Министерство просвещения Республики Башкортостан

Государственное автономное профессиональное образовательное учреждение

Уфимский колледж статистики, информатики и вычислительной техники

|  |  |
| --- | --- |
|  | УТВЕРЖДАЮ  Заведующий отделением  вычислительной техники  \_\_\_\_\_\_\_\_\_\_Т.Р. Тимашев  « » 2025 г. |

«СОЗДАНИЕ СИСТЕМЫ АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ СЕТЬЮ SDN»

Пояснительная записка к курсовому проекту

|  |  |
| --- | --- |
|  | Руководитель проекта  Р.А. Озерова  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2025 г. |
|  | Студент группы 22СА-1  Идрисов И.Р.  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2025 г. |

2025

Министерство образования и науки Республики Башкортостан

Государственное автономное профессиональное образовательное учреждение

Уфимский колледж статистики, информатики и вычислительной техники

|  |  |
| --- | --- |
|  | УТВЕРЖДАЮ  Заведующий отделением  вычислительной техники  Т.Р. Тимашев  « » 2025 г. |

ЗАДАНИЕ

На курсовой проект студенту дневного отделения, группы 22СА-1, специальности 09.02.06 Сетевое и системное администрирование.

Фамилия, имя, отчество: Идрисов Ильсур Рашитович

Тема курсового проекта: «Создание системы автоматического управления сетью SDN»

В результате выполнения курсового проекта должны быть представлены:

1. пояснительная записка, состоящая из следующих разделов:

Введение

1. Общая часть

2. Специальная часть

3. Экспериментальная часть

Заключение

Список использованной литературы

б) презентация курсового проекта

Список рекомендуемой литературы

Немет, Эви, Снайдер, Гарт, Хейн, Трент, Уэйли, Бен, Макни, Дэн.

Unix и Linux: руководство системного администратора, 5-е изд.: Пер.

С англ. -СБП.: ООО «Диалектика», 2020. – 1168 с.

Милл, И., Хобсон, Э.С., Docker на практике / пер. с англ. Д.А.

Беликов. – М. ДМК Пресс, 2020. 516 с.

Шрёдер Карла Linux. Книга рецептов. 2-е изд. – СПБ.: Питер,

2020. – 592 с.

Задание к выполнению получил «1» сентября 2025 г.

Студент \_\_\_\_\_\_\_\_\_\_\_\_ Идрисов Ильсур Рашитович

Срок окончания «05» ноября 2025 г.

Руководитель курсового проекта Р.А. Озерова

Задание рассмотрено на заседании цикловой комиссии вычислительной техники   
протокол № \_\_от «\_\_» \_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

Председатель цикловой комиссии Р.А. Озерова

Министерство образования и науки Республики Башкортостан

Государственное автономное профессиональное образовательное учреждение

Уфимский колледж статистики, информатики и вычислительной техники

ЗАКЛЮЧЕНИЕ

на курсовой проект

Студент Идрисов Ильсур Рашитович

Группа 22СА-1

Специальность 09.02.06 Сетевое и системное администрирование

Тема: Создание системы автоматического управления сетью SDN

Объем курсового проекта:

количество листов пояснительной записки

количество листов графической части

Заключение о степени соответствия заданию на курсовое проектирование

Характеристика качеств, проявленных студентом при работе над проектом: самостоятельность, дисциплинированность, умение планировать работу и пользоваться литературным материалом и т.д.

Положительные стороны курсового проекта

Недостатки курсового проекта

Характеристика общетехнической и специальной подготовки студента

Заключение и предлагаемая оценка за курсовой проект

Руководитель курсового проекта Роза Алексеевна Озерова

Должность Преподаватель

Место работы ГАПОУ УКСИВТ

«\_\_» 2025 г.

Подпись\_\_\_\_\_\_\_\_\_\_\_\_\_

Изм.

Лист

№ докум.

Подпись

Дата

Лист

x

40.И-1020-25 09.02.06 КП-ПЗ

Разраб.

Идрисов И. Р.

Провер.

Тимашев Т.Р.

Реценз.

Н. Контр.

Озерова Р.А.

Утверд.

Тимашев Т.Р.

«СОЗДАНИЕ СИСТЕМЫ АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ СЕТЬЮ SDN»

Лит.

Листов

x

УКСИВТ 22СА-1

АННОТАЦИЯ

В данной курсовой работе исследуются методы построения системы автоматизированного управления программно-конфигурируемыми сетями (SDN). Работа направлена на разработку архитектуры мониторинга сетевой инфраструктуры, включая оборудование данных и прикладные сервисы. Особое внимание уделяется анализу требований к сбору и обработке метрик производительности, а также автоматизации конфигурации плоскостей данных. В рамках исследования реализованы специализированные шаблоны мониторинга, развернуты агенты сбора метрик и прокси-компоненты, разработана система оповещений и отчетности с учетом критериев доступности и производительности сетевых сервисов.

Содержание

[1.1. ОСНОВНЫЕ ПРИНЦИПЫ SDN 12](#_Toc212542427)

[1.2 Архитектура SDN. 16](#_Toc212542428)

[1.2.1. Ключевые принципы архитектуры SDN 16](#_Toc212542429)

[1.2.2. Уровни архитектуры SDN 17](#_Toc212542430)

[1.2.3. Интерфейсы SDN: Northbound и Southbound API 18](#_Toc212542431)

[1.2.4. Обоснование выбора архитектуры для проекта 19](#_Toc212542432)

[2. Специальная часть. 21](#_Toc212542433)

[2.1. Обоснование выбора технологий и инструментов. 21](#_Toc212542434)

[2.2.1. Уровни архитектуры. 22](#_Toc212542435)

[2.2.2. Логика работы системы автоматического управления. 23](#_Toc212542436)

[2.3. Физическая и логическая схема тестового стенда. 24](#_Toc212542437)

[2.4. Выводы по главе. 24](#_Toc212542438)

[3. Практическая часть. 26](#_Toc212542439)

[3.1. Создание виртуальной сети. 26](#_Toc212542440)

[3.2 Создание конфигурационного файла SDN. 27](#_Toc212542441)

[3.3 Развёртка SDN на сервере. 28](#_Toc212542442)

[3.4 Настройка OpenVSwitch. 28](#_Toc212542443)

[3.5 Проверка firewall 28](#_Toc212542444)

[ЗАКЛЮЧЕНИЕ 29](#_Toc212542445)

[СПИСОК ЛИТЕРАТУРЫ 32](#_Toc212542446)

ВВЕДЕНИЕ

Современные сети передачи данных становятся все более сложными и многоуровневыми, что требует от администраторов и инженеров по сетям применения новых подходов к управлению и оптимизации сетевых ресурсов. В условиях стремительного роста объемов информации и увеличения числа подключенных устройств, традиционные методы управления сетями часто оказываются недостаточно эффективными. В этой связи концепция программно-определяемых сетей (SDN, Software-Defined Networking) приобретает все большее значение. SDN представляет собой революционный подход к архитектуре сетей, который позволяет отделить управление сетью от её аппаратной реализации, обеспечивая более гибкое, масштабируемое и эффективное управление сетевыми ресурсами.

Суть технологии SDN заключается в создании абстрактного уровня управления, который позволяет администраторам сети программно управлять сетевыми устройствами и их поведением. Это достигается за счет внедрения контроллера, который служит центральным элементом управления, взаимодействующим с сетевыми устройствами через открытые интерфейсы, такие как OpenFlow. Такой подход открывает новые горизонты для автоматизации процессов управления сетью, что в свою очередь способствует повышению надежности, безопасности и производительности сетевой инфраструктуры.

Создание системы автоматического управления сетью на основе технологии SDN представляет собой актуальную и значимую задачу, поскольку она позволяет оптимизировать использование сетевых ресурсов, минимизировать время реагирования на изменения в сети и упростить процессы администрирования. В рамках данной работы будет рассмотрен процесс проектирования и реализации системы автоматического управления сетью SDN, а также проведено тестирование полученной системы с целью оценки её эффективности и выявления возможных направлений для дальнейшего совершенствования.

Первый раздел работы будет посвящен технологии SDN, где мы подробно рассмотрим её основные принципы, архитектурные компоненты и преимущества по сравнению с традиционными подходами к управлению сетями. Важным аспектом этого раздела станет анализ существующих решений и инструментов, используемых в рамках SDN, что позволит лучше понять, как именно эта технология может быть применена для автоматизации управления сетевой инфраструктурой.

Во втором разделе будет описан процесс проектирования системы автоматического управления сетью. Здесь мы рассмотрим ключевые этапы проектирования, включая определение требований к системе, выбор архитектуры, а также разработку алгоритмов и протоколов, необходимых для эффективного управления сетевыми ресурсами. Важным аспектом проектирования станет интеграция различных компонентов системы, что позволит создать единое целое, способное адаптироваться к изменениям в сетевой среде и обеспечивать высокую производительность.

Третий раздел будет посвящен реализации и тестированию разработанной системы. Мы подробно рассмотрим процесс реализации, включая выбор технологий и инструментов, а также практические аспекты разработки. Важной частью этого раздела станет тестирование системы, в ходе которого будут оцениваться её функциональные возможности, производительность и стабильность работы. Мы также рассмотрим сценарии использования системы в реальных условиях, что позволит оценить её практическую применимость и выявить возможные недостатки.

Таким образом, данная работа направлена на исследование и разработку системы автоматического управления сетью на основе технологии SDN. Мы надеемся, что результаты нашего исследования и практические наработки будут полезны как для дальнейшего изучения технологии SDN, так и для практического применения в области сетевого администрирования и управления сетями. В условиях постоянного роста и усложнения сетевой инфраструктуры, создание эффективных инструментов для автоматизации управления сетью становится не только актуальной задачей, но и важным шагом к созданию более надежных и производительных сетевых решений.

Исследование «Создание системы автоматического управления сетью SDN» является крайне актуальным в условиях стремительного развития информационных технологий и возрастающих требований к гибкости и эффективности сетевой инфраструктуры. Технология SDN (Software-Defined Networking) предлагает новые подходы к управлению сетями, позволяя централизованно контролировать и оптимизировать их работу. В рамках работы, охватывающей проектирование системы автоматического управления и её реализацию с последующим тестированием, рассматриваются ключевые аспекты, такие как адаптивность, масштабируемость и безопасность сетевых решений. Это исследование не только способствует углублению теоретических знаний в области SDN, но и предлагает практические решения для повышения производительности и надежности современных сетевых систем, что делает его значимым для специалистов в области информационных технологий и телекоммуникаций.

В работе «Создание системы автоматического управления сетью SDN» объектом исследования является архитектура и функциональные возможности сетей с программно-определяемой сетью (SDN), которые предоставляют новые подходы к управлению сетевыми ресурсами. Предметом исследования выступает разработка и внедрение системы автоматического управления, способной оптимизировать процессы конфигурирования, мониторинга и управления сетевой инфраструктурой на основе технологии SDN. В рамках исследования рассматриваются ключевые аспекты проектирования системы, включая выбор алгоритмов и протоколов, а также этапы реализации и тестирования, что позволяет оценить эффективность предложенного решения в реальных условиях эксплуатации.

Целью исследования в работе «Создание системы автоматического управления сетью SDN» является разработка эффективной системы, способной оптимизировать управление сетевой инфраструктурой на основе технологии программно-определяемых сетей (SDN). В рамках данной работы поставлены задачи, включающие анализ и обоснование применения технологии SDN, проектирование архитектуры системы автоматического управления, а также реализацию и тестирование разработанной системы для обеспечения ее функциональности и надежности. Эти этапы направлены на достижение высокой степени автоматизации сетевых процессов, что позволит значительно улучшить производительность и управляемость сетевой среды.

1. Общая часть

1.1. ОСНОВНЫЕ ПРИНЦИПЫ SDN

Технология программно-определяемых сетей (SDN, Software-Defined Networking) представляет собой революционный подход к проектированию и управлению компьютерными сетями, который кардинально отличается от традиционных методов. В основе SDN лежит концепция разделения плоскостей управления и передачи данных, что позволяет значительно упростить управление сетевой инфраструктурой, повысить ее гибкость и адаптивность к изменениям в требованиях бизнеса и пользователей. В отличие от традиционных сетей, где маршрутизаторы и коммутаторы принимают решения о маршрутизации и передаче данных на основе встроенных алгоритмов и протоколов, SDN использует централизованный контроллер, который принимает решения о том, как трафик должен обрабатываться и направляться по сети.

Одним из основных принципов SDN является абстракция сети. В традиционных сетях устройства управления и передачи данных жестко связаны друг с другом, что затрудняет их настройку и управление. В SDN контроллер предоставляет единую точку управления, которая абстрагирует детали реализации сети и позволяет администраторам взаимодействовать с сетью через высокоуровневые интерфейсы и API. Это позволяет значительно упростить процессы настройки и управления, а также ускорить развертывание новых услуг и приложений. Например, с помощью программных интерфейсов администраторы могут легко изменять правила маршрутизации, управлять полосой пропускания и настраивать политики безопасности без необходимости вносить изменения в каждое отдельное устройство.

Другим важным аспектом SDN является возможность динамического управления сетью в реальном времени. Традиционные сети часто требуют значительных временных затрат на внесение изменений в конфигурацию и обновление программного обеспечения устройств. В SDN изменения могут быть внесены мгновенно через контроллер, что позволяет быстро реагировать на изменения в сетевом трафике, атаки или сбои в работе оборудования. Это также открывает новые возможности для автоматизации процессов управления сетью, таких как автоматическое масштабирование ресурсов, балансировка нагрузки и управление качеством обслуживания (QoS).

Одним из ключевых компонентов архитектуры SDN является контроллер, который отвечает за принятие решений о маршрутизации и управлении трафиком. Контроллер получает информацию о состоянии сети и ее устройствах, анализирует данные о трафике и применяет заданные политики для оптимизации работы сети. В зависимости от архитектуры SDN, контроллер может быть реализован в виде программного обеспечения, работающего на стандартном сервере, или в виде облачного сервиса. Современные контроллеры поддерживают различные протоколы взаимодействия с сетевыми устройствами, такие как OpenFlow, который позволяет контроллеру управлять потоками данных на уровне пакетов.

Протокол OpenFlow является одним из самых известных и широко используемых протоколов в экосистеме SDN. Он позволяет контроллеру устанавливать правила обработки трафика на коммутаторах и маршрутизаторах, а также получать информацию о состоянии потоков данных. OpenFlow обеспечивает взаимодействие между контроллером и сетевыми устройствами, позволяя динамически изменять маршруты передачи данных и управлять трафиком в реальном времени. Это делает возможным создание сложных сценариев управления трафиком, таких как выделение полосы пропускания для определенных приложений или пользователей, а также реализацию политик безопасности, направленных на предотвращение несанкционированного доступа и атак.

Важным аспектом SDN является поддержка виртуализации сетевых ресурсов. В традиционных сетях физические устройства жестко связаны с конкретными функциями и задачами, что ограничивает их гибкость и возможность масштабирования. В SDN сетевые функции могут быть виртуализированы и реализованы в виде программных приложений, работающих на стандартных серверах. Это позволяет администраторам создавать виртуальные сети, которые могут динамически изменяться в зависимости от потребностей бизнеса. Например, можно создать несколько виртуальных сетей, каждая из которых будет иметь свои собственные политики безопасности и управления трафиком, при этом все они будут использовать одну и ту же физическую инфраструктуру.

Программно-определяемые сети также открывают новые возможности для интеграции с облачными сервисами и приложениями. В условиях стремительного роста облачных технологий и увеличения объема данных, передаваемых по сети, SDN позволяет более эффективно управлять ресурсами и оптимизировать сетевую инфраструктуру. С помощью SDN можно легко интегрировать облачные сервисы, такие как хранение данных, вычислительные ресурсы и приложения, что позволяет создавать гибкие и масштабируемые решения для бизнеса. Например, компании могут быстро развертывать новые облачные приложения, автоматически настраивая сетевые ресурсы в зависимости от изменяющихся требований пользователей.

Кроме того, SDN предоставляет возможности для улучшения управления безопасностью в сетях. Традиционные методы защиты, основанные на статических правилах и конфигурациях, часто не могут эффективно справляться с современными угрозами и атаками. SDN позволяет динамически изменять политики безопасности на основе анализа трафика и поведения пользователей. Например, при обнаружении подозрительной активности контроллер может автоматически изменять правила доступа, блокировать определенные потоки данных или перенаправлять трафик для дополнительного анализа. Это позволяет значительно повысить уровень безопасности сети и снизить риски, связанные с кибератаками.

Однако, несмотря на множество преимуществ, внедрение SDN также сопряжено с определенными вызовами и рисками. Одним из основных вопросов является безопасность самого контроллера и его взаимодействия с сетевыми устройствами. Поскольку контроллер является центральной точкой управления, его компрометация может привести к серьезным последствиям для всей сети. Поэтому необходимо разрабатывать надежные механизмы аутентификации и авторизации, а также обеспечивать защиту данных, передаваемых между контроллером и устройствами. Кроме того, важно учитывать вопросы совместимости и интеграции с существующими сетевыми инфраструктурами, так как многие организации могут использовать устаревшее оборудование и программное обеспечение, которое не поддерживает современные протоколы SDN.

В заключение, технология программно-определяемых сетей (SDN) представляет собой значительный шаг вперед в области управления сетями, предлагая новые подходы к проектированию, развертыванию и управлению сетевой инфраструктурой. Основные принципы SDN, такие как абстракция сети, динамическое управление, виртуализация ресурсов и интеграция с облачными сервисами, открывают новые горизонты для автоматизации и оптимизации сетевых процессов. Несмотря на существующие вызовы, связанные с безопасностью и совместимостью, SDN продолжает развиваться и внедряться в различные сферы, от крупных корпоративных сетей до облачных решений, что делает его важным элементом современного сетевого ландшафта.

1.2 Архитектура SDN.

Архитектура программно-конфигурируемых сетей (SDN) является фундаментальной основой для создания гибких, управляемых и программируемых сетевых инфраструктур. Её ключевая идея заключается в отделении плоскости управления (Control Plane), которая принимает решения о передаче трафика, от плоскости данных (Data Plane), которая непосредственно отвечает за пересылку пакетов. Это разделение радикально меняет традиционный подход к построению и управлению сетями, создавая предпосылки для внедрения сложной автоматизации.

1.2.1. Ключевые принципы архитектуры SDN

В основе архитектуры SDN лежат три основных принципа:

Разделение плоскости управления и плоскости данных: В традиционных сетях каждый коммутатор или маршрутизатор совмещает в себе обе функции. В SDN «интеллектуальная» составляющая (управление) централизована в логически едином контроллере, в то время как сетевые устройства (коммутаторы, маршрутизаторы) превращаются в простые, высокопроизводительные элементы пересылки, которые выполняют инструкции контроллера.

Централизованное управление и контроль: логически централизованный контроллер обладает глобальным представлением о состоянии всей сети. Это позволяет применять непротиворечивую политику на всей сети, оптимизировать маршрутизацию и управлять ресурсами более эффективно по сравнению с распределенными протоколами, где каждое устройство имеет лишь локальное представление о топологии.

Программируемость сети: Сеть становится платформой, для которой можно разрабатывать приложения. Через открытые программные интерфейсы (API), такие как Northbound API, приложения могут взаимодействовать с контроллером, запрашивать информацию о сети и диктовать ей поведение. Это позволяет автоматизировать сложные задачи, такие как обеспечение безопасности, балансировка нагрузки и управление качеством обслуживания (QoS).

1.2.2. Уровни архитектуры SDN

Архитектура SDN традиционно представляется в виде трех уровней (см. Рисунок 1), взаимодействующих через стандартизированные интерфейсы.

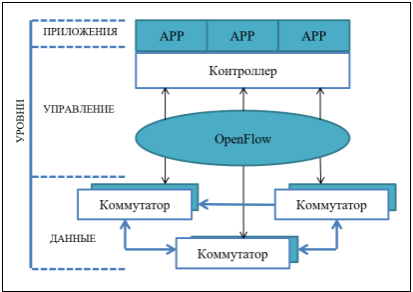


Рисунок 1 - схема

1. Уровень приложения (Data Plane). Данный уровень состоит из физических и виртуальных сетевых устройств: коммутаторов, маршрутизаторов и точек доступа. Их основная функция – быстрая пересылка пакетов на основе правил (flow entries), которые хранятся в их таблицах потоков (flow tables). Эти правила предписаны контроллером. Ключевым протоколом для взаимодействия с контроллером на этом уровне является OpenFlow.

2. Уровень управления (Control Plane). Это «мозг» сети SDN, представленный одним или несколькими контроллерами (например, OpenDaylight, ONOS, Ryu). Контроллер выполняет следующие функции:

Обнаружение топологии: формирует глобальное представление о сетевой топологии и подключенных устройствах.

Управление сетевыми потоками: принимает решения о маршрутизации и вычисляет пути для сетевых потоков.

Программирование плоскости данных: устанавливает, изменяет и удаляет правила в таблицах потоков коммутаторов через протоколы Southbound API (например, OpenFlow).

Предоставление API: предоставляет Northbound API для уровня приложений.

3. Уровень приложений (Application Plane). На этом уровне функционируют бизнес-приложения и сервисы, которые используют программируемость сети для реализации своей логики. Примеры приложений включают:

Системы обнаружения и предотвращения вторжений (IDS/IPS).

Средства балансировки нагрузки.

Системы управления мобильностью.

Механизмы обеспечения качества обслуживания (QoS).

1.2.3. Интерфейсы SDN: Northbound и Southbound API

Связь между уровнями архитектуры осуществляется через два критически важных интерфейса:

Southbound API (Интерфейс «южного направления»): Это интерфейс между контроллером и сетевыми устройствами. Наиболее распространенным и стандартизированным протоколом Southbound API является OpenFlow. Он определяет, как контроллер может добавлять, удалять и изменять записи в таблицах потоков коммутатора, а также как коммутатор должен отправлять контроллеру пакеты для принятия решений (например, первый пакет нового потока).

Northbound API (Интерфейс «северного направления»): Это интерфейс между контроллером и сетевыми приложениями. В отличие от Southbound API, он не имеет единого стандарта. Обычно это RESTful API или специализированные SDK, которые позволяют приложениям запрашивать информацию о сети (топология, статистика) и давать команды на изменение сетевого поведения («разрешить трафик», «направить поток по определенному пути»). Именно Northbound API является ключевым элементом для создания Системы Автоматического Управления, так как через него будет осуществляться взаимодействие между разрабатываемым программным обеспечением и сетевой инфраструктурой.

1.2.4. Обоснование выбора архитектуры для проекта

Для целей данного проекта — создания Системы Автоматического Управления — архитектура SDN является оптимальной по следующим причинам:

Единая точка управления: логически централизованный контроллер предоставляет глобальное представление о сети, что является необходимым условием для реализации корректной и непротиворечивой автоматизации. Все решения могут приниматься на основе полной информации.

Динамическое перепрограммирование: Возможность мгновенно изменять правила пересылки в коммутаторах через Southbound API (OpenFlow) позволяет системе быстро реагировать на события в сети (сбои, перегрузки, атаки) и применять новые политики в реальном времени.

Открытость и программируемость: Northbound API контроллера позволяет разрабатываемой Системе Автоматического Управления выступать в роли специализированного приложения. Система сможет получать телеметрию, анализировать её и отдавать контроллеру команды на перенастройку сети без необходимости взаимодействия с каждым устройством в отдельности.

# 2. Специальная часть.

## 2.1. Обоснование выбора технологий и инструментов.

Для реализации системы автоматического управления была выбрана программно-конфигурируемая сеть (SDN) как наиболее подходящая парадигма, позволяющая централизованно и программно управлять сетевыми процессами. В рамках данного проекта использовался следующий стек технологий:

Контроллер POX - выбор обусловлен его простотой, лёгкостью в освоении и идеальной пригодностью для учебных и исследовательских проектов. POX написан на Python, что упрощает разработку логики управления и интеграцию с другими скриптами. Его модульная архитектура позволяет легко расширять функциональность, что и было использовано для реализации dns\_firewall.

Протокол OpenFlow - является стандартом для взаимодействия между контроллером (POX) и коммутаторами (маршрутизаторами в роли OpenFlow-совместимых коммутаторов). OpenFlow предоставляет необходимый набор инструкций для удалённого управления таблицами потоков на сетевых устройствах.

Операционная система ALT Linux: Использование единой ОС на всех узлах сети (серверах, маршрутизаторах, рабочей станции) упрощает развёртывание и конфигурирование системы, обеспечивая согласованность программной среды и минимизируя риски совместимости.

Модуль dns\_firewall: Этот кастомный модуль, написанный на Python для контроллера POX, является ядром системы автоматического управления в данном проекте. Он реализует функцию безопасности, анализируя DNS-трафик и динамически блокируя нежелательные соединения.

2.2. Архитектура разрабатываемой SDN-сети.

Архитектура спроектированной системы основана на классической трёхуровневой модели SDN и отражает физическую и логическую структуру стенда.

## 2.2.1. Уровни архитектуры.

Уровень инфраструктуры (Data Plane):

Главный маршрутизатор (ISP) выполняет роль ключевого OpenFlow-коммутатора. На нём работает контроллер POX с модулем dns\_firewall. Он является точкой принятия решений для всего проходящего через него трафика.

Дополнительные маршрутизаторы (HQ\_RTR, BR\_RTR) функционируют как простые OpenFlow-коммутаторы, пересылающие пакеты в соответствии с правилами, установленными контроллером на ISP.

Серверы (HQ\_SRV, BR\_SRV) и рабочая станция (HQ\_CLI) являются конечными узлами сети (хостами), генерирующими и потребляющими сетевой трафик.

Уровень управления (Control Plane). Контроллер POX с модулем dns\_firewall: Размещён на маршрутизаторе HQ\_RTR. Этот уровень является "мозгом" сети. Контроллер устанавливает соединение по протоколу OpenFlow со всеми маршрутизаторами (HQ\_RTR, ISP, BR\_RTR), получает от них информацию о событиях (например, о поступлении первого пакета нового потока) и программирует их таблицы потоков.

Уровень приложений (Application Plane). Модуль dns\_firewall является сетевым приложением, работающим поверх контроллера POX. Через Northbound API POX (фактически, через вызовы функций Python) приложение получает пакеты для анализа и диктует политику управления, которую контроллер транслирует в правила OpenFlow.

## 2.2.2. Логика работы системы автоматического управления.

Логическим ядром системы является модуль dns\_firewall. Его алгоритм работы подробно описан на схеме последовательности действий

Описание алгоритма:

Перехват пакета, когда хост (например, HQ\_CLI) пытается установить соединение с google.com, его первый пакет (ARP, DNS-запрос или сам TCP-пакет) попадает на коммутатор HQ\_RTR. Если для этого потока нет правила в таблице потоков, коммутатор инкапсулирует пакет в сообщение Packet-In и отправляет его контроллеру POX.

Анализ DNS-трафика: Модуль dns\_firewall в POX перехватывает пакет. Если это DNS-ответ, он извлекает из него доменное имя и запрошенный IP-адрес.

Проверка политики извлечённое доменное имя (google.com или example.com) проверяется против списка запрещённых. В данном проекте список "чёрный" и статичен, жёстко прописан в коде модуля.

Автоматическое применение политики:

Если домен разрешён, контроллер устанавливает в коммутаторе правило NORMAL, разрешающее коммутатору обрабатывать трафик для этого потока стандартным образом (например, с помощью традиционных таблиц маршрутизации).

Если домен запрещён (например, google.com), система автоматического управления активируется. Контроллер вычисляет правило для таблицы потоков с действием DROP для IP-адреса, который был получен в DNS-ответе для google.com. Это правило немедленно устанавливается на всех управляемых коммутаторах (или только на том, где был обнаружен трафик, в зависимости от реализации).

Блокировка трафика все последующие пакеты, направленные с хоста HQ\_CLI на IP-адрес google.com, будут соответствовать установленному DROP-правилу на коммутаторе ISP и будут отброшены. Соединение не будет установлено.

## 2.3. Физическая и логическая схема тестового стенда.

На основе приведённого описания построена схема тестового стенда (Рисунок 2).

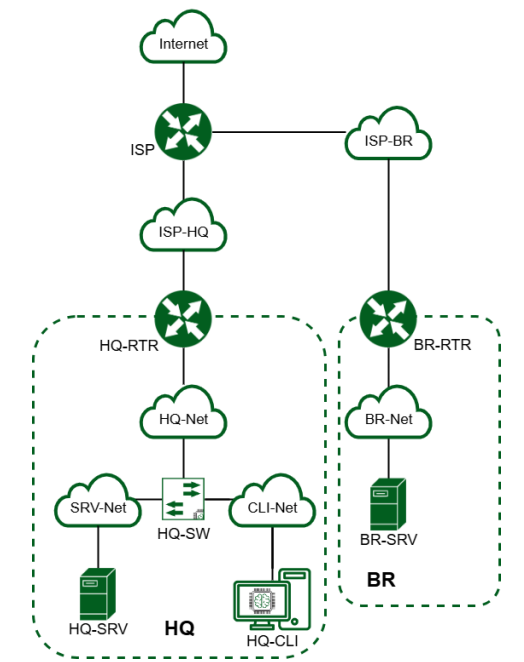


Рисунок 2 – Схема тестового стенда.

## 2.4. Выводы по главе.

В данной главе было проведено проектирование системы автоматического управления сетью SDN. В результате:

Было обосновано использование стека технологий POX, OpenFlow и ALT Linux как эффективного набора для создания прототипа системы.

Была разработана трёхуровневая архитектура, где чётко разделены функции плоскости данных (маршрутизаторы) и плоскости управления (контроллер POX).

Определено ядро системы автоматического управления — модуль dns\_firewall, который реализует функцию динамической блокировки нежелательного трафика на основе анализа DNS.

Подробно описан алгоритм работы системы, демонстрирующий полный цикл автоматизации: от перехвата пакета до применения политики безопасности без вмешательства оператора.

Построена схема тестового стенда, которая будет использована для развёртывания и тестирования системы в следующей главе.

Разработанная архитектура является гибкой и позволяет в будущем расширить функционал системы автоматического управления, добавив новые модули для мониторинга, балансировки нагрузки или защиты от других типов угроз.

3. Практическая часть.

3.1. Создание виртуальной сети.

Для настройки используется готовый макет сети (рис. 3) с настроенными IP-адресами. Разворачиваться SDN будет на машине ISP. Архитектура спроектированной сети базируется на платформе виртуализации Proxmox VE. В качестве программных маршрутизаторов используются легковесные виртуальные машины на базе Alt Linux JeOS, для развертывания сетевых сервисов применяется Alt Linux Server, а в качестве рабочего места конечного пользователя — Alt Linux Workstation.

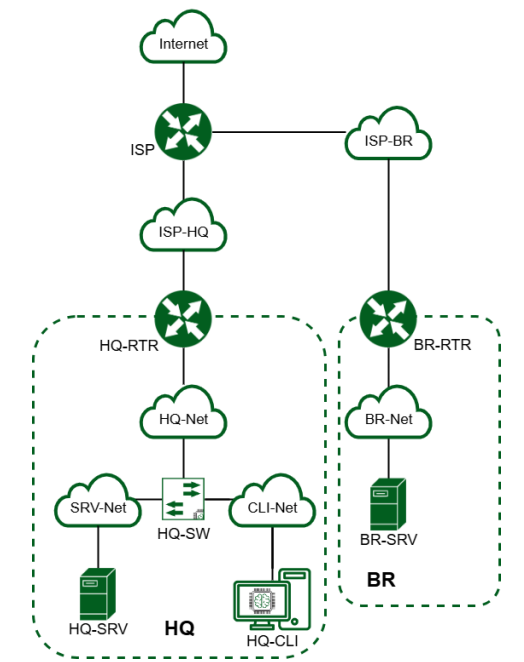


Рисунок 3 – схема сети.

## 3.2 Создание конфигурационного файла SDN.

В качестве SDN сервиса будет использоваться «The POX Network software platform». Платформа использует язык Python. Создаем небольшой конфиг для нашего сервиса. Конфиг будет блокировать все пакеты, исходящие с сайтов «example.com» и «test.org». Загружаем файл (рис. 4) на «GitHub.com».



Рисунок 4 – конфигурация POX

## 3.3 Развёртка SDN на сервере.

Для начала устанавливаем все необходимые программы (Git, Python). Прописываем команду: «apt-get update && apt-get install git python3». Теперь загружаем платформу с Git Hub по команде «cd /opt && git clone [https://github.com/noxrepo/pox»](https://github.com/noxrepo/pox). После этого переходим в директорию /opt/pox/ext и скачиваем наш файл созданный ранее по команде «git clone [https://github.com/P0wix-1/SDN»](https://github.com/P0wix-1/SDN). Запускаем нашу платформу по команде «cd /opt/pox && ./pox.py dns\_firewall».

## 3.4 Настройка OpenVSwitch.

Устанавливаем программу на машинах HQ-RTR и BR-RTR по команде «apt-get install openvswitch». Создаем мост на всех маршрутизаторах по команде «ovs-vsctl add-br br0», после добавляем интерфейсы в OVS по команде «ovs-vsctl add-port br0 ens\*\*». Включаем протокол OpenFlow1.0 «sudo ovs-vsctl set bridge br0 protocols=OpenFlow10». Назначаем контроллер «ovs-vsctl set-controller br0 tcp:192.168.2.1:6633». На сервере должно появиться сообщение о подключении (рис. 5).

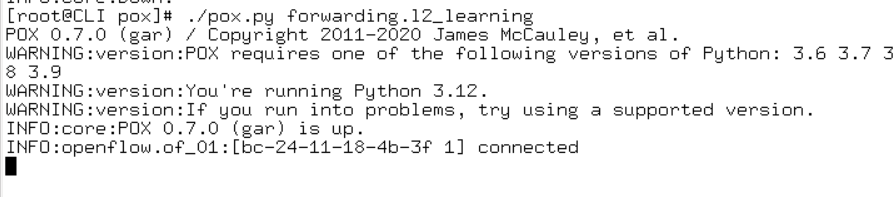


Рисунок 5 – status POX.

## 3.5 Проверка firewall

На HQ-CLI проверяем работоспособность firewall (рис 6).

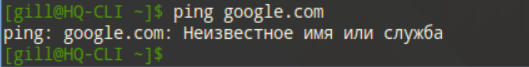


Рисунок 6 – ping google.com.

ЗАКЛЮЧЕНИЕ

Заключение данной работы на тему «Создание системы автоматического управления сетью SDN» подводит итоги проведенного исследования и разработки, а также акцентирует внимание на значимости и перспективах применения технологии программно-определяемых сетей (SDN) в современных условиях. В ходе работы были рассмотрены ключевые аспекты, касающиеся технологии SDN, проектирования системы автоматического управления, а также ее реализации и тестирования. Эти элементы составляют основу для понимания не только текущего состояния, но и будущих направлений развития сетевых технологий.

Технология SDN представляет собой революционный подход к управлению сетями, который позволяет отделить управление сетевыми ресурсами от их физической реализации. Это отделение дает возможность централизованно управлять сетью, что значительно упрощает администрирование и повышает гибкость сетевых решений. В рамках исследования было подробно рассмотрено, как SDN меняет традиционные подходы к проектированию сетевой инфраструктуры, позволяя создавать более динамичные и адаптивные сети, способные быстро реагировать на изменения в требованиях пользователей и бизнес-процессах.

Проектирование системы автоматического управления сетью SDN стало ключевым этапом работы. В этом разделе было акцентировано внимание на разработке архитектуры системы, которая включает в себя компоненты, такие как контроллеры, сетевые устройства и пользовательские интерфейсы. Важным аспектом проектирования стало определение функциональных требований к системе, что позволило создать эффективный и интуитивно понятный интерфейс для администраторов сети. Также был рассмотрен вопрос интеграции существующих сетевых решений в новую архитектуру, что является критически важным для обеспечения совместимости и минимизации затрат на внедрение.

Реализация и тестирование системы стали завершающим этапом работы, в ходе которого была создана рабочая модель системы автоматического управления. В процессе реализации использовались современные инструменты и технологии, что позволило создать гибкое и масштабируемое решение. Тестирование системы подтвердило ее работоспособность и эффективность, выявив при этом ряд возможных улучшений и оптимизаций. В частности, результаты тестирования показали, что система способна обеспечивать высокую производительность и надежность, что является ключевым требованием для современных корпоративных сетей.

Одним из наиболее значимых выводов, сделанных в ходе работы, является то, что технология SDN не только улучшает управление сетями, но и открывает новые возможности для их оптимизации. Автоматизация процессов управления сетевой инфраструктурой позволяет значительно снизить временные затраты на администрирование и повысить уровень безопасности. В условиях постоянного роста объемов данных и увеличения требований со стороны пользователей, автоматизированные системы управления становятся неотъемлемой частью эффективной работы современных организаций.

Кроме того, работа над системой автоматического управления сетью SDN показала, что внедрение таких технологий требует комплексного подхода. Это включает в себя не только технические аспекты, но и организационные изменения внутри компаний. Для успешного внедрения SDN необходимо обеспечить соответствующую подготовку кадров, а также создать культуру инноваций и гибкости в подходах к управлению сетевой инфраструктурой. Безусловно, это является вызовом для многих организаций, однако преимущества, которые предоставляет SDN, делают этот процесс необходимым.

В заключение, создание системы автоматического управления сетью SDN является важным шагом на пути к модернизации и оптимизации сетевых решений. Технология SDN открывает новые горизонты для управления сетями, позволяя создавать более эффективные и адаптивные инфраструктуры. В ходе работы были рассмотрены ключевые аспекты, которые помогут в дальнейшем развитии и внедрении подобных систем. Перспективы развития SDN и автоматизации управления сетями выглядят многообещающими, и можно ожидать, что в будущем мы увидим еще больше инновационных решений, основанных на этой технологии.

Таким образом, работа над темой «Создание системы автоматического управления сетью SDN» не только продемонстрировала практическую применимость технологий SDN, но и выявила ряд направлений для дальнейших исследований и разработок. В частности, стоит обратить внимание на интеграцию SDN с другими современными технологиями, такими как виртуализация сетевых функций (NFV), что может привести к созданию еще более мощных и гибких сетевых решений. Также важным направлением является изучение вопросов безопасности в контексте SDN, поскольку централизованное управление сетевыми ресурсами может представлять собой как возможности, так и риски.

В заключение, можно сказать, что работа над созданием системы автоматического управления сетью SDN является не только теоретически интересной, но и практически значимой. Она открывает новые горизонты для сетевых технологий и предоставляет возможности для создания более эффективных и надежных сетевых решений, что в свою очередь будет способствовать развитию бизнеса и улучшению качества предоставляемых услуг. Надеемся, что результаты данной работы будут полезны как для практиков, так и для исследователей, стремящихся к внедрению передовых технологий в области сетевого управления.

СПИСОК ЛИТЕРАТУРЫ

1. Немет, Э. Unix и Linux: руководство системного администратора [Текст] : 5-е изд. / Э. Немет, Г. Снайдер, Т. Хейн и др. ; пер. с англ. – СПб. : Диалектика, 2020. – 1168 с.
2. Олифер, В. Г. Компьютерные сети. Принципы, технологии, протоколы [Текст] : учебник для вузов / В. Г. Олифер, Н. А. Олифер. – 6-е изд. – СПб. : Питер, 2021. – 944 с. – (Классика computer science).
3. Шрёдер, К. Linux. Книга рецептов [Текст] / К. Шрёдер. – 2-е изд. – СПб. : Питер, 2020. – 592 с.
4. Милл, И. Docker на практике [Текст] / И. Милл, Э.С. Хобсон ; пер. с англ. Д.А. Беликов. – М. : ДМК Пресс, 2020. – 516 с.
5. Столлингс, В. Современные компьютерные сети [Текст] / В. Столлингс ; пер. с англ. – 4-е изд. – СПб. : Питер, 2021. – 864 с.
6. Таненбаум, Э. Компьютерные сети [Текст] / Э. Таненбаум, Н. Феер, Д. Уэзеролл ; пер. с англ. – 6-е изд. – СПб. : Питер, 2022. – 848 с.
7. Вишневский, В. М. Теоретические основы проектирования компьютерных сетей [Текст] / В. М. Вишневский. – М. : Техносфера, 2020. – 512 с.
8. Кульгин, М. В. Технологии корпоративных сетей [Текст] : учебник / М. В. Кульгин. – 2-е изд., перераб. и доп. – СПб. : БХВ-Петербург, 2019. – 864 с.
9. Гордейчик, А. В. Программно-конфигурируемые сети (SDN). Архитектура и применение [Текст] / А. В. Гордейчик, В. И. Гольденберг // Труды института системного программирования РАН. – 2018. – Т. 30, № 2. – С. 67-88.
10. Берсенев, А. С. Анализ протокола OpenFlow и его применение в программно-конфигурируемых сетях [Текст] / А. С. Берсенев, Д. А. Галкин // Информационно-управляющие системы. – 2019. – № 4. – С. 45-53.
11. McKeown, N. OpenFlow: Enabling Innovation in Campus Networks [Электронный ресурс] / N. McKeown, T. Anderson, H. Balakrishnan et al. // ACM SIGCOMM Computer Communication Review. – 2008. – Vol. 38, № 2. – P. 69–74. – Режим доступа: https://dl.acm.org/doi/10.1145/1355734.1355746 (дата обращения: 15.10.2023).
12. Kreutz, D. Software-Defined Networking: A Comprehensive Survey [Электронный ресурс] / D. Kreutz, F. M. V. Ramos, P. E. Veríssimo et al. // Proceedings of the IEEE. – 2015. – Vol. 103, № 1. – P. 14–76. – Режим доступа: https://ieeexplore.ieee.org/document/6994333 (дата обращения: 15.10.2023).
13. POX Controller Official Wiki [Электронный ресурс]. – Режим доступа: https://noxrepo.github.io/pox-doc/html/ (дата обращения: 15.10.2023).
14. OpenFlow Switch Specification (Version 1.5.1) [Электронный ресурс] / Open Networking Foundation. – 2015. – Режим доступа: https://opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf (дата обращения: 15.10.2023).
15. Официальная документация ALT Linux [Электронный ресурс]. – Режим доступа: https://www.altlinux.org/Документация (дата обращения: 15.10.2023).
16. Mininet: An Instant Virtual Network on your Laptop (or other PC) [Электронный ресурс]. – Режим доступа: http://mininet.org/ (дата обращения: 15.10.2023).