НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

(УНИВЕРСИТЕТ ИТМО)

Реферат

Дисциплина: Основы технологии программно-конфигурируемых сетей

Тема: OpenDaylight SDN

Выполнил студент гр. K41114: Левенцов Дмитрий Владимирович

Преподаватель: Шкребец Александр Евгеньевич

Санкт-Петербург

2020

Введение

OpenDaylight (ODL) — проект с открытым исходным кодом под эгидой Linux Foundation, направленный на содействие внедрению и развитию программно-определяемых сетей (SDN) путем создания общей отраслевой платформы.

ODL — это промышленное ПО на основе Java, управляемое консорциумом Linux Founsion включающее около 50 корпоративных представителей, таких как Brocade, Cisco, Citrix, Dell, Ericsson, HP, IBM, Juniper, Microsoft и Red Hat. Благодаря тому что ODL – открытая платформа, представители сообщества, конечные пользователи и даже клиенты могут участвовать в определении, анализе, разработке и тестировании его архитектуры и существующих модулей. Более того, они могут внести свой вклад, предлагая новые инициативы и представляя новые предложения техническому сообществу, уделяющему особое внимание ODL. Задача проекта ODL - создать совместное сообщество, способствующее успеху и принятию SDN.

Software Defined Networking

SDN (Программно-конфигурируемые сети) - это отделение программного обеспечения, управляющего функцией пересылки пакетов (control plane - управляющий уровень) от сетевого аппаратного элемента, который пересылает пакеты (data plane - передающий уровень). Правила пересылки вырабатываются логически централизованным программируемым контроллером для обеспечения возможности пересылки данных по дампам сетевых элементов. **Дамп данных** – пересылка данных между двумя системами для их дальнейшего использования сторонними приложениями или анализа человеком. Защищенный **южный интерфейс** (SB – SouthBound – с помощью него приложение обращается к нижестоящему в архитектуре системы, приложению) устанавливается между каждым элементом базовой сети в целом и контроллером, через который передаются правила. Сетевые

элементы хранят правила в цепочке потоковых таблиц, и, когда ни одна запись не соответствует полям заголовка пакета принятого пакета в таблицах потоков, маршрутизаторы или коммутаторы отправляют его в контроллер. Если найдено соответствующее правило, применяется определенное действие (сброс, пересылка на определенный порт). Если совпадений не найдено, пакет может быть либо отброшен, либо отправлен на контроллер. На «рисунке 1» показан протокол OpenFlow (OF), наиболее распространенный интерфейс SB между многочисленными разработанными контроллерами SDN и элементами сети.

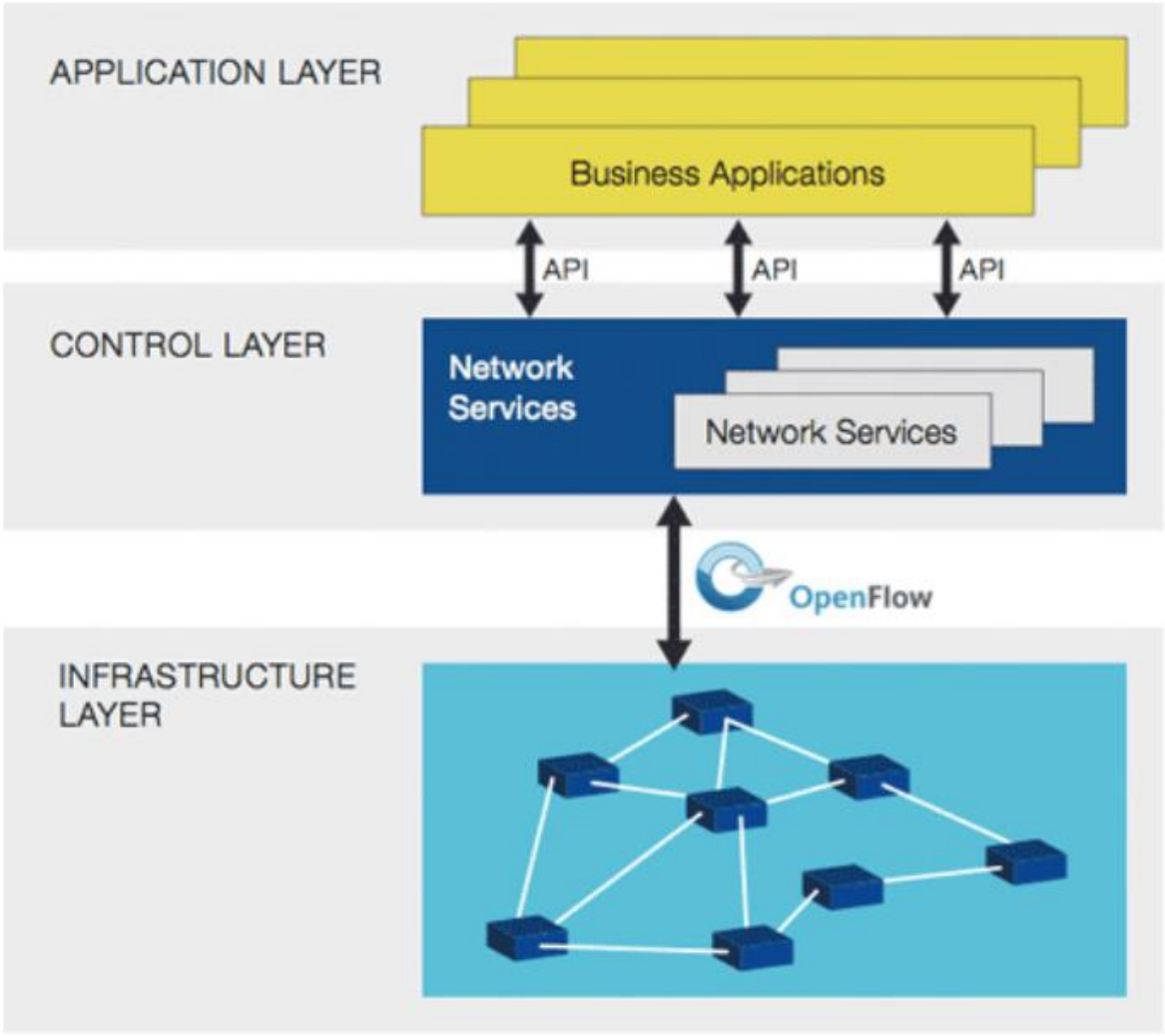

Рисунок 1 – Архитектура SDN

Моделирование и программирование конфигурации сети

Эволюция подхода, основанного на моделировании, приводит к разработке моделей для автоматического управления и настройки сетей.

NETCONF — это протокол для управления и настройки сетевых элементов. Данный протокол поддерживает конфигурации в хранилищах данных и предусматривает набор операций нижнего уровня: извлечение, конфигурирование, копирование и удаление, которые можно выполнять в хранилищах данных. NETCONF использует кодирование данных на основе XML как для данных конфигурации, так и для сообщений протокола. NETCONF поддерживает RemoteProcedureCalls (RPCs), оповещения, любые модели данных и операции отката; отделяет конфигурации от состояния рабочих данных; позволяет сохранять и восстанавливать, сравнивать конфигурации. Конфигурирование и операции хранения данных в NETCONF реализованы в виде RPCs. Yet Another Next Generation (YANG) — это язык моделирования данных, изначально разработанный для моделирования RPCs, оповещений, конфигураций, данных о состоянии элементов сети, а также ограничений, которые должны быть применены к данным. Кроме того, YANG можно использовать для моделирования сервисов, протоколов, политик сети и клиентов. YANG структурирует данные в дерево, которое можно использовать для доступа к конфигурации данных, определенной в NETCONF, манипулирования этими данными. YANG также определяет модели данных в модулях и подмодулях, где данные могут быть импортированы и экспортированы между ними.

RESTCONF — это интерфейс прикладного программирования, который обеспечивает доступ к операциям NETCONF по HTTP, которые определены как RPCs в YANG. Данные конфигурации — это ресурсы, направляемые универсальными идентификаторами ресурсов (URI), которые могут быть возвращены методом GET и могут обрабатываться методами

PATCH, DELETE, POST и PUT. Данные форматируются с помощью XML или JavaScript Object Notation (JSON).

Архитектура OpenDaylight

Архитектура ODL разработана на основе Инициативы Open ServicesGateway (OSGi), которая представляет собой модульную структуру разработки, в которой слабосвязанные модули образуют всю платформу. Модули могут быть построены независимо с возможностью импорта и экспорта данных друг от друга.

Архитектура ODL формируется в виде многоуровневой структуры: уровень сетевых приложений сверху, уровень контроллера платформы в середине, а сетевые элементы представляют нижний уровень. Сердцем ODL является средний уровень, который содержит: основные сетевые функции, такие как: топология, статистика и службы пересылки; сетевые функции платформы, которые включают модули для конкретных сетевых задач; а также уровень абстрагирования услуг (SAL) обеспечивающий доступ от служб плоскости мониторинга, управления и приложений к службам и приложениям плоскости приложений. Этот уровень абстракции услуг помогает сообществу ODL сосредоточиться на разработке приложений, а не на кодировании взаимодействия между различными уровнями.

Архитектура ODL представлена следующими слоями: сетевые приложения и сервисы, платформа контроллера, а также интерфейсы и протоколы SB (Рисунок 2). Все эти слои образуют уровень управления (control plane). Передающий уровень (data plane) не является частью архитектуры ODL, он в основном описывает различные сетевые элементы (физические и виртуальные) в базовой сети.

\

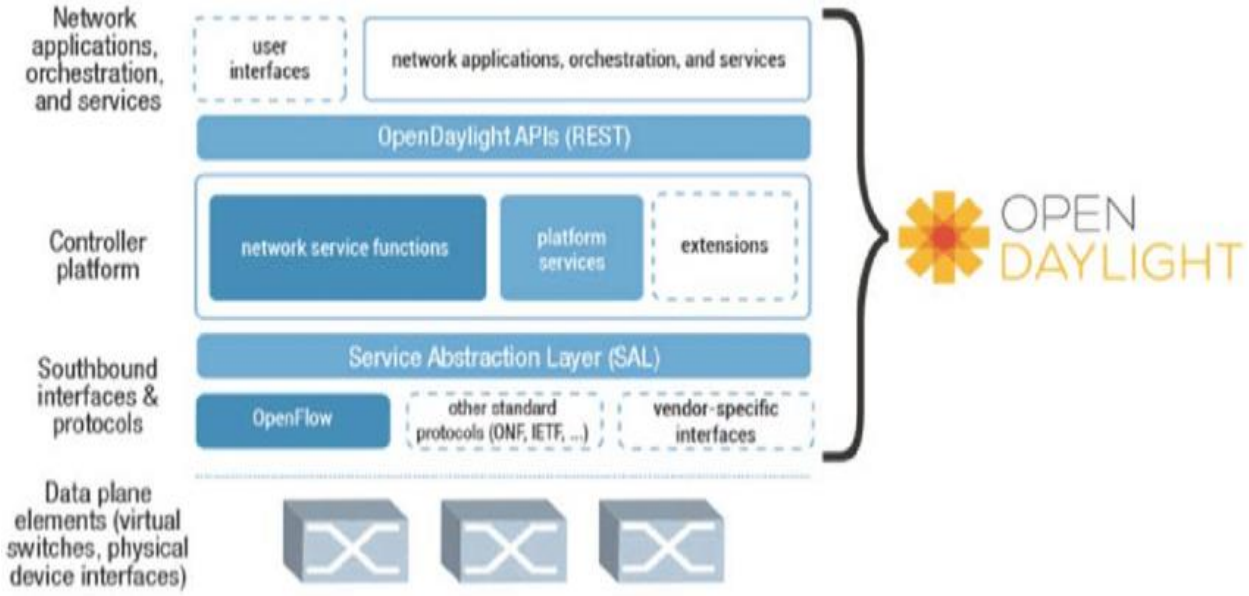

Рисунок 2 – упрощённая схема архитектуры OpenDaylight

Далее будут приведены подробные сведения о модулях и компонентах, входящих в каждый слой ODL на основе полной схемы архитектуры ODL (рисунок 3).

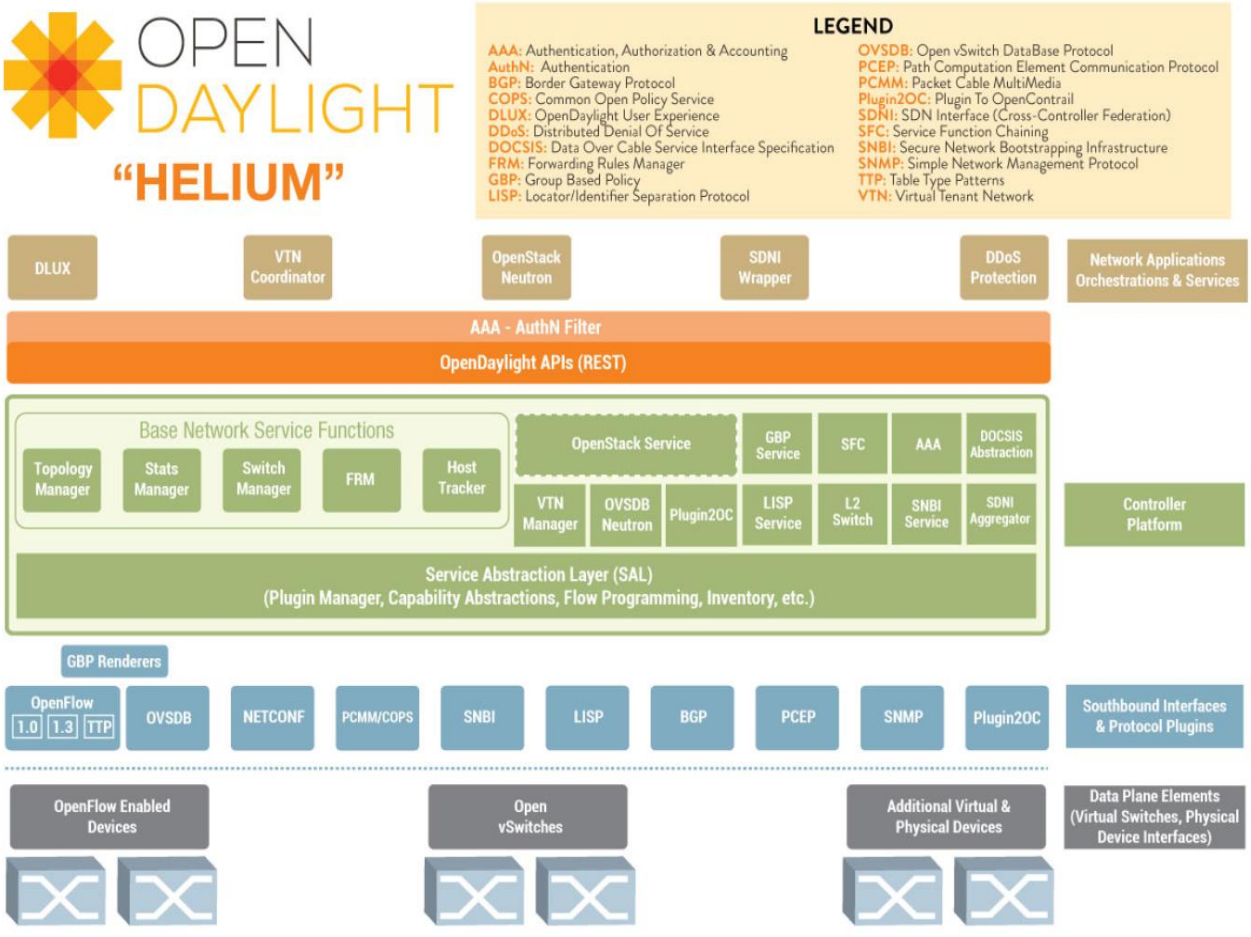

Рисунок 3 – полная схема архитектуры OpenDaylight

Controller Platform

ODL является модульным, подключаемым и гибким контроллером. Платформа контроллера является основным уровнем в архитектуре ODL, который обеспечивает абстракцию SDN. Этот уровень предоставляет открытые API северного интерфейса (NorthBound - NB) сетевым приложениям для управления и настройки физических и виртуальных элементов в сети. Он также состоит из базовых функций сетевых служб (BNSF – Base Network Service Functions), функций сетевых служб платформы (PNSF – Platform Network Service Functions) и служб уровня абстракции (SAL - Service Abstraction Layer), которые будут рассмотрены далее.

1. **Base Network Service Functions:** BNSF отвечает за сбор статистики и информации о элементах в пределах всей сети, возможностях данных элементов. Они предоставляют NB API приложениям для доступа к собранной информации и статистике. Рассмотрим следующие встроенные в ODL сетевые сервисы:
    - **Topology Manager.** Менеджер топологии хранит информацию об управляемых коммутаторах в операционном поддереве топологии. Он формирует это поддерево, прослушивая оповещения о добавлении или удалении коммутатора. Сетевые приложения, которым необходимо просмотреть сеть, могут использовать Topology Manager через NB API.
    - **Statistic Manager.** Диспетчер статистики собирает статистическую информацию из управляемых коммутаторов. Он отправляет статистические запросы всем коммутаторам и обслуживает все отклики в оперативном дереве статистики. Статистическая информация о портах коммутатора, таблицах и потоках предоставляется менеджером статистики.

- **Switch Manager.** Диспетчер коммутаторов хранит сведения о коммутаторах и их портах для идентификации обнаруженных коммутаторов. Для каждого обнаруженного коммутатора он сохраняет свои параметры в дереве данных диспетчера коммутаторов.
- **Forwarding Rules Manager (FRM).** Диспетчер правил пересылки проверяет наличие обновлений потока, разрешает их конфликты и проверяет их. Он предоставляет основные правила пересылки, такие как правила OpenFlow, а также устанавливает правила пересылки в управляемые коммутаторы через интерфейс SB независимо от спецификаций коммутатора.
- **Host Tracker** отслеживает местоположение конечного хоста во всей сети (коммутатор и порт, к которому он подключается) и сохраняет соответствующую информацию (MAC-адрес, сетевой адрес, тип коммутатора и тип порта). База данных HostTracker заполняется динамически с использованием MAC-адресов хостов или вручную с использованием NB API.

2. **Platform Network Service Functions**: Контроллер ODL содержит подключаемые службы, выполняющие определенные сетевые задачи а также другие расширения для улучшения функциональности SDN. Рассмотрим некоторые службы:
    - **Affinity Metadata Service** (Служба метаданных) обеспечивает NB API данными сетевых требований приложений для передачи рабочей нагрузки на контроллер. Контроллер может обеспечивать инфраструктуру сети между конечными точками, чтобы удовлетворять данным требованиям или уменьшить результирующую рабочую нагрузку.
    - **Virtual Tenant Network (VTN) Manager** (Менеджер виртуальной сети арендатора) - создает и управляет мульти-арендной виртуальной сетью. VTN позволяет пользователям проектировать

логическую сеть (внешний вид сети L2 / L3) независимо от топологии физической сети.

- **L2 Switch (коммутатор L2)** - обеспечивает функциональность коммутатора L2 и создает несколько общих повторно используемых сервисов, таких как отслеживание адресов, базовый протокол связующего дерева, модульная обработка пакетов, управляемая событиями, и вычисления оптимального пути.
- **Service Function Chaining (SFC).** Цепочка служебных функций предоставляет возможность определять цепочку сетевых служб (таких как брандмауэр, маршрутизаторы, балансировщики нагрузки) в упорядоченном списке для определения пути обслуживания для передачи данных.
- **Group-Based Policy (GBP).** Групповая политика отделяет требования к подключению приложений от базовых деталей сетевых элементов посредством модели политики, ориентированной на приложения. Он классифицирует конечные точки сети по группам на основе требований приложений и применяет к этим группам политику, ориентированную на приложения.
- **Authentication, Authorization, and Accounting (AAA) Service.** Служба аутентификации, авторизации и учета предложена для предоставления обобщенной модели для функций AAA в проекте ODL.

3. **Service Abstraction Layer (SAL):** Являясь сердцем ODL, SAL позволяет ODL поддерживать несколько протоколов SB (через плагины SB) и предоставлять единый набор служб для других модулей и сетевых приложений. Служба обнаружения устройств предоставляется SAL и используется Диспетчером топологии для формирования топологии сети и создания функциональных возможностей элементов.

Подавляющее число служб SAL построены на основе функций плагинов SB.

Southbound Interface and Protocols Plugins

Протоколы SB используется для обеспечения безопасности между контроллером и элементами сети. Сетевые элементы могут управляться, настраиваться и контролироваться с помощью данных протоколов. ODL поддерживает несколько протоколов SB (через плагины SB). Эти протоколы SB позволяют ODL поддерживать гетерогенные сети (соединяют персональные компьютеры и другие устройства с различными операционными системами или протоколами передачи данных) и обеспечивать взаимодействие с другими технологиями и между другими поставщиками. Рассмотрим некоторые протоколы:

- **OpenFlow Plugin** реализует спецификации протокола OF по мере его развития.
- **Open vSwtich Database (OVSDB) Plugin** управляет и настраивает open vswitches.
- **Simple Network Management Protocol (SNMP) Plugin** – SB плагин для управления off-the-shelf commodity Ethernet коммутаторами. Конфигурация потока на этих коммутаторах может быть реализована через таблицу пересылки, таблицу ACL и таблицу VLAN
- **BGP-LS/PCEP Plugins** реализует Border Gateway Protocol на основе Java и Path Computation Element Protocol. Плагин BGP-LS рассматривается как источник информации о топологии L3 для ODL, в то время как плагин PCCEP используется для создания путей к базовой сети.
- **Network Configuration Protocol (NETCONF) Plugin** разработан, чтобы позволить ODL управлять и конфигурировать сетевые

элементы, поддерживающие NETCONF protocol. Он также помогает обнаруживать такие элементы, их возможности и обеспечивает все функции протокола NETCONF.

The Network Applications and Services

На верхний уровень ODL выходят сетевые приложения и сервисы, которые контролируют, управляют, контролируют всю сеть. Большинство этих приложений и сервисов связаны с соответствующими сетевыми сервисами платформы, такими как координатор VTN и менеджер VTN. Этот уровень также включает в себя сервисы оркестровки, которые проектируют трафик в соответствии с требованиями NFV и облака.

Рассмотрим некоторые сетевые приложения архитектуры ODL:

- **openDayLight User eXperience (DLUX)** – пользовательский веб-интерфейс
- **VTN coordinator** – внешнее приложение, предоставляющее пользователям REST APIs для создания VTN и координирующее расширение виртуальных сетей через несколько контроллеров ODL
- **SDNi Wrapper** (Интерфейс оболочки программно-конфигурируемых сетей) – часть приложения ODL-SDNi для обеспечения коммуникаций между контроллерами SDN. Для сбора информации, которая будет распределена между контроллерами, используется SDNi REST API. API REST SDNi извлекает обобщенную информацию из агрегатора SDNi, которую будет обрабатывать SDNi Wrapper.
- **Защита от DDoS –** функция для обнаружения и смягчения распределённых атак типа «отказ в обслуживании». Функция использует NB REST API для мониторинга поведения

защищенного трафика и перенаправления трафика атакующих к Системам смягчения атак (Attack Mitigation Systems - AMSs).

Служба кластеризации в OpenDaylight

ODL поддерживает службу кластеризации, в которой несколько экземпляров ODL действуют как один логический контроллер. Кластеризация не только обеспечивает избыточность, но также обеспечивает линейное масштабирование экземпляров ODL. Каждый экземпляр связывается друг с другом, образуя кластер, и обменивается информацией друг с другом, чтобы обеспечить доступность в случае сбоя любого экземпляра.

На рисунке 4 изображен кластер из 3 экземпляров ODL, где MD-SAL в одном из этих экземпляров подключается к хранилищу данных кластера и шине сообщений (для маршрутизации запросов на обслуживание между экземплярами). Ниже приведена информация, которая должна быть синхронизирована между экземплярами:

- Файлы конфигурации
- База данных топологии
- База данных пользователей
- Master controller для данного элемента сети.

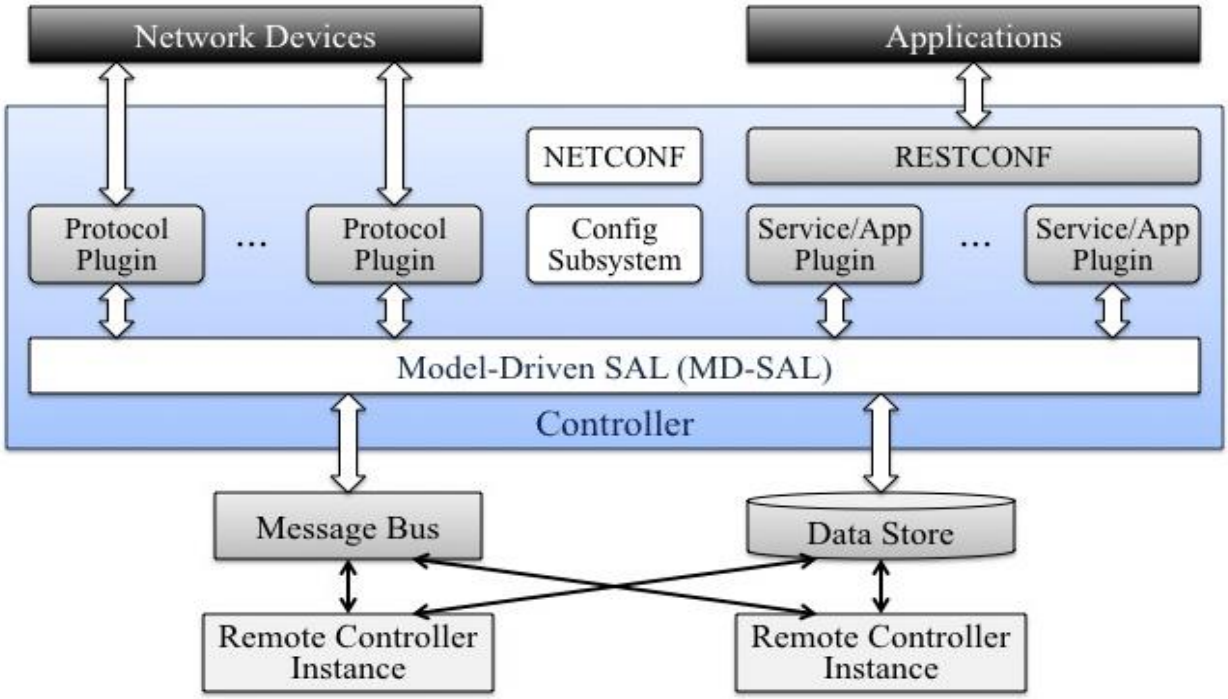

Рисунок 4 - Служба кластеризации в OpenDaylight

С помощью службы кластеризации ODL сетевые элементы и приложения могут подключаться к нескольким экземплярам. Коммутаторы подключаются к двум или более экземплярам через постоянное соединение point-to-point TCP/IP (в случае сбоя одного экземпляра доступен другой). приложения подключаются к экземпляру через (HTTP, основанный на непостоянных соединениях) веб-сервисы RESTful, что означает, что в случае сбоя экземпляра приложение восстановит новое соединение при следующей транзакции.

Заключение

SDN может оказать значительное влияние на различные типы сетей, а также сформировать сети следующего поколения. Контроллер SDN не должен ограничиваться OpenFlow или каким-либо отдельным протоколом SB, взаимодействующим с элементами сети, а API-интерфейсы протоколов SB или приложений NB должны создаваться автоматически для предоставления или запроса услуг. Архитектура ODL позволяет адаптировать любые загруженные плагины SB или NB. Эта способность является результатом развитой MD-SAL

в ODL, которая не имеет каких-либо специфичных для плагина API, а также API SB и API NB, автоматически генерируемых из моделей плагинов. Несколько экземпляров контроллера ODL могут формировать кластер для обеспечения избыточности, высокой доступности и масштабируемости.

Список использованных источников

1. https://www.opendaylight.org/
2. https://en.wikipedia.org/wiki/OpenDaylight_Project
3. https://wiki-archive.opendaylight.org/view/Main_Page
4. https://www.researchgate.net/publication/317057083_Facilitation_of_The_OpenDaylight_Architecture
5. https://www.researchgate.net/publication/319547264_Open_Daylight_as_a_Controller_for_Software_Defined_Networking