Программируемые Сети ООО
117342, г. Москва, ул. Бутлерова, д.17

+7 (499) 649 46 46
www.Brain4Net.ru
need@brain4net.ru

РУКОВОДСТВО ПО ПЛАНИРОВАНИЮ И ВНЕДРЕНИЮ

B4N SERVICE PLATFORM

Версия 2.5

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ **2**

ВВЕДЕНИЕ **5**

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ, ТЕРМИНОВ И СОКРАЩЕНИЙ **6**

НАЗНАЧЕНИЕ **11**

АРХИТЕКТУРА И СОСТАВ ПЛАТФОРМЫ **12**

Архитектура платформы B4N Service Platform 12

B4N Orchestrator 13

Назначение и основные функции 13

Портал управления 14

Оркестрация сетевых сервисов 15

Оркестрация вычислительных ресурсов 17

Каталог и конструктор сервисов 17

Тенанты и портал самоуправления 19

Интеграция с внешними системами 20

B4N Controller 20

B4N SwitchOS 21

B4N SwitchOS для платформы x86 22

B4N SwitchOS для Bare-metal коммутаторов 22

SDN ТРАНСПОРТ **24**

Построение топологии сети 24

Транспортные туннели 25

Механизмы инкапсуляции транспортных туннелей 26

VLAN swapping 26

MPLS swapping 28

Обеспечение гарантированного качества обслуживания 29

Сервисные модели 30

L2 point-to-point service 31

L2 Multipoint-to-Multipoint service 32

L2 Point-to-Multipoint service 33
Сервис L3VPN со статической маршрутизацией 35
Сервис L3 IP Multicast source tree 37
Резервные сервисные интерфейсы Reserve SI 39
Зеркалирование и отвод трафика (TAP) 39
Обеспечение отказоустойчивости сети 39
Отказ канала связи 40
Переключение трафика на резервный маршрут при возрастании количества ошибок или при наличии перегрузки на каналах связи 41
Link Dampening 41
Отказ коммутатора 43
Отказ контроллера 44
ОРКЕСТРАЦИЯ ВИРТУАЛЬНЫХ СЕТЕВЫХ ФУНКЦИЙ И СЕРВИСОВ 46
Архитектура NFV 46
Управление и оркестрация VNF 47
Состав пакета VNF 47
VNF дескриптор 48
Способы оптимизации производительности VNF 49
Политика использования ресурсов процессора 50
NUMA-топология 50
Использование больших страниц памяти 51
Управление сетевыми устройствами 51
Начальная конфигурация VNF 51
Политики размещения VDU (Affinity/Anti-Affinity) 52
Мониторинг VNF 52
Масштабирование VNF 54
Автоматическое восстановление VNF 54
Операции с VNF и VDU 54
Доступ к консоли VNF 57
Физические сетевые функции 57
Определение и характеристики PNF 57

Требования к PNF с точки зрения автоматизации управления 57
PNF Descriptor 58
Настройка PNF в каталоге оркестратора 58
Создание и запуск виртуальных сетевых сервисов 60
Выделение ресурсов тенанту 60
Создание сетевого сервиса с помощью графического конструктора сервисов 61
Изменение работающего сетевого сервиса 62
ТРЕБОВАНИЯ К АППАРАТНОМУ И ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ 64
Требования к аппаратному обеспечению для установки B4N Service Platform 64
Требования к программному обеспечению для установки B4N Service Platform 65
Требования к коммутаторам OpenFlow 65
ОБЕСПЕЧЕНИЕ БЕЗОПАСНОСТИ 66
СЦЕНАРИИ ПРИМЕНЕНИЯ 68
Применение технологий SDN/NFV при проектировании и модернизации сети агрегации региональных филиалов сервис-провайдера 68
Задачи и сервисы региональной сети передачи данных (РСПД) 68
Классы обслуживания, предоставляемые пакетной сетью РСПД 68
Архитектура сети РСПД. Транспортный уровень 70
Архитектура сети РСПД. Транспортный уровень 70
Механизм предоставления сервиса B2C абонентам 71
Механизм предоставления сервиса B2B абонентам 73
Доставка трафика IPTV 74
Организация присоединения сети РСПД к магистральной сети 74
Резервирование трафика в сети РСПД 74
ЗАКЛЮЧЕНИЕ 76

ВВЕДЕНИЕ

Платформа B4N Service Platform предназначена для модернизации сетевой инфраструктуры операторов связи и центров обработки данных с целью перехода на использование технологий SDN и NFV.

Фундаментальная концепция SDN (Software Defined Networking) меняет сложившиеся представления о современной архитектуре сетей путем введения понятия программируемых сетей и абстрагирования от физической сетевой инфраструктуры. Начальное, более узкое, определение SDN - разделение функционала управления сетью и потоками трафика (Control Plane) и функционала коммутации пакетных данных (Data Plane). Функции расчета сетевой топологии, таблиц коммутации трафика выполняются на отдельном устройстве – контроллере сети. Весь интеллект сети при этом находиться на контроллере сети. Функции коммутации и маршрутизации пакетов (Data Plane) выполняются на сетевом оборудовании с поддержкой загрузки таблиц коммутации с центрального контроллера сети, как правило, с помощью протокола OpenFlow.

Это позволяет масштабировать ресурсы, необходимые для контроля передачи трафика, независимо от ресурсов, которые требуются для коммутации пакетов, с целью обеспечить максимально эффективную загрузку оборудования. Кроме того, централизация уровня контроля уменьшает количество управляемых интеллектуальных устройств, упрощает эксплуатацию сети и делает возможным оркестрацию сервисов.

Развитие идеи централизации функций контроля сети позволяет дать более широкое определение SDN – это разработка стандартных протоколов и моделей данных, которые позволят организовать логически централизованное управление мультивендорными и многоуровневыми сетями. SDN-контроллеры предоставляют системам более высокого уровня абстрактную топологию и модели сервисов, упрощая оркестрацию сервисов и делая возможным появление новых приложений, использующих возможности программирования сетей.

Виртуализация сетевых функций NFV (Network Functions Virtualization) не связана напрямую, но во многих решениях дополняет SDN. NFV – это способ виртуализации функций элементов сети, позволяющий выполнять сетевые функции не с помощью специализированного оборудования, а реализовать их в виде программных модулей, работающих на стандартных серверах и виртуальных машинах (VM).

Внедрение решения B4N Service Platform позволит построить сеть, обладающую следующими характеристиками:

- Разделение функций контроля сети и передачи данных;
- Виртуализация сетевых функций;
- Программируемое управление сетью;
- Программируемое управление вычислительными ресурсами с помощью оркестратора;
- Использование стандартных протоколов;
- Единый механизм управления и выделения аппаратных ресурсов;
- Автоматизация управления и внедрения услуг в соответствии с бизнес-процессами;
- Автоматическая оркестрация ресурсов в соответствии с потребностями приложений или функций.

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ, ТЕРМИНОВ И СОКРАЩЕНИЙ

Сокращение	Расшифровка
Администратор	Функциональная роль по управлению распределенной сетью SDN.
Виртуальная машина (ВМ)	Полностью изолированный программный контейнер (модуль), работающий с собственной операционной системой и приложениями, аналогично физическому серверу. ВМ обладает собственными: – процессорными ресурсами, – местом на жестком диске, – оперативной памятью, – операционной системой.
Виртуальный ЦОД	Сервис аренды вычислительных мощностей для создания собственной инфраструктуры в «облаке». Характеризуется следующими параметрами: – Суммарная вычислительная мощность процессоров – Оперативная память – Дисковое пространство: объем и скорость – IP-сети: количество и размер, количество внешних IP адресов – Максимальное количество виртуальных машин
Домен	Множество элементов инфраструктуры, объединенных единой системой политик, предназначенных для реализации определенного функционала.
Оператор тенанта	Функциональная роль по управлению логически обособленной частью ресурсов - тенантом.
Конечный пользователь	Корпоративный пользователь сервисов сети SDN.
ЦОД	Центр обработки данных (Data Center) – специализированная технологическая площадка, предназначенная для размещения серверного и коммуникационного оборудования.
Customer edge	Граничный маршрутизатор клиента.
CP	Connection Point - Точка подключения пользователя к сетевому домену, без привязки к какому-либо сервису. Имеет разные типы инкапсуляции (Access, VLAN, QinQ и т.д.)
DC	Логическая сущность, представляющая агрегацию сетевых и вычислительных ресурсов (DataCenter, Telco PoP). Всегда вложена в Infrastructure Domain.

DCI	Data Center Interconnect – технологии и инфраструктура для объединения ресурсов разнесенных сайтов ЦОД
DPDK	Data Plane Development Kit - набор библиотек и драйверов для быстрой обработки сетевых пакетов в приложениях, работающих на архитектуре Intel.
eNNI	External network-to-network interface - Интерфейс, подключенный к внешней сети. Обычно используется совместно с динамическим протоколом маршрутизации с внешней сетью.
E-LAN	То же самое, что и M2M service
E-Line	То же самое, что и P2P service
E-Tree	То же самое, что и P2M service
EVC	Ethernet virtual circuits - виртуальное соединение Ethernet - - соединение двух или более пользовательских сетевых интерфейсов на втором уровне (данные передаются в виде Ethernet фреймов).
IaaS	Инфраструктура как услуга (Infrastructure as a service) – бизнес-модель обеспечения и использования виртуальной и аппаратной ИТ-инфраструктуры, при которой вычислительные мощности ЦОД предоставляются абонентам во временное пользование.
IGMP	Internet Group Management Protocol — протокол управления групповой (multicast) передачей данных в сетях, основанных на протоколе IP.
Infrastructure Domain	Инфраструктурный домен - Логическая сущность для группировки нескольких ЦОД (DC), SDN контроллеров.
KVM	Kernel-based Virtual Machine — программное решение, обеспечивающее виртуализацию в среде Linux на платформе x86.
L3VPN	Сервис контроллера, представляющий собой логический маршрутизатор между двумя или более L3-SI
M2M-service	Соединение типа «каждый с каждым». Например, соединение всех офисов в единую сеть, где каждый филиал может общаться с другими через логические туннели.
MPLS	Multiprotocol label switching — многопротокольная коммутация по меткам) — механизм в телекоммуникационной сети, осуществляющий передачу данных от одного узла сети к другому с помощью меток.

MPLS_TC	Поле «класс трафика» (traffic class) в метке MPLS; используется для реализации механизмов качества обслуживания (QoS) и явного уведомления о перегрузке; занимает 3 бита.
NaaS	Сеть как сервис (Network as a Service) – предоставление сетевых услуг конечным пользователям на инфраструктуре провайдера с применением единого процесса заказа и обеспечением стандартных процессов обслуживания.
NS (Network Service)	Сетевой сервис, набор виртуальных сетевых функций и соединений между ними.
NFV	Network Function Virtualization – виртуализация сетевых функций или реализация сетевых функций на виртуальной инфраструктуре.
NFVI	Network Function Virtualization Infrastructure – инфраструктура для обеспечения функционирования приложений NFV.
NFVO	NFV Orchestration – Оркестрация виртуализированных сетевых функций
NNI	Network-to-network interface - Интерфейс в сторону другого узла сети.
NS (Network Service)	Сетевой сервис, набор виртуальных сетевых функций и соединений между ними.
OF-port	Логический порт коммутатора, который организуется Openflow-агентом и через который контроллер взаимодействует с коммутатором по протоколу OpenFlow. Openflow-контроллер оперирует OF-портами коммутатора, для него Interface = OF-port.
Orchestrator	Система управления NFV / SDN-платформой
OSS	Домен систем эксплуатации и поддержки операций (Operation Support System) – набор корпоративных информационных систем, предназначенных для автоматизации процессов взаимодействия управляющего персонала, сервисов и элементов инфраструктуры.
OSS / BSS	Operation Support System / Billing Support System - Системы для управления услугами с точки зрения бизнеса
P2M-service	Соединение типа «все с одним». Например, соединение всех филиалов с центральным офисом, при котором филиалы не имеют возможности общаться между собой напрямую.

P2P-service	Туннель от одной точки до другой.
PE router	Provider edge. Граничный маршрутизатор провайдера услуги.
PCP	Priority code point - поле длиной 3 бита в заголовке 802.1Q, которое в соответствии со стандартом IEEE 802.1p определяет класс обслуживания трафика и влияет на приоритет его обработки.
SI	Service Interface - Сервисный интерфейс пользователя к конкретному сервису. Представляет собой CP, включенный в состав какого-либо сервиса.
QinQ	QinQ (IEEE 802.1QinQ) — расширение к стандарту IEEE 802.1Q, описывающее как тегированный трафик может передаваться внутри уже тегированного по 802.1Q трафика.
QoS (Quality of service)	Гарантированное качество сервиса.
RHEV	Red Hat Enterprise Virtualization - платформа виртуализации на базе гипервизора KVM.
SDN	Software-Defined Networking - программно-конфигурируемая сеть - сеть передачи данных, в которой уровень управления сетью отделён от устройств передачи данных и реализуется программно.
SDN контроллер	Базовый элемент сетевой инфраструктуры SDN - центральное управляющее устройство, осуществляющее контроль и программирование связанных между собой коммутаторов по какому-либо протоколу управления.
SLA (Service Level Agreement)	Соглашение между заказчиком и его поставщиком о параметрах и уровне качества сервиса
Tenant (Pool / Network Slice)	Логический пул ресурсов, выделенный клиенту или под конкретные задачи оператора. Тенант может объединять в себе ресурсы нескольких ЦОД, доменов и т. д. Например, если офисы клиента разнесены по разным городам страны, то в каждом городе ему могут быть выделены вычислительные и сетевые ресурсы, которые все вместе образуют один тенант.
UNI	User Network Interface (интерфейс подключения пользователя) - физический интерфейс/точка разграничения ответственности между сервис-провайдером и клиентом.

VDU	Virtual Deployment Unit - Виртуальная машина/ы в составе VNF.
VIM (Virtualized Infrastructure Manager)	Менеджер виртуальной инфраструктуры, ПО, отвечающее за конфигурацию и управление виртуальной инфраструктурой. В данный момент наиболее популярные VIM — это OpenStack и VMware vSphere vCenter.
VL (virtual links)	Связь любого типа между устройствами. Примеры: P2P, M2M и т. д.
VLAN	Virtual Local Area Network - виртуальная локальная сеть.
VM	Виртуальная машина.
VNF (Virtual Network Function)	Виртуальная сетевая функция. Пользователи применяют эти функции для создания виртуальной сетевой инфраструктуры. Сетевые функции представлены в каталоге. VNF могут быть простыми (состоящими из одной функции) или комплексными (состоящими из нескольких объединенных в группу виртуальных сетевых функций).
VNFD	Дескриптор, описывающий VNF.
VNFM	Менеджер виртуальных сетевых функций.
VNF Package	Пакет содержащий образы VDU, VNFD, конфигурационные скрипты.
YANG	YANG - модель данных, используемая для описания конфигурации и изменения состояния с использованием протокола NETCONF.

НАЗНАЧЕНИЕ

Платформа B4N Service Platform – это комплексное решение для модернизации сетевой инфраструктуры операторов связи, центров обработки данных и распределенных корпоративных сетей на базе технологий SDN и NFV. Платформа включает в себя компоненты для оркестрации сетевых сервисов, организации высокоскоростной обработки трафика и виртуализации сетевых функций.

Основные задачи системы могут быть разделены на следующие категории:

- управление ресурсами (контроллерами, коммутаторами, вычислительными серверами и т.п.);
- управление сервисами, задание шаблонов сетевых сервисов, их конфигурация и запуск сервисов;
- управление заказчиками (тенантами), выделение им сетевых и вычислительных ресурсов;
- объединение оборудования в кластеры для создания отказоустойчивых систем с динамическим распределением нагрузки;
- демонстрация текущего состояния сети в виде различных типов метрик, а также в виде физической и логической топологии;
- оперативное уведомление пользователей системы об ошибках или критическом состоянии оборудования;
- вывод логов о событиях и действиях пользователей в системе;
- управление пользователями и их ролями в системе.

Имеется возможность выбрать язык интерфейса - русский или английский.

Функциональность платформы включает в себя два взаимосвязанных и взаимодополняющих блока - SDN транспорт и оркестрация виртуальных сетевых функций. Далее в этом документе в соответствующих разделах приводится описание транспортной составляющей и возможностей платформы с точки зрения оркестрации VNF.

Описание разработанного на базе платформы решения B4N SD-WAN содержится в отдельном документе “B4N SD-WAN. DesignGuide”.

АРХИТЕКТУРА И СОСТАВ ПЛАТФОРМЫ

3.1. Архитектура платформы B4N Service Platform

Архитектура платформы представлена на рис. 1.

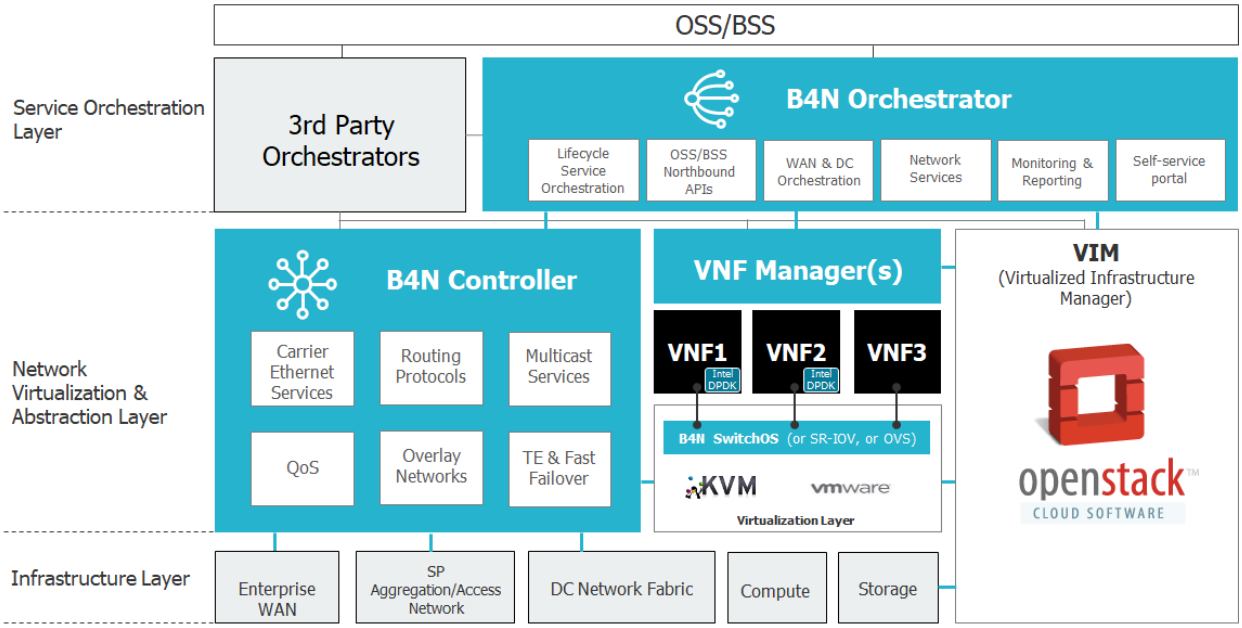


Рис.1. Архитектура платформы B4N Service Platform

Платформа B4N Service Platform включает в себя следующие компоненты:

- B4N Orchestrator – Единый графический интерфейс для управления платформой. Отвечает за выделение ресурсов, добавление и активацию виртуальных сетевых функций, оркестрацию и управление жизненным циклом сетевых сервисов. Оркестратор выполняет также функции интеграции с внешними системами (OSS/BSS, OpenStack с KVM, VMWare, RHEV или Hyper-V и др.);
- VNF Manager - отвечает за конфигурацию виртуальных сетевых функций;
- B4N Controller – высокопроизводительный кластер SDN-контроллеров с поддержкой OpenFlow v1.3;
- B4N SwitchOS – OpenFlow-агент для bare-metal коммутаторов или виртуальный коммутатор для работы в виртуализированных средах.

3.2. B4N Orchestrator

3.2.1. Назначение и основные функции

B4N Оркестратор обеспечивает сквозную (end-to-end) автоматизацию создания, развертывания и управления жизненным циклом сетевых сервисов, отвечает за согласованное функционирование всех компонентов, составляющих сервис. Оркестратор управляет одновременно доменами SDN (физическими и виртуальными) и NFV, декомпозируя запрос на создание сервиса на отдельные запросы к SDN-контроллерам и VIM, отвечающим за ресурсы в своих доменах, и затем “склеивая” все части вместе для создания единого end-to-end сервиса.

B4N Оркестратор включает в себя:

- Графический интерфейс управления инфраструктурой - сетевыми устройствами (через SDN-контроллеры), вычислительными и системными ресурсами;
- Каталог виртуальных сетевых функций (VNF) и шаблонов сетевых сервисов;
- Конструктор сервисов для создания цепочек сервисов;
- Портал самоуправления для Заказчиков, позволяющий управлять выделенными им ресурсами, логической топологией и сервисами;
- Северный (Northbound) интерфейс REST API для интеграции с оркестраторами других производителей и системами OSS/BSS.

Иерархическая структура сетевых, вычислительных и системных ресурсов, которыми оперирует оркестратор, представлена на рис. 2. В системе может быть несколько доменов, и домен может включать в себя несколько центров обработки данных.

Оркестратор не управляет напрямую ни коммутаторами, ни вычислительными узлами (серверами). Вместо этого SDN-контроллеры предоставляют оркестратору абстрактную топологию и модели сервисов, а менеджеры виртуальной инфраструктуры (VIM) - информацию о доступных вычислительных ресурсах.

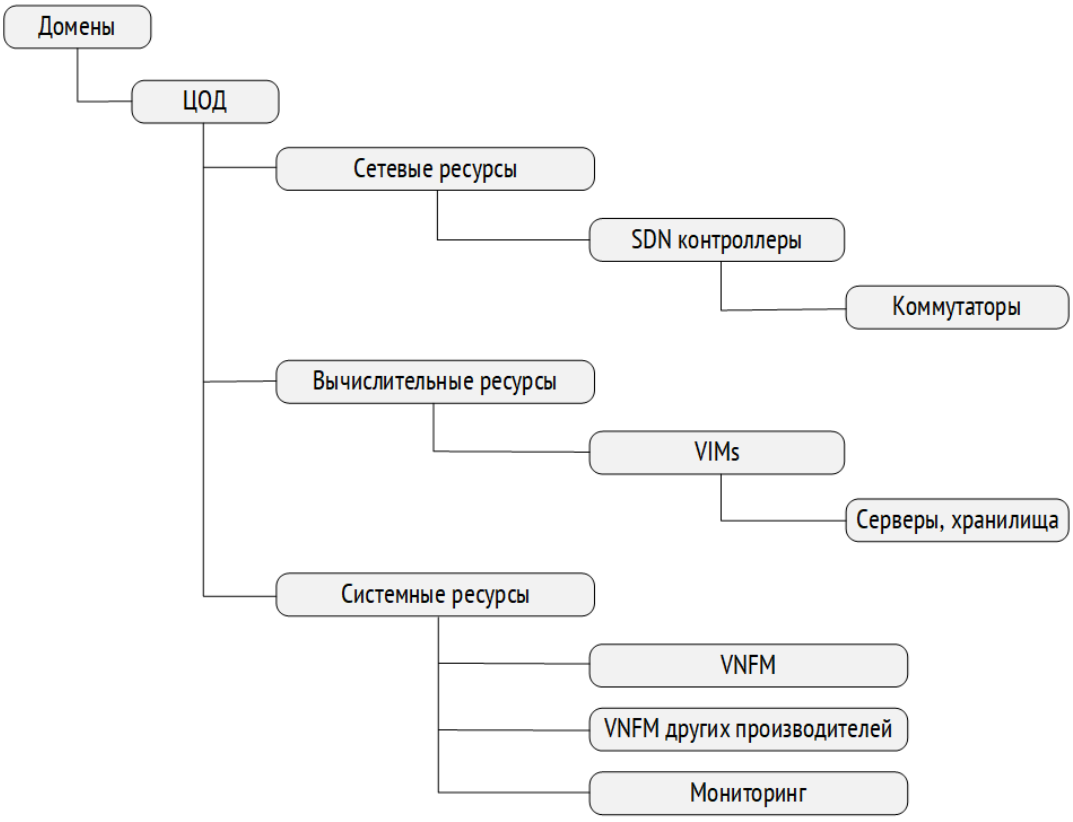


Рис.2. Централизованное управление ресурсами

3.2.2. Портал управления

Платформа B4N Service Platform предоставляет централизованный иерархический портал управления и контроля активации сервисов. Логическая архитектура интерфейса представлена на рис. 3.

Портал управления обеспечивает отображение всей инфраструктуры и состояний компонентов, в том числе с привязкой к географической карте. Интерфейс рассчитан на операторские сети с большим количеством объектов и имеет фильтры для сортировки и быстрого поиска необходимых компонентов.

Добавление и изменение конфигурации компонентов осуществляются с помощью мыши, без необходимости использования командной строки. Из этого же интерфейса осуществляется просмотр статистики и мониторинг компонентов системы.

Пользователи системы делятся на два типа:

- пользователи платформы имеют доступ ко всем ресурсам системы;
- пользователи тенанта при входе в систему могут видеть только ресурсы и объекты, принадлежащие данному тенанту.

Права пользователей на создание и изменение объектов определяются присвоенной им ролью.

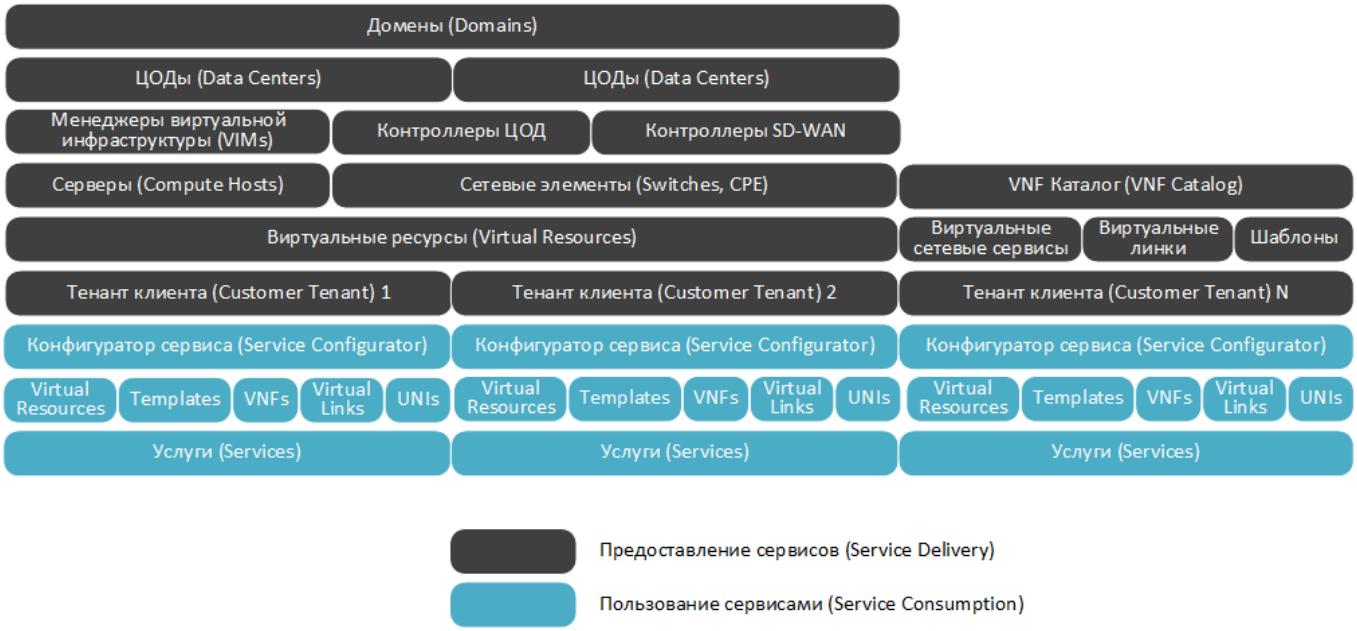


Рис.3. Логическая структура портала управления

Пользователи и группы могут быть локальными или импортироваться из внешней системы LDAP.

Возможен импорт из следующих систем LDAP:

- Linux OpenLDAP с Simple и Simple SSL аутентификацией
- MS Active Directory с Kerberos и Kerberos SSL аутентификацией.

Для импортированных пользователей и групп назначается роль в системе и при необходимости осуществляется привязка к тенанту,

3.2.3. Оркестрация сетевых сервисов

B4N Оркестратор, с помощью SDN-контроллеров, позволяет построить надежную и отказоустойчивую сетевую фабрику, состоящую из физических сетевых элементов - коммутаторов с поддержкой протокола OpenFlow, и виртуальных коммутаторов. Физические коммутаторы используются в сетях Metro Ethernet на уровне агрегации и в сетевой фабрике ЦОД, виртуальные коммутаторы применяются в облачных средах операторов (Telco Cloud). Это позволяет провайдерам услуг обеспечить связность между VNF клиентов и создать единую “SDN-фабрику” от сети доступа до конечного порта в ЦОД оператора (рис. 4). Такой подход обеспечивает гарантированную доставку и контроль трафика клиентов на всем пути от сети доступа до виртуальных сетевых функций.

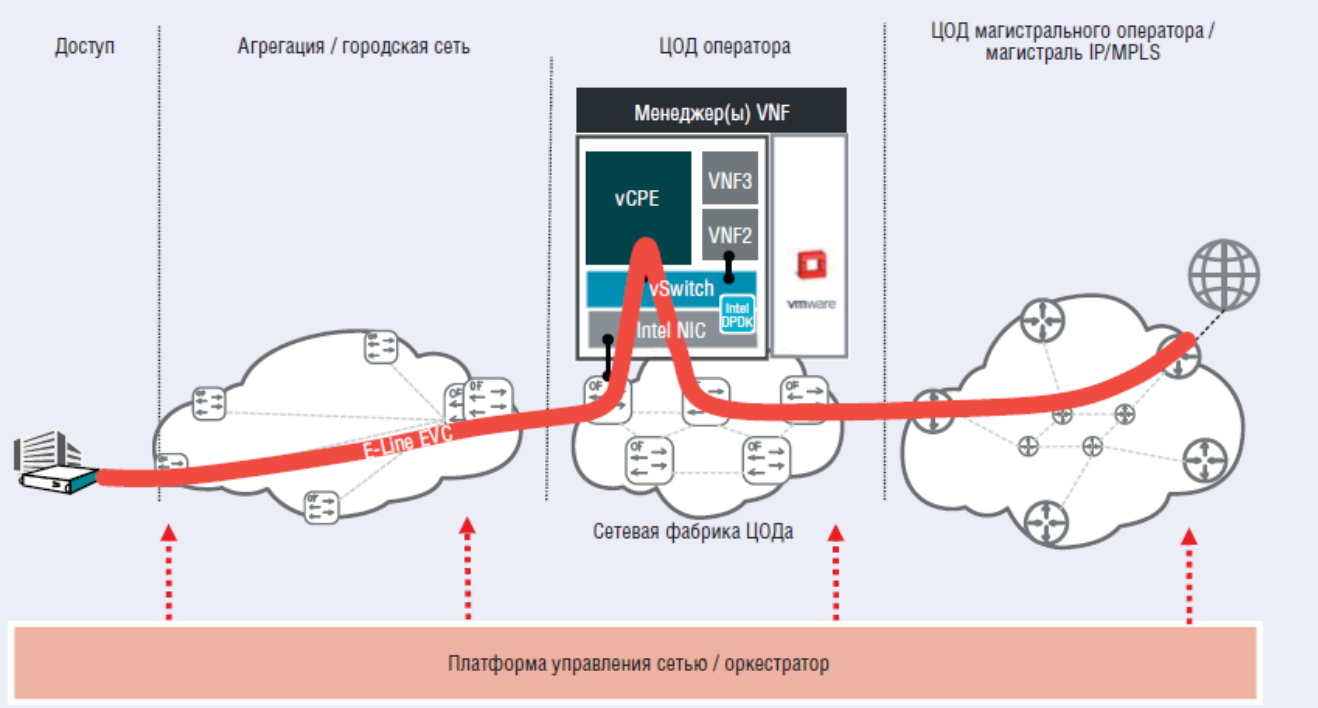


Рис. 4. Интеграция сервисов Carrier Ethernet с сетевыми сервисами в облаке оператора

Для сетей Metro Ethernet реализованы следующие функции:

- Сервисы Carrier Ethernet - E-Line (P2P), E-Tree (P2M), E-LAN EVCs (M2M);
- Сервис L3VPN со статической маршрутизацией;
- Типы UNI - untagged, VLAN, QinQ;
- Модель QoS с поддержкой до 8 классов обслуживания, классификации по полям VLAN PCP, полисинга на входных интерфейсах, назначения очередей на выходных портах;
- Маршрутизация по сегментам (Segment Routing) для определения кратчайшего пути при маршрутизации от источника;
- Traffic Engineering - расчет путей с учетом полосы пропускания или заданных ограничений;
- Построение карт - топология сети, движение трафика, схемы сервисов.

Для сетевой фабрики ЦОД'ов реализован набор сервисов, аналогичный решению для сетей Metro Ethernet. Сеть не разделяется на Underlay и Overlay. Оркестратор через контроллеры управляет физическими сетевыми устройствами и виртуальными коммутаторами. Коммутаторы должны поддерживать протокол Openflow 1.3.

Поддержка Intel DPDK на виртуальных коммутаторах позволяет приблизить их производительность к аппаратным коммутаторам за счет исключения сетевого стека

операционной системы из обработки пакетов. Приложение, в данном случае программный коммутатор, напрямую общается с сетевыми картами физического сервера.

В случае, если необходимо связать несколько отдельных сетевых доменов OpenFlow через участки сети, построенные с помощью классических коммутаторов и маршрутизаторов, могут быть использованы туннели с применением overlay-технологий.

Интеграция с сетевой частью инфраструктуры виртуализации под управлением OpenStack реализуется через встраиваемый модуль (plug-in) Modular Layer 2 (ML2) для neutron - компонента платформы OpenStack, отвечающего за предоставление сетевых услуг как сервиса. Взаимодействие с оркестратором осуществляется через Mechanism driver к модулю ML2.

3.2.4. Оркестрация вычислительных ресурсов

B4N Оркестратор контролирует виртуальные вычислительные ресурсы независимо от того, на каких физических серверах они развернуты, обеспечивает централизованное управление и мониторинг путем интеграции с внешней системой мониторинга. Данные мониторинга позволяют своевременно проводить автоматическое масштабирование и восстановление сетевых функций. Оркестратор обеспечивает автоматическое восстановление виртуальной сетевой функции в случае сбоя.

Управление вычислительными ресурсами реализовано с помощью менеджеров виртуальной инфраструктуры (VIM) других производителей. В настоящий момент поддерживается OpenStack. В качестве гипервизоров могут быть выбраны QEMU/KVM, XenServer и Xen Cloud Platform, Microsoft Hyper-V, VMware ESXi под управлением vCenter.

Список поддерживаемых гостевых операционных систем включает следующие популярные ОС:

- Windows Server 2003 / 2008 / 2008 R2 / 2012 / 2016;
- Red Hat Enterprise Linux 5 / 6 / 7;
- SUSE Enterprise Linux 10 / 11 / 12;
- CentOS 5 / 6 / 7 /8;
- Ubuntu Linux 12 / 14 / 16.

Полный список поддерживаемых гостевых ОС содержится в документации выбранного гипервизора.

Поддерживаются как 32-разрядные, так и 64-разрядные операционные системы.

Возможность объединять виртуальные ресурсы, принадлежащие нескольким VIM, в логические группы (домены) и назначать их тенантам обеспечивает гибкость при выделении ресурсов для отдельных клиентов или административных подразделений заказчиков.

3.2.5. Каталог и конструктор сервисов

Далее в данном документе будут использоваться следующие определения.

Сетевой сервис (Network service - NS) - цепочка из одной или нескольких VNF, соединенных виртуальными линками (Virtual Link - VL), имеющая начальную и конечную точки (Connection Point-CP).

Виртуальный линк имеет определенную топологию (P2P, или E-Line; P2M, или E-Tree; M2M, или E-LAN) и может содержать дополнительные требования (например, полоса пропускания, QoS и другие характеристики).

VNF Forwarding Graph определяет логическую топологию сетевого сервиса и представляет собой упорядоченную последовательность VNF, соединенных виртуальными линками.

Каталог оркестратора содержит следующие категории объектов:

- Виртуальные сетевые функции;
- Шаблоны сетевых сервисов;
- Виртуальные линки;
- Точки подключения сетевого сервиса.

Администратор платформы может добавлять в каталог новые виртуальные сетевые функции.

Пользователь создает требуемый сетевой сервис, выбирая нужные VNF, виртуальные линки и точки подключения из каталога и с помощью мыши перетаскивая их на граф в окно конфигуратора сервисов.

Администратор может заранее создать шаблоны сетевых сервисов, выбрав необходимые VNF, последовательность их выполнения и способы связи между ними. Эти шаблоны помещаются в каталог и могут быть в дальнейшем использованы для быстрого добавления сервисов пользователями.

Пример экрана графического интерфейса с конфигуратором сервисов приведен на рисунке 5.

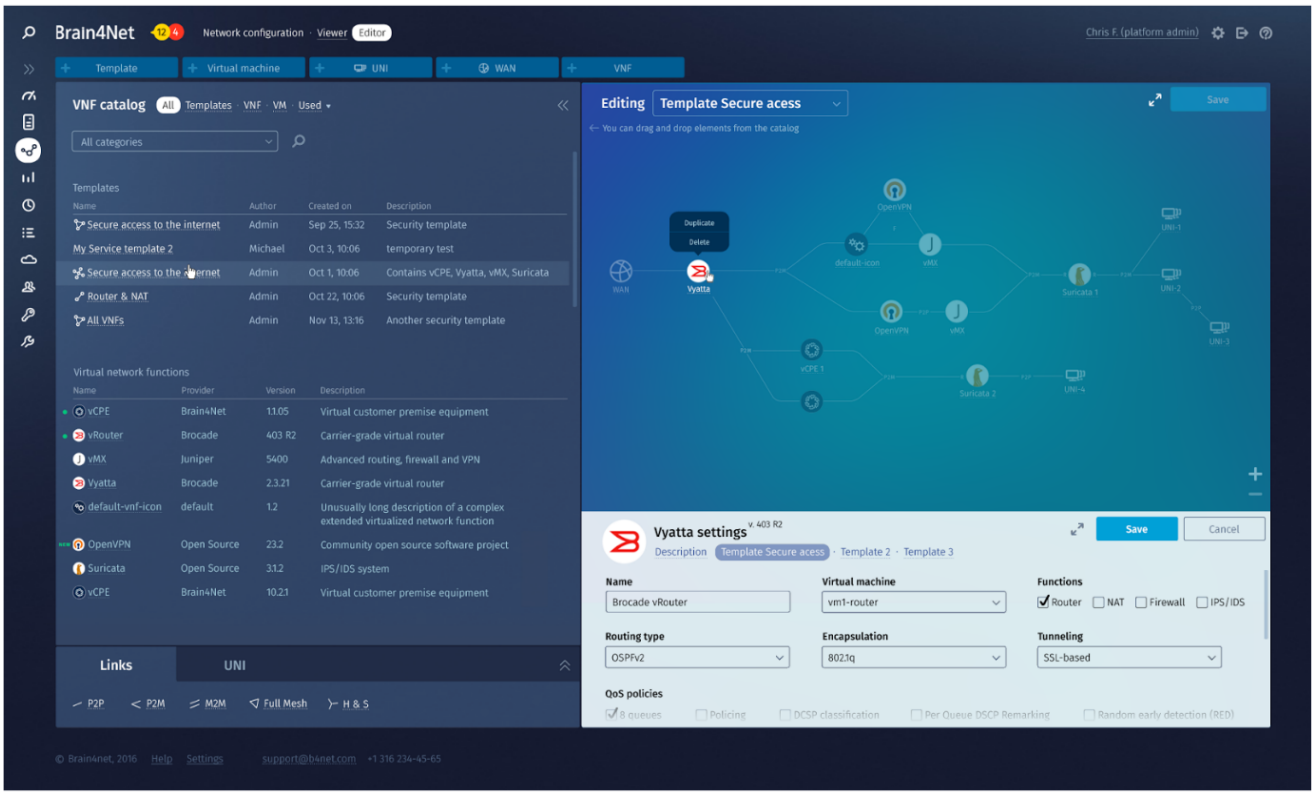


Рис. 5. Оркестрация сервисов

После того как сетевой сервис создан, пользователь может отредактировать параметры каждой виртуальной функции (масштабируемость, производительность, контроль доступа и другие настройки, специфичные для каждой VNF). После этого пользователь может активировать сервис. Далее развертывание сервиса происходит полностью в автоматическом режиме.

Оркестратор, используя информацию о ресурсах, предоставленную системами управления виртуальной инфраструктурой и SDN-контроллерами, определяет наиболее эффективное место развертывания VNF, осуществляет проверку доступности и выделение вычислительных ресурсов, в том числе находящихся в разных ЦОД и управляемых разными VIM, обеспечивает сетевую связность внутри и за пределами ЦОД до конечной точки услуги.

3.2.6. Тенанты и портал самоуправления

Тенант представляет собой логический набор ресурсов, предоставляемых Заказчику или выделяемых из общего пула с какой-либо другой целью. Каждый тенант содержит набор точек подключения (пользовательские сетевые интерфейсы - UNI), виртуальные сетевые функции из каталога, которые могут быть использованы для создания сервисов, ЦОД, где находятся вычислительные ресурсы, и группу VIM, под управлением которых будут работать виртуальные функции VNF.

Управление доступом в мультитенантной архитектуре включает в себя:

- управление пользователями и ролями,
- возможность доступа к нескольким тенантам,
- доступ к тенантам для инженеров технической поддержки.

К каждому тенанту привязываются учетные записи. Администратор тенанта в рамках выделенных ему ресурсов может выполнять следующие функции:

- просмотр топологии и переход к управлению выбранным компонентом сети,
- создание, удаление и редактирование сетевых сервисов в Конфигураторе сервисов, просмотр загрузки компонентов,
- управление виртуальными сетевыми функциями,
- управление пользователями и ролями,
- мониторинг системы.

Администратор платформы имеет возможность подключиться к тенанту в качестве администратора тенанта для настройки и устранения неполадок.

На рисунке 6 приведен пример страницы графического интерфейса для управления тенантами.

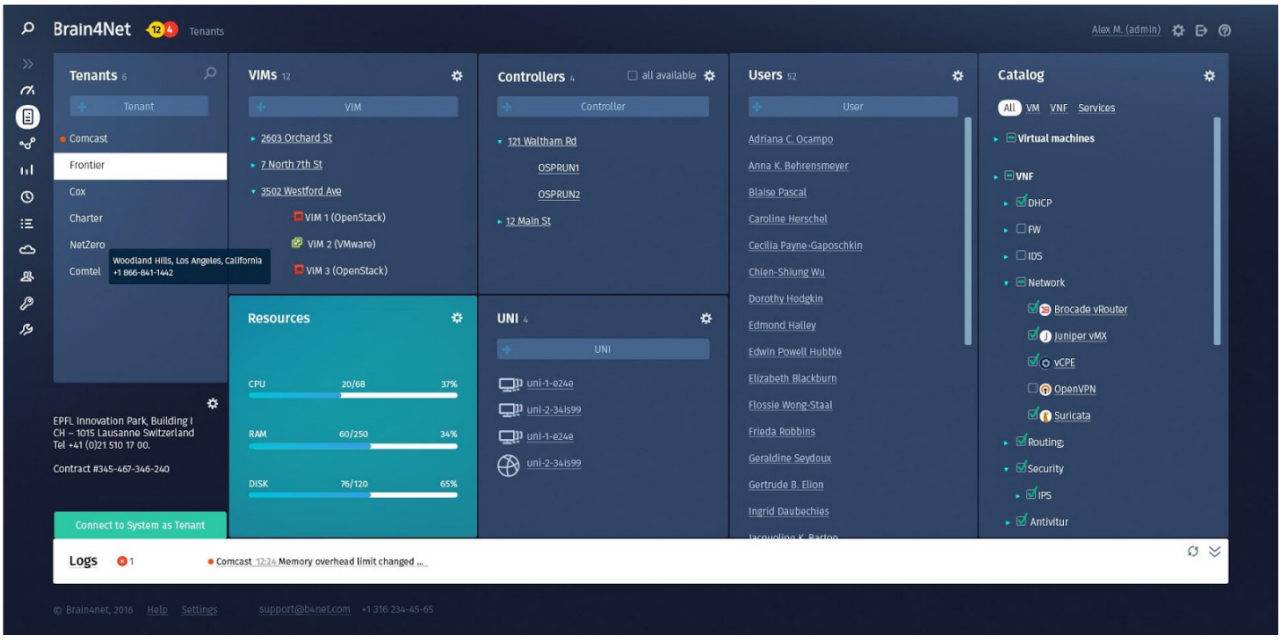


Рис. 6. Управление тенантами

3.2.7. Интеграция с внешними системами

Для интеграции с оркестраторами других производителей и OSS платформами в системе предусмотрен стандартный “северный” (Northbound) REST API, через который доступен весь имеющийся функционал.

Взаимодействие осуществляется по протоколу HTTP в формате json. Используется модель данных YANG.

Для интеграции с внешними системами мониторинга поддерживается протокол SNMP. Контроллер может выполнять роль SNMP-proxy для коммутаторов OpenFlow. При наступлении определенных событий (например, отказ устройства или обрыв соединения) контроллер посылает сообщения об обнаруженной неисправности, а также обеспечивает оперативное уведомление об ошибках или критическом состоянии оборудования.

3.3. B4N Controller

B4N Controller – высокопроизводительный SDN-контроллер на основе OpenFlow v1.3, осуществляющий управление OpenFlow-коммутаторами.

SDN-контроллер получает полную информацию о сетевых портах, соединениях и топологии, что позволяет в каждый момент времени иметь корректную и полную картину сети, централизованно внедрять политики и находить наиболее эффективные пути передачи трафика.

Архитектура продукта учитывает специфику распределенных сетей – протяженность, различные типы подключения филиалов / региональных узлов оператора связи, объемы трафика, ограничения полосы пропускания между филиалами / регионами и требования к качеству обслуживания.

Функции B4N Controller:

- Построение топологии сети и реакция на ее изменения
- Сегментирование сети, назначение уникальных меток, нахождение кратчайшего пути между сегментами
- Обработка входящих пакетов OpenFlow версии 1.3
- Обеспечение гарантированного качества обслуживания (QoS)
- Поддержка сервисов P2P, P2M, M2M
- Обработка трафика IP multicast, поддержка протоколов IGMPv2 / IGMPv3
- Сервис L3 VPN со статической маршрутизацией
- Обеспечение отказоустойчивости с помощью механизма кластеризации Master-Slave.

3.4. B4N SwitchOS

B4N SwitchOS – компонент B4N Service Platform, выполняющий функции высокоскоростной обработки трафика в SDN сетях. SwitchOS поддерживает работу со сторонними SDN контроллерами, обладает высокой производительностью и прост в использовании.

Существуют версии SwitchOS для различных платформ – виртуальный коммутатор для платформы x86 и OpenFlow-агент для Bare-metal коммутаторов (рис.7).

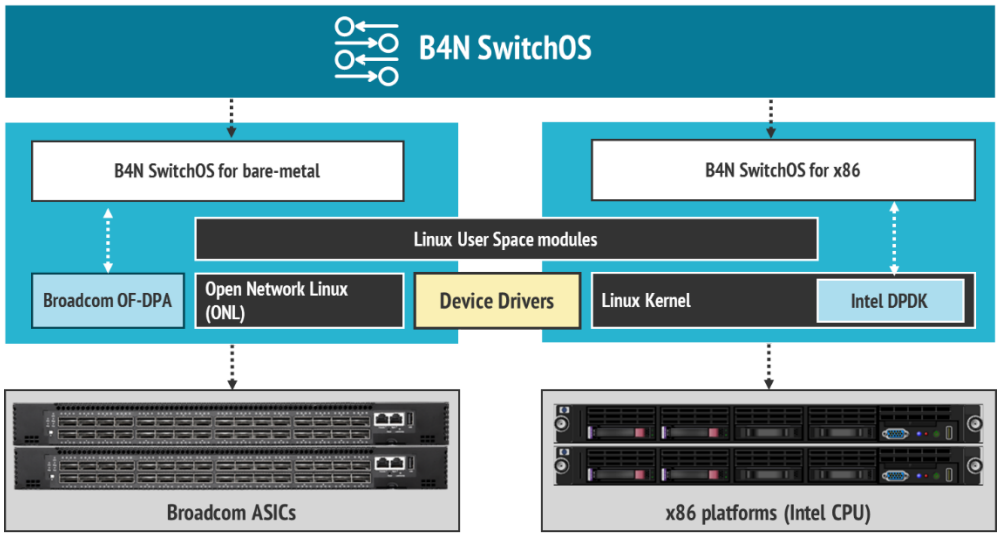


Рис.7. Архитектура виртуального коммутатора B4N SwitchOS

3.4.1. B4N SwitchOS для платформы x86

B4N SwitchOS для платформы x86 – это программный OpenFlow-коммутатор, реализующий функции высокоскоростной обработки трафика.

Благодаря использованию библиотек Intel DPDK и патентованных алгоритмов Brain4Net, решение позволяет добиться скорости обработки трафика, максимально приближенной к аппаратным коммутаторам.

Использование DPDK также позволяет заранее распределить аппаратные ресурсы наиболее оптимальным образом с точки зрения производительности программного коммутатора. Например, возможность привязывать порты сетевого адаптера к определенным ядрам процессора позволяет исключить накладные расходы, создаваемые планировщиком операционной системы при переключении задач. Благодаря использованию многопоточности, DPDK сокращает количество обращений к памяти и шине PCI, более эффективно используя процессорные мощности. Кроме того, DPDK позволяет оптимизировать использование памяти.

По сравнению с программными коммутаторами с открытым кодом B4N SwitchOS изначально создавался для установки в ЦОД сервис-провайдеров, поэтому обеспечивает поддержку функций, необходимых в операторских сетях:

- QinQ,
- QoS,
- Connectivity Fault Management (CFM),
- in-band и out-of-band подключение к SDN-контроллеру,
- одновременное подключение к нескольким SDN-контроллерам,

- возможность запросить статусы текущих контроллеров (подключен/отключен, master/slave),
- шифрованные соединения с контроллером (TLS).

В настоящий момент B4N SwitchOS поддерживает OpenFlow версии 1.3.5, в том числе queues и metering, select groups и group chaining, расширения OpenFlow для Hierarchical QoS. Список реализованных спецификаций протокола OpenFlow приведен в Приложении 1.

Требования для установки B4N SwitchOS:

- Сетевые карты, поддерживаемые DPDK - http://dpdk.org/doc/nics
- x86_64 процессор
- ОС GNU/Linux x86_64
- Установленный DPDK 16.0.4.

3.4.2. B4N SwitchOS для Bare-metal коммутаторов

B4N SwitchOS может быть установлен в качестве OpenFlow агента на коммутаторах с поддержкой OF-DPA 3.0.

OF-DPA - OpenFlow Data Plane Abstraction - модель абстрактного коммутатора, созданная для коммутаторов Ethernet на базе микросхем (Application-Specific Integrated Circuits - ASICs) Broadcom. Это открытая спецификация, основной целью которой является возможность программирования коммутирующей матрицы Broadcom с помощью инструкций протокола OpenFlow и превращение коммутатора с Broadcom ASIC в логический Openflow коммутатор.

Модель базируется на спецификации OpenFlow 1.3.4 и использует поддержку этой спецификацией нескольких таблиц flow tables для доступа к таблицам, реализованным в микросхеме Broadcom.

OF-DPA является программным компонентом, обеспечивающим уровень адаптации между агентом OpenFlow и коммутатором с Broadcom ASIC. Он располагается выше уровня Broadcom SDK (software development kit), который представляет собой низкоуровневый драйвер для настройки, программирования и управления чипсетом Broadcom (рис. 8).

Bare-metal коммутаторы на базе чипсета Broadcom поставляются с предустановленным ПО Open Network Installation Environment (ONIE). Фактически ONIE представляет собой загрузчик на основе Linux, который при первой загрузке ищет источник для установки полноценной сетевой ОС.

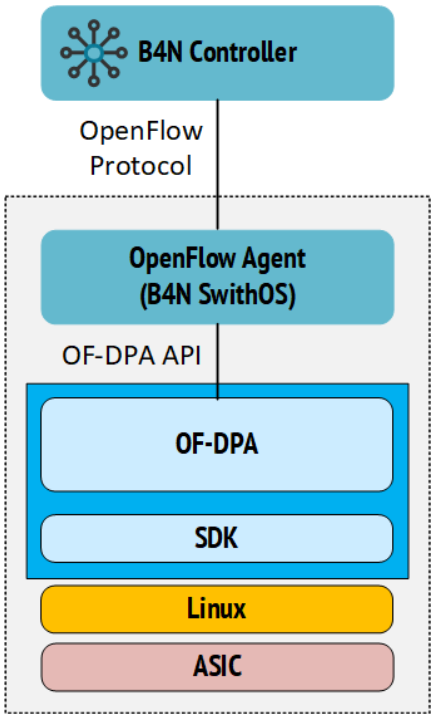


Рис.8. Компоненты OF-DPA

При первом включении коммутатора выполняется следующая последовательность действий:

- загружается ONIE;
- настраивается Ethernet интерфейс управления коммутатора;
- после установления соединения с сетью управления скачивается установщик ONL (Open Network Linux) installer;
- скачивается и устанавливается пакет OF-DPA и OpenFlow агент B4N SwitchOS.

При последующих перезагрузках устройства загружается OF-DPA и OpenFlow агент. ONIE остается на коммутаторе и используется только когда требуется переустановить ОС.

Список протестированных моделей коммутаторов постоянно пополняется и в настоящий момент включает в себя:

- EdgeCore 5812, 5912, 6712,
- Bulat BS7510, BR100
- Netberg Aurora 420
- Eltex MES5448

SDN ТРАНСПОРТ

1.1. Построение топологии сети

В решении B4N основной структурной единицей является сетевой домен, состоящий из следующих элементов:

- набора коммутаторов Ethernet, имеющих программный интерфейс управления OpenFlow версии 1.3 и выше, соединенных между собой каналами с произвольной топологией;
- контроллера или кластера контроллеров B4N Controller, имеющих IP-соединение с интерфейсами управления коммутаторов по выделенной сети управления (out-of-band) или через коммутирующие порты (in-band).

B4N Controller автоматически строит топологию сети и поддерживает ее актуальность в процессе работы. Топология может быть представлена с привязкой к географической карте или в виде логической схемы. При добавлении в сеть нового коммутатора он автоматически обнаруживается контроллером.

После построения топологии порты коммутаторов разделяются на:

- Магистральные порты, соединяющие коммутаторы домена между собой;
- Внешние (сервисные) порты, которые могут использоваться для клиентских подключений.

Коммутатор, который имеет и магистральные, и внешние порты, является пограничным коммутатором.

На магистральных портах организуются транспортные туннели, предназначенные для переноса трафика между узлами домена.

На сервисных портах могут быть организованы сервисные интерфейсы. Сервисный интерфейс представляет собой связку: конкретный коммутатор плюс конкретный порт на этом коммутаторе плюс инкапсуляция (Access, VLAN, QinQ).

Обобщенная модель сети представлена на рисунке 9.

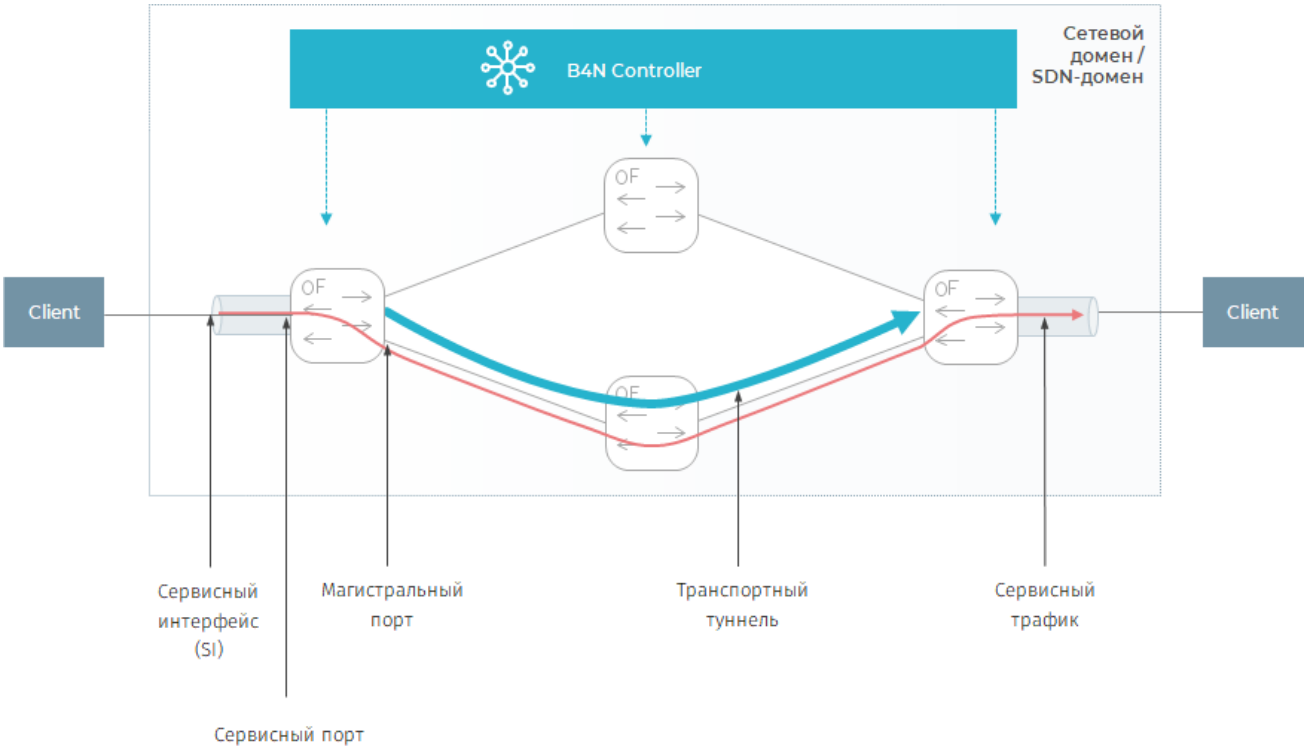


Рис. 9. Сетевой домен

Транспортный уровень сетевого домена строится на основе транспортных туннелей, соединяющих узлы домена через магистральные порты по принципу «каждый с каждым».

Транспортный уровень существует независимо от сервисного уровня и обеспечивает максимально возможную связность между коммутаторами и построение актуальной топологии.

1.2. Транспортные туннели

Транспортный туннель строится как направленный граф от начального узла до конечного и включает в себя:

1. узел инициации (начальный узел)
2. 1 или более магистральных каналов
3. 0 или более транзитных узлов
4. узел терминации (конечный узел)

Обобщенная схема транспортного туннеля показана на рисунке 10.

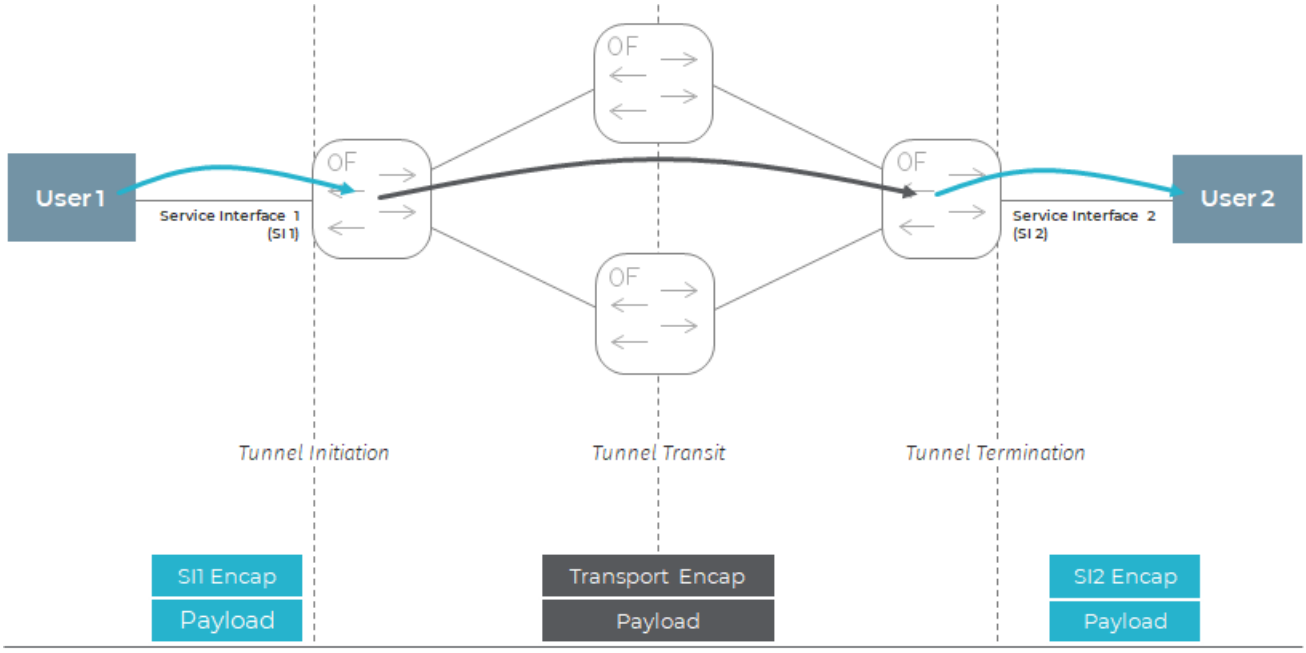


Рис. 10. Транспортный туннель

Для построения транспортных туннелей контроллер применяет алгоритм расчета кратчайших путей между узлами сетевого домена на основе метрики стоимости (cost) магистральных каналов. На данный момент веса каналов рассчитываются по формуле:

Cost = 1 000 000 / Bandwidth(Mbps)

При этом между любой парой коммутаторов рассчитываются основной и резервный путь транспортного уровня. Эти пути используются при программировании FastFailover-групп, определяющих основной выходной порт коммутации трафика к определенному узлу и резервный порт в случае отказа основного. Таким образом реализуется механизм FastReroute на транспортном уровне.

Механизм инкапсуляции транспортных туннелей может быть различным в зависимости от используемых коммутаторов и их характеристик. На данный момент поддерживаются механизмы инкапсуляции VLAN swapping и MPLS swapping.

1.3. Механизмы инкапсуляции транспортных туннелей

1.3.1. VLAN swapping

При использовании данного механизма инкапсуляции транспортный уровень адресуется стеком из одной дополнительной метки VLAN, вставляемым в пакет при отправке в магистраль домена. При этом метка меняется на каждом промежуточном узле в соответствии с рассчитанным контроллером кратчайшим или резервным путем.

На пограничных коммутаторах OpenFlow производятся следующие действия с пакетами:

- сбрасывается метка VLAN (pop-VLAN), в метаданные заносится информация о номере VLAN получателя и значении поля PCP;
- добавляется метка сервисного VLAN (push-VLAN), по ней будет определяться получатель пакета на конечном узле сети;
- добавляется метка транспортного VLAN (push-VLAN);
- заносится требуемое значение в поле VLAN_PCP транспортного VLAN (в зависимости от настроек QoS и информации в метаданных);
- назначается номер очереди исходящего порта коммутатора;
- пакет направляется в магистральный порт.

Схема инкапсуляции трафика при использовании механизма VLAN swapping показана на рисунке 11.

Все действия с метками (сброс, добавление) всегда относятся к внешней метке пакета. Например, когда добавляется новая метка VLAN, она вставляется сразу после заголовка Ethernet, перед другими метками, если они присутствуют.

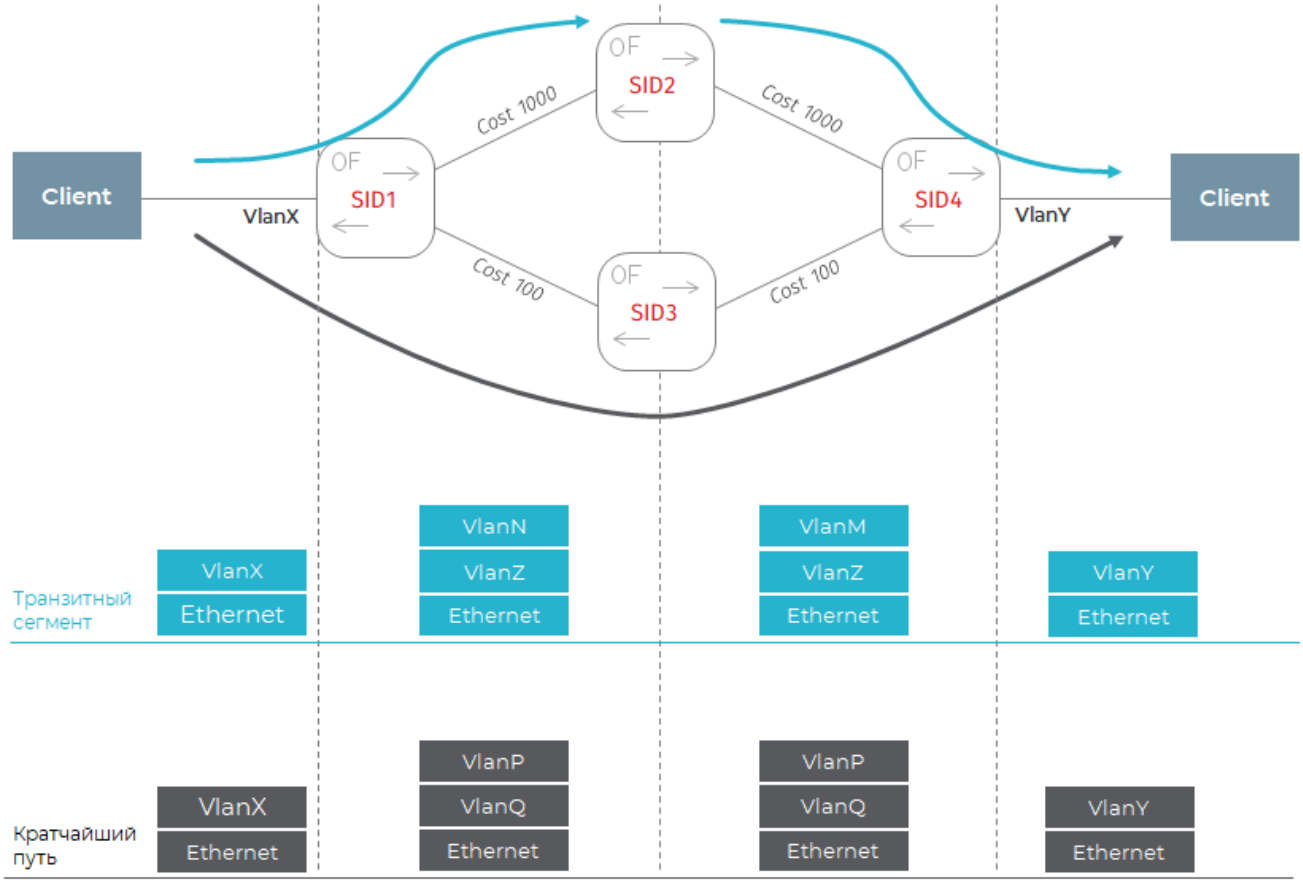


Рис. 11. VLAN swapping

1.3.2. MPLS swapping

Механизм инкапсуляции транспортных туннелей MPLS Swapping использует тот же алгоритм назначения меток на транспортный сегмент, что и VLAN swapping. Диапазон разрешенных значений в этом случае меняется с 8-4094 на 16-1048576 (нижние 16 меток зарезервированы и их использовать запрещено).

После подключения коммутатора к контроллеру и обнаружения (обновления) топологии сети выполняется алгоритм расчета основного и резервного транспортных сегментов от начального до конечного коммутаторов. Контроллер выбирает набор mpls меток для этих транспортных сегментов, метка меняется на каждом промежуточном узле.

На пограничных коммутаторах выполняются следующие действия:

- в метаданные заносится информация о номере входящего порта, значении поля PCP;
- в соответствии с настройками CoS определяется внутренний класс обслуживания, выполняется полисинг;
- формируется новый Ethetnet заголовок (MAC адреса источника и назначения, VLAN тип и ID, PCP в соответствии с внутренним классом обслуживания);
- добавляется сервисная MPLS метка (push_mpls), уникальная для каждого сервиса. По ней будет определяться получатель пакета на конечном узле сети;
- добавляется транспортная метка MPLS (push_mpls);
- назначается номер очереди исходящего порта коммутатора;
- пакет направляется в магистральный порт.

На каждом транзитном коммутаторе транспортная MPLS метка меняется на другую, рассчитанную алгоритмом, и на конечном коммутаторе обе MPLS метки снимаются (рис. 12).

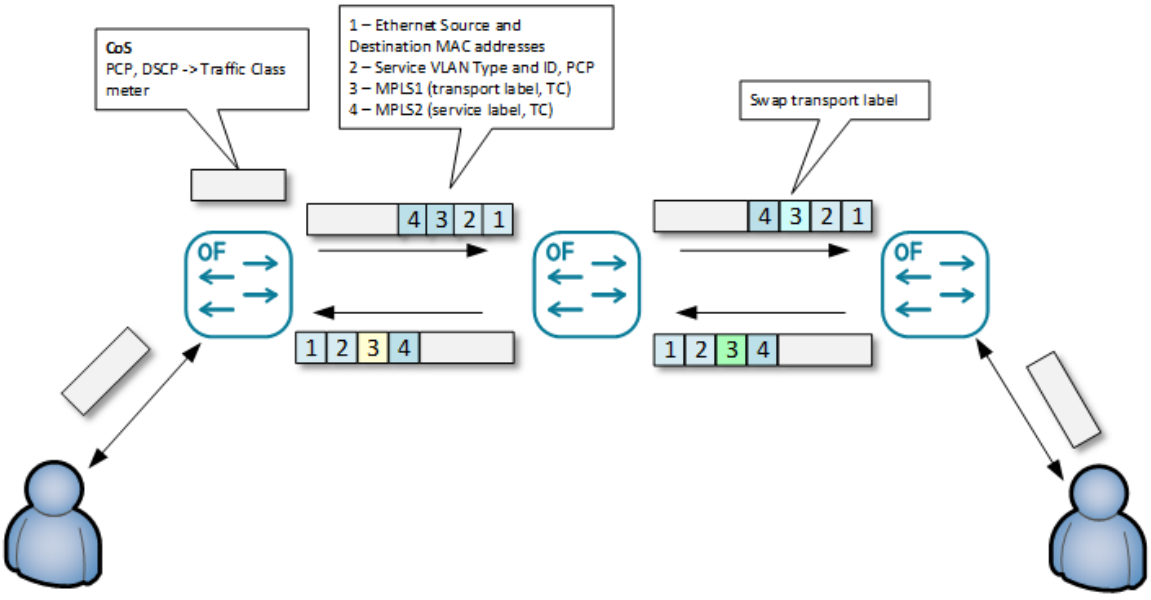


Рис. 12. MPLS swapping

1.4. Обеспечение гарантированного качества обслуживания

B4N Controller обеспечивает поддержку качества обслуживания (QoS), в том числе:

- Полисинг входящего трафика;
- Классификацию входящего трафика;
- Отображение классов трафика на аппаратные очереди коммутаторов.

Модель QoS внутри домена может включать от 3 до 7 классов обслуживания:

1. RealTime – трафик реального времени (голос, видео). Для данного класса трафика задается максимальная полоса. При выходе из магистрального или сервисного порта трафик класса направляется в очередь StrictPriority.
2. Business Critical – трафик критичных приложений. Можно создать один или несколько классов данного типа. Для трафика Business-Critical задается гарантированная и максимальная полоса пропускания. При выходе из магистрального или сервисного порта трафик класса отправляется в очередь WFQ или WRR.
3. BestEffort – трафик, не предъявляющий жестких требований к качеству обслуживания. Трафик класса маппируется в очередь с наименьшим приоритетом при выходе из магистрального или сервисного порта.

Внутри домена классы обслуживания маркируются с помощью VLAN_PCP. При входе в магистральный порт используется доверительная политика указанной маркировке.

Для классификации трафика при входе в сервисный интерфейс используется конфигурируемый классификатор, определяющий соответствие критериев входящего пакета внутреннему классу обслуживания. Критерии, которые могут быть использованы для классификации:

1. Весь трафик сервисного интерфейса
2. VLAN PCP
3. IP DSCP

Для настройки полисинга при входе в сервисный интерфейс используется полисер трафика, определяющий:

- Максимальную полосу пропускания на весь сервисный интерфейс;
- Распределение полосы пропускания между определенными ранее классами трафика.

Классификатор и полисер объединяются в QoS-профиль, который применяется к сервисному интерфейсу.

Точки и политики применения QoS на сети показаны на рисунке 13.

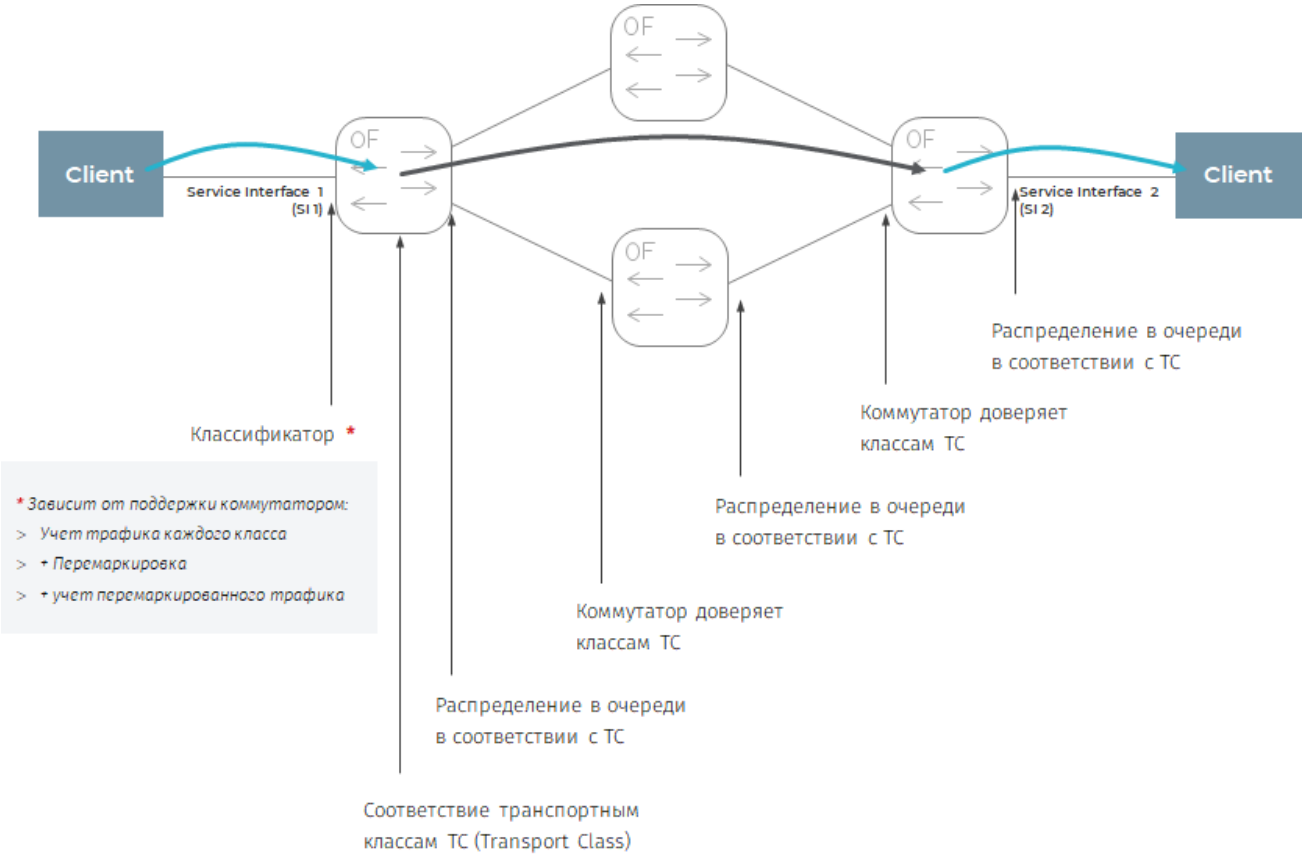


Рис. 13. Механизм работы QoS внутри домена

1.5. Сервисные модели

Основным примитивом для построения сервисов является сервисный интерфейс. Он определяется на внешних сервисных портах одним из следующих способов:

- весь физический порт
- 802.1Q VLAN на физическом порту
- Внешний 802.1ad VLAN + внутренний VLAN

Сервисный интерфейс при включении в сервис приобретает следующие дополнительные атрибуты:

1. Входной QoS-профиль, содержащий классификатор и полисер входящего трафика;
2. Счетчики входящего и исходящего трафика;
3. Параметры состояния;
4. Обработчики OAM-трафика;
5. IP и MAC адрес (для L3-сервисов).

Сервис в сетевой модели B4N создается путем объединения сервисных интерфейсов на разных коммутаторах через транспортные туннели с применением определенной логики прохождения трафика между сервисными интерфейсами.

С точки зрения технологии OpenFlow сетевой сервис представляет собой набор правил обработки пакетов (flows) на коммутаторах, реализующих определенную логику передачи пакетов.

Реализованы следующие сервисные модели:

- Сервисы L2 Ethernet
    - L2 point-to-point service (E-Line в классификации MEF);
    - L2 multipoint-to-multipoint service (E-LAN в классификации MEF);
    - L2 point-to-multipoint service (E-TREE в классификации MEF);
- Сервис L3VPN со статической маршрутизацией
- L3 IP Multicast source tree (Service Provider Multicast).

1.5.1. L2 point-to-point service

Сервис L2 point-to-point состоит из следующих элементов:

1. Два сервисных интерфейса;
2. Транспортный туннель, соединяющий узлы, на которых находятся указанные сервисные интерфейсы.

Ethernet-фреймы прозрачно проходят между двумя сервисными интерфейсами, за исключением нормализации инкапсуляции сервисных интерфейсов, если они различны.

L2-протоколы на сервисных интерфейсах не обрабатываются.

Схема сервиса представлена на рисунке 14.

При создании сервиса можно применить механизм TrafficEngineering для данного сервиса, указав промежуточные пути прохождения транспортного туннеля. При этом для MPLS-swapping это достигается указанием стека меток в начале туннеля, а для VLAN-swapping прокладывается новый транспортный туннель в соответствии с указанными промежуточными точками.

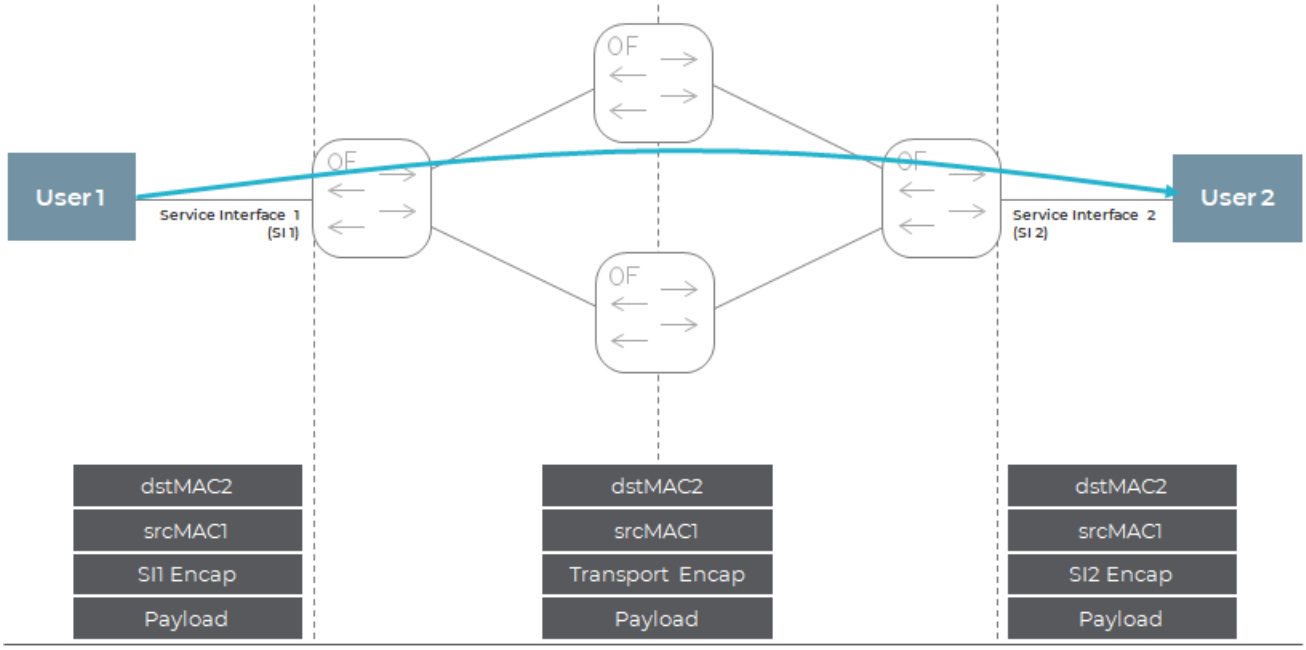


Рис. 14. Сервис L2 point-to-point

1.5.2. L2 Multipoint-to-Multipoint service

Сервис L2 Multipoint-to-Multipoint реализует функционал сервиса MEF E-LAN с поддержкой как EP-LAN, так и EVP-LAN. Сервис состоит из следующих элементов:

1. Два или более сервисных интерфейсов;
2. Общая сервисная таблица MAC-адресов, образующая общий распределенный bridge-domain сервиса.

Обеспечивается полная связность между сервисными интерфейсами. Каждый SI может передавать трафик на другой SI, входящий в сервис.

Для каждого сервисного интерфейса есть возможность указать Reserve SI – резервный сервисный интерфейс, который может находиться на этом же или на другом коммутаторе.

Схема сервиса представлена на рисунке 15.

Сервис использует механизм mac-learning через контроллер для заполнения MAC-таблицы. В начальном состоянии любой входящий в сервисный интерфейс пакет пересылается на контроллер. Имеется возможность указать максимальное количество MAC-адресов на сервис и выбрать действие контроллера при переполнении MAC-таблицы – производить широковещательную рассылку пакетов с невыученными MAC-адресами или отбрасывать их.

Изучение Ethernet MAC адреса происходит путем отправки части пакета на контроллер в сообщении при его получении на сервисном интерфейсе. Отправка пакета на контроллер

происходит только в случае, если Ethernet MAC адрес в поле адреса источника заголовка Ethernet пакета не был изучен ранее или запись об этом MAC адресе устарела.

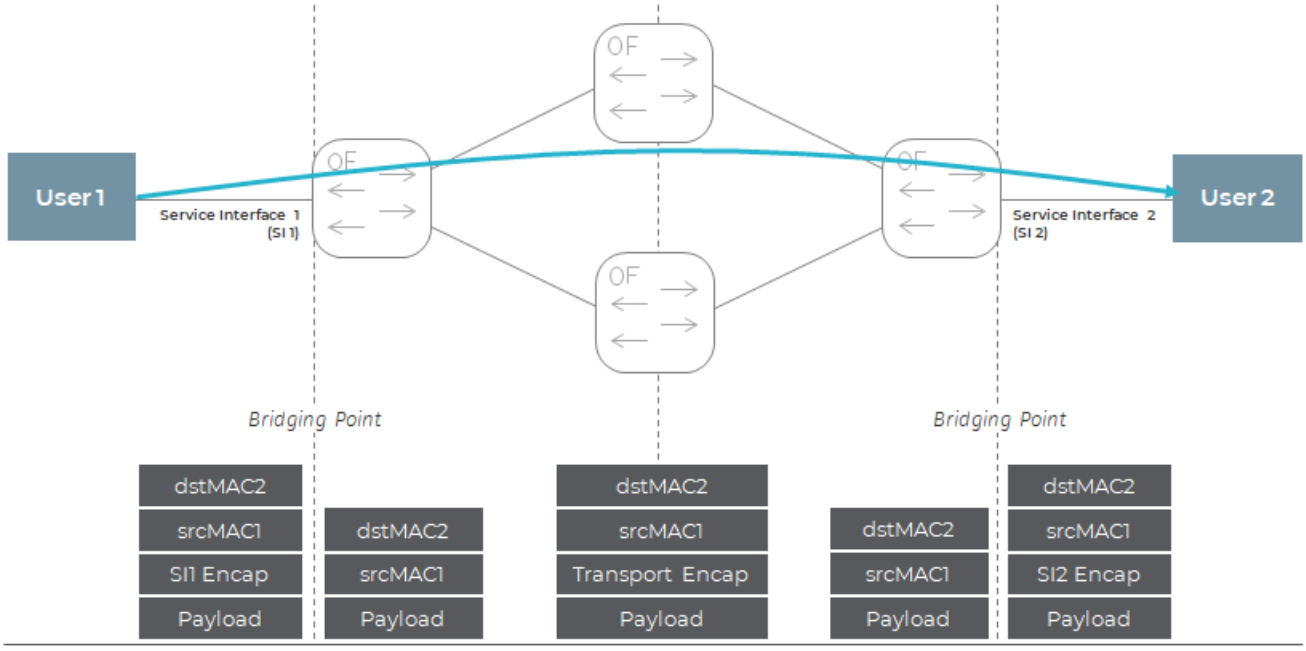


Рис. 15. Сервис L2 multipoint-to-multipoint

Поддерживается два режима изучения Ethernet MAC адресов:

- Learn and flood - одновременно с отправкой сообщения OFPT_PACKET_IN на контроллер происходит репликация трафика на все порты по правилам для BUM (broadcast, unknown-unicast and multicast) трафика. Это позволяет изучать Ethernet MAC адреса без потери пакетов, но создает избыточную нагрузку на контроллер.
- Learn and drop - при формировании сообщения OFPT_PACKET_IN на контроллер пакет отбрасывается. Этот механизм приводит к потерям пакетов при изучении MAC-адресов, но уменьшает нагрузку на контроллер.

При создании или редактировании сервиса можно указать MAC Age Timeout - время жизни Ethernet MAC-адреса в MAC таблице, от 0 до 65535 сек. Если выбрано значение 0, то записи в MAC таблице не устаревают. Значение по умолчанию – 3600.

1.5.3. L2 Point-to-Multipoint service

Сервис point-to-multipoint (P2M) является разновидностью сервиса multipoint-to-multipoint, работа которого соответствует сервису E-Tree MEF.

Сервис поддерживает произвольные топологии с любым количеством SI, при этом каждому SI должна быть назначена одна из ролей:

1. Root
2. Leaf

Роль определяет ограничения на прохождение трафика:

- Пакет, вошедший в сеть через Root, может быть направлен как в Root, так и в Leaf.
- Пакет, вошедший в сеть через Leaf, может быть направлен только в Root.

Сервис должен содержать два или более SI. Как минимум один из сервисных интерфейсов должен иметь роль Root.

Изучение MAC адресов происходит для всех сервисных интерфейсов, независимо от их роли, а коммутация трафика осуществляется в соответствии с ролями SI.

Кроме классической схемы работы сервиса P2M возможны варианты, когда интерфейс Root назначается интерфейсом по умолчанию - Default forwarding interface (DFI):

- DFI с таблицами FIB на сервисных интерфейсах Root и Leaf
- DFI с таблицами FIB на сервисных интерфейсах Leaf.

Все режимы с DFI рассчитаны на увеличение масштабируемости сервиса. Например в случаях подключения B2C клиентов к услуге доступа в Интернет, количество клиентов B2C значительно превышает количество BRAS. При этом нет необходимости идентифицировать каждого клиента по его уникальному MAC адресу, достаточно знать, за каким коммутатором доступа он расположен. Учитывая это, можно существенно уменьшить количество OpenFlow правил на коммутаторах.

Примеры топологий для сервиса P2M с использованием DFI приведены на рис. 16.

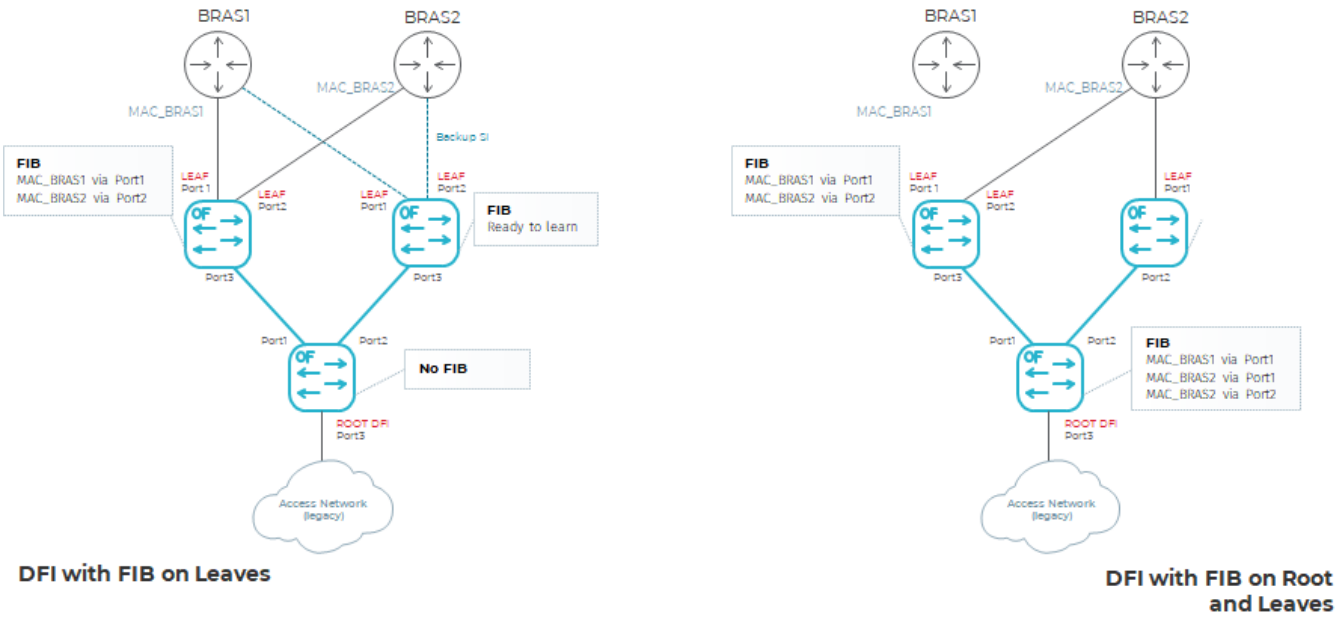


Рис. 16. Примеры топологий сервиса L2 point-to-multipoint с DFI

Все варианты работы сервиса с DFI рассчитаны на увеличение масштабируемости сервиса. Например в случае подключения B2C клиентов к услуге доступа в Интернет количество клиентов B2C значительно превышает количество BRAS. При этом нет необходимости идентифицировать каждого клиента по его уникальному MAC адресу, достаточно знать, за каким коммутатором доступа он расположен. Учитывая это, можно существенно уменьшить количество правил OpenFlow на коммутаторах.

При использовании варианта DFI с таблицами FIB на сервисных интерфейсах Root и Leaf изученные за SI Leaf MAC-адреса прописываются в таблицы OpenFlow на всех коммутаторах, участвующих в сервисе. Это позволяет оптимизировать пути распространения трафика до SI Leaf и также поддерживать конфигурации, при которых SI Leaf и SI Root могут располагаться на одном и том же коммутаторе.

Вариант DFI с таблицами FIB на сервисных интерфейсах Leaf может быть использован в ситуациях, когда два BRAS подключены к одному коммутатору и резервируются по схеме ACTIVE-ACTIVE, для чего используется FIB на портах с ролями Leaf. При этом для каждого порта с ролью Leaf можно задать резервный интерфейс Reserve SI, на который будет переключен трафик в случае отказа коммутатора с основным портом.

При выборе этих типов сервиса необходимо учитывать, что сервис может содержать только один Root SI (который назначается интерфейсом по умолчанию DFI). С целью обеспечения отказоустойчивости для любого типа SI может быть определен резервный интерфейс Reserve SI.

1.5.4. Сервис L3VPN со статической маршрутизацией

Сервис L3VPN со статической маршрутизацией позволяет предоставить клиентам услугу создания маршрутизируемой сети между непосредственно подключенными сетями клиента с возможностью добавления статических маршрутов. Поддерживается полносвязная топология Full-Mesh и топология с общим центром и несколькими удаленными узлами Hub-and-Spoke (рис.17).

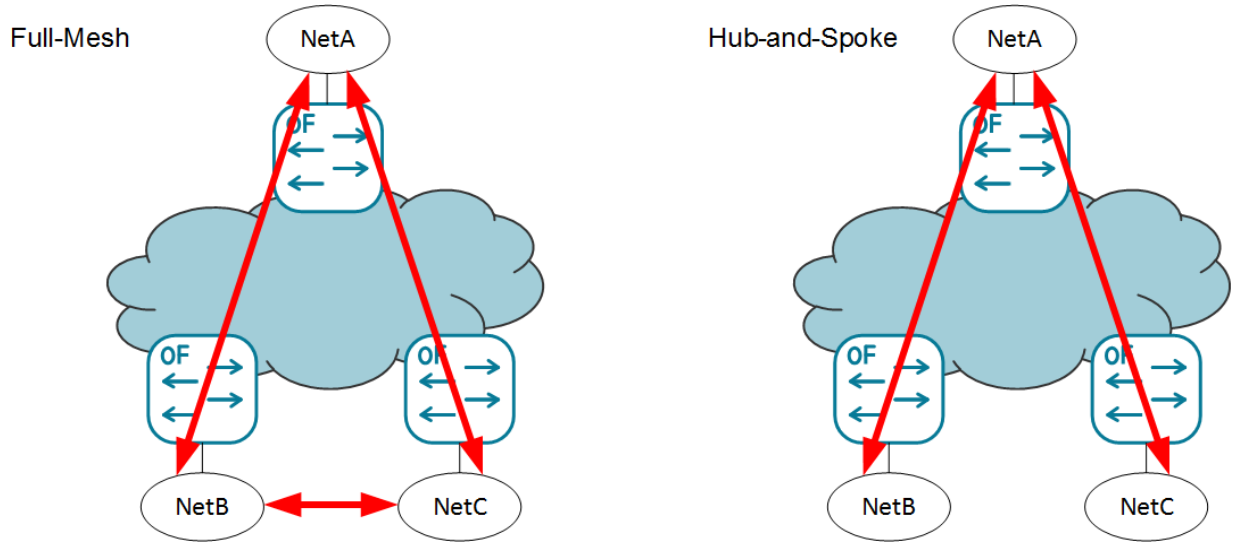


Рис. 17. Примеры топологий

В Full-Mesh топологии разрешается взаимодействие между любыми сетями клиента. В Hub-and-Spoke топологии разрешается взаимодействие только между центральным узлом и удаленными узлами клиента. При необходимости взаимодействие между сетями, подключенными к удаленным узлам, может осуществляться через центральный узел (рис 18).

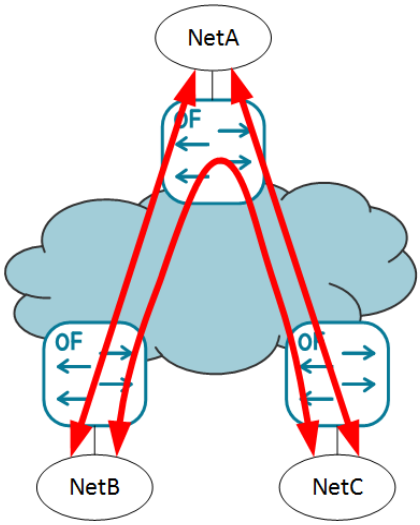


Рис. 18. Сервис L3VPN Hub-and-Spoke с разрешенными соединениями между сетями удаленных узлов через центральный узел

Сервис L3VPN поддерживает добавление статических маршрутов на подсети клиента, которые располагаются за маршрутизаторами, находящимися не под управлением B4N и подключенными одним из своих интерфейсов к сети, непосредственно подключенную к сети SDN (например, netD). Примеры таких сетей представлены на рисунке 19.

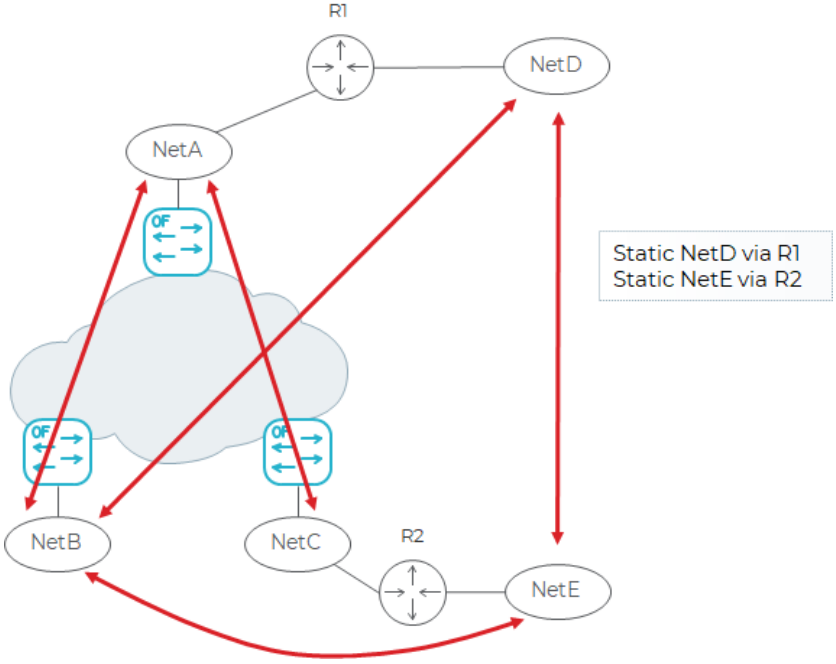


Рис. 19. Пример топологии со статической маршрутизацией

Подсети NetD и NetE с точки зрения разрешенного взаимодействия между подсетями наследуют характеристики типа сервиса Full-Mesh либо Hub-and-Spoke и, как следствие, роль подключения либо Hub либо Spoke.

Независимо от топологии сервиса может быть указан только один маршрут по умолчанию (default route). Для топологии Hub-and-Spoke маршрут по умолчанию может быть задан только на одном SI с ролью Hub.

При создании сервиса для обоих типов топологии можно выбрать логический тип интерфейса SI: M2M point (аналог interface vlan) или Switch/port point (аналог subinterface на физическом порту).

1.5.5. Сервис L3 IP Multicast source tree

Сервис L3 IP Multicast source tree состоит из следующих элементов:

1. Один или более сервисных интерфейсов, к которым подключены источники IP Multicast, вещающие в режиме статической подписки;
2. Один или более сервисных интерфейсов, к которым подключены абоненты, подписавшиеся на группы IP Multicast по протоколу IGMPv2/v3;
3. Список групп Multicast.

Данный сервис организует дерево распространения IP Multicast внутри домена. Корнями деревьев являются указанные при создании сервиса источники. Подписчики присоединяются к ближайшему узлу дерева, если не заданы ограничения (указание конкретного источника, запрет на подключение к источнику данному абоненту и т.д.).

Схема сервиса представлена на рисунке 20.

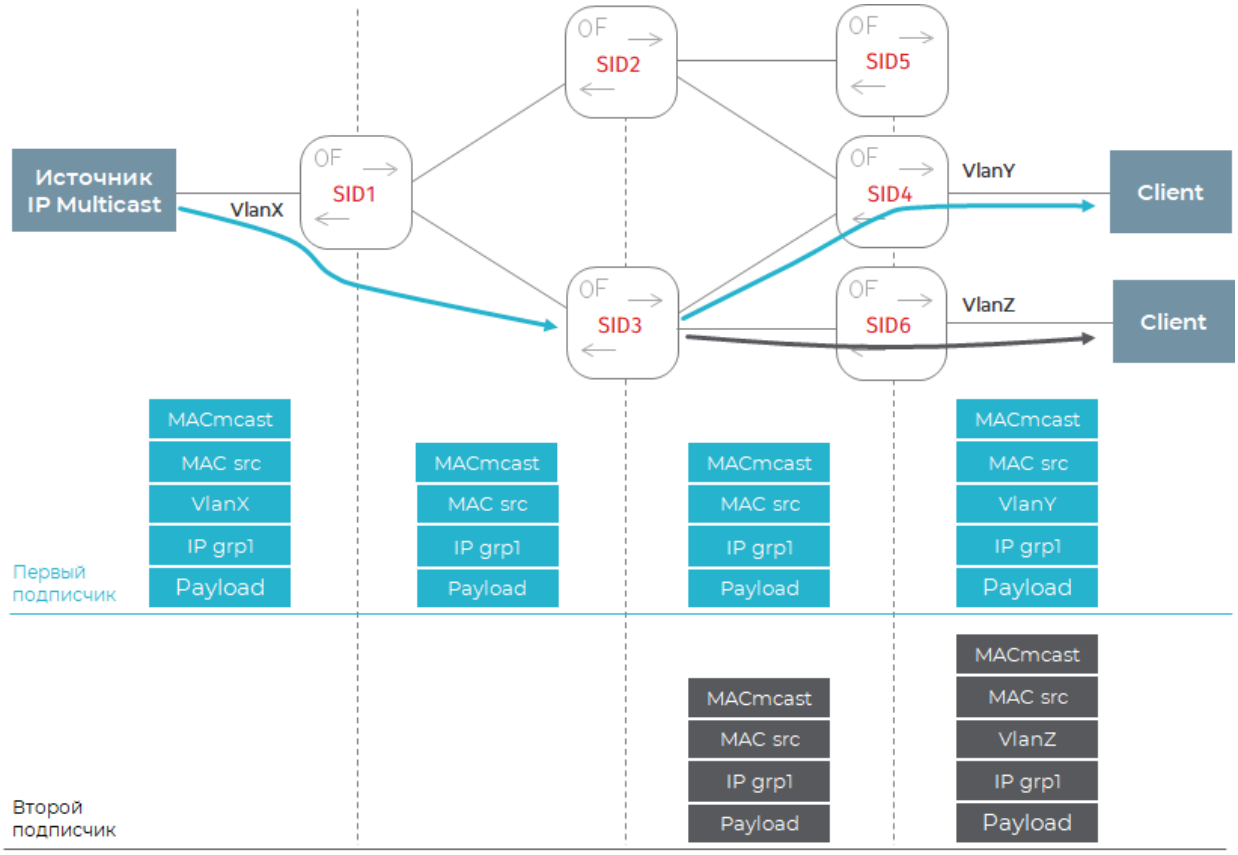


Рис. 20. Сервис L3 IP Multicast source tree

Репликация трафика в узлах дерева производится путем определения Openflow групп типа All. Трафик данного сервиса внутри домена идет как Ethernet-фрейм с IP-payload, без дополнительной инкапсуляции.

После создания сервиса для каждого источника Multicast Source SI строится единый граф транспортного дерева, по которому будут распространяться все мультикаст-группы. Расчет транспортного дерева производится при создании или изменении конфигурации сервиса, и динамически пересчитывается при изменении топологии. Транспортное дерево включает коммутаторы с основным и резервным источниками, коммутаторы с разрешенными потребителями и транзитные коммутаторы.

Поверх транспортного дерева при подписке потребителей на активные группы строятся частные деревья для каждой группы.

Трафику сервиса присваивается класс обслуживания RealTime.

Сервисные интерфейсы, к которым подключены абоненты, могут одновременно использоваться и при создании других сервисов (P2P, P2M, M2M, L3VPN).

1.5.6. Резервные сервисные интерфейсы Reserve SI

Для организации отказоустойчивого подключения клиентов к сервисным интерфейсам имеется возможность использовать резервные сервисные интерфейсы - Reserve SI. Добавление Reserve SI доступно для всех L2 и L3 сервисов и для всех сервисных интерфейсов.

После выхода из строя основного интерфейса трафик автоматически перенаправляется на резервный. Если основной интерфейс снова становится доступным, сервис может продолжить использовать резервный интерфейс или переключиться обратно на основной, в зависимости от настроек. Предусмотрена возможность принудительной смены сервисного интерфейса между основным и резервным.

1.6. Зеркалирование и отвод трафика (TAP)

Для сервисов point-to-point, multipoint-to-multipoint, point-to-multipoint можно включить функцию зеркалирования или отвода (перенаправления) трафика с определенных портов коммутаторов. Данная функция позволяет агрегировать, фильтровать и реплицировать трафик от различных участков сети с целью передачи его на централизованные системы мониторинга для обеспечения безопасности, наблюдения или аналитики.

Для того, чтобы выделить только интересующий трафик, предварительно создаются правила фильтрации, в которых могут использоваться значения следующих полей заголовка пакета:

- MAC источника,
- MAC получателя,
- Внешний VLAN ID,
- Внешний VLAN PCP,
- Внутренний VLAN ID,
- Внутренний VLAN PCP,
- IP адрес/маска источника,
- IP адрес/маска получателя,
- ICMP, TCP, UDP,
- TCP, UDP порт источника,
- TCP, UDP порт получателя.

Для одного входного сервисного интерфейса может быть задано несколько фильтров. И наоборот, один фильтр может применяться для разных SI.

1.7. Обеспечение отказоустойчивости сети

Архитектура решения и используемые механизмы резервирования обеспечивают работоспособность сети при возникновении следующих видов сбоев:

- отказ или перегрузка канала связи;
- отказ коммутатора;
- отказ узла кластера SDN-контроллеров.

Для того, чтобы была возможность использовать эти механизмы резервирования, необходимо при проектировании сети предусмотреть избыточные устройства и соединения между ними.

1.7.1. Отказ канала связи

Сразу после включения контроллеров и регистрации коммутаторов рассчитываются кратчайшие пути (shortest path) и резервные пути (fast-failover) от каждого узла к каждому. Далее контроллер программирует коммутаторы OpenFlow, заранее определяя основной выходной порт коммутации трафика к определенному узлу и резервный порт в случае отказа основного.

Таким образом, при отказе линка не производится переписывание правил коммутации пакетов. Коммутатор сразу начинает отправлять пакеты через резервный порт (рис. 21).

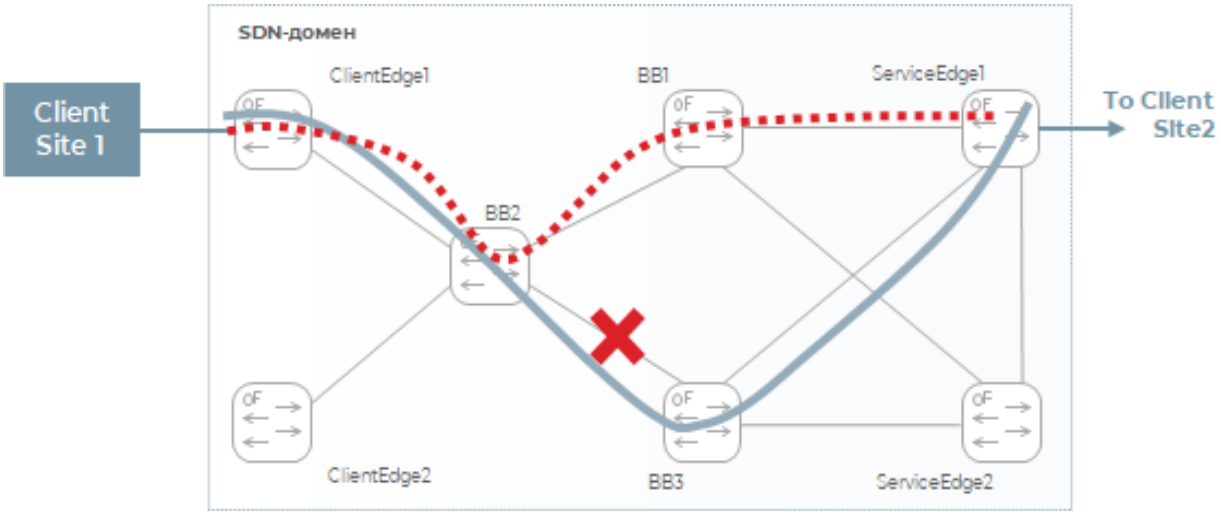


Рис. 21. Автоматическое перенаправление трафика через резервный порт

Если линк не восстановился, то по истечении таймаута (Fail throttling, 1-3 сек) контроллер пересчитывает топологию транспортных туннелей и прописывает новые транспортные правила для каждого сегмента, начиная с предпоследнего узла. После этого старые правила на промежуточных узлах удаляются, и трафик передается по новому маршруту.

Не всегда при аварии на канале связи интерфейс коммутатора переходит в состояние interface down. В общем случае каналообразующее оборудование должно деактивировать клиентский интерфейс при нарушении целостности канала.

Если каналообразующее оборудование не позволяет это сделать, необходимо использовать коммутаторы OpenFlow с поддержкой протокола BFD (Bidirectional Forwarding Detection) или CFM (Ethernet Connectivity Fault Management) для быстрого детектирования обрыва канала.

1.7.2. Переключение трафика на резервный маршрут при возрастании количества ошибок или при наличии перегрузки на каналах связи

Реализована возможность переключения трафика на резервный маршрут при возрастании количества ошибок или при наличии перегрузки по трафику на каналах передачи данных между устройствами.

Данная функция может быть включена при запуске контроллера с указанием требуемых параметров или использованием значений по умолчанию. Кроме того, для каждого линка возможно задавать индивидуальные значения для всех параметров (рис. 22).

Для детектирования событий достижения пороговых значений контроллер через заданные интервалы времени сравнивает пороговые значения с текущими значениями, полученными из модуля сбора статистики реального трафика. При обнаружении превышения любого из установленных пороговых значений контроллер изменяет стоимость линка и пересчитывает топологию. Таким образом, линк не выключается из топологии, а становится очень невыгодным по стоимости. Если это единственный линк, то он продолжает использоваться.

При обнаружении снижения всех параметров ниже установленных пороговых значений контроллер возвращает стоимость линка к обычному значению и пересчитывает топологию.

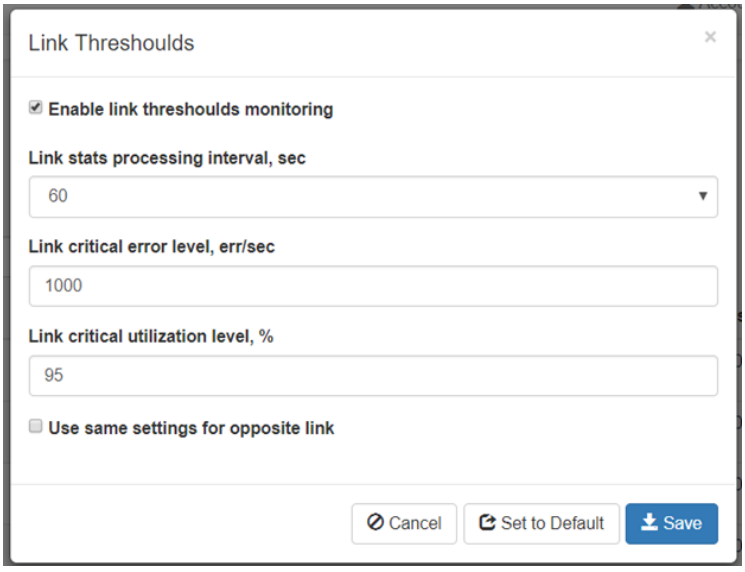


Рис. 22. Задание пороговых значений для линков

1.7.3. Link Dampening

Основная задача функции Link Dampening - детектирование частых изменений состояния интерфейса (UP-DOWN или наоборот), перемещение всех текущих сервисов, проходящих через интерфейс, на запасные более стабильные пути, и выведение транспортного сегмента, привязанного к этому интерфейсу, из расчета путей для сервисов.

Учитываются следующие переходы состояния порта:

- Up/Live → Down/Not-Live
- Down/Not-Live → Up/Live
- Up/Live → Up/Not-Live
- Up/Not-Live → Up/Live

Состояния Live/Not-Live используются для интеграции функции Link Dampening с функционалом Ethernet Connectivity Fault Management (CFM), который детектирует пропадание двусторонней связности Ethernet сегмента между соседними коммутаторами без перехода интерфейса в состояние down (пропадание сигнала Rx).

Link Dampening применяется на обоих концах Ethernet сегмента.

Нестабильное состояние порта в сети SDN не имеет такого катастрофического эффекта, как в традиционных сетях, где каждое изменение состояния интерфейса требует, чтобы маршрутизаторы в сети пересчитали лучшие пути, добавили или удалили маршруты в маршрутные таблицы и далее сообщили об изменениях своим соседям. В результате увеличивается загрузка процессора на устройствах и рассинхронизируются протоколы маршрутизации. Тем не менее в сети SDN при переходе интерфейса коммутатора OpenFlow из состояния Up в Down и из Down в Up отрабатывается функционал FastFailover группы, для которой этот интерфейс является основным путем. Также каждый переход состояния интерфейса приводит к перерасчету проложенных сервисов через этот интерфейс, а значит, увеличению количества openflow сообщений между контроллером и коммутаторами.

Администратор может активировать функцию Link Dampening глобально для кластера контроллеров и изменить параметры отдельно для каждого линка, если значения по умолчанию не подходят (рис. 23).

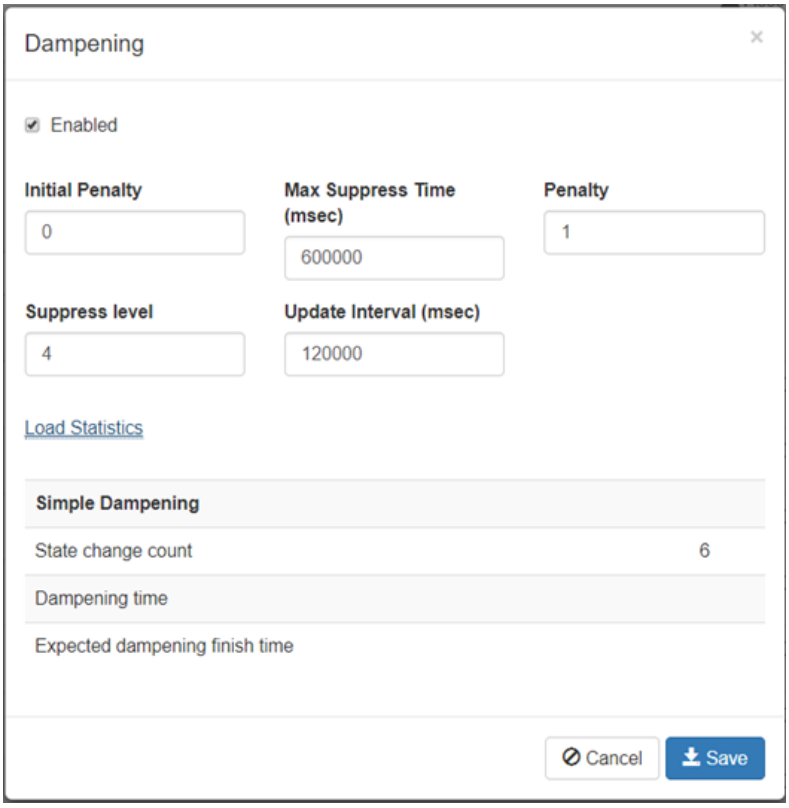


Рис. 23. Настройка Link Dampening для отдельного линка

Значения полей окна “Dampening”:

- Max Suppress Time — Максимальное время, на которое может быть заблокирован интерфейс. По истечении этого времени счетчики сбрасываются;
- Suppress level — Значение Penalty, при котором активируется ограничение доступа к интерфейсу;
- Update Interval - Интервал времени, за которое интерфейс должен набрать Penalty до Suppress level.

По умолчанию, возврат интерфейса и связанного с ним транспортного сегмента в работу происходит, если в течении 10 минут интерфейс не испытывает изменения состояния.

1.7.4. Отказ коммутатора

При отказе коммутатора он автоматически выводится из топологии. Производится перерасчет сегментов и прописывание новых правил на оставшиеся коммутаторы.

Используется следующий алгоритм обнаружения отказа устройства. Если контроллер не получает сообщения OpenFlow от коммутатора в течение определенного таймаута (по умолчанию 1 сек), контроллер начинает посылать Echo request на коммутатор с периодичностью 1 раз в секунду.

Если получен ответ от коммутатора Echo reply на посылаемый Echo request контроллера, предполагается, что коммутатор исправный, и работа продолжается в штатном режиме.

Если контроллер не получил ответа в течение заданного интервала времени (по умолчанию 5 сек), проверяется, получал ли контроллер сообщения об изменении статуса портов с коммутаторов, непосредственно подключенных к проверяемому, и относятся ли полученные сообщения к линкам, направленным в сторону проверяемого коммутатора. Если таких сообщений не было, считается, что коммутатор потерял контрольный канал с контроллером, но продолжает работу. Если были сообщения об изменении статуса порта на неактивный, считается, что коммутатор вышел из строя. В этом случае он удаляется из топологии.

1.7.5. Отказ контроллера

Для обеспечения отказоустойчивости контроллера рекомендуется устанавливать несколько серверов контроллеров (лучше на разных площадках) и объединять их в кластер. Количество серверов в кластере должно быть нечетным (2N+1). Обычно используют кластеры из трех серверов (рис. 24).

Один из серверов в кластере автоматически выбирается мастером.

Алгоритм выбора мастера оптимизирован для выполнения следующих условий:

- Предоставление услуг без потери трафика (или минимизация потери трафика) в случаях отказа мастера и перехода роли мастера на сервер slave, а также отказа всего кластера контроллеров с последующим его восстановлением;
- Исключение ситуации, когда при потере связи между контроллерами сразу несколько контроллеров переходят в роль мастера (split brain).

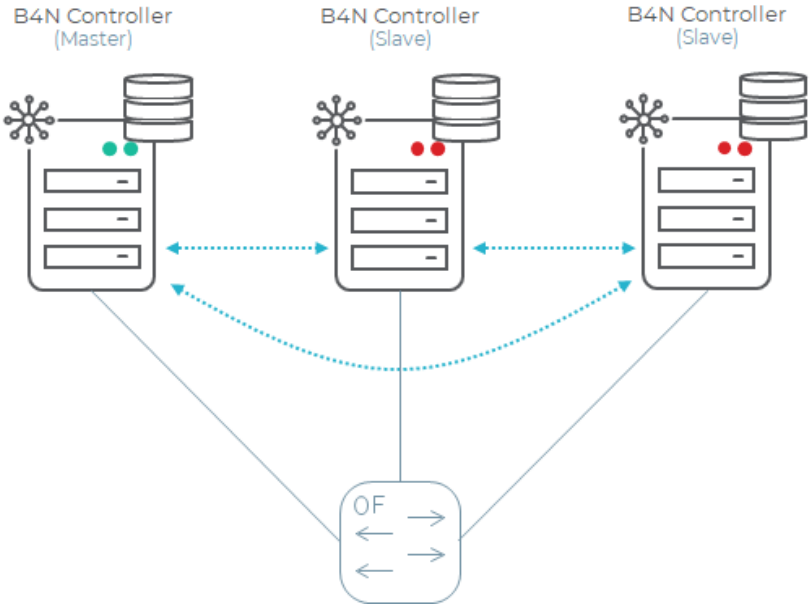


Рис. 24. Кластер контроллеров OpenFlow

Для предотвращения split brain используется концепция кворума.

Кворум определяет минимальное количество узлов, которые должны остаться работоспособными, чтобы кластер считался функционирующим. Когда один или несколько узлов выходят из строя, необходимо удостовериться, что на работающих узлах данные и процессы остались актуальными. Для этого сравниваются копии данных на оставшихся узлах и, если обнаруживается несоответствие, применяется правило большинства - информация считается верной, если она совпадает на большинстве узлов.

Для реализации такого алгоритма кластер должен иметь нечетное количество узлов, а кворум определяется как число, большее чем половина узлов. Если в результате отказа нескольких узлов размер кластера станет меньше величины кворума, кластер в целом будет неработоспособным.

Например, для кластера из семи узлов размер кворума будет равен 4. Если одновременно выйдут из строя 4 узла, кластер не будет функционировать, тогда как если только 3 узла станут неработоспособными, кластер продолжит работу. Концепция кворума особенно важна в случаях, когда из-за потери связи между узлами кластер разделяется на 2 или более частей. Если кластер разделится на две части, состоящие из 3-х и 4-х узлов, кворум из 4-х узлов продолжит работать, а та часть, где осталось 3 узла, будет выведена из кластера.

Для кластера из двух узлов ни одна из частей не будет иметь большинство, что может привести к ситуации, когда обе части прекратят работу или будут работать независимо друг от друга. Кластеры с четным количеством узлов будут иметь ту же проблему, если произойдет разделение на две равные части.

При выходе из строя мастера и наличии кворума выбирается новый контроллер с ролью мастера.

На коммутаторе OpenFlow может быть настроено подключение к одному или нескольким контроллерам. В случае использования кластера контроллеров указываются все серверы кластера. Коммутатор подключается ко всем контроллерам, присутствующим в его конфигурации, и постоянно поддерживает соединение с ними.

Контроллер в роли slave имеет ограниченные права на управление коммутатором. Фактически, он может только получать информацию о его статусе. Контроллер не может посылать команды коммутатору, связанные с обработкой пакетов или изменением состояния коммутатора.

Контроллер, который стал мастером, сообщает коммутатору OpenFlow об изменении своей роли, и после этого имеет полный доступ к коммутатору. Однако коммутатор должен предварительно проверить, что только один из контроллеров, к которым он подключен, имеет роль master.

В случае обрыва соединения между контроллером и коммутатором обе стороны не прекращают попытки установить новое соединение до тех пор, пока связь не будет восстановлена или IP адрес устройства не будет удален из конфигурации.

Коммутатор OpenFlow, который потерял контакт со всеми контроллерами, продолжает принимать и передавать пакеты в соответствии с прописанными на нем правилами. При этом правила можно разделить на постоянные и динамические.

Постоянные правила используются в статических сервисах, например, L2 point-to-point. Постоянные правила остаются в памяти коммутаторов до восстановления связи с контроллером.

Динамические правила создаются контроллером как результат какого-либо события, например, получения коммутатором пакета от неизвестного MAC-адреса для сервисов, использующих MAC learning. Динамические правила имеют время жизни, которое может быть абсолютным значением или относительным, когда таймер сбрасывается после каждого получения пакета, удовлетворяющего правилу. По истечении времени жизни правило удаляется из таблицы потоков (flow table) коммутатора.

При отсутствии связи с контроллером невозможно установление новых сервисов.

ОРКЕСТРАЦИЯ ВИРТУАЛЬНЫХ СЕТЕВЫХ ФУНКЦИЙ И СЕРВИСОВ

1.8. Архитектура NFV

B4N Service Platform соответствует референсной архитектуре ETSI NFV MANO. Функции NFV Orchestrator реализуются оркестратором B4N, VNF Manager отвечает за управление виртуальными сетевыми функциями (созданными другими производителями или с открытым кодом), в качестве VIM в настоящее время поддерживается OpenStack. На рис. 25 показана взаимосвязь между логическими компонентами системы с точки зрения управления виртуализацией сетевых функций. Цветом выделены компоненты, входящие в состав платформы B4N Service Platform.

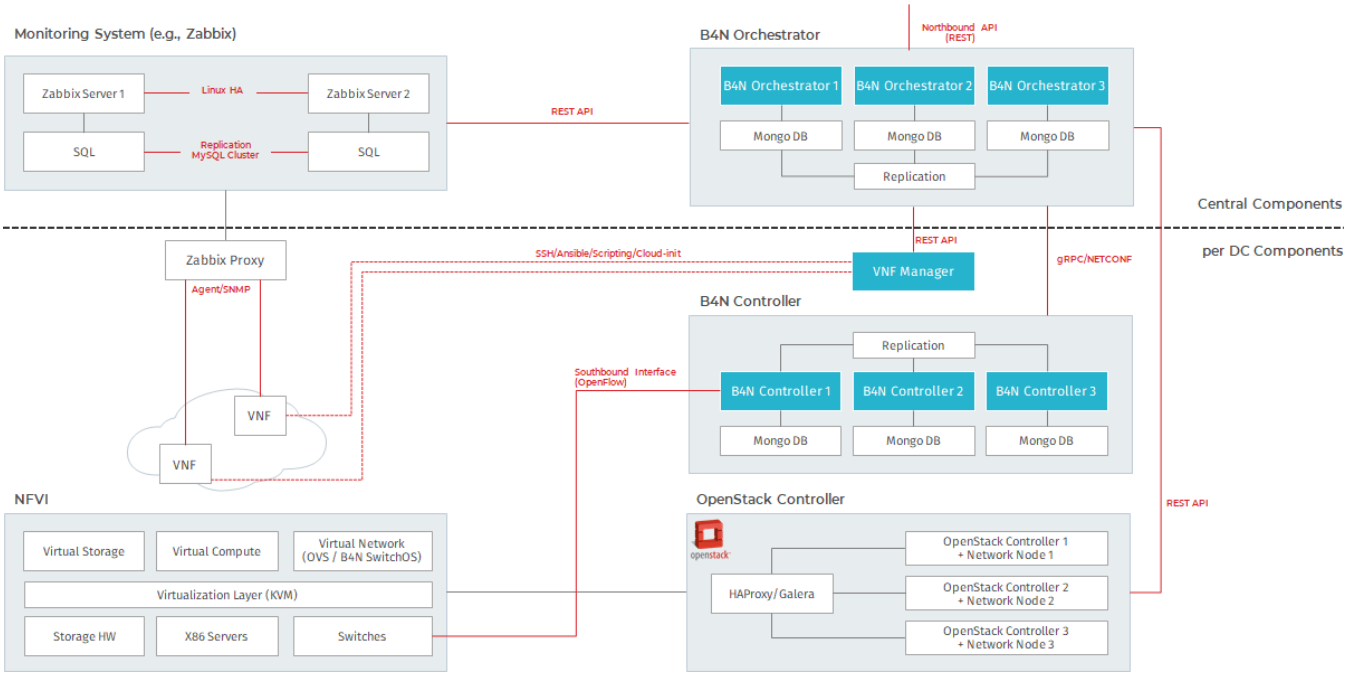


Рис.25. Логическая структура платформы B4N Service Platform

1.9. Управление и оркестрация VNF

B4N Orchestrator позволяет загружать в каталог виртуальные сетевые функции, созданные другими производителями. Возможность быстрого портирования виртуальных функций с применением стандартных VNF-дескрипторов позволяет сервис-провайдерам предлагать своим клиентам широкий набор виртуальных сетевых функций, коммерческих или с открытым кодом.

Оркестрация VNF включает следующие действия:

- Выбор места размещения высокопроизводительных и высоконагруженных VNF,
- Управление жизненным циклом VNF (создание, обновление, удаление),
- Мониторинг VNF,
- Обеспечение доступности - автоматический перезапуск VNF в случае необходимости на другом хосте или в другом ЦОДе,
- Обеспечение масштабируемости - добавление ресурсов в случае их нехватки или освобождение ресурсов, если загрузка VNF низкая.

VNF Manager отвечает за начальную настройку VNF после развертывания. Так как в настоящее время не определен единый стандартный язык сценариев для настройки виртуальных функций, VNF Manager поддерживает различные методы загрузки начальной конфигурации: SSH, Ansible, скрипты (python, bash), Cloud-init.

В каждом ЦОД разворачивается отдельный экземпляр VNF Manager. Оркестратор осуществляет централизованное управление распределенной архитектурой VNF Manager.

1.9.1. Состав пакета VNF

VNF загружаются в систему в виде пакета VNF (VNF Package).

Пакет VNF представляет собой tar-архив, содержащий информацию, необходимую для развертывания и управления жизненным циклом VNF (рис. 26).

Пакет VNF включает:

- образы виртуальных машин, входящих в состав виртуальной сетевой функции (обычно в формате qcow2);
- VNF дескриптор (в формате XML);
- папку со скриптами (scripts), которые могут потребоваться для начального развертывания VNF и управления ее жизненным циклом. В этой же папке могут находиться другие необходимые файлы, например, файлы лицензий;
- опционально - файл с описанием виртуальной сетевой функции (в формате pdf).

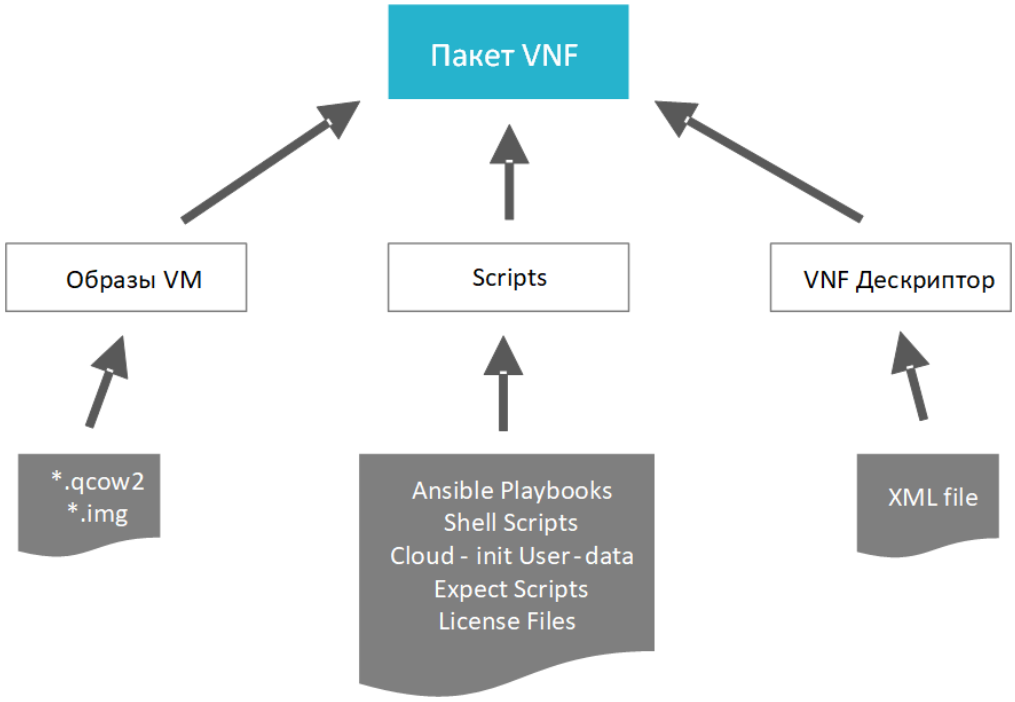


Рис. 26. Состав пакета VNF

В репозитории могут храниться несколько версий пакетов VNF одной виртуальной функции. Каждый пакет имеет собственный VNF дескриптор. Разные версии VNF пакетов могут соответствовать различным вариантам имплементации одной и той же функции, разным версиям ПО или предназначаться для работы под управлением разных гипервизоров.

1.9.2. VNF дескриптор

Вся необходимая для развертывания VNF информация содержится в файле VNF дескриптор. VNF дескриптор описывает список образов, количество виртуальных узлов (VDU) в составе VNF, требования к ресурсам (процессор, оперативная память, количество и объем дисков и т.п.), требования к сети и настройкам сетевых интерфейсов.

Структура файла VNF дескриптор представлена на рис. 27. Файл содержит следующие основные разделы:

- Общая информация о VNF - производитель VNF, версия, ссылка на файл с описанием VNF и т.п.;
- Virtual Links - описание типа сетевой связности между двумя или более точками подключения, в соответствии с требованиями производителя VNF;
- Connection Points - точки подключения - описание портов виртуальной сетевой функции, в том числе используемых для управления;
- Configurations - информация, необходимая для инициализации и начальной настройки VNF;
- Images - содержит имена и формат образов виртуальных машин для развертывания VNF, а также требования к гипервизору;
- Flavours - описание вариантов развертывания экземпляров VNF.

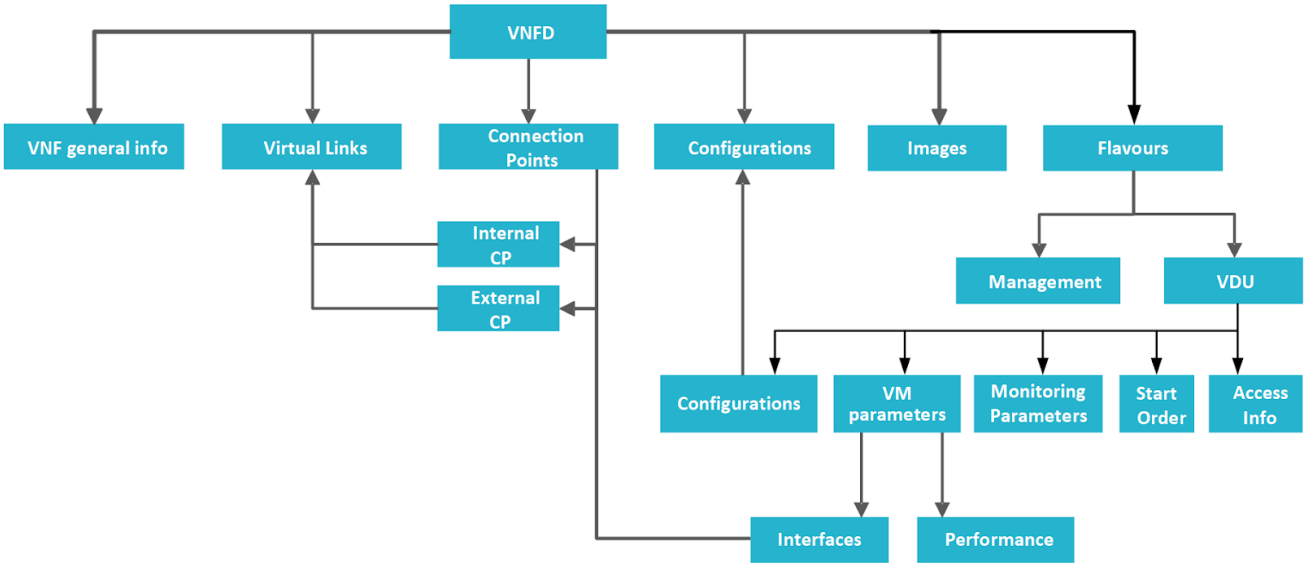


Рис. 27. Структура файла VNF дескриптор

Для каждого варианта развертывания экземпляра VNF (flavour) указывается:

- имя, описание и порядок расположения в меню выбора flavours в графическом интерфейсе;
- способы доступа к интерфейсам управления VNF (ssh, VNC, Web-интерфейс);
- информация о виртуальных узлах (VDUs - Virtual Deployment Units), входящих в состав виртуальной сетевой функции.

Описание каждого VDU содержит:

- имя, способы мониторинга и название шаблона для мониторинга параметров VDU;
- информацию для развертывания - название имиджа, объем виртуального дискового хранилища;
- описание способов начальной настройки, учетную запись администратора по умолчанию;
- требования к аппаратным ресурсам - процессору, оперативной памяти, сетевым интерфейсам;
- описание политики автоматического восстановления VDU.

Кроме того, VNF дескриптор содержит описание политики ручного и автоматического масштабирования VNF.

Просмотреть файл VNF дескриптор для виртуальной сетевой функции можно, выбрав в каталоге требуемую VNF и нажав кнопку “Открыть VNFD” в верхней правой части экрана.

1.9.3. Способы оптимизации производительности VNF

Если в ЦОД в качестве VIM установлен OpenStack, для оптимизации производительности VNF используются возможности Intel EPA (Enhanced Platform Awareness). OpenStack определяет характеристики аппаратных платформ, в том числе относящиеся к реализации расширенной функциональности CPU. В процессе развертывания VNF запрошенные характеристики

сравниваются с возможностями оборудования и в результате для размещения VNF выделяется оптимальное местоположение. Например, для высоконагруженных VNF, таких как балансировщики, программные коммутаторы и т.п., выделяются ресурсы с поддержкой оптимизации под архитектуру NUMA, возможностью использования больших страниц памяти (hugepage), “привязки” ядер процессора к приложению (CPU Pinning), прямого управления сетевыми картами (PCI Passthrough или Single Root I/O Virtualization - SR-IOV).

Требуемые параметры EPA указываются в файле VNF Descriptor (Flavour -> VDUs). Можно выделить следующие группы параметров:

- Политика использования ресурсов процессора,
- Параметры, относящиеся к NUMA-топологии,
- Использование больших страниц памяти.

1.9.3.1. Политика использования ресурсов процессора

Доступны следующие настройки:

- CPU Pinning - требуется ли выделить виртуальной машине персональные ядра процессора (dedicated), или она может работать в режиме разделения ядер с другими VM (shared).
- Simultaneous multithreading (SMT) - использовать ли логическое разделение ядер (в терминах Intel - Hyper-threading).

1.9.3.2. NUMA-топология

NUMA (Non-Uniform Memory Access — «неравномерный доступ к памяти» или Non-Uniform Memory Architecture — «Архитектура с неравномерной памятью») — схема реализации компьютерной памяти, используемая в мультипроцессорных системах, когда время доступа к памяти определяется ее расположением по отношению к процессору.

Оптимально, когда виртуальная машина работает в рамках одного узла NUMA, не используя шину QPI.

На обычных, не-NUMA системах, нагрузка распределяется между всеми сокетами в режиме "round-robin". Такой подход ведет к увеличению производительности. Но на NUMA системе размещение VM целиком на одном NUMA узле, включая виртуальные процессоры и память, позволяет снизить задержки при обращении к памяти примерно на 30% (рис. 28).

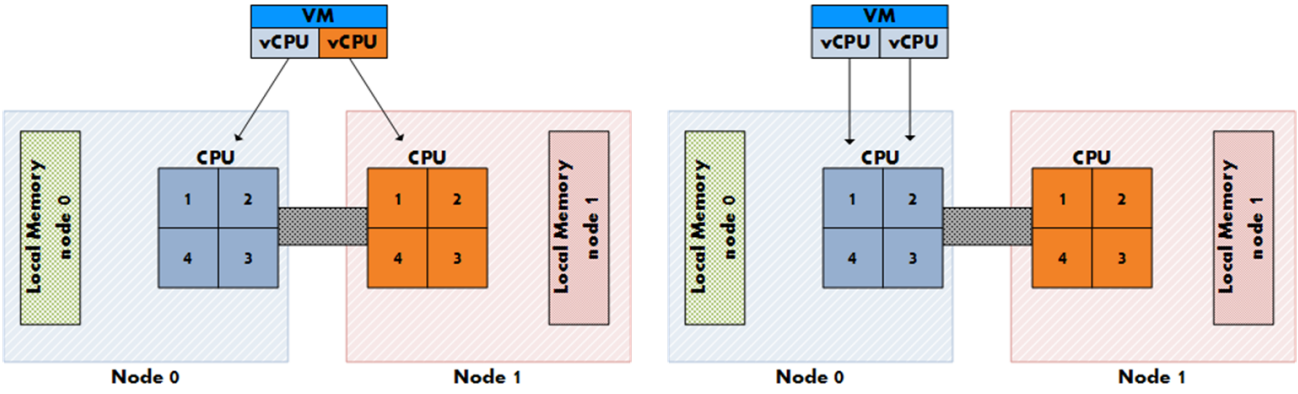


Рис. 28 Размещение vCPU на не-NUMA системе (слева) и на NUMA системе (справа)

В файле VNF Descriptor можно указать следующие параметры:

- Количество узлов NUMA, которыми следует ограничить виртуальную машину. Бывает, что VM не может поместиться в рамках одного узла.
- Номер NUMA и номера ядер процессора, выделяемых виртуальной машине.
- Количество оперативной памяти в мегабайтах, выделяемое с определенного узла NUMA.

1.9.3.3. Использование больших страниц памяти

Большие страницы (Hugepages) - это технология использования страниц памяти размером 2 Мбайт или 1 Гбайт вместо стандартных 4 Кбайт.

В некоторых случаях использование больших страниц памяти может увеличить производительность VNF за счет снижения частоты обращения к страницам памяти и связанных с этим потерь процессорного времени.

1.9.3.4. Управление сетевыми устройствами

Технологии PCI Passthrough и SR-IOV позволяют полностью или частично передать виртуальной машине управление PCI устройством ввода/вывода. Это увеличивает эффективность сетевых операций и снижает накладные расходы, связанные с обработкой сетевого трафика.

Для задания режима работы применяется параметр тип порта - vnic-type. Возможные значения:

- virtio (используется по умолчанию)
- direct - SR-IOV или PCI-Passthrough (зависит от того, какая технология выбрана в ЦОД).

1.9.4. Начальная конфигурация VNF

Начальная конфигурация VNF может быть автоматизирована с помощью скриптов, вызываемых из файла VNF Descriptor. Предварительно должен быть настроен доступ на виртуальную машину по протоколу ssh и задан пароль пользователя с правами root.

Если VNF не поддерживает управление по протоколу SSH, альтернативным вариантом может быть использование пакета cloud-init.

Cloud-init позволяет выполнить конфигурацию виртуальных машин при начальной загрузке, при этом VM может не иметь подключения к сети.

Cloud-init представляет собой набор скриптов и утилит, предназначенных для инициализации и предварительной настройки виртуальной машины или сетевой функции. Пакет cloud-init должен быть установлен на виртуальную машину.

С помощью cloud-init можно, например, выполнить следующие настройки при первой загрузке:

- Выбрать локализацию операционной системы,
- Назначить имя хоста,
- Задать пароль пользователя root,
- Сгенерировать ключи SSH и добавить их в файл .ssh/authorized_keys,
- Задать конфигурацию сетевых устройств,
- Выполнить другие команды.

Если VNF предъявляет определенные требования к сложности паролей пользователей, необходимо указать в VNF Descriptor схему создания пароля. Для этого используется регулярное выражение (regex expression).

С помощью регулярного выражения можно указать, что пароль должен содержать:

- Прописные буквы английского алфавита;
- Строчные буквы английского алфавита;
- Десятичные цифры;
- Специальные символы (например, @, #, $, %);
- Не менее требуемого количества символов.

Регулярное выражение указывается в разделе <vdu> файла VNF Descriptor.

1.9.5. Политики размещения VDU (Affinity/Anti-Affinity)

Часто бывает необходимо обеспечить контроль размещения виртуальных машин на аппаратных ресурсах. Например, VDU, требующие минимальных задержек при обмене информацией, должны быть развернуты максимально близко друг к другу, в идеальном случае на одном хосте. VDU, для которых может потребоваться вертикальное масштабирование или необходимо обеспечить отказоустойчивость, должны размещаться на разных аппаратных серверах.

Политики размещения VDU Affinity/Anti-Affinity позволяют указать, следует ли создавать виртуальные машины, входящие в VNF, на одном хосте или разных хостах:

- Affinity - все VDU в составе VNF размещаются на одном хосте (гипервизоре);
- Anti-Affinity - VDU размещаются на разных хостах.

Политики Affinity/Anti-Affinity реализуются с помощью групп. Если для группы назначается политика Affinity, все виртуальные машины - члены группы - развертываются на одном хосте. Если назначается политика Anti-Affinity, VDU размещаются на разных хостах.

Принадлежность группе не может быть назначена или изменена для запущенных виртуальных машин, но может быть указана при создании виртуальной машины из образа. Политики Affinity/Anti-Affinity описываются в файле VNF Descriptor в разделе <flavor>.

1.9.6. Мониторинг VNF

Функции обеспечения доступности (auto healing) и автоматического добавления/освобождения ресурсов (auto scaling) требуют постоянного мониторинга виртуальных сетевых функций. Для этой цели реализована интеграция с системой мониторинга Zabbix, которая включается в состав решения.

На VNF (на каждый VDU) устанавливаются агенты Zabbix. Если на VNF нельзя поставить агент, используется мониторинг по протоколу SMNP.

Система мониторинга включает в себя сервер Zabbix и серверы Zabbix Proxy, устанавливаемые в каждом ЦОД с целью масштабируемости решения. Для мониторинга используется сеть управления VNF, поэтому сервер Zabbix Proxy должен быть добавлен в соответствующий

служебный сетевой сервис P2M (рис. 29).

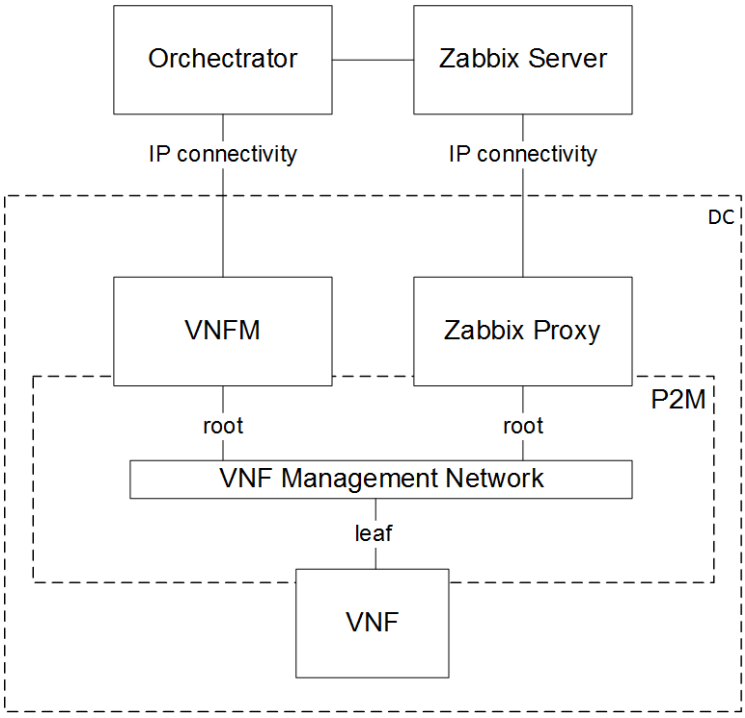


Рис. 29. Мониторинг VNF с помощью Zabbix

При создании VNF хост добавляется в соответствующую группу на сервере Zabbix и для него выбирается шаблон, определяющий, какие параметры должны мониториться. Выбор шаблона зависит от операционной системы VDU и от виртуальной сетевой функции. Если для VNF имеется специфический шаблон Zabbix, то его имя может быть указано в VNF дескрипторе.

Таким образом, получив всю необходимую информацию (идентификаторы сервера Zabbix Proxy, группы и шаблона), оркестратор настраивает мониторинг виртуальной функции.

Образ VDU уже содержит предустановленный Zabbix-агент. Его настройка может быть выполнена после получения доступа к VDU с помощью скрипта. Например, соответствующие команды могут быть добавлены в скрипт начальной конфигурации VNF.

Администратор или оператор тенанта из графического интерфейса могут просматривать графики изменения параметров, для которых включен мониторинг, а также изменять настройки отображения:

- Выбрать нужный VDU (если VNF состоит из нескольких серверов),
- Выбрать график из списка доступных,
- Выбрать интервал (предустановленные интервалы – Real-Time (1 час, обновление раз в 20 секунд), 1 день, 1 неделя, 1 месяц) или задать требуемый интервал с указанием даты начала и конца периода.

1.9.7. Масштабирование VNF

Масштабирование VNF выполняется с целью увеличить или уменьшить выделенные данному экземпляру виртуальной функции ресурсы (ядра процессора, объем оперативной памяти и дисков).

Необходимость в добавлении ресурсов (переходе на более производительный шаблон flavor) возникает, если загрузка виртуальной машины становится постоянно высокой. Если же, наоборот, экземпляр виртуальной функции недозагружен, имеет смысл использовать менее производительный flavor, чтобы освободившиеся ресурсы могли быть использованы другими виртуальными функциями или приложениями.

С помощью API система мониторинга Zabbix уведомляет оркестратор о необходимости масштабирования конкретной виртуальной функции.

Уведомления посылаются по следующим событиям:

- Высокая загрузка процессора (high_cpu_utilization),
- Низкая загрузка процессора (low_cpu_utilization),
- Высокое потребление оперативной памяти (high_memory_usage),
- Высокая загрузка сетевого интерфейса (high_net_utilization).

В VNF Descriptor добавляются параметры, описывающие, при каких условиях необходимо добавить или уменьшить ресурсы для виртуальной машины (VDU). Если хотя бы одно из условий выполняется, принимается решение провести вертикальное масштабирование VNF. Переход на другой flavor требует перезагрузки виртуальной машины.

1.9.8. Автоматическое восстановление VNF

Функция автоматического восстановления (Auto-Healing) позволяет автоматически восстановить работоспособность VDU в составе виртуальной сетевой функции, если виртуальная машина перестала отвечать на запросы системы мониторинга (например, зависла или была удалена).

В данной версии автоматическое восстановление возможно в рамках одного VIM.

В VNF дескрипторе в разделе <flavor> - <auto-healing> указываются действия (Actions) и наборы действий (Action-sets), необходимые для автоматического восстановления работоспособности виртуальной функции.

Возможно указать следующие действия:

- выполнить скрипт;
- перезагрузить виртуальную машину;
- пересоздать виртуальную машину.

Пересоздание виртуальной машины (reprovision) предполагает ее удаление, если она все еще существует, и создание нового экземпляра VDU на этом же или другом хосте. Далее VDU

включается в сервисную цепочку, на ней исполняются требуемые скрипты и восстанавливается последняя сохраненная конфигурация из резервной копии.

Действия в Action-set выполняются последовательно, до достижения положительного результата.

1.9.9. Операции с VNF и VDU

С VNF и VDU возможно производить действия как с экземплярами виртуальных машин под управлением OpenStack.

С активным экземпляром VNF можно выполнить следующие действия (рис. 30):

- **Start VNF** - запустить VNF. Действие выполняется для всех VDU, входящих в состав VNF и имеющих статус SHUTOFF. Для того чтобы действие было доступно, необходимо, чтобы хотя бы одна VDU была в указанном статусе.
- **Stop VNF** - Выключить VNF. Действие выполняется для всех VDU, входящих в состав VNF, имеющих статус ACTIVE или ERROR. Для того чтобы действие было доступно, необходимо, чтобы хотя бы одна VDU была в указанном статусе.

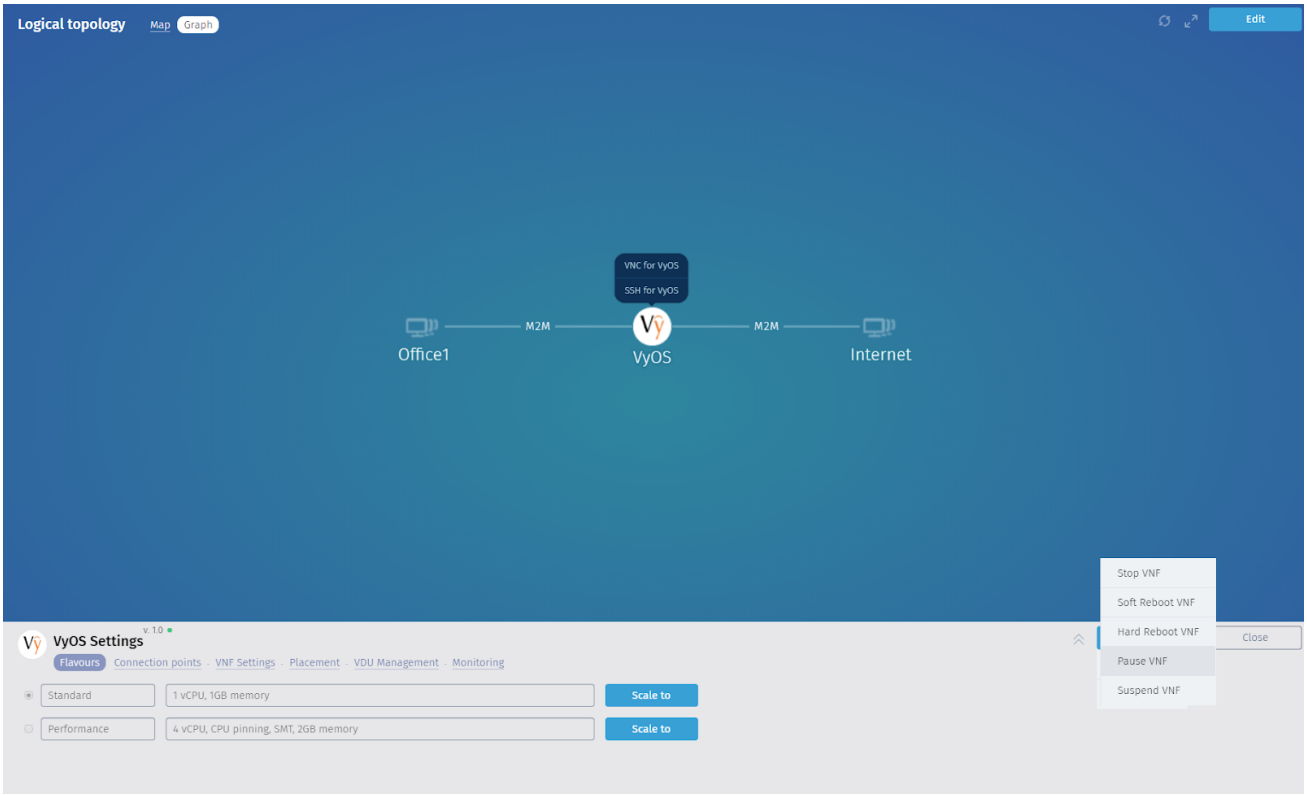


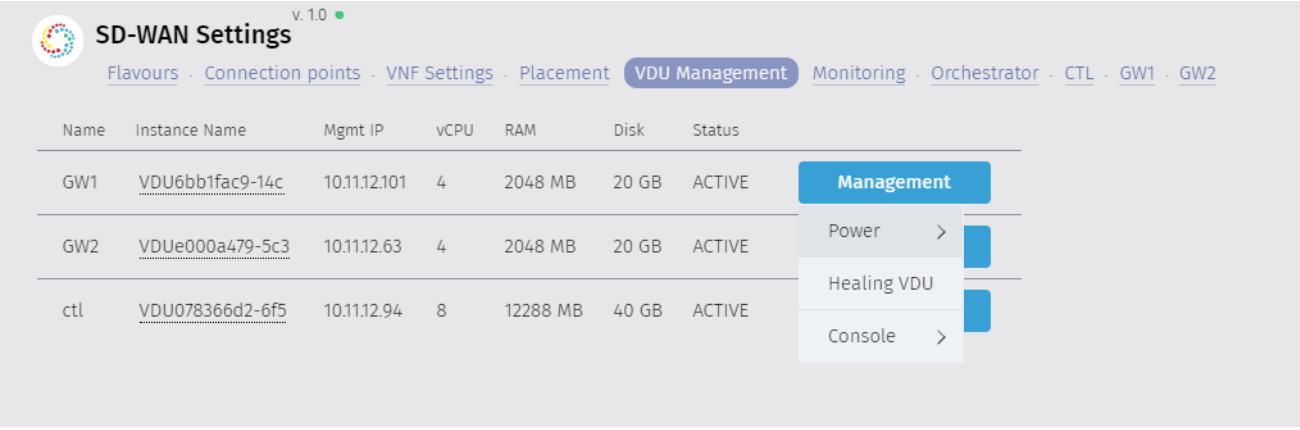
Рис. 30. Операции с VNF

- **Soft Reboot VNF** - Перезагрузить VNF. Перезагрузка с опцией Soft выполняется для всех VDU, входящих в состав VNF, имеющих статус ACTIVE. Для того чтобы действие было доступно, необходимо, чтобы хотя бы одна VDU была в указанном статусе. При

перезагрузке операционная система получает сигнал о необходимости перезапуска, что позволяет корректно завершить все процессы.

- **Hard Reboot VNF** - Перезагрузить VNF. Перезагрузка с опцией Hard выполняется для всех VDU, входящих в состав VNF. Перезагрузка имитирует выключение/включение питания для виртуальной машины.
- **Pause VNF** - Остановить VNF. Действие выполняется для всех VDU, входящих в состав VNF, имеющих статус ACTIVE. При выполнении команды, виртуальная машина "замораживается", состояние машины хранится в оперативной памяти.
- **Unpause VNF** - Снова запустить VNF. Действие выполняется для всех VDU, входящих в состав VNF, имеющих статус Paused. При выполнении команды виртуальная машина продолжает работу с учетом ранее сохраненного состояния.
- **Suspend VNF** - Остановить VNF. Действие выполняется для всех VDU, входящих в состав VNF, имеющих статус ACTIVE. При выполнении команды виртуальная машина "замораживается", состояние машины сохраняется на диск.
- **Resume Suspended VNF** - Снова запустить VNF. Действие выполняется для всех VDU, входящих в состав VNF, имеющих статус Suspended. При выполнении команды виртуальная машина продолжает работу с учетом ранее сохраненного состояния.
- **Healing VNF** - принудительная процедура восстановления для всех VDU в VNF. Ручное восстановление VNF возможно при любых состояниях VDU.

Аналогичные действия можно выполнять для каждого VDU в отдельности. Просмотр состояния VDU и меню управления доступны на закладке “VDU Management” (рис. 31).



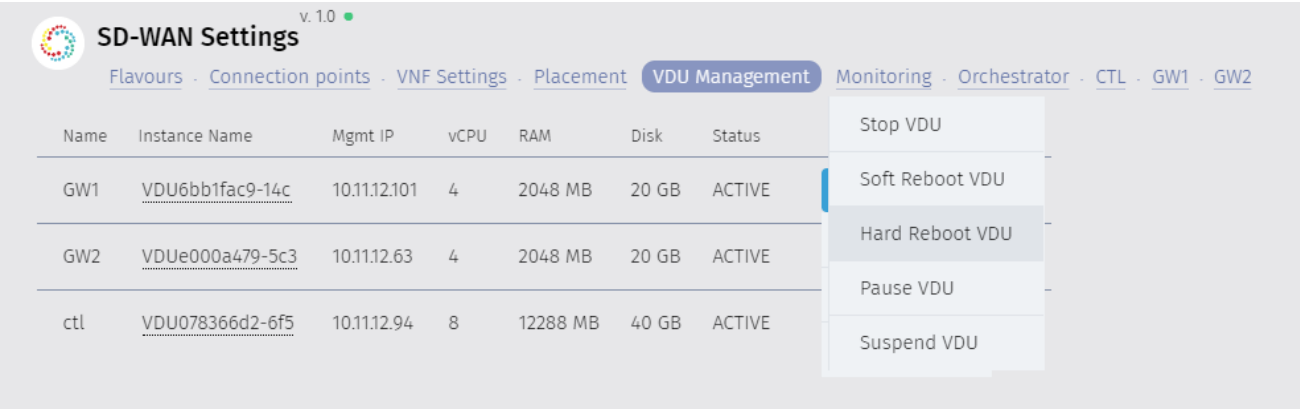


Рис. 31. Операции с VDU

1.9.10. Доступ к консоли VNF

Администратор тенанта может получить доступ к управлению активной VNF. Для этого необходимо кликнуть мышью иконку VNF в сервисной цепочке и выбрать один из предложенных способов:

- VNC консоль к виртуальной машине
- SSH сессия, отображающаяся в браузере
- Web-интерфейс VNF (при его наличии)

Доступ к интерфейсам управления VNF проксируется через VNF Manager и не требует IP-связности между рабочей станцией администратора и виртуальной сетевой функцией.

Доступные опции управления VNF описываются в VNF дескрипторе в разделе <Flavor> отдельно для каждого VDU.

1.10. Физические сетевые функции

1.10.1. Определение и характеристики PNF

Физическая сетевая функция (Physical Network Functions, PNF) - это сетевая функция, выполненная производителем в виде программного обеспечения, установленного на выделенное для этой функции оборудование, в отличие от VNF, которые используют облачные ресурсы под управлением VIM для развертывания виртуальных программных модулей.

PNF может представлять собой либо отдельное устройство, либо отдельно установленные виртуальные машины, не развернутые оркестратором. Для оркестрации PNF не используется VIM.

Производитель PNF может предоставлять информацию о внутренней архитектуре и способах доступа к внутренним компонентам функции, или устройство может поставляться как «черный

ящик», без раскрытия внутренних характеристик и логики работы. Программная и аппаратная части PNF могут производиться одним или разными вендорами.

Несмотря на то, что виртуализации сетевых функций является одним из основных трендов в сетевой архитектуре, в обозримом будущем останутся сетевые функции в виде физических устройств. Должна быть обеспечена возможность интеграции PNF с облачной инфраструктурой, включая взаимодействие с VNF в рамках сервисной цепочки. PNF должна соответствовать требованиям отказоустойчивости, масштабируемости, централизованного управления и безопасности.

1.10.2. Требования к PNF с точки зрения автоматизации управления

Возможность централизованного управления через оркестратор требует от PNF наличия открытого API. Должна предоставляться возможность мониторинга программной и аппаратной части. Отказоустойчивость PNF должна обеспечиваться применением резервного оборудования, позволяющего заменить вышедшее из строя устройство в режиме Active-Active или Active-Standby.

Настройка PNF не должна требовать ручного ввода информации или команд. В идеальном случае PNF должна использовать стандартные средства для развертывания, настройки и обновления в полностью автоматическом режиме.

Оркестрация PNF подразумевает возможность выполнять следующие действия:

- Получение оповещений о событиях и предупреждений при срабатывании триггеров;
- Обновление ПО;
- Настройку конфигурации.

Управление жизненным циклом PNF и VNF унифицировано и включает в себя процедуру онбординга, настройку и мониторинг. Основное отличие связано с процессом развертывания сетевой функции. PNF должна быть вначале установлена и подключена к сети вручную. В случае VNF за развертывание и запуск экземпляра функции отвечает оркестратор.

Добавление PNF в каталог оркестратора осуществляется таким же способом, как и добавление VNF. Для этого нужно нажать кнопку «+ PNF» и указать путь к архиву с PNF-пакетом. Основное отличие связано с содержимым пакета – в нем отсутствуют образы виртуальных машин.

Кроме того, для PNF, по сравнению с VNF, не применимы следующие действия:

- Масштабирование;
- Автоматическое восстановление (появится в следующих версиях);
- Управление аппаратными ресурсами (CPU, RAM, Disk);
- Создание виртуальных линков;
- Подключение к консоли VNC;
- Начальная настройка с помощью Cloud-init.

1.10.3. PNF Descriptor

PNF Descriptor (PNFD) используется оркестратором для обеспечения связности между PNF и VNF внутри сервисной цепочки. Для этого в PNF дескрипторе в разделе «connection-points» описываются точки подключения PNF.

Другие необходимые разделы PNFD:

- <configurations> - указываются скрипты для настройки PNF;
- <flavors> - <vdus> - используется для указания начальных учетных данных для подключения к PNF, а также содержит информацию для постановки PNF на мониторинг.

1.10.4. Настройка PNF в каталоге оркестратора

При выборе PNF в каталоге становится доступным окно для редактирования настроек PNF (рис.32):

- Закладка **DC Placement**. Указывается ЦОД, где установлена PNF.
- Закладка **Connection Points**. Определяет соответствие точек подключения PNF сервисным интерфейсам (SI). Для каждого сервисного интерфейса нужно указать:
  - Домен
  - Data Center
  - SDN Controller
  - Коммутатор и порт коммутатора
  - Инкапсуляцию (VLAN ID, QinQ или весь трафик порта).

Каждой точке подключения может соответствовать несколько SI.

Допустимым является не указывать SI для точки подключения. В этом случае к ней нельзя будет подключить другой элемент на графе в окне конфигуратора сервисов.

- Закладка **Management IP.** Один или несколько адресов управления для PNF должны быть выделены и настроены заранее. Эти адреса должны быть доступны для VNFM. Данные IP адреса будут использоваться оркестратором для настройки и мониторинга PNF и/или ее компонентов.

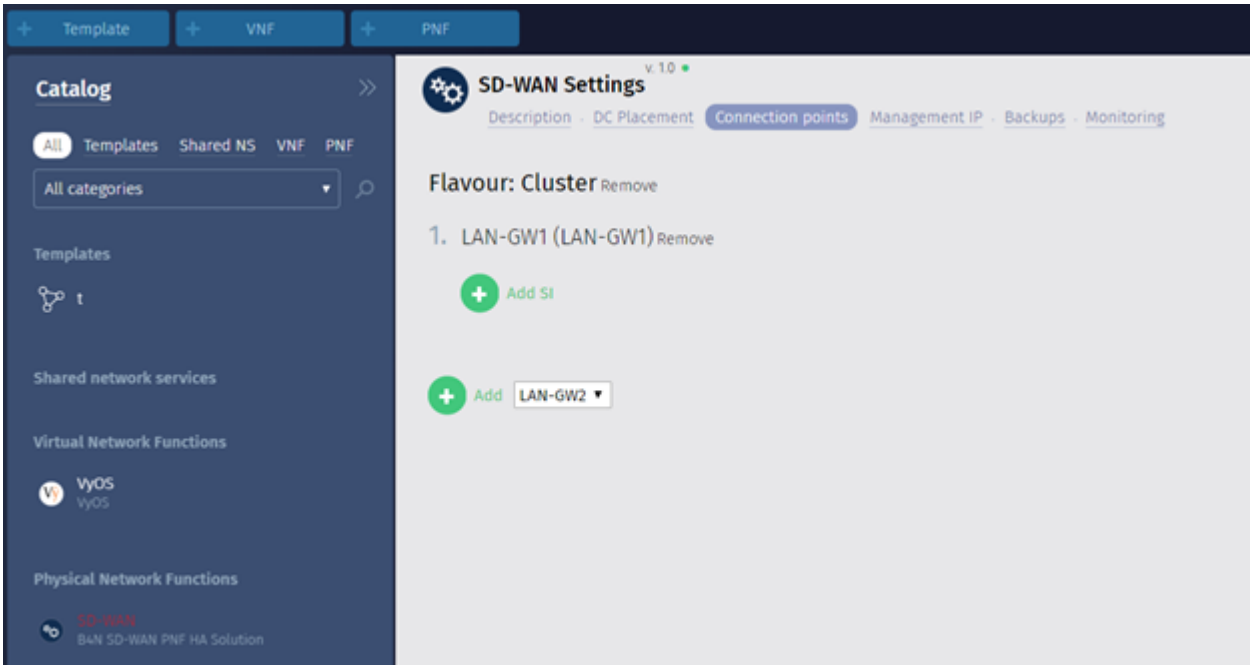


Рис. 32. Настройка PNF

IP адреса управления PNF не должны входить в динамический диапазон IP адресов управления VNF («Management IP Range» на странице управления DC). Для каждого тенанта необходимо разрешить использование PNF так же, как и VNF, в разделе Catalog на странице Tenants.

PNF не может быть удален, если выделен тенанту или использован в шаблоне сетевого сервиса. Шаблон сетевого сервиса с PNF, на базе которого развернут сетевой сервис, не может быть удален.

1.11. Создание и запуск виртуальных сетевых сервисов

1.11.1. Выделение ресурсов тенанту

Сетевые сервисы создаются и развертываются администраторами тенантов.

Предварительно администратор платформы должен выделить тенанту необходимые ресурсы (рис. 33):

- Вычислительные ресурсы под управлением одного или нескольких VIM, которые могут располагаться в одном или нескольких ЦОД;
- Точки подключения трафика тенанта (UNI);
- Элементы каталога, доступные тенанту (VNF, PNF, шаблоны сетевых сервисов). Если шаблон сервиса отмечен как доступный тенанту, автоматически ему доступны VNF и PNF, входящие в состав шаблона.

Администратор платформы может ограничить тенанту количество вычислительных ресурсов (ядра процессора, объем оперативной памяти, дисковое пространство). Пул ресурсов является общим для всех VIM, доступных тенанту. Далее администратор платформы должен выбрать пользователей или группу пользователей, которые будут обладать правами администратора тенанта.

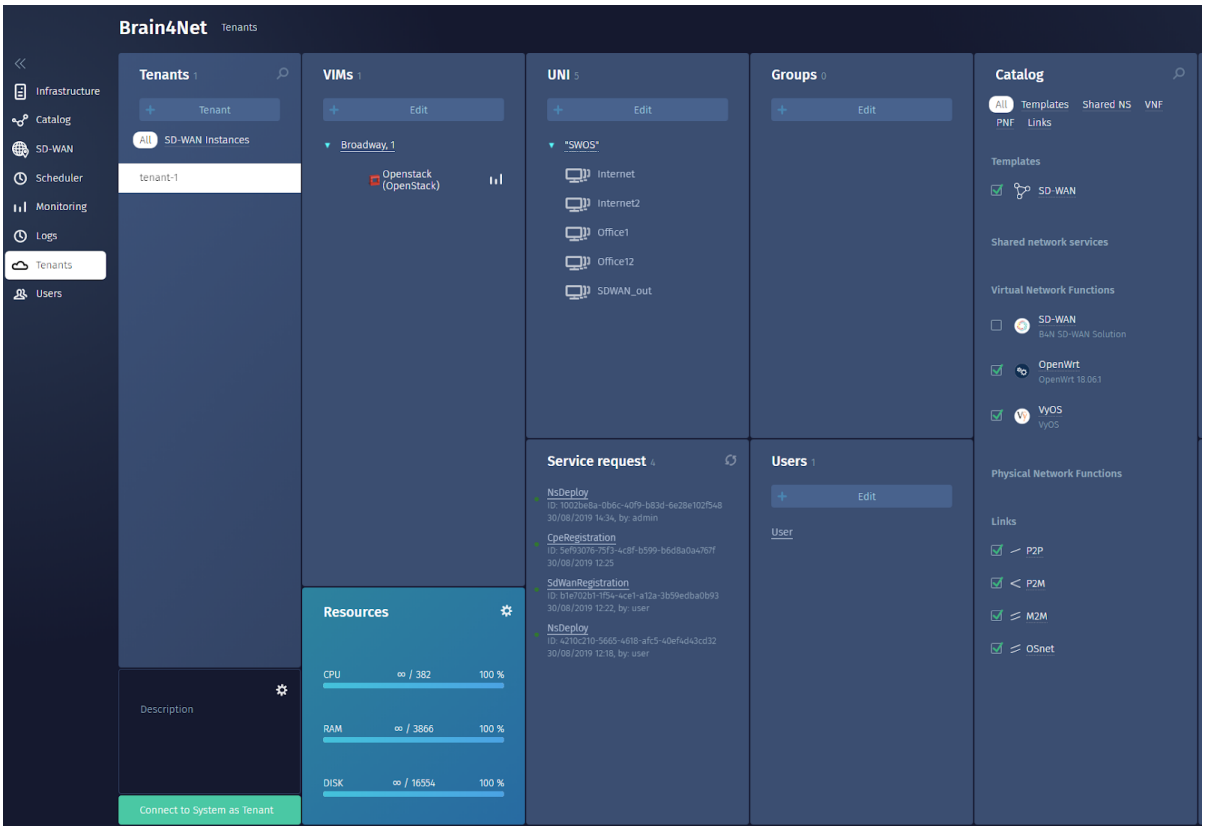


Рис. 33. Создание тенанта

Администратор платформы может подключиться к системе с правами администратора тенанта, нажав кнопку “Connect to System as Tenant” в нижней левой части экрана. Это позволяет проверить правильность настроек тенанта, а также оказывать помощь администратору тенанта в случае возникновения вопросов или проблем при создании сервисов.

1.11.2. Создание сетевого сервиса с помощью графического конструктора сервисов

Администратор тенанта при входе в систему попадает в портал самообслуживания. Для создания сервиса необходимо выполнить следующие действия:

1. В меню «Services» нажать кнопку «New service».
2. Включить в сетевой сервис необходимые элементы, выбрав их из каталога и перетащив с помощью мыши в окно конструктора сервисов:
    - Виртуальные и / или физические сетевые функции

- Виртуальные линки («Links»)
- Точки подключения («UNI»)

3. Соединить элементы в соответствии с топологией сервиса.
4. Назначить сервису имя и сохранить.
5. Нажать кнопку «Deploy».

На рис. 34 приведен пример экрана создания сервиса.

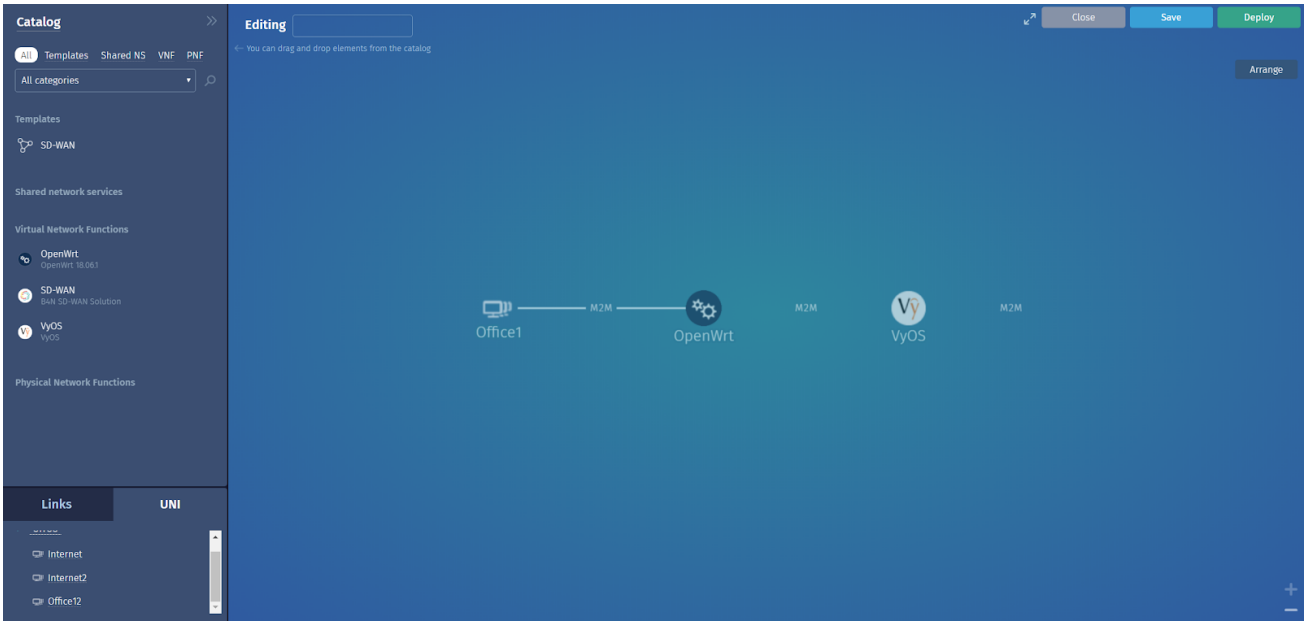


Рис. 34. Создание сервиса

После этого система автоматически переходит в режим просмотра и отображает статус развертывания сервиса.

Альтернативный способ создания сервиса - на шаге 2 выбрать в каталоге созданный ранее администратором платформы шаблон сервиса и перетащить его в окно конструктора сервисов. После этого поместить на граф элементы «UNI» и заменить ими абстрактные точки подключения шаблона.

Администратор тенанта может наблюдать за статусом развертывания компонентов сервиса, нажав на кнопку настройки сервиса и выбрав пункт меню «Open log».

Пример журнального файла развертывания сервиса приведен на рис. 35. Все журнальные файлы сохраняются в системе и доступны для просмотра администратором тенанта в портале самообслуживания («Open log» в меню настройки сервиса) и администратором платформы (раздел “Service Request” на странице тенанта).

Логируются все действия оркестратора, время, затраченное на операцию, статус операции и ее результат. Если на каком-то шаге произошла ошибка, оркестратор откатывает назад все действия системы по развертыванию сервиса, и отмечает этот файл красной точкой. При просмотре файла

можно найти операцию, при выполнении которой произошел сбой, и, кликнув на ней мышью, получить подробное описание ошибки.

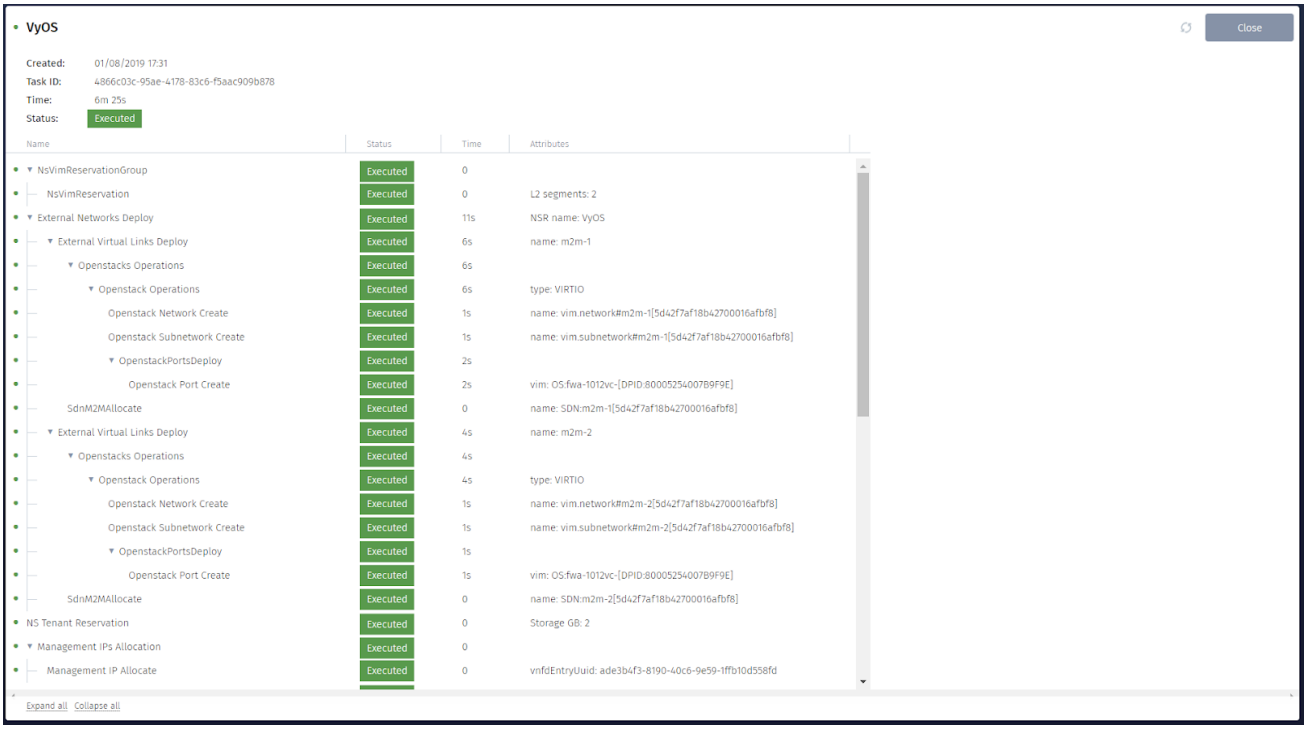


Рис. 35. Журнальный файл развертывания сервиса

1.11.3. Изменение работающего сетевого сервиса

Иногда возникает необходимость внести изменения в уже развернутый сетевой сервис.

Разрешаются следующие изменения сервисной цепочки активного сервиса.

1. Действия с точками подключения (UNI):
   - Добавление нового UNI на граф
   - Замена одного UNI на другой
   - Удаление UNI из сервисной цепочки
2. Действия с VNF:
   - Добавление VNF в начало или конец сервисной цепочки
   - Добавление VNF в середину сервисной цепочки (не для всех топологий сервиса)
   - Удаление VNF из сервисной цепочки (не для всех топологий сервиса)

Для изменения работающего сервиса необходимо нажать кнопку “Edit” в окне конфигуратора сервисов. После этого граф можно редактировать. Для активации изменений нажать кнопку “Deploy Changes”.

ТРЕБОВАНИЯ К АППАРАТНОМУ И ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ

1.12. Требования к аппаратному обеспечению для установки B4N Service Platform

B4N Service Platform имеет распределенную архитектуру и состоит из нескольких программных модулей:

- Оркестратор (B4N Orchestrator)
- Веб-интерфейс оркестратора
- Контроллер (B4N Controller)
- VNF Manager (VNFM)
- База данных MongoDB.

Модули поставляются в виде docker-контейнеров, и, как следствие, быстро развертываются и обновляются. Компоненты могут устанавливаться и масштабироваться независимо друг от друга, что позволяет при необходимости увеличить производительность системы в целом. Все компоненты могут быть установлены на один физический сервер или виртуальную машину, или могут быть распределены по нескольким серверам, находящимся на разных площадках.

Требования к аппаратным ресурсам зависят от количества управляемых коммутаторов и виртуальных сетевых функций (табл. 1.).

Таблица 1. Требования к аппаратным ресурсам для установки B4N Service Platform

Назначение	Кол-во ядер процессора	Оперативная память, Гбайт	Дисковое пространство, Гбайт	Сетевые адаптеры
Управление SDN транспортом, без учета VNF				
до 32 коммутаторов	8	32	150	2x10G
до 64 коммутаторов	12	32	150	2x10G
Оркестрация VNF				
до 1000 VNF	4	8	150	2x10G
от 1000 до 10000 VNF	8	16	250	2x10G
более 10000 VNF	12	32	500	2x10G
VNF Manager				
до 1000 VNF	2	4	150	2x10G
от 1000 до 10000 VNF	4	8	250	2x10G
более 10000 VNF	8	16	500	2x10G

Установка платформы без резервирования рекомендуется только для тестирования системы в лаборатории. Для промышленных систем необходимо использовать кластер из трех узлов.

Мониторинг виртуальных сетевых функций осуществляют серверы Zabbix и Zabbix Proxy. Требования к Zabbix приведены в таблице 2. Если платформа будет использоваться для управления несколькими ЦОД, необходимо предусмотреть установку VNF Manager и Zabbix Proxy в каждом ЦОД.

Таблица 2. Требования к аппаратным ресурсам для установки Zabbix

Компонент, назначение	Кол-во ядер процессора	Оперативная память, Гбайт	Дисковое пространство, Гбайт	Сетевые адаптеры
Zabbix Server				
до 1000 VNF	2	2	50	2x1G
от 1000 до 10000 VNF	4	8	150	2x10G
более 10000 VNF	8	16	1024	2x10G
Zabbix Proxy				
до 1000 VNF	2	2	50	2x1G
от 1000 до 10000 VNF	4	8	50	2x10G
более 10000 VNF	8	16	50	2x10G

1.13. Требования к программному обеспечению для установки B4N Service Platform

Поддерживаемые операционные системы (x64):

- Debian версии 8.1 и выше
- Ubuntu версии 14.04 и выше
- CentOS версии 7.1 и выше
- RedHat версии 7.1 и выше
- SUSE версии 12.4 и выше

Для установки системы требуется Docker версии 1.5 или выше.

1.14. Требования к коммутаторам OpenFlow

Список поддерживаемых коммутаторов OpenFlow составляется по результатам тестирования в лаборатории Brain4Net и постоянно пополняется новыми моделями. Актуальная версия доступна по ссылке: https://brain4net.ru/supported-devices/.

ОБЕСПЕЧЕНИЕ БЕЗОПАСНОСТИ

Сеть SDN состоит из трех уровней - уровень коммутации (data plane), уровень управления сетью (control plane) и уровень оркестрации. Безопасность решения в целом определяется безопасностью каждого уровня в отдельности, а также взаимосвязей между ними. Каждый компонент платформы B4N Service Platform на своем уровне обеспечивает необходимые функции безопасности.

Наиболее критичным компонентом сети SDN является контроллер. Поэтому необходимо обеспечить физическую безопасность серверов, на которых установлено ПО контроллера, а также защиту операционной системы. Установка и обновление ПО контроллера должно производиться с использованием доверенных дистрибутивов. Административные интерфейсы контроллера должны быть доступны только из изолированной частной сети.

Контроллер B4N не имеет графического интерфейса. Управление контроллером осуществляется через пользовательский интерфейс оркестратора или API.

Для обмена сообщениями между контроллером и оркестратором B4N в качестве API используется gRPC. Аутентификация и шифрование данных обеспечиваются с помощью SSL/TLS. В качестве альтернативы поддерживается NETCONF Configuration Protocol over Secure SHell (SSH).

“Южный” интерфейс для связи с коммутаторами OpenFlow может использовать защищенный транспортный протокол TLS.

Осуществляется запись всех событий в журнальные файлы с возможностью задания степени подробности - Error, Warning, Info, Debug, Trace. Журнальные файлы привязаны к каждому объекту системы. Предусмотрена возможность экспорта выбранного лога в файл на внешней системе (через графический интерфейс оркестратора).

На уровне оркестрации разделение прав доступа пользователей к системе реализовано с помощью назначения роли каждой учетной записи.

Ведется отдельный журнальный файл, в который записывается информация о действиях пользователей, в том числе:

- успешные и неуспешные попытки входа в систему с указанием времени входа, имени пользователя и IP адреса, с которого осуществлялся доступ к системе;
- действия и задачи, запускаемые от имени пользователя.

Из списка действий и событий есть возможность быстро перейти на страницу указанного в записи объекта (пользователя, VNF и т.п.).

Для обеспечения безопасного обмена данными с контроллером коммутаторы OpenFlow должны поддерживать протокол TLS и иметь возможность загрузки сертификатов.

Применительно к NFV можно выделить следующие направления обеспечения безопасности:

- Безопасность логических интерфейсов,
- Защита гостевой ОС,
- Целостность дистрибутивов VNF.

Безопасность логических интерфейсов VNF и данных, которые передаются через них, обычно достигается комбинацией списков контроля доступа и безопасных протоколов, таких как IPSec, а также дополняется внешними сетевыми экранами. Платформа B4N позволяет разделять логические интерфейсы по принадлежности к разным тенантам, не допуская передачи трафика между ними.

Целостность дистрибутивов VNF реализуется подсчетом контрольных сумм или с помощью сертификатов X.509 и обеспечивается поставщиком VNF.

Безопасность гостевой операционной системы входит в зону ответственности VNF, которая ее использует. Защита гостевой ОС обычно подразумевает удаление или деактивацию сервисов, не требующихся для функционирования сетевой функции, а также использование безопасных протоколов, таких как SSH, везде где это возможно.

СЦЕНАРИИ ПРИМЕНЕНИЯ

1.15. Применение технологий SDN/NFV при проектировании и модернизации сети агрегации региональных филиалов сервис-провайдера

1.15.1. Задачи и сервисы региональной сети передачи данных (РСПД)

Региональная сеть передачи данных обеспечивает решение следующих задач:

- Транзит L2 трафика B2C и B2B абонентов к сервисным устройствам;
- Обеспечение качества обслуживания (SLA) клиентам сети;
- Бесперебойная доставка клиентского трафика, минимальное время восстановления сервиса в случае обрыва линий связи или выхода из строя оборудования;
- Транзит multicast и VoIP трафика;
- Снижение операционных и капитальных издержек, вывод из эксплуатации и замена устаревшей инфраструктуры современными аналогами.

Сеть РСПД оказывает сервис транспорта для следующих видов трафика:

- B2B: Интернет, VoIP, IPTV, L2/L3 VPN;
- B2C: Интернет, VoIP, IPTV.

1.15.2. Классы обслуживания, предоставляемые пакетной сетью РСПД

Для обеспечения качества обслуживания на пакетной сети РСПД филиалов организуется 3 класса обслуживания для клиентского трафика. Параметры классов обслуживания и их назначение описаны в Табл. 3.

Таблица 3. Классы обслуживания в сети РСПД

Название	Рекомендуемое назначение	Контроль соблюдения контрактных обязательств	Вариация задержки (не более)	Потеря пакетов (не более)	Задержка
Real-Time	Передача интерактивного голоса и видео (VoIP, ВКС). B2B: VoIP B2C:VoIP	Трафик свыше контракта подлежит сбросу	10мс	0.1%	до 20 мс плюс время прохождения луча (из расчёта 5 мкс на километр для оптического кабеля)
Business Critical	Интерактивный трафик корпоративных информационных систем, критичный к потерям пакетов и задержкам (SAP, корпоративные порталы и все прочие приложения с графическим интерфейсом). B2B: Интернет	трафик свыше контракта будет обработан по классу Best Effort	не нормируется	0.5%	до 40мс плюс время прохождения луча (из расчёта 5 мкс на километр для оптического кабеля)
Best effort	Трафик низкого приоритета, не блокирующий работу организации (backup, доступ в Internet, почта и др.) B2B:Интернет B2C:Интернет	трафик свыше суммарного объёма контракта будет сброшен	не нормируется	1%	не нормируется

Качество сервисов для указанных классов в сети SDN обеспечивается созданием на коммутаторах правил, помещающих трафик классов в соответствующие аппаратные очереди коммутаторов на основе маркировки пакетов. Предварительно аппаратные очереди единообразно конфигурируются для обеспечения требуемых характеристик обработки трафика.

Сеть на основе технологии SDN может в дополнение к используемой в традиционной сети модели агрегированной DiffServ QoS использовать интегрированный per-PW QoS и обеспечивать отслеживание использования приоритетной полосы транспортных каналов (Admission Control).

1.15.3. Архитектура сети РСПД. Транспортный уровень

Транспортный уровень сети формируется оборудованием магистрального сегмента и каналами связи между узлами (арендованными или собственными).

Топология транспортного уровня определяется доступным набором магистральных каналов. Для организации каналов связи могут быть использованы следующие способы:

- Темная оптика;
- Арендованные выделенные каналы;
- Арендованные виртуальные выделенные каналы “точка-точка” (PW, L2-tunnel);
- Арендованные виртуальные частные сети L2VPN.

При организации каналов должны быть соблюдены следующие условия:

1. Каналообразующее оборудование должно иметь клиентский интерфейс Ethernet;
2. Каналообразующее оборудование должно пропускать фреймы Ethernet длиной не менее 1538 байт;
3. Каналообразующее оборудование должно деактивировать клиентский интерфейс при нарушении целостности канала. В случае невозможности организции этого функционала в каналообразующем оборудовании необходимо использовать OF-коммутаторы с поддержкой функционала BFD для быстрого детектирования обрыва канала. Примеры организации такого канала показаны на рис.36.

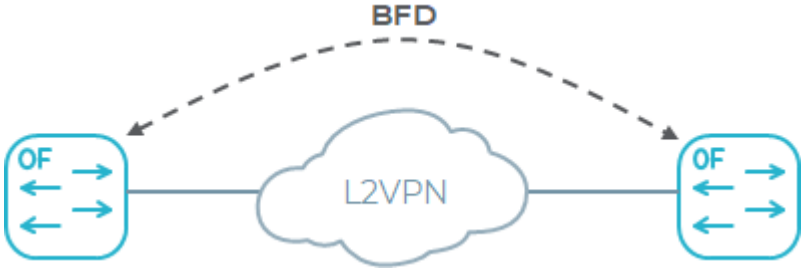


Рис. 36. Пример организации транспортных каналов с функцией быстрого детектирования отказа канала

1.15.4. Архитектура сети РСПД. Транспортный уровень

На уровне пакетной сети организуется логическая топология «двойная звезда» (рис. 37). Топология «двойная звезда» существенно упрощает эксплуатацию, планирование и масштабирование сети, позволит снизить операционные издержки и упростить поиск неисправности в аварийных ситуациях.

Пакетная сеть региональной сети решает задачи транзита L2 трафика клиентов Сервис-провайдера к сервисным устройствам магистральной сети. Кроме того, в отличие от традиционной сети SDN-сеть региона может содержать часть локализованных VNF-сервисов на базе NFV-фермы под управлением общего контроллера SDN и оркестратора.

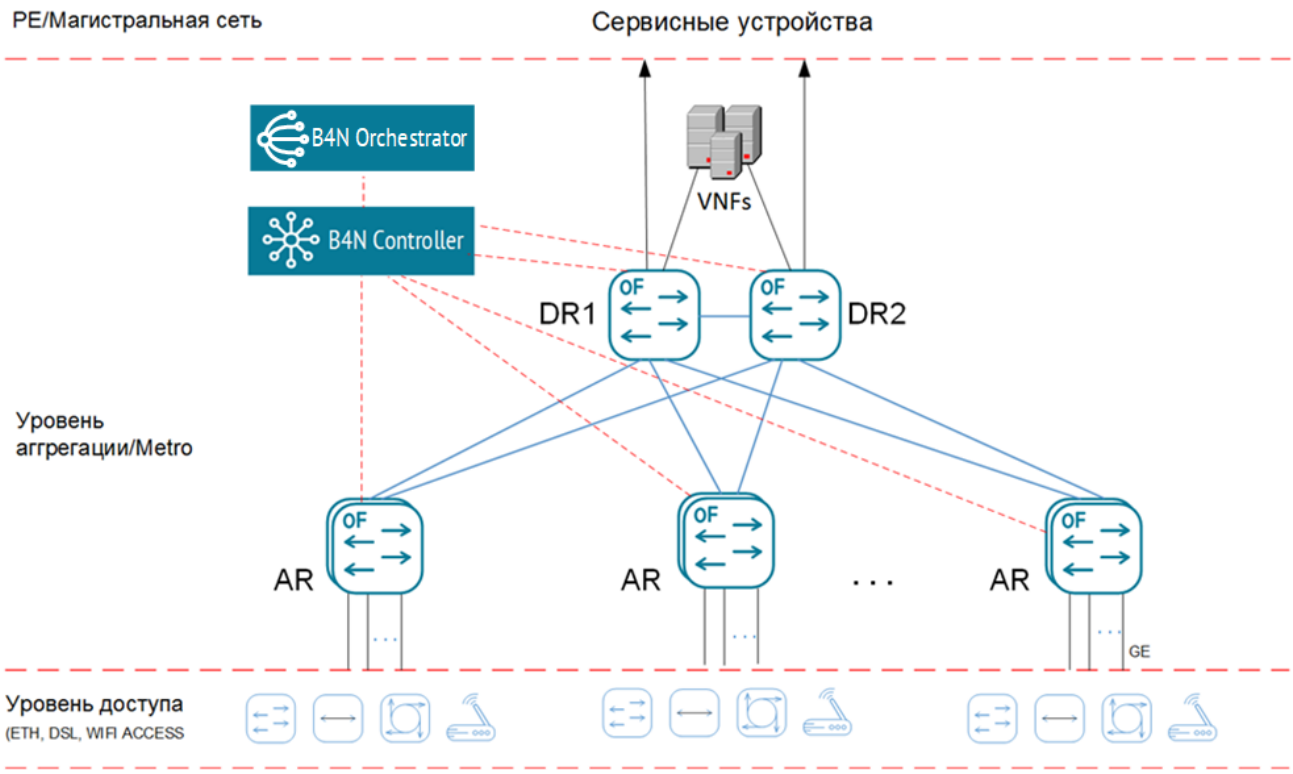


Рис. 37. Топология «двойная звезда»

В SDN-сети на уровне агрегации не используется никакого внутреннего протокола маршрутизации между коммутаторами. Всю маршрутизацию внутри сети выполняет контроллер SDN, используя алгоритм Дейкстры поиска кратчайшего пути на основе весов (cost). Кроме кратчайшего пути, рассчитывается наиболее предпочтительный резервный маршрут, для быстрой перемаршрутизации к той же конечной точке. Веса транспортным каналам назначаются исходя из их пропускной способности. При использовании механизма резервирования полосы для приоритетного трафика (Admission Control) для расчета пути используется отдельная система весов, которая динамически корректируется исходя из уже использованной полосы.

Управление пакетными устройствами сети РСПД (коммутаторы агрегации) рекомендуется организовывать с помощью служебной сети управления транспортным DWDM оборудованием – OSC (Optical Service Channel). При невозможности организации управления по OSC каналу или по причине отсутствия оборудования DWDM, сеть управления организуется с помощью стандартных средств, с логическим, и по возможности физическим, разделением каналов управления и каналов сети передачи данных. При этом трафик управления маркируется как приоритетный класс и помещается в приоритетную очередь с резервированием не менее 512 Кбит/с на устройство.

1.15.5. Механизм предоставления сервиса B2C абонентам

Основная сервисная модель предоставления услуг B2C абонентам - это использование C-VLAN и туннельного протокола PPPoE. Схема предоставления услуги доступа в Интернет для B2C абонентов представлена на рис. 38. Для каждого абонента выделяется отдельный C-VLAN.

Коммутатор агрегации (AR) добавляет вторую (S-VLAN или внешнюю) метку к Ethernet пакетам абонентов. Допускается подключать GPON OLT с 10 GE интерфейсами в узел AR. OLT, обслуживающие большое количество абонентов (> 4096) должны добавлять 2 и более S Vlan метки.

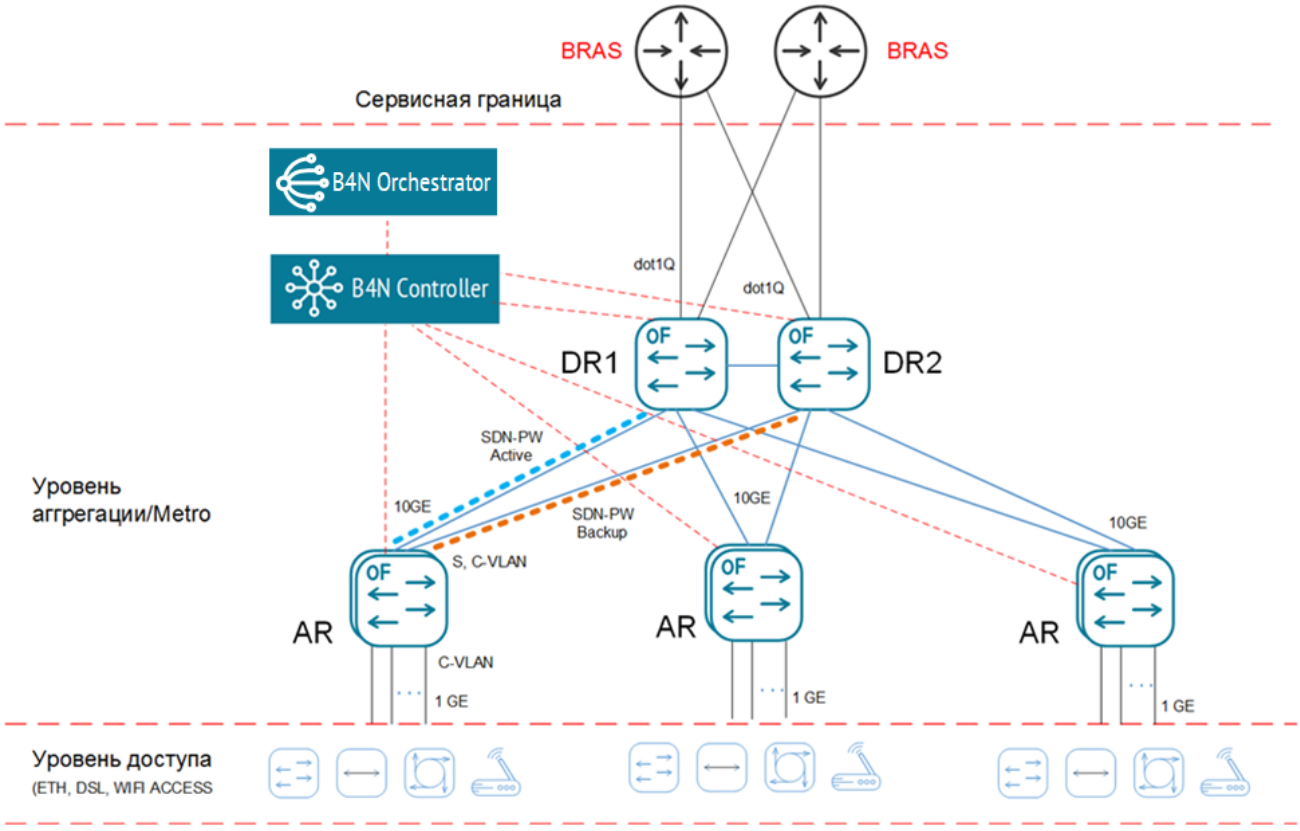


Рис. 38. Схема сети для предоставления сервиса B2C абонентам

Коммутатор агрегации (AR) подключается к двум центральным узлам агрегации (DR1 и DR2) с помощью 10 GE каналов. Для обеспечения резервирования на участке AR - DR контроллер SDN организует два point-to-point L2 туннеля (SDN Pseudowire, аналог MPLS Pseudowire). Трафик B2C, агрегированный на коммутаторе агрегации, доставляется от AR к DR с помощью выделенного SDN Pseudowire (SDN-PW), каждый S-VLAN доставляется на DR в отдельном SDN-PW. При неработоспособности основного SDN-PW происходит переключение на резервный SDN-PW, который терминируется на другом коммутаторе DR. Центральные коммутаторы сети агрегации (DR1 и DR2) подключены к сервисным устройствам BRAS по полносвязной схеме таким образом, что каждый из DR подключен ко всем BRAS. Для каждого S-VLAN на узле DR должны быть доступны два BRAS, это необходимо для резервирования и балансировки нагрузки на BRAS.

1.15.6. Механизм предоставления сервиса B2B абонентам

В сети оператора связи могут быть организованы различные схемы предоставления сервиса B2B абонентам:

4. Схема, аналогичная используемой для предоставления сервиса B2C абонентам, - подключение B2B абонентов к оборудованию BRAS;
5. Подключение B2B абонентов к выделенному оборудованию BPE (Business PE) с организацией дополнительных мер на сети РСПД по защите и резервированию абонентского трафика.

Схема предоставления сервиса B2B абонентам представлена на рис. 39.

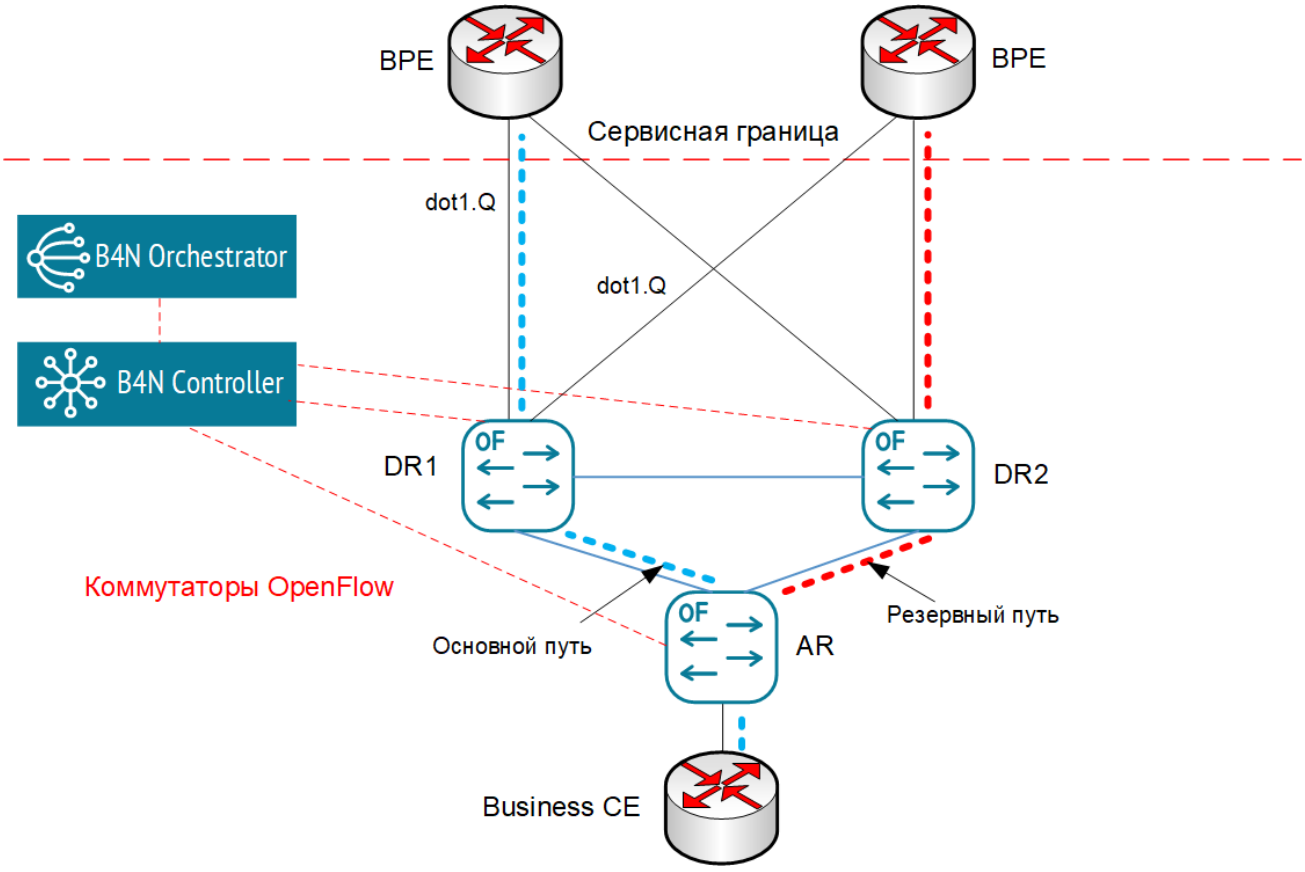


Рис. 39. Схема предоставления сервиса для B2B абонентов

B2B абоненты подключаются к коммутатору агрегации. Для таких абонентов на коммутаторе агрегации выделяется отдельная внешняя метка и создаются 2 SDN-PW между AR и DR - один активный, другой резервный. Для защиты от отказа одного из BPE необходимо создать 2 отдельных логических L2 канала между CE клиента и каждым BPE. Два SDN-PW между AR и BPE должны, в нормальном режиме работы РСПД, проходить по разным географическим маршрутам ВОЛС на участке AR – DR. На каналах между DR и BPE настраивается инкапсуляция dot1Q.

1.15.7. Доставка трафика IPTV

В SDN-сети РСПД не используется внутренний протокол IP Multicast маршрутизации, расчет дерева распространения multicast-групп выполняется на контроллере на основе расчета кратчайших путей от ближайшей точки прохождения группы (или источника) до потребителя. При этом для группы резервируется указанная гарантированная полоса. Инжектирование IPTV трафика в сеть должно происходить через выделенный для этих целей канал 10 Гбит/с между узлами DR и INTERNET GW. От устройств агрегации трафик IP Multicast доставляется на устройства концентрации и доступа в одном Multicast VLAN, в котором используется технология IGMP Snooping. На устройствах доступа рекомендуется использовать технологию IGMP Proxy. На устройствах агрегации необходимо использовать динамическую подписку на каналы и резервировать полосу для multicast трафика 2,5 Гбит/с. Возможно использование статической подписки. Резервирование multicast трафика на участке AR-DR обеспечивается SDN-контроллером. Каждый DR получает все multicast группы со стороны INTERNET GW. При отказе одного из коммутаторов DR дерево PIM перестраивается для работы с другим работоспособным коммутатором DR.

1.15.8. Организация присоединения сети РСПД к магистральной сети

Для присоединения региональной сети к магистральной используются 10 GE каналы с инкапсуляцией IEEE 802.1ad. На каналах между DR и сервисными устройствами (BRAS, BPE) организуется съем трафика для СОРМ с помощью пассивных оптических сплиттеров.

Сеть SDN эффективно решает основную задачу региональной сети передачи данных оператора - транспорт L2 трафика клиентов к сервисным устройствам. Для создания и сигнализации туннелей SDN-PW используется протокол OpenFlow.

Между DR и сервисными устройствами настраивается инкапсуляция dot1.Q.

В нормальном режиме работы типовой AR потребляет до 20 Гбит/с трафика, при аварии на одном из каналов – 10 Гбит/с. В случае крупного районного центра AR может быть подключен более чем одной парой каналов 10G. В связи с низкой экономической эффективностью, интерфейсы 40GE и 100GE рассматриваются как исключение при отдельном обосновании.

Сервисные устройства – NAT, DPI, CDN размещаются в магистральной сети. Инфраструктурный ЦОД предусматривается в каждом региональном филиале. В этом же ЦОДе могут размещаться виртуальные сетевые функции (маршрутизаторы, межсетевые экраны, системы IDS/IPS и т.д.) под управлением единого оркестратора с сетью агрегации.

1.15.9. Резервирование трафика в сети РСПД

Для обеспечения необходимого качества сервиса клиентам необходимо обеспечить резервирование и защиту трафика в сети РСПД. Способ организации резервирования зависит от участка сети:

1. **Коммутатор уровня доступа – коммутатор агрегации (AR).** Резервирование обычно не используется. При необходимости резервирования рекомендуется подключать коммутатор уровня доступа к коммутатору агрегации двумя каналами, проложенными по

разным трассам ВОЛС с агрегацией их в один логический канал. Агрегация портов может быть организована с использованием протокола LACP LAG (если коммутаторы с обеих сторон его поддерживают). Другой вариант обеспечения отказоустойчивости на уровне AR - установка двух коммутаторов AR и создание основного интерфейса сервиса на первом коммутаторе и резервного интерфейса - на втором (multi-homing).

2. **Коммутатор агрегации (AR) – центральный коммутатор сети агрегации (DR).** Использование кольцевых топологий на уровне ВОЛС, использование SDN-PW в режиме активный/резервный. Между устройствами AR и DR1,2 создаются активный и резервный SDN-PW. Принадлежность к SDN-PW определяется внешней Vlan меткой (S-tag). Основное требование – обеспечить быстрое переключение на резервный канал в случае неработоспособности основного.

3. **DR – BRAS/BPE.** Резервирование BRAS достигается путем подключения каждого маршрутизатора DR одновременно как минимум к двум BRAS филиала. При выходе из строя одного из BRAS, его абоненты переключаются на второй BRAS.

Резервирование от отказа BPE достигается использованием 2-х логических каналов между CE клиента и каждым из BPE. Подключение каждого DR к каждому BPE необходимо для защиты от отказа DR. При отказе одного из DR происходит переключение SDN-PW на другой DR, клиентский трафик пойдет к BPE через другой работоспособный DR.

4. **Резервирование контроллера и оркестратора SDN.** Контроллер и оркестратор SDN должны быть развернуты в виде отказоустойчивых кластеров с горячим резервированием, размещенных на разных, георезервированных площадках. Контроллер рассчитывает и загружает на сетевые устройства предрассчитанные правила маршрутизации/коммутации пакетных данных в сети РСПД. Сеть SDN должна обеспечивать основной сервис SDN-PW при недоступности всех контроллеров сети. Динамические сервисы типа IGMP, P2M, M2M должны при этом работать в snapshot-режиме (с сохранением состояния сервиса на момент обрыва соединения с контроллером). Каналы управления коммутаторов в случае использования out-of-band интерфейсов должны быть зарезервированы на транспортном уровне.

ЗАКЛЮЧЕНИЕ

Подход компании Brain4Net при разработке сервисной платформы для операторов связи, сервис-провайдеров и корпоративных заказчиков заключается в создании единой унифицированной сетевой фабрики, которая дает возможность доставлять пользовательский трафик через сеть передачи данных в NFV-платформу, расположенную в ЦОД, и бесшовно интегрировать сетевые сервисы с NFV или другими облачными приложениями.