МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ»

Кафедра Электроники и робототехники

«ДОПУЩЕН К ЗАЩИТЕ»

Зав. кафедрой _____
(ученая степень, звание, Ф.И.О.)

_____ «____» _____ 201__ г.
(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Исследование технологий локальных сетей

Специальность 5В071600

Выполнил (а) Бижанов Даурен (Ф.И.О.) Группа ПС 15-3

Научный руководитель доцент, к.т.н., Шабельников Е.А.
(ученая степень, звание, Ф.И.О.)

Консультанты:

по экономической части:

к.э.н., доцент Бекишева А.Д.
(ученая степень, звание, Ф.И.О.)

_____ «16» 03 2019 г.
(подпись)

по безопасности жизнедеятельности:

д.х.н., проф. Трихлебов Н.Г.
(ученая степень, звание, Ф.И.О.)

_____ «15» 05 2019 г.
(подпись)

Нормоконтролер: Ахтамбаев Г.О.
(ученая степень, звание, Ф.И.О.)

_____ «3» 06 2019 г.
(подпись)

Рецензент: _____
(ученая степень, звание, Ф.И.О.)

_____ «____» _____ 201___ г.
(подпись)

Алматы 2019

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ»

Институт космической инженерии и телекоммуникаций

Кафедра Электроники и робототехники

Специальность 5В071600 - Приборостроение

ЗАДАНИЕ
на выполнение дипломного проекта

Студенту Бикенову Даурену Ермануловичу
(Ф.И.О.)

Тема проекта Исследование технологий мобильных сетей

Утверждена приказом по университету №124 от «26» октября 2018 г.

Срок сдачи законченного проекта «10» июня 2019 г.

Исходные данные к проекту (требуемые параметры результатов исследования (проектирования) и исходные данные объекта): Разработать и смоделировать сегмент компьютерной сети согласно концепции SDN, а также провести тестирование внедрения механизмов обеспечения информационной безопасности.

Перечень вопросов, подлежащих разработке в дипломном проекте, или краткое содержание дипломного проекта: 1. изучить архитектуру, основные компоненты и особенности сетей SDN; 2. исследовать проблемы безопасной передачи информации в сетях SDN; 3. разработать и смоделировать сегмент компьютерной сети согласно концепции SDN; 4. провести тестирование внедрения известных механизмов обеспечения информационной безопасности.

Перечень графического материала (с точным указанием обязательных чертежей): Отличие классической сетевой архитектуры от SDN, архитектура сетевого устройства, разделение функций управления и пересылки, трехуровневая архитектура SDN, Southbound и Northbound интерфейсы, выделение потока трафика, на основе общих признаков, связь коммутатора и контроллера по OpenFlow, компоненты OpenFlow коммутатора, формат записей в таблице сетевых протоколов OpenFlow коммутатора, формат заголовка FT.

Основная рекомендуемая литература:

1. Шаньгин А.В. Технологии SDN/OpenFlow
2. Семенеский Р.А. Технологии SDN и NFV: новые возможности для телекоммуникаций
3. Курочкин И.И., Гуменный О.Г. Безопасность сетей SDN. Классификация атак.
4. Борисов М.А., Заводцев И.В., Чижов И.В. Основы программно-аппаратной защиты информации.

Консультации по проекту с указанием относящихся к ним разделов проекта

Раздел	Консультант	Сроки	Подпись
БЖД	Трихович И.Г.	13.02–15.05.19	
Экономич. часть	Бекешева А.Ч.	08.04–15.05.19	
Архитектура SDN	Шабельников Е.А.	05.02.2019	
OpenFlow коммутатор	Шабельников Е.А	10.02.2019	
Механизмы обеспеч. безоп. SDN	Шабельников Е.А.	15.02.2019	
Контроллер OpenDayLight	Шабельников Е.А.	02.03.2019	
Разработка сети SDN	Шабельников Е.А.	25.04.2019	
Демонстрация работосп-и	Шабельников Е.А.	10.05.2019	

График
подготовки дипломного проекта

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечание
1. Описание принципов работы SDN сетей	03.01.2019 - 05.02.2019	
2. Информационная безопасность сетей SDN	10.02.2019 - 25.02.2019	
3. Разработка и моделирование сегмента сети SDN	02.03.2019 - 20.03.2019	
4. Внедрение механизмов защиты в данном сегменте	23.03.2019 - 25.04.2019	
5. Анализ полученных в работе результатов, обобщение	26.04.2019 - 11.05.2019	
6. Технико-экономическое обоснование проекта	15.05.2019	
7. Безопасность жизнедеятельности	16.05.2019	

Дата выдачи задания «03» января 2019 г.

Заведующий кафедрой _____ (Ситников Г.О.)
(подпись) (Ф.И.О.)

Научный руководитель проекта _____ (Шабельников Е.А.)
(подпись) (Ф.И.О.)

Задание принял к исполнению студент _____ (Бикмаев Р.Е.)
(подпись) (Ф.И.О.)

Аннотация

В данной дипломной работе исследована концепция программно-конфигурируемых сетей (SDN). Проведен анализ архитектуры сети SDN и сравнение с традиционной сетевой архитектурой. Исследованы аспекты обеспечения информационной безопасности в сетях SDN и способы их внедрения. Произведена разработка и моделирование сегмента сети SDN на базе платформы виртуализации Oracle VM VirtualBox. В ходе моделирования рассмотрен функционал контроллера SDN и внедрен механизм обеспечения безопасного канала связи. Результаты разработки и моделирования подтверждают актуальность применения программно-конфигурируемых сетей и возможность обеспечения их надежной и безопасной эксплуатации.

Annotation

In this thesis work investigated the concept of software-configured networks (SDN). An analysis of the SDN network architecture and comparison with the traditional network architecture is carried out. The aspects of ensuring information security in SDN networks and the ways of their implementation are investigated. The development and modeling of the SDN network segment based on the Oracle VM VirtualBox virtualization platform has been carried out. During the simulation, the functionality of the SDN controller was considered and the mechanism for providing a secure communication channel was implemented. The results of development and modeling confirm the relevance of the use of software-configured networks and the possibility of ensuring their reliable and safe operation.

Аңдатпа

Диссертациялық жұмыста бағдарламалық жасақтама конфигурацияланған желілер (SDN) тұжырымдамасы зерттелді. SDN желісінің архитектурасын талдау және дәстүрлі желі архитектурасымен салыстыру жүргізілді. SDN желілеріндегі ақпараттық қауіпсіздікті қамтамасыз ету аспектілері және оларды жүзеге асыру жолдары зерттелді. Oracle VM VirtualBox виртуалдандыру платформасына негізделген SDN желісінің сегментін жасау және модельдеу жүргізілді. Модельдеу кезінде SDN контроллерінің функционалдылығы қарастырылды және қауіпсіз байланыс арнасын ұсыну тетігі енгізілді. Даму және модельдеудің нәтижелері бағдарламалық қамтамасыз етудің конфигурацияланған желілерін пайдаланудың өзектілігін және олардың сенімді және қауіпсіз жұмыс істеуін қамтамасыз ету мүмкіндігін растайды.

Содержание

Введение

Появление новых технологий и различных сервисов в сфере ИТ таких как: виртуализация серверов, облачные платформы, большое количество сервисов хранения и центров обработки данных, постоянно модифицирующихся мобильных сетей, решений на базе BYOD, а также увеличение нагрузки на сети и рост объёмов пользовательского трафика, приводит к снижению эффективности применения традиционных сетей [1-3].

Традиционная сетевая архитектура стала более ресурсоемкой, сложной для управления и масштабирования, а также требует большого количества денежных средств на эксплуатацию и модернизацию. Кроме того, зависимость от закрытого программного обеспечения и поставщиков оборудования значительно усложняет использование мультивендорного оборудования в рамках одной сети, и выбор наиболее верных архитектурных решений. Созданную и запущенную в промышленную эксплуатацию сетевую инфраструктуру сложно развивать, а внедрение новых версий существующих протоколов или новых протоколов и сервисов, становится трудно реализуемой задачей [4].

Существующие сетевые архитектуры перестают соответствовать требованиям рынка и возможностям современных сетей. Это стало причиной появления принципиально нового подхода к построению сетей. В 2006 г специалистами университетов Стэнфорда и Беркли была предложена концепция «Программно-Конфигурируемых Сетей» – ПКС (Softwaredefined Networking – SDN).

В рамках концепции SDN была разработана динамичная, легко масштабируемая, гибкая в управлении сетевая архитектура, главная особенность которой заключается в наличии централизованного управляющего устройства – контроллера SDN, и разделении плоскости управления и плоскости передачи данных. А также нового подхода к программированию сети, и возможности динамического управления сетевыми элементами через открытые интерфейсы и приложения, как на реальном, так и на виртуальном уровне. Целью этой архитектуры является гибкость, автоматизация, и снижение общих издержек сети.

С точки зрения обеспечения безопасности сети SDN, существуют особенности, связанные с архитектурным подходом к разделению плоскостей и выделению централизованного управляющего устройства.

Контроллер как ключевой компонент в управлении всей инфраструктурой SDN является наиболее уязвимым элементом, и атака на него может повлечь критичные для всей инфраструктуры последствия.

Основными угрозами, возникающими со стороны сетевых устройств, работающих по принципу программно-конфигурируемой сети, остаются вариации таких атак, как «отказ в обслуживании», подмена контроллера и т.д.

На сегодняшний день концепция SDN всё больше набирает популярность и постепенно внедряется в сетевую инфраструктуру. Но

технология постоянно развивается и имеет множество интерпретаций. Так как, по многим аспектам, в том числе безопасности SDN, полностью или частично отсутствует подробная документация, представляется интересным его актуальное исследование.

Исследование механизмов защиты сетей SDN, таких как: TLS/SSL, IDS, репликация контроллера, и возможности их внедрения для обеспечения безопасности информационного обмена в сетях SDN.

При написании данной работы были поставлены следующие задачи:

– изучить архитектуру, основные компоненты и особенности сетей на основе концепции SDN;

– исследовать проблемы, связанные с защитой передачи информации в сетях SDN;

– разработать и смоделировать сегмент компьютерной сети согласно концепции SDN;

– провести тестирование внедрения известных механизмов обеспечения информационной безопасности.

Объектом исследования данной работы является организация безопасного обмена информацией в сетях SDN.

Предметом исследования является реализация механизмов защиты передачи данных в сетях SDN.

В ходе исследований применялись следующие методы: сравнение, анализ, классификация, моделирование.

Результаты исследования могут быть использованы при внедрении технологии SDN как в существующие, так и в новые сегменты компьютерных сетей. Эксплуатации контроллеров SDN в центрах обработки данных и SD-WAN и обеспечении безопасности сетей SDN. А также, для дальнейших исследований в этой области, в рамках учебного курса и лабораторных работ.

Работа состоит из введения, пяти глав, заключения, списка литературы и приложения.

В первой главе рассматривается архитектура сетей SDN, основные особенности и компоненты, а также область применения SDN.

Вторая глава посвящена классификации актуальных атак и угроз информационной безопасности в сети SDN и исследованию механизмов защиты, вводятся основные понятия и определения.

Третья глава посвящена разработке сегмента SDN сети, и проведению эксперимента по внедрению механизмов защиты TLS/SSL на данном сегменте.

В четвертой главе проведено технико-экономическое обоснование проекта.

Пятая глава посвящена комплексу мероприятий, направленных на обеспечении безопасности работников на рабочем месте – безопасность жизнедеятельности.

1 Описание принципов работы SDN сетей

Главная особенность заключается в наличии централизованного управляющего устройства – контроллера SDN. В основе данного подхода лежит принцип физического разделения плоскости управления и плоскости передачи данных, передача функций маршрутизации контроллеру сети, и реализации на основе этого принципа легко масштабируемой, быстро и гибко настраиваемой виртуальной сети. Отличие классической архитектуры от SDN представлено на рисунке 1 [2].

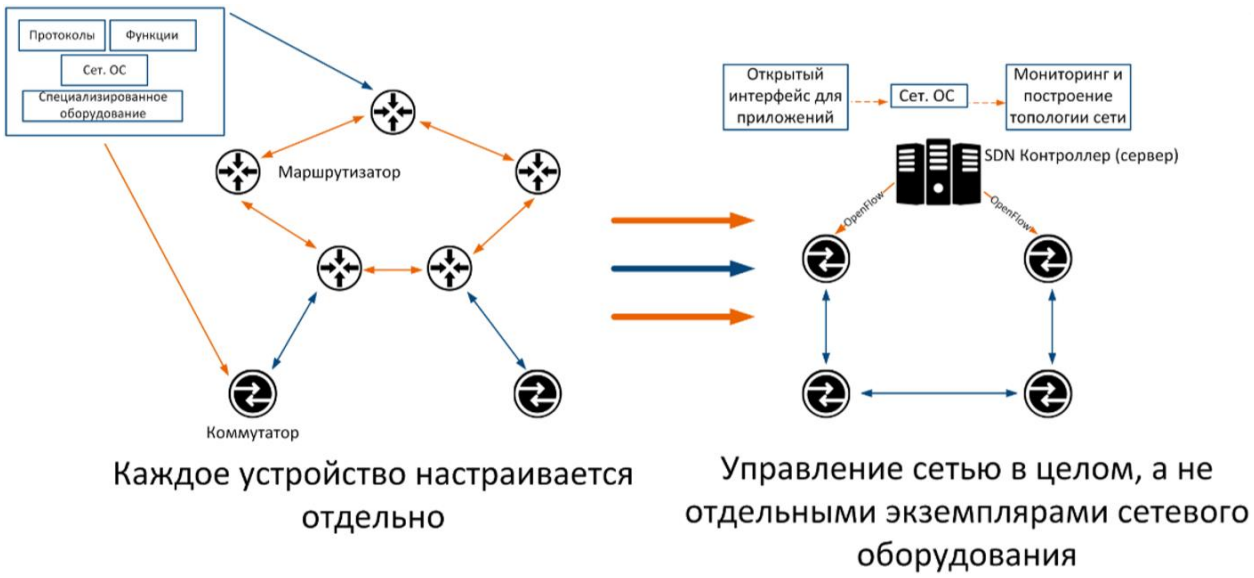

Рисунок 1 – Отличие классической сетевой архитектуры от SDN

Ещё одной ключевой особенностью технологии SDN является объединение в отдельные классы (потоки) передаваемых по сети данных, и применение различной логики управления для каждого потока. Такой подход актуален в применении к виртуализованным сетям, и открывает новые возможности в этой области.

1.1 Архитектура SDN

В основном, в сетях с традиционной архитектурой, сетевые устройства имеют три друг от друга независимых уровня. Уровни представлены на рисунке 2.

Рисунок 2 – Архитектура сетевого устройства

Уровни сетевых устройств:

– Data plane – отвечает за пересылку Protocol Data Unit (PDU) IP-пакеты, фреймы, сегменты и т.д. по определенным правилам, хранящимся в таблицах пересылок(маршрутизации/коммутации/меток). Приходящие пакеты должны быть максимально быстро переданы на порт назначения. На этом уровне выполняется много однообразных операций – это требует значительных вычислительных мощностей;

– Control plane – формирует условия для работы Data plane и обеспечения форвардинга. Обеспечивает логику для пересылки пакетов (заполнение таблиц маршрутизации, отработку различных служебных протоколов ARP/STP/и пр.). В отличие от data plane, имеет достаточно сложную логику работы и не требует большого кол-ва операций;

– Management Plane (может не выделяться в отдельную плоскость, сливаясь с плоскостью контроля) – предоставляет интерфейс управления и мониторинга отвечает за конфигурацию и общее состояние узла. В том числе следит за физическими параметрами, такими как температура, электропитание, вентиляция, работоспособность плат и модулей.

Разделение функций управления и пересылки показаны на рисунке 3.

Архитектура SDN основывается на логическом и физическом разделении функций уровней сетевых устройств – отделении Control Plane от Data Plane и переносе плоскости управления сетью на выделенный SDN контроллер, который имеет точную информацию о структуре и топологии сети. Таким образом, в рамках классической концепции SDN сетевое устройство – передающее устройство без функций управления [5].

SDN состоит из трех уровней:

– уровень инфраструктуры;

– уровень управления;
– уровень приложений.

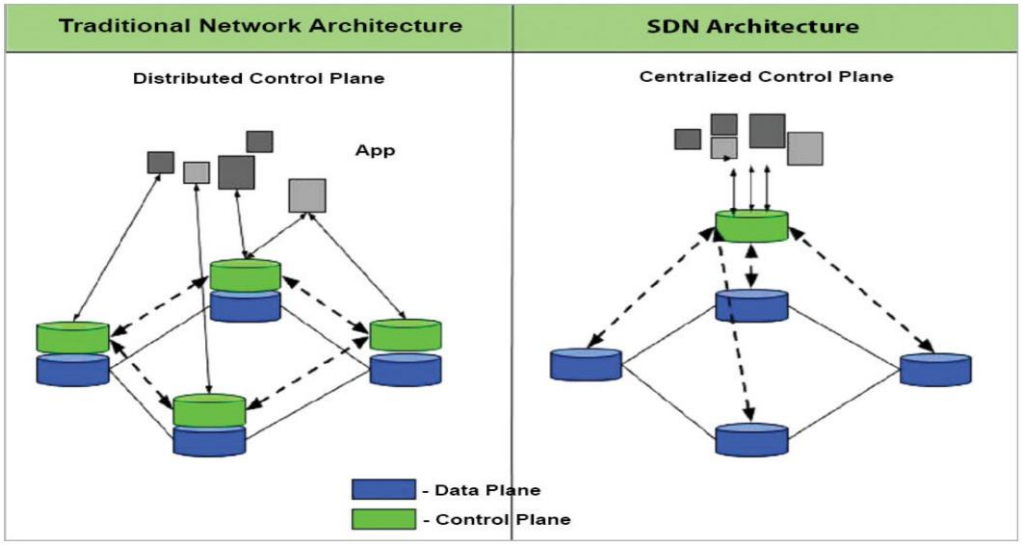

Рисунок 3 – Разделение функций управления и пересылки

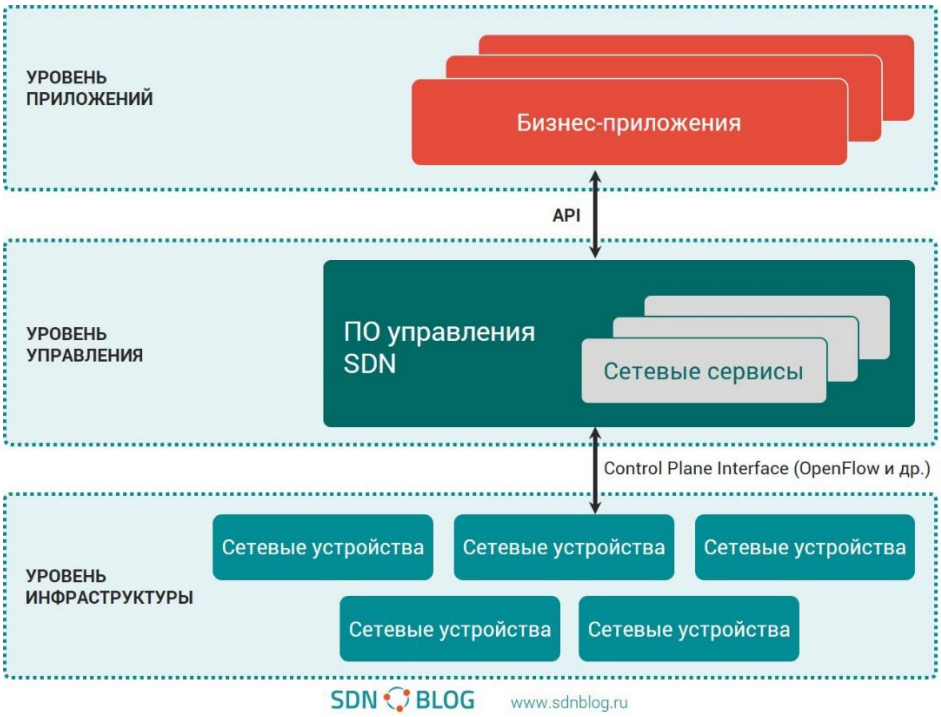

Рисунок 4 – Трехуровневая архитектура SDN

Трехуровненвая архитектура SDN представлена на рисунке 4.

Описание уровней SDN:

а) уровень инфраструктуры или передачи данных (data plane) – включает в себя набор передающих сетевых устройств, которые доступны через унифицированный интерфейс и выполняют инструкции таблиц потоков для передачи сетевого трафика. Это могут быть как физические, так и виртуальные коммутаторы, и маршрутизаторы, а также каналы связи [6-7];

б) уровень управления (control plane) «контрольная плоскость» – регулирует обмен информацией о таблицах маршрутизации и вырабатывает правила маршрутизации на основе заложенных в контроллер алгоритмов анализа потоков пакетов, пересылаемых от коммутаторов. Реализуется на самом контроллере SDN или кластере таких контроллеров, обеспечивающих функции управления сетевой инфраструктурой.

На данном уровне есть три интерфейса для взаимодействия с другими элементами сетевой инфраструктуры. С помощью этих интерфейсов уровень управления связывает все три плоскости [5-7]:

– Southbound API – (нисходящий интерфейс для уровня передачи данных) обеспечивает взаимодействие SDN контроллера с сетевыми устройствами посредством специальных протоколов, например, OpenFlow. Это позволяет контроллеру динамически вносить изменения и конфигурировать сеть в реальном времени. Интерфейс способен поддерживать несколько протоколов (в виде отдельных плагинов), таких как OpenFlow 1.0, OpenFlow 1.3 и BGPLS;

– Northbound API – (для уровня приложений) интерпретирует бизнес логику в сетевые инструкции, предоставляет основные сетевые функции, такие как вычисление пути данных, маршрутизация и безопасность, позволяет гибко выделять сетевые ресурсы, исходя из требований приложений, абстрагируя сетевую инфраструктуру. Используется для автоматизации и управления данными. Данный интерфейс еще не стандартизирован, Большинство контроллеров в качестве интерфейса восходящего взаимодействия поддерживают RESTful API. API-интерфейсы Northbound могут использовать Python, Java, C, REST, XML, JSON.

Интерфейс NorthBound (NBI) можно назвать границей между контроллером и прикладным уровнем. Он поддерживает большинство протоколов уровня приложения для взаимодействия, т. е. HTTPS, SFTP и т. д. Мы можем управлять контроллером с помощью основных HTTP методов POST, GET, PUT и DELETE;

– восточный/западный интерфейс (для взаимодействия между элементами уровня управления). Отслеживает топологию всей сети и предоставляет программный интерфейс (API) для сетевых приложений;

в) уровень приложений состоит из высокоуровневых приложений, которые взаимодействуют с сетевым контроллером или набором контроллеров, запрашивая и передавая необходимые ресурсы и инструкции посредством API для реализации конкретных функций, отвечающих требованиям сетевой инфраструктуры. Это взаимодействие осуществляется с помощью такого компонента SDN как Northbound API. Здесь реализуются различные функции обработки сетевого трафика. Примером таких приложений могут служить средства мониторинга (сбор информации о сети и отправка её SDN приложениям), аналитики балансировки нагрузки или бизнес-приложения [6].

Southbound и Northbound интерфейсы представлены на рисунке 5.

Отделение управления от функций форвардинга и рассмотрение каждого уровня как самостоятельной единицы предоставляет приложениям более подробную информацию от контроллера о состоянии сети, по сравнению с традиционными сетями.

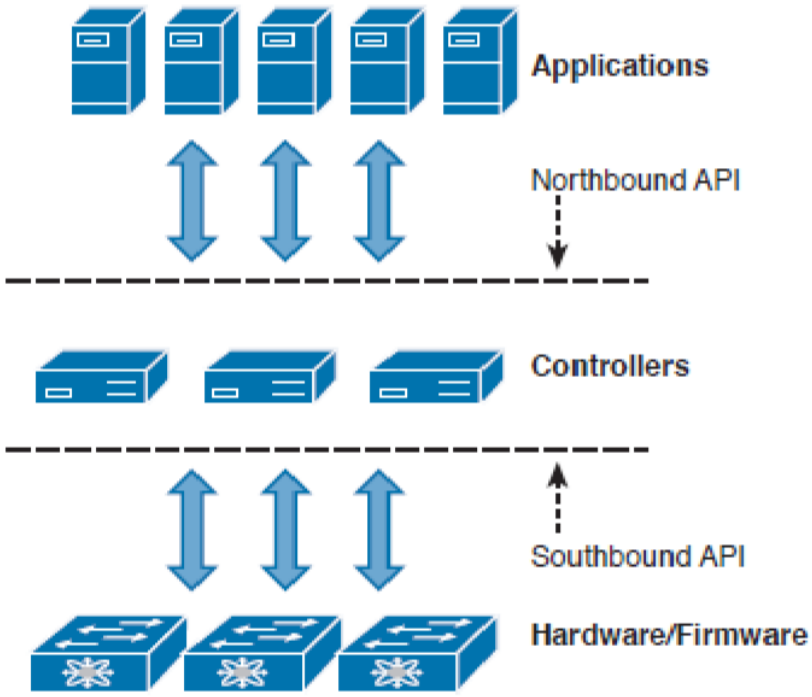

Рисунок 5 – Southbound и Northbound интерфейсы

1.2 Контроллер OpenFlow и его функции

SDN Контроллер – управляющий центр сети, включающий центральную сетевую операционную систему и управляющие приложения. Отвечает за построение и отображение топологии, программирование сетевых устройств и служит единой точкой управления всей сетью. SDN контроллер предоставляет возможность объединять несколько контроллеров, и устанавливать иерархические связи между ними [5-6]. SDN контроллер реализуется в качестве ПО для управления плоскостью пересылки.

Платформа контроллера содержит набор динамически подключаемых модулей для выполнения необходимых сетевых задач с помощью программирования функций сетевой инфраструктуры. Это является фундаментальным свойством архитектуры SDN и её основным преимуществом. Разделение функциональных особенностей приложений позволяет осуществлять несколько задач в сети одновременно.

Основные функции контроллера представлены в таблице 1.

Таблица 1 – Основные функции контроллера

Управление ресурсами сервера	– поддержка многопоточности; – мониторинг загрузки сервера.
Обеспечение связи между уровнем приложений и сетью	– взаимодействие приложения-коммутатор; – динамическая конфигурация сети в зависимости от параметров трафика;
Предоставление сервисов	– построение топологии; – мониторинг хостов; – добавление/удаление новых элементов сети.
Управление коммутацией	– программирование flow table и правил пересылки трафика.
Создание приложений на основе API	– одновременный запуск нескольких приложений; – регистрация приложений; – обработка ошибок приложений и ошибок взаимодействия приложений; – подписка на события от приложений; – приоритизация приложений; – обеспечение взаимодействия между приложениями посредством соответствующей инфраструктуры; – балансировка нагрузки приложений (по потокам); – разграничение прав доступа приложений к элементам сети.

Открытый программный интерфейс API (от англ. Application Programming Interface), позволяет разрабатывать приложения и программные расширения, которые реализуют логику, необходимую, для определения, обновления и адаптации правил потоков.

Поток пакетов – последовательность пакетов, проходящих через сеть, с общим набором значений полей заголовка, представленных на рисунке 6.

Управление можно осуществлять как для индивидуальных потоков, так и для сгруппированных по определенным признакам: источнике, назначении, приложении или любой их комбинации. Это предоставляет широкие возможности для QoS или гарантированного обслуживания для определенных приложений [7].

Рисунок 6 – Выделение потока трафика, на основе общих признаков

Контроллер статически или динамически программирует и устанавливает правила соответствия для потоков. Правила потока определяют базовые инструкции, которые регулируют пересылку, изменение или удаление каждого пакета, который попадает в SDN-коммутатор [4-6]. Установка правил передачи пакетов контроллером для новых потоков может производиться двумя способами [6-8]:

Реактивный – устройства передачи запрашивают контроллер, а контроллер формирует одно правило и устанавливает его на устройство передачи, отправившее запрос.

Проактивный – контроллер передает политики по иерархии устройств передачи таким образом, что необходимость в контроллере для обработки новых потоков почти не возникает.

В зависимости от режима управления, топология с централизованным управлением может иметь различные уязвимости безопасности, которые будут рассмотрены во второй главе.

На сегодняшний день на рынке представлено множество SDN контроллеров таких как: OpenDayLight ONOS Floodlight Runos и др. Но все они должны отвечать основным требованиям (табл.2), сформировавшимся в ходе разработки и внедрения контроллеров в эксплуатацию.

Из всего вышесказанного следует, что контроллер выступает в роли элемента реагирования: получает сообщения от коммутаторов по каналам управления и вырабатывает отклики, которые изменяют содержимое таблиц коммутации. За счёт логически централизованного интеллекта в программном SDN контроллере, сеть для приложений и политик становится как единый логический коммутатор.

Основные требования к контроллеру представлены в таблице 2.

Таблица 2 – Основные требования к контроллеру

Гибкость	Масштабируемость	Временная расширяемость	Производительность
– совместимость API с использованием общих структур и моделей программирования; – системная интеграция и объединение приложений на организованные рабочие процессы более высокого уровня	– архитектура должна позволять создавать плагины независимо друг от друга и инфраструктуры контроллера, – минимальное время системной интеграции поддержка кластеризации	– инфраструктура контроллера адаптивна к схемам данных (моделям), из динамически загружаемых плагинов или устройств	– устойчивость к различным видам нагрузки – для поддержки разработки SDN приложений обеспечивать связь со средой разработки приложений

1.3 Протоколы SDN. OpenFlow

Два наиболее известных протокола, которые используются контроллерами SDN для связи с коммутаторами и маршрутизаторами, это OpenFlow и OVSDB.

Протокол OpenFlow – предназначен для централизованного управления сетями, обеспечивает прямой доступ контроллера к управлению сетью. Определяет организацию обмена сообщениями об изменениях таблиц переадресации [7], [9-10]. OpenFlow – единственный стандартизованный протокол SDN. Он представляет собой «opensource» проект – все детали внутреннего алгоритма доступны для разработчиков. Фонд Open Networking занимается разработкой конфигураций, тестированием и стандартизацией Open Flow через технические рабочие группы, помогая обеспечить взаимодействие между сетевыми устройствами и программным обеспечением управления от разных поставщиков.

Сетевые устройства могут поддерживать переадресацию на основе OpenFlow, а также традиционную переадресацию – это обеспечивает гибкость протокола для постепенного внедрения SDN в существующие инфраструктуры. Поддержка OpenFlow зачастую реализуется с помощью обновления программного обеспечения.

Для взаимодействия между сетевыми устройствами с помощью OpenFlow поддерживается три типа сообщений [9-10].

Основные типы сообщений OpenFlow представлены в таблице 3.

Таблица 3 – Основные типы сообщений OpenFlow

Тип сообщения	Назначение
Контроллер-коммутатор	– конфигурация коммутатора – управление и контроль состояния (потеря канала, ошибки соединения) – управления таблицами протоколов
Симметричные сообщения	– отправка в обоих направлениях – обнаружение проблем соединения контроллера с коммутатором … Hello, Echo
Ассиметричные сообщения	– отправка от коммутатора к контроллеру – объявляют об изменении состояния сети и коммутаторов Packet-in, flow-removed, port-status, error

Протокол OpenFlow реализуется с обеих сторон интерфейса между устройствами сетевой инфраструктуры и программным обеспечением управления SDN. Связь коммутатора и контроллера по OpenFlow представлена на рисунке 7. Для установления соединения каждая из сторон

высылает OpenFlow-сообщение OFPT_HELLO с максимальным номером версии протокола, которая поддерживается отправителем. Если данная версия поддерживается контроллером, взаимодействие продолжается, иначе коммутатор получает сообщение об ошибке, и дальнейшая настройка становится невозможной [9], [11].

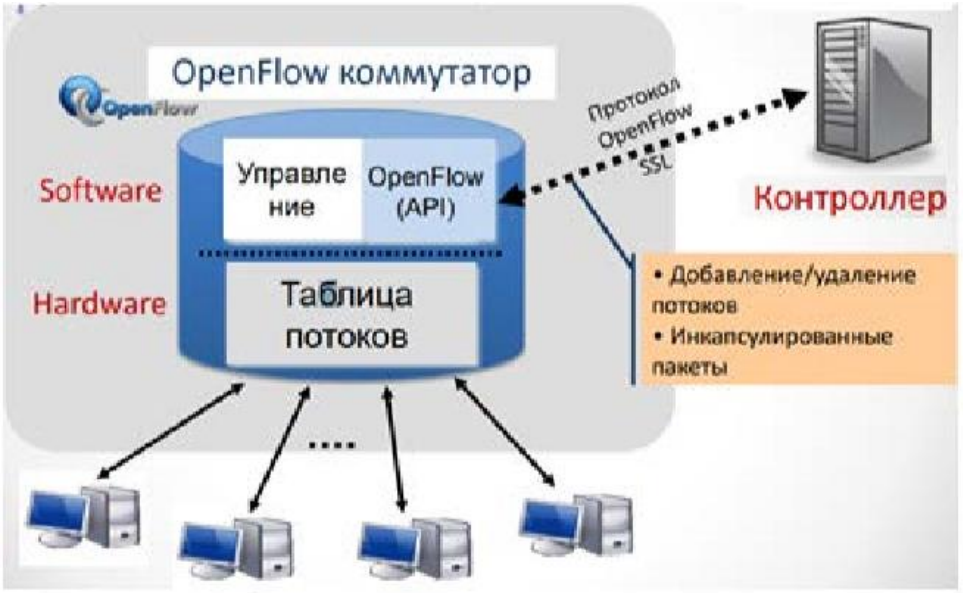

Рисунок 7 – Связь коммутатора и контроллера по OpenFlow

Для идентификации соединения используется уникальный URI соединения – информация о соединении – адрес и порт. URI должен соответствовать синтаксису URI, определенному в RFC 3986, в частности, в отношении кодировки символов.

Форма записи URI:

tcp: 198.168.10.98: 6653
tls: test.opennetworking.org: 6653
ssl: [3ffe: 2a00: 100: 7031 :: 1]: 6653

Поле протокола определяет транспортный протокол. Согласно спецификации, OpenFlow для безопасного соединения рекомендуется использовать протокол TLS (Transport Layer Security) [7]. Коммутатор и контроллер могут связываться через TLS-соединение для обеспечения аутентификации и шифрования соединения. Они взаимно аутентифицируются путем обмена сертификатами, подписанными конкретными сайтами закрытым ключом (рекомендуется). Протокол OpenFlow не реализует безопасность сам по себе, поэтому транспортный протокол должен обеспечивать безопасность, если это необходимо. Коммутатор и контроллер могут дополнительно связываться с использованием TCP соединения для незашифрованного сообщения (не рекомендуется) или для реализации альтернативных

механизмов безопасности таких как IPsec, VPN или отдельная физическая сеть.

Поле имени или адреса – имя хоста или IP-адрес контроллера. IPv6 адрес должен быть заключен в квадратные скобки, как рекомендовано в RFC 2732.

Поле порта – транспортный порт, используемый на контроллере. Если поле порта не указано, оно должно быть эквивалентно установке этого поля в стандартный порт OpenFlow 6633. Если тип транспорта для соединения OpenFlow не определен этой спецификации, должен использоваться протокол, который правильно идентифицирует этот тип транспорта. В этом случае остальная часть URI соединения определяется протоколом.

1.4 OpenFlow коммутатор

OpenFlow коммутатор – сетевое устройство Data Plane, поддерживающее работу по протоколу OpenFlow и выполняющее функции перенаправления данных согласно инструкциям SDN контроллера. Задача – передача и обработка пакетов в соответствии с таблицей сетевых потоков и правил, сформированной контроллером на основании определенных критериев кадра или пакета. Устройства лишены функции управления и могут только передавать статистику и обращаться за новыми правилами потоков к внешнему SDN-контроллеру [7].

Согласно спецификации стандарта, OpenFlow – каждый коммутатор должен содержать одну или более таблиц потоков (flow tables), групповую таблицу (group table) и поддерживать канал (OpenFlow channel) для связи с удаленным контроллером по протоколу OpenFlow [5]. Компоненты OpenFlow коммутатора представлены на рисунке 8 [7], [9-10].

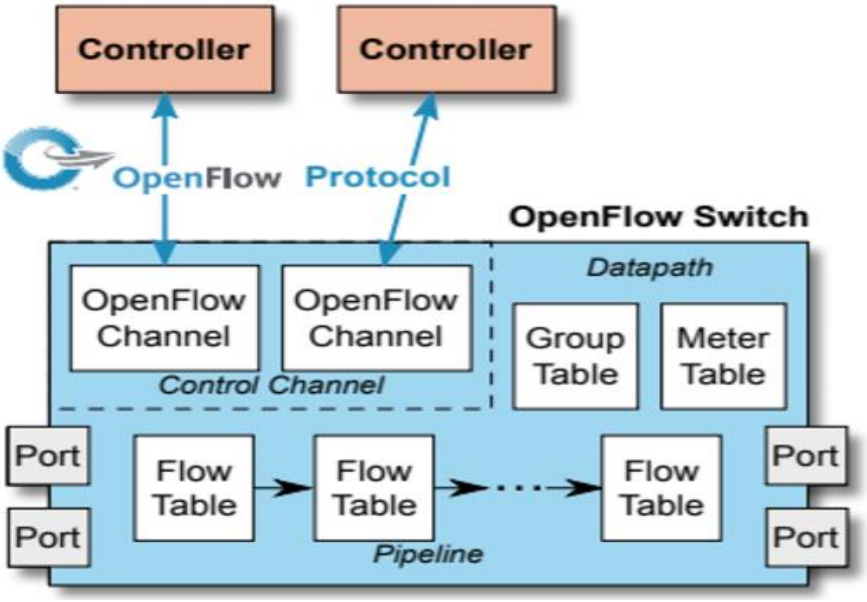

Рисунок 8 – Компоненты OpenFlow коммутатора

Канал OpenFlow (OpenFlow Channel) – интерфейс, который соединяет каждый логический коммутатор OpenFlow с контроллером. Для каждого контроллера, подключаемого к коммутатору, устанавливается отдельный канал [9-10].

Выделяют два подхода к построению такого канала:

а) Out-of-band – передача сообщений протокола по физически независимому каналу (используется отдельный порт коммутатора), не используя сеть передачи данных. Обеспечивает более высокую надежность и безопасность за счет отсутствия влияния передаваемого по сети трафика на управляющий трафик и исключения физического доступа к сети управления со стороны узлов в сети передачи данных;

б) In-band – передача управляющей информации через сеть передачи данных. Менее надежный и безопасный способ. К преимуществам подхода можно отнести более высокую экономичность и сокращение количества сетевого оборудования, упрощение проектирования и сети.

Формат записей в таблице сетевых протоколов OpenFlow коммутатора представлен на рисунке 9.

Формат заголовка Flow table представлен на рисунке 10.

OpenFlow порт – сетевой интерфейс для передачи пакетов между устройствами с поддержкой OpenFlow. Выполняют функции портов стандартного L2-коммутатора. Пакеты, приходящие на OpenFlow-порт, классифицируются по потокам в таблицах потоков с помощью Match классификаторов. Подробная структура таблицы потоков представлена на рисунке 12.

Сравнение(match fields	Приоритет (priority)	Счетчики(counters)	Действия (Actions)	Тайм-ауты(time-outs)	Метаданные

Рисунок 9 – Формат записей в таблице сетевых протоколов OpenFlow коммутатора

Ingress port	MAC Src	MAC dstn	Eth type	Vlan ID	Vlan Priority	IP src	IP dstn	IP Pport	TCP sPort	TCP dPort	Action

Рисунок 10 – Формат заголовка Flow table

Обработка пакетов на основе таблицы представлена на рисунке 11.

Записи в таблице сетевых протоколов OpenFlow коммутатора:

– match – правило выделения пакетов, принадлежащих конкретному потоку с помощью сравнения заголовков пакета (данные L2-L4). Набор полей «match» задан в специальном формате OpenFlow Extensible Match fields (OXM);

– actions – определяет каким образом будут обработаны элементы данного потока. В спецификации OpenFlow и Open vSwitch указаны основные действия, но также есть возможность расширения [5], [7], [9-10];

– статистика – отслеживает количество принятых пакетов конкретного потока и количество байт с помощью обновления фрейм-счетчиков. Счетчики доступны для таблиц, потоков, портов и очередей;

– priority – числовое поле от 0 до 65535;

– timeouts – максимальное время до удаления соответствующей flow записи;

– метаданные – не используются при обработке пакетов. Может использоваться контроллером для фильтрации статистики потока, изменения

потока и удаления потока, используется для переноса информации от одной таблицы к другой.

Соответствующая запись определяется по полям сравнения (поле «protocol») и приоритету. При совпадении полей выбирается запись с наибольшим приоритетом. В соответствии с ней будет обрабатываться пакет [7]. Процесс обработки пакета представлен на рисунке 13.

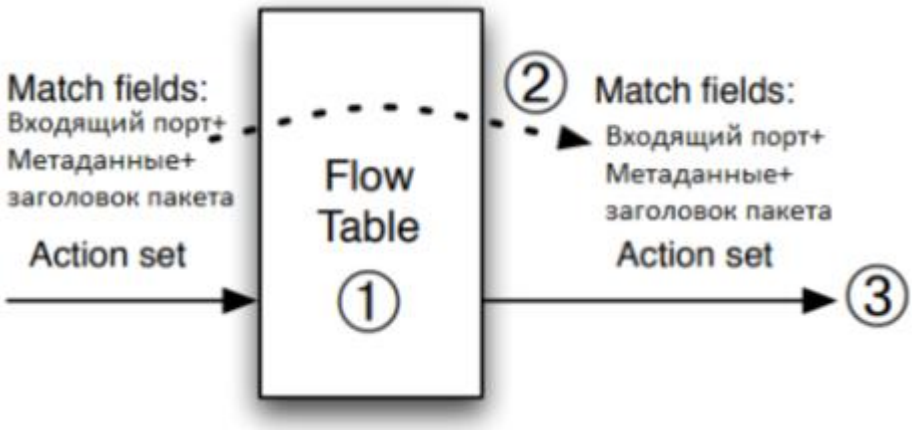

Рисунок 11 – Обработка пакетов на основе таблицы

Другим элементом записи таблицы является «действие» (action), которое сопоставляется каждой записи в таблице потоков, применяется к пакетам, принадлежащим данному потоку, и определяет обработку пакетов потока [9-10]. Основными действиями являются: пересылка на определенный порт или несколько портов, отбрасывание всех пакетов конкретного потока, преобразование пакета (модификация заголовка пакета) − инкапсуляция и передача пакетов на контроллер ПКС по безопасному каналу. Кроме того, «стандартная» обработка, позволяет разделить потоки данных на потоки, управляемые OpenFlow, и потоки, управляемые другими механизмами (существующие протоколы маршрутизации). Предоставляет возможность изолировать экспериментальный трафик, используя общую инфраструктуру.

В OpenFlow есть механизм обработки пакетов с использованием групп OpenFlow. Группа – это набор подмножеств, каждое подмножество состоит из набора Actions, которые могут быть наборами действий для широковещательной рассылки и более сложных механизмов пересылки.

Функционал GROUPS расширяет возможности пересылки пакетов. Например, позволяет реализовать ECMP Load Balancing – «отправка на одно подмножество из группы» – для балансировки нагрузки в LAG или ECMP.

Таблица групп (group table) содержит записи о группах, содержащие список контейнеров действий со специальной семантикой в зависимости от типа группы.

Механизм работы OpenFlow коммутатора достаточно прост. Начиная с версии 1.3 OpenFlow поддерживает множественные таблицы потоков. Для обработки потоков, коммутатором используется механизм конвейерной

обработки (pipeline) – обработка пришедшего пакета начинается до окончания обработки предыдущего [7], [9-10].

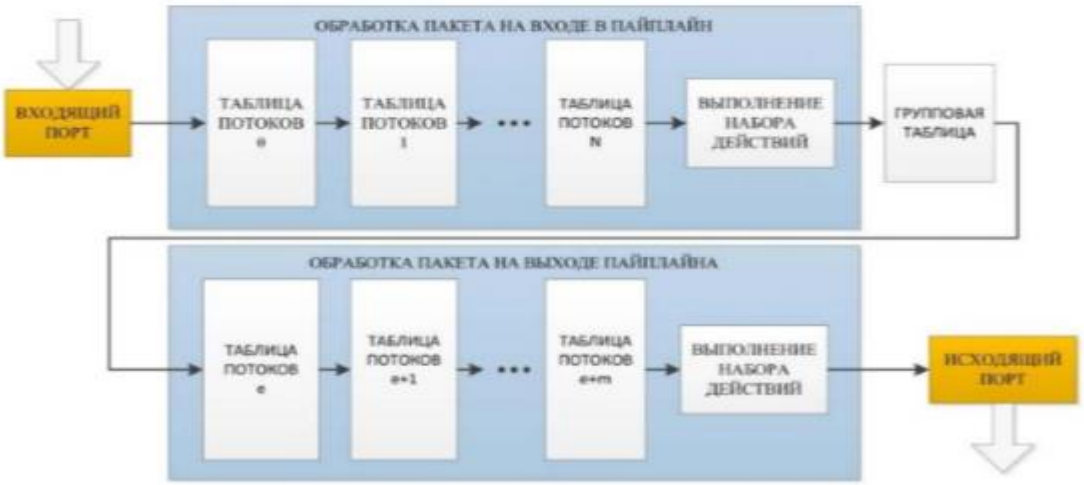

Рисунок 12 – Конвейерная обработка пакета

Конвейер (Pipeline) – набор связанных таблиц, которые обеспечивают проверку заголовков, пересылку и модификацию пакетов в OpenFlow коммутаторе.

OpenFlow агент получает команды от контролера и формирует таблицу flow, которая содержит инструкции по обработке пришедшего PDU пакета. Таким образом OpenFlow связывает устройства управления и устройства передачи.

OpenFlow широко внедряется производителями сетевого оборудования из-за простой структуры OpenFlow-коммутатора, которая может быть реализована за счет небольших модификаций программного и аппаратного обеспечения. В результате переход на протокол OpenFlow может быть произведен пошагово с внедрением протокола в те сетевые сегменты, которые требуют функций OpenFlow.

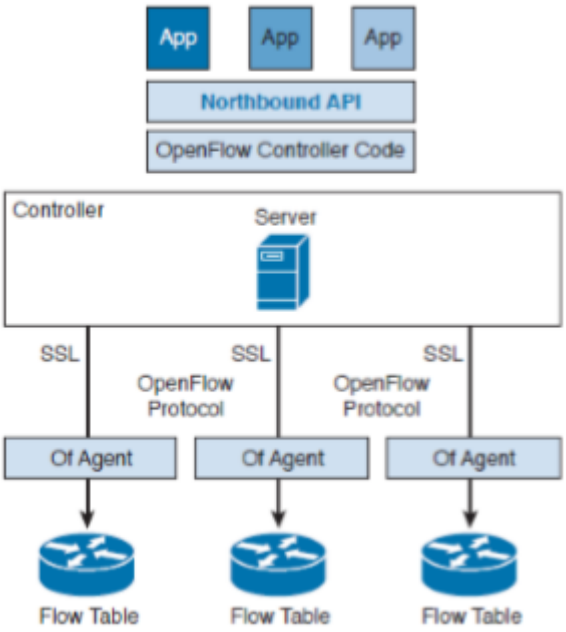

Рисунок 13 – Схема соединения с помощью OF – агента

1.5 SDN и виртуализация

Одним из вариантов развертывания программно-конфигурируемой сети является интеграция SDN и NFV (Network Functions Virtualization) [8]. NFV-концепция сетевой архитектуры, предлагающая использовать технологии виртуализации для виртуализации целых классов функций сетевых узлов в виде составных элементов. Виртуализация сетевых функций, это замена привычного оборудования (маршрутизаторов, коммутаторов и пр.) виртуальными аналогами. Виртуализируемая сетевая функция (англ. virtualized network function, VNF) может включать одну или несколько виртуальных машин, использующих разное программное обеспечение и процессы, поверх отраслевых стандартов, серверы, коммутаторы и хранилища большого объема, или даже инфраструктуру облачных вычислений, вместо отдельных аппаратных решений для каждой сетевой функции. Кроме того, концепция SDN также подразумевает использование мощных стандартизированных серверов и коммутаторов. В качестве платформы для работы NFV используются стандартные x86-серверы и широко используемые технологии виртуализации (VMware, KVM, и др.). Интеграция SDN и VMware NSX представлена на рисунке 14.

Платформа виртуализации осуществляет взаимосвязь между вычислительной сетью, управляемой протоколом OpenFlow, и внешними OpenFlow контроллерами. Все управляющие сообщения, которые проходят по управляемой протоколом OpenFlow сети к внешним OpenFlow-контроллерам, тем или иным образом проходят через компоненты платформы виртуализации. Процедуры создания и анализа сетевых слоев (сетевых срезов) требуют изоляции слоёв, идентификации области заголовков, которая не

пересекается с какими-либо существующими областями. С помощью заголовка пакета определяется слой, к которому этот пакет относится.

Платформы виртуализации осуществляют изолирование слоев, контролируя то, что потоки трафика, принадлежащие разным слоям, не перекрываются в сетевой инфраструктуре. Это необходимо для оптимального использования ресурсов.

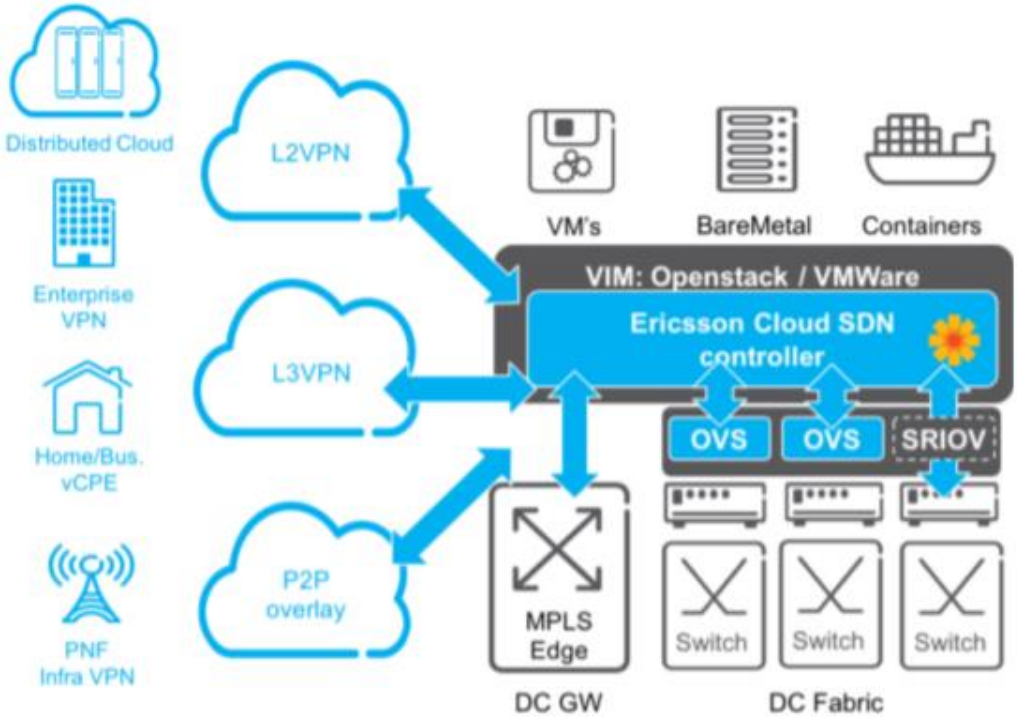

Рисунок 14 – Интеграция SDN и VMware NSX

1.6 Область применения SDN

На сегодняшний день перспективной областью применения SDN являются центры обработки данных. В силу больших объёмов проходящего трафика и его изменяющейся, в зависимости от нагрузок и конфигурации ПО, структурой. Облачные и виртуализованные ЦОД создают новые требования к базовой сети. В таких ЦОД используется динамическая модель – произвольные ресурсы виртуальных машин распределяются на вычислительные узлы в случайном порядке. Это не позволяет заранее эффективно настроить сеть, так как виртуальные машины (ВМ), находящиеся в ЦОД должны функционировать, как ВМ организованные в логические серверные пулы в сети каждого клиента [3], [8], [12].

Кроме того, трафик от разных арендаторов должен быть изолирован, как для безопасности, так и для производительности. Различные сетевые функции, такие как Firewall, системы распознавания интернет-трафика (DPI), балансировка нагрузки должны быть добавлены по требованию клиента и в соответствии с его трафиком. Таким образом, сетевые функции должны быть как никогда тесно связаны с вычислительным функционалом поскольку сетевые политики должны соответствовать политикам вычислений и обычной статической конфигурации сети не может быть достаточным.

Перенастройка политик без централизации всего управления крайне затруднительна. С внедрением SDN менеджер ЦОД имеет возможность управлять API (интерфейсом прикладного программирования) для применения новых требований к сетевому контроллеру. Сетевой контроллер может затем использовать интерфейс API, например, OpenFlow, чтобы применить требования по обеспечению сетевого доступа и политики к программному коммутатору.

Наиболее распространенным решением внедрения является развертывание наложения SDN для сопоставления динамической конфигурации со статической сетью [3].

Программный коммутатор динамически маршрутизирует пакеты с виртуальных машин на разных статических туннелях, установленных на сети. Центры обработки данных являются важной внутренней частью многих крупных компаний. Например, Google Facebook, Amazon и Yahoo, и применение SDN к автоматизации процессов в ЦОД, упрощает их эксплуатацию, и позволяют значительно снизить CAPEX и OPEX, и повысить производительность, за счет централизованного управления.

2 Информационная безопасность сетей SDN

Под термином информационная безопасность (ИБ), принято понимать состояние защищенности информационной системы (ИС), включая информацию и поддерживающую ее инфраструктуру. ИС находится в состоянии защищенности, если обеспечены ее конфиденциальность, доступность и целостность [13].

Конфиденциальность – свойство информации, которое указывает на необходимость введения ограничений на круг лиц, имеющих доступ к данной информации, и обеспечиваемое способностью системы сохранять указанную информацию в тайне от лиц, не имеющих прав на доступ к ней.

Доступность – такое свойство информации, которое характеризуется обеспечением своевременного санкционированного доступа субъектов к информации и соответствующим автоматизированным службам всегда, когда в обращении к ним возникает необходимость.

Целостность – свойство информации, которое заключается в ее существовании в виде неизменном по отношению к некоторому фиксированному ее состоянию [14].

Под угрозой принято подразумевать любое действие, которое может быть направлено на нарушение ИБ системы.

Угроза является следствием наличия уязвимых мест или уязвимостей в информационной системе. Уязвимости могут возникать по разным причинам, например, в результате непреднамеренных ошибок программистов при написании программ.

Атака – это реализованная угроза.

Риск – это вероятностная оценка величины возможного ущерба, который может понести владелец информационного ресурса в результате успешно проведенной атаки.

2.1 Классификация возможных атак на сети SDN

Архитектура SDN значительно упрощает конфигурацию и управление сетевой архитектурой, за счет централизованного подхода и возможности динамически программировать сеть. Но также имеет определенные потенциальные уязвимости сетевой инфраструктуры и особенности обеспечения информационной безопасности [8]. На практике, сетевая архитектура SDN, так же, как и сети классической архитектуры уязвимы для многих способов несанкционированного вторжения в процесс обмена данными. По мере развития компьютерных и сетевых технологий список возможных типов сетевых атак на IP-сети постоянно расширяется [15-16].

С учетом трехуровневой архитектуры SDN и централизованного подхода к организации сети, можно обозначить возможные точки атак. Возможные точки атак представлены на рисунке 15.

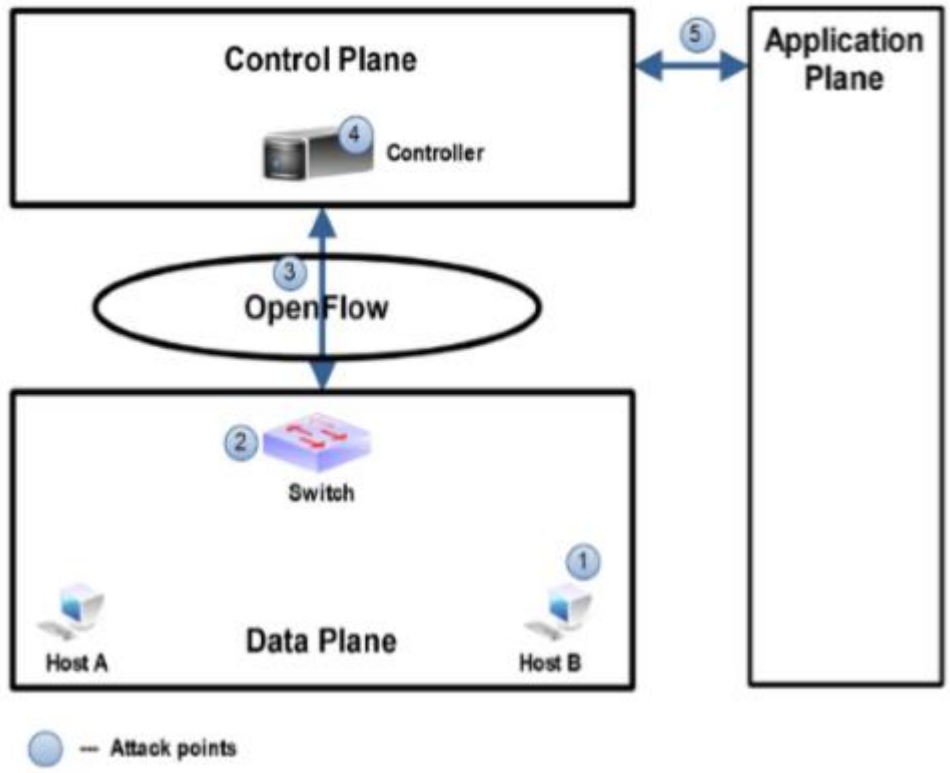

Рисунок 15 – Векторы атак на сеть SDN

Исходя из этого подход к обеспечению безопасности сети SDN можно разделить на 4 вектора. Векторы безопасности сети SDN представлены в таблице 4.

Для обеспечения ИБ в сетях SDN необходимо обеспечить безопасность каждого уровня архитектуры и особое внимание уделить безопасности контроллера, который является единой точкой отказа и критически важным компонентом сети SDN. Контроллер как ключевой компонент в управлении SDN является наиболее уязвимым элементом и большинство атак используют

эту особенность. Их можно классифицировать по вектору и типу угроз [11], [17-19], [20].

Проекция возможных атак на плоскости архитектуры SDN представлена в таблице 5.

Таблица 4 – Векторы безопасности SDN

Безопасность Data Plane	Безопасность Control Plane	Безопасность каналов связи	Безопасность API
– уязвимость программного обеспечения – ddos атаки – атака сетевых устройств изнутри сети	– DDoS атаки на контроллер – целостность контроллера – внедрение нежелательной информации в контроллер – уязвимости ОС поверх которой работает контроллер	– в каналах Open Flow используются SSL/TLS, но данные протоколы не являются обязательными – аутентификация между контроллером и OF устройствами – DDoS атаки – поддержание насыщенности канала	– HTTP spoofing – некорректные инструкции для контроллера – возможность к осуществлению других типов атак

Таблица 5 – Классификация атак в SDN

Вектор атак	Специфично для SDN	Атаки
Атаки на топологию сети (Data Plane)	Нет	Spoofing атаки: ARP Poisoning, Fake topology UDP spoofing. TCAM exhaustion
Атаки на уровень управления (Control Plane)	Да	DoS контроллера, MiTM, PACKET IN Flooding, Switch blackhole
Атака на уровень приложений SDN	Да	HTTP spoofing
Атака на канал связи	Нет	MiTM

Описание возможных атак в SDN:

а) атаки на плоскость данных.

Spoofing (подмена) – атака может считаться атакой подмены в том случае, когда злоумышленник генерирует трафик, представляемый от другого, легитимного источника. Подмена менеджмент трафика (ARP, IGMP, LLDP и

другие пакеты). Существует несколько типов атаки подмены, в том числе следующие:

– подмена IP адреса. В этом случае, атакующий подменяет свой IP адрес (т.е. адрес источника в IP пакете);

– подмена MAC адреса. Данный тип подмены осуществляется на канальном уровне модели OSI. Атакующий подменяет свой MAC на чужой;

– подмена сервиса или приложения. Например, подмена легитимного DHCP сервера в сети, что может привести к выдаче заведомо ложных настроек сети в ответ клиентов.

ARP Poisoning – подмена информации о хостах в сети с помощью поддельных ARP запросов. Возможность перехвата трафика, применение вредоносных правил для потоков в сети и изменение их направления.

Fake topology (перестройка топологии сети) – атакующий хост создает поддельное звено в сети, происходит пересчет путей маршрутизации, что может нарушить маршрутизацию и правильное направление потоков в сети [11].

UDP spoofing – атакующий отправляет большое количество UDP-пакетов в случайные порты на целевом объекте, целевой хост постоянно проверяет приложение на этом порту. Поскольку ни одно прослушивающее приложение на этом порту не найдено, оно отвечает с недостижимым пакетом ICMP. Этот процесс потребляет много ресурсов, в результате чего хост становится недоступным. DDoS - атаки.

Для SDN характерна атака переполнения типа TCAM exhaustion (переполнение памяти коммутаторов) – зараженный хост отправляет множество сообщений в сеть и вынуждает контроллер устанавливать большое количество правил для потоков, расходуя TCAM. Это приводит к высоким задержкам и потере пакетов. Атаки переполнения буфера – вектор атак, производимый на стороне клиента. Вредоносный код может быть внедрен в файлы данных, и код может быть выполнен, в случае открытия его уязвимым клиентским приложением.

Использование системы обнаружения (IDS), чтобы помочь в выявлении любых аномальных потоков. Применение списков доступа (ACL) и аутентификации и авторизации устройств в сети. Использование зашифрованных паролей и отключение неактивных сервисов. Применение таких механизмов защиты коммутаторов как: IP source guard, port security, uRPF, private Vlan. А также различные автоматические решения для управления трафиком для программных компонентов.

б) атаки на плоскость управления.

Атаки переполнения буфера. Атакующий может проанализировать сетевые серверные приложения на наличие ошибок. Уязвимость переполнения буфера - одна из уязвимостей. Если какая-либо служба принимает входные данные и ожидает, что ввод будет в пределах определенного размера, но при этом не проверяет размер фактического ввода, то такая служба может быть уязвима для атаки переполнения буфера. Это

означает, что злоумышленник может предоставить входные данные, размер которых превышает ожидаемый, вследствие чего, служба будет принимать и записывать входные данные в память, заполняя соответствующий буфер, а также перезаписывая соседнюю память. Такое переписывание памяти может привести к сбою системы, в результате чего произойдет DoS. В худшем случае злоумышленник сможет ввести вредоносный код в буфер, что приведет к компрометации системы.

Цель атаки – сделать недоступными ресурсы системы для легальных пользователей путем истощения ресурсов (вычислительных, пропускной способности сети и т.п.) атакуемой системы. Атака TCP SYN Flood является классическим примером атаки DoS. Атака TCP SYN Flood использует трехэтапный процесс установления связи TCP, отправляя несколько пакетов TCP SYN со случайными адресами источника на атакуемый хост. Жертва отправляет в ответ SYN ACK к произвольному исходному адресу и добавляет запись в таблицу соединений. Поскольку SYN ACK предназначен для некорректного или несуществующего хоста, последняя часть трехстороннего подтверждения никогда не завершается, и запись остается в таблице соединений до истечения времени таймера. Путем создания с высокой частотой TCP SYN пакетов от случайных IP-адресов, злоумышленник может заполнить таблицу подключений и, фактически, лишить легальных пользователям служб TCP (таких как электронная почта, передача файлов или WWW). Из-за того, что, IP адрес источника является ложным, не существует простого способа отслеживания источника атаки. Некоторые атаки DoS, такие как Ping of Death, могут привести к сбою службы, системы или группы систем. В атаках Ping of Death злоумышленник создает фрагмент, размером более 65 536 байт. 65 536 байт - это максимальный размер пакета, определенный IP-протоколом. Уязвимый хост, при получении этого сообщения, производит попытку настройки буфера повторной сборки пакетов, которая в итоге приводит к сбою или перезагрузке системы. Атака Ping of Death эксплуатирует уязвимость в обработке на IP уровне, но существуют и аналогичные атаки, использующие уязвимости на прикладном уровне. Атакующие используют уязвимости, вызывая системные сбои отправляя специальном образом подготовленные SNMP, syslog, DNS или другие протокольные сообщения на основе протокола UDP. Эти сообщения могут приводить к сбоям в работе различных функций синтаксического анализа и обработки, приводящих большинстве случаев к сбою системы и перезагрузке. Также существует и IPv6 версия атаки Ping of Death.

Когда попытка DoS происходит от одного хоста сети, она представляет собой DoS-атаку. В DoS атаке могут потенциально принимать участие тысячи источников, в этом случае такая атака называется DDoS-атакой. Зараженные хосты и коммутаторы могут провести DoS (Denial-of-service) атаку чтобы вывести из строя ресурсы на уязвимых коммутаторах и/или сам SDN контроллер, влияя на пересылку данных.

Атака «человек по середине» (MiTM) - это комплексная атака, которая включает в себя успешные атаки на IP-маршрутизацию или протоколы (например, ARP, DNS или DHCP), приводящие к перенаправлению трафика. Методом компрометации канала связи атака направлена на обход взаимной аутентификации злоумышленник устанавливает независимые связи с жертвами и передаёт сообщения между ними. Атака такого типа, скорее обобщенная концепция, которая может быть реализована во многих сценариях, в отличие от конкретных видов атак. Как правило, атакующий имеет возможность просматривать данные между двумя узлами сети. Атака «человек по середине» может быть пассивной или активной. В пассивном режиме злоумышленники похищают конфиденциальную информацию. При активных атаках злоумышленники модифицируют данные при передаче или вводят собственные данные [16].

PACKET IN Flooding – генерирует ряд потоков на хосты, чтобы вызвать поток PACKET IN сообщения контроллеру и тем самым снизить производительность. Вредоносная программа может установить несколько правил потока с помощью сообщений FLOW MOD к целевому коммутатору для переполнения таблицы потоков [21-22].

Switch blackhole (создание «черной дыры» в сети) – зараженный коммутатор может сбрасывать или перенаправлять пакеты и в результате поток не доходит. Это может значительно нарушить работу сети и бизнес приложений.

Репликация контроллера, восстановление системы, включая периодическое обновление системы для поддержания надежного состояния, ограничение скорости, фильтрация событий, сброс пакетов, настройка тайм-аута, агрегация потока, использование протоколов TLS /SSL. Использование механизмов аутентификации и RBAC. Использование МСЭ, обеспечение безопасности нижестоящей ОС, актуальные обновления, надежные пароли, обеспечение безопасности сервисных портов;

в) атаки на уровень приложений SDN.

HTTP spoofing – простые HTTP-запросы отправляются GET и POST, которые требуют огромного количества данных в ответ, потребляющих большой объем полосы пропускания, сбрасывая сервер. распространенными DDoS-атаками в сети.

Кроме того, наличие открытых программных интерфейсов и низкий уровень стандартизации, предполагает возможные уязвимости в API, через которые можно получить полный контроль сети SDN, используя контроллер [11].

Наиболее важные критерии решения заключаются в использовании механизмов, которые могут создавать автономное доверительное управление для проверки доверия приложения в течение его срока службы. Программные модули для контроллера должны проходить предварительную проверку на уязвимости перед внедрением. Management security (SSH HTTPS).

2.2 Механизмы обеспечения безопасности контроллера SDN

Исходя из анализа возможных атак, можно сделать вывод, что из-за наличия централизованного управления атаки с подменой и разновидности DDOS имеют более высокий потенциал для сетей SDN, чем в традиционных сетях.

Подмена менеджмент-трафика, передаваемого между коммутаторами и сетевыми контроллерами, может повлечь несанкционированный доступ к сетевым устройствам, получение контроллером некорректной статистики или данных о состоянии сети, что может сказаться на работе всей сети [17]. A DDOS атаки в реактивном режиме контроллера приводят к долгосрочному блокированию контроллера и полностью останавливают обработку сетевого трафика.

Атаки на уровень топологии влекут за собой дальнейшую возможность получения доступа к контроллеру и являются первым этапом к компрометации контроллера к нарушению ИБ системы. Для того чтобы исключить любую возможность компрометации контроллера, необходимо обеспечить три составляющие безопасного контроллера SDN. Основные требования безопасности для контроллеров SDN представлены на рисунке 16.

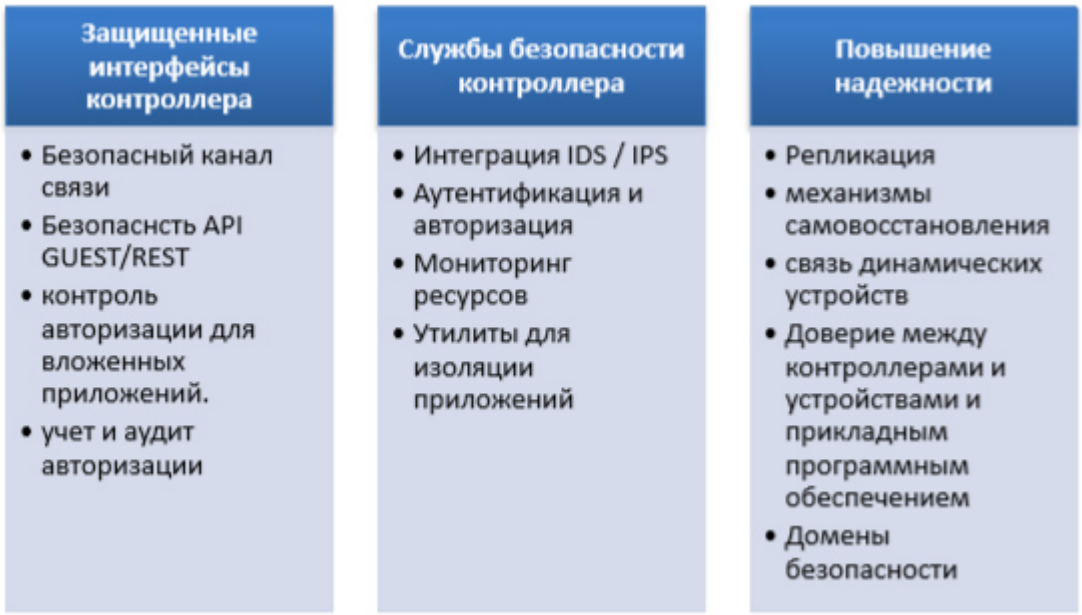

Рисунок 16 – Основные требования безопасности для контроллеров SDN

Защищенные кейсы контроллера. В контроллерах SDN существует три интерфейса: D-CPI, I-CPI и A-CPI для контроллера данных, промежуточного контроллера и контроллера приложений соответственно. Кроме того, графический интерфейс пользователя (GUI) – это еще один широко используемый интерфейс для контроллера. GUI предоставляется для упрощения управления и предоставления информации о сетевых устройствах, топологии сети, таблицы потоков. Для всех этих интерфейсов, будь то D -CPI,

ICPI, A-CPI, TLS или GUI, конфиденциальная связь должна быть защищена [11], [23].

Для проверки подлинности и создания безопасного канала обмена Open Flow используются способы на основе таких протоколов как:

– Transport Layer Security (TLS);

– Secure Shell (SSH);

– IPSec.

а) TLS и его предшественник, Secure Sockets Layer (SSL) – это криптографические сетевые протоколы для обеспечения передачи данных. TLS обеспечивает конфиденциальность и целостность данных, а также аутентификацию сервером, а иногда и клиентом. Основан на асимметричном шифровании, использует асимметричный алгоритм (RSA) для аутентификации и безопасного обмена ключами. Безопасный канал на основе протокола TLS представлен на рисунке 17.

Рисунок 17 – Безопасный канал на основе протокола TLS

Связь между коммутатором и контроллером устанавливается с помощью механизма рукопожатия и аутентификации с последующим шифрованием данных. Установление TLS соединения представлено на рисунке 18.

TLS использует цифровые сертификаты для проверки подлинности пользователь/хост. Цифровые сертификаты используют открытый ключ, а также другие данные, такие как имя субъекта, имя эмитента, срок действия и алгоритм, используемый для подписания. Цифровой сертификат выдается органами сертификации (CA).

Осуществление поддержки TLS различными коммутаторами представлено в таблице 6.

Когда пользователь запрашивает сертификат субъекта от центра сертификации, ЦС подписывает сертификат, используя его закрытый ключ, перед отправкой пользователю; пользователь затем расшифровывает подписанный сертификат, используя открытый ключ CA. Таким образом, пользователь может проверить, что сертификат действителен и не был подделан.

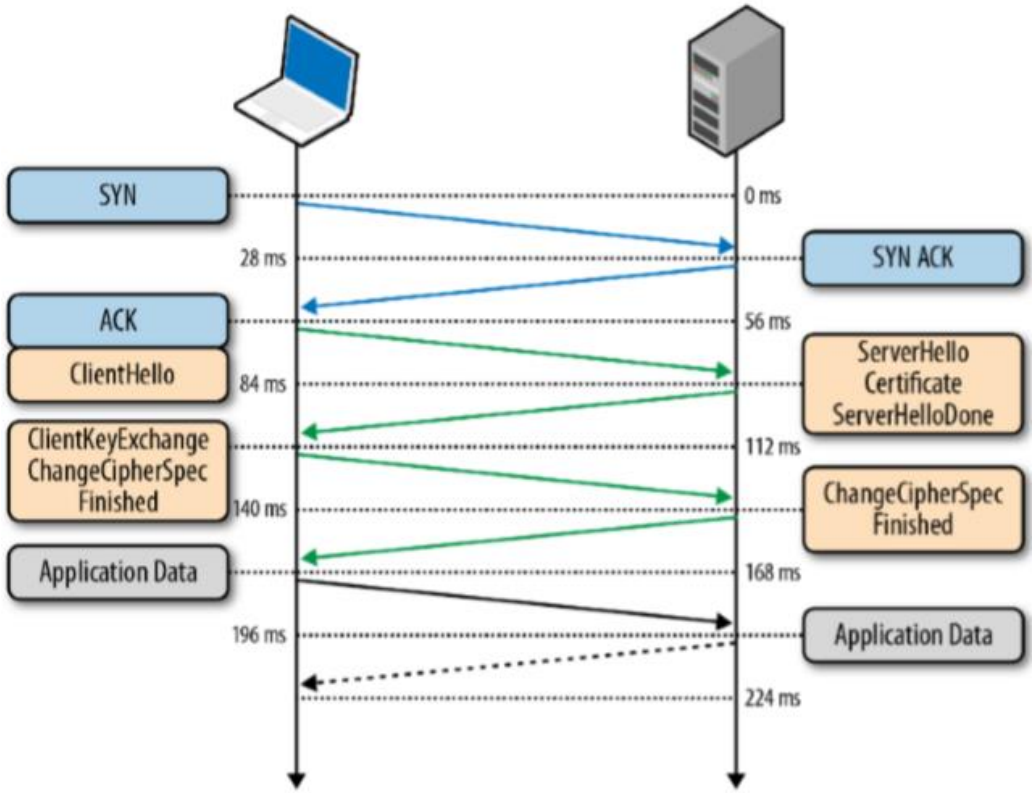

Рисунок18 – Установление TLS соединения

В архитектуре OpenFlow используются само заверенные корневые сертификаты – специальный тип сертификата, подписанный самим его субъектом. Создатель сертификата сам является в данном случае ЦС. Все корневые сертификаты доверенных УЦ являются само заверенными. Реализуется, используя OpenSSL для реализации открытого ключа X.509 инфраструктуры.

Сначала коммутатор отправляет пакет Hello на контроллер. Этот Hello указывает версию TLS и алгоритм шифрования, который поддерживает коммутатор. контроллер затем отвечает собственным пакетом Hello, после чего он отправляет свой сертификат коммутатору. Таким образом, коммутатор может аутентифицировать контроллер.

Затем коммутатор генерирует закрытый ключ и использует открытый ключ контроллера для шифрования закрытого ключа и отправляет его на контроллер. После этого контроллер переключается на TLS, таким образом, аутентифицируется контроллер; однако это оставляет сеть открытой для атак из скомпрометированных коммутаторов. Коммутатор тоже должен пройти аутентификацию. Это делается путем аутентификации коммутаторов с использованием их открытого ключа. Когда контроллер отправит свой сертификат на коммутатор, он запросит коммутатор проверить его личность, отправив открытый ключ. Это предотвращает добавление противником вредоносных коммутаторов в сеть, что значительно увеличивает безопасность сети.

Таблица 6 – Осуществление поддержки TLS различными коммутаторами

Вендор	Поддержка TLS
HP	Нет
Brocade	Нет
Dell	Нет
Open vSwitch	Да
Cisco	Да

б) SSH является общим названием для всего семейства протоколов Secure shell. Используется для передачи файлов (SCP, SFTP), дистанционного управления ресурсами, туннелирования и т.д.

Защищенный SSH тоннель представлен на рисунке 19.

В этом подходе используются автоматически сгенерированные ключи. Контроллер является сервером для коммутаторов, а коммутатор - клиент SSH.

Клиент подключается к серверу и аутентифицируется с помощью ключа. При этом он аутентифицирует также ключ сервера [2]. Затем с помощью сообщения OpenFlow создается защищенный туннель между портом на стороне сервера (контроллер) и настроенном портом на коммутаторе. Коммутатор отправляет незашифрованный трафик на локальный порт, и трафик передается в порт на контроллере используя зашифрованный и безопасный туннель.

Рисунок 19 – Защищенный SSH тоннель

в) IPSec – это набор протоколов для обеспечения безопасности соединения и шифрования обмена ключами между хостами.

Защищенный тоннель IPSec, переносящий данные OpenFlow по зашифрованному каналу представлен на рисунке 20.

IPSec состоит, по меньшей мере, из двух каналов связи между подключенными устройствами:

– обменный канал, через который данные связаны с аутентификацией и шифрованием (ключи);

– канал (один или несколько), который переносит пакеты, переданные по уже установленной линии.

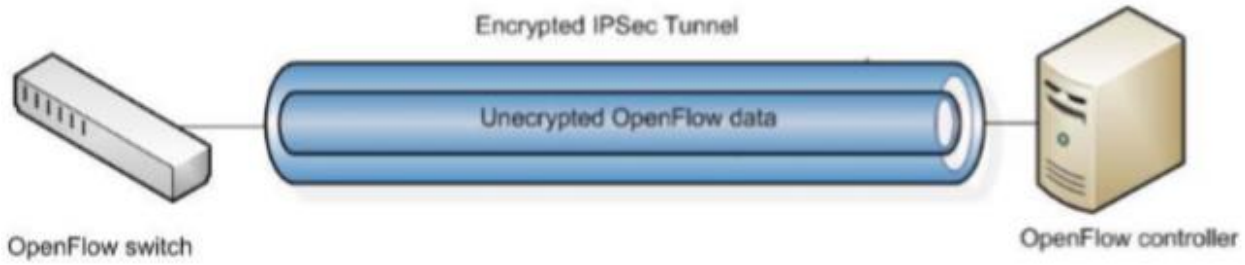

Рисунок 20 – Защищенный тоннель IPSec, переносящий данные OpenFlow по зашифрованному каналу

Графический интерфейс пользователя (GUI) может быть защищен с помощью авторизации или аутентификации пользователей перед доступом к контроллеру GUI. Например, в OpenDayLight Controller для входа необходимы имя пользователя и пароль. Аналогичным образом, в ОНОС применяется GUI API Security, но без требования авторизации или аутентификации, вместо этого ONOS требует IP-адрес машины, на которой установлен контроллер [14].

Помимо защиты платформы управления и её интерфейсов, важна также безопасность самого контроллера SDN. Например, многие контроллеры имеют внутреннюю интеграцию IDS / IPS, авторизацию и аутентификацию, мониторинг ресурсов и ведение журнала / проверка безопасности сервисов. Реализация этих методов на контроллере повышает его безопасность.

Интеграция системы обнаружения вторжений (IDS) и системы предотвращения вторжений (IPS), необходимы для обнаружения и предотвращения потенциальных угроз. Реализуется оценкой отклонения статистики данного типа трафика от стандартной, и генерацией предупреждений на основе набора правил. Принцип проектирования отделяет мониторинг трафика от управления потоком, путем зеркалирования трафика от портов доступа коммутатора на коммутируемый анализатор портов (SPAN) / зеркальный порт, который затем пересылает трафик в систему обнаружения вторжений (IDS) для мониторинга.

Интеграция межсетевого экрана с контроллером SDN – для выявления и изолирования подозрительных компьютеров внутри сегмента, ограниченного межсетевым экраном, и блокировка внутреннего распространения зараженного трафика.

Интеграция с IDS с помощью SPAN-порта представлена на рисунке 21.

Мониторинг – мониторинг поведения приложений и проверка пакетов. При проникновении злоумышленника система должна реагировать должным образом для восстановления работы контроллера. На этапе мониторинга трафика весь трафик должен контролироваться через сеть подходы мониторинга, например, sFlow, NetFlow, порт SPAN и т.д.

Сканер уязвимостей – может быть реализован как программное обеспечение, прошивка, аппаратное обеспечение или их комбинация. Сеть сканируется на уязвимые потоки и на основе результата, генерируемого

сканером уязвимостей, представляет директивы политики потока в исполнительный механизм безопасности.

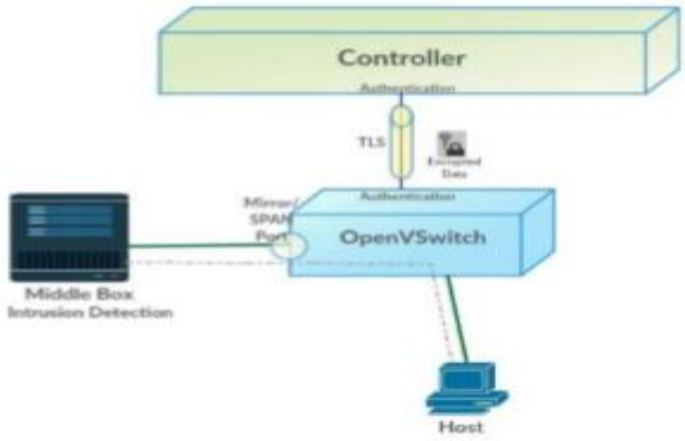

Рисунок 21 – Интеграция с IDS с помощью SPAN-порта

Например, перенаправлять внешних посетителей из уязвимой комбинации хостов и портов, продолжая разрешать внутренний доступ к уязвимому узлу / порту, перенаправлять весь трафик от уязвимой комбинации хоста / порта или реализовывать любую другую желаемую политику сетевого потока. Таким образом, директива политики одного потока может приводить к введению множества правил изменения потока с помощью исполнительного механизма безопасности на одном или нескольких сетевых переключателях [18-19].

Аутентификация, авторизация и учет (AAA) – обеспечивают эффективный контроль доступа к ресурсам, для пользователей и приложений. Процесс аутентификации включает в себя создание учетных данных во время выполнения уникальной идентификации приложения. Для привилегированных системных вызовов авторизация статуса заявки оценивается модулем авторизации приложения с помощью подписанного ключа, соответствующего приложению.

Мониторинг ресурсов необходим для управления ресурсами в различных контроллерах. Наиболее предпочтительным является менеджер ресурсов (например, ЦП, файл дескриптор и память) с использованием различных приложений. Мониторинг помогает в отслеживании системы для определения аномального поведения и обнаружения ошибок или вредоносных приложений. Также, правильное управление ресурсами гарантирует, что ресурсы контроллера поддерживают максимальное количество приложений.

В SDN механизмы обеспечения надежности можно разделить на два самостоятельных: защитное переключение (резервирование) и восстановление (перемаршрутизация). Для повышения отказоустойчивости плоскости данных используют оба метода, а в плоскости управления используют методы защитного резервирования контроллеров и их виртуализацию, которая выполняется путем перемаршрутизации.

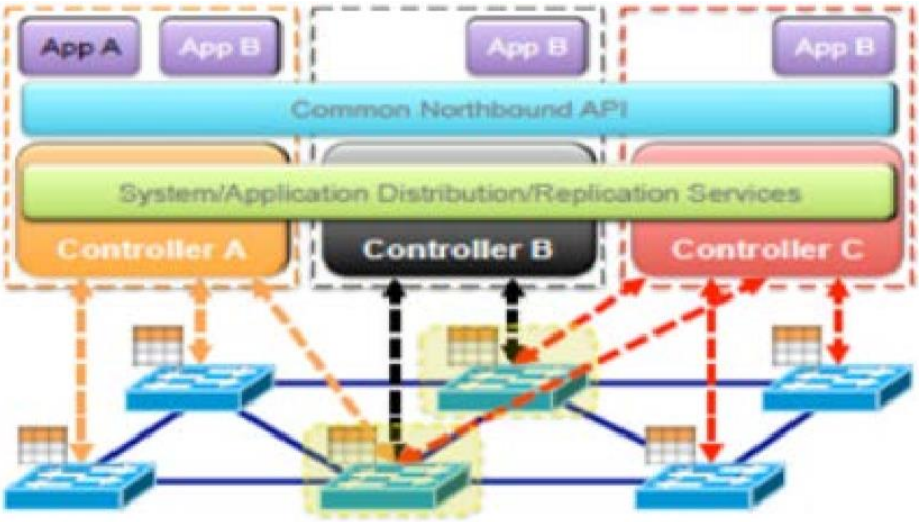

Рисунок 22 – Репликация контроллера SDN

Репликация – метод повышения надежности и безопасности систем SDN. Репликация контроллера несколькими экземплярами обеспечивает устойчивость к программным и аппаратным авариям и вредоносным ошибкам. Реплицировать рекомендуется с помощью контроллеров на разных платформах, для предотвращения общих ошибок, включая уязвимости и ошибки программного обеспечения. Часто, операционные системы разных поставщиков, характеризуются ограниченным числом пересекающихся уязвимостей. Таким образом, снижается влияние на безопасность. Репликация контроллера SDN представлена на рисунке 22.

Кроме того, важно обеспечить динамическую и надежную связь каждого коммутатора с несколькими контроллерами. Это предотвратит сбой в операциях управления коммутатора. В этом случае обеспечение безопасности гарантируется использованием пороговой криптографии для обнаружения вредоносных контроллеров и предотвращения атак типа «человек в середине».

Контроллеры должны быть настроены на сообщение о вредоносных или ошибочных контроллерах в соответствии с алгоритмом обнаружения отказа или аномалии. Вредоносный контроллер должен быть автоматически изолирован, когда его надежность падает ниже недопустимого порога [11], [18-19].

Также, рекомендуется использовать изолированную технику доменов безопасности для защиты сети от атак. Например, разграничение доступа в зависимости от уровня привилегий пользователя. В SDN-контроллерах домен безопасности достигается с помощью механизмов, таких как виртуализация песочницы. При таком дизайне активный режим изоляции может быть установлен с использованием четко определенного интерфейса, который позволяет устанавливать связи и операции между изолированными доменами.

2.3 Анализ существующих комплексных решений обеспечения безопасности SDN

На данный момент существуют готовые комплексные решения по обеспечению безопасности для внедрения в SDN сети. Они могут быть реализованы в качестве программного обеспечения, дополнительных расширений для контроллеров, облачных платформ и сервисов. В таблице 7 приведен краткий анализ данных решений.

Таблица 7 – Анализ комплексных решений безопасности

FORTNOX [22]	Расширение для контроллера NOX – система контроля доступа на основе ролей, фокусируется на плоскости управления. Проверяет правила потока с цифровой подписью перед вставкой flow table. Аутентификация осуществляется с помощью цифровых подписей. Аутентификация ролей с тремя уровнями: 1) пользователь с наивысшим приоритетом; 2) приложение безопасности; 3) приложение с наименьшим приоритетом.
FlowVisor [24]	FlowVisor действует как прозрачный прокси между контроллерами и переключателями. Он позволяет создавать виртуальные сети поверх SDN, использовать несколько экспериментальных сетевых срезов (комбинации переключателей порты, MAC-адреса, IP-адреса, адреса портов или ICMP тип) и отделять производственный трафик. Делегирует управление срезами для разных контроллеров - один срез не может контролировать трафик другого.
VeriFlow [17]	Система для проверки состояния сети во время пересылки пакетов
AvantGuard [25]	Расширение SDN, которое повышает безопасность и устойчивость самого OpenFlow. Включает в себя два новых модуля: миграция подключений модулем и исполнительным триггерным модулем фильтрации неполные TCP-соединения, установив сеанс рукопожатия перед поступлением пакетов контроллер. Соединения TCP поддерживаются модулем подключения миграции чтобы избежать угроз атаки насыщения TCP. Управляющий триггерный модуль позволяет плоскости данных, чтобы сообщать о состоянии сети и активировать определенное правило потока, основанное на предопределенных условий движения

Продолжение таблицы 7

OpenSec	Основан на политиках безопасности. Действует как виртуальный уровень между пользователем и контроллером OpenFlow. автоматически преобразует политики безопасности в набор правил, которые вставляются в сетевые устройства. OpenSec позволяет сетевым операторам определять, как автоматически реагировать при обнаружении вредоносного трафика
SE-Floodlight	Расширение для повышения безопасности SDN с помощью проверки подлинности на основе ролей. Функция «Наименьшая привилегия» позволяет приложениям OpenFlow работать вне контекста процесса управления. Функция Security Audit представляет новую подсистему аудита OpenFlow, которая отслеживает все связанные с безопасностью события, создаваемые стеком OpenFlow.

С точки зрения обеспечения безопасности, в динамически программируемой сети есть много плюсов. Например, возможность реализовывать политику безопасности как файл данных, база данных, таблица или другая подходящая компьютеризированная структура данных. Теперь термин «сетевой администратор» может относиться, как к человеческому оператору, так и к программному приложению сетевой безопасности или компьютеризированному агенту, или делегату человеческого оператора, например, к программному приложению, которое действует под руководством или в ответ на исходные данные оператора [11].

Политика безопасности может быть реализована на коммутаторах в виде ряда правил сетевого потока, которые поддерживаются на коммутаторах в локальных таблицах. Каждый из переключателей обновляет свою соответствующую таблицу локального потока в соответствии с обновлениями размещения пакетов. В некоторых вариантах осуществления коммутаторы могут сообщать изменения в локальных таблицах потока обратно в службу медиа-посредничества.

Теперь можно применять правильную защиту к конкретному потоку, идентифицировать потоки, требующие более глубокой проверки, выделять услуги на совместно используемой инфраструктуре и достигать более хорошей видимости сетевого трафика.

Для сбора и мониторинга статистики трафика используется протокол Flow. Затем можно построить предельные уровни безопасного трафика и сравнивать трафик в реальном времени с сохраненными уровнями. в традиционных сетях практически невозможно достичь такой видимости. Сравнивая трафик в реальном времени с базовым уровнем, можно выявлять отклонения и динамически перенаправлять подозрительный трафик на

дополнительную проверку или даже временно изолировать его до устранения проблем. Тем временем проверенный трафик направляется на соответствующие целевые узлы. Это позволяет применять динамическое управление политикой, которое можно настраивать в реальном времени в соответствии с происходящими событиями [26].

3 Моделирование и анализ работы сегмента сети SDN

Для отладки, тестирования и анализа работы сети. А также для разработки новых технологий сети SDN применяются средства эмуляции и моделирования сети с помощью виртуализованного стенда, который работает в реальном времени, соединяет реальные устройства с виртуальными устройствами или с реальными приложениями.

Средства моделирования и эмуляции позволяют разработчикам системы определять правильность и эффективность дизайна до развертывания системы. Виртуальное сетевое моделирование имеет низкую стоимость для сборки, гибкое для любой топологии и легко масштабируемое. Наиболее простым и популярным вариантом эмуляции топологии SDN является сетевой эмулятор Mininet, который реализуется в рамках одной виртуальной машины, и позволяет создавать топологию сети с различным количеством хостов. Но данный вид эмуляции не представляет в полной мере информации о работе реальной SDN сети, поэтому в данной работе, будет использоваться моделирование сегмента SDN на базе платформы виртуализации Oracle VM VirtualBox и отдельных хостов в качестве виртуальных машин, что позволит получит наиболее полное представление о работе и эффективности SDN.

3.1 Контроллер OpenDayLight

Одним из контроллеров с поддержкой OpenFlow является OpenDayLight контроллер. OpenDayLight – Open-source проект с гибкой модульной платформой MD-SAL для встраивания различных плагинов с поддержкой нескольких протоколов и обеспечения согласованных услуг для модулей и приложений. Модельный уровень абстракции сервиса (MD-SAL) — это среда OpenDayLight, для создания новых функций в виде служб и драйверов протоколов. SAL предоставляет такие базовые сервисы, как Device Discovery (обнаружение сетевых устройств в сети и подключение к ним), который в свою очередь используются такими модулями как Topology Manager (хранение, обновление информации о топологии сети) и т.д.

Проект разрабатывается под лицензией EPL v1.0(Eclipse public license) при поддержке Linux Foundation. Разрабатывается на языке Java и является кроссплатформенным [27].

Контроллер OpenDayLight реализован исключительно в программном обеспечении и хранится в его собственной виртуальной машине Java (JVM).

OpenDayLight построен из пакетов OSGi и платформы сборки - Java Karaf на основе инфраструктуры Apache Felix Framework или Eclipse Equinox OSGi использует Maven как инструмент сборки.

Apache Karaf обеспечивает установку функций Karaf, и входит в программное обеспечение платформы OpenDayLight. По умолчанию OpenDayLight не имеет предварительно установленных функций.

OSGi – Java-специфическая среда, которая улучшает способ взаимодействия Java-классов в рамках одной JVM. Используется для запуска приложений и позволяет динамически подключать плагины для использования и связи различных Southbound протоколов.

Как Karaf, так и OSGi обеспечивают определенный уровень изоляции с явными границами кода, импортом пакетов, экспортом пакетов и другими функциями, связанными с безопасностью.

Архитектурные принципы OpenDayLight:

– модульность и расширяемость – модульный, расширяемый контроллер поддерживает установку, удаление и обновление используемых служб, во время работы (без выключения);

– множественная поддержка Southbound протоколов – использование различных сетевых протоколов (OpenFlow, BGP и т.д.) через плагины SB протокола, обеспечение единого набора услуг и API для приложений через общий набор NB API-интерфейсов (Rest API, OSGi framework);

– служебный абстрактный уровень (Service Abstraction Layer) – использование различных Northbound протоколов;

– consistent clustering (распределенная версия) – предоставляет отказоустойчивость и поддержку распределенной версии (High Avalability Model), позволяет развертывать контроллер на нескольких физических

серверах, гарантируя тем самым отказоустойчивость и балансировку нагрузки;

– разделяемость – управление различными частями сети, при помощи разных компонентов контроллера.

Архитектура OpenDayLight представлена на рисунке 23.

Рисунок 23 – Архитектура OpenDayLight

Последний релиз Oxygen полностью поддерживает протокол OpenFlow 1.3, BGP-LS, протокол OVSDB, PCEP, SNMP. Есть дополнительная поддержка OVSDB protocol, следовательно, может использовать все особенности и расширения Open vSwitch [27].

3.2 Open vSwitch коммутатор

Виртуальный коммутатор – ключевой компонент для виртуализации сетевой инфраструктуры. Он соединяет виртуальные сетевые адаптеры с физическими сетевыми адаптерами, установленными на сервере, устанавливает связь между виртуальными сетевыми адаптерами для локального взаимодействия в рамках сервера.

На данный момент существует три наиболее популярных виртуальных коммутатора – VMware virtual switch (standard и distributed), Cisco Nexus 1000v и Open vSwitch.

Open vSwitch – наиболее универсальное решение. Многоуровневый виртуальный коммутатор с открытым исходным кодом, разрабатываемый под лицензией Apache 2.0. Полностью управляемый, независимый коммутатор, который реализует виртуальную коммутацию на базе протокола OpenFlow. Поддержка OpenFlow предоставляет возможность для интеграции промышленных облачных систем с существующей ПКС. Совместим с популярными платформами для коммутации за счет программных дополнений.

Поддерживает наиболее популярные гипервизоры, включая KVM, Virtual Box, Xen и XenServer. Open vSwitch состоит из службы-коммутатора (user-space) и сопутствующего модуля ядра (kernelspace), который управляет процессом поточной (flowbased) коммутации.

Open vSwitch поддерживает широкий набор технологий, включая NetFlow, sFlow, port mirroring, VLAN, LACP, TLS/SSL.

В рамках классической модели SDN для управления OVS (формирования FIB) применяются сторонние компоненты. например, плагин для OpenStack Neutron или SDN-контроллер OpenDayLight.

Также OVS можно использовать в режиме стандартного коммутатора, без внешнего управляющего элемента и применять MAC learning для формирования таблиц коммутации.

3.3 Разработка сегмента сети SDN

Разработка сегмента SDN проводилась с помощью технологии виртуализации – Oracle VM VirtualBox (VB). VirtualBox - это универсальный полнофункциональный виртуализатор для оборудования x86, предназначенный для использования на сервере, настольном компьютере и встраиваемых системах [25].

Обзор возможностей Oracle VM VirtualBox представлены в таблице 8.

Таблица 8 – Возможности Oracle VM VirtualBox

Переносимость	Oracle VM VirtualBox работает на большом количестве 32-битных и 64-битных хост-ОС. Это так называемый размещенный гипервизор, иногда называемый гипервизором типа 2. В то время как гипервизор типа 1 будет работать непосредственно на оборудовании, Oracle VM VirtualBox требует установки существующей ОС. Таким образом, он может работать вместе с существующими приложениями на этом хосте.
Не требуется аппаратная виртуализация	Для многих сценариев Oracle VM VirtualBox не требует функций процессора, встроенных в новое оборудование, такое как Intel VT-x или AMD-V. В отличие от многих других решений для виртуализации, вы можете использовать Oracle VM VirtualBox даже на старом оборудовании, где эти функции отсутствуют.
Гостевые дополнения: общие папки, бесшовные окна, 3D виртуализация	Гостевые дополнения Oracle VM VirtualBox представляют собой пакеты программного обеспечения, которые можно устанавливать внутри поддерживаемых гостевых систем для повышения их производительности и обеспечения дополнительной интеграции и связи с хост-системой. После установки Guest Additions виртуальная машина будет поддерживать автоматическую настройку разрешений видео, бесшовных окон, ускоренной трехмерной графики и многого другого.

Продолжение таблицы 8

Многопоколенные разветвленные снимки	Oracle VM VirtualBox может сохранять произвольные снимки состояния виртуальной машины. Вы можете вернуться назад во времени и вернуть виртуальную машину к любому такому снимку и запустить альтернативную конфигурацию виртуальной машины, создав эффективное дерево снимков.
ВМ группы	Oracle VM VirtualBox предоставляет функцию групп, которая позволяет пользователю организовывать и контролировать виртуальные машины как коллективно, так и индивидуально. В дополнение к базовым группам любая виртуальная машина также может находиться в более чем одной группе, а группы могут быть вложены в иерархию. Это означает, что вы можете иметь группы групп. Как правило, операции, которые можно выполнять с группами, аналогичны операциям, которые можно применять к отдельным виртуальным машинам: запуск, пауза, сброс, закрытие (состояние сохранения, отправка выключения, отключение питания), сброс сохраненного состояния, отображение в файловой системе, Сортировать.

После установки вы можете запустить Oracle VM VirtualBox следующим образом:

- на хосте Windows в меню «Программы» выберите элемент в группе VirtualBox. В Vista или Windows 7 вы также можете войти VB в поле поиска в меню «Пуск»;
- на хосте Mac OS X в Finder дважды щелкните элемент VirtualBox в папке «Программы». Вы можете перетащить этот элемент на свой док;
- на хосте Linux или Oracle Solaris, в зависимости от среды рабочего стола, элемент Oracle VM VirtualBox может быть помещен в группу

«Система» или «Системные инструменты» меню «Приложения». Кроме того, вы можете войти VB в окно терминала.

Окно VirtualBox Manager, после первоначального запуска представлено на рисунке 24.

Окно VirtualBox Manager, после создания виртуальных машин представлено на рисунке 25.

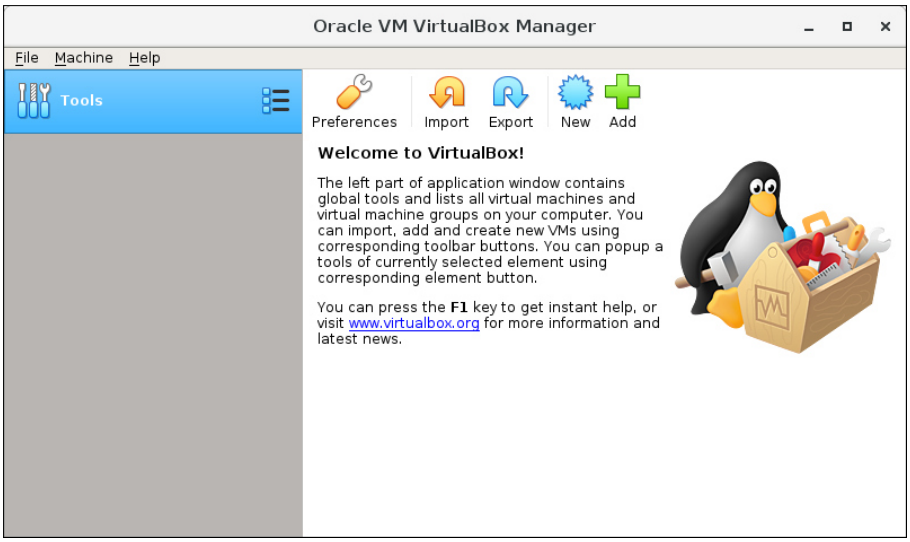

Рисунок 24 – Первоначальное окно VirtualBox Manager

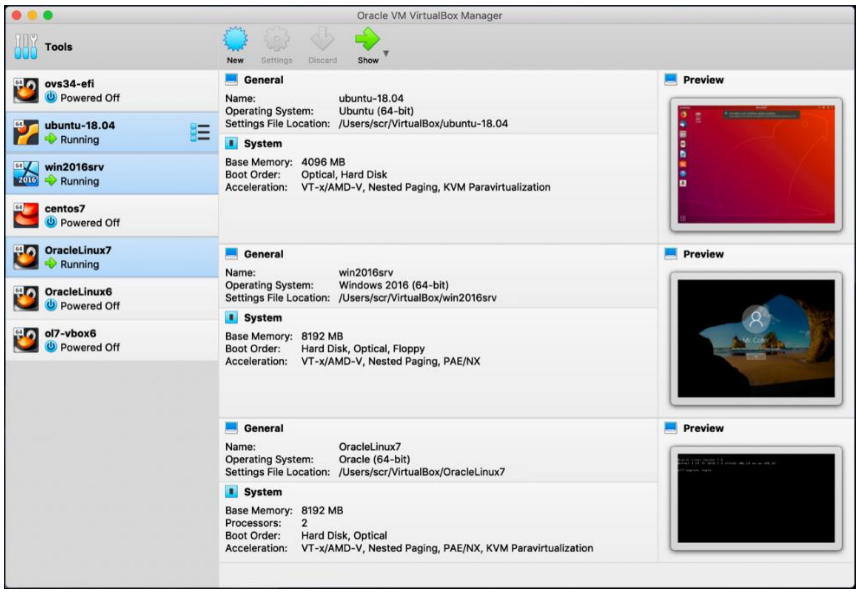

Рисунок 25 – Окно VirtualBox Manager, после создания виртуальных машин

Группы виртуальных машин позволяют пользователю создавать специальные группы виртуальных машин, а также управлять и выполнять функции на них как коллективно, так и индивидуально.

На рисунке 26 показаны группы виртуальных машин, отображаемые в VirtualBox Manager.

Сегмент был смоделирован в виртуальной среде с несколькими виртуальными машинами (VM). Сервер Ubuntu 16.04.1 LTS 64-бит был выбран в качестве гостевой ОС. Для проекта использовались 2 виртуальные машины: 1 для контроллера OpenDayLight, 1 для Open vSwitch.

Задачи моделирования:

а) разработка модели сегмента SDN сети, способной передавать исходный трафик;

б) тестирование внедрения механизмов защиты для обеспечения безопасности передачи трафика.

Технические характеристики оборудования, используемые при моделировании представлены в таблице 9.

Таблица 9 – Описание объектов модели

Номер ВМ	ОС	HW	Назначение ВМ
1	Ubuntu 16/04 LTS x64	4 CPU	Контроллер ODL
2	Ubuntu 16.04 LTS x64	4 CPU	OVS

Для настройки конфигурации сети SDN определяется пространство IP адресов. Используется внутренняя сеть. Интерфейсы контроллера и коммутаторов настраиваются в соответствии с таблицей для обеспечения связи между устройствами Open vSwitch. Адресация в топологии сети представлена в таблице 10. Схема топологии сегмента сети SDN представлена на рисунке 27.

Таблица 10 – Адресация в топологии сети

Элемент	IP Адрес	Версия
OpenDayLight SDN контроллер	Ens160 192.168.31.163/23 Ens192 192.168.31.69/23	ODL ver Nitrogen
Open vSwitch SW1	Ens160 192.168.31.131/23 Ens256 192.168.31.27/23	OF v1.3 OVS 2.5.4
Сеть лаборатории Б329	192 192.168.30.0	–

Виртуальные машины имеют несколько интерфейсов. Интерфейс ens160 используется для связи с контроллером и коммутатора, интерфейс коммутатора ens256 используется для связи с VirtualBox Switch, интерфейс контроллера ens192 используется для связи с VirtualBox Switch и сетью лаборатории.

Модель топологии сети отображает сегмент SDN с контроллером ODL, программируемым коммутатором OVS и соединением с сетью лаборатории Б329.

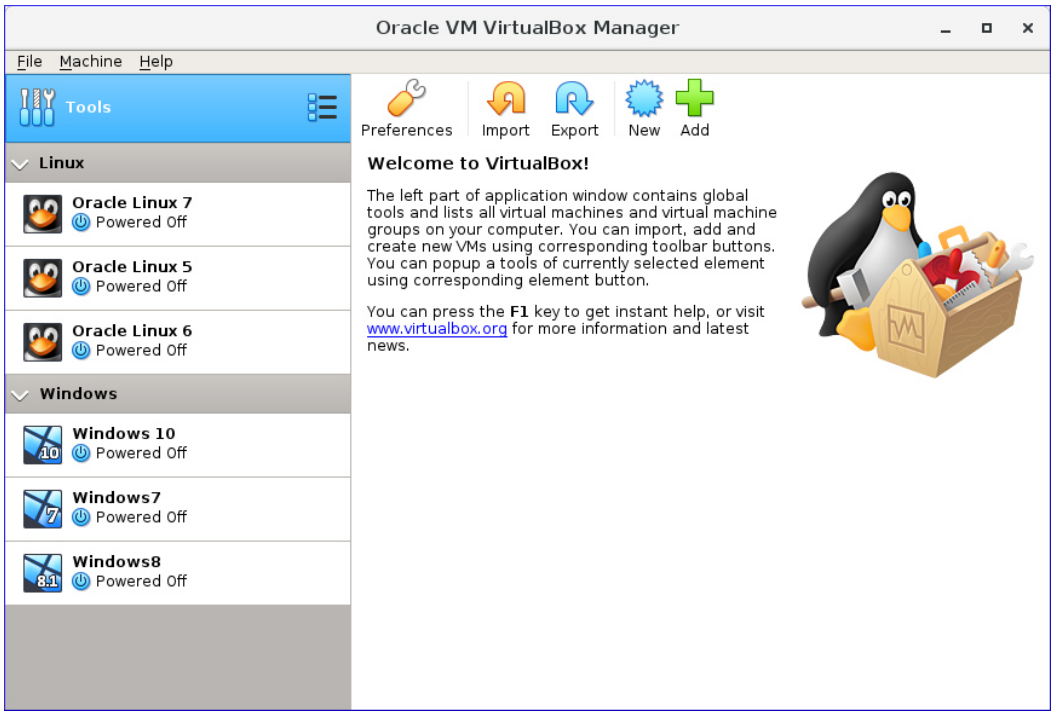

Рисунок 26 – Группы виртуальных машин в VB Manager

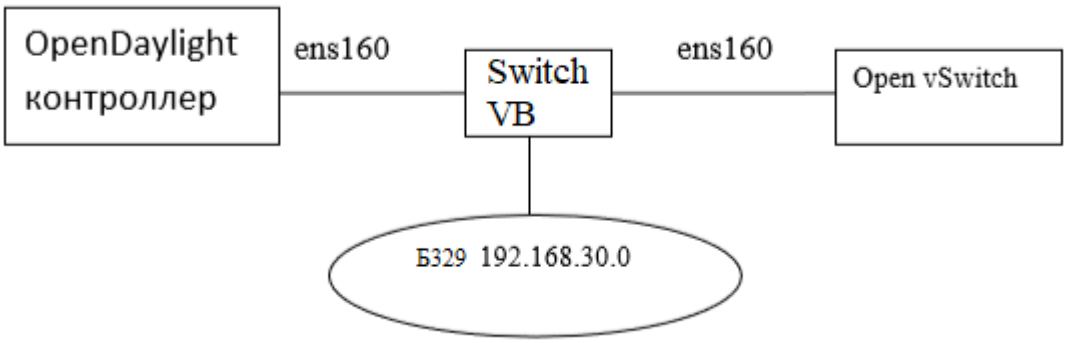

Рисунок 27 – Схема топологии сегмента сети SDN

3.4 Документирование конфигурации сетевого оборудования

В ходе выполнения моделирования, было установлено программное обеспечение контроллера OpenDayLight, Open vSwitch и произведены изменения в конфигурациях коммутатора и контроллера. Далее приведены основные настройки для каждого сетевого элемента.

Установка и настройка контроллера OpenDayLight. Контроллер OpenDayLight Nitrogen SR-3 был выбран, поскольку он поддерживает версии OpenFlow1.0 и 1.3, которые совместимы с версиями Open vSwitch. Перед установкой контроллера требуется установить, необходимые для работы библиотеки. CLI интерфейс контроллера OpenDayLight представлен на рисунке 28.

JDK (JavaDevelopmentKit) – бесплатно распространяемый компанией OracleCorporation комплект разработчика приложений на языке Java, включающий в себя компилятор Java (javac), стандартные библиотеки классов Java, примеры, документацию, различные утилиты и исполнительную систему Java (JRE) (приложение А).

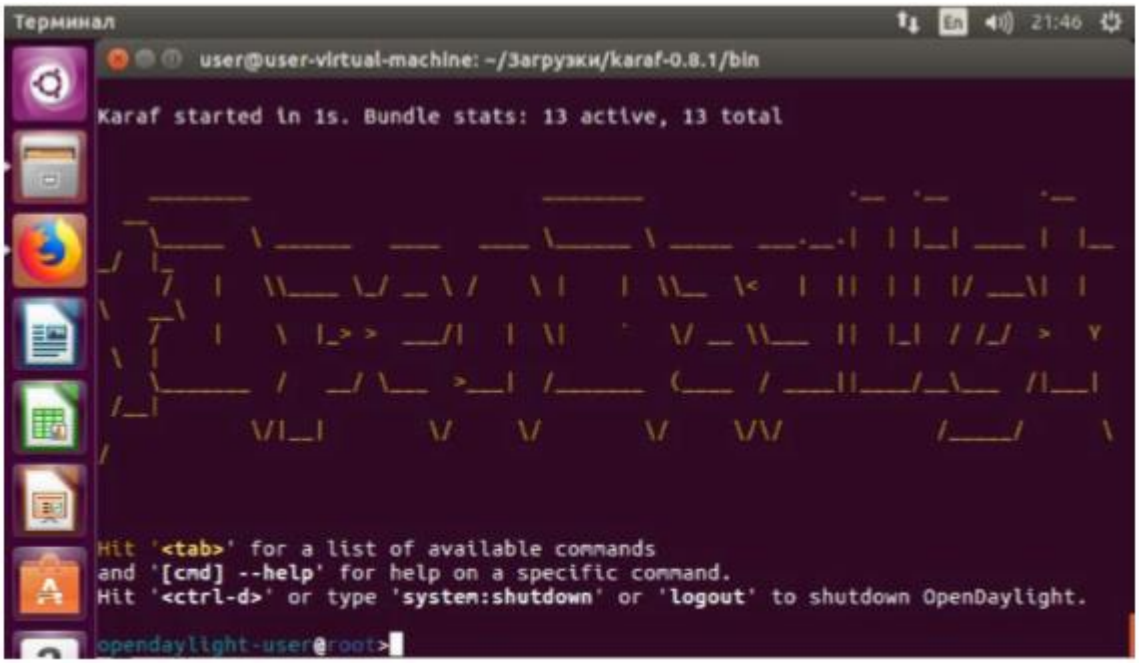

Рисунок 28 – CLI интерфейс контроллера OpenDayLight

Для обеспечения полного функционала контроллера, необходимо установить дополнительные функции с помощью CLI.

Синтаксис команды установки функций:

Feature: install [MODULE_NAME]

Синтаксис команды проверки установленных функций:

Feature: list –i | grep [MODULE_NAME]

Перечень и назначение функций приведены в таблице 11.

Таблица 11 – Функции контроллера OpenDayLight

Функция	Назначение
odl-l2switch-switch-all	Коммутация уровня L2
odl-restconf	API REST для приложений. Обеспечивает «GET», «PUT», «DELETE» запросы и поддержку HyperText Transfer Protocol (HTTP).
odl-mdsal – apidocs	Предоставляет apidocs explorer и перечисляет все доступные API на контроллере.
odl-dlux-all odl-dlux-core odl-dluxapps-nodes odl-dluxapps-topology odl-dluxapps-yangutils odl-dluxapps-yangui odl-dluxapps-yangvisualizer odl-dluxapps-yangman	DLUX отвечает за графический интерфейс пользователя (GUI) в контроллере OpenDayLight.устанавливает GUI-функции для контроллера OpenDayLight.

Контроллер OpenDayLight прослушивает порт 6633 OpenFlow для подключения к его узлам. Команда netstat –a | grep 6633 увидеть состояние порта. Проверка состояния OpenFlow порта контроллера представлена на рисунке 29.

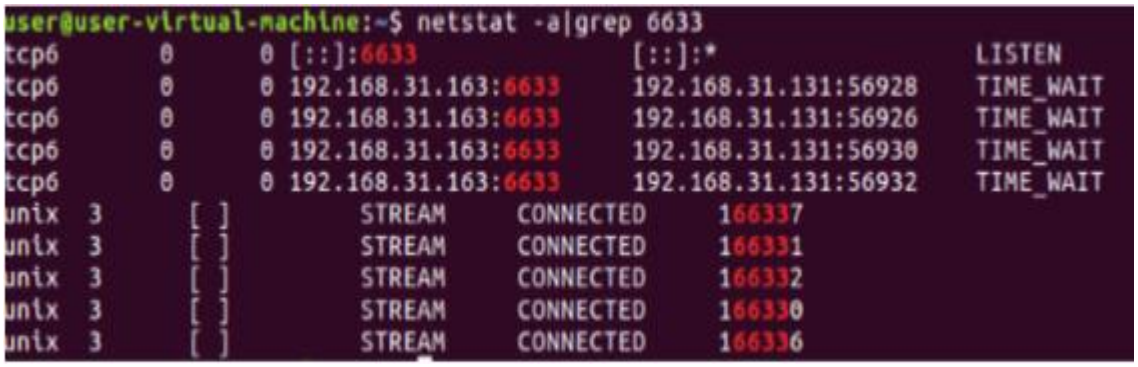

Рисунок 29 – Проверка состояния OpenFlow порта контроллера

Установка и настройка OVS. Для создания соединения между ВМ 1 и ВМ 2. На виртуальную машину 2 был установлен программируемый модульный коммутатор Open vSwitch. А также для настройки связи с контроллером создан интерфейс моста. (приложение Б, В). Для сохранения конфигурации интерфейсов, настройки сохранялись в файле «Interface», расположенном в папке «/ etc / network /» в системном корне.

Конфигурация всех Open vSwitch коммутаторов, портов, настройки поддерживаемых протоколов хранятся в собственной базе данных OVS. Конфигурация интерфейсов OVS представлена на рисунке 30. Утилита ovs-vsctl предоставляет интерфейс для внесения изменений в эту БД.

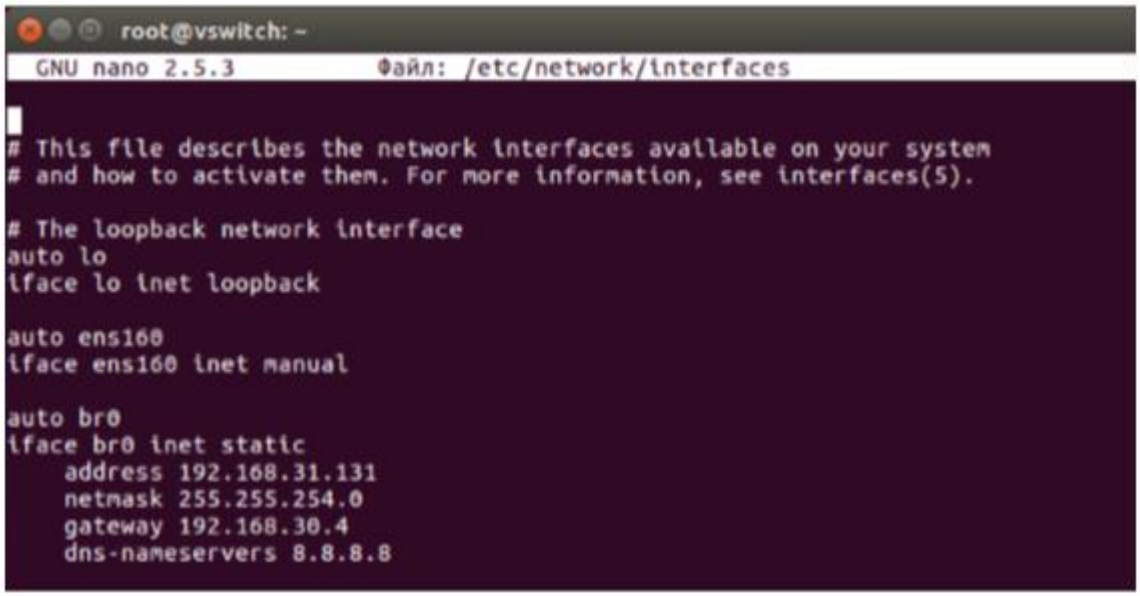

Рисунок 30 – Конфигурация интерфейсов OVS

Перечень используемых технологий представлен в таблице 12.

Как было отмечено во второй главе. Для обеспечения безопасности контроллера необходим безопасный канал связи. Коммутатор и контроллер должны пройти аутентификацию, чтобы избежать подключения скомпрометированных устройств.

В контексте нашей структуры, канал шифруется с использованием соединения SSL / TLS на основе инфраструктуры открытого ключа (PKI). Чтобы настроить базовую PKI необходимо создать орган по сертификации (CA) и сгенерировать открытый и закрытый ключи контроллера и коммутатора. Для этого служит ovs-pki – утилита для управления инфраструктурой открытого ключа OpenFlow. ovs-pki использует OpenSSL для управления ключами.

Чтобы использовать OVS с SSL, необходимо подключить поддержку SSL - загрузить необходимые пакеты SSL для своего дистрибутива: «sudo apt-get install openssl».

Создание и управление инфраструктурой открытого ключа для коммутаторов OpenFlow, выполняется с помощью команд ovs-pki. Синтаксис команд: Ovs-pki [опции] команда [args]. Инициализация PKI представлена на рисунке 31.

Реализованные команды и их аргументы приведены в приложении В, таблица В.1.

Таблица 12 – Перечень используемых технологий

Технология	Настраиваемые элементы	Описание
Fail-mode secure	OVS	«отказоустойчивый» режим моста – только контроллер отвечает за пересылку пакетов, если контроллер выключен все пакеты будут удалены.
Fail standalone		Автономный: OVS будет пересылать пакеты, если контроллер не работает
SSL	OVS ODL	Протокол защиты транспортного уровня – криптографические протоколы, обеспечивающие защищенную передачу данных между узлами в сети
OpenSSL	OVS ODL	Криптографический пакет с открытым исходным кодом для работы с SSL/TLS. Позволяет создавать ключи RSA, DH, DSA и сертификаты X.509, подписывать их. Также имеется возможность шифрования данных и тестирования SSL/TLS соединений, для управления сертификатами и генерации ключей.

Рисунок 31 – Инициализация PKI

Каталог pki расположен в директории var/lib/openvswitch/pki и содержит два важных подкаталога Controllerca и Switchca. Каталоги PKI представлены на рисунке 32.

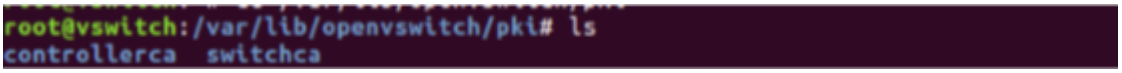

Рисунок 32 – Каталоги PKI

Подкаталоги:

а) Controllerca содержит CA – файл контроллера:

– Ctl-cert.pet – корневой сертификат для органа сертификации контроллера. Каждый Open vSwitch должен иметь копию того файла, чтобы он мог аутентифицировать допустимые контроллеры с помощью проверки подписи этого сертификата, одинаковый для всех коммутаторов в данном административном блоке.

– Ctl-privkey.pet – закрытый ключ подписи для органа сертификации контроллера. Этот файл должен храниться в секрете. Нет необходимости в том, чтобы коммутаторы или контроллеры имели копию.

б) Подкаталог Switchca – содержит CA файлы коммутатора:

– SC-cert.pem – файл корневого сертификата коммутатора, подписанный собственным личным ключом коммутатора. Удостоверяющий, что закрытый ключ является надежным. Файл должен быть сгенерирован на машине с закрытым ключом для центра сертификации коммутатора, в идеале, это должна быть машина, которая вообще не подключена к сети.

Его содержание не является секретом. Контроллер OpenFlow должен иметь этот файл, чтобы он мог аутентифицировать действительные коммутаторы.

– SC-privkey.pem – файл закрытого ключа коммутатора, который содержит частную половину ключа RSA или DSA. Закрытый ключ подписи для центра сертификации сертификатов. Этот файл может быть сгенерирован в самом коммутаторе, для обеспечения максимальной безопасности, или может быть сгенерирован в другом месте и скопирован в Open vSwitch. Для обеспечения безопасности содержимое этого файла должно оставаться секретным. Обычно нет необходимости копировать этот файл с Open vSwitch на коммутаторы или контроллеры.

Создание пар ключей для коммутатора и контроллера представлено на рисунке 33.

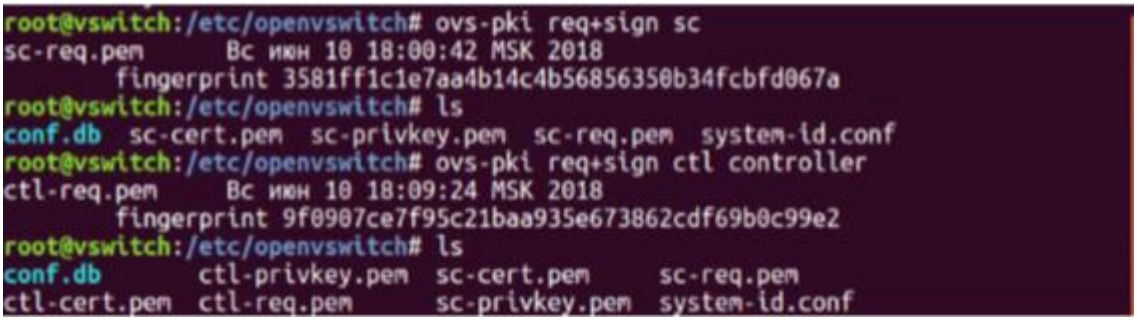

Рисунок 33 – Создание пар ключей для коммутатора и контроллера

В ходе создания пары ключей и сертификатов для контроллера и коммутатора были сгенерированы шесть файлов «.pem», из которых мы будем использовать четыре. Закрытый ключ для каждой стороны, подписанный неофициальным корневым сертификатом. Публичный сертификат для каждой стороны. Существуют также два сертификата запроса sc-req.pem и ctl-req.pem, которые использовались для создания подписанных сертификатов.

Закрытый ключ контроллера и сертификат в файлах ctl-privkey.pem и ctlcert.pem, необходимо скопировать в контроллер. Очень важно убедиться, что не созданы копии ctl-privkey.pem, поскольку они могут использоваться для выдачи себя за легитимный хост.

Для переноса закрытого ключа и сертификата контроллера на контроллер создано хранилище ключей в формате PKCS12, содержащего закрытые и открытые ключи контроллера. Экспорт хранилища ключей и сертификата контроллера в формат PKCS#12 представлен на рисунке 34.

```
root@vswitch:~# openssl pkcs12 -export -in ctl-cert.pem -inkey ctl-privkey.pem \
> -out ctl.p12 -name odlserver \
> -CAfile /var/lib/openvswitch/pki/controllerca/cacert.pem -caname root -chain
Enter Export Password:
Verifying - Enter Export Password:
```

Рисунок 34 – Экспорт хранилища ключей и сертификата контроллера в формат PKCS#12

Результатом является новый файл ctl.p12. Новый формат хранилища представлен на рисунке 35.

```
root@vswitch:~# ls /etc/openvswitch
conf.db       ctl.p12          ctl-req.pem  sc-privkey.pem  system-id.conf
ctl-cert.pem  ctl-privkey.pem  sc-cert.pem  sc-req.pem
```

Рисунок 35 – Новый формат хранилища

OVS теперь имеет открытый и закрытый ключ, а также имеет копию сертификата контроллера. Необходимо предоставить копию сертификата контроллеру, чтобы он мог проверить открытый ключ OVS при квитировании TLS / SSL. Скопировать файл ctl.p12 на контроллер и импортировать его в существующее хранилище ключей Java. Ключевые форматы хранилища ключей Java и OVS отличаются друг от друга, поэтому нужно преобразовать сертификат в формат хранилища ключей контроллера. Для этого экспортируйте хранилище ключей PKCS12 в формат JKS. Перевод в формат jks представлен на рисунке 36.

```
root@user-virtual-machine:/etc/openvswitch# keytool -importkeystore          -de
tstorepass opendaylight -destkeypass opendaylight -destkeystore ctl.jks
-srckeystore ctl.p12 -srcstoretype PKCS12 -srcstorepass opendaylight          -a
ias odlserver
Importing keystore ctl.p12 to ctl.jks...

Warning:
The JKS keystore uses a proprietary format. It is recommended to migrate to PKC
12 which is an industry standard format using "keytool -importkeystore -srckeys
ore ctl.jks -destkeystore ctl.jks -deststoretype pkcs12".
root@user-virtual-machine:/etc/openvswitch#
```

Рисунок 36 – Перевод в формат jks

Рисунок 37 – Создание хранилища ключей

Создание хранилища ключей представлено на рисунке 37.

В результате, были созданы три новых файла для каждого элемента - открытый ключ, закрытый ключ и сертификат, который является самозаверенным. Хранилище ключей, которое действует как репозиторий для пары ключей контроллеров. Хранилище доверия, в котором хранится ключ коммутатора.

В завершении, необходимо настроить ovs-vswitchd для использования SSL. Нужно указать OVS на его новые генерируемые общедоступные и закрытые ключи вместе с сертификатом контроллера «cacert.pem».

ovs-vsctl set-ssl /etc/openvswitch/sc-privkey.pem \

/etc/openvswitch/sc-cert.pem /etc/openvswitch/cacert.pem

На этом этапе конфигурация контроллера и коммутатора OVS для установления безопасного соединения готова.

3.5 Демонстрация работоспособности оборудования

Для демонстрации работы оборудования использовался ряд консольных команд, вывод которых свидетельствует о правильности конфигурации. А также графический интерфейс контроллера OpenDayLight.

С целью исследования трафика, проходящего через активное сетевое оборудование, применен анализатор трафика Wireshark (версия 1.12.5).

Проверим доступность интерфейса контроллера командой ping. Проверка соединения контроллера и коммутатора представлена на рисунке 38.

```
root@vswitch:~# ping -c 4 192.168.31.163
PING 192.168.31.163 (192.168.31.163) 56(84) bytes of data.
64 bytes from 192.168.31.163: icmp_seq=1 ttl=64 time=0.572 ms
64 bytes from 192.168.31.163: icmp_seq=2 ttl=64 time=0.524 ms
64 bytes from 192.168.31.163: icmp_seq=3 ttl=64 time=0.510 ms
64 bytes from 192.168.31.163: icmp_seq=4 ttl=64 time=0.577 ms

--- 192.168.31.163 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3056ms
rtt min/avg/max/mdev = 0.510/0.545/0.577/0.040 ms
```

Рисунок 38 – Проверка соединения контроллера и коммутатора

Ping выполнен успешно, связь между контроллером и коммутатором осуществляется.

Команда ovs-vsctl show выводит информацию об интерфейсах моста, соединении с контроллером, и версии Open vSwitch. В строке controller отображен тип установленного соединения. Для текущей конфигурации SSL. Вывод состояния интерфейсов OVS представлен на рисунке 39.

Команда ovs-vsctl list controller выводит информацию обо всех подключенных контроллерах. Вывод всех подключенных контроллеров представлен на рисунке 40.

IP-адрес контроллера SDN, для подключения к коммутатору 192.168.31.188.

Тип подключения – SSL и порт для связи - порт 6633.

Значение is_connected = true подтверждает, что соединение успешно.

```
root@vswitch:~# ovs-vsctl show
7855e23c-7d05-4a39-abd9-64dbbe39c6f0
    Bridge "br0"
        Controller "ssl:192.168.31.163:6633"
            is_connected: true
        Port "ens160"
            Interface "ens160"
        Port "br0"
            Interface "br0"
                type: internal
    ovs_version: "2.5.4"
```

Рисунок 39 – Вывод состояния интерфейсов OVS

Рисунок 40 – Вывод всех подключенных контроллеров

Для мониторинга коммутатора OpenFlow используется инструмент командной строки. Утилита ovs-ofctl используется для сбора статистики по портам, таблицам или потокам и управления OpenFlow-потоками OVS. Различные параметры мониторинга могут использоваться в соответствии со списком ниже. Функции мониторинга OVS представлены в таблице 13.

Сопоставление интерфейсов моста OpenFlow – портам, представлено в таблице 14.

Вывод статистики через CLI Open vSwitch представлен на рисунке 41.

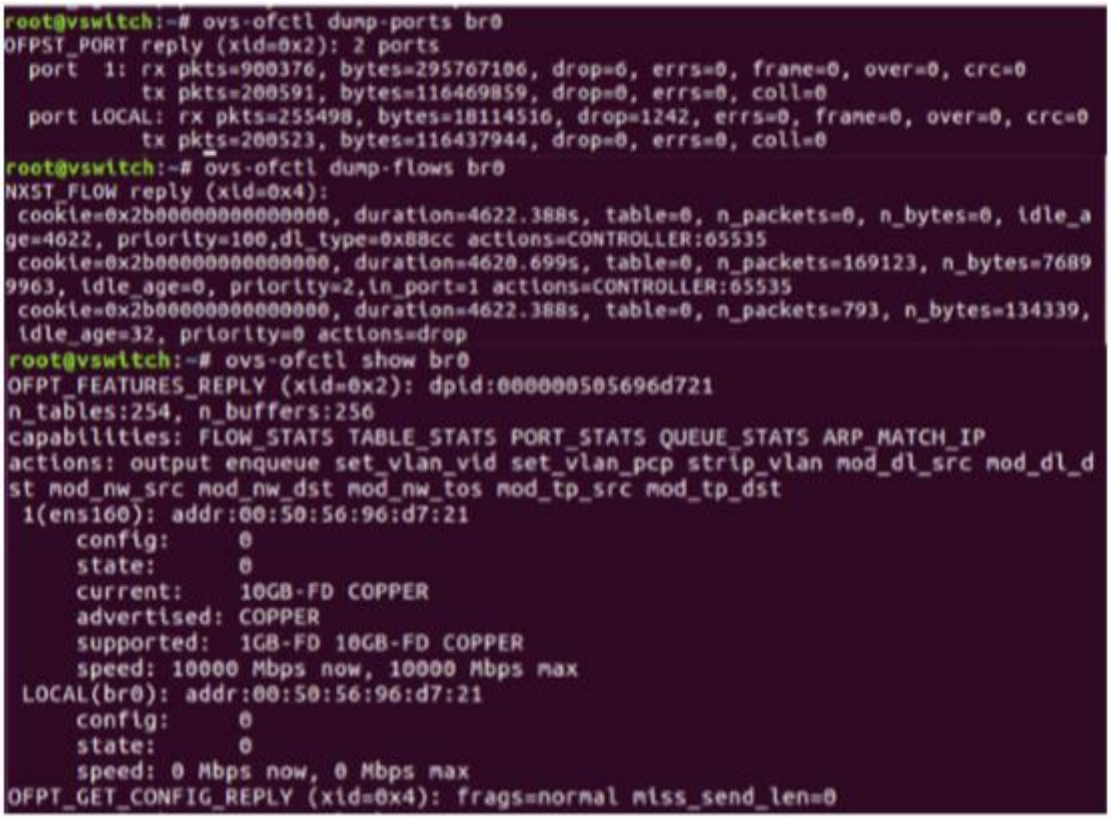

Рисунок 41 – Вывод статистики через CLI Open vSwitch

Таблица 13 – Функции мониторинга OVS

Команда	Функция
show -O <версия openflow> <bridge> <if> show -O openflow13 br0 LOCAL / eth1 / eth2	Проверяет порты
dump-tables -O <версия openflow> <мост> dump-tables -O openflow13 br0	Распечатывает статистику консоли для каждой из таблиц потоков, используемых коммутатором.
dump-flow -O <версия openflow> <мост> dump-flow -O openflow13 br0	Выводит все записи потока в таблицах коммутаторов, которые соответствуют потокам
ovs-vsctl get Interface eth0 ofport	Таблица интерфейсов в базе данных Open vSwitch также отображает имена портов OpenFlow в числа. Чтобы напечатать номер порта OpenFlow, связанный с интерфейсом.
dump-ports -O <версия openflow> <bridge> <netdev> dump-ports -O openflow13 of-switch eth0	Распечатывает статистику консоли для сетевых устройств, связанных с коммутатором. Если netdev будет указана только статистика, связанная с этим устройством. NETDEV может быть номером назначенного порта OpenFlow или именем устройства, например.
ovs-ofctl show br0	Сопоставление между именами портов OpenFlow и номерами
ovs-appctl fdb/show <bridge_name>	Посмотреть таблицу MAC - адресов

Таблица 14 – Сопоставление интерфейсов моста OpenFlow – портам

Интерфейс	MAC	Номер порта OpenFlow
Br0	00:50:56:96: d7:21	429496
Ens160	00:50:56:96: d7:21	1

Кроме того, для сопоставления конфигурации используется GUI OpenDaylight. Графический интерфейс включает в себя меню «Yang UI» «Topology» «Yang Visualizer» «Nodes» «Yangman».

Для доступа к GUI в адресную строку веб-браузера на хосте контроллера необходимо скопировать ссылку доступа к GUI контроллера ODL. Авторизация для доступа к GUI представлена на рисунке 42.

Ссылка доступа «http: // [CONTROLLER_IP]: 8181 / index.html».

Рисунок 42 – Авторизация для доступа к GUI

С помощью вкладки «Topology» графического интерфейса контроллера OpenDaylight можно проверить топологию сети и убедится в правильности конфигурации, созданной функциями L2Switch. Меню «Топология» отображает подключенный OVS коммутатор и хосты, находящиеся в сети лаборатории Б329. Отображение топологии сегмента в GUI контроллера представлено на рисунке 43.

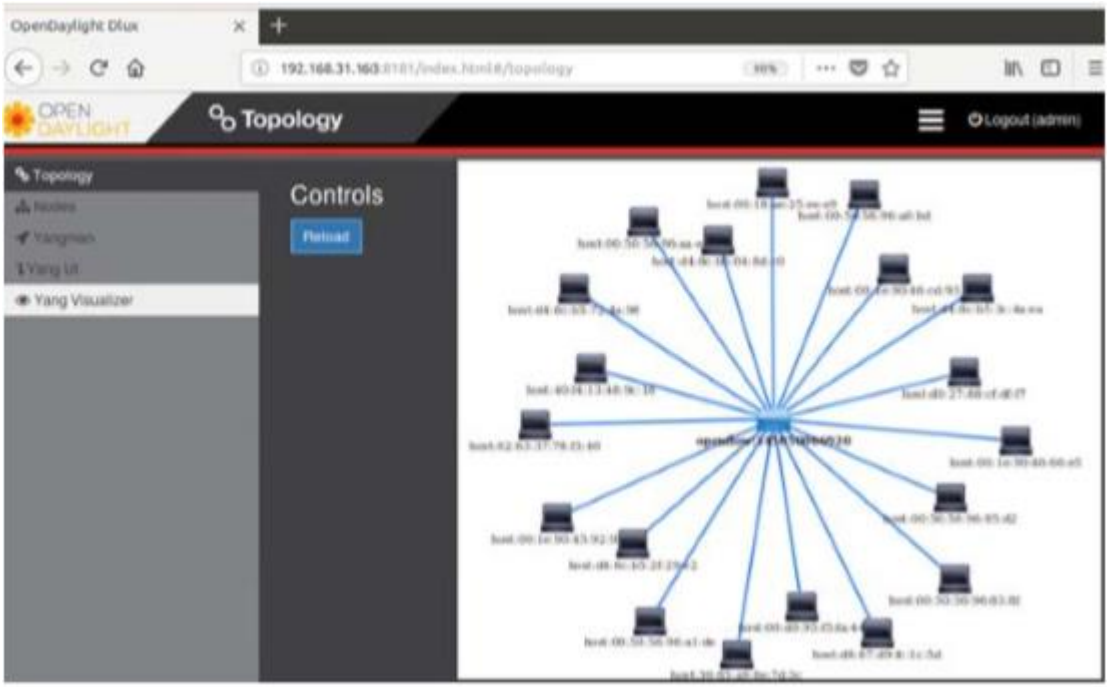

Рисунок 43 – Отображение топологии сегмента в GUI контроллера

Вкладка «Nodes» предоставляет информацию об id узла openflow, интерфейсах узла, их номерах и адресах. А также возможность просмотра статистики интерфейсов, аналогичную статистике CLI Open vSwitch.

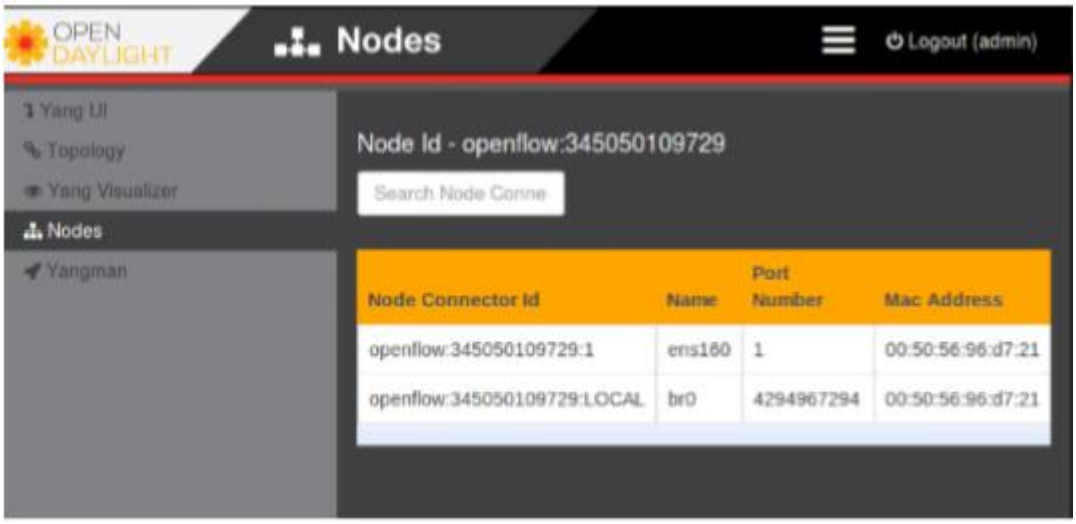

Рисунок 46 – Информация о интерфейсах коммутатора

Рисунок 47 – Вывод статистики с помощью GUI контроллера

Функционал «Yang UI» предоставляет расширенные возможности сбора статистики и получения информации о элементах сети, и их конфигурации с помощью GET POST запросов. Вывод информации возможен в различных форматах: как в графическом представлении, так и виде XML или JSON. Информация о интерфейсах коммутатора представлена на рисунке 46. Вывод статистики с помощью GUI контроллера представлен на рисунке 47.

GET запрос на инвентаризацию узлов представлен на рисунке 48.

Информация о инвентаризации узлов с помощь GET запроса представлена на рисунке 49.

Рисунок 48 – GET запрос на инвентаризацию узлов

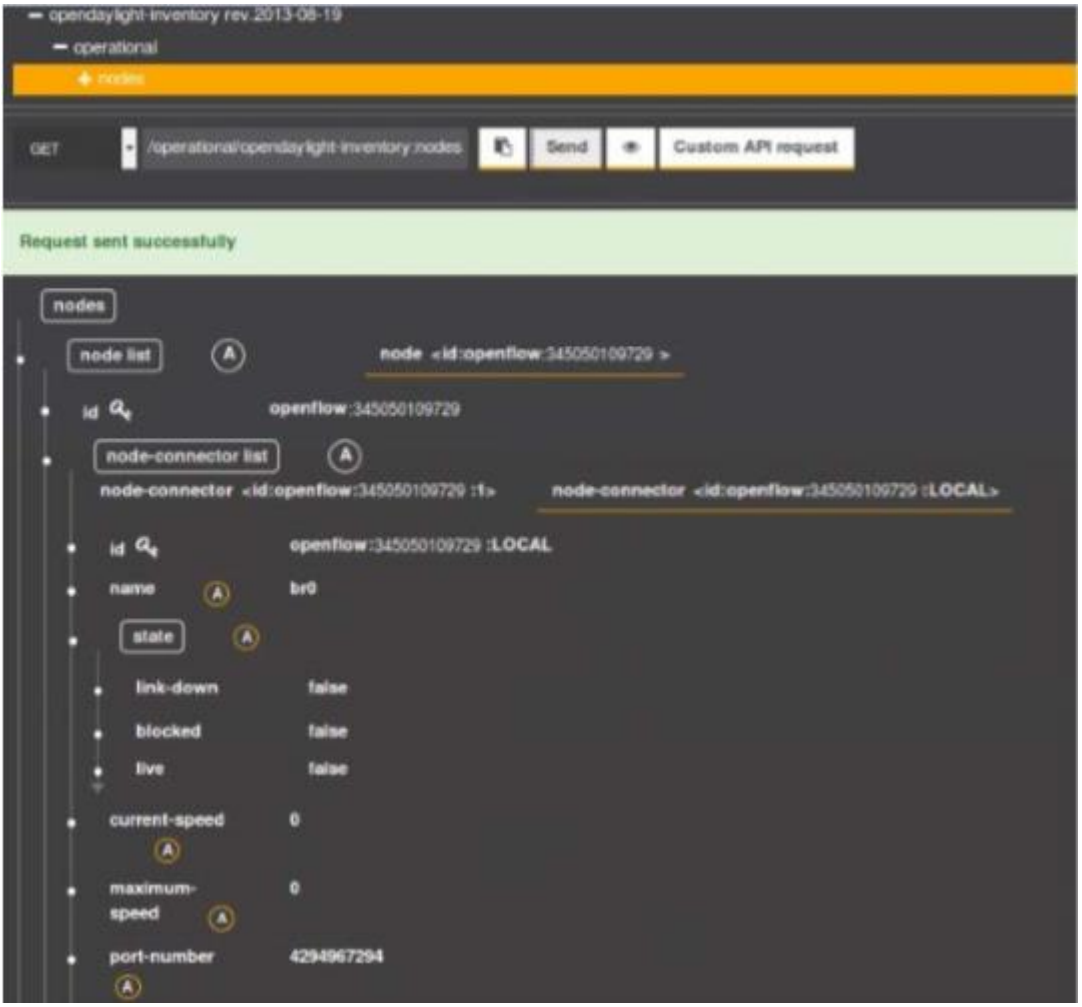

Рисунок 49 – Информация о инвентаризации узлов с помощь GET запроса

Данные статистики, полученные с помощью GUI контроллера, совпадают со статистикой, полученной при помощи CLI Open vSwitch. Безопасный тип соединения вместо TCP соединения по умолчанию установлен успешно.

4 Технико-экономическое обоснование дипломной работы

В данной дипломной работе основой задачей является разработка и обеспечение информационной безопасности в компьютерной сети согласно концепции SDN.

4.1 Определение трудоемкости разработки ПП

Для начала, чтобы определить трудоемкость разработки ПП, составим перечень видов и основных этапов работ, которые будут выполнены. Также уделим внимание последовательности и параллельному выполнению работ, для того чтобы сократить длительность разработки ПП [28].

Распределение работ по этапам, видам и оценка их трудоемкости представлены в таблице 14.

Таблица 14 – Распределение работ по этапам и видам и оценка их трудоемкости

Этап разработки ПП	Сотрудник	Время работы, ч	Количество дней
Составление технического задания на разработку системы	Руководитель	40	5
Покупка оборудования и перевозка на место сборки	Инженер-сетевик	24	3
Установка оборудования и монтаж	Инженер-монтажник	24	3
Разработка сегмента сети SDN, способной передавать исходный трафик	Инженер-сетевик	56	7
Тестирование внедрения механизмов защиты для обеспечения безопасности передачи трафика	Инженер-сетевик	56	7
Подготовка и сдача отчетных материалов	Инженер-сетевик	16	2
ИТОГО			27

4.2 Расчет затрат на разработку ПП

4.2.1 Материальные затраты

Материальные затраты состоят из основных и вспомогательных.

Затраты на основные ресурсы представлены в таблице 15.

Монтаж оборудования, пуско-наладка производится инженерами-монтажниками, расходы составляют 10% от стоимости всего оборудования и рассчитываются по формуле [28]:

$$К_{мон} = К_{обор} \times 0.1, \quad (4.1)$$

$$К_{мон} = 223074 * 0,1 = 22308 \text{ тенге.}$$

Транспортные расходы, составляют 5% от стоимости всего оборудования и рассчитываются по формуле:

$$К_{пер} = К_{обор} \times 0,05, \quad (4.2)$$

$$К_{пер} = 223074 \times 0,05 = 11154 \text{ тенге.}$$

Таблица 15 – Затраты на основные материальные ресурсы

Наименование материального ресурса	Количество израсходованного материала	Цена за единицу, тг	Сумма, тг
Системный блок: i3-8350K +16GB +120SSD +1TBHDD +600W	1	157053	157053
Монитор: Philips 200V4QSBR/00 19.5	1	29580	29580
Клавиатура: Crown CMK02 Black USB	1	1070	1070
Мышь: Ritmix ROM-111 Black USB	1	705	705
Принтер: HP LaserJet Pro M15a A4	1	34666	34666
ubuntu 16.04.1 lts	2	Бесплатно	
OpenDayLight + Open vSwitch	1	Бесплатно	
Oracle VM VirtualBox	1	Бесплатно	
ИТОГО затраты на основные материальные ресурсы			223074

Тогда капитальные затраты составят:

$$К_{кап} = 223074 + 22308 + 11154 = 256536 \text{ тенге.} \quad (4.3)$$

Затраты на вспомогательные ресурсы представлены в таблице 16.

Затраты на электроэнергию для производственных нужд, включают в себя расходы электроэнергии на производственное оборудование и дополнительные нужды.

Стоимость необходимых затрат на электроэнергию для представлена в таблице 17.

Таблица 16 – Затраты на вспомогательные материальные ресурсы

Наименование материального ресурса	Единица измерения	Количество израсходованного материала	Цена за единицу, тг	Сумма, тг
Бумага А4: SvetoCopy, A4, 80 гр/м2, 500 листов	пачка	1	1305	1305
Картридж: Sprint SP-H-CF244A (HP 44A (CF244A))	шт.	1	10480	10480
ИТОГО затраты на вспомогательные материальные ресурсы				11785

Таблица 17 – Затраты на электроэнергию

Наименование	Паспортная мощность, кВт	Коэффициент использования мощности	Время работы оборудования для разработки ПП, ч	Цена электроэнергии тенге/кВч*ч	Сумма, тг
Системный блок + клавиатура + мышь	0.6	0,9	192	17	1763
Монитор	0,014	0,9	192	17	42
Лазерный принтер	0,21	0,9	8	17	26
ИТОГО затраты на электроэнергию					1831

4.2.2 Расчет фонда оплаты труда

Для вычисления ФОТ приведем среднемесячную заработную плату работников, которую сведем в таблицу 18.

Фонд оплаты труда включают в себя затраты на основную и дополнительную заработную плату и рассчитывается по формуле [28]:

$$ФОТ = З_{осн} + З_{доп}. \quad (4.4)$$

Основная заработная плата определяется как сумма оплаты труда всех исполнителей по формуле:

$$З_{осн} = \sum_{i=1}^{n} З_i \times Т_i, \quad (4.5)$$

где $З_i$ – зарплата i – го работника в день, тенге;

$Т_i$ – затраты времени i-го работника, дней.

Таблица 18 – Количество задействованных в проекте работников, и их заработная плата

Исполнитель	Количество, человек	Месячная заработная плата, тенге
Руководитель	1	200000
Инженер-сетевик	1	150000
Инженер-монтажник	1	100000
Итого:	3	450000

Дополнительная заработная плата составляет 20% от основной заработный платы:

$$З_{доп} = 0.2 \times З_{осн}. \quad (4.6)$$

Так как участники, задействованные в проекте, работают в различные промежутки времени выделенного на реализацию проекта, необходимо установить этапы проектирования и рассчитать количество рабочих дней каждого задействованного работника.

Разработка системы состоит из 6 этапов, общей продолжительностью 27 дней. Согласно Статье 68 Трудового кодекса Республики Казахстан предусмотрено, что нормальная продолжительность рабочего времени не должна превышать 40 часов в неделю, при пятидневной рабочей неделе.

Этапы и количество рабочих дней, выделенных на проектирование системы предоставлены в таблице 19.

Стоимость человека-дня вычисляется по формуле

$$Д = \frac{ЗПм}{Др}, \quad (4.7)$$

где ЗПм – заработная плата за месяц, тенге;

Др – среднемесячное количество рабочих дней (среднемесячное количество рабочих дней – 24).

Таблица 19 – Этапы реализации проектирование сети

Этап разработки ПП	Сотрудник	Время работы, ч	Кол-во дней
Составление технического задания на разработку системы	Руководитель	40	5
Покупка оборудования и перевозка на место сборки	Инженер-сетевик	24	3
Установка оборудования и монтаж	Инженер-монтажник	24	3
Разработка сегмента сети SDN, способной передавать исходный трафик	Инженер-сетевик	56	7
Тестирование внедрения механизмов защиты для обеспечения безопасности передачи трафика	Инженер-сетевик	56	7
Подготовка и сдача отчетных материалов	Инженер-сетевик	16	2
ИТОГО			27

Дневная зарплата инженера-монтажника, соответствии с формулой (4.7) будет:

$$Д = \frac{100000}{24} = 4167 \text{ тенге.}$$

Дневная зарплата для инженера-сетевика, согласно формуле (4.7) будет:

$$Д = \frac{150000}{24} = 6250 \text{ тенге.}$$

Дневная зарплата для руководителя, согласно формуле (4.7) будет:

$$Д = \frac{200000}{24} = 8334 \text{ тенге.}$$

На основе рассчитанных данных одного человека дня, рассчитаем затраты на оплату труда для каждой категории работников. Трудозатраты представлены в таблице 20.

Таблица 20 – Трудозатраты

Наименование должностей	Дневная зарплата, тенге	Количества дней	Сумма, тенге
Инженер-сетевик	6250	19	118750
Инженер-монтажник	4167	3	12501
Руководитель	8334	5	41670
Итого			172921

Дополнительная заработная плата составляет 20% от основной заработной платы, согласно формуле (4.6) будет

$$З_{доп} = 0{,}2 \times 172921 = 34585 \text{ тенге.}$$

Суммарный фонд оплаты труда (ФОТ), согласно формуле (4.4) будет

$$ФОТ = 172921 + 34585 = 207506 \text{ тенге.}$$

При расчете фонда заработной платы, нужно учитывать, социальный налог в размере 9,5% от общего фонда оплаты труда после отчисления в пенсионный фонд, отчисления в пенсионный фонд составляют 10 % от ФОТ:

$$С_н = 0.095 \times (ФОТ - ПО). \quad (4.8)$$

где ПО – отчисления в пенсионный фонд; ФОТ – фонд оплаты труда;
Пенсионные отчисления будут равны:

$$ПО = 207506 \times 0{,}1 = 20751 \text{ тенге.}$$

тогда, исходя из формулы (4.8) социальный налог будет равен:

$$С_н = 0{,}095 \times (207506 - 20751) = 17742 \text{ тенге.}$$

4.2.3 Расчет затрат на накладные расходы

К накладным затратам относятся расходы на арендную плату, включая коммунальные платежи и прочие хозяйственные расходы. Прочие затраты представлены в таблице 21.

Таблица 21 – Прочие затраты

№	Наименование материального ресурса	Количество	Цена за единицу, тг	Сумма, тг
1	Интернет-услуги	1 месяц	4600	4600
2	Аренда офиса	15 кв. м	5000	75000
3	Расходы на содержание офиса	15 кв. м	250	3750
4	Система пожарной сигнализации	1 месяц	12000	12000
5	Охрана помещения	1 месяц	13000	13000
ИТОГО затраты на материальные нужды				108350

На основании полученных данных по отдельным статьям составлена смета затрат на разработку ПП таблице 22.

Таблица 22 – Смета затрат на разработку ПП

№	Статьи затрат	Сумма, тг
1	Материальные затраты, в том числе: – материалы – электроэнергия	 11785 1831
2	Затраты на оплату труда	207506
3	Отчисления на социальные нужды	17742
4	Амортизация основных фондов	857,22
5	Прочие затраты	108350
ИТОГО по смете		348071,22

4.3 Оценка социально – экономических результатов функционирования ПП

Применение концепции SDN дает множество преимуществ, среди них:

– возможность видеть топологию всей сети;

– динамическая конфигурация всей сети в целом, а не отдельных единиц оборудования;

– независимое от поставщиков обновление оборудования;

– возможность контроля всей сети из высокоуровневого приложения.

Технология SDN быстро совершенствуется и на данный момент наиболее актуальна для применения в ЦОД в связи с большим объемом проходящего трафика и необходимостью быстрого реагирования на изменения в сети, в том числе атаки на критически важные ресурсы [8].

Различные атаки мошенников могут привести к финансовым потерям при утрате компьютера. Так как компьютер при атаке мошенников может выйти из строя, необходимо внедрять различные механизмы обеспечения информационной безопасности, таких как шифрование.

Согласно исследованиям Ponemon Institute, средняя стоимость утечки информации для фирм в Великобритании составила 1,7 млн фунтов.

Для того чтобы снизить финансовые потери, необходимо воспользоваться шифрованием данных. Эксперты подсчитали, что при шифровании информации, потеря обходится в 37 443 долларов, если не использовать шифрование, - то в 56 165 долларов.

Также финансовые потери напрямую зависят от того, какую должность в компании занимает человек, потерявший информацию. Наибольшей ценностью обладает информация не высшего должностного лица компании, а директора или менеджера. Убыток от потери информации топ-менеджером обходится в среднем в 28 449 долларов, но, если его потеряли директор или менеджер, сумма возрастает до 60 781 долларов и 61 040 долларов соответственно.

ПКС подход имеет много преимуществ в области безопасности, особенно в части физической безопасности сетевого оборудования и конфигурирования политик доступа к оборудованию. Разделение плоскости данных и плоскости управления дает дополнительные преимущества. Она упрощает конфигурирование и управление большим количеством различных сетевых устройств. Позволяет отслеживать статистику для конкретных элементов сети и отправлять инструкции.

5 Безопасность жизнедеятельности

Крайне важно, чтобы организации соблюдали конкретные законы о здоровье и безопасности, чтобы на рабочем месте не пострадал работник. Во-первых, большинство организаций будут рассматривать опасности на рабочем месте; это факторы, которые могут нанести кому-либо вред или нанести ущерб во время процессов, используемых организацией. Тогда большинство организаций будет рассматривать риск каждой опасности, это означает вероятность того, что опасность действительно причиняет вред или травму [29]. Выявление опасностей и рисков позволяет организации выработать конкретные меры предосторожности, которые помогут минимизировать риск путем создания правил или положений на рабочем месте. Схема рабочего места представлена на рисунке 50.

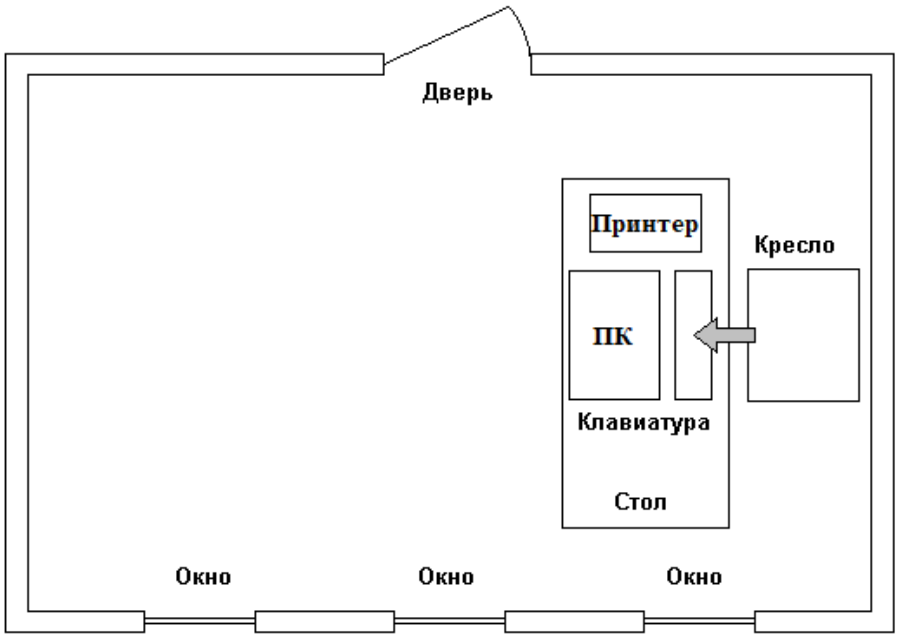

Рисунок 50 – Схема рабочего места

Большинство организаций будет показывать эту оценку здоровья и безопасности на своем рабочем месте как оценку риска, что очень важно, поскольку позволяет работодателям и работникам знать, как быть в безопасности, и какие меры предосторожности им необходимо предпринять, чтобы уменьшить вероятность кого-либо или что-то повреждено [29].

Раздел 4 «Безопасность и охрана труда» в трудовом Кодексе Республики Казахстан был создан для «обеспечения безопасности жизни и здоровья работников в процессе трудовой деятельности» [30]. Это включает в себя выявление опасностей в каждом отделе организации для снижения рисков в процессе или методе, которые сотрудники использовали для производства продукта или развития предоставляемых услуг. Кроме того, Закон запрещает использование чрезвычайно опасных, опасных и легковоспламеняющихся продуктов, если только оценка риска не показывает, каким образом будут

предотвращаться опасности, например, использование защитной одежды, такой как лабораторные халаты, перчатки и защитные очки, а также строгий надзор, который может потребоваться, чтобы процесс, в котором используется вещество, было разрешено продолжать.

Эти вещества могут быть опасны не только для работников, но и для окружающей среды. Закон о безопасности и гигиене труда пытается уменьшить количество вредных выбросов в атмосферу. Кроме того, он может включать устойчивость рабочего места путем анализа состояния здания, чтобы гарантировать, что здание не может рухнуть, когда сотрудники на работе, а также он гарантирует, что на рабочем месте есть особые функции безопасности, такие как противопожарные двери, так что риск количество работников, сжигаемых при пожаре, уменьшается [29].

В офисе, работодатель несет ответственность за соблюдение положений Закона об охране здоровья и технике безопасности на рабочем месте, а также за поддержание использования оценки риска на рабочем месте, главным образом путем разработки специальной письменной политики безопасности, которая может быть легко понятна сотрудникам, что показывает подходящие меры предосторожности при оценке риска.

5.1 Пожаробезопасность

Из-за опасных и легковоспламеняющихся материалов, которые часто используются или хранятся в офисах, и их близости к работникам, пожары могут иметь катастрофические последствия, но промышленные объекты не являются единственными рабочими местами, уязвимыми к опасностям пожара [31]. Несмотря на то, что офисные работники часто мало думают о возможности пожара при выполнении своей работы, каждый год в офисных помещениях происходит более 1000 пожаров.

Пределы огнестойкости здания представлены в таблице 23.

Таблица 23 – Пределы огнестойкости здания

Степень огнестойкости здания Df	Пределы огнестойкости конструкций			
	Несущие элементы	Наружные ненесущие стены	Перекрытия	Стены лестничных клеток
I	R 120	E 30	REI 60	REI 120

Такие факторы, как беспечность сотрудников, офисное оборудование и приборы, которые выделяют тепло, а также большое количество бумаги и других горючих материалов, могут способствовать пожарам в офисе, что может привести к серьезному материальному ущербу, серьезным травмам и гибели людей.

Соответствие запроектированной конструкций здания требованиям СП 2.13130.2009, указано в таблице 24.

Для того чтобы обезопасить сотрудников от возможности появления пожара, в офисе соблюден следующий противопожарный режим [31]:

а) наличие табличек с номером телефона вызова пожарной охраны и фамилиями ответственных за противопожарное состояние помещений.

б) наличие планов эвакуации при пожаре. План разработан для каждого этажа офисного здания. Копии всех поэтажных планов используются для создания сводного плана, хранящегося у ответственного лица. Также изготавливается локальный план эвакуации. В соответствии с требованиями план эвакуации является фотолюминесцентным и размещается в хорошо просматриваемых местах. Расстояние от одного плана до другого на одном этаже не превышает 60 метров.

в) наличие знаков пожарной безопасности в офисе. К ним относятся информационные стенды и плакаты по пожарной безопасности и информационные таблички нескольких видов:

– запрещающие таблички (о запрете курения);

– предупреждающие, указывают на возможную опасность;

– предписывающие, представляют собой команду на совершение определенных действий, например, «Работать здесь!»;

– указательные знаки для средств противопожарной защиты красного цвета с изображением местонахождения пожарного гидранта, телефона, водоема;

– указательные для целей эвакуации — зеленые таблички с указанием направления движения, эвакуационного или аварийного выхода.

Таблица 24 – Соответствие конструкции здания по СП

Наименование конструкций здания	Требуется по СП	Соответствие требованиям норм
Стены наружные	E 30	соответствует
Стены лестничных клеток	REI 120	соответствует
Плиты перекрытий	REI 60	соответствует
Колонны	R 120	соответствует
Балки	R 120	соответствует

г) наличие первичных средств пожаротушения в офисе. Это средства, которые эффективны на начальной стадии возгорания. Они подразделяются на огнетушащие вещества (вода, песок), огнетушащие материалы (асбестовое полотно, металлическая сетка с мелкими ячейками, кошма) и пожарный ручной инструмент и инвентарь (лопаты, ломы, крюки, пожарные краны и огнетушители). Все оборудование находится в доступных местах. Из числа работников офиса назначены лица, несущие ответственность за наличие и сохранность первичных средств пожаротушения. Они проводят их ежеквартальный осмотр и при необходимости производят замену или устранение неполадок.

д) наличие журнала инструктажа. Журнал инструктажа по пожарной безопасности создается в соответствии с требованиями Правил противопожарного режима РК. В журнал заносятся данные о проведении инструктажа, содержащие сведения о личных данных инструктируемого, его профессии, даты прохождения инструктажа и личные подписи инструктирующего и инструктируемого.

е) доступ к эвакуационным выходам и путям эвакуации. Одной из действенных мер пожарной безопасности в офисе является обеспечение свободного доступа к эвакуационным путям и выходам. Для указания их расположения можно использовать различные средства оповещения, как речевые, так и световые. Так как наш офис является небольшим помещением, то главное требование к аварийному выходу — это обеспечение свободного выхода на улицу. Офис находится в арендованном помещении в бизнес-центре, значит за оборудование аварийных выходов и путей несет ответственность арендодатель.

5.2 Электробезопасность

Почти каждый процесс в любом офисе сегодня основан на оборудовании, которое работает от электричества и является потенциально опасным, если неправильно используется или не обслуживается регулярно. Это может быть вызвано неисправной электропроводкой, небезопасными установками, потертыми шнурами и нестандартными отключениями питания.

Пороговые величины тока представлены в таблице 25.

Во избежание каких-либо исходов, которые могут нести риск здоровью работников, в офисе будет соблюден следующий контрольный список [29]:

– каждый отдельный элемент оборудования, машина и устройство имеют двойную изоляцию и надлежащим образом заземлены;

– розетки не перегружены;

– отсутствие подключений штекеров с несколькими выходами к другим штекерам с несколькими выходами;

– запрет на использование оборудования мокрыми руками;

– проверены все удлинители, чтобы убедиться, что они не перегружены, и помещены в хорошо проветриваемые помещения для адекватного рассеивания тепла;

– отсутствие подключения заземленных шнуров к незаземленным розеткам;

– не связывайте и не завязывайте электрические шнуры и не прячьте их под коврами, где стулья могут их пережать;

– отключены все устройства и гаджеты, когда они не используются, чтобы сэкономить больше энергии и устранить риск возникновения пожара и электрических опасностей;

– используются правильные соединители, разъемы, а также электрические провода и кабели, чтобы объединить вместо того, чтобы прикреплять какие-либо соединения;

– проверена и обслужена каждая установка компетентным электриком.

Как правило, первые ощущения возникают при воздействии тока 1-1,5 мА. Эта величина считается пороговой. Дальнейшее повышение приводит к непроизвольным сокращениям мышечной системы, сопровождающимися болезненными ощущениями.

После рубежа от 12 до 15 мА мышечная система не поддается контролю. В некоторых случаях из-за этого попавший под напряжение не имеет возможности самостоятельно освободиться (например, разжать кулак с зажатым проводом). Ток, начиная с указанного рубежа, считается «не отпускающим». Дальнейшее его повышение вызывает судорожные сокращения сердца, а величина – 100 мА приводит к летальному исходу.

Электрические установки, к которым относится практически все оборудование ЭВМ, представляют для человека большую потенциальную опасность, так как в процессе эксплуатации или проведении

профилактических работ человек может коснуться частей, находящихся под напряжением.

Таблица 25 – Пороговые величины тока

Напряжение	Ощутимый (мА)	Не отпускающий (мА)	Фибрилляционный с летальным исходом
Переменное (50Гц)	1,0-1,5	12,0-15,0	100,0
Постоянное	6,0	60,0	300,0

Любое воздействие тока может привести к электрической травме, то есть к повреждению организма, вызванному действием электрического тока или электрической дуги.

При рассмотрении вопроса обеспечения электробезопасности разработчика необходимо выделить три основных фактора:

– электроустановки рабочего места программиста;

– вспомогательное электрооборудование;

– окружающая среда помещения.

К электроустройствам рабочего места относятся: компьютер, видеомонитор, принтер. К вспомогательному оборудованию относятся лампы местного освещения, вентиляторы и другие электрические приборы. Электрооборудование, перечисленное выше, относится к установкам напряжением до 1000 В, исключение составляют лишь дисплей, электронно-лучевые трубки, которых имеют напряжение в несколько киловольт.

Окружающая среда помещений, в которых работает программист, воздействует на электрическую изоляцию приборов и устройств, электрическое сопротивление тела человека и может создавать условия для поражения электрическим током.

Помещения, оборудованные вычислительной техникой, как правило, относятся к категории помещений без повышенной опасности так как:

– относительная влажность воздуха не превышает 75%;

– нет токопроводящей пыли;

– температура не превышает длительное время 30 °С;

– отсутствует возможность одновременного прикосновения человека с имеющими соединение с землей металлическими конструкциями;

– отсутствие доступа к токоведущим частям оборудования;

– нет токопроводящих полов.

Таким образом, для предотвращения электротравматизма пользователя, необходимо соблюдать требования безопасности.

5.3 Расчетная часть

5.3.1 Расчет уровня шума

Одним из неблагоприятных факторов производственной среды в ИВЦ является высокий уровень шума, создаваемый печатными устройствами, оборудованием для кондиционирования воздуха, вентиляторами систем охлаждения в самих ЭВМ.

Для решения вопросов о необходимости и целесообразности снижения шума необходимо знать уровни шума на рабочем месте оператора.

Уровень шума, возникающий от нескольких некогерентных источников, работающих одновременно, подсчитывается на основании принципа энергетического суммирования излучений отдельных источников [32]:

$$L_{\Sigma} = 10lg \sum_{i+1}^{n} 10^{0,1L_i}, \quad (5.1)$$

где L_i – уровень звукового давления i – того источника шума;

n – количество источников шума.

Полученные результаты расчета сравнивается с допустимым значением уровня шума для данного рабочего места. Если результаты расчета выше допустимого значения уровня шума, то необходимы специальные меры по снижению шума. К ним относятся: облицовка стен и потолка зала звукопоглощающими материалами, снижение шума в источнике, правильная планировка оборудования и рациональная организация рабочего места оператора. Уровни звукового давления источников шума, действующих на оператора на его рабочем месте представлены в таблице 26.

Таблица 26 – Уровни звукового давления различных источников

Источник шума	Уровень шума, дБ
Монитор	17
Жесткий диск	40
Вентилятор (Кулер)	45
Клавиатура	10
Принтер	45

Обычно рабочее место оператора оснащено следующим оборудованием: винчестер в системном блоке, вентилятор(ы) систем охлаждения ПК, монитор, клавиатура, принтер и сканер.

Подставив значения уровня звукового давления для каждого вида оборудования в формулу, получим:

$$L_{\Sigma} = 10lg \sum_{i+1}^{n} 10^{0,1L_i} = 10\,lg(10^{1,7} + 10^{4} + 10^{4,5} + 10^{1} + 10^{4,5}) = 48{,}6 \text{ дБ}$$

Согласно санитарно-эпидемиологическим требованиям к эксплуатации персональных компьютеров, видеотерминалов и условиям работы с ними, полученное значение не превышает допустимый уровень шума для рабочего места оператора, равный 65 дБ [33].

5.3.2 Расчет освещенности

Для расчета освещенности рабочего места, необходимо выбрать систему освещения, определить необходимое количество светильников, типа выбранных светильников и их размещение. Таким образом мы рассчитаем параметры искусственного освещения.

Обычно искусственное освещение выполняется посредством электрических источников света двух видов: ламп накаливания и люминесцентных ламп. Будем использовать люминесцентные лампы, которые по сравнению с лампами накаливания имеют ряд существенных преимуществ:

Чаще всего искусственное освещение осуществляется за счет двух видов электрических источников света:

– по спектральному составу света они близки к дневному, естественному свету;

– обладают более высоким КПД (в 1,5-2 раза выше, чем КПД ламп накаливания);

– обладают повышенной светоотдачей (в 3-4 раза выше, чем у ламп накаливания);

– более длительный срок службы.

Расчет освещения производится для комнаты площадью 15 м^2, ширина которой 5 м, высота - 3 м. Схема рабочего места представлена на рисунке 50. Воспользуемся методом светового потока.

Для определения количества светильников определим световой поток, падающий на поверхность по формуле:

$$F = \frac{E \times K \times S \times Z}{n}, \qquad (5.2)$$

где F – рассчитываемый световой поток, лм;

E – нормированная минимальная освещенность, лк (определяется по таблице). Работу программиста, в соответствии с этой таблицей, можно отнести к разряду точных работ, следовательно, минимальная освещенность будет E = 300 лк;

S – площадь освещаемого помещения (в нашем случае S = 15 м^2);

Z – отношение средней освещенности к минимальной (обычно принимается равным 1,1…1,2, пусть Z = 1,1);

K – коэффициент запаса, учитывающий уменьшение светового потока лампы в результате загрязнения светильников в процессе эксплуатации (его значение зависит от типа помещения и характера проводимых в нем работ и в нашем случае K = 1,5);

n – коэффициент использования, (выражается отношением светового потока, падающего на расчетную поверхность, к суммарному потоку всех ламп и исчисляется в долях единицы; зависит от характеристик светильника, размеров помещения, окраски стен и потолка, характеризуемых коэффициентами отражения от стен (РС) и потолка (РП)), значение

коэффициентов РС и РП были указаны выше: РС=40%, РП=60%. Значение n определим по таблице коэффициентов использования различных светильников. Для этого вычислим индекс помещения по формуле:

$$I = \frac{S}{h(A+B)}, \tag{5.3}$$

где S – площадь помещения, S = 15 м^2;

h – расчетная высота подвеса, h = 2,92 м;

A – ширина помещения, A = 3 м;

B – длина помещения, B = 5 м.

Подставив значения получим:

$$I = \frac{15}{2{,}92 * (3+5)} = 0{,}64.$$

Зная индекс помещения I=0,64, коэффициент использования n = 0,22.

Подставим все значения в формулу для определения светового потока F:

$$F = \frac{300 * 1{,}5 * 15 * 1{,}1}{0{,}22} = 33750 \text{ лм.}$$

Для освещения выбираем люминесцентные лампы типа ЛБ40-1, световой поток которых $F = 4320$ лк.

Рассчитаем необходимое количество ламп по формуле:

$$N = \frac{F}{F_Л}, \tag{5.4}$$

где N – определяемое число ламп;

F – световой поток, F = 33750 лм;

$F_Л$ – световой поток лампы, $F_Л$ = 4320 лм.

$$N = \frac{33750}{4320} = 8 \text{ шт.}$$

При выборе осветительных приборов используем светильники типа ОД. Каждый светильник комплектуется двумя лампами.

Согласно требованиям освещенность при комбинированном освещении должна быть $E^{Н}_{КОМБ} = 1250$ лк, поэтому [34]:

$$E^{Н}_{МЕСТН} = E^{Н}_{КОМБ} - E^{Н}_{ОБЩ} = 1250 - 300 = 950 \text{ лк,} \tag{5.5}$$

Для расчета местного освещения воспользуемся точечным методом.

Для определения светового потока Fл от лампы местного освещения, создающей на рабочей поверхности освещенность $E_{местн}$, будем использовать формулу:

$$F_Л = \frac{1000 \times E^Н_{МЕСТН} \times К}{\mu \times \varepsilon}, \quad (5.6)$$

где К = 1,3 – коэффициент запаса;

μ = 1,1 – коэффициент, учитывающий влияние отраженного света и удаленных светильников (например, светильников местного освещения соседних рабочих мест) [29];

$E^Н_{МЕСТН} = 950$ лк– нормированная местная освещенность;

ε – условная освещенность.

Условная освещенность, создаваемая условной лампой со световым потоком $F_Л = 1000$ лм, зависит от светораспределения светильника и определяется по графикам пространственных изолюкс. По этим графикам для каждого типа светильника по координатам h (высота подвеса светильника над уровнем рабочей поверхности) и d (расстояние от следа светильника на уровень рабочей поверхности до расчетной точки) находится расположение рабочей точки и определяется ее условная освещенность путем интерполирования между значениями, указанными у кривых пространственных изолюкс [34].

При h=0,4 м и d=0,3 м для светильника типа «Альфа» получим ε = 400. Тогда, подставляя все значения в формулу получаем, что необходимо выбрать лампу для местного освещения с таким световым потоком:

$$F_Л = \frac{1000 \times E^Н_{МЕСТН} \times К}{\mu \times \varepsilon} = \frac{1000 \times 950 \times 1{,}3}{1{,}1 \times 400} = 2807 \text{ лм.}$$

Выбираем светильник МО24 - 100 с мощностью 100 Вт.

Максимально допустимое значение коэффициента пульсации $К_П$ светового потока при работе с дисплеями и видеотерминалами не должно превышать 10% [34]. Такой коэффициент пульсации можно обеспечить при включении половины ламп в светильнике по схеме опережающего и половины - по схеме отстающего тока, и чтобы число ламп в светильнике было кратно двум.

В соответствии с СП РК 2.04-104-2012 показатель дискомфорта (М) не должен превышать 40% [34].

Применяемый светильник относится к группе Ie. Зная группу, к которой относится выбранный светильник, при РС=50%, РП=10%, определяем индекс помещения $I_м$, при котором обеспечивается нормируемый максимально допустимый показатель дискомфорта (М). $I_м = 2{,}5$.

Сравнивая $I_м$ с I, определяем соответствие требованиям по дискомфорту осветительной установки. Требования выполняются при соблюдении условия: $I < I_м$.

В нашем случае $I_м$ = 0,64, т.е. условие выполняется, и выбранная осветительная установка соответствует требованиям по дискомфорту.

Заключение

На данном этапе развития сетевых технологий возникает всё больше предпосылок к замене традиционных архитектур, на более гибкие и эффективные. В связи с этим интерес к программно-конфигурируемым сетям (ПКС) возрастает с каждым днём. Появление технологии SDN позволило перейти с экстенсивного (увеличение объёмов и усложнение сетевых топологий) к интенсивному (возможность внедрения новых технологий на основе уже существующей топологии и оборудования) пути развития сетевой инфраструктуры.

Применение концепции SDN дает множество преимуществ, среди них:

– возможность видеть топологию всей сети;

– динамическая конфигурация всей сети в целом, а не отдельных единиц оборудования;

– независимое от поставщиков обновление оборудования;

– возможность контроля всей сети из высокоуровневого приложения.

Технология SDN быстро совершенствуется и на данный момент наиболее актуальна для применения в ЦОД в связи с большим объемом проходящего трафика и необходимостью быстрого реагирования на изменения в сети, в том числе атаки на критически важные ресурсы. ПКС подход имеет много преимуществ в области безопасности, особенно в части физической безопасности сетевого оборудования и конфигурирования политик доступа к оборудованию. Разделение плоскости данных и плоскости управления дает дополнительные преимущества. Она упрощает конфигурирование и управление большим количеством различных сетевых устройств. Позволяет отслеживать статистику для конкретных элементов сети и отправлять инструкции.

Однако, в области концепции SDN, необходимы дальнейшие исследования. Примером такой области исследований может являться безопасность API интерфейсов контроллера для связи с приложениями и облачными сервисами.

Целью данной выпускной квалификационной работы являлось исследование архитектуры SDN, с точки зрения ИБ и механизмов защиты, применяемых для обеспечения безопасности и надежности в сетях SDN. В ходе исследования были решены следующие задачи:

– осуществлен сравнительный анализ безопасности традиционной сетевой архитектуры и архитектуры sdn;

– исследованы проблемы, связанные с обеспечением безопасности контроллера sdn;

– разработан и реализован сегмент компьютерной сети согласно концепции sdn. Внедрен механизм установления безопасного соединения контроллера и коммутатора. Полученные результаты могут быть использованы на практике при планировании или в целях имитационного моделирования и изучения концепции sdn сетей.

Список литературы

1 Сергей Орлов. SDN и другие. Журнал сетевых решений. https://habr.com/post/315524/, (03.01.2019)

3 Сергей Орлов. Журнал сетевых решений. Оптимизация сетевого трафика. https://www.osp.ru/lan/2014/11/13043730, (15.01.2019)

4 Шалимов А.В. Технологии SDN/OpenFlow http://lvk.cs.msu.su/~sveta/SDN_OpenFlow_basics_lecture1.pdfSDN&NFV, (03.02.2019)

5 Open Networking Foundation. SDN architecture Issue 1 June, 2014 ONF TR502. https://www.opennetworking.org/images/stories/downloads/sdnresources/technical-reports/TR_SDN_ARCH_1.0_06062014.pdf, (03.02.2019)

6 Open Networking Foundation. SDN Architecture Overview Version 1.0 December 12, 2013. https://www.opennetworking.org/images/stories/downloads/sdnresources/technical-reports/SDN-architecture-overview-1.0.pdf, (07.02.2019)

7 Open Networking Foundation. OpenFlow Switch Specification Version 1.5.1 (Protocol version 0x06) March 26, 2015 ONF TS-025. https://www.opennetworking.org/wp-content/uploads/2014/10/openflow-switchv1.5.1.pdf, (07.02.2019)

8 Смелянский Р.Л Технологии SDN и NFV: новые возможности для телекоммуникаций. https://www.osp.ru/netcat_files/userfiles/Ethernet_2015/Smelyansky_R.pdf, (09.02.2019)

9 “OpenFlow switch specification: Version 1.4.0”. https://www.opennetworking.org/images/stories/downloads/specification/openflow-spec-v1.4.0.pdf, (13.02.2019)

10 Software-Defined Networking: The New Norm for Networks ONF White Paper April 13, 2012. https://www.opennetworking.org/images/stories/downloads/sdnresources/white-papers/wp-sdn-newnorm.pdf, (13.02.2019)

11 Курочкин И.И., Гуменный Д.Г. Безопасность сетей SDN.Классификация атак // Современные информационные технологии и ИТ-образование. 2015. №11. https://cyberleninka.ru/article/n/bezopasnost-seteysdn-klassifikatsiya-atak, (15.02.2019)

12 Дмитрий Ганьжа Инфраструктура ЦОД. https://www.osp.ru/lan/2018/04/13054102 18.04.2018, (21.02.2019)

13 Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы. 4-е изд. Санкт-Петербург: Питер, 2010. 944 с.

14 Борисов М.А., Заводцев И.В., Чижов И.В. Основы программно-аппаратной защиты информации. 3-е изд. Москва: ЛЕНАНД, 2014. 416

15 Галицкий А.В., Рябко С.Д., Шаньгин В.Ф. Защита информации в сети.
Анализ технологий и синтез решений. Москва: ДМК пресс, 2004. 616 с.

16 Catherine P. Implementing Cisco IOS Network Security (IINS). Indianapolis: CiscoPress, 2009.

17 OpenFlow: A Security Analysis Rowan Kloti ¨ ETH Zurich Zurich, Vasileios Kotronis ETH Zurich Zurich, Paul Smith AIT Austrian Institute of Technology 2444 Seibersdorf

18 Ayesha Imran SDN Controllers Security Issues MS Thesis document in Web Intelligence and Service Engineering November 9, 2017

19 University of Jyväskylä Department of Mathematical Information Technology.

https://jyx.jyu.fi/dspace/bitstream/handle/123456789/55934/URN:NBN:fi:jyu-201711204305.pdf?sequence=1, (25.02.2019)

20 D. Kreutz, F.M. Ramos, P. EstevesVerissimo, C. Esteve Rothenberg, S. Azodolmolky, S. Uhlig Software-defined networking: A comprehensive survey, 103

21 P. Porras et al., S. Shin, V. Yegneswaran, M. Fong, M. Tyson, and G. Gu, “A security enforcement kernel for OpenFlow networks,” in Proc. Of HotSDN, 2012.

22 R. Sherwood and et al., “Flowvisor: A network virtualization layer,” OpenFlow Switch Consortium, Tech. Rep, 2009

23 Dotsenko S., Vladyko A., Letenko I. A fuzzy logic-based information security management for software-defined networks. 16th International conference on advanced communication technology. 2014. Pp. 167–171

24 A. Khurshid et al., “VeriFlow: verifying network-wide invariants in real time,” in Proc. of HotSDN, 2012.
http://www.tik.ee.ethz.ch/file/9ee69e89be779fd1448a7356d79ddb18/openflow_sec.pdf, (05.03.2019)

25 Документация VirtualBox.
https://www.virtualbox.org/manual/, (05.03.2019)

26 Cisco security data center.
https://www.cisco.com/c/en/us/solutions/security/secure-data-centersolution/index.html, (05.03.2019)

27 Документация Opendaylight.
https://www.opendaylight.org/, (05.03.2019)

28 Бекишева А.И. Методические указания к выполнению экономической части дипломной работы – Алматы: АУЭС; 2013. – 24с.

29 https://ohsonline.com/Blogs/The-OHS-Wire/2017/06/, (03.02.2019)

30 Трудовой кодекс республики Казахстан (с изменениями и дополнениями по состоянию на 01.01.2019 г.)

31 https://officeblindsandglazing.co.uk/fire-safety-workplace/, (03.02.2019)

32 СанПиН РК № 3.01.030-97 «Предельно допустимые уровни инфразвука и низкочастотного шума в помещениях жилых и общественных зданий и на территории жилой застройки»

33 https://egov.kz/wps/portal/!ut/p/b0/, (16.03.2019)
34 СП РК 2.04-104-2012 «Естественное и искусственное освещение» (с изменениями и дополнениями по состоянию на 01.08.2018 г.)

Приложение А

Текст программы для подключения необходимых библиотек

Sudo apt-get install default-jre-headless

Sudo apt-get install maven

Sudo apt-get install openjdk-8-jdk

Sudo apt-get install openjdk-8-jre

Установка контроллера OpenDayLight:

- скачиваем tar файл с контролером с сайта opendaylight
cd Загрузки/;
- распаковываем: tar –xvzf karaf-0.8.1.tar.gz
Cd bin/. /karaf.
Для запуска контроллера из CLI: cd Загрузки/ karaf-0.8.1/bin. /karaf

Приложение Б

Текст программы для установки ovs

sudo apt-get update

sudo apt-get upgrade

sudo apt-get install qemu-kvm libvirt-bin ubuntu-vm-builder bridge-utils cpuchecker vnc4server virtinst

Install by Open vSwitch by running:

sudo apt-get install openvswitch-switch openvswitch-common

Приложение В

Текст программы создания интерфейса моста

ovs-vsctl add-br br0

ovs-vsctl add-port br0 ens160

Ifconfig ens160 0

Ifconfig br0 up

Ifconfig br0 192.168.31.131 netmask 255.255.254.0

Route add default gw 192.168.31. br0

ovs-vsctl set-controller br0 ssl: ip:6633

ovs-ofctl add-flow br0 action-normal

ovs-vsctl set bridge br0 protocols=OpenFlow10, OpenFlow11, OpenFlow12, OpenFlow13

Таблица В.1 – Реализованные команды ovs-pli и их аргументы

Компонент	Назначение
Init	Инициализирует новую PKI и заполняет ее парой центров сертификации для контроллеров и коммутаторов.
req name	Создает новый закрытый ключ с именем name - privkey.pem и соответствующий запрос сертификата с именем name req.pem.
sign name [type]	Вызывает запрос сертификата с именем name-req.pem для создания сертификата с именем name-cert.pem.
req + sign name [type]	Объединяет команды req и sign в один шаг,поместив все файлы, созданные каждым. Имя-privkey.pem и файлы name-cert.pem должны быть надежно скопированы на коммутатор или контроллер.
self – sign name	Признает запрос сертификата с именем name-req.pem, используя pri-vate key nameprivkey.pem, создающий самозаверяющий сертификат named name-cert.pem. Необходимо было создать входные файлы с ovs-pki req.