Internet Engineering Task Force (IETF)
Request for Comments: 9182
Category: Standards Track
ISSN: 2070-1721

S. Barguil
O. Gonzalez de Dios, Ed.
Telefonica
M. Boucadair, Ed.
Orange
L. Munoz
Vodafone
A. Aguado
Nokia

February 2022

A YANG Network Data Model for Layer 3 VPNs

Модель данных YANG для L3 VPN

Аннотация

В качестве дополнения к модели услуг виртуальных частных сетей L3 (Layer 3 Virtual Private Network Service Model или L3SM), применяемой для взаимодействия между клиентами и сервис-провайдерами, этот документ задаёт модель сети L3VPN (L3VPN Network Model или L3NM), которая может служить для предоставления услуг виртуальных частных сетей L3 (Layer 3 Virtual Private Network или L3VPN) внутри сети сервис-провайдера. Эта модель обеспечивает сетецентричное представление услуг L3VPN.

Модель L3NM предназначена для использования сетевыми контроллерами для вывода конфигурационной информации, которая будет передаваться вовлечённым сетевым устройствам. Модель также может облегчить связь между оркестратором (организатором) служб и контроллером/оркестратором сети.

Статус документа

Документ относится к категории Internet Standards Track.

Документ является результатом работы IETF[1] и представляет согласованный взгляд сообщества IETF. Документ прошёл открытое обсуждение и был одобрен для публикации IESG[2]. Дополнительную информацию о стандартах Internet можно найти в разделе 2 в RFC 7841.

Информация о текущем статусе документа, найденных ошибках и способах обратной связи доступна по ссылке https://www.rfc-editor.org/info/rfc9182.

Оглавление

[1]Internet Engineering Task Force - комиссия по решению инженерных задач Internet.
[2]Internet Engineering Steering Group - комиссия по инженерным разработкам Internet.

1. Введение

В [RFC8299] определена модель YANG L3SM, которая может применяться для взаимодействия между клиентами и поставщиками услуг. Модель сосредоточена на описании точки зрения клиента на услуги виртуальной частной сети (Virtual Private Network или VPN) и обеспечивает абстрактное представление запрашиваемых клиентом услуг. Такой подход ограничивает применение L3SM ролью модели обслуживания клиентов (согласно [RFC8309]).

Данный документ определяет модуль YANG названный моделью сети L3VPN (L3VPN Network Model или L3NM). Модель L3NM предназначена для сетецентричного представления услуг L3 VPN. Модель может применяться для упрощения взаимодействий между организатором (оркестратором) службы и сетевым контроллером/оркестратором, позволяя включать больше сетецентрических сведений. Модель обеспечивает дополнительные возможности, такие как управление ресурсами, или служит многодоменным интерфейсом оркестровки, где должны согласовываться логические ресурсы (такие как цели или различители маршрутов).

Этот документ использует базовый модуль YANG VPN, определённый в [RFC9181].

Этот документ не отменяет [RFC8299]. Оба модуля применяются для схожих целей, но с разными областями действия и представлениями. Модуль YANG L3NM был создан на основе подхода «упрощать и расширять» (prune and extend) к послужившему базой модулю YANG из [RFC8299]. Тем не менее, L3NM не задаёт дополнений к L3SM, поскольку требуется конкретная структура для удовлетворения потребностей L3.

Некоторые сведения, зафиксированные в L3SM, могут передаваться оркестратором в L3NM (например, клиент) или применяться для передачи отдельных атрибутов L3NM (например, фактических правил пересылки). Кроме того, такие сведения могут поддерживаться локально в оркестраторе, который отвечает за поддержку сопоставления между представлением клиента и его экземпляром сети. Некоторые сведения зафиксированные и раскрываемые с помощью L3NM, могут передаваться уровню службы (например, возможности) для управления обработкой заказов на услуги VPN и, следовательно, службой L3SM.

В параграфе 5.1 [RFC8969] показано, как можно использовать L3NM в архитектуре автоматизации управления сетью.

L3NM не пытается охватить все варианты развёртывания, особенно те, где связность L3VPN поддерживается путём координации разных VPN в разных базовых сетях. Более сложные варианты внедрения включают координацию разных экземпляров VPN и различных технологий для обеспечения сквозной связности VPN, решаются с помощью дополнительных модулей YANG, например, [YANG-Composed-VPN].

Модель L3NM сосредоточена на L3 VPN, основанных на BGP PE, как описано в [RFC4026], [RFC4110] и [RFC4364], а также Multicast VPN, описанных в [RFC6037] и [RFC6513].

Модель данных YANG в этом документе соответствует архитектуре хранилищ данных управления сетью (Network Management Datastore Architecture или NMDA), заданной в [RFC8342].

2. Терминология

Ключевые слова **должно** (MUST), **недопустимо** (MUST NOT), **требуется** (REQUIRED), **нужно** (SHALL), **не следует** (SHALL NOT), **следует** (SHOULD), **не нужно** (SHOULD NOT), **рекомендуется** (RECOMMENDED), **не рекомендуется** (NOT RECOMMENDED), **возможно** (MAY), **необязательно** (OPTIONAL) в данном документе интерпретируются в соответствии с BCP 14 [RFC2119] [RFC8174] тогда и только тогда, когда они выделены шрифтом, как показано здесь.

Этот документ предполагает знакомство читателя с [RFC6241], [RFC7950], [RFC8299], [RFC8309], [RFC8453] и использует определённые в этих документах термины. Документ использует термин «модель сети» (network model) в соответствии с параграфом 2.1 в [RFC8969]. На диаграммах деревьев применяется нотация [RFC8340].

Ниже приведены термины, определённые в этом документе.

Layer 3 VPN Service Model (L3SM) - модель услуг L3SM

Модель данных YANG, описывающая требования сервиса L3VPN, объединяющего набор сайтов, с точки зрения клиента. Клиентская модель сервиса не содержит деталей сети сервис-провайдера. Клиентская модель сервиса L3VPN определена в [RFC8299].

Layer 3 VPN Network Model (L3NM) - модель сети L3NM
Модель данных YANG, описывающая службу VPN в сети сервис-провайдера. Модель содержит сведения о сети провайдера и может включать выделенные ресурсы. Модель могут применять контроллеры сети для управления и поддержки конфигурации услуг VPN в сети сервис-провайдера. Соответствующий модуль YANG может применяться организатором услуг для запроса услуги VPN у контроллера сети.

Service orchestrator - оркестратор (организатор) услуг
Функциональный объект, взаимодействующий с клиентом L3VPN через L3SM. Организатор отвечает за устройства присоединения CE к PE (CE-PE), выбор PE и запросы услуг VPN у сетевого контроллера.

Network orchestrator - сетевой оркестратор (организатор)
Функциональный объект, размещённый в иерархии между оркестратором сервиса и сетевыми контроллерами. Оркестратор сети может управлять одним или несколькими сетевыми контроллерами.

Network controller - сетевой контроллер
Функциональный объект, отвечающий за управление и поддержку сети сервис-провайдера.

VPN node - узел VPN
Абстракция, представляющая набор применяемых на PE правил, относящихся к одной службе VPN. Служба VPN включает один или множество узлов VPN. Поскольку это абстракция, способ реализации узла VPN выбирает контроллер сети. Например, в VPN на основе BGP узел VPN обычно может отображаться на экземпляр виртуальной маршрутизации и пересылки (Virtual Routing and Forwarding или VRF).

VPN network access - доступ в сеть VPN
Абстракция, представляющая сетевые интерфейсы, связанные с данным узлом VPN. Трафик, поступающий от доступа в VPN, относится к VPN. Устройства присоединения (bearer - носитель, канал) между CE и PE завершаются доступом в сеть VPN. Поддерживается ссылка на носитель для сохранения канала между L3SM и L3NM, если в данном развёртывании применяются обе модели.

VPN site - сайт VPN
Местоположение клиента VPN, соединённое с сетью сервис-провайдера через канал CE-PE, у которого может быть доступ хотя бы в одну сеть VPN [RFC4176].

VPN service provider - поставщик услуг VPN
Сервис-провайдер, обеспечивающий связанные с VPN услуги [RFC4176].

Service provider network - сеть сервис-провайдера
Сеть, способная предоставлять связанные с VPN услуги.

Этот документ предназначен для моделирования BGP VPN на основе PE в сети сервис-провайдера, поэтому в нем применяются термины, определённые в [RFC4026] и [RFC4176].

3. Сокращения

ACL
Access Control List - список управления доступом.

AS
Autonomous System - автономная система.

ASM
Any-Source Multicast - групповая передача Any-Source.

ASN
AS Number - номер автономной системы.

BFD
Bidirectional Forwarding Detection - обнаружение двухсторонней пересылки.

BGP
Border Gateway Protocol - протокол граничного шлюза (междоменной маршрутизации).

BSR
Bootstrap Router - маршрутизатор начальной загрузки.

CE
Customer Edge - граница клиента.

CsC
Carriers' Carriers - операторы для операторов.

IGMP
Internet Group Management Protocol - протокол управления группами Internet.

L3NM
L3VPN Network Model - модель сети L3VPN.

L3SM
L3VPN Service Model - модель службы L3VPN.

L3VPN
Layer 3 Virtual Private Network - виртуальная частная сеть L3.

MLD
Multicast Listener Discovery - обнаружение получателей группового трафика.

MSDP
Multicast Source Discovery Protocol - протокол обнаружения получателей группового трафика.

MVPN
Multicast VPN - групповая VPN.

NAT
Network Address Translation - трансляция сетевых адресов.

OAM
Operations, Administration, and Maintenance - операции, администрирование, управление (поддержка)

OSPF
Open Shortest Path First - сначала кратчайший путь.

PE
Provider Edge - граница провайдера.

PIM
Protocol Independent Multicast - независимая от протокола групповая передача.

QoS
Quality of Service - качество обслуживания.

RD
Route Distinguisher - отличитель маршрута.

RP
Rendezvous Point - точка встречи.

RT
Route Target - цель маршрута.

SA
Security Association - защищённая связь.

SSM
Source-Specific Multicast - зависящая от источника групповая передача.

VPN
Virtual Private Network - виртуальная частная сеть.

VRF
Virtual Routing and Forwarding - виртуальная маршрутизация и пересылка.

4. Эталонная архитектура L3NM

На рисунке 1 показана эталонная архитектура L3NM. Рисунок является расширением архитектуры, представленной в разделе 5 [RFC8299] и разбивает поле оркестровки (orchestration) на 3 части - организация службы, организация сети и организация домена.

Хотя в некоторых развёртываниях может быть выбрана монолитная оркестровка (охватывающая службу и сеть), этот документ выступает за чёткое разделение между компонентами организации службы и сети для большей гибкости. Такое устройство соответствует эталонной архитектуре L3VPN, определённой в параграфе 1.3 [RFC4176]. Разделение основано на выделенном коммуникационном интерфейсе между компонентами и соответствующими модулями YANG, которые отражают связанную с сетью информацию. Эти сведения скрыты от клиентов.

Интеллектуальные средства трансляции предназначенных для клиента сведений в ориентированную на сеть информацию (и обратно) зависят от реализаци.

Здесь применяется терминология из [RFC8309], чтобы показать разницу между моделью обслуживания клиентов, моделью предоставления услуг и моделью конфигурации устройства. В этом контексте роли организации домена и управления конфигурацией могут исполнять контроллеры.

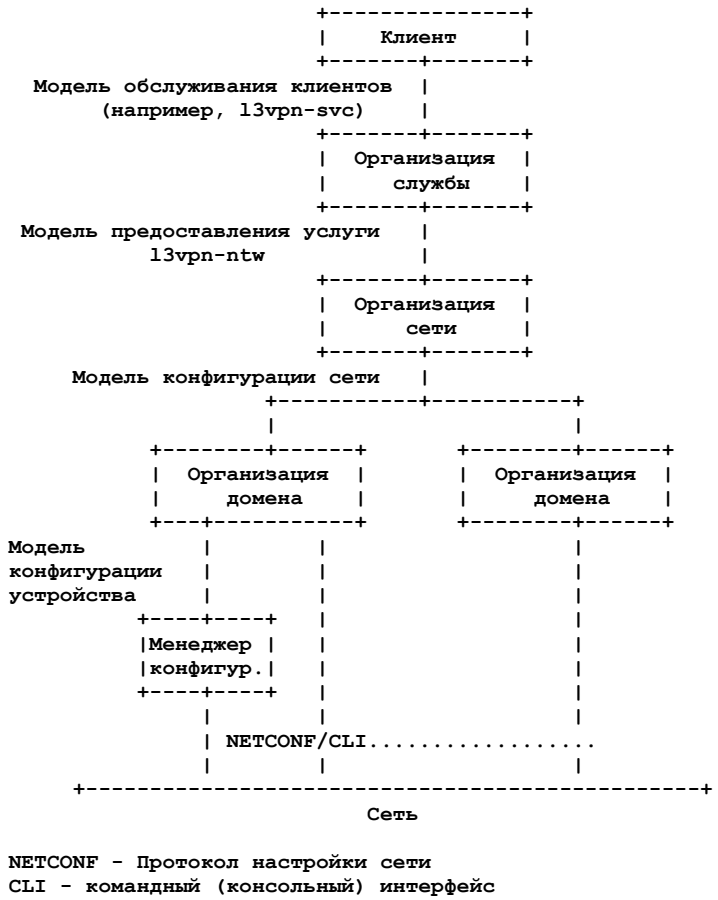

Рисунок 1. Эталонная архитектура L3NM.

Клиент может применять разные средства для запроса услуги, которая может вызывать создание экземпляра L3NM. Клиент может пользоваться L3SM или более абстрактными моделями для запроса услуг, основанных на сервисе L3VPN. Например, клиент может представить профиль обеспечения связности IP (IP Connectivity Provisioning Profile или CPP) с характеристиками запрашиваемых услуг [RFC7297], расширенных услуг VPN (VPN+) [Enhanced-VPN-Framework] или услуг сетевого слоя IETF (network slice) [Network-Slices-Framework].

Отметим, что можно использовать L3SM и L3NM в контексте модели абстракции и управления сетями TE (Abstraction and Control of TE Networks или ACTN) [RFC8453]. На рисунке 2 показан клиентский контроллер сети (Customer Network Controller или CNC), многодоменный координатор услуг (Multi-Domain Service Coordinator или MDSC), обеспечивающий сетевой контроллер (Provisioning Network Controller или PNC) и интерфейсы, где применяется L3SM и L3NM.

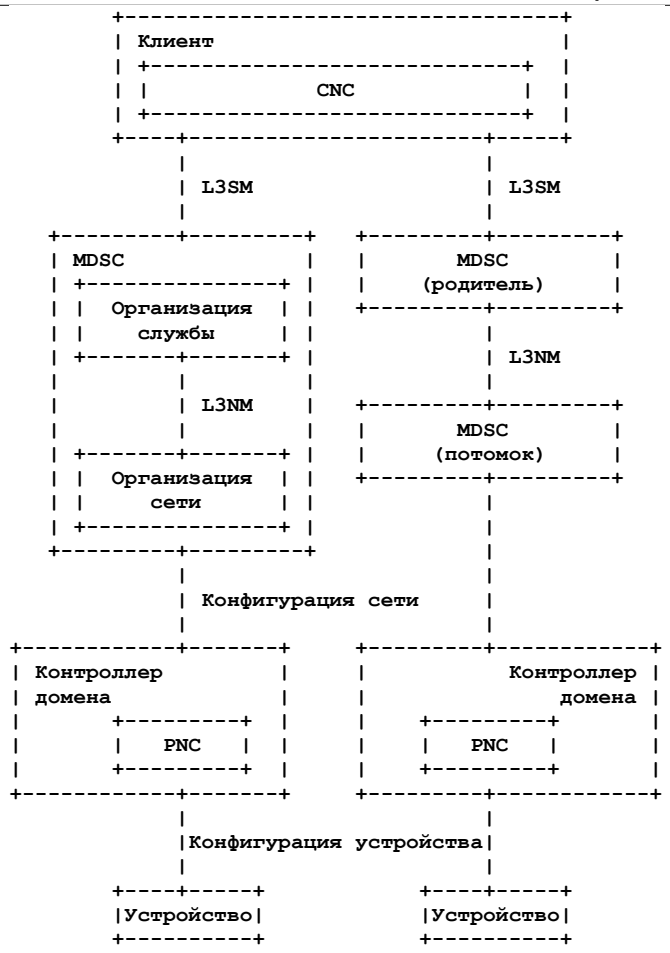

Рисунок 2. L3SM и L3NM в контексте ACTN.

5. Связи с другими моделями данных YANG

Модуль ietf-vpn-common [RFC9181] включает набор идентификаторов, типов и группироваок, предназначенных для использования в связанных с VPN модулях YANG независимо от уровня (например, L2, L3) и типа модуля (например, модель сети или службы), включая будущие выпуски имеющихся моделей (например, [RFC8299] или [RFC8466]). L3NM использует эти базовые типы и группировки.

Чтобы избежать дублирования данных и упростить их передачу между уровнями, когда это требуется (от уровня сервиса на сетевой и обратно), в ранних вресиях L3NM применяется много узлов данных из [RFC8299]. Тем не менее, от такого подхода отказались в пользу модуля ietf-vpn-common, поскольку исходное решение интерпретировалось как внедрение L3NM в зависимости от L3SM, а это не так. Например, сервис-провайдер может использовать L3NM для создания своих служб L3VPN без раскрытия L3SM.

Как отмечено в разделе 4, модель L3NM предназначена для управления услугами L3VPN в сети сервис-провайдера. Модуль обеспечивает сетевое представление сервиса, которое видимо лишь внутри сети провайдера и не раскрывается наружу (например, клиентам). Приведенные ниже элементы показывают интерфейсы L3NM с другими модулями YANG.

L3SM

L3NM не является моделью обслуживания клиентов. Внутреннее представление службы (т. е., L3NM) может отображаться на внешнее, которое доступно клиентам - модель службы L3VPN (L3SM) [RFC8299].

В L3NM можно вводить данные из запроса клиента. Такие запросы обычно основаны на шаблоне L3SM. В частности, некоторые части модуля L3SM можно напрямую сопоставить с L3NM, хотя другие части генерируются в зависимости от запрошенной услуги и местных правил. Некоторые части являются локальными для сервис-провайдера и не отображаются напрямую в L3SM.

Отметим, что использование L3NM внутри сервис-провайдера не предполагает и не запрещает раскрытие сервиса VPN через L3SM (это зависит от развертывания). Тем не менее, L3NM пытается максимально согласоваться с функциями, поддерживаемыми L3SM, для упрощения использования L3NM и L3SM ради автоматизированного предоставления услуг VPN.

Модули топологии сети

L3VPN включает узлы, являющиеся частью топологии, поддерживаемой сетью сервис-провайдера. Топологию можно представить с помощью топологического модуля YANG из [RFC8345] или его расширения, такого как модуль YANG для точек подключения к сервису (Service Attachment Point или SAP) [YANG-SAPs].

Модули устройств

L3NM не является моделью устройства.

Как только глобальная служба VPN зафиксирована с помощью L3NM, фактическая активация и предоставление услуг VPN будет включать модули устройств для настройки требуемых функций предоставления услуг. Эти функции поддерживаются узлами VPN и могут управляться с применением модулей YANG для устройств. Список некоторых модулей YANG для устройств приведён ниже.

- Управление маршрутизацией [RFC8349].
- BGP [BGP-YANG].
- PIM [PIM-YANG].

- Поддержка NAT [RFC8512].
- Поддержка QoS [QoS-YANG].
- ACL [RFC8519].

Использование L3NM для вывода зависящих от устройств действий зависит от реализации.

6. Пример использования модели данных L3NM

В этом разделе представлены некоторые примеры, иллюстрирующие контекст примерения L3NM.

6.1. Корпоративные услуги L3 VPN

Корпоративные сети L3VPN являются одними из наиболее распространённых служб и модель L3NM может быть полезна для автоматизации предоставления и поддержки таких VPN. Можно создать шаблоны и пакетные процессы, в результате чего многие параметры, которые для VPN нужно создавать с нуля, можно абстрагировать на верхний уровень программно определяемой сети (Software-Defined Networking или SDN) [RFC7149] [RFC7426], но некоторая часть ручной работы сохранится.

Общей функцией VPN является добавление и удаление узлов VPN. В рабочих процессах для этого можно использовать L3NM при добавлении и удалении узлов из модели данных сети по мере необходимости.

6.2. Управление ресурсами нескольких доменов

Реализация услуг L3VPN, охватывающих разные административные домены (т. е., администрируемые разными системами управления или контроллерами), требует синхронизации сетевых ресурсов между системами. В частности, ресурсами нужно адекватно управлять в каждом домене, чтобы не возникало неверных конфигураций. Например, нужно синхронизировать цели маршрутов (RT) между устройствами PE. Когда все PE контролируются одной системой управления, эта же система может управлять назначением RT. Если служба охватывает несколько систем управления, назначение RT нужно согласовывать между доменами, поэтому модель сети должна обеспечивать способ указания RT. Кроме того, нужно синхронизировать отличители маршрутов (RD), чтобы избежать конфликтов RD между разными системами управления. Некорректное выделение может привести к экспорту одних RD и префиксов IP разными PE.

6.3. Управление групповыми службами

Групповые услуги можно реализовать через L3VPN с использованием двойных PIM MVPN (модель draft-rosen) [RFC6037] или MVPN на основе Multiprotocol BGP (MP-BGP) [RFC6513] [RFC6514]. Оба метода поддерживаются и одинаково эффективны, главное различие состоит в том, что MVPN на основе MP-BGP не требуют настройки групповой передачи в сети сервис-провайдера. MP-BGP MVPN реализуют плоскость управления BGP внутри AS и PIM Sparse Mode [RFC7761] в качестве плоскости данных. Сведения о состоянии PIM передаются между PE с использованием той же архитектуры, которая применяется для unicast VPN.

Решение [RFC6037] имеет ограничения в таких аспектах, как возможности для транспорта, расширяемость плоскости управления, доступность, эксплуатационная согласованность, и требует поддержки состояния в магистрали. Из-за этих ограничений в качестве базовой архитектурной модели для реализации групповых услуг в L3VPN была выбрана модель MP-BGP MVPN. В этом варианте BGP служит для автоматического обнаружения PE, входящих в MVPN, а клиентская сигнализация PIM передаётся через ядро провайдера с помощью MP-BGP. Групповой трафик доставляется по путям P2MP[1] LSP[2].

7. Описание модуля YANG L3NM

Модуль L3NM (ietf-l3vpn-ntw) определён для управления L3VPN в сети сервис-провайдера и может применяться, в частности, для создания, изменения и нахождения услуг L3VPN в сети.

Полное дерево модуля можно сгенерировать с помощью pyang [PYANG]. Это дерево не показано здесь из-за большого размера (параграф 3.3 в [RFC8340]) и для удобства в документе приведены отдельные ветви.

7.1. Общая структура модуля

Модуль ietf-l3vpn-ntw использует два основных контейнера - vpn-profiles и vpn-services (Рисунок 3). Контейнер vpn-profiles используется провайдером для поддержки набора базовых профилей VPN, применяемых для одной или нескольких служб VPN (параграф 7.2). Контейнер vpn-services содержит набор служб VPN, поддерживаемых в сети сервис-провайдера. Структура данных vpn-service абстрагирует службы VPN (параграф 7.3).

```
module: ietf-l3vpn-ntw
  +--rw l3vpn-ntw
     +--rw vpn-profiles
     |  ...
     +--rw vpn-services
        +--rw vpn-service* [vpn-id]
           ...
           +--rw vpn-nodes
              +--rw vpn-node* [vpn-node-id]
                 ...
                 +--rw vpn-network-accesses
                    +--rw vpn-network-access* [id]
                       ...
```

Рисунок 3. Общая структура дерева L3NM.

Некоторые узлы данных связаны с семейством адресов. Для компактного представления данных, имеющих одно значение для IPv4 и IPv6, **рекомендуется** использовать семейство адресов двойного стека. Если узел данных представлен для двойного стека и IPv4 (или IPv6), значение для двойного стека имеет приоритет.

[1]MPLS Point-to-Multipoint - MPLS «один со многими».
[2]Label Switched Path - путь с коммутацией по меткам.

7.2. Профили VPN

Контейнер vpn-profiles (Рисунок 4) позволяет поставщику услуг VPN задать и поддерживать набор профилей VPN [RFC9181], применяемых для одной или нескольких служб VPN.

```
+--rw l3vpn-ntw
   +--rw vpn-profiles
   |  +--rw valid-provider-identifiers
   |     +--rw external-connectivity-identifier* [id]
   |     |       {external-connectivity}?
   |     |  +--rw id    string
   |     +--rw encryption-profile-identifier* [id]
   |     |  +--rw id    string
   |     +--rw qos-profile-identifier* [id]
   |     |  +--rw id    string
   |     +--rw bfd-profile-identifier* [id]
   |     |  +--rw id    string
   |     +--rw forwarding-profile-identifier* [id]
   |     |  +--rw id    string
   |     +--rw routing-profile-identifier* [id]
   |        +--rw id    string
   +--rw vpn-services
      ...
```

Рисунок 4. Структура ветви профилей VPN.

В этом документе не принимается допущений о точных определениях этих профилей. Такие определения локально задаёт каждый сервис-провайдер VPN. Модель лишь включает идентификатор для таких профилей, чтобы упростить их указание и привязку к локальным правилам при создании службы VPN. Список идентификаторов приведён ниже.

external-connectivity-identifier
: Указывает профиль, определяющий внешнюю связность для службы VPN (или подмножества сайтов VPN). Внешняя связность может быть доступом в Internet или ограниченным подключением, например, к облаку.

encryption-profile-identifier
: Указывает набор правил, относящихся к схемам и организации шифрования, которые могут применяться при создании и предоставлении услуги VPN.

qos-profile-identifier
: Указывает набор правил QoS, таких как классификация, маркировка и действия (например, [RFC3644]).

bfd-profile-identifier
: Указывает набор правил обнаружения двухсторонней пересылки (BFD) [RFC5880], которые можно вызывать при создании службы VPN.

forwarding-profile-identifier
: Указывает правила, которые применяются при пересылке пакетов, предаваемых внутри VPN. Такие правила могут включать, например, списки управления доступом (ACL).

routing-profile-identifier
: Указывает набор правил маршрутизации (например, правил BGP) которые будут применяться для предоставления услуг VPN.

7.3. Услуги VPN

Структура данных vpn-service абстрагирует службу VPN в сети сервис-провайдера. Каждая структура vpn-service однозначно указывается vpn-id, имеющим локальную значимость (например, в контроллере сети). Ветви vpn-services показана на рисунке 5.

```
+--rw l3vpn-ntw
   +--rw vpn-profiles
   |  ...
   +--rw vpn-services
      +--rw vpn-service* [vpn-id]
         +--rw vpn-id                    vpn-common:vpn-id
         +--rw vpn-name?                 string
         +--rw vpn-description?          string
         +--rw customer-name?            string
         +--rw parent-service-id?        vpn-common:vpn-id
         +--rw vpn-type?                 identityref
         +--rw vpn-service-topology?     identityref
         +--rw status
         |  +--rw admin-status
         |  |  +--rw status?          identityref
         |  |  +--rw last-change?   yang:date-and-time
         |  +--ro oper-status
         |     +--ro status?          identityref
         |     +--ro last-change?   yang:date-and-time
         +--rw vpn-instance-profiles
         |  ...
         +--rw underlay-transport
         |  +-- (type)?
         |     +--:(abstract)
         |     |  +--rw transport-instance-id?   string
         |     |  +--rw instance-type?           identityref
         |     +--:(protocol)
         |        +--rw protocol*                identityref
         +--rw external-connectivity
         |                    {vpn-common:external-connectivity}?
         |  +--rw (profile)?
         |     +--:(profile)
```

```
            |        +--rw profile-name?            leafref
            +--rw vpn-nodes
               ...
```

Рисунок 5. Структура ветви службы VPN.

Узлы данных, показанные на рисунке 5, перечислены ниже.

vpn-id
Идентификатор, однозначно указывающий службу L3VPN в области действия L3NM.

vpn-name
Имя службы для упрощения её идентификации.

vpn-description
Текстовое описание службы, внутреннюю структуру которого задаёт сервис-провайдер.

customer-name
Имя клиента, заказавшего услугу.

parent-service-id
Идентификатор родительской службы (например, L3SM, сетевой срез IETF, VPN+), вызвавшей создание услуги VPN. Этот идентификатор служит для упрощения сопоставления (сетевой) услуги, встроенной в сеть с заказом на услугу. Контроллер может использовать такое сопоставление для заполнения некоторых полей (например, описаний) в зависимости от локального развёртывания.

vpn-type
Указывает тип VPN значением мз [RFC9181]. Для L3NM это обычно BGP/MPLS L3VPN, но могут быть определены иные значения для поддержки конкретных возможностей L3 VPN (например, [RFC9136]).

vpn-service-topology
Указывает топологию для службы - hub-spoke, any-to-any, custom. Реализация этого атрибута в сети определяется корректным использованием целей импорта и экспорта (параграф 4.3.5 в [RFC4364]).

status
Служит для отслеживания состояния данной услуги VPN. Поддерживается административный и рабочий статус с указанием временной метки. Например, услуга может быть создана, но не использовтаься. Административный и рабочий статус могут применяться для обнаружения аномалий службы. Например, служба указана как активная на уровне сервиса, но остаётся неактивной на уровне сети, что может указывать необходимость действий по согласованию наблюдаемого состояния с ожидаемым.

vpn-instance-profiles
Задаёт многократно используемые параметры для vpn-service (см. 7.4. Профили экземпляров VPN).

underlay-transport
Описывает предпочтения для выбора транспортной технологии для передачи трафика VPN. Это особенно полезно в сетях с несколькими доменами и типами межсетевых интерфейсов (Network-to-Network Interface или NNI). Базовый транспорт можно указывать экземпляром абстрактного транспорта (например, идентификатором экземпляра VPN+ или виртуальной сети, именем сетевого среза) или упорядоченным списком фактических протоколов, применяемых в сети. Набор идентификаторов протоколов для указания транспорта задан в [RFC9181].

external-connectivity
Указывает возможность и способ внешних подключений для услуги VPN. Например, сервис-провайдер может предоставлять внешнюю связность клиенту VPN (например, к общедоступному облаку). Такая услуга может включать настройку правил фильтрации и NAT (например, привязку интерфейса VRF к экземпляру NAT, как описано в параграфе 2.10 [RFC8512]). Эти добавочные свойства можно связать со всем или частью доступа в сеть. Некоторые из таких свойств могут быть реализованы в PE или иных узлах (например, в узле P или даже на выделенном узле, обеспечивающем функции NAT).
В этом документе поддерживается лишь указатель на локальный профиль, определяющий внешнюю связность.

vpn-node
Абстракция набора правил, применяемых к узлу сети и относящихся к одному vpn-service. Услуга VPN обычно создаётся путём добавления экземпляров vpn-node в контейнер vpn-nodes. В vpn-node содержатся vpn-network-accesses - интерфейсы, соединённые с VPN, через которые принимается трафик клиентов. Поэтому сайты клиента соединяются с vpn-network-accesses. Поскольку эта модель является моделью сети, в ней не требуется сведений о сайтах клиентов. Такая информация относится скорей к L3SM, а её включение в L3NM, например, для заполнения узлов описания (description), определяется реализацией. Дополнительные сведения приведены в параграфе 7.5.

7.4. Профили экземпляров VPN

Профили экземпляров VPN предназначены для факторизации узлов данных, используемых на многих уровнях модели. Базовые профили экземпляров VPN определяются на уровне сервиса VPN, а затем вызываются на уровне узла VPN и доступа в сеть VPN. Каждый профиль экземпляра VPN указывается profile-id. Этот идентификатор используется для одного или нескольких узлов VPN (параграф 7.5), чтобы контроллер мог идентифицировать базовые ресурсы (например, RT и RD), которые нужно настроить для данного экземпляра VRF. Ветвь vpn-instance-profiles показана на рисунке 6.

```
   +--rw l3vpn-ntw
      +--rw vpn-profiles
      |  ...
      +--rw vpn-services
         +--rw vpn-service* [vpn-id]
            +--rw vpn-id                    vpn-common:vpn-id
            ...
            +--rw vpn-instance-profiles
            |  +--rw vpn-instance-profile* [profile-id]
            |     +--rw profile-id                 string
            |     +--rw role?                      identityref
            |     +--rw local-as?                  inet:as-number
            |     |      {vpn-common:rtg-bgp}?
            |     +--rw (rd-choice)?
            |     |  +--:(directly-assigned)
            |     |  |  +--rw rd?
```

```
|      |  |         rt-types:route-distinguisher
|      |  +--:(directly-assigned-suffix)
|      |  |  +--rw rd-suffix?           uint16
|      |  +--:(auto-assigned)
|      |  |  +--rw rd-auto
|      |  |     +--rw (auto-mode)?
|      |  |     |  +--:(from-pool)
|      |  |     |  |  +--rw rd-pool-name?   string
|      |  |     |  +--:(full-auto)
|      |  |     |     +--rw auto?           empty
|      |  |     +--ro auto-assigned-rd?
|      |  |           rt-types:route-distinguisher
|      |  +--:(auto-assigned-suffix)
|      |  |  +--rw rd-auto-suffix
|      |  |     +--rw (auto-mode)?
|      |  |     |  +--:(from-pool)
|      |  |     |  |  +--rw rd-pool-name?        string
|      |  |     |  +--:(full-auto)
|      |  |     |     +--rw auto?               empty
|      |  |     +--ro auto-assigned-rd-suffix?   uint16
|      |  +--:(no-rd)
|      |     +--rw no-rd?               empty
|      +--rw address-family* [address-family]
|      |  +--rw address-family          identityref
|      |  +--rw vpn-targets
|      |  |  +--rw vpn-target* [id]
|      |  |  |  +--rw id                   uint8
|      |  |  |  +--rw route-targets* [route-target]
|      |  |  |  |  +--rw route-target
|      |  |  |  |       rt-types:route-target
|      |  |  |  +--rw route-target-type
|      |  |  |          rt-types:route-target-type
|      |  |  +--rw vpn-policies
|      |  |     +--rw import-policy?   string
|      |  |     +--rw export-policy?   string
|      |  +--rw maximum-routes* [protocol]
|      |     +--rw protocol          identityref
|      |     +--rw maximum-routes?   uint32
|      +--rw multicast {vpn-common:multicast}?
|         ...
```

Рисунок 6. Структура ветви профилей экземпляров VPN.

Описания узлов данных приведены ниже.

profile-id

Служит для однозначного указания профиля экземпляра VPN.

role

Указывает роль профиля экземпляра VPN в VPN. Значения ролей заданы в [RFC9181] (например, any-to-any-role, spoke-role, hub-role).

local-as

Указывает номер автономной системы (Autonomous System Number или ASN), настроенный для узла VPN.

rd

Как указано в [RFC9181], поддерживается несколько режимов назначения RD: прямое назначение, полностью автоматическое назначение, автоматическое назначение из заданного пула и отсутствие назначения. В иллюстративных целях можно использовать указанные ниже режимы в вариантах развёртывания.

directly-assigned

Поставщик услуг VPN (организатор сервиса) явно назначает RD. Это соответствует случаю, когда провайдеру VPN нужно обновить некоторые имеющиеся службы.

full-auto

Контроллер сети автоматически назначает RD. Это подходит для развёртывания новых служб.

no-rd

Поставщик услуг VPN (организатор сервиса) явно хочет не назначать RD. Это может применяться для тестирования CE в сети и устранения неполадок.

Кроме того, модуль поддерживает развёртывания, где в RD назначается лишь поле Assigned Number (параграф 4.2 в [RFC4364]) из пула, а в поле Administrator помещается, например, Router ID маршрутизатора, назначенного узлу VPN. Модуль поддерживает для управления полем Assigned Number режимы явного назначения, автоматического назначения из пула и полностью автоматического назначения.

address-family

Набор узлов данных по семействам адресов.

address-family

Указывает семейство адресов (ipv4, ipv6, dual-stack).

vpn-targets

Задаёт правила импорта и экспорта RT для службы VPN (параграф 4.3 в [RFC4364]).

maximum-routes

Максимальное число префиксов, которые узел VPN может воспринять для данного протокола маршрутизации. Если protocol имеет значение any, это указывает применимость заданного максимума ко всем активным протоколам маршрутизации.

multicast

Включает поддержку группового трафика в VPN (см. 7.7. Групповая передача).

7.5. Узлы VPN

Абстракция vpn-node представляет набор общих правил, применяемых на данном узле сети (обычно PE) и относящихся к одной службе L3VPN. В vpn-node включён параметр для индикации узла сети, где эти правила применяются. Если ne-id указывает конкртеный PE, vpn-node, скорей всего, будет отображён на экземпляр VRF в этом узле. Однако модель позволяет указывать и абстрактный узел и в этом случае сетевой контроллер будет определять расщепление vpn-node по экземплярам VRF. Структура ветви для узла VPN показана на рисунке 7.

```
+--rw l3vpn-ntw
   +--rw vpn-profiles
   |  ...
   +--rw vpn-services
      +--rw vpn-service* [vpn-id]
         ...
         +--rw vpn-nodes
            +--rw vpn-node* [vpn-node-id]
               +--rw vpn-node-id                vpn-common:vpn-id
               +--rw description?               string
               +--rw ne-id?                     string
               +--rw local-as?                  inet:as-number
               |       {vpn-common:rtg-bgp}?
               +--rw router-id?                 rt-types:router-id
               +--rw active-vpn-instance-profiles
               |  +--rw vpn-instance-profile* [profile-id]
               |     +--rw profile-id                 leafref
               |     +--rw router-id* [address-family]
               |     |  +--rw address-family    identityref
               |     |  +--rw router-id?        inet:ip-address
               |     +--rw local-as?            inet:as-number
               |     |     {vpn-common:rtg-bgp}?
               |     +--rw (rd-choice)?
               |     |  ....
               |     +--rw address-family* [address-family]
               |     |  +--rw address-family          identityref
               |     |  |  ...
               |     |  +--rw vpn-targets
               |     |  |  ...
               |     |  +--rw maximum-routes* [protocol]
               |     |     ...
               |     +--rw multicast {vpn-common:multicast}?
               |        ...
               +--rw msdp {msdp}?
               |  +--rw peer?            inet:ipv4-address
               |  +--rw local-address?   inet:ipv4-address
               |  +--rw status
               |     +--rw admin-status
               |     |  +--rw status?         identityref
               |     |  +--rw last-change?   yang:date-and-time
               |     +--ro oper-status
               |        +--ro status?         identityref
               |        +--ro last-change?   yang:date-and-time
               +--rw groups
               |  +--rw group* [group-id]
               |     +--rw group-id    string
               +--rw status
               |  +--rw admin-status
               |  |  +--rw status?         identityref
               |  |  +--rw last-change?   yang:date-and-time
               |  +--ro oper-status
               |     +--ro status?         identityref
               |     +--ro last-change?   yang:date-and-time
               +--rw vpn-network-accesses
                  ...
```

Рисунок 7. Структура ветви узла VPN.

Узлы данных vpn-node перечислены ниже.

vpn-node-id
Идентификатор, однозначно указывающий узел, разрешающий доступ в сеть VPN.

description
Текстовое описание узла VPN.

ne-id
Уникальный идентификатор элемента сети, где развернут узел VPN.

local-as
Номер автономной системы (ASN), заданный для узла VPN.

router-id
32-битовый уникальный идентификатор маршрутизатора в AS.

active-vpn-instance-profiles
Список активных профилей экземпляров VPN для этого узла VPN. Один или несколько профилей экземпляров VPN, определённых на уровне службы VPN, могут быть разрешены на уровне узла VPN и каждый из этих профилей однозначно указывается profile-id. Структура active-vpn-instance-profiles совпадает с описанной в параграфе 7.4. Профили экземпляров VPN, за исключением того, что active-vpn-instance-profiles включает router-id и не имеет листа role. Значение router-id в active-vpn-instance-profiles имеет предпочтение перед router-id в vpn-node для указанного семейства адресов. Например, Router ID могут быть настроены по семействам адресов. Это

свойство может применяться, например, для настройки адреса IPv6 как Router ID, когда такая возможность поддерживается вовлеченными маршрутизаторами. Значения в active-vpn-instance-profiles переопределяют заданные на уровне сервиса VPN, см. пример в A.3. Переопределение параметров профиля экземпляра VPN.

msdp

Для резервирования может быть включён протокол обнаружения групповых источников (Multicast Source Discovery Protocol или MSDP) [RFC3618], используемый для общего доступа к сведениям об источниках разных точек встречи (Rendezvous Point или RP). В этом контексте цель MSDP заключается в повышении отказоустойчивости групповой услуги. MSDP можно настроить на маршрутизаторах, не являющихся RP, это полезно в доменах, не поддерживающих групповые источники, но разрешающих транзит группового трафика.

groups

Список групп, в которые входит узел VPN [RFC9181]. Например, group-id служит для связывания ограничений резервирования или защиты с узлами VPN.

status

Отслеживает рабочий и административный статус узла, вовлечённого в сервис VPN. Несовпадение административного и оперативного статуса может применяться для поиска аномалий.

vpn-network-accesses

Представляет точку, к которой присоединяются сайты.

Отметим, что в отличие от L3SM, в L3NM не нужно моделировать сайт клиента, достаточно точек, получающих трафик с сайтов (т. е. сторона PE соединений PE-CE между клиентом и провайдером). Поэтому для доступа в сеть VPN указывается соединение сети провайдера с помещением клиента. Профили VPN (vpn-profiles) включают наборы правил маршрутизации, которые могут применяться при организации услуги.

См. также параграф 7.6.

7.6. Доступ в сеть VPN

Контейнер vpn-network-access включает набор узлов данных со сведениями относительно доступа для трафика, относящегося к конкретной сети L3VPN (Рисунок 8).

```
...
+--rw vpn-nodes
   +--rw vpn-node* [vpn-node-id]
      ...
      +--rw vpn-network-accesses
         +--rw vpn-network-access* [id]
            +--rw id                          vpn-common:vpn-id
            +--rw interface-id?               string
            +--rw description?                string
            +--rw vpn-network-access-type?    identityref
            +--rw vpn-instance-profile?       leafref
            +--rw status
            |  +--rw admin-status
            |  |  +--rw status?          identityref
            |  |  +--rw last-change?    yang:date-and-time
            |  +--ro oper-status
            |     +--ro status?          identityref
            |     +--ro last-change?    yang:date-and-time
            +--rw connection
            |  ...
            +--rw ip-connection
            |  ...
            +--rw routing-protocols
            |  ...
            +--rw oam
            |  ...
            +--rw security
            |  ...
            +--rw service
               ...
```

Рисунок 8. Структура ветви доступа в сеть VPN.

Узлы данных контейнера vpn-network-access перечислены ниже.

id

Идентификатор доступа в сеть VPN.

interface-id

Указывает физический или логический интерфейс, к которому привязан доступ в сеть VPN.

description

Текстовое описание доступа в сеть VPN.

vpn-network-access-type

Служит для выбора типа сетевого интерфейса для развёртывания на устройствах.

point-to-point

Прямое соединение между конечными точками. Контроллер должен сохранять связь между физическим и логическим интерфейсом на устройстве с id vpn-network-access.

multipoint

Многоточечное соединение между сайтом клиента и PE. Контроллер должен сохранять связь между физическим и логическим интерфейсом на устройстве с id vpn-network-access.

irb

Соединение, исходящее от службы L2VPN. Идентификатор такой службы (l2vpn-id) может быть включён в контейнер connection, как показано на рисунке 9 (параграф 7.6.1). Контроллер должен сохранять связь между физическим и логическим интерфейсом на устройстве с id vpn-network-access.

loopback
Представляет создание логического интерфейса на устройстве. Пример использования loopback-интерфейса в L3NM представлен в Приложении A.2.

vpn-instance-profile
Указывает активный профиль экземпляра VPN на уровне узла VPN. Ссылка на активный профиль экземпляра VPN предполагает, что все связанные узлы данных будут наследоваться доступом в VPN. Однако некоторые унаследованные данные (например, поддержка групповой передачи) могут быть переопределены на уровне доступа в сеть VPN и в таком случае уточненные значения будут иметь преимущество над унаследованными.

status
Указывает административный и рабочий статус доступа в сеть VPN.

connection
Представляет и группирует набор подключений L2, откуда приходит трафик L3VPN в конкретном доступе в сеть VPN (см. 7.6.1. Соединение).

ip-connection
Сведения о подключении L3 для доступа в сеть VPN (например, адреса IP, см. 7.6.2. Соединение IP).

routing-protocols
Сведения о конфигурации маршрутизации CE-PE (см. 7.6.3. Протоколы маршрутизации CE-PE).

oam
Задаёт механизмы OAM, используемые для доступа в сеть VPN (см. 7.6.4. OAM).

security
Задаёт аутентификацию и шифрование, применяемые для данного доступа в сеть VPN (см. 7.6.5. Безопасность).

service
Параметры обслуживания (например, QoS, групповая передача) для данного доступа в сеть VPN (см. 7.6.6. Услуги.

7.6.1. Соединение

Контейнер connection представляет связность L2 с L3VPN для конкретного доступа в сеть VPN. Как показано на рисунке 9, контейнер connection задаёт протоколы и параметрв для включения такой связности L2. Трафик может входить в VPN с инкапсуляцией или без неё (например, VLAN, QinQ). Контейнер encapsulation задаёт применяемую инкапсуляцию L2 (при наличии) и позволяет настроить соответствующие теги.

Интерфейс, который подключается к L3VPN, указывает interface-id на уровне vpn-network-access. С точки зрения модели сети предполагается, что interface-id достаточно для идентификации этого интерфейса. Однако для некоторых реализаций и развёртываний может потребоваться настройка конкретных субинтерфейсов L2. Такие интерфейсы, связанные с L2, могут быть включены в l2-termination-point.

Если нужен туннель L2 для завершения службы в соединении CE-PE, применяется контейнер l2-tunnel-service, задающий параметры, требуемые для организации такой туннельной службы (например, VPLS[1] или VXLAN[2]). Задан идентификатор l2-tunnel-type для указания типа туннеля L2. Контейнер может также указывать псевдопровод (параграф 6.1 в [RFC8077]).

Как указано в параграфе 7.6, l2vpn-id служит для идентификации услуги L2VPN, связанной с интегрированным интерфейсом моста и маршрутизации (Integrated Routing and Bridging или IRB).

Для реализаций, требующих внутренних мостов, в local-bridge-reference можно указать ссылку на локальный мост (это может быть локальный домен мостов).

Согласно [RFC4176], сайт представляет местоположение клиента VPN, подключённое к сети сервис-провайдера через канал CE-PE, который может иметь доступ хотя бы к одной VPN. Соединение сайта клиента с сетью провайдера является опорным (bearer) и с каждым сайтом связан список таких соединений. Опорное соединение - это соединение L2 с сайтом. Для L3NM предполагается, что опорное соединение выделено сервис-провайдером на этапе оркестровки (организации) службы. Это соединение связано с элементом сети и портом, поэтому указывается просто ссылкой bearer-reference для связывания запроса услуги (например, L3SM) и L3NM.

L3NM можно использовать для создания интерфейса агрегата каналов (Link Aggregation Group или LAG) для данной услуги L3VPN (lag-interface) [IEEE802.1AX]. Такой интерфейс LAG можно указать в ветви interface-id (параграф 7.6).

```
...
+--rw connection
|  +--rw encapsulation
|  |  +--rw type?              identityref
|  |  +--rw dot1q
|  |  |  +--rw tag-type?   identityref
|  |  |  +--rw cvlan-id?   uint16
|  |  +--rw priority-tagged
|  |  |  +--rw tag-type?   identityref
|  |  +--rw qinq
|  |     +--rw tag-type?   identityref
|  |     +--rw svlan-id    uint16
|  |     +--rw cvlan-id    uint16
|  +--rw (l2-service)?
|  |  +--:(l2-tunnel-service)
|  |  |  +--rw l2-tunnel-service
|  |  |     +--rw type?         identityref
|  |  |     +--rw pseudowire
|  |  |     |  +--rw vcid?      uint32
|  |  |     |  +--rw far-end?   union
|  |  |     +--rw vpls
|  |  |     |  +--rw vcid?      uint32
|  |  |     |  +--rw far-end*   union
|  |  |     +--rw vxlan
```

[1]Virtual Private LAN Service - служба виртуальной частной ЛВС.
[2]Virtual eXtensible Local Area Network - расширяемая виртуальная частная ЛВС.

```
|  |  |        +--rw vni-id             uint32
|  |  |        +--rw peer-mode?         identityref
|  |  |        +--rw peer-ip-address*   inet:ip-address
|  |  +--:(l2vpn)
|  |     +--rw l2vpn-id?            vpn-common:vpn-id
|  +--rw l2-termination-point?      string
|  +--rw local-bridge-reference?    string
|  +--rw bearer-reference?          string
|  |       {vpn-common:bearer-reference}?
|  +--rw lag-interface {vpn-common:lag-interface}?
|     +--rw lag-interface-id?   string
|     +--rw member-link-list
|        +--rw member-link* [name]
|           +--rw name    string
...
```

Рисунок 9. Структура ветви соединений.

7.6.2. Соединение IP

Контейнер ip-connection служит для группировки сведений о связности L3, в частности, адреса IP доступа в сеть VPN. Выделенный адрес представляет нофигурацию интерфейса PE. Отметим, что для завершения службы L3 может потребоваться отдельный адрес интерфейса L3, отличный от указанного в контейнере connection. Идентификатор такого интерфейса включается в l3-termination-point. Этот узел данных может служить, например, для передачи интерфейса домена мостов. Как показано на рисунке 10, контейнер ip-connection может содержать адреса IPv4, IPv6 или оба, если поддерживается dual-stack.

```
...
+--rw vpn-network-accesses
   +--rw vpn-network-access* [id]
      ...
      +--rw ip-connection
      |  +--rw l3-termination-point?     string
      |  +--rw ipv4 {vpn-common:ipv4}?
      |  |  ...
      |  +--rw ipv6 {vpn-common:ipv6}?
      |     ...
      ...
```

Рисунок 10. Структура ветви соединений IP.

Для IPv4 и IPv6 соединение IP поддерживает 3 режима назначения адресов IP для клиентов: DHCP провайдера, ретранслятор DHCP и статическое назначение. Отметим, что для IPv6 можно использовать автоматическую настройку адресов без учёта состояния (Stateless Address Autoconfiguration или SLAAC) [RFC4862]. Для адресов IPv4 и IPv6 применяется address-allocation-type для указания режима выделения адресов IP, активируемого для данного доступа в сеть VPN. При установке для address-allocation-type значения provider-dhcp назначение DHCP может происходить локально или от внешнего сервера DHCP, в зависимости от dhcp-service-type.

На рисунке 11 показана структура динамического выделения адреса IPv4 (т. е. с помощью DHCP).

```
...
+--rw ip-connection
|  +--rw l3-termination-point?     string
|  +--rw ipv4 {vpn-common:ipv4}?
|  |  +--rw local-address?             inet:ipv4-address
|  |  +--rw prefix-length?             uint8
|  |  +--rw address-allocation-type?   identityref
|  |  +--rw (allocation-type)?
|  |     +--:(provider-dhcp)
|  |     |  +--rw dhcp-service-type?   enumeration
|  |     |  +--rw (service-type)?
|  |     |     +--:(relay)
|  |     |     |  +--rw server-ip-address*
|  |     |     |          inet:ipv4-address
|  |     |     +--:(server)
|  |     |        +--rw (address-assign)?
|  |     |           +--:(number)
|  |     |           |  +--rw number-of-dynamic-address?
|  |     |           |           uint16
|  |     |           +--:(explicit)
|  |     |              +--rw customer-addresses
|  |     |                 +--rw address-pool* [pool-id]
|  |     |                    +--rw pool-id          string
|  |     |                    +--rw start-address
|  |     |                    |           inet:ipv4-address
|  |     |                    +--rw end-address?
|  |     |                                inet:ipv4-address
|  |     +--:(dhcp-relay)
|  |     |  +--rw customer-dhcp-servers
|  |     |     +--rw server-ip-address*   inet:ipv4-address
|  |     +--:(static-addresses)
|  |        ...
...
```

Рисунок 11. Структура ветви соединений IP (IPv4).

На рисунке 12 показана структура динамического выделения адреса IPv6 (например, DHCPv6 и/или SLAAC). Отметим, что при установке для address-allocation-type значения slaac опция Prefix Information в Router Advertisements для SLAAC будет передавать префикс IPv6, определяемый local-address и prefix-length. Например при local-address со значением 2001:db8:0:1::1 и prefix-length со значением 64 будет использован префикс IPv6 2001:db8:0:1::/64.

```
...
+--rw ip-connection
|  +--rw l3-termination-point?     string
|  +--rw ipv4 {vpn-common:ipv4}?
|  |  ...
|  +--rw ipv6 {vpn-common:ipv6}?
|     +--rw local-address?                inet:ipv6-address
|     +--rw prefix-length?                uint8
|     +--rw address-allocation-type?      identityref
|     +--rw (allocation-type)?
|        +--:(provider-dhcp)
|        |  +--rw provider-dhcp
|        |     +--rw dhcp-service-type?
|        |     |       enumeration
|        |     +--rw (service-type)?
|        |        +--:(relay)
|        |        |  +--rw server-ip-address*
|        |        |          inet:ipv6-address
|        |        +--:(server)
|        |           +--rw (address-assign)?
|        |              +--:(number)
|        |              |  +--rw number-of-dynamic-address?
|        |              |          uint16
|        |              +--:(explicit)
|        |                 +--rw customer-addresses
|        |                    +--rw address-pool*  [pool-id]
|        |                       +--rw pool-id      string
|        |                       +--rw start-address
|        |                       |       inet:ipv6-address
|        |                       +--rw end-address?
|        |                               inet:ipv6-address
|        +--:(dhcp-relay)
|        |  +--rw customer-dhcp-servers
|        |     +--rw server-ip-address*
|        |             inet:ipv6-address
|        +--:(static-addresses)
|           ...
```

Рисунок 12. Структура ветви соединений IP (IPv6).

При статической адресации (Рисунок 13) модель поддерживает назначение нескольких адресов IP в одном vpn-network-access. Для идентификации основного адреса в соединении должна быть указана ссылка primary-address с соответствующим значением address-id.

```
...
+--rw ip-connection
|  +--rw l3-termination-point?     string
|  +--rw ipv4 {vpn-common:ipv4}?
|  |  +--rw address-allocation-type?         identityref
|  |  +--rw (allocation-type)?
|  |     ...
|  |     +--:(static-addresses)
|  |        +--rw primary-address?        -> ../address/address-id
|  |        +--rw address* [address-id]
|  |           +--rw address-id          string
|  |           +--rw customer-address?   inet:ipv4-address
|  +--rw ipv6 {vpn-common:ipv6}?
|     +--rw address-allocation-type?         identityref
|     +--rw (allocation-type)?
|        ...
|        +--:(static-addresses)
|           +--rw primary-address?     -> ../address/address-id
|           +--rw address* [address-id]
|              +--rw address-id          string
|              +--rw customer-address?   inet:ipv6-address
...
```

Рисунок 13. Структура ветви соединений IP (статический режим).

7.6.3. Протоколы маршрутизации CE-PE

Поставщик услуг VPN может настроить 1 или несколько протоколов маршрутизации, связанных с конкретным vpn-network-access, для работы между PE и CE. Каждый экземпляр указывается однозначно, чтобы можно было настроить на канале несколько экземпляров одного протокола маршрутизации. Ветвь routing-protocols показана на рисунке 14.

```
   ...
   +--rw vpn-network-accesses
      +--rw vpn-network-access* [id]
         ...
         +--rw routing-protocols
         |  +--rw routing-protocol* [id]
         |     +--rw id   string
```

```
|     +--rw type?              identityref
|     +--rw routing-profiles* [id]
|     |  +--rw id      leafref
|     |  +--rw type?   identityref
|     +--rw static
|     |  ...
|     +--rw bgp
|     |  ...
|     +--rw ospf
|     |  ...
|     +--rw isis
|     |  ...
|     +--rw rip
|     |  ...
|     +--rw vrrp
|        ...
+--rw security
    ...
```

Рисунок 14. Структура ветви маршрутизации.

Можно задать несколько экземпляров протокола маршрутизации, каждый из которых однозначно указывается значением id. Тип экземпляра маршрутизации указывает лист type. Значения этих атрибутов определены в [RFC9181] (routing-protocol-type). Настройка нескольких экземпляров протокола маршрутизации не означает автоматически наличие (с точки зрения конфигурации) параллельных экземпляров на канале PE-CE. Каждая реализация (обычно организатор сети, как показано на рисунке 1) самостоятельно выбирает подходящую конфигурацию в зависимости от базовых возможностей и рабочих рекомендаций сервис-провайдера. Например, если нужно реализовать несколько партнёров BGP, потребуется настроить несколько экземпляров BGP как часть этой модели. Однако с точки зрения конфигурации устройства это можно реализовать разными способами:

- несколько процессов BGP с одним соседом для всех процессов;
- 1 процесс BGP с несколькими соседями;
- комбинация предыдущих вариантов.

Конфигурация маршрутизации не включает правила нижнего уровня, они обрабатываются на уровне конфигурации устройств. Локальные правила сервис-провайдера (например, фильтрация) реализуются как часть конфигурации устройства, они не фиксируются в L3NM, но модель позволяет связать локальные профили с экземплярами маршрутизации (routing-profiles). Отметим, что эти профили маршрутизации могут охватывать параметры, которые глобально применяются ко всем службам L3VPN в сети сервис-провайдера, тогда как настраиваемые параметры L3VPN фиксируются средствами L3NM. Таким образом, предоставление услуги L3VPN будет основываться на создании экземпляров этих глобальных профилей маршрутизации и настройке L3NM.

7.6.3.1. Статическая маршрутизация

L3NM поддерживает конфигурации с 1 или несколькими статическими маршрутами IPv4/IPv6. Поскольку для IPv4 и IPv6 применяется одна структура, рассматривался вариант использования одного контейнера для группировки статических записей независимо от семейства адресов. Однако от этого отказались, чтобы упростить сопоставление с использованием структуры из [RFC8299]. Ветвь для статической маршрутизации показана на рисунке 15.

```
...
+--rw routing-protocols
|  +--rw routing-protocol* [id]
|     ...
|     +--rw static
|     |  +--rw cascaded-lan-prefixes
|     |     +--rw ipv4-lan-prefixes*
|     |     |       [lan next-hop]
|     |     |       {vpn-common:ipv4}?
|     |     |  +--rw lan          inet:ipv4-prefix
|     |     |  +--rw lan-tag?       string
|     |     |  +--rw next-hop       union
|     |     |  +--rw bfd-enable?    boolean
|     |     |  +--rw metric?        uint32
|     |     |  +--rw preference?    uint32
|     |     |  +--rw status
|     |     |     +--rw admin-status
|     |     |     |  +--rw status?          identityref
|     |     |     |  +--rw last-change?   yang:date-and-time
|     |     |     +--ro oper-status
|     |     |        +--ro status?          identityref
|     |     |        +--ro last-change?   yang:date-and-time
|     |     +--rw ipv6-lan-prefixes*
|     |             [lan next-hop]
|     |             {vpn-common:ipv6}?
|     |        +--rw lan          inet:ipv6-prefix
|     |        +--rw lan-tag?       string
|     |        +--rw next-hop       union
|     |        +--rw bfd-enable?    boolean
|     |        +--rw metric?        uint32
|     |        +--rw preference?    uint32
|     |        +--rw status
|     |           +--rw admin-status
|     |           |  +--rw status?          identityref
|     |           |  +--rw last-change?   yang:date-and-time
```

```
|     |           +--ro oper-status
|     |              +--ro status?         identityref
|     |              +--ro last-change?   yang:date-and-time
...
```

Рисунок 15. Структура ветви статической маршрутизации.

Узлы данных для префикса IP указаны ниже.

lan-tag
Указывает локальный тег (например, myfavorite-lan), используемый для применения локальных правил.

next-hop
Указывает следующий узел статического маршрута. Это может быть адрес IP, предопределённый тип next-hop (например, discard или local-link) и т. п.

bfd-enable
Включает и отключает BFD для статической записи.

metric
Указывает метрику, связанную со статической записью и применяемую при экспорте маршрута в IGP.

preference
Указывает предпочтение для записи, используемое для выбора среди маршрутов к одному префиксу.

status
Указывает статус записи и может служить для (де)активации отдельных статических маршрутов.

7.6.3.2. BGP

L3NM позволяет настраивать соседей BGP, включая набор параметров, которые нужно задать на уровне интерфейса для настройки службы. Контейнер bgp не предназначен для включения всех параметров BGP, это относится скорее к полной модели устройства BGP. Ветвь для маршрутизации BGP показана на рисунке 16.

```
...
+--rw routing-protocols
|  +--rw routing-protocol* [id]
|     ...
|     +--rw bgp
|     |  +--rw description?               string
|     |  +--rw local-as?                  inet:as-number
|     |  +--rw peer-as                    inet:as-number
|     |  +--rw address-family?            identityref
|     |  +--rw local-address?             union
|     |  +--rw neighbor*                  inet:ip-address
|     |  +--rw multihop?                  uint8
|     |  +--rw as-override?               boolean
|     |  +--rw allow-own-as?              uint8
|     |  +--rw prepend-global-as?         boolean
|     |  +--rw send-default-route?        boolean
|     |  +--rw site-of-origin?            rt-types:route-origin
|     |  +--rw ipv6-site-of-origin?       rt-types:ipv6-route-origin
|     |  +--rw redistribute-connected* [address-family]
|     |  |  +--rw address-family    identityref
|     |  |  +--rw enable?           boolean
|     |  +--rw bgp-max-prefix
|     |  |  +--rw max-prefix?          uint32
|     |  |  +--rw warning-threshold?   decimal64
|     |  |  +--rw violate-action?      enumeration
|     |  |  +--rw restart-timer?       uint32
|     |  +--rw bgp-timers
|     |  |  +--rw keepalive?   uint16
|     |  |  +--rw hold-time?   uint16
|     |  +--rw authentication
|     |  |  +--rw enable?            boolean
|     |  |  +--rw keying-material
|     |  |     +--rw (option)?
|     |  |        +--:(ao)
|     |  |        |  +--rw enable-ao?         boolean
|     |  |        |  +--rw ao-keychain?       key-chain:key-chain-ref
|     |  |        +--:(md5)
|     |  |        |  +--rw md5-keychain?      key-chain:key-chain-ref
|     |  |        +--:(explicit)
|     |  |        |  +--rw key-id?            uint32
|     |  |        |  +--rw key?               string
|     |  |        |  +--rw crypto-algorithm?  identityref
|     |  |        +--:(ipsec)
|     |  |           +--rw sa?             string
|     |  +--rw status
|     |     +--rw admin-status
|     |     |  +--rw status?         identityref
|     |     |  +--rw last-change?   yang:date-and-time
|     |     +--ro oper-status
|     |        +--ro status?         identityref
|     |        +--ro last-change?   yang:date-and-time
...
```

Рисунок 16. Структура ветви маршрутизации BGP.

Ниже приведены узлы данных контейнера. Реализации (например, организатор сети) должна вывести соответствующую конфигурацию устройства BGP.

description
Описание сессии BGP.

local-as
Указывает локальный номер AS (ASN), если нужен номер, отличающийся от ASN, заданного на уровне узла VPN.

peer-as
Указывает ASN клиента.

address-family
Указывает семейство адресов партнёра (ipv4, ipv6, dual-stack), которое используется вместе с vpn-type для вывода подходящих идентификаторов AFI/SAFI[1], которые будут частью выведенных конфигураций устройств (например, unicast IPv4 MPLS L3VPN - AFI,SAFI = 1,128, как указано в параграфе 4.3.4 [RFC4364]).

local-address
Указывает адрес или ссылку на интерфейс, используемый для организации транспортной сессии BGP.

neighbor
Указывает двух (с одним семейством адресов) или одного соседа (если для address-family установлено dual-stack). Список адресов IP соседей BGP может передаваться затем в этот узел данных.

multihop
Указывает число разрешённых узлов пересылки (IP hop) между PE и его партнёром BGP.

as-override
Установка этого листа показывает, разрешено ли переопределение ASN, т. е. замена ASN клиента в атрибуте BGP AS_PATH значением ASN в атрибуте local-as.

allow-own-as
Применяется в некоторых топологиях (например, hub-and-spoke — звезда), чтобы разрешить включение ASN провайдера в атрибут BGP AS_PATH, полученный от CE. Петли предотвращаются установкой в allow-own-as максимального числа вхождений ASN провайдера. По умолчанию установлено значение 0, т. е. атрибуты AS_PATH с ASN провайдера отвергаются.

prepend-global-as
При настройке разных ASN на уровне узла VPN и доступа в сеть этот параметр определяет, добавляется ли ASN от уровня узла VPN в начало атрибута AS_PATH.

send-default-route
Управляет анонсированием партнёру принятого по умолчанию маршрута.

site-of-origin
Однозначно идентифицирует набор маршрутов, полученных с сайта через определённое соединение CE-PE. Это служит для предотвращения петель в маршрутизации (раздел 7 в [RFC4364]). Атрибут Site of Origin кодируется как Route Origin Extended Community.

ipv6-site-of-origin
Передаёт группу IPv6 Address Specific BGP Extended Community, служащую для индикации Site of Origin в сведениях VRF [RFC5701]. Применяется для предотвращения петель в маршрутизации.

redistribute-connected
Управляет анонсированием канала PE-CE другим PE.

bgp-max-prefix
Управляет поведением при достижении максимального числа префиксов.

max-prefix
Указывает максимальное число префиксов BGP, разрешённых в сессии BGP. При достижении предела выполняется действие, заданное violate-action.

warning-threshold
При достижении этого предела передаётся предупреждение.

violate-action
Указывает действие, выполняемое при достижении максимального числа префиксов BGP. Примерами таких действий служат отправка предупреждения, отбрасывание избыточных путей от партнёра, перезапуск сессии.

restart-timer
Указывает интервал времени (в секундах) после которого сессия BGP будет организована заново.

bgp-timers
Этот контейнер содержит два таймера - (1) hold-time задаёт интервал, используемый для таймера удержания (Hold Timer, параграф 4.2 в [RFC4271]) при организации сессии BGP, (2) keepalive задаёт интервал для KeepaliveTimer между PE и партнёром BGP (параграф 4.4 в [RFC4271]). Оба таймера указываются в секундах.

authentication
Модуль следует рекомендациям параграфа 13.2 в [RFC4364], поскольку позволяет включить опцию аутентификации TCP (TCP Authentication Option или TCP-AO) [RFC5925] и поддерживает установленные системы, использующие MD5. Дополнительно модуль позволяет включить возможность использовать IPsec.
В этой версии L3NM предполагается, что параметры, относящиеся к TCP-AO, настроены заранее как часть цепочки ключей, на которую ссылается L3NM. Допущений о способе предварительной настройки цепочки ключей не принимается, однако структуре такой цепочки следует охватывать узлы данных сверх указанных в [RFC8177], в основном это SendID и RecvID (параграф 3.1 в [RFC5925]).

status
Указывает статус экземпляра маршрутизации BGP.

7.6.3.3. OSPF

OSPF можно настроить для работы как протокол маршрутизации в vpn-network-access (Рисунок 17).

```
...
+--rw routing-protocols
|  +--rw routing-protocol* [id]
|     ...
|     +--rw ospf
|     |  +--rw address-family?   identityref
|     |  +--rw area-id           yang:dotted-quad
|     |  +--rw metric?           uint16
```

[1]Address Family Identifier - идентификатор семейства адресов, Subsequent Address Family Identifier - следующий AFI.

```
|      |  +--rw sham-links  {vpn-common:rtg-ospf-sham-link}?
|      |  |  +--rw sham-link* [target-site]
|      |  |     +--rw target-site    string
|      |  |     +--rw metric?        uint16
|      |  +--rw max-lsa?          uint32
|      |  +--rw authentication
|      |  |  +--rw enable?            boolean
|      |  |  +--rw keying-material
|      |  |     +--rw (option)?
|      |  |        +--:(auth-key-chain)
|      |  |        |  +--rw key-chain?
|      |  |        |          key-chain:key-chain-ref
|      |  |        +--:(auth-key-explicit)
|      |  |        |  +--rw key-id?      uint32
|      |  |        |  +--rw key?         string
|      |  |        |  +--rw crypto-algorithm?
|      |  |        |          identityref
|      |  |        +--:(ipsec)
|      |  |           +--rw sa?    string
|      |  +--rw status
|      |     +--rw admin-status
|      |     |  +--rw status?        identityref
|      |     |  +--rw last-change?  yang:date-and-time
|      |     +--ro oper-status
|      |        +--ro status?        identityref
|      |        +--ro last-change?  yang:date-and-time
...
```

Рисунок 17. Структура ветви маршрутизации OSPF.

address-family
Указывает активацию IPv4, IPv6 или обоих семейств адресов. При запросе IPv4 или dual-stack реализация (например, организатор сети) сама выбирает OSPFv2 [RFC4577] или OSPFv3 [RFC6565] для анонсирования маршрутов IPv4. Решение обычно отражается в конфигурации устройства, выведенной для применения в L3VPN.

area-id
Указывает OSPF Area ID.

metric
Связывает метрику с маршрутами OSPF.

sham-links
Служит для создания фиктивных (sham) каналов OSPF между 2 доступами в сеть VPN, использующими общую область и имеющими обходной (backdoor) канал (параграф 4.2.7 в [RFC4577] и раздел 5 в [RFC6565]).

max-lsa
Максимальное число анонсов состояния канала (Link State Advertisement или LSA), которые воспримет экземпляр OSPF.

authentication
Управляет схемами аутентификации для экземпляра OSPF. Поддерживается IPsec для аутентификации OSPFv3 [RFC4552] и Authentication Trailer для OSPFv2 [RFC5709] [RFC7474] и OSPFv3 [RFC7166].

status
Указывает статус экземпляра маршрутизации OSPF.

7.6.3.4. IS-IS

Модель позволяет включить IS-IS [ISO10589] [RFC1195] [RFC5308] для работы на интерфейсе vpn-network-access. Ветвь IS-IS показана на рисунке 18.

```
...
+--rw routing-protocols
|  +--rw routing-protocol* [id]
|     ...
|     +--rw isis
|     |  +--rw address-family?   identityref
|     |  +--rw area-address      area-address
|     |  +--rw level?            identityref
|     |  +--rw metric?           uint16
|     |  +--rw mode?             enumeration
|     |  +--rw authentication
|     |  |  +--rw enable?            boolean
|     |  |  +--rw keying-material
|     |  |     +--rw (option)?
|     |  |        +--:(auth-key-chain)
|     |  |        |  +--rw key-chain?
|     |  |        |          key-chain:key-chain-ref
|     |  |        +--:(auth-key-explicit)
|     |  |           +--rw key-id?             uint32
|     |  |           +--rw key?                string
|     |  |           +--rw crypto-algorithm?   identityref
|     |  +--rw status
|     |     +--rw admin-status
|     |     |  +--rw status?        identityref
|     |     |  +--rw last-change?  yang:date-and-time
|     |     +--ro oper-status
|     |        +--ro status?        identityref
|     |        +--ro last-change?  yang:date-and-time
...
```

Рисунок 18. Структура ветви маршрутизации IS-IS.

Ниже перечислены узлы данных IS-IS.

address-family
Указывает активацию IPv4, IPv6 или обоих семейств адресов.
area-address
Указывает адрес области IS-IS.
level
Указывает уровень IS-IS: Level 1, Level 2 или оба.
metric
Связывает метрику с маршрутами IS-IS.
mode
Указывает тип режима интерфейса IS-IS - активный (active, т. е. передающий и принимающий пакеты управления протокола IS-IS) или пассивный (passive, без передачи обновлений IS-IS через этот интерфейс).
authentication
Управляет схемами аутентификации для экземпляра IS-IS. Поддерживаются цепочки ключей [RFC8177], а также прямое задание ключа и алгоритмов проверки подлинности.
status
Указывает статус экземпляра маршрутизации IS-IS.

7.6.3.5. RIP

Модель позволяет пользователю настроить RIP для работы на интерфейсе vpn-network-access (Рисунок 19).

```
...
+--rw routing-protocols
|  +--rw routing-protocol* [id]
|     ...
|     +--rw rip
|     |  +--rw address-family?   identityref
|     |  +--rw timers
|     |  |  +--rw update-interval?     uint16
|     |  |  +--rw invalid-interval?    uint16
|     |  |  +--rw holddown-interval?   uint16
|     |  |  +--rw flush-interval?      uint16
|     |  +--rw default-metric?   uint8
|     |  +--rw authentication
|     |  |  +--rw enable?            boolean
|     |  |  +--rw keying-material
|     |  |     +--rw (option)?
|     |  |        +--:(auth-key-chain)
|     |  |        |  +--rw key-chain?
|     |  |        |          key-chain:key-chain-ref
|     |  |        +--:(auth-key-explicit)
|     |  |           +--rw key?                string
|     |  |           +--rw crypto-algorithm?   identityref
|     |  +--rw status
|     |     +--rw admin-status
|     |     |  +--rw status?        identityref
|     |     |  +--rw last-change?  yang:date-and-time
|     |     +--ro oper-status
|     |        +--ro status?        identityref
|     |        +--ro last-change?  yang:date-and-time
...
```

Рисунок 19. Структура ветви маршрутизации RIP.

Ниже перечислены узлы данных для протокола RIP.

address-family
Указывает активацию IPv4, IPv6 или обоих семейств адресов и служит для включения RIPv2 [RFC2453], RIP Next Generation (RIPng) или обоих протоколов [RFC2080].
timers
Задаёт указанные ниже таймеры (в секундах).
update-interval
Интервал передачи обновлений RIP.
invalid-interval
Интервал, по истечении которого маршрут RIP объявляется недействительным.
holddown-interval
Интервал, по истечении которого лучшие маршруты RIP освобождаются.
flush-interval
Интервал, по истечении которого маршрут удаляется из таблицы маршрутизации.
default-metric
Задаёт принятую по умолчанию метрику RIP.
authentication
Управляет схемами аутентификации для экземпляра RIP.
status
Указывает статус экземпляра маршрутизации RIP.

7.6.3.6. VRRP

Модель позволяет включить протокол резервирования виртуального маршрутизатора (Virtual Router Redundancy Protocol или VRRP) на интерфейсе vpn-network-access (Рисунок 20).

```
...
+--rw routing-protocols
|  +--rw routing-protocol* [id]
```

```
|      ...
|      +--rw vrrp
|         +--rw address-family*   identityref
|         +--rw vrrp-group?       uint8
|         +--rw backup-peer?      inet:ip-address
|         +--rw virtual-ip-address*   inet:ip-address
|         +--rw priority?         uint8
|         +--rw ping-reply?       boolean
|         +--rw status
|            +--rw admin-status
|            |  +--rw status?        identityref
|            |  +--rw last-change?  yang:date-and-time
|            +--ro oper-status
|               +--ro status?        identityref
|               +--ro last-change?  yang:date-and-time
...
```

Рисунок 20. Структура ветви маршрутизации VRRP.

Ниже укаазны поддерживаемые узлы данных.

address-family
Указывает активацию IPv4, IPv6 или обоих семейств адресов. VRRP версии 3 [RFC5798] поддерживает IPv4 и IPv6.
vrrp-group
Служит для идентификации группы VRRP.
backup-peer
Передаёт IP-адрес партнёра.
virtual-ip-address
Виртуальные адреса IP для одной группы VRRP.
priority
Назначает приоритет выбора VRRP для резервного виртуального маршрутизатора.
ping-reply
Управляет откликами узла VRRP на запросы ping.
status
Указывает статус экземпляра VRRP.
Узел для аутентификации отсутствует, поскольку аутентификация в VRRP не поддерживается (раздел 9 в [RFC5798]).

7.6.4. OAM

Контейнер oam (Рисунок 21) задаёт механизмы OAM, используемые для доступа в сеть VPN. В текущей версии L3NM поддерживается лишь BFD.

```
...
+--rw oam
|  +--rw bfd {vpn-common:bfd}?
|     +--rw session-type?               identityref
|     +--rw desired-min-tx-interval?    uint32
|     +--rw required-min-rx-interval?   uint32
|     +--rw local-multiplier?           uint8
|     +--rw holdtime?                   uint32
|     +--rw profile?                    leafref
|     +--rw authentication!
|     |  +--rw key-chain?    key-chain:key-chain-ref
|     |  +--rw meticulous?   boolean
|     +--rw status
|        +--rw admin-status
|           |  +--rw status?         identityref
|           |  +--rw last-change?   yang:date-and-time
|           +--ro oper-status
|              +--ro status?         identityref
|              +--ro last-change?   yang:date-and-time
...
```

Рисунок 21. Структура ветви соединения IP (OAM)

Узлы данных для OAM перечислены ниже.

session-type
Указывает вариант BFD для настройки сессии (например, классический BFD [RFC5880], Seamless BFD [RFC7880]). По умолчанию предполагается, что поведение сессии BFD соответствует [RFC5880].
desired-min-tx-interval
Желаемый минимальный интервал (в мсек), который PE будет применять при передаче пакетов BFD Control, за вычетом применяемых вариаций (jitter).
required-min-rx-interval
Минимальный интервал (в мсек) между получением пакетов BFD Control, который PE способен поддерживать, за вычетом применяемых отправителем вариаций (jitter).
local-multiplier
Множитель для согласованного интервала передачи, определяющий время обнаружения партнёра.
holdtime
Указывает ожидаемое время удержания BFD в миллисекундах. Значение может наследоваться из запроса услуги (см. параграф 6.3.2.2.2 в [RFC8299]).
profile
Указывает профиль BFD (7.2. Профили VPN). Профиль может задаваться провайдером или наследоваться из запроса услуги (см. параграф 6.3.2.2.2 в [RFC8299]).

authentication

Сведения для включения режимов аутентификации BFD, рассмотренных в параграфе 6.7 [RFC5880]. В частности, лист meticulous управляет активацией «дотошного» режима, как указано в параграфах 6.7.3 и 6.7.4 [RFC5880].

status

Указывает состояние BFD.

7.6.5. Безопасность

Контейнер security задаёт аутентификацию и шифрование, применяемые к трафика для данного доступа в сеть VPN. Как показано на рисунке 22, L3NM можно применять для непосредственного управления применяемым шифрованием (например, L2 или L3) или для вызова профиля шифрования.

```
...
+--rw vpn-services
   +--rw vpn-service* [vpn-id]
      ...
      +--rw vpn-nodes
         +--rw vpn-node* [vpn-node-id]
            ...
            +--rw vpn-network-accesses
               +--rw vpn-network-access* [id]
                  ...
                  +--rw security
                  |  +--rw encryption {vpn-common:encryption}?
                  |  |  +--rw enabled?   boolean
                  |  |  +--rw layer?     enumeration
                  |  +--rw encryption-profile
                  |     +--rw (profile)?
                  |        +--:(provider-profile)
                  |        |  +--rw profile-name?        leafref
                  |        +--:(customer-profile)
                  |           +--rw customer-key-chain?
                  |                    key-chain:key-chain-ref
                  +--rw service
                      ...
```

Рисунок 22. Структура ветви безопасности.

7.6.6. Услуги

7.6.6.1. Обзор

Контейнер service задаёт параметры, применяемые к данному доступу в сеть VPN (Рисунок 23).

```
...
+--rw vpn-network-accesses
   +--rw vpn-network-access* [id]
      ...
      +--rw service
         +--rw pe-to-ce-bandwidth?   uint64 {vpn-common:inbound-bw}?
         +--rw ce-to-pe-bandwidth?   uint64 {vpn-common:outbound-bw}?
         +--rw mtu?                  uint32
         +--rw qos {vpn-common:qos}?
         |  ...
         +--rw carriers-carrier
         |       {vpn-common:carriers-carrier}?
         |  +--rw signaling-type?   enumeration
         +--rw ntp
         |  +--rw broadcast?        enumeration
         |  +--rw auth-profile
         |  |  +--rw profile-id?    string
         |  +--rw status
         |     +--rw admin-status
         |     |  +--rw status?         identityref
         |     |  +--rw last-change?   yang:date-and-time
         |     +--ro oper-status
         |        +--ro status?         identityref
         |        +--ro last-change?   yang:date-and-time
         +--rw multicast {vpn-common:multicast}?
            ...
```

Рисунок 23. Структура ветви служб.

pe-to-ce-bandwidth

Указывает входную (от сервис-провайдера к сайту) пропускную способность соединения (bps - бит/с).

ce-to-pe-bandwidth

Указывает выходную (от сайта к сервис-провайдеру) пропускную способность соединения (bps - бит/с).

mtu

Указывает MTU на уровне службы.

qos

Служит для установки правил QoS, применяемых к данному соединению (см. 7.6.6.2. QoS).

carriers-carrier

Группирует набор параметров, используемых при включении режима «оператор для операторов» (Carriers' Carriers или CsC), таких как использование BGP для сигнализации [RFC8277].

ntp

Служит для управления синхронизацией часов по протоколу NTP [RFC5905], которая может требоваться для некоторых VPN (например, VPN инфраструктуры и управления).

multicast

Управляет групповой передаче и содержит другие узлы, такие как семейство адресов, см. 7.7. Групповая передача.

7.6.6.2. QoS

Контейнер qos служит для определения набора правил QoS применяемых для данного соединения (Рисунок 24). Правила QoS могут задавать классификацию и действия, например, действием QoS может быть установка пределной скорости на входе или выходе для данного класса обслуживания.

```
...
+--rw qos {vpn-common:qos}?
|  +--rw qos-classification-policy
|  |  +--rw rule* [id]
|  |     +--rw id              string
|  |     +--rw (match-type)?
|  |     |  +--:(match-flow)
|  |     |  |  +--rw (l3)?
|  |     |  |  |  +--:(ipv4)
|  |     |  |  |  |  ...
|  |     |  |  |  +--:(ipv6)
|  |     |  |  |     ...
|  |     |  |  +--rw (l4)?
|  |     |  |     +--:(tcp)
|  |     |  |     |  ...
|  |     |  |     +--:(udp)
|  |     |  |        ...
|  |     |  +--:(match-application)
|  |     |     +--rw match-application?
|  |     |              identityref
|  |     +--rw target-class-id?    string
|  +--rw qos-action
|  |  +--rw rule* [id]
|  |     +--rw id                      string
|  |     +--rw target-class-id?        string
|  |     +--rw inbound-rate-limit?     decimal64
|  |     +--rw outbound-rate-limit?    decimal64
|  +--rw qos-profile
|     +--rw qos-profile* [profile]
|        +--rw profile      leafref
|        +--rw direction?   identityref
...
```

Рисунок 24. Структура ветви QoS.

Классификация QoS может быть основана на множестве критериев, включая указанные ниже.

Layer 3

Как показано на рисунке 25, классификация возможна на основе любой комбинации полей заголовков IPv4 и IPv6.

```
+--rw qos {vpn-common:qos}?
|  +--rw qos-classification-policy
|  |  +--rw rule* [id]
|  |     +--rw id           string
|  |     +--rw (match-type)?
|  |     |  +--:(match-flow)
|  |     |  |  +--rw (l3)?
|  |     |  |  |  +--:(ipv4)
|  |     |  |  |  |  +--rw ipv4
|  |     |  |  |  |     +--rw dscp?              inet:dscp
|  |     |  |  |  |     +--rw ecn?               uint8
|  |     |  |  |  |     +--rw length?            uint16
|  |     |  |  |  |     +--rw ttl?               uint8
|  |     |  |  |  |     +--rw protocol?          uint8
|  |     |  |  |  |     +--rw ihl?               uint8
|  |     |  |  |  |     +--rw flags?             bits
|  |     |  |  |  |     +--rw offset?            uint16
|  |     |  |  |  |     +--rw identification?    uint16
|  |     |  |  |  |     +--rw (destination-network)?
|  |     |  |  |  |     |  +--:(destination-ipv4-network)
|  |     |  |  |  |     |     +--rw destination-ipv4-network?
|  |     |  |  |  |     |              inet:ipv4-prefix
|  |     |  |  |  |     +--rw (source-network)?
|  |     |  |  |  |        +--:(source-ipv4-network)
|  |     |  |  |  |           +--rw source-ipv4-network?
|  |     |  |  |  |  inet:ipv4-prefix
|  |     |  |  |  +--:(ipv6)
|  |     |  |  |     +--rw ipv6
|  |     |  |  |        +--rw dscp?              inet:dscp
|  |     |  |  |        +--rw ecn?               uint8
|  |     |  |  |        +--rw length?            uint16
|  |     |  |  |        +--rw ttl?               uint8
|  |     |  |  |        +--rw protocol?          uint8
|  |     |  |  |        +--rw (destination-network)?
|  |     |  |  |        |  +--:(destination-ipv6-network)
|  |     |  |  |        |     +--rw destination-ipv6-network?
```

```
|  |     |  |  |        |             inet:ipv6-prefix
|  |     |  |  |        +--rw (source-network)?
|  |     |  |  |        |  +--:(source-ipv6-network)
|  |     |  |  |        |     +--rw source-ipv6-network?
|  |     |  |  |        |             inet:ipv6-prefix
|  |     |  |  |        +--rw flow-label?
|  |     |  |  |                   inet:ipv6-flow-label
...
```

Рисунок 25. Структура ветви QoS (L3).

Layer 4

Как описано в [RFC9181], любой протокол L4 можно указать в узле protocol ветви l3 (Рисунок 25), но в этой версии доступны лишь критерии для TCP и UDP, поскольку эти протоколы наиболее часто применяются в контексте услуг VPN. В будущем могут быть разработаны дополнения для других узлов данных L4. Связанные с TCP и UDP сопоставления могут быть заданы в L3NM, как показано на рисунке 26.

Как отмечено в [RFC9181], некоторые транспортные протоколы используют имеющийся протокол (например, TCP или UDP) как основу. Критерии сопоставления для таких протоколов могут основываться на установке protocol в ветви l3, критериях сопоставления TCP/UDP, как показано на рисунке 26, части данных (payload) TCP/UDP или сочетания указанного. Эта версия модуля не поддерживает такие расширенные критерии сопоставления. Будущие версии базового модуля VPN или дополнения к L3NM могут добавить сопоставления на основе данных (payload) транспортного протокола (например, сопоставление с битовой маской).

```
+--rw qos {vpn-common:qos}?
|  +--rw qos-classification-policy
|  |  +--rw rule* [id]
|  |     +--rw id            string
|  |     +--rw (match-type)?
|  |     |  +--:(match-flow)
|  |     |  |  +--rw (l3)?
|  |     |  |  |  ...
|  |     |  |  +--rw (l4)?
|  |     |  |     +--:(tcp)
|  |     |  |     |  +--rw tcp
|  |     |  |     |     +--rw sequence-number?          uint32
|  |     |  |     |     +--rw acknowledgement-number?   uint32
|  |     |  |     |     +--rw data-offset?              uint8
|  |     |  |     |     +--rw reserved?                 uint8
|  |     |  |     |     +--rw flags?                    bits
|  |     |  |     |     +--rw window-size?              uint16
|  |     |  |     |     +--rw urgent-pointer?           uint16
|  |     |  |     |     +--rw options?                  binary
|  |     |  |     |     +--rw (source-port)?
|  |     |  |     |     |  +--:(source-port-range-or-operator)
|  |     |  |     |     |     +--rw source-port-range-or-operator
|  |     |  |     |     |        +--rw (port-range-or-operator)?
|  |     |  |     |     |           +--:(range)
|  |     |  |     |     |           |  +--rw lower-port
|  |     |  |     |     |           |  |       inet:port-number
|  |     |  |     |     |           |  +--rw upper-port
|  |     |  |     |     |           |          inet:port-number
|  |     |  |     |     |           +--:(operator)
|  |     |  |     |     |              +--rw operator? operator
|  |     |  |     |     |              +--rw port
|  |     |  |     |     |                      inet:port-number
|  |     |  |     |     +--rw (destination-port)?
|  |     |  |     |        +--:(destination-port-range-or-operator)
|  |     |  |     |          +--rw destination-port-range-or-operator
|  |     |  |     |             +--rw (port-range-or-operator)?
|  |     |  |     |                +--:(range)
|  |     |  |     |                |  +--rw lower-port
|  |     |  |     |                |  |       inet:port-number
|  |     |  |     |                |  +--rw upper-port
|  |     |  |     |                |          inet:port-number
|  |     |  |     |                +--:(operator)
|  |     |  |     |                   +--rw operator? operator
|  |     |  |     |                   +--rw port
|  |     |  |     |                           inet:port-number
|  |     |  |     +--:(udp)
|  |     |  |        +--rw udp
|  |     |  |           +--rw length?                    uint16
|  |     |  |           +--rw (source-port)?
|  |     |  |           |  +--:(source-port-range-or-operator)
|  |     |  |           |     +--rw source-port-range-or-operator
|  |     |  |           |        +--rw (port-range-or-operator)?
|  |     |  |           |           +--:(range)
|  |     |  |           |           |  +--rw lower-port
|  |     |  |           |           |  |       inet:port-number
|  |     |  |           |           |  +--rw upper-port
|  |     |  |           |           |          inet:port-number
|  |     |  |           |           +--:(operator)
|  |     |  |           |              +--rw operator?  operator
|  |     |  |           |              +--rw port
|  |     |  |           |                      inet:port-number
|  |     |  |           +--rw (destination-port)?
|  |     |  |              +--:(destination-port-range-or-operator)
```

```
|  |     |  |                +--rw destination-port-range-or-operator
|  |     |  |                   +--rw (port-range-or-operator)?
|  |     |  |                      +--:(range)
|  |     |  |                      |  +--rw lower-port
|  |     |  |                      |  |       inet:port-number
|  |     |  |                      |  +--rw upper-port
|  |     |  |                      |          inet:port-number
|  |     |  |                      +--:(operator)
|  |     |  |                         +--rw operator?   operator
|  |     |  |                         +--rw port
|  |     |  |                                 inet:port-number
...
```

Рисунок 26. Структура ветви QoS (L4).

Соответствие приложения

Зависящая от приложения классификация (Рисунок 24).

7.7. Групповая передача

Групповая передача может быть разрешена доя отдельной VPN на уровне узла VPN и доступа в сеть VPN (Рисунок 27). Некоторые узлы данных (например, max-groups на рисунке 28) могут контролироваться на разных уровнях - служба VPN, узел VPN, доступ в сеть VPN.

```
      ...
      +--rw vpn-services
         +--rw vpn-service* [vpn-id]
            ...
            +--rw vpn-instance-profiles
            |  +--rw vpn-instance-profile* [profile-id]
            |     ....
            |     +--rw multicast {vpn-common:multicast}?
            |        ...
            +--rw vpn-nodes
               +--rw vpn-node* [vpn-node-id]
                  ...
                  +--rw active-vpn-instance-profiles
                  |  +--rw vpn-instance-profile* [profile-id]
                  |     ...
                  |     +--rw multicast {vpn-common:multicast}?
                  |        ...
                  +--rw vpn-network-accesses
                     +--rw vpn-network-access* [id]
                        ...
                        +--rw service
                           ...
                           +--rw multicast {vpn-common:multicast}?
                              ...
```

Рисунок 27. Структура ветви групповой передачи.

Связанные с групповой передачей узлы данных на уровне профиля экземпляра VPN показаны на рисунке 28.

```
...
+--rw vpn-services
   +--rw vpn-service* [vpn-id]
      ...
      +--rw vpn-instance-profiles
      |  +--rw vpn-instance-profile* [profile-id]
      |     ....
      |     +--rw multicast {vpn-common:multicast}?
      |        +--rw tree-flavor?   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 anycast
      |        |  |     |     +--rw local-address?    inet:ip-address
      |        |  |     |     +--rw rp-set-address*   inet:ip-address
      |        |  |     +--rw rp-address          inet:ip-address
      |        |  |     +--rw groups
      |        |  |        +--rw group* [id]
      |        |  |           +--rw id                     uint16
      |        |  |           +--rw (group-format)
      |        |  |              +--:(group-prefix)
      |        |  |              |  +--rw group-address?
      |        |  |              |          inet:ip-prefix
      |        |  |              +--:(startend)
      |        |  |                 +--rw group-start?
      |        |  |                 |       inet:ip-address
      |        |  |                 +--rw group-end?
      |        |  |                 |       inet:ip-address
      |        |  +--rw rp-discovery
      |        |     +--rw rp-discovery-type?   identityref
```

```
|        |     +--rw bsr-candidates
|        |        +--rw bsr-candidate-address*
|        |        |       inet:ip-address
|        |     +--rw igmp {vpn-common:igmp and vpn-common:ipv4}?
|        |     |  +--rw static-group* [group-addr]
|        |     |  |  +--rw group-addr
|        |     |  |  |       rt-types:ipv4-multicast-group-address
|        |     |  |  +--rw source-addr?
|        |     |  |          rt-types:ipv4-multicast-source-address
|        |     |  +--rw max-groups?     uint32
|        |     |  +--rw max-entries?    uint32
|        |     |  +--rw version?        identityref
|        |     +--rw mld {vpn-common:mld and vpn-common:ipv6}?
|        |     |  +--rw static-group* [group-addr]
|        |     |  |  +--rw group-addr
|        |     |  |  |       rt-types:ipv6-multicast-group-address
|        |     |  |  +--rw source-addr?
|        |     |  |          rt-types:ipv6-multicast-source-address
|        |     |  +--rw max-groups?     uint32
|        |     |  +--rw max-entries?    uint32
|        |     |  +--rw version?        identityref
|        |     +--rw pim {vpn-common:pim}?
|        |        +--rw hello-interval?
|        |        |       rt-types:timer-value-seconds16
|        |        +--rw dr-priority?      uint32
|     ...
```

Рисунок 28. Структура ветви групповой передачи (уровень профиля экземпляра VPN).

Модель поддерживает 1 тип дерева на доступ в VPN (tree- flavor): Any-Source Multicast (ASM), Source-Specific Multicast (SSM) или bidirectional.

При использовании ASM модель поддерживает настройку точек встречи (RP), которые могут быть статическими (static), bsr-rp или auto-rp. При статическом задании точек встречи их сопоставление с multicast-группой должно задаваться в контейнере rp-group-mappings. RP может быть узлом провайдера или клиента. Для клиентского RP адрес RP должен указываться в листе rp-address.

Модель поддерживает резервирование RP через лист rp-redundancy, однако детали этого выходят за рамки документа.

Когда конкретной сети VPN, применяющей ASM, нужна более оптимальная доставка трафика (например, в соответствии с [RFC8299]), можно установить optimal-traffic-delivery. При установке значения true реализация должна использовать любой механизм для более оптимальной доставки трафика клиента. Например, anycast является одним из механизмом повышения избыточности RP, обеспечивая устойчивость к отказам и более быстрое восстановление.

При настройке связанных с групповой передачей параметров на уровне узла VPN (Рисунок 29), применяется такая же структура, какая показана на рисунке 30. При задании на уровне узла VPN параметры IGMP [RFC1112] [RFC2236] [RFC3376], MLD [RFC2710] [RFC3810] и PIM [RFC7761] применяются к каждому доступу в сеть VPN этого узла VPN, если только на уровне доступа в сеть VPN не заданы конкретные узлы.

```
...
+--rw vpn-nodes
   +--rw vpn-node* [vpn-node-id]
      ...
      +--rw active-vpn-instance-profiles
      |  +--rw vpn-instance-profile* [profile-id]
      |     ...
      |     +--rw multicast {vpn-common:multicast}?
      |        +--rw tree-flavor*   identityref
      |        +--rw rp
      |        |  ...
      |        +--rw igmp {vpn-common:igmp and vpn-common:ipv4}?
      |        |  ...
      |        +--rw mld {vpn-common:mld and vpn-common:ipv6}?
      |        |  ...
      |        +--rw pim {vpn-common:pim}?
      |           ...
```

Рисунок 29. Структура ветви групповой передачи (уровень узла VPN).

Связанные с групповой передачей узлы данных на уровне доступа в сеть VPN показаны на рисунке 30. Настроенные на этом уровне значения переопределяют значения, заданные на других уровнях.

```
...
+--rw vpn-network-accesses
   +--rw vpn-network-access* [id]
      ...
      +--rw service
         ...
         +--rw multicast {vpn-common:multicast}?
            +--rw access-type?      enumeration
            +--rw address-family?   identityref
            +--rw protocol-type?    enumeration
            +--rw remote-source?    boolean
            +--rw igmp  {vpn-common:igmp}?
            |  +--rw static-group* [group-addr]
            |  |  +--rw group-addr
            |  |          rt-types:ipv4-multicast-group-address
```

```
|  |  +--rw source-addr?
|  |          rt-types:ipv4-multicast-source-address
|  +--rw max-groups?          uint32
|  +--rw max-entries?         uint32
|  +--rw max-group-sources?   uint32
|  +--rw version?             identityref
|  +--rw status
|     +--rw admin-status
|     |  +--rw status?         identityref
|     |  +--rw last-change?   yang:date-and-time
|     +--ro oper-status
|        +--ro status?         identityref
|        +--ro last-change?   yang:date-and-time
+--rw mld {vpn-common:mld}?
|  +--rw static-group* [group-addr]
|  |  +--rw group-addr
|  |          rt-types:ipv6-multicast-group-address
|  |  +--rw source-addr?
|  |          rt-types:ipv6-multicast-source-address
|  +--rw max-groups?          uint32
|  +--rw max-entries?         uint32
|  +--rw max-group-sources?   uint32
|  +--rw version?             identityref
|  +--rw status
|     +--rw admin-status
|     |  +--rw status?         identityref
|     |  +--rw last-change?   yang:date-and-time
|     +--ro oper-status
|        +--ro status?         identityref
|        +--ro last-change?   yang:date-and-time
+--rw pim {vpn-common:pim}?
   +--rw hello-interval?   rt-types:timer-value-seconds16
   +--rw dr-priority?      uint32
   +--rw status
      +--rw admin-status
      |  +--rw status?         identityref
      |  +--rw last-change?   yang:date-and-time
      +--ro oper-status
         +--ro status?         identityref
         +--ro last-change?   yang:date-and-time
```

Рисунок 30. Структура ветви групповой передачи (уровень доступа в сеть VPN).

8. Модуль YANG L3NM

Этот модуль использует типы из [RFC6991], [RFC8343], [RFC9181] и группировки из [RFC8519], [RFC8177], [RFC8294].

```
<CODE BEGINS> file "ietf-l3vpn-ntw@2022-02-14.yang"
module ietf-l3vpn-ntw {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-l3vpn-ntw";
  prefix l3nm;

  import ietf-vpn-common {
    prefix vpn-common;
    reference
      "RFC 9181: A Common YANG Data Model for Layer 2 and Layer 3
                 VPNs";
  }
  import ietf-inet-types {
    prefix inet;
    reference
      "RFC 6991: Common YANG Data Types, Section 4";
  }
  import ietf-yang-types {
    prefix yang;
    reference
      "RFC 6991: Common YANG Data Types, Section 3";
  }
  import ietf-key-chain {
    prefix key-chain;
    reference
      "RFC 8177: YANG Data Model for Key Chains";
  }
  import ietf-routing-types {
    prefix rt-types;
    reference
      "RFC 8294: Common YANG Data Types for the Routing Area";
  }
  import ietf-interfaces {
    prefix if;
    reference
      "RFC 8343: A YANG Data Model for Interface Management";
  }

  organization
```

```
    "IETF OPSAWG (Operations and Management Area Working Group)";
  contact
    "WG Web:   <https://datatracker.ietf.org/wg/opsawg/>
     WG List:  <mailto:opsawg@ietf.org>

     Author:   Samier Barguil
               <mailto:samier.barguilgiraldo.ext@telefonica.com>
     Editor:   Oscar Gonzalez de Dios
               <mailto:oscar.gonzalezdedios@telefonica.com>
     Editor:   Mohamed Boucadair
               <mailto:mohamed.boucadair@orange.com>
     Author:   Luis Angel Munoz
               <mailto:luis-angel.munoz@vodafone.com>
     Author:   Alejandro Aguado
               <mailto:alejandro.aguado_martin@nokia.com>";
  description
    "Этот модуль YANG задаёт базовую, ориентированную на сеть модель
     для настройки L3 VPN.

     Авторские права (Copyright (c) 2022) принадлежат IETF Trust и
     лицам, указанным как авторы. Все права защищены.

     Распространение и применение модуля в исходной или двоичной
     форме с изменениями или без таковых разрешено в соответствии с
     лицензией Simplified BSD License, изложенной в параграфе 4.c
     IETF Trust's Legal Provisions Relating to IETF Documents
     (https://trustee.ietf.org/license-info).

     Эта версия модуля YANG является частью RFC 9182, где правовые
     аспекты приведены более полно.";

  revision 2022-02-14 {
    description
      "Исходный выпуск.";
    reference
      "RFC 9182: A YANG Network Data Model for Layer 3 VPNs";
  }

  /* Свойства (возможности) */

  feature msdp {
    description
      "Указывает поддержку протокола MSDP в VPN.";
    reference
      "RFC 3618: Multicast Source Discovery Protocol (MSDP)";
  }

  /* Идентификаторы */

  identity address-allocation-type {
    description
      "Базовый идентификатор типа назначения адресов на канале
       PE-CE.";
  }

  identity provider-dhcp {
    base address-allocation-type;
    description
      "Сеть провайдера предоставляет клиенту услуги DHCP.";
  }

  identity provider-dhcp-relay {
    base address-allocation-type;
    description
      "Сеть провайдера предоставляет клиенту услуги DHCP relay.";
  }

  identity provider-dhcp-slaac {
    if-feature "vpn-common:ipv6";
    base address-allocation-type;
    description
      "Сеть провайдера предоставляет клиенту услуги DHCP и SLAAC).";
    reference
      "RFC 4862: IPv6 Stateless Address Autoconfiguration";
  }

  identity static-address {
    base address-allocation-type;
    description
      "Сеть провайдера предоставляет клиенту статический адрес IP.";
  }

  identity slaac {
    if-feature "vpn-common:ipv6";
    base address-allocation-type;
    description
```

```
        "Сеть провайдера использует IPv6 SLAAC для предоставления
         адресов клиентам.";
      reference
        "RFC 4862: IPv6 Stateless Address Autoconfiguration";
    }

    identity local-defined-next-hop {
      description
        "Базовый идентификатор для заданных локально next hop.";
    }

    identity discard {
      base local-defined-next-hop;
      description
        "Указывает отбрасывание трафика для соответствующего адресата.
         Например, это может создавать «черную дыру» (black-hole).";
    }

    identity local-link {
      base local-defined-next-hop;
      description
        "Считать трафик, адресованный в префикс заданного next-hop,
         как подключённый к локальному каналу.";
    }

    identity l2-tunnel-type {
      description
        "Базовый идентификатор для выбора туннеля L2 в доступе в VPN.";
    }

    identity pseudowire {
      base l2-tunnel-type;
      description
        "Завершение туннельного псевдопровода в доступе в VPN.";
    }

    identity vpls {
      base l2-tunnel-type;
      description
        "Завершение туннеля VPLS в доступе в VPN.";
    }

    identity vxlan {
      base l2-tunnel-type;
      description
        "Завершение туннеля VXLAN в доступе в VPN.";
    }

    /* Определения типов */

    typedef predefined-next-hop {
      type identityref {
        base local-defined-next-hop;
      }
      description
        "Предопределённый next-hop для локально созданных маршрутов.";
    }

    typedef area-address {
      type string {
        pattern '[0-9A-Fa-f]{2}(\.[0-9A-Fa-f]{4}){0,6}';
      }
      description
        "Задаёт формат адреса для области.";
    }

    /* Группировки */

    grouping vpn-instance-profile {
      description
        "Группировка для узлов данных, которые могут быть факторизованы
         на многих уровнях модели. Группировка может служить для
         определения базовых профилей на уровне сервиса VPN, которые
         будут указываться на уровне узла VPN и доступа в сеть VPN.";
      leaf local-as {
        if-feature "vpn-common:rtg-bgp";
        type inet:as-number;
        description
          "Номер AS провайдера, используемый при маршрутизации BGP.";
      }
      uses vpn-common:route-distinguisher;
      list address-family {
        key "address-family";
        description
          "Набор параметров на семейство адресов.";
        leaf address-family {
```

```
        type identityref {
          base vpn-common:address-family;
        }
        description
          "Семейство адресов (IPv4 и/или IPv6).";
      }
      container vpn-targets {
        description
          "Набор целей маршрутов для сопоставления при импорте
           и экспорте маршрутов VRF.";
        uses vpn-common:vpn-route-targets;
      }
      list maximum-routes {
        key "protocol";
        description
          "Максимальное число маршрутов для VRF.";
        leaf protocol {
          type identityref {
            base vpn-common:routing-protocol-type;
          }
          description
            "Протокол маршрутизации. Значение any может служить для
             указания пределов, применимых ко всем активным
             протоколам.";
        }
        leaf maximum-routes {
          type uint32;
          description
            "Максимальное число префиксов, которые VRF может
             воспринять для данного семейства адресов и протокола.";
        }
      }
    }
    container multicast {
      if-feature "vpn-common:multicast";
      description
        "Глобальные параметры групповой передачи.";
      leaf tree-flavor {
        type identityref {
          base vpn-common:multicast-tree-type;
        }
        description
          "Тип используемого дерева групповой передачи.";
      }
      container rp {
        description
          "Параметры точки встречи (RP).";
        container rp-group-mappings {
          description
            "Параметры сопоставления RP с группой.";
          list rp-group-mapping {
            key "id";
            description
              "Список сопоставлений RP с группами.";
            leaf id {
              type uint16;
              description
                "Уникальный идентификатор сопоставления.";
            }
            container provider-managed {
              description
                "Параметры для управляемой провайдером точки RP.";
              leaf enabled {
                type boolean;
                default "false";
                description
                  "true, если RP должна быть управляемым провайдером
                   узлом, false для управляемого клиентом узла.";
              }
              leaf rp-redundancy {
                type boolean;
                default "false";
                description
                  "true указывает, что нужен механизм резервирования
                   для RP.";
              }
              leaf optimal-traffic-delivery {
                type boolean;
                default "false";
                description
                  "true указывает, что сервис-провайдер (SP) должен
                   гарантировать использование оптимального пути для
                   трафика. SP может использовать архитектуру
                   переключения Anycast RP или RP-tree-to-SPT
                   (SPT - дерево кратчайшего пути).";
              }
```

```
            container anycast {
              when "../rp-redundancy = 'true' and
                    ../optimal-traffic-delivery = 'true'" {
                description
                  "Применимо лишь при активации резервирования RP и
                   доставки по оптимальному пути.";
              }
              description
                "PIM Anycast-RP parameters.";
              leaf local-address {
                type inet:ip-address;
                description
                  "Локальный адрес IP для PIM RP. Обычно
                   соответствует Router ID или основному адресу.";
              }
              leaf-list rp-set-address {
                type inet:ip-address;
                description
                  "Адрес IP набора маршрутизаторов RP.";
              }
            }
          }
          leaf rp-address {
            when "../provider-managed/enabled = 'false'" {
              description
                "Применяется для RP, не управляемой провайдером.";
            }
            type inet:ip-address;
            mandatory true;
            description
              "Адрес RP, управляемой клиентом.";
          }
          container groups {
            description
              "Multicast-группы, связанные с RP.";
            list group {
              key "id";
              description
                "Список multicast-групп.";
              leaf id {
                type uint16;
                description
                  "Идентификатор группы.";
              }
              choice group-format {
                mandatory true;
                description
                  "Выбор формата multicast-группы.";
                case group-prefix {
                  leaf group-address {
                    type inet:ip-prefix;
                    description
                      "Один префикс multicast-группы.";
                  }
                }
                case startend {
                  leaf group-start {
                    type inet:ip-address;
                    description
                      "Первый адрес в диапазоне адресов
                       multicast-группы.";
                  }
                  leaf group-end {
                    type inet:ip-address;
                    description
                      "Последний адрес в диапазоне адресов
                       multicast-группы.";
                  }
                }
              }
            }
          }
        }
      }
      container rp-discovery {
        description
          "Параметры обнаружения RP.";
        leaf rp-discovery-type {
          type identityref {
            base vpn-common:multicast-rp-discovery-type;
          }
          default "vpn-common:static-rp";
          description
            "Используемый тип обнаружения RP.";
        }
        container bsr-candidates {
```

```
            when "derived-from-or-self(../rp-discovery-type, "
               + "'vpn-common:bsr-rp')" {
              description
                "Применяется лишь при обнаружении типа bsr-rp.";
            }
            description
              "Контейнер для адресов кандидатов в BSR (Bootstrap
               Router — маршрутизатор начальной загрузки) клиента.";
            leaf-list bsr-candidate-address {
              type inet:ip-address;
              description
                "Адрес кандидата в BSR.";
            }
          }
        }
      }
      container igmp {
        if-feature "vpn-common:igmp and vpn-common:ipv4";
        description
          "Параметры, связанные с IGMP.";
        list static-group {
          key "group-addr";
          description
            "Статический источник/группа, связанный с сессией IGMP.";
          leaf group-addr {
            type rt-types:ipv4-multicast-group-address;
            description
              "Адрес IPv4 для группы.";
          }
          leaf source-addr {
            type rt-types:ipv4-multicast-source-address;
            description
              "Адрес отправителя IPv4 для группы.";
          }
        }
        leaf max-groups {
          type uint32;
          description
            "Максимальное число групп.";
        }
        leaf max-entries {
          type uint32;
          description
            "Максимальное число записей IGMP.";
        }
        leaf version {
          type identityref {
            base vpn-common:igmp-version;
          }
          default "vpn-common:igmpv2";
          description
            "Версия IGMP.";
          reference
            "RFC 1112: Host Extensions for IP Multicasting
             RFC 2236: Internet Group Management Protocol,
                       Version 2
             RFC 3376: Internet Group Management Protocol,
                       Version 3";
        }
      }
      container mld {
        if-feature "vpn-common:mld and vpn-common:ipv6";
        description
          "Связанные с MLD параметры.";
        list static-group {
          key "group-addr";
          description
            "Статический источник/группа, связанный с сессией MLD.";
          leaf group-addr {
            type rt-types:ipv6-multicast-group-address;
            description
              "Адрес IPv6 для группы.";
          }
          leaf source-addr {
            type rt-types:ipv6-multicast-source-address;
            description
              "Адрес источника IPv6 для группы.";
          }
        }
        leaf max-groups {
          type uint32;
          description
            "Максимальное число групп.";
        }
        leaf max-entries {
          type uint32;
```

```
            description
              "Максимальное число записей MLD.";
          }
          leaf version {
            type identityref {
              base vpn-common:mld-version;
            }
            default "vpn-common:mldv2";
            description
              "Версия протокола MLD.";
            reference
              "RFC 2710: Multicast Listener Discovery (MLD) for IPv6
               RFC 3810: Multicast Listener Discovery Version 2
                         (MLDv2) for IPv6";
          }
        }
        container pim {
          if-feature "vpn-common:pim";
          description
            "Применимо лишь с типом протокола pim.";
          leaf hello-interval {
            type rt-types:timer-value-seconds16;
            default "30";
            description
              "Интервал между сообщениями PIM Hello При infinity или
               not-set периодические сообщения Hello не передаются.";
            reference
              "RFC 7761: Protocol Independent Multicast - Sparse
                         Mode (PIM-SM): Protocol Specification
                         (Revised), Section 4.11
               RFC 8294: Common YANG Data Types for the Routing
                         Area";
          }
          leaf dr-priority {
            type uint32;
            default "1";
            description
              "Предпочтение при выборе назначенного маршрутизатора
               (Designated Router или DR). Большее значение указывает
               более высокий приоритет.";
            reference
              "RFC 7761: Protocol Independent Multicast - Sparse
                         Mode (PIM-SM): Protocol Specification
                         (Revised), Section 4.3.2";
          }
        }
      }
    }

    /* Основные блоки */
    /* l3vpn-ntw */

    container l3vpn-ntw {
      description
        "Основной контейнер для управления службами L3VPN.";
      container vpn-profiles {
        description
          "Набор профилей VPN, пригодных для указания в службе VPN.";
        uses vpn-common:vpn-profile-cfg;
      }
      container vpn-services {
        description
          "Контейнер для услуг VPN.";
        list vpn-service {
          key "vpn-id";
          description
            "List of VPN services.";
          uses vpn-common:vpn-description;
          leaf parent-service-id {
            type vpn-common:vpn-id;
            description
              "Указатель на родительскую службу, если она есть. Это
               может быть L3SM, запрос среза (slice), VPN+ и т. п.";
          }
          leaf vpn-type {
            type identityref {
              base vpn-common:service-type;
            }
            description
              "Указывает тип службы.";
          }
          leaf vpn-service-topology {
            type identityref {
              base vpn-common:vpn-topology;
            }
            default "vpn-common:any-to-any";
```

```
        description
          "Топология службы VPN.";
      }
      uses vpn-common:service-status;
      container vpn-instance-profiles {
        description
          "Контейнер для списка профилей экземпляров VPN.";
        list vpn-instance-profile {
          key "profile-id";
          description
            "Список профилей экземпляров VPN.";
          leaf profile-id {
            type string;
            description
              "Идентификатор профиля экземпляра VPN.";
          }
          leaf role {
            type identityref {
              base vpn-common:role;
            }
            default "vpn-common:any-to-any-role";
            description
              "Роль узла VPN в сети VPN.";
          }
          uses vpn-instance-profile;
        }
      }
      container underlay-transport {
        description
          "Контейнер для базового транспорта.";
        uses vpn-common:underlay-transport;
      }
      container external-connectivity {
        if-feature "vpn-common:external-connectivity";
        description
          "Контейнер для внешних соединений.";
        choice profile {
          description
            "Выбор профиля для внешних соединений.";
          case profile {
            leaf profile-name {
              type leafref {
                path "/l3vpn-ntw/vpn-profiles"
                   + "/valid-provider-identifiers"
                   + "/external-connectivity-identifier/id";
              }
              description
                "Имя профиля от провайдера для применения на
                 уровне службы VPN.";
            }
          }
        }
      }
      container vpn-nodes {
        description
          "Контейнер для узлов VPN.";
        list vpn-node {
          key "vpn-node-id";
          description
            "Список узлов VPN.";
          leaf vpn-node-id {
            type vpn-common:vpn-id;
            description
              "Идентификатор узла VPN.";
          }
          leaf description {
            type string;
            description
              "Текстовое описание узла VPN.";
          }
          leaf ne-id {
            type string;
            description
              "Уникальный идентификатор элемента сети, где
               реализован узел VPN.";
          }
          leaf local-as {
            if-feature "vpn-common:rtg-bgp";
            type inet:as-number;
            description
              "Номер AS у провайдера, применяемый, если клиенту
               нужна маршрутизация BGP.";
          }
          leaf router-id {
            type rt-types:router-id;
            description
```

```
        "32-битовое значение с разделением точками, служащее
         для указания узла внутри AS. Применяется для
         IPv4 и IPv6.";
    }
    container active-vpn-instance-profiles {
      description
        "Контейнер для активных профилей экземпляров VPN.";
      list vpn-instance-profile {
        key "profile-id";
        description
          "Список активных профилей экземпляров VPN.";
        leaf profile-id {
          type leafref {
            path "/l3vpn-ntw/vpn-services/vpn-service"
               + "/vpn-instance-profiles"
               + "/vpn-instance-profile/profile-id";
          }
          description
            "Активный профиль экземпляра VPN для узла.";
        }
        list router-id {
          key "address-family";
          description
            "Router ID по семействам адресов.";
          leaf address-family {
            type identityref {
              base vpn-common:address-family;
            }
            description
              "Семейство адресов, к которому относится
               Router ID.";
          }
          leaf router-id {
            type inet:ip-address;
            description
              "В качестве router-id может служить адрес IPv4
               или IPv6. Это может применяться, например, для
               настройки адреса IPv6 как Router ID, когда
               такая возможность поддерживается базовыми
               маршрутизаторами. В таком случае настроенное
               значение переопределяет базовое, заданное на
               уровне узла VPN.";
          }
        }
        uses vpn-instance-profile;
      }
    }
    container msdp {
      if-feature "msdp";
      description
        "Параметры, относящиеся к MSDP.";
      leaf peer {
        type inet:ipv4-address;
        description
          "Адрес IPv4 партнёра MSDP.";
      }
      leaf local-address {
        type inet:ipv4-address;
        description
          "Локальный адрес IPv4, который должен настраиваться
           на узле.";
      }
      uses vpn-common:service-status;
    }
    uses vpn-common:vpn-components-group;
    uses vpn-common:service-status;
    container vpn-network-accesses {
      description
        "Список доступов в сеть.";
      list vpn-network-access {
        key "id";
        description
          "Список доступов в сеть.";
        leaf id {
          type vpn-common:vpn-id;
          description
            "Идентификатор доступа в сеть.";
        }
        leaf interface-id {
          type string;
          description
            "Идентификатор физического или логического
             интерфейса. Идентификация субинтерфейсов
             обеспечивается на уровне соединения и
             соединения IP.";
        }
```

```
        leaf description {
          type string;
          description
            "Текстовое описание доступа в сеть.";
        }
        leaf vpn-network-access-type {
          type identityref {
            base vpn-common:site-network-access-type;
          }
          default "vpn-common:point-to-point";
          description
            "Тип соединения, например point to point.";
        }
        leaf vpn-instance-profile {
          type leafref {
            path "/l3vpn-ntw/vpn-services/vpn-service"
               + "/vpn-nodes/vpn-node"
               + "/active-vpn-instance-profiles"
               + "/vpn-instance-profile/profile-id";
          }
          description
            "Идентификатор активного профиля экземпляра VPN";
        }
        uses vpn-common:service-status;
        container connection {
          description
            "Протоколы и параметры L2, требуемые для связи
             между PE и CE.";
          container encapsulation {
            description
              "Контейнер для инкапсуляции L2.";
            leaf type {
              type identityref {
                base vpn-common:encapsulation-type;
              }
              default "vpn-common:priority-tagged";
              description
                "Тип инкапсуляции. Для интерфейса с тегами
                 по умолчанию применяется priority-tagged.";
            }
            container dot1q {
              when "derived-from-or-self(../type, "
                 + "'vpn-common:dot1q')" {
                description
                  "Применяется лишь к интерфейсам с тегами
                   dot1q.";
              }
              description
                "Tagged interface.";
              leaf tag-type {
                type identityref {
                  base vpn-common:tag-type;
                }
                default "vpn-common:c-vlan";
                description
                  "Тип тега. По умолчанию c-vlan.";
              }
              leaf cvlan-id {
                type uint16 {
                  range "1..4094";
                }
                description
                  "Идентификатор VLAN.";
              }
            }
            container priority-tagged {
              when "derived-from-or-self(../type, "
                 + "'vpn-common:priority-tagged')" {
                description
                  "Применяется только с теговыми интерфейсами
                   типа priority-tagged.";
              }
              description
                "Priority tagged.";
              leaf tag-type {
                type identityref {
                  base vpn-common:tag-type;
                }
                default "vpn-common:c-vlan";
                description
                  "Тип тега. По умолчанию c-vlan.";
              }
            }
            container qinq {
              when "derived-from-or-self(../type, "
                 + "'vpn-common:qinq')" {
```

```
          description
            "Применяется только с теговыми интерфейсами
             типа qinq.";
        }
        description
          "Includes QinQ parameters.";
        leaf tag-type {
          type identityref {
            base vpn-common:tag-type;
          }
          default "vpn-common:s-c-vlan";
          description
            "Тип тега.";
        }
        leaf svlan-id {
          type uint16;
          mandatory true;
          description
            "Идентификатор Service VLAN (S-VLAN).";
        }
        leaf cvlan-id {
          type uint16;
          mandatory true;
          description
            "Идентификатор Customer VLAN (C-VLAN).";
        }
      }
    }
    choice l2-service {
      description
        "Услуга связности L2 может быть предоставлена
         указателем на L2VPN или заданием туннеля L2.";
      container l2-tunnel-service {
        description
          "Определяет завершение туннеля L2. Применимо
           лишь при необходимости туннеля. Возможны
           значения pseudowire, vpls, vxlan. При
           необходимости можно задать иные значения.";
        leaf type {
          type identityref {
            base l2-tunnel-type;
          }
          description
            "Вариант завершение туннеля для каждого
             доступа в сеть VPN.";
        }
        container pseudowire {
          when "derived-from-or-self(../type, "
             + "'pseudowire')" {
            description
              "Применимо лишь к сервису L2 pseudowire";
          }
          description
            "Параметры завершения псевдопровода.";
          leaf vcid {
            type uint32;
            description
              "Идентификатор псевдопровода (PW) или
               виртуального устройства (VC).";
          }
          leaf far-end {
            type union {
              type uint32;
              type inet:ip-address;
            }
            description
              "Указание соседа.";
            reference
              "RFC 8077: Pseudowire Setup and
                         Maintenance Using the Label
                         Distribution Protocol
                         (LDP), Section 6.1";
          }
        }
        container vpls {
          when "derived-from-or-self(../type, "
             + "'vpls')" {
            description
              "Применимо лишь для сервиса L2 vpls.";
          }
          description
            "Параметры завершения VPLS.";
          leaf vcid {
            type uint32;
            description
              "VC identifier.";
```

```
        }
        leaf-list far-end {
          type union {
            type uint32;
            type inet:ip-address;
          }
          description
            "Указание соседа.";
        }
      }
      container vxlan {
        when "derived-from-or-self(../type, "
           + "'vxlan')" {
          description
            "Применимо лишь для сервиса L2 vxlan.";
        }
        description
          "Параметры завершения VXLAN.";
        leaf vni-id {
          type uint32;
          mandatory true;
          description
            "Идентификатор сети VXLAN (VNI).";
        }
        leaf peer-mode {
          type identityref {
            base vpn-common:vxlan-peer-mode;
          }
          default "vpn-common:static-mode";
          description
            "Режим доступа VXLAN. По умолчанию принят
             партнёрский режим static-mode.";
        }
        leaf-list peer-ip-address {
          type inet:ip-address;
          description
            "Список IP-адресов партнёра.";
        }
      }
    }
    case l2vpn {
      leaf l2vpn-id {
        type vpn-common:vpn-id;
        description
          "Служба L2VPN, связанная с интегрированным
           интерфейсом моста и маршрутизации (IRB).";
      }
    }
  }
  leaf l2-termination-point {
    type string;
    description
      "Ссылка на локальную точку завершения L2, такую
       как субинтерфейс L2.";
  }
  leaf local-bridge-reference {
    type string;
    description
      "Ссылка на локальный мост, например, для
       реализации, которой нужен внутренний мост. Это
       может быть ссылка на локальный домен мостов.";
  }
  leaf bearer-reference {
    if-feature "vpn-common:bearer-reference";
    type string;
    description
      "Внутренняя ссылка для сервис-провайдера, чтобы
       указать опорный канал, связанный с VPN.";
  }
  container lag-interface {
    if-feature "vpn-common:lag-interface";
    description
      "Контейнер для настройки атрибутов интерфейса
       объединения каналов (LAG).";
    leaf lag-interface-id {
      type string;
      description
        "Идентификатор интерфейса LAG.";
    }
    container member-link-list {
      description
        "Контейнер для элементов группы каналов.";
      list member-link {
        key "name";
        description
          "Канал из группы.";
```

```
          leaf name {
            type string;
            description
              "Имя канала из группы.";
          }
        }
      }
    }
  }
  container ip-connection {
    description
      "Параметры соединения IP.";
    leaf l3-termination-point {
      type string;
      description
        "Ссылка на локальную точку завершения L3, такую
         как интерфейс домена мостов.";
    }
    container ipv4 {
      if-feature "vpn-common:ipv4";
      description
        "Параметры, связанные с IPv4.";
      leaf local-address {
        type inet:ipv4-address;
        description
          "Адрес IP на интерфейсе провайдера.";
      }
      leaf prefix-length {
        type uint8 {
          range "0..32";
        }
        description
          "Размер префикса подсети в битах, применяемый
           для локального и клиентского адреса.";
      }
      leaf address-allocation-type {
        type identityref {
          base address-allocation-type;
        }
        must "not(derived-from-or-self(current(), "
           + "'slaac') or "
           + "derived-from-or-self(current(), "
           + "'provider-dhcp-slaac'))" {
          error-message "SLAAC is only applicable "
                      + "to IPv6.";
        }
        description
          "Задаёт выделение адресов для сайта партнёра.
           Если тип выделения не указан, адресация IPv4
           не поддерживается.";
      }
      choice allocation-type {
        description
          "Выбор типа выделения адресов IPv4.";
        case provider-dhcp {
          description
            "Параметры для адресов, полученных по DHCP,
             предоставляемому оператором.";
          leaf dhcp-service-type {
            type enumeration {
              enum server {
                description
                  "Локальный сервер DHCP.";
              }
              enum relay {
                description
                  "Локальный ретранслятор DHCP. Запросы
                   передаются серверу провайдера.";
              }
            }
            description
              "Тип службы DHCP для этого доступа.";
          }
          choice service-type {
            description
              "Выбор по типу услуги DHCP.";
            case relay {
              description
                "Контейнер для списка серверов DHCP у
                 провайдера (для dhcp-service-type
                 relay).";
              leaf-list server-ip-address {
                type inet:ipv4-address;
                description
                  "Адрес IPv4 сервера DHCP у провайдера
                   для использования локальным
```

```
                ретранслятором DHCP.";
            }
          }
          case server {
            description
              "Выбор для назначения адресов когда
               разрешён локальный сервер DHCP.";
            choice address-assign {
              default "number";
              description
                "Выбор способа назначения адресов
                 IPv4.";
              case number {
                leaf number-of-dynamic-address {
                  type uint16;
                  default "1";
                  description
                    "Число адресов IP, выделяемых
                     клиенту для этого доступа.";
                }
              }
              case explicit {
                container customer-addresses {
                  description
                    "Контейнер для адресов клиента,
                     назначаемых сервером DHCP.";
                  list address-pool {
                    key "pool-id";
                    description
                      "Описание адресов IP,
                       назначаемых DHCP.

                       При наличии лишь start-address
                       выделяется только один адрес.

                       Когда заданы start-address и
                       end-address выделяется весь
                       диапазон вместе с границами.";
                    leaf pool-id {
                      type string;
                      description
                        "Идентификатор пула адресов
                         от start-address до
                         end-address, включительно.";
                    }
                    leaf start-address {
                      type inet:ipv4-address;
                      mandatory true;
                      description
                        "Первый адрес пула.";
                    }
                    leaf end-address {
                      type inet:ipv4-address;
                      description
                        "Последний адрес пула.";
                    }
                  }
                }
              }
            }
          }
        }
        case dhcp-relay {
          description
            "Ретранслятор DHCP от оператора";
          container customer-dhcp-servers {
            description
              "Контейнер для списка клиентских
               серверов DHCP.";
            leaf-list server-ip-address {
              type inet:ipv4-address;
              description
                "Адрес IPv4 клиентского сервера DHCP.";
            }
          }
        }
        case static-addresses {
          description
            "Списки используемых адресов IPv4.";
          leaf primary-address {
            type leafref {
              path "../address/address-id";
            }
            description
              "Основной адрес соединения.";
```

```
        }
        list address {
          key "address-id";
          description
            "Списки используемых адресов IPv4.";
          leaf address-id {
            type string;
            description
              "Идентификатор статического адреса
               IPv4.";
          }
          leaf customer-address {
            type inet:ipv4-address;
            description
              "Адрес IPv4 на стороне клиента.";
          }
        }
      }
    }
  }
  container ipv6 {
    if-feature "vpn-common:ipv6";
    description
      "Параметры, связанные с IPv6.";
    leaf local-address {
      type inet:ipv6-address;
      description
        "Адрес IPv6 на стороне провайдера.";
    }
    leaf prefix-length {
      type uint8 {
        range "0..128";
      }
      description
        "Размер префикса подсети в битах для
         локальных и клиентских адресов.";
    }
    leaf address-allocation-type {
      type identityref {
        base address-allocation-type;
      }
      description
        "Задаёт способ назначения адресов. Если
         тип не задан, адреса IPv6 не выделяются.";
    }
    choice allocation-type {
      description
        "Выбор по типу назначения адресов IPv6.";
      container provider-dhcp {
        when "derived-from-or-self(../address-allo"
           + "cation-type, 'provider-dhcp') or "
           + "derived-from-or-self(../address-allo"
           + "cation-type, 'provider-dhcp-slaac')" {
          description
            "Применимо лишь при назначении адресов
             DHCPv6, обеспечиваемом оператором.";
        }
        description
          "Параметры, относящиеся к адресам от DHCP";
        leaf dhcp-service-type {
          type enumeration {
            enum server {
              description
                "Локальный сервер DHCPv6.";
            }
            enum relay {
              description
                "Ретранслятор DHCPv6.";
            }
          }
          description
            "Тип службы DHCPv6, включённой для этого
             доступа.";
        }
        choice service-type {
          description
            "Выбор по типу сервиса DHCPv6.";
          case relay {
            leaf-list server-ip-address {
              type inet:ipv6-address;
              description
                "Адреса IPv6 провайдерского сервера
                 DHCPv6.";
            }
          }
          case server {
```

```
        choice address-assign {
          default "number";
          description
            "Выбор назначения префиксов IPv6
             сервером DHCPv6.";
          case number {
            leaf number-of-dynamic-address {
              type uint16;
              default "1";
              description
                "Число префиксов IPv6, выделяемых
                 клиенту для этого доступа.";
            }
          }
          case explicit {
            container customer-addresses {
              description
                "Контейнер для клиентских адресов
                 IPv6, выделенных через DHCPv6.";
              list address-pool {
                key "pool-id";
                description
                  "Описывает адреса IPv6,
                   выделенные DHCPv6.

                   При наличии лишь start-address
                   выделяется только один адрес.

                   При указании start-address и
                   end-address выделяется
                   диапазон адресов с границами.";
                leaf pool-id {
                  type string;
                  description
                    "Идентификатор пула для блока
                     start-address -
                     end-address.";
                }
                leaf start-address {
                  type inet:ipv6-address;
                  mandatory true;
                  description
                    "Первый адрес пула.";
                }
                leaf end-address {
                  type inet:ipv6-address;
                  description
                    "Последний адрес пула.";
                }
              }
            }
          }
        }
      }
    }
    case dhcp-relay {
      description
        "Ретранслятор DHCPv6 от операторома.";
      container customer-dhcp-servers {
        description
          "Контейнер для списка клиентских
           серверов DHCP.";
        leaf-list server-ip-address {
          type inet:ipv6-address;
          description
            "Адреса IP клиентского сервера
             DHCPv6.";
        }
      }
    }
    case static-addresses {
      description
        "Связанные с IPv6 параметры для
         статического назначения.";
      leaf primary-address {
        type leafref {
          path "../address/address-id";
        }
        description
          "Основной адрес соединения.";
      }
      list address {
        key "address-id";
        description
          "Используемые адреса IPv6.";
```

```
            leaf address-id {
              type string;
              description
                "Идентификатор адреса IPv6.";
            }
            leaf customer-address {
              type inet:ipv6-address;
              description
                "An IPv6 address of the customer
                 side.";
            }
          }
        }
      }
    }
  }
  container routing-protocols {
    description
      "Задаёт протоколы маршрутизации.";
    list routing-protocol {
      key "id";
      description
        "Список протоколов маршрутизации на канале
         CE-PE. Список может дополняться.";
      leaf id {
        type string;
        description
          "Уникальный идентификатор протокола
           маршрутизации.";
      }
      leaf type {
        type identityref {
          base vpn-common:routing-protocol-type;
        }
        description
          "Тип протокола маршрутизации.";
      }
      list routing-profiles {
        key "id";
        description
          "Профили маршрутизации.";
        leaf id {
          type leafref {
            path "/l3vpn-ntw/vpn-profiles"
               + "/valid-provider-identifiers"
               + "/routing-profile-identifier/id";
          }
          description
            "Профиль маршрутизации для применения.";
        }
        leaf type {
          type identityref {
            base vpn-common:ie-type;
          }
          description
            "Импорт, экспорт или оба.";
        }
      }
      container static {
        when "derived-from-or-self(../type, "
           + "'vpn-common:static-routing')" {
          description
            "Применимо лишь для статической
             маршрутизации.";
        }
        description
          "Настройка статической маршрутизации.";
          routing.";
        container cascaded-lan-prefixes {
          description
            "Префиксы ЛВС от клиента.";
          list ipv4-lan-prefixes {
            if-feature "vpn-common:ipv4";
            key "lan next-hop";
            description
              "Список префиксов ЛВС для сайта.";
            leaf lan {
              type inet:ipv4-prefix;
              description
                "Префиксы ЛВС.";
            }
            leaf lan-tag {
              type string;
              description
                "Внутренний тег, применяемый политикой
                 VPN.";
```

```
        }
        leaf next-hop {
          type union {
            type inet:ip-address;
            type predefined-next-hop;
          }
          description
            "Значение next hop для статического
             маршрута - адрес IP или
             предопределённый тип next-hop
             например, discard или local-link).";
        }
        leaf bfd-enable {
          if-feature "vpn-common:bfd";
          type boolean;
          description
            "Управляет включением BFD.";
        }
        leaf metric {
          type uint32;
          description
            "Указывает метрику, связанную со
             статическим маршрутом.";
        }
        leaf preference {
          type uint32;
          description
            "Указывает предпочтение, связанное со
             статическим маршрутом.";
        }
        uses vpn-common:service-status;
      }
      list ipv6-lan-prefixes {
        if-feature "vpn-common:ipv6";
        key "lan next-hop";
        description
          "Список префиксов ЛВС для сайта.";
        leaf lan {
          type inet:ipv6-prefix;
          description
            "Префиксы ЛВС.";
        }
        leaf lan-tag {
          type string;
          description
            "Внутренний тег для правил VPN.";
        }
        leaf next-hop {
          type union {
            type inet:ip-address;
            type predefined-next-hop;
          }
          description
            "Значение next hop для статического
             маршрута - адрес IP или
             предопределённый тип next-hop
             например, discard или local-link).";
        }
        leaf bfd-enable {
          if-feature "vpn-common:bfd";
          type boolean;
          description
            "Управляет включением BFD.";
        }
        leaf metric {
          type uint32;
          description
            "Указывает метрику, связанную со
             статическим маршрутом.";
        }
        leaf preference {
          type uint32;
          description
            "Указывает предпочтение, связанное со
             статическим маршрутом.";
        }
        uses vpn-common:service-status;
      }
    }
  }
  container bgp {
    when "derived-from-or-self(../type, "
       + "'vpn-common:bgp-routing')" {
      description
        "Применимо лишь для протокола BGP.";
    }
```

```
description
  "Конфигурация, связанная с BGP.";
leaf description {
  type string;
  description
    "Описание сессии BGP, предназначенное
     для диагностики. Семантика описания
     зависит от реализации.";
}
leaf local-as {
  type inet:as-number;
  description
    "Локальный номер AS (ASN), если он не
     совпадает с ASN, заданным на уровне
     узла VPN.";
}
leaf peer-as {
  type inet:as-number;
  mandatory true;
  description
    "ASN клиента, который запрашивает
     маршрутизацию BGP.";
}
leaf address-family {
  type identityref {
    base vpn-common:address-family;
  }
  description
    "Активируемые семейства адресов. Значение
     dual-stack активирует IPv4 и IPv6.";
}
leaf local-address {
  type union {
    type inet:ip-address;
    type if:interface-ref;
  }
  description
    "Локальный адрес IP для использования в
     транспортной сессии BGP. Это может быть
     адрес IP или ссылка на интерфейс.";
}
leaf-list neighbor {
  type inet:ip-address;
  description
    "IP-адреса соседа BGP. Можно указывать
     IPv4 и IPv6, если будут созданы сессии
     для IPv4 и IPv6.";
}
leaf multihop {
  type uint8;
  description
    "Число интервалов пересылки IP (hop),
     разрешённых между данным соседом BGP
     и PE.";
}
leaf as-override {
  type boolean;
  default "false";
  description
    "Управляет переопределением ASN, т. е.
     заменой ASN, заданного в атрибуте AS_PATH
     от клиента локальным номером ASN.";
}
leaf allow-own-as {
  type uint8;
  default "0";
  description
    "Максимальное число включений ASN
     провайдера в AS_PATH, при котором атрибут
     будет отклонен.";
}
leaf prepend-global-as {
  type boolean;
  default "false";
  description
    "В некоторых случаях ASN, заданный на
     уровне узла VPN может отличаться от ASN,
     настроенного на уровне доступа в сеть VPN.
     Когда такие ASN представлены, оба
     помещаются в начало обновлений маршрута
     BGP для этого доступа. Для запрета такого
     поведения нужно установить в
     prepend-global-as значение false и ASN от
     уровня узла VPN не будет включаться в
     обновления маршрута BGP.";
}
```

```
leaf send-default-route {
  type boolean;
  default "false";
  description
    "Управляет анонсированием принятых по
     умолчанию маршрутов партнеру.";
}
leaf site-of-origin {
  when "../address-family = 'vpn-common:ipv4' "
     + "or 'vpn-common:dual-stack'" {
    description
      "Применяется лишь при активном IPv4.";
  }
  type rt-types:route-origin;
  description
    "Атрибут Site of Origin кодируется как
     Route Origin Extended Community. Это
     служит для однозначного указания набора
     маршрутов, изученного от сайта через
     конкретное соединение CE-PE и служащего
     для предотвращения маршрутных петель.";
  reference
    "RFC 4364: BGP/MPLS IP Virtual Private
               Networks (VPNs), Section 7";
}
leaf ipv6-site-of-origin {
  when "../address-family = 'vpn-common:ipv6' "
     + "or 'vpn-common:dual-stack'" {
    description
      "Применяется лишь при активном IPv6.";
  }
  type rt-types:ipv6-route-origin;
  description
    "Атрибут IPv6 Site of Origin кодируется как
     IPv6 Route Origin Extended Community. Это
     однозначно указывает набор маршрутов,
     изученный с сайта через сведения VRF.";
  reference
    "RFC 5701: IPv6 Address Specific BGP
               Extended Community
               Attribute";
}
list redistribute-connected {
  key "address-family";
  description
    "Указывает правила (по семействам адресов)
     для подключённых маршрутов.";
  leaf address-family {
    type identityref {
      base vpn-common:address-family;
    }
    description
      "Семейство адресов.";
  }
  leaf enable {
    type boolean;
    description
      "Разрешает распространение подключённых
       маршрутов.";
  }
}
container bgp-max-prefix {
  description
    "Задаёт поведение при достижении
     максимального числа префиксов.";
  leaf max-prefix {
    type uint32;
    default "5000";
    description
      "Максимальное число префиксов BGP,
       разрешённое в сессии BGP. Это позволяет
       контролировать число полученных от
       соседа префиксов. При достижении предела
       выполняется действие, указанное листом
       violate-action.";
    reference
      "RFC 4271: A Border Gateway Protocol 4
                 (BGP-4), Section 8.2.2";
  }
  leaf warning-threshold {
    type decimal64 {
      fraction-digits 5;
      range "0..100";
    }
    units "percent";
    default "75";
```

```
        description
          "При превышении порога передаётся
           уведомление.";
      }
      leaf violate-action {
        type enumeration {
          enum warning {
            description
              "Партнёру передаётся предупреждение";
          }
          enum discard-extra-paths {
            description
              "Избыточные пути отбрасываются.";
          }
          enum restart {
            description
              "Сессия BGP перезапускается после
               заданного интервала.";
          }
        }
        description
          "Если для соседа BGP превышен порог
           max-prefix, выполняется действие,
           указанное violate-action.";
      }
      leaf restart-timer {
        type uint32;
        units "seconds";
        description
          "Интервал времени, после которого сессия
           BGP организуется заново.";
      }
    }
    container bgp-timers {
      description
        "Два таймера BGP, которые могут быть заданы
         при организации услуги VPN с BGP как
         протокола маршрутизации CE-PE.";
      leaf keepalive {
        type uint16 {
          range "0..21845";
        }
        units "seconds";
        default "30";
        description
          "Интервал передачи сообщений KEEPALIVE
           между PE и партнёром BGP. Значение 0
           отключает передачу KEEPALIVE.
           Предлагается устанавливать максимальный
           интервал между KEEPALIVE в 1/3 интервала
           Hold Time.";
        reference
          "RFC 4271: A Border Gateway Protocol 4
                     (BGP-4), Section 4.4";
      }
      leaf hold-time {
        type uint16 {
          range "0 | 3..65535";
        }
        units "seconds";
        default "90";
        description
          "Максимальное число секунд между приёмом
           последовательных сообщений KEEPALIVE и/или
           UPDATE от партнёра. Время удержания может
           быть 0 или не меньше 3 секунд.";
        reference
          "RFC 4271: A Border Gateway Protocol 4
                     (BGP-4), Section 4.2";
      }
    }
    container authentication {
      description
        "Контейнер для параметров аутентификации
         BGP между PE и CE.";
      leaf enable {
        type boolean;
        default "false";
        description
          "Управляет применением аутентификации.";
      }
      container keying-material {
        when "../enable = 'true'";
        description
          "Контейнер для описания защиты сессии BGP
           между PE и CE.";
```

```
        choice option {
          description
            "Выбор вариантов аутентификации.";
          case ao {
            description
              "Опция TCP-AO.";
            reference
              "RFC 5925: The TCP Authentication
                         Option";
            leaf enable-ao {
              type boolean;
              description
                "Включение TCP-AO.";
            }
            leaf ao-keychain {
              type key-chain:key-chain-ref;
              description
                "Ссылка на цепочку ключей TCP-AO.";
              reference
                "RFC 8177: YANG Data Model for
                           Key Chains";
            }
          }
          case md5 {
            description
              "Применение MD5 для защиты сессии.";
            reference
              "RFC 4364: BGP/MPLS IP Virtual
                         Private Networks
                         (VPNs), Section 13.2";
            leaf md5-keychain {
              type key-chain:key-chain-ref;
              description
                "Ссылка на цепочку ключей MD5.";
              reference
                "RFC 8177: YANG Data Model for
                           Key Chains";
            }
          }
          case explicit {
            leaf key-id {
              type uint32;
              description
                "Идентификатор ключа.";
            }
            leaf key {
              type string;
              description
                "Ключ аутентификации BGP. Эта
                 модель поддерживает лишь ключи,
                 представимые строками ASCII.";
            }
            leaf crypto-algorithm {
              type identityref {
                base key-chain:crypto-algorithm;
              }
              description
                "Криптографический алгоритм,
                 связанный с ключом.";
            }
          }
          case ipsec {
            description
              "Ссылка на защищённую связь IKE SA.";
            leaf sa {
              type string;
              description
                "Заданное администратором имя SA.";
            }
          }
        }
      }
    }
    uses vpn-common:service-status;
  }
  container ospf {
    when "derived-from-or-self(../type, "
       + "'vpn-common:ospf-routing')" {
      description
        "Применимо лишь с протоколом OSPF.";
    }
    description
      "Конфигурация, связанная с OSPF.";
    leaf address-family {
      type identityref {
        base vpn-common:address-family;
```

```
    }
    description
      "Указывает активагию IPv4, IPv6 или обоих";
  }
  leaf area-id {
    type yang:dotted-quad;
    mandatory true;
    description
      "Area ID.";
    reference
      "RFC 4577: OSPF as the Provider/Customer
                 Edge Protocol for BGP/MPLS IP
                 Virtual Private Networks
                 (VPNs), Section 4.2.3
       RFC 6565: OSPFv3 as a Provider Edge to
                 Customer Edge (PE-CE) Routing
                 Protocol, Section 4.2";
  }
  leaf metric {
    type uint16;
    default "1";
    description
      "Метрика канала PE-CE, применяемая при
       расчёте состояния маршрутизации и выборе
       пути.";
  }
  container sham-links {
    if-feature "vpn-common:rtg-ospf-sham-link";
    description
      "Список фиктивных (sham) каналов.";
    reference
      "RFC 4577: OSPF as the Provider/Customer
                 Edge Protocol for BGP/MPLS IP
                 Virtual Private Networks
                 (VPNs), Section 4.2.7
       RFC 6565: OSPFv3 as a Provider Edge to
                 Customer Edge (PE-CE) Routing
                 Protocol, Section 5";
    list sham-link {
      key "target-site";
      description
        "Создает sham-канал с другим сайтом.";
      leaf target-site {
        type string;
        description
          "Целевой сайт для sham-канала, заданный
           идентификатором.";
      }
      leaf metric {
        type uint16;
        default "1";
        description
          "Метрика sham-канала, применяемая при
           расчёте состояния маршрутизации и
           выборе пути. По умолчанию 1.";
        reference
          "RFC 4577: OSPF as the
                     Provider/Customer Edge
                     Protocol for BGP/MPLS IP
                     Virtual Private Networks
                     (VPNs), Section 4.2.7.3
           RFC 6565: OSPFv3 as a Provider Edge
                     to Customer Edge (PE-CE)
                     Routing Protocol,
                     Section 5.2";
      }
    }
  }
  leaf max-lsa {
    type uint32 {
      range "1..4294967294";
    }
    description
      "Максимальное число LSA, воспринимаемых
       экземпляром OSPF.";
  }
  container authentication {
    description
      "Настройка аутентификации.";
    leaf enable {
      type boolean;
      default "false";
      description
        "Включает и отключает аутентификацию.";
    }
    container keying-material {
```

```
          when "../enable = 'true'";
          description
            "Контроллер для описания защиты сессии
             OSPF между CE и PE.";
          choice option {
            description
              "Опции аутентификации OSPF.";
            case auth-key-chain {
              leaf key-chain {
                type key-chain:key-chain-ref;
                description
                  "Имя цепочки ключей.";
              }
            }
            case auth-key-explicit {
              leaf key-id {
                type uint32;
                description
                  "Идентификатор ключа.";
              }
              leaf key {
                type string;
                description
                  "Ключ аутентификации OSPF. Эта
                   модель поддерживает лишь ключи,
                   представимые строками ASCII.";
              }
              leaf crypto-algorithm {
                type identityref {
                  base key-chain:crypto-algorithm;
                }
                description
                  "Криптографический алгоритм,
                   связанный с ключом.";
              }
            }
            case ipsec {
              leaf sa {
                type string;
                description
                  "Заданное администратором имя SA.";
                reference
                  "RFC 4552: Authentication/
                              Confidentiality for
                              OSPFv3";
              }
            }
          }
        }
      }
      uses vpn-common:service-status;
    }
    container isis {
      when "derived-from-or-self(../type, "
         + "'vpn-common:isis-routing')" {
        description
          "Применимо лишь для протокола IS-IS.";
      }
      description
        "Конфигурация, связанная с IS-IS.";
      leaf address-family {
        type identityref {
          base vpn-common:address-family;
        }
        description
          "Активация IPv4, IPv6 или обоих.";
      }
      leaf area-address {
        type area-address;
        mandatory true;
        description
          "Адрес области.";
      }
      leaf level {
        type identityref {
          base vpn-common:isis-level;
        }
        description
          "level-1, level-2, level-1-2.";
        reference
          "RFC 9181: A Common YANG Data Model for
                     Layer 2 and Layer 3 VPNs";
      }
      leaf metric {
        type uint16;
        default "1";
```

```
        description
          "Метрика канала PE-CE, применяемая при
           расчёте состояния маршрутизации и выборе
           пути.";
      }
      leaf mode {
        type enumeration {
          enum active {
            description
              "Интерфейс передаёт или принимает
               пакеты управления протокола IS-IS.";
          }
          enum passive {
            description
              "Отключает передачу обновлений IS-IS
               через указанный интерфейс.";
          }
        }
        default "active";
        description
          "Тип режима интерфейса IS-IS.";
      }
      container authentication {
        description
          "Настройка аутентификации.";
        leaf enable {
          type boolean;
          default "false";
          description
            "Включает и отключает аутентификацию.";
        }
        container keying-material {
          when "../enable = 'true'";
          description
            "Контейнер для описания защиты сессии
             IS-IS между CE и PE.";
          choice option {
            description
              "Опции аутентификации IS-IS.";
            case auth-key-chain {
              leaf key-chain {
                type key-chain:key-chain-ref;
                description
                  "Имя цепочки ключей.";
              }
            }
            case auth-key-explicit {
              leaf key-id {
                type uint32;
                description
                  "Идентификатор ключа.";
              }
              leaf key {
                type string;
                description
                  "Ключ аутентификации IS-IS. Эта
                   модель поддерживает лишь ключи,
                   представимые строками ASCII.";
              }
              leaf crypto-algorithm {
                type identityref {
                  base key-chain:crypto-algorithm;
                }
                description
                  "Криптографический алгоритм,
                   связанный с ключом.";
              }
            }
          }
        }
      }
      uses vpn-common:service-status;
    }
    container rip {
      when "derived-from-or-self(../type, "
         + "'vpn-common:rip-routing')" {
        description
          "Применимо лишь для протокола RIP. Для IPv4
           предполагается протокол RIP версии 2.";
      }
      description
        "Конфигурация, связанная с RIP.";
      leaf address-family {
        type identityref {
          base vpn-common:address-family;
        }
```

```
    description
      "Активация IPv4, IPv6 или обоих.";
  }
  container timers {
    description
      "Таймеры RIP.";
    reference
      "RFC 2453: RIP Version 2";
    leaf update-interval {
      type uint16 {
        range "1..32767";
      }
      units "seconds";
      default "30";
      description
        "Время обновления RIP, т. е. интервал,
         для которого передаётся обновление RIP";
    }
    leaf invalid-interval {
      type uint16 {
        range "1..32767";
      }
      units "seconds";
      default "180";
      description
        "Интервал, по истечение которого маршрут
         считается недействительным, если нет
         обновлений. Это значение по меньшей мере
         втрое больше update-interval.";
    }
    leaf holddown-interval {
      type uint16 {
        range "1..32767";
      }
      units "seconds";
      default "180";
      description
        "Интервал перед освобождением лучших
         маршрутов.";
    }
    leaf flush-interval {
      type uint16 {
        range "1..32767";
      }
      units "seconds";
      default "240";
      description
        "Таймер очистки RIP, т. е. время, которое
         должно пройти, прежде чем маршрут будет
         удалён из таблицы маршрутизации.";
    }
  }
  leaf default-metric {
    type uint8 {
      range "0..16";
    }
    default "1";
    description
      "Задаёт принятую по умолчанию метрику.";
  }
  container authentication {
    description
      "Настройка аутентификации.";
    leaf enable {
      type boolean;
      default "false";
      description
        "Включает и отключает аутентификацию.";
    }
    container keying-material {
      when "../enable = 'true'";
      description
        "Контейнер для описания защиты сессии RIP
         между CE и PE.";
      choice option {
        description
          "Задаёт схему аутентификации.";
        case auth-key-chain {
          leaf key-chain {
            type key-chain:key-chain-ref;
            description
              "Имя цепочки ключей.";
          }
        }
        case auth-key-explicit {
          leaf key {
```

```
                    type string;
                    description
                      "Ключ аутентификации RIP. Эта
                       модель поддерживает лишь ключи,
                       представимые строками ASCII.";
                  }
                  leaf crypto-algorithm {
                    type identityref {
                      base key-chain:crypto-algorithm;
                    }
                    description
                      "Криптографический алгоритм,
                       связанный с ключом.";
                  }
                }
              }
            }
          }
          uses vpn-common:service-status;
        }
        container vrrp {
          when "derived-from-or-self(../type, "
             + "'vpn-common:vrrp-routing')" {
            description
              "Применимо лишь для протокола VRRP.";
          }
          description
            "Конфигурация, связанная с VRRP.";
          reference
            "RFC 5798: Virtual Router Redundancy
                       Protocol (VRRP) Version 3 for
                       IPv4 and IPv6";
          leaf address-family {
            type identityref {
              base vpn-common:address-family;
            }
            description
              "Активация IPv4, IPv6 или обоих.";
          }
          leaf vrrp-group {
            type uint8 {
              range "1..255";
            }
            description
              "Идентификатор группы VRRP.";
          }
          leaf backup-peer {
            type inet:ip-address;
            description
              "IP-адрес партнёра.";
          }
          leaf-list virtual-ip-address {
            type inet:ip-address;
            description
              "Виртуальный IP-адрес для одной группы
               VRRP.";
            reference
              "RFC 5798: Virtual Router Redundancy
                         Protocol (VRRP) Version 3 for
                         IPv4 and IPv6,
                         Sections 1.2 and 1.3";
          }
          leaf priority {
            type uint8 {
              range "1..254";
            }
            default "100";
            description
              "Локальный приоритет узла VRRP.";
          }
          leaf ping-reply {
            type boolean;
            default "false";
            description
              "Управляет откликами узла VRRP на ping.";
          }
          uses vpn-common:service-status;
        }
      }
    }
    container oam {
      description
        "Задаёт применяемые механизмы OAM. BFD
         задан как механизм обнаружения отказов,
         но в будущем механизмы могут добавляться.";
      container bfd {
```

```
if-feature "vpn-common:bfd";
description
  "Контейнер для BFD.";
leaf session-type {
  type identityref {
    base vpn-common:bfd-session-type;
  }
  default "vpn-common:classic-bfd";
  description
    "Тип сессии BFD.";
}
leaf desired-min-tx-interval {
  type uint32;
  units "microseconds";
  default "1000000";
  description
    "Минимальный интервал передачи пакетов
     BFD Control, желаемый для оператора.";
  reference
    "RFC 5880: Bidirectional Forwarding
                Detection (BFD),
                Section 6.8.7";
}
leaf required-min-rx-interval {
  type uint32;
  units "microseconds";
  default "1000000";
  description
    "Минимальный интервал приёма пакетов BFD
     Control, который следует поддерживать PE.";
  reference
    "RFC 5880: Bidirectional Forwarding
                Detection (BFD),
                Section 6.8.7";
}
leaf local-multiplier {
  type uint8 {
    range "1..255";
  }
  default "3";
  description
    "Коэффициент обнаружения, передаваемый
     партнёру BFD. Интервал обнаружения для
     принимающего партнёра BFD рассчитывается
     путём умножения значения согласованного
     интервала передачи на этот коэффициент.";
  reference
    "RFC 5880: Bidirectional Forwarding
                Detection (BFD),
                Section 6.8.7";
}
leaf holdtime {
  type uint32;
  units "milliseconds";
  description
    "Ожидаемое время удержания BFD. Клиент может
     вносить некие фиксированные значения для
     периода удержания, если провайдер разрешает
     ему применять эту функцию. Без разрешения
     провайдера значения установить нельзя.";
  reference
    "RFC 5880: Bidirectional Forwarding
                Detection (BFD),
                Section 6.8.18";
}
leaf profile {
  type leafref {
    path "/l3vpn-ntw/vpn-profiles"
       + "/valid-provider-identifiers"
       + "/bfd-profile-identifier/id";
  }
  description
    "Общеизвестное имя профиля сервис-провайдера.
     Провайдер может предлагать клиентам
     некоторые профили в зависимости от желаемого
     клиентом уровня обслуживания.";
}
container authentication {
  presence "Разрешена аутентификация BFD";
  description
    "Параметры аутентификации BFD.";
  leaf key-chain {
    type key-chain:key-chain-ref;
    description
      "Имя цепочки ключей.";
  }
```

```
        leaf meticulous {
          type boolean;
          description
            "Включает «дотошный» (meticulous) режим.";
          reference
            "RFC 5880: Bidirectional Forwarding
                       Detection (BFD),
                       Section 6.7";
        }
      }
      uses vpn-common:service-status;
    }
  }
  container security {
    description
      "Зависящие от сайта параметры безопасности.";
    container encryption {
      if-feature "vpn-common:encryption";
      description
        "Контейнер защитного шифрования CE-PE.";
      leaf enabled {
        type boolean;
        default "false";
        description
          "Значение true задаёт использование
           шифрования, иначе оно отключено.";
      }
      leaf layer {
        when "../enabled = 'true'" {
          description
            "Включается лишь при использовании
             шифрования.";
        }
        type enumeration {
          enum layer2 {
            description
              "Шифрование происходит на уровне L2.";
          }
          enum layer3 {
            description
              "Шифрование происходит на уровне L3.
               Например, может применяться IPsec, если
               клиент запросил шифрование L3.";
          }
        }
        description
          "Уровень, на котором происходит шифрование.";
      }
    }
    container encryption-profile {
      when "../encryption/enabled = 'true'" {
        description
          "Уровень, где включено шифрование.";
      }
      description
        "Контейнер для профиля шифрования.";
      choice profile {
        description
          "Выбор профиля шифрования.";
        case provider-profile {
          leaf profile-name {
            type leafref {
              path "/l3vpn-ntw/vpn-profiles"
                 + "/valid-provider-identifiers"
                 + "/encryption-profile-identifier/id";
            }
            description
              "Имя применяемого профиля от провайдера";
          }
        }
        case customer-profile {
          leaf customer-key-chain {
            type key-chain:key-chain-ref;
            description
              "Указанная клиентом цепочка ключей.";
          }
        }
      }
    }
  }
  container service {
    description
      "Параметры службы для присоединения.";
    leaf pe-to-ce-bandwidth {
      if-feature "vpn-common:inbound-bw";
      type uint64;
```

```
        units "bps";
        description
          "Пропускная способность соединения на вход (от
           SP к сайту) с точки зрения сайта клиента. В
           L3SM для этого служит input-bandwidth.";
      }
      leaf ce-to-pe-bandwidth {
        if-feature "vpn-common:outbound-bw";
        type uint64;
        units "bps";
        description
          "Пропускная способность соединения на выход (от
           сайта клиента к SP) с точки зрения сайта
           клиента. В L3SM для этого служит
           output-bandwidth.";
      }
      leaf mtu {
        type uint32;
        units "bytes";
        description
          "MTU на уровне службы. Для сервиса IP это будет
           IP MTU. Если включён режим «оператор для
           операторов» (CsC), запрашиваемое значение MTU
           будет указывать максимальный размер пакета с
           меткой MPLS, а не IP MTU.";
      }
      container qos {
        if-feature "vpn-common:qos";
        description
          "Конфигурация QoS.";
        container qos-classification-policy {
          description
            "Конфигурация правил классификации трафика.";
          uses vpn-common:qos-classification-policy;
        }
        container qos-action {
          description
            "Список действий для правил QoS.";
          list rule {
            key "id";
            description
              "Список действий QoS.";
            leaf id {
              type string;
              description
                "Идентификатор правила для действия QoS";
            }
            leaf target-class-id {
              type string;
              description
                "Указание класса обслуживания (внутреннее
                 значение для администратора).";
            }
            leaf inbound-rate-limit {
              type decimal64 {
                fraction-digits 5;
                range "0..100";
              }
              units "percent";
              description
                "Задаёт условия и способ ограничения
                 скорости входящего трафика для правила
                 QoS. Указывается в процентах от значения
                 input-bandwidth.";
            }
            leaf outbound-rate-limit {
              type decimal64 {
                fraction-digits 5;
                range "0..100";
              }
              units "percent";
              description
                "Задаёт условия и способ ограничения
                 скорости исходящего трафика для правила
                 QoS. Указывается в процентах от значения
                 output-bandwidth.";
            }
          }
        }
        container qos-profile {
          description
            "Конфигурация профиля QoS.";
          list qos-profile {
            key "profile";
            description
              "Профиль QoS - стандартный или
```

```
            настраиваемый.";
        leaf profile {
          type leafref {
            path "/l3vpn-ntw/vpn-profiles"
               + "/valid-provider-identifiers"
               + "/qos-profile-identifier/id";
          }
          description
            "Профиль QoS для применения.";
        }
        leaf direction {
          type identityref {
            base vpn-common:qos-profile-direction;
          }
          default "vpn-common:both";
          description
            "Направление для применения профиля QoS";
        }
      }
    }
  }
  container carriers-carrier {
    if-feature "vpn-common:carriers-carrier";
    description
      "Применяется, когда клиент предоставляет
       услуги на основе MPLS, т. е. в случае CsC
       (клиент создаёт службы MPLS, используя IP VPN
       для передачи трафика службы).";
    leaf signaling-type {
      type enumeration {
        enum ldp {
          description
            "Использование LDP как сигнального
             протокола между PE и CE. Это требует
             также настройки протокола IGP.";
        }
        enum bgp {
          description
            "Использование BGP как сигнального
             протокола между PE и CE. Это требует
             также настройки BGP как протокола
             маршрутизации.";
          reference
            "RFC 8277: Using BGP to Bind MPLS
                       Labels to Address
                       Prefixes";
        }
      }
      default "bgp";
      description
        "Тип сигнализации MPLS.";
    }
  }
  container ntp {
    description
      "В некоторых VPN (например, инфраструктура и
       управление) может потребоваться синхронизация
       часов. Этот контейнер содержит параметры для
       включения услуг NTP.";
    reference
      "RFC 5905: Network Time Protocol Version 4:
                 Protocol and Algorithms
                 Specification";
    leaf broadcast {
      type enumeration {
        enum client {
          description
            "Узел VPN будет прослушивать
             широковещательные сообщения NTP в этом
             доступе в сеть VPN.";
        }
        enum server {
          description
            "Узел VPN будет вести себя как
             широковещательный сервер.";
        }
      }
      description
        "Указывает широковещательный режим NTP для
         использования в этом доступе в сеть VPN.";
    }
    container auth-profile {
      description
        "Указатель на локальный профиль.";
      leaf profile-id {
        type string;
```

```
        description
          "Предоставляется указатель на локальный
           профиль аутентификации на узле VPN.";
      }
    }
    uses vpn-common:service-status;
  }
  container multicast {
    if-feature "vpn-common:multicast";
    description
      "Multicast-параметры для доступа в сеть.";
    leaf access-type {
      type enumeration {
        enum receiver-only {
          description
            "На партнёрском имеются лишь получатели";
        }
        enum source-only {
          description
            "На партнёрском имеются лишь источники";
        }
        enum source-receiver {
          description
            "На партнёрском сайте имеются источники и
             получатели.";
        }
      }
      default "source-receiver";
      description
        "Тип multicast-сайта.";
    }
    leaf address-family {
      type identityref {
        base vpn-common:address-family;
      }
      description
        "Указывает семейство адресов.";
    }
    leaf protocol-type {
      type enumeration {
        enum host {
          description
            "Хосты напрямую подключены к сети SP.
             На хостах нужны протоколы, такие как
             IGMP или MLD.";
        }
        enum router {
          description
            "Хосты находятся за маршрутизатором
             клиента, будет применяться PIM.";
        }
        enum both {
          description
            "Часть хостов расположена за клиентским
             маршрутизатором, другие подключены
             напрямую к сети SP. Нужно использовать
             протоколы для хостов и маршрутизации.
             Обычно применяются IGMP и PIM.";
        }
      }
      default "both";
      description
        "Тип группового протокола для использования с
         сайтом клиента.";
    }
    leaf remote-source {
      type boolean;
      default "false";
      description
        "Удалённый multicast-источник не находится в
         одной подсети с доступом в сеть VPN.
         Значение true указывает, что групповой
         трафик от удалённого источника
         воспринимается.";
    }
    container igmp {
      when "../protocol-type = 'host' and "
         + "../address-family = 'vpn-common:ipv4' "
         + "or 'vpn-common:dual-stack'";
      if-feature "vpn-common:igmp";
      description
        "Связанные с IGMP параметры.";
      list static-group {
        key "group-addr";
        description
          "Статические источники/группы, связанные
```

```
          с сессией IGMP.";
        leaf group-addr {
          type rt-types:ipv4-multicast-group-address;
          description
            "Адрес IPv4 для multicast-группы.";
        }
        leaf source-addr {
          type
            rt-types:ipv4-multicast-source-address;
          description
            "Адрес IPv4 для multicast-источника.";
        }
      }
      leaf max-groups {
        type uint32;
        description
          "Максимальное число групп.";
      }
      leaf max-entries {
        type uint32;
        description
          "Максимальное число записей IGMP.";
      }
      leaf max-group-sources {
        type uint32;
        description
          "Максимальное число групповых источников.";
      }
      leaf version {
        type identityref {
          base vpn-common:igmp-version;
        }
        default "vpn-common:igmpv2";
        description
          "Версия IGMP.";
      }
      uses vpn-common:service-status;
    }
    container mld {
      when "../protocol-type = 'host' and "
         + "../address-family = 'vpn-common:ipv6' "
         + "or 'vpn-common:dual-stack'";
      if-feature "vpn-common:mld";
      description
        "Параметры, связанные с MLD.";
      list static-group {
        key "group-addr";
        description
          "Статические источники/группы, связанные
           с сессией MLD.";
        leaf group-addr {
          type rt-types:ipv6-multicast-group-address;
          description
            "Адрес IPv6 для multicast-группы.";
        }
        leaf source-addr {
          type
            rt-types:ipv6-multicast-source-address;
          description
            "Адрес IPv6 для multicast-источника.";
        }
      }
      leaf max-groups {
        type uint32;
        description
          "Максимальное число групп.";
      }
      leaf max-entries {
        type uint32;
        description
          "Максимальное число записей MLD.";
      }
      leaf max-group-sources {
        type uint32;
        description
          "Максимальное число групповых источников.";
      }
      leaf version {
        type identityref {
          base vpn-common:mld-version;
        }
        default "vpn-common:mldv2";
        description
          "Версия протокола MLD.";
      }
      uses vpn-common:service-status;
```

```
                    }
                    container pim {
                      when "../protocol-type = 'router'";
                      if-feature "vpn-common:pim";
                      description
                        "Применимо лишь для протокола pim.";
                      leaf hello-interval {
                        type rt-types:timer-value-seconds16;
                        default "30";
                        description
                          "Интервал между сообщениями PIM Hello.
                           Значение infinity и not-set отменяют
                           передачу периодических сообщений Hello.";
                        reference
                          "RFC 7761: Protocol Independent
                                     Multicast - Sparse Mode
                                     (PIM-SM): Protocol
                                     Specification (Revised),
                                     Section 4.11
                           RFC 8294: Common YANG Data Types for
                                     the Routing Area";
                      }
                      leaf dr-priority {
                        type uint32;
                        default "1";
                        description
                          "Предпочтение, связанное с выбором DR.
                           Большее значение даёт преимущество.";
                        reference
                          "RFC 7761: Protocol Independent
                                     Multicast - Sparse Mode
                                     (PIM-SM): Protocol
                                     Specification (Revised),
                                     Section 4.3.2";
                      }
                      uses vpn-common:service-status;
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
  }
  <CODE ENDS>
```

9. Вопросы безопасности

Заданные в этом документе модули YANG определяют схемы для данных, которые предназначены для доступа через протоколы управления сетью, такие как NETCONF [RFC6241] или RESTCONF [RFC8040]. Нижним уровнем для NETCONF является уровень защищённого транспорта с обязательно для реализации поддержкой Secure Shell (SSH) [RFC6242]. Нижним уровнем RESTCONF является HTTPS с обязательной реализацией защищённого транспорта TLS [RFC8446].

Модель управления доступом к конфигурации сети (Network Configuration Access Control Model или NACM) [RFC8341] обеспечивает средства для предоставления доступа лишь определенным пользователям NETCONF или RESTCONF к заранее заданному набору содержимого и протокольных операций NETCONF или RESTCONF.

В этом модуле данных YANG определено множество узлов данных, которые разрешают запись, создание и удаление (т. е. config true, как принято по умолчанию). Эти узлы могут быть конфиденциальными или уязвимыми в некоторых сетевых средах. Запись в такие узлы (например, edit-config) без должной защиты может негативно влиять на работу сети. Ниже перечислены ветви и узлы, которые могут быть конфиденциальны или уязвимы.

vpn-profiles

Этот контейнер содержит набор деликатных сведений, которые влияют на предоставление услуг L3VPN. Например, атакующий с доступом к этим узлам данных сможет манипулировать правилами маршрутизации, QoS и свойствами шифрования. Эти узлы помечены тегом nacm:default-deny-write в [RFC9181].

vpn-services

Злоумышленник с доступом к узалм сети может предпринять различные атаки, такие как удаление работающих служб L3VPN, прерывающее весь трафик клиентов. Кроме того, атакующий сможет изменять атрибуты работающих служб (например, QoS, пропускную способность, протоколы маршрутизации, ключевой материал), что приведёт к некорректной работе службы и нарушению соглашения об уровне обслуживания (Service Level Agreement или SLA). Злоумышленник может также попытаться создать службу L3VPN или добавить доступ в сеть. Помимо применения NACM для предотвращения несанкционированного доступа указанные действия можно обнаружить путём адекватного мониторинга и отслеживания изменений в конфигурации сети.

Некоторые из доступных для чтения узлов в этом модуле YANG могут быть конфиденциальны или уязвимы в той или иной сетевой среде. Важно контролировать доступ к таким объектам (например, get, get-config, notification). Ниже перечислены ветви и узлы, которые могут быть конфиденциальны или уязвимы.

customer-name и ip-connection

Злоумышленник может получить связанные с приватностью сведения, которые можно использовать для отслеживания клиента. Раскрытие таких сведений может считаться нарушением доверительных отношений между клиентом и поставщиком услуг.

keying-material

Злоумышленник может узнать криптографические ключи, применяемые для защиты базовых служб VPN (например, маршрутизации CE-PE). Эти ключи можно использовать для внедрения поддельных анонсов маршрутов.

Некоторые узлы данных (bgp, ospf, isis, rip, bfd) применяют [RFC8177] для аутентификации, поэтому модуль наследует соображения безопасности, рассмотренные в разделе 5 [RFC8177]. Кроме того, эти узлы данных поддерживают явное представление ключей строками ASCII. Использование шестнадцатеричного формата для ключей позволило бы повысить энтропию при том же числе октетов в строке ключа. Однако такой формат не включён в эту версию L3NM, поскольку не поддерживается базовыми модулями устройств (например, [RFC8695]).

Как отмечено в параграф 7.6.3, модуль поддерживает MD5 для соответствия установленной базе BGP. MD5 имеет множество слабостей, рассмотренных в разделе 2 [RFC6151] и параграфе 2.1 [RFC6952].

В [RFC8633] описан опыт, который следует учитывать в VPN, использующих NTP. Кроме того, механизм криптографической защиты NTP задан в [RFC8915].

10. Взаимодействие с IANA

Агентство IANA зарегистрировало URI в субреесте ns реестра IETF XML Registry [RFC3688]

```
URI:  urn:ietf:params:xml:ns:yang:ietf-l3vpn-ntw
Registrant Contact:  The IESG.
XML:  N/A; запрошенный URI является пространством имён XML.
```

Агентство IANA зарегистрировало модуль YANG в субреестре YANG Module Names [RFC6020] реестра YANG Parameters.

```
Name:  ietf-l3vpn-ntw
Maintained by IANA?  N
Namespace:  urn:ietf:params:xml:ns:yang:ietf-l3vpn-ntw
Prefix:  l3nm
Reference:  RFC 9182
```

11. Литература

11.1. Нормативные документы

[ISO10589] ISO, "Information technology - Telecommunications and information exchange between systems - Intermediate System to Intermediate System intra-domain routeing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode network service (ISO 8473)", ISO/IEC 10589:2002, 2002, <https://www.iso.org/standard/30932.html>.

[RFC1112] Deering, S., "Host extensions for IP multicasting", STD 5, RFC 1112, DOI 10.17487/RFC1112, August 1989, <https://www.rfc-editor.org/info/rfc1112>.

[RFC1195] Callon, R., "Use of OSI IS-IS for routing in TCP/IP and dual environments", RFC 1195, DOI 10.17487/RFC1195, December 1990, <https://www.rfc-editor.org/info/rfc1195>.

[RFC2080] Malkin, G. and R. Minnear, "RIPng for IPv6", RFC 2080, DOI 10.17487/RFC2080, January 1997, <https://www.rfc-editor.org/info/rfc2080>.

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

[RFC2236] Fenner, W., "Internet Group Management Protocol, Version 2", RFC 2236, DOI 10.17487/RFC2236, November 1997, <https://www.rfc-editor.org/info/rfc2236>.

[RFC2453] Malkin, G., "RIP Version 2", STD 56, RFC 2453, DOI 10.17487/RFC2453, November 1998, <https://www.rfc-editor.org/info/rfc2453>.

[RFC2710] Deering, S., Fenner, W., and B. Haberman, "Multicast Listener Discovery (MLD) for IPv6", RFC 2710, DOI 10.17487/RFC2710, October 1999, <https://www.rfc-editor.org/info/rfc2710>.

[RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, DOI 10.17487/RFC3376, October 2002, <https://www.rfc-editor.org/info/rfc3376>.

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

[RFC3810] Vida, R., Ed. and L. Costa, Ed., "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, DOI 10.17487/RFC3810, June 2004, <https://www.rfc-editor.org/info/rfc3810>.

[RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <https://www.rfc-editor.org/info/rfc4271>.

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

[RFC4552] Gupta, M. and N. Melam, "Authentication/Confidentiality for OSPFv3", RFC 4552, DOI 10.17487/RFC4552, June 2006, <https://www.rfc-editor.org/info/rfc4552>.

[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, <https://www.rfc-editor.org/info/rfc4577>.

[RFC5308] Hopps, C., "Routing IPv6 with IS-IS", RFC 5308, DOI 10.17487/RFC5308, October 2008, <https://www.rfc-editor.org/info/rfc5308>.

[RFC5701] Rekhter, Y., "IPv6 Address Specific BGP Extended Community Attribute", RFC 5701, DOI 10.17487/RFC5701, November 2009, <https://www.rfc-editor.org/info/rfc5701>.

[RFC5709] Bhatia, M., Manral, V., Fanto, M., White, R., Barnes, M., Li, T., and R. Atkinson, "OSPFv2 HMAC-SHA Cryptographic Authentication", RFC 5709, DOI 10.17487/RFC5709, October 2009, <https://www.rfc-editor.org/info/rfc5709>.

[RFC5798] Nadas, S., Ed., "Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6", RFC 5798, DOI 10.17487/RFC5798, March 2010, <https://www.rfc-editor.org/info/rfc5798>.

[RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <https://www.rfc-editor.org/info/rfc5880>.

[RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, DOI 10.17487/RFC5905, June 2010, <https://www.rfc-editor.org/info/rfc5905>.

[RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, DOI 10.17487/RFC5925, June 2010, <https://www.rfc-editor.org/info/rfc5925>.

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

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

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

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

[RFC6514] Aggarwal, R., Rosen, E., Morin, T., and Y. Rekhter, "BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs", RFC 6514, DOI 10.17487/RFC6514, February 2012, <https://www.rfc-editor.org/info/rfc6514>.

[RFC6565] Pillay-Esnault, P., Moyer, P., Doyle, J., Ertekin, E., and M. Lundberg, "OSPFv3 as a Provider Edge to Customer Edge (PE-CE) Routing Protocol", RFC 6565, DOI 10.17487/RFC6565, June 2012, <https://www.rfc-editor.org/info/rfc6565>.

[RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <https://www.rfc-editor.org/info/rfc6991>.

[RFC7166] Bhatia, M., Manral, V., and A. Lindem, "Supporting Authentication Trailer for OSPFv3", RFC 7166, DOI 10.17487/RFC7166, March 2014, <https://www.rfc-editor.org/info/rfc7166>.

[RFC7474] Bhatia, M., Hartman, S., Zhang, D., and A. Lindem, Ed., "Security Extension for OSPFv2 When Using Manual Key Management", RFC 7474, DOI 10.17487/RFC7474, April 2015, <https://www.rfc-editor.org/info/rfc7474>.

[RFC7761] Fenner, B., Handley, M., Holbrook, H., Kouvelas, I., Parekh, R., Zhang, Z., and L. Zheng, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", STD 83, RFC 7761, DOI 10.17487/RFC7761, March 2016, <https://www.rfc-editor.org/info/rfc7761>.

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

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

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

[RFC8177] Lindem, A., Ed., Qu, Y., Yeung, D., Chen, I., and J. Zhang, "YANG Data Model for Key Chains", RFC 8177, DOI 10.17487/RFC8177, June 2017, <https://www.rfc-editor.org/info/rfc8177>.

[RFC8294] Liu, X., Qu, Y., Lindem, A., Hopps, C., and L. Berger, "Common YANG Data Types for the Routing Area", RFC 8294, DOI 10.17487/RFC8294, December 2017, <https://www.rfc-editor.org/info/rfc8294>.

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

[RFC8343] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 8343, DOI 10.17487/RFC8343, March 2018, <https://www.rfc-editor.org/info/rfc8343>.

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

[RFC8466] Wen, B., Fioccola, G., Ed., Xie, C., and L. Jalil, "A YANG Data Model for Layer 2 Virtual Private Network (L2VPN) Service Delivery", RFC 8466, DOI 10.17487/RFC8466, October 2018, <https://www.rfc-editor.org/info/rfc8466>.

[RFC8519] Jethanandani, M., Agarwal, S., Huang, L., and D. Blair, "YANG Data Model for Network Access Control Lists (ACLs)", RFC 8519, DOI 10.17487/RFC8519, March 2019, <https://www.rfc-editor.org/info/rfc8519>.

[RFC9181] Barguil, S., Gonzalez de Dios, O., Ed., Boucadair, M., Ed., and Q. Wu, "A Common YANG Data Model for Layer 2 and Layer 3 VPNs", RFC 9181, DOI 10.17487/RFC9181, February 2022, <https://www.rfc-editor.org/info/rfc9181>.

11.2. Дополнительная литература

[BGP-YANG] Jethanandani, M., Patel, K., Hares, S., and J. Haas, "BGP YANG Model for Service Provider Networks", Work in Progress, Internet-Draft, draft-ietf-idr-bgp-model-12, 25 October 2021, <https://datatracker.ietf.org/doc/html/draft-ietf-idr-bgp-model-12>.

[Enhanced-VPN-Framework] Dong, J., Bryant, S., Li, Z., Miyasaka, T., and Y. Lee, "A Framework for Enhanced Virtual Private Network (VPN+) Services", Work in Progress, Internet-Draft, draft-ietf-teas-enhanced-vpn-09, 25 October 2021, <https://datatracker.ietf.org/doc/html/draft-ietf-teas-enhanced-vpn-09>.

[IEEE802.1AX] IEEE, "802.1AX-2020 - IEEE Standard for Local and Metropolitan Area Networks--Link Aggregation", IEEE Std 802.1AX-2020, <https://ieeexplore.ieee.org/document/9105034>.

[Network-Slices-Framework] Farrel, A., Ed., Gray, E., Drake, J., Rokui, R., Homma, S., Makhijani, K., Contreras, LM., and J. Tantsura, "Framework for IETF Network Slices", Work in Progress, Internet-Draft, draft-ietf-teas-ietf-network-slices-05, 25 October 2021, <https://datatracker.ietf.org/doc/html/draft-ietf-teas-ietf-network-slices-05>.

[PIM-YANG] Liu, X., McAllister, P., Peter, A., Sivakumar, M., Liu, Y., and F. Hu, "A YANG Data Model for Protocol Independent Multicast (PIM)", Work in Progress[1], Internet-Draft, draft-ietf-pim-yang-17, 19 May 2018, <https://datatracker.ietf.org/doc/html/draft-ietf-pim-yang-17>.

[PYANG] "pyang", commit 524cf61, December 2021, <https://github.com/mbj4668/pyang>.

[QoS-YANG] Choudhary, A., Jethanandani, M., Aries, E., and I. Chen, "A YANG Data Model for Quality of Service (QoS)", Work in Progress, Internet-Draft, draft-ietf-rtgwg-qos-model-06, 8 November 2021, <https://datatracker.ietf.org/doc/html/draft-ietf-rtgwg-qos-model-06>.

[RFC3618] Fenner, B., Ed. and D. Meyer, Ed., "Multicast Source Discovery Protocol (MSDP)", RFC 3618, DOI 10.17487/RFC3618, October 2003, <https://www.rfc-editor.org/info/rfc3618>.

[RFC3644] Snir, Y., Ramberg, Y., Strassner, J., Cohen, R., and B. Moore, "Policy Quality of Service (QoS) Information Model", RFC 3644, DOI 10.17487/RFC3644, November 2003, <https://www.rfc-editor.org/info/rfc3644>.

[RFC4026] Andersson, L. and T. Madsen, "Provider Provisioned Virtual Private Network (VPN) Terminology", RFC 4026, DOI 10.17487/RFC4026, March 2005, <https://www.rfc-editor.org/info/rfc4026>.

[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, <https://www.rfc-editor.org/info/rfc4110>.

[RFC4176] El Mghazli, Y., Ed., Nadeau, T., Boucadair, M., Chan, K., and A. Gonguet, "Framework for Layer 3 Virtual Private Networks (L3VPN) Operations and Management", RFC 4176, DOI 10.17487/RFC4176, October 2005, <https://www.rfc-editor.org/info/rfc4176>.

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

[RFC6037] Rosen, E., Ed., Cai, Y., Ed., and IJ. Wijnands, "Cisco Systems' Solution for Multicast in BGP/MPLS IP VPNs", RFC 6037, DOI 10.17487/RFC6037, October 2010, <https://www.rfc-editor.org/info/rfc6037>.

[RFC6151] Turner, S. and L. Chen, "Updated Security Considerations for the MD5 Message-Digest and the HMAC-MD5 Algorithms", RFC 6151, DOI 10.17487/RFC6151, March 2011, <https://www.rfc-editor.org/info/rfc6151>.

[RFC6952] Jethanandani, M., Patel, K., and L. Zheng, "Analysis of BGP, LDP, PCEP, and MSDP Issues According to the Keying and Authentication for Routing Protocols (KARP) Design Guide", RFC 6952, DOI 10.17487/RFC6952, May 2013, <https://www.rfc-editor.org/info/rfc6952>.

[RFC7149] Boucadair, M. and C. Jacquenet, "Software-Defined Networking: A Perspective from within a Service Provider Environment", RFC 7149, DOI 10.17487/RFC7149, March 2014, <https://www.rfc-editor.org/info/rfc7149>.

[RFC7297] Boucadair, M., Jacquenet, C., and N. Wang, "IP Connectivity Provisioning Profile (CPP)", RFC 7297, DOI 10.17487/RFC7297, July 2014, <https://www.rfc-editor.org/info/rfc7297>.

[RFC7426] Haleplidis, E., Ed., Pentikousis, K., Ed., Denazis, S., Hadi Salim, J., Meyer, D., and O. Koufopavlou, "Software-Defined Networking (SDN): Layers and Architecture Terminology", RFC 7426, DOI 10.17487/RFC7426, January 2015, <https://www.rfc-editor.org/info/rfc7426>.

[RFC7880] Pignataro, C., Ward, D., Akiya, N., Bhatia, M., and S. Pallagatti, "Seamless Bidirectional Forwarding Detection (S-BFD)", RFC 7880, DOI 10.17487/RFC7880, July 2016, <https://www.rfc-editor.org/info/rfc7880>.

[RFC8077] Martini, L., Ed. and G. Heron, Ed., "Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)", STD 84, RFC 8077, DOI 10.17487/RFC8077, February 2017, <https://www.rfc-editor.org/info/rfc8077>.

[RFC8277] Rosen, E., "Using BGP to Bind MPLS Labels to Address Prefixes", RFC 8277, DOI 10.17487/RFC8277, October 2017, <https://www.rfc-editor.org/info/rfc8277>.

[1] Опубликовано в RFC 9128. *Прим. перев.*

[RFC8299] Wu, Q., Ed., Litkowski, S., Tomotaki, L., and K. Ogaki, "YANG Data Model for L3VPN Service Delivery", RFC 8299, DOI 10.17487/RFC8299, January 2018, <https://www.rfc-editor.org/info/rfc8299>.

[RFC8309] Wu, Q., Liu, W., and A. Farrel, "Service Models Explained", RFC 8309, DOI 10.17487/RFC8309, January 2018, <https://www.rfc-editor.org/info/rfc8309>.

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

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

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

[RFC8349] Lhotka, L., Lindem, A., and Y. Qu, "A YANG Data Model for Routing Management (NMDA Version)", RFC 8349, DOI 10.17487/RFC8349, March 2018, <https://www.rfc-editor.org/info/rfc8349>.

[RFC8453] Ceccarelli, D., Ed. and Y. Lee, Ed., "Framework for Abstraction and Control of TE Networks (ACTN)", RFC 8453, DOI 10.17487/RFC8453, August 2018, <https://www.rfc-editor.org/info/rfc8453>.

[RFC8512] Boucadair, M., Ed., Sivakumar, S., Jacquenet, C., Vinapamula, S., and Q. Wu, "A YANG Module for Network Address Translation (NAT) and Network Prefix Translation (NPT)", RFC 8512, DOI 10.17487/RFC8512, January 2019, <https://www.rfc-editor.org/info/rfc8512>.

[RFC8633] Reilly, D., Stenn, H., and D. Sibold, "Network Time Protocol Best Current Practices", BCP 223, RFC 8633, DOI 10.17487/RFC8633, July 2019, <https://www.rfc-editor.org/info/rfc8633>.

[RFC8695] Liu, X., Sarda, P., and V. Choudhary, "A YANG Data Model for the Routing Information Protocol (RIP)", RFC 8695, DOI 10.17487/RFC8695, February 2020, <https://www.rfc-editor.org/info/rfc8695>.

[RFC8792] Watsen, K., Auerswald, E., Farrel, A., and Q. Wu, "Handling Long Lines in Content of Internet-Drafts and RFCs", RFC 8792, DOI 10.17487/RFC8792, June 2020, <https://www.rfc-editor.org/info/rfc8792>.

[RFC8915] Franke, D., Sibold, D., Teichel, K., Dansarie, M., and R. Sundblad, "Network Time Security for the Network Time Protocol", RFC 8915, DOI 10.17487/RFC8915, September 2020, <https://www.rfc-editor.org/info/rfc8915>.

[RFC8969] Wu, Q., Ed., Boucadair, M., Ed., Lopez, D., Xie, C., and L. Geng, "A Framework for Automating Service and Network Management with YANG", RFC 8969, DOI 10.17487/RFC8969, January 2021, <https://www.rfc-editor.org/info/rfc8969>.

[RFC9136] Rabadan, J., Ed., Henderickx, W., Drake, J., Lin, W., and A. Sajassi, "IP Prefix Advertisement in Ethernet VPN (EVPN)", RFC 9136, DOI 10.17487/RFC9136, October 2021, <https://www.rfc-editor.org/info/rfc9136>.

[YANG-Composed-VPN] Even, R., Wu, B., Wu, Q., and Y. Cheng, "YANG Data Model for Composed VPN Service Delivery", Work in Progress, Internet-Draft, draft-evenwu-opsawg-yang-composed-vpn-03, 8 March 2019, <https://datatracker.ietf.org/doc/html/draft-evenwu-opsawg-yang-composed-vpn-03>.

[YANG-SAPs] Gonzalez de Dios, O., Barguil, S., Wu, Q., Boucadair, M., and V. Lopez, "A Network YANG Model for Service Attachment Points", Work in Progress, Internet-Draft, draft-ietf-opsawg-sap-00, 25 January 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-opsawg-sap-00>.

Приложение A. Примеры L3VPN

A.1. Предоставление 4G VPN

Сети L3VPN широко применяются для внедрения 3G/4G, стационарных и корпоративных служб главным образом потому, что в сети можно применять разные политики разделения трафика для предоставления мобильным клиентам услуг в соответствии с требованиями SLA.

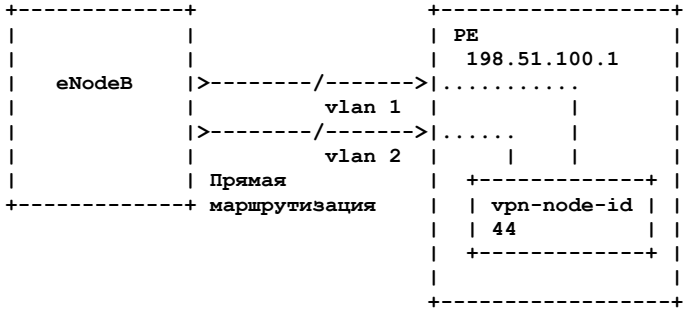

Рисунок 31. Пример мобильного подключения.

Обычно (Рисунок 31) eNodeB (CE) напрямую подключается к маршрутизаторам доступа транспортной сети, а их логические интерфейсы (1 или несколько в зависимости от типа обслуживания) настраиваются для VPN, доставляющей пакеты в платформы ядра. В этом примере vpn-node создаётся с двумя vpn-network-accesses. Этапы организации службы L3VPN с использованием L3NM описаны ниже.

Сначала создаётся служба 4G VPN (Рисунок 32).

```
POST: /restconf/data/ietf-l3vpn-ntw:l3vpn-ntw/vpn-services
Host: example.com
Content-Type: application/yang-data+json

{
 "ietf-l3vpn-ntw:vpn-services": {
   "vpn-service": [
     {
       "vpn-id": "4G",
       "vpn-description": "VPN to deploy 4G services",
       "customer-name": "mycustomer",
       "vpn-service-topology": "custom",
       "vpn-instance-profiles": {
         "vpn-instance-profile": [
           {
             "profile-id": "simple-profile",
             "local-as": 65550,
             "rd": "0:65550:1",
             "address-family": [
               {
                 "address-family": "ietf-vpn-common:dual-stack",
                 "vpn-targets": {
                   "vpn-target": [
                     {
                       "id": 1,
                       "route-targets": [
                         {
                           "route-target": "0:65550:1"
                         }
                       ],
                       "route-target-type": "both"
                     }
                   ]
                 }
               }
             ]
           }
         ]
       }
     }
   ]
 }
}
```

Рисунок 32. Создание службы VPN.

Затем создаётся узел VPN (Рисунок 33). В этом примере узел VPN эквивалентен VRF на физическом устройстве ('ne-id'=198.51.100.1). Отметим, что символы \ в конце строк на рисунках 33и 34 используются в соответствии с [RFC8792].

```
POST: /restconf/data/ietf-l3vpn-ntw:l3vpn-ntw/\
      vpn-services/vpn-service=4G
Host: example.com
Content-Type: application/yang-data+json

{
  "ietf-l3vpn-ntw:vpn-nodes": {
    "vpn-node": [
      {
        "vpn-node-id": "44",
        "ne-id": "198.51.100.1",
        "active-vpn-instance-profiles": {
          "vpn-instance-profile": [
            {
              "profile-id": "simple-profile"
            }
          ]
        }
      }
    ]
  }
}
```

Рисунок 33. Создание узла VPN.

Наконец, создаётся два доступа в сеть VPN с использованием одного физического порта ('interface-id'=1/1/1). В каждом vpn-network-access имеется отдельный интерфейс VLAN (1, 2): SYNC и DATA (Рисунок 34). Эти интерфейсы разделяют трафик.

```
POST: /restconf/data/ietf-l3vpn-ntw:l3vpn-ntw/\
      vpn-services/vpn-service=4G/vpn-nodes/vpn-node=44
content-type: application/yang-data+json

{
  "ietf-l3vpn-ntw:vpn-network-accesses": {
    "vpn-network-access": [
      {
        "id": "1/1/1.1",
        "interface-id": "1/1/1",
        "description": "Интерфейс SYNC для eNODE-B",
        "vpn-network-access-type": "ietf-vpn-common:point-to-point",
        "vpn-instance-profile": "simple-profile",
        "status": {
          "admin-status": {
            "status": "ietf-vpn-common:admin-up"
          }
        },
        "connection": {
          "encapsulation": {
            "type": "ietf-vpn-common:dot1q",
            "dot1q": {
              "cvlan-id": 1
            }
          }
        },
        "ip-connection": {
          "ipv4": {
            "local-address": "192.0.2.1",
            "prefix-length": 30,
            "address-allocation-type": "static-address",
            "static-addresses": {
              "primary-address": "1",
              "address": [
                {
                  "address-id": "1",
                  "customer-address": "192.0.2.2"
                }
              ]
            }
          },
          "ipv6": {
            "local-address": "2001:db8::1",
            "prefix-length": 64,
            "address-allocation-type": "static-address",
            "primary-address": "1",
            "address": [
              {
                "address-id": "1",
                "customer-address": "2001:db8::2"
              }
            ]
          }
        },
        "routing-protocols": {
          "routing-protocol": [
            {
              "id": "1",
              "type": "ietf-vpn-common:direct"
            }
          ]
        }
      },
      {
        "id": "1/1/1.2",
        "interface-id": "1/1/1",
        "description": "Интерфейс DATA для eNODE-B",
        "vpn-network-access-type": "ietf-vpn-common:point-to-point",
        "vpn-instance-profile": "simple-profile",
        "status": {
          "admin-status": {
            "status": "ietf-vpn-common:admin-up"
          }
        },
        "connection": {
          "encapsulation": {
            "type": "ietf-vpn-common:dot1q",
            "dot1q": {
              "cvlan-id": 2
            }
          }
        },
        "ip-connection": {
          "ipv4": {
            "local-address": "192.0.2.1",
            "prefix-length": 30,
```

```
            "address-allocation-type": "static-address",
            "static-addresses": {
              "primary-address": "1",
              "address": [
                {
                  "address-id": "1",
                  "customer-address": "192.0.2.2"
                }
              ]
            }
          },
          "ipv6": {
            "local-address": "2001:db8::1",
            "prefix-length": 64,
            "address-allocation-type": "static-address",
            "primary-address": "1",
            "address": [
              {
                "address-id": "1",
                "customer-address": "2001:db8::2"
              }
            ]
          }
        },
        "routing-protocols": {
          "routing-protocol": [
            {
              "id": "1",
              "type": "ietf-vpn-common:direct"
            }
          ]
        }
      }
    ]
  }
}
```

Рисунок 34. Создание доступа в сеть VPN.

A.2. Петлевой интерфейс

Пример loopback-интерфейса представлен на рисунке 35.

```
{
  "ietf-l3vpn-ntw:vpn-network-accesses": {
    "vpn-network-access": [
      {
        "id": "vpn-access-loopback",
        "interface-id": "Loopback1",
        "description": "Пример loopback-интерфейса.",
        "vpn-network-access-type": "ietf-vpn-common:loopback",
        "status": {
          "admin-status": {
            "status": "ietf-vpn-common:admin-up"
          }
        },
        "ip-connection": {
          "ipv6": {
            "local-address": "2001:db8::4",
            "prefix-length": 128
          }
        }
      }
    ]
  }
}
```

Рисунок 35. Доступ в сеть VPN с Loopback-интерфейсом (тело сообщения).

A.3. Переопределение параметров профиля экземпляра VPN

На рисунке 36 показан упрощённый пример предоставления на уровне службы VPN (как часть vpn-instance-profiles) информации, которая может быть переопределена сведениями, заданными на уровне узла VPN. В этом примере PE3 и PE4 наследуют параметры vpn-instance-profiles, заданные на уровне службы VPN, а PE1 и PE2 снабжены значениями maximum-routes на уровне узла VPN, которые переопределяют значения, заданные на уровне службы VPN.

```
{
  "ietf-l3vpn-ntw:vpn-services": {
    "vpn-service": [
      {
        "vpn-id": "override-example",
        "vpn-service-topology": "ietf-vpn-common:hub-spoke",
        "vpn-instance-profiles": {
          "vpn-instance-profile": [
            {
              "profile-id": "HUB",
              "role": "ietf-vpn-common:hub-role",
```

```
            "local-as": 64510,
            "rd-suffix": 1001,
            "address-family": [
              {
                "address-family": "ietf-vpn-common:dual-stack",
                "maximum-routes": [
                  {
                    "protocol": "ietf-vpn-common:any",
                    "maximum-routes": 100
                  }
                ]
              }
            ]
          },
          {
            "profile-id": "SPOKE",
            "role": "ietf-vpn-common:spoke-role",
            "local-as": 64510,
            "address-family": [
              {
                "address-family": "ietf-vpn-common:dual-stack",
                "maximum-routes": [
                  {
                    "protocol": "ietf-vpn-common:any",
                    "maximum-routes": 1000
                  }
                ]
              }
            ]
          }
        ]
      },
      "vpn-nodes": {
        "vpn-node": [
          {
            "vpn-node-id": "PE1",
            "ne-id": "pe1",
            "router-id": "198.51.100.1",
            "active-vpn-instance-profiles": {
              "vpn-instance-profile": [
                {
                  "profile-id": "HUB",
                  "rd": "1:198.51.100.1:1001",
                  "address-family": [
                    {
                      "address-family":
                               "ietf-vpn-common:dual-stack",
                      "maximum-routes": [
                        {
                          "protocol": "ietf-vpn-common:any",
                          "maximum-routes": 10
                        }
                      ]
                    }
                  ]
                }
              ]
            }
          },
          {
            "vpn-node-id": "PE2",
            "ne-id": "pe2",
            "router-id": "198.51.100.2",
            "active-vpn-instance-profiles": {
              "vpn-instance-profile": [
                {
                  "profile-id": "SPOKE",
                  "address-family": [
                    {
                      "address-family":
                               "ietf-vpn-common:dual-stack",
                      "maximum-routes": [
                        {
                          "protocol": "ietf-vpn-common:any",
                          "maximum-routes": 100
                        }
                      ]
                    }
                  ]
                }
              ]
            }
          },
          {
            "vpn-node-id": "PE3",
            "ne-id": "pe3",
```

```
                "router-id": "198.51.100.3",
                "active-vpn-instance-profiles": {
                  "vpn-instance-profile": [
                    {
                      "profile-id": "SPOKE"
                    }
                  ]
                }
              },
              {
                "vpn-node-id": "PE4",
                "ne-id": "pe4",
                "router-id": "198.51.100.4",
                "active-vpn-instance-profiles": {
                  "vpn-instance-profile": [
                    {
                      "profile-id": "SPOKE"
                    }
                  ]
                }
              }
            ]
          }
        }
      ]
    }
  }
```

Рисунок 36. Пример профиля экземпляра VPN (тело сообщения).

A.4. Пример предоставления Multicast VPN

IPTV является основным применением групповой передачи в ЛВС. В приведённом ниже примере включён разреженный режим PIM (PIM - Sparse Mode или PIM-SM) между PE и CE. PE получает групповой трафик от CE, напрямую соединённого с multicast-источником. Сигнализация между PE и CE обеспечивается с помощью BGP. Для multicast-группы статически задана точка встречи RP.

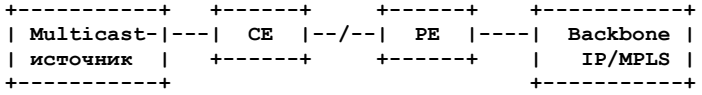

Рисунок 37. Пример групповой услуги L3VPN.

На рисунке 38 показано, как настроить групповую услугу L3VPN с использованием L3NM.

Сначала создаётся multicast-служба вместе с базовым профилем экземпляра VPN.

```
{
  "ietf-l3vpn-ntw:vpn-services": {
    "vpn-service": [
      {
        "vpn-id": "Multicast-IPTV",
        "vpn-description": "Multicast IPTV VPN service",
        "customer-name": "a-name",
        "vpn-service-topology": "ietf-vpn-common:hub-spoke",
        "vpn-instance-profiles": {
          "vpn-instance-profile": [
            {
              "profile-id": "multicast",
              "role": "ietf-vpn-common:hub-role",
              "local-as": 65536,
              "multicast": {
                "rp": {
                  "rp-group-mappings": {
                    "rp-group-mapping": [
                      {
                        "id": 1,
                        "rp-address": "203.0.113.17",
                        "groups": {
                          "group": [
                            {
                              "id": 1,
                              "group-address": "239.130.0.0/15"
                            }
                          ]
                        }
                      }
                    ]
                  },
                  "rp-discovery": {
                    "rp-discovery-type": "ietf-vpn-common:static-rp"
                  }
                }
              }
            }
          ]
        }
      }
```

```
      ]
    }
  }
```

Рисунок 38. Организация групповой службы VPN (без тела запроса).

Затем создаются узлы VPN (Рисунок 39). В этом примере узел VPN представляет VRF, настроенную на физическом устройстве.

```
{
  "ietf-l3vpn-ntw:vpn-node": [
    {
      "vpn-node-id": "500003105",
      "description": "VRF-IPTV-MULTICAST",
      "ne-id": "198.51.100.10",
      "router-id": "198.51.100.10",
      "active-vpn-instance-profiles": {
        "vpn-instance-profile": [
          {
            "profile-id": "multicast",
            "rd": "65536:31050202"
          }
        ]
      }
    }
  ]
}
```

Рисунок 39. Организация группового узла VPN (без тела запроса).

В заключение создаётся доступ в сеть VPN со включённой групповой передачей (Рисунок 40).

```
{
  "ietf-l3vpn-ntw:vpn-network-access": {
    "id": "1/1/1",
    "description": "Connected-to-source",
    "vpn-network-access-type": "ietf-vpn-common:point-to-point",
    "vpn-instance-profile": "multicast",
    "status": {
      "admin-status": {
        "status": "ietf-vpn-common:admin-up"
      },
      "ip-connection": {
        "ipv4": {
          "local-address": "203.0.113.1",
          "prefix-length": 30,
          "address-allocation-type": "static-address",
          "static-addresses": {
            "primary-address": "1",
            "address": [
              {
                "address-id": "1",
                "customer-address": "203.0.113.2"
              }
            ]
          }
        }
      },
      "routing-protocols": {
        "routing-protocol": [
          {
            "id": "1",
            "type": "ietf-vpn-common:bgp-routing",
            "bgp": {
              "description": "Connected to CE",
              "peer-as": "65537",
              "address-family": "ietf-vpn-common:ipv4",
              "neighbor": "203.0.113.2"
            }
          }
        ]
      },
      "service": {
        "pe-to-ce-bandwidth": "100000000",
        "ce-to-pe-bandwidth": "100000000",
        "mtu": 1500,
        "multicast": {
          "access-type": "source-only",
          "address-family": "ietf-vpn-common:ipv4",
          "protocol-type": "router",
          "pim": {
            "hello-interval": 30,
            "status": {
              "admin-status": {
                "status": "ietf-vpn-common:admin-up"
              }
            }
```

```
                }
              }
            }
          }
        }
      }
```

Рисунок 40. Организация группового доступа в сеть VPN (без тела запроса).

Благодарности

В ходе обсуждения этой работы были получены ценные замечания и рецензии от (в алфавитном порядке) Raul Arco, Miguel Cros Cecilia, Joe Clarke, Dhruv Dhody, Adrian Farrel, Roque Gagliano, Christian Jacquenet, Kireeti Kompella, Julian Lucek, Greg Mirsky, Tom Petch. Спасибо им за это. Спасибо Philip Eardley за обзор черновой версии документа.

Daniel King, Daniel Voyer, Luay Jalil, Stephane Litkowski внесли вклад в ранние версии документа. Большое спасибо Robert Wilton за рецензию AD. Спасибо Andrew Malis за обзор для директората маршрутизации, Rifaat Shekh-Yusef за обзор для директората безопасности, Qin Wu за рецензию opsdir и Pete Resnick за рецензию для директората genart. Спасибо Michael Scharf за обсуждение TCP-AO. Спасибо Martin Duke, Lars Eggert, Zaheduzzaman Sarker, Roman Danyliw, Erik Kline, Benjamin Kaduk, Francesca Palombini, Éric Vyncke за рецензии IESG.

Эта работа частично поддерживалась в проекте Европейской комиссии H2020-ICT-2016-2 METRO-HAUL (G.A. 761727) и проекте Horizon 2020 Secured по автономному управления трафиком Tera в потоках SDN (Teraflow) (G.A. 101015857).

Участники работы

Victor Lopez
Nokia
Madrid
Spain
Email: victor.lopez@nokia.com

Qin Wu
Huawei
Email: bill.wu@huawei.com

Manuel Lopez
Vodafone
Spain
Email: manuel-julian.lopez@vodafone.com

Lucia Oliva Ballega
Telefonica
Email: lucia.olivaballega.ext@telefonica.com

Erez Segev
Ribbon Communications
Email: erez.segev@rbbn.com

Paul Sherratt
Gamma Telecom
Email: paul.sherratt@gamma.co.uk

Адреса авторов

Samier Barguil
Telefonica
Madrid
Spain
Email: samier.barguilgiraldo.ext@telefonica.com

Oscar Gonzalez de Dios (editor)
Telefonica
Madrid
Spain
Email: oscar.gonzalezdedios@telefonica.com

Mohamed Boucadair (editor)
Orange
35000 Rennes
France
Email: mohamed.boucadair@orange.com

Luis Angel Munoz
Vodafone
Spain
Email: luis-angel.munoz@vodafone.com

Alejandro Aguado
Nokia
Madrid
Spain
Email: alejandro.aguado_martin@nokia.com

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru