

Б. ШЫРЫН¹*, А. ЖУМАДИЛЛАЕВА², М. ОТНМАН³

^{1,2}Евразийский национальный университет имени Л.Н.Гумилева,
Астана, Казахстан;

³Университет Путра Малайзия, Куала-Лумпур, Малайзия.

e-mail: *bexultan.shyryn@gmail.com; ay8222@mail.ru; mothmanupm@gmail.com

СРАВНЕНИЕ КОНТРОЛЛЕРОВ SDN: ОЦЕНКА ПРОИЗВОДИТЕЛЬНОСТИ И ФУНКЦИОНАЛЬНОСТИ

Программно-определяемые сети (SDN) становятся все более популярными в последние годы благодаря своей способности централизовать управление сетью и обеспечить большую гибкость и маневренность. Контроллеры SDN являются ключевым компонентом архитектуры SDN и отвечают за контроль и управление сетью. Однако при широком разнообразии контроллеров SDN, доступных на рынке, выбор подходящего контроллера может оказаться непростой задачей. В этой статье представлен сравнительный анализ нескольких популярных SDN-контроллеров, включая OpenDaylight, Floodlight, Ryu и ONOS. Сравнение основано на двух основных критериях: производительность и функциональность. Показатели производительности включают скорость пересылки пакетов, задержку в плоскости управления и использование памяти, а показатели функциональности – поддержку виртуализации сети, безопасность и масштабируемость. Наша оценка показала, что производительность и функциональность контроллеров значительно различаются, причем некоторые контроллеры в определенных областях работают лучше, чем другие. Например, у OpenDaylight была самая высокая скорость пересылки пакетов, а у Ryu – самая низкая задержка плоскости управления. С точки зрения функциональности, Floodlight обеспечивал наиболее широкие возможности безопасности, а ONOS обладал лучшей масштабируемостью. В целом, в этой статье дана комплексная оценка производительности и функциональности контроллеров SDN, что может помочь сетевым администраторам выбрать подходящий контроллер для своих конкретных нужд.

Ключевые слова: программно-определяемые сети, SDN контроллер, инженерия трафика, OpenFlow, Mininet

Б. А. ШЫРЫН¹*, А. К. ЖУМАДИЛЛАЕВА², М. ОТНМАН³

^{1,2}Л.Н.Гумилев атындағы Еуразия ұлттық университеті, Астана, Қазақстан;

³Путра Малайзия университеті, Куала-Лумпур, Малайзия.

e-mail: *bexultan.shyryn@gmail.com; ay8222@mail.ru; mothmanupm@gmail.com

SDN КОНТРОЛЛЕРІН САЛЫСТЫРУ: ӨНІМДІЛІКТІ ЖӘНЕ ФУНКЦИОНАЛДЫЛЫҚТЫ БАҒАЛАУ

Бағдарламалық қамтамасыз етумен анықталған желі (SDN) соңғы жылдары желіні басқаруды орталықтандыру және үлкен икемділік пен ептілікті қамтамасыз ету қабілетіне байланысты барған сайын танымал бола бастады. SDN контроллері SDN архитектурасының негізгі құрамдас бөлігі болып табылады және желіні басқаруға және басқаруға жауапты. Дегенмен, нарықта қол жетімді SDN контроллерінің алуан түрлілігімен дұрыс контроллерді таңдау өте қиын міндет болуы мүмкін. Бұл мақалада OpenDaylight, Floodlight, Ryu және ONOS сияқты бірнеше танымал SDN контроллерінің салыстырмалы талдауы берілген. Салыстыру екі негізгі критерийге негізделген: өнімділік және функционалдылық. Өнімділік көрсеткіштері пакеттерді

қайта жіберу жылдамдығын, басқару жазықтығының кідірісін және жадты пайдалануды қамтиды, ал функционалдық көрсеткіштерге желіні виртуалдандыруды қолдау, қауіпсіздік және масштабтау кіреді. Біздің бағалауымыз контроллерлердің өнімділігі мен функционалдығы айтарлықтай өзгеретінін көрсетті, кейбір контроллерлер белгілі бір аймақтарда басқаларына қарағанда жақсырақ жұмыс істейді. Мысалы, OpenDaylight ең жоғары пакетті қайта жіберу жылдамдығына ие болды, ал Ryu басқару ұшақтарының ең аз кідірісіне ие болды. Функционалдылық тұрғысынан Floodlight ең кең қауіпсіздік мүмкіндіктерін қамтамасыз етті, ал ONOS ең жақсы масштабтауға ие болды. Тұтастай алғанда, бұл мақала SDN контроллерлерінің өнімділігі мен функционалдығын жан-жақты бағалауды қамтамасыз етеді, бұл желі әкімшілеріне олардың нақты қажеттіліктері үшін дұрыс контроллерді таңдауға көмектесе алады.

Түйін сөздер: Программалы-анықталған желілер, SDN контроллер, трафик инженериясы, OpenFlow, Mininet.

B. SHYRYN¹*, A. ZHUMADILLAYEVA², M. OTHMAN³

^{1,2}L.N. Gumilyov Eurasian National University, Astana, Kazakhstan;

³Universiti Putra Malaysia, Kuala Lumpur, Malaysia.

e-mail: *bexultan.shyryn@gmail.com; ay8222@mail.ru; mothmanupm@gmail.com

COMPARISON OF SDN CONTROLLERS: PERFORMANCE AND FUNCTIONALITY EVALUATION

In recent times, there has been a surge in popularity for Software-defined networking (SDN) due to its ability to centralize network management, resulting in greater flexibility and agility. SDN architecture relies heavily on SDN controllers, which play a significant role in controlling and managing the network. However, with a wide range of SDN controllers available on the market, selecting the ideal one can be a daunting task. To assist with this challenge, this article conducts a comparative analysis of several renowned SDN controllers, such as OpenDaylight, Floodlight, Ryu, and ONOS. The assessment is based on two key criteria: performance and functionality. Performance metrics include packet forwarding speed, control plane latency, and memory usage. In contrast, functionality metrics include network virtualization support, security, and scalability. Our evaluation indicates that there is a significant variation in the performance and functionality of the controllers, with some controllers excelling in specific areas compared to others. For example, OpenDaylight had the highest packet forwarding speed, while Ryu exhibited the lowest control plane latency. Concerning functionality, Floodlight presented the most comprehensive security features, whereas ONOS demonstrated the best scalability. Overall, this article presents a thorough evaluation of SDN controllers' performance and functionality, which can guide network administrators in choosing the most appropriate controller to suit their specific requirements.

Key words: Software-defined networks, SDN controller, traffic engineering, OpenFlow, Mininet.

Введение. Программно-определяемые сети (SDN) произвели революцию в управлении и эксплуатации сетей. В отличие от традиционных сетей, где сетевые устройства самостоятельно принимают решения на основе протоколов маршрутизации, SDN обеспечивает централизованный подход к управлению сетью, отделяя плоскость управления от плоскости данных. Контроллеры SDN являются ключевым компонентом архитектуры SDN и отвечают за управление сетью, контролируя поток трафика и конфигурируя сетевые устройства.

С ростом популярности концепции SDN на рынке появилось большое количество контроллеров SDN с разнообразным набором функций и возможностей. Однако вы-

бор наиболее подходящего SDN-контроллера представляет собой сложную задачу, которая требует учета множества факторов, включая производительность, функциональность, масштабируемость и безопасность.

В этой статье мы представляем сравнительный анализ нескольких популярных SDN-контроллеров, включая OpenDaylight [23], Floodlight [25], Ryu [26] и ONOS [24]. Сравнение основано на двух основных критериях: производительность и функциональность. Показатели производительности включают скорость пересылки пакетов, задержки в плоскости управления и использование памяти, а показатели функциональности – поддержку виртуализации сети, безопасность и масштабируемость.

Научная новизна работы предопределяется тем, что были даны всесторонние оценки производительности и функциональности контроллеров SDN, что может помочь сетевым администраторам выбрать подходящий контроллер для своих конкретных нужд. Мы считаем, что этот сравнительный анализ станет ценным ресурсом для всех, кто заинтересован в развертывании сети на базе SDN и ищет подходящий SDN-контроллер для управления своей сетью.

Архитектура SDN. Программно-определяемые сети (SDN) – это сетевая парадигма, которая позволяет отделить плоскость управления сетью от плоскости данных, обеспечивая программный контроль и управление сетью [1]. SDN обеспечивает централизованную платформу управления, где администраторы сети могут легко контролировать топологию сети, протоколы маршрутизации и политики безопасности с помощью программного контроллера.

Основные функции архитектуры SDN включают:

1. Абстракция плоскости управления. Архитектура SDN отличается от других сетевых архитектур тем, что она разделяет плоскость управления от плоскости данных, что позволяет абстрагировать управление от конкретной аппаратной инфраструктуры.

2. Программируемость. Архитектура SDN обладает возможностью программной настройки сетевой инфраструктуры, что позволяет ее динамически перенастраивать в соответствии с изменяющимися требованиями и условиями сетевой среды [2].

3. Централизованное управление. SDN обеспечивает централизованную платформу управления, которая позволяет администраторам сети управлять топологией сети, протоколами маршрутизации и политиками безопасности с легкостью.

4. Виртуализация сети: SDN позволяет эффективно использовать ресурсы физической сети путем создания нескольких виртуальных сетей поверх нее. Это дает возможность достижения изоляции между сетями и оптимального использования ресурсов.

5. Инженерия трафика. SDN дает возможность управлять потоками трафика, опираясь на запросы, поступающие от приложений, что способствует оптимизации производительности и эффективного использования сети.

6. Автоматизация. SDN обеспечивает среду для автоматизации настройки и управления сетью, что способствует более быстрому развертыванию сети и уменьшению затрат на эксплуатацию [3].

7. Управление сетью на основе политик. С помощью SDN можно создавать правила, которые применяются к потокам сетевого трафика. Эти правила могут использоваться для обеспечения безопасности и соответствия требованиям, определения приоритетов трафика и повышения эффективности использования сетевых ресурсов.

8. Цепочки услуг. SDN предоставляет возможность создавать последовательности сетевых функций, которые называются цепочками сервисов [4]. Эти цепочки могут быть применены к потокам трафика и использоваться для реализации различных сетевых сервисов, включая брандмауэры, балансировщики нагрузки и системы обнаружения вторжений. Создание цепочек сервисов является одним из способов использования SDN для обеспечения безопасности и оптимизации производительности сети.

9. Видимость сети. SDN предоставляет инструменты для мониторинга и анализа потоков сетевого трафика, что позволяет сетевым администраторам получить более глубокое понимание поведения сети и более эффективно диагностировать и устранять проблемы в сети.

10. Управление сетью, не зависящее от поставщика. SDN обеспечивает возможность создания независимых от производителя решений для управления сетью, которые могут управлять сетевыми устройствами различных производителей с помощью единого интерфейса. Это позволяет упростить управление сетью и уменьшить зависимость от конкретных поставщиков оборудования.

11. Динамическая реконфигурация сети: SDN дает возможность сетевым администраторам динамически изменять топологию сети и протоколы маршрутизации, реагируя на изменения в структуре сетевого трафика и устраняя сбои в сети [5].

SDN становится особенной благодаря разделению управляющей плоскости от плоскости передачи данных. На рисунке 1 можно увидеть компоненты архитектуры SDN, включая плоскость данных, управления и приложений.

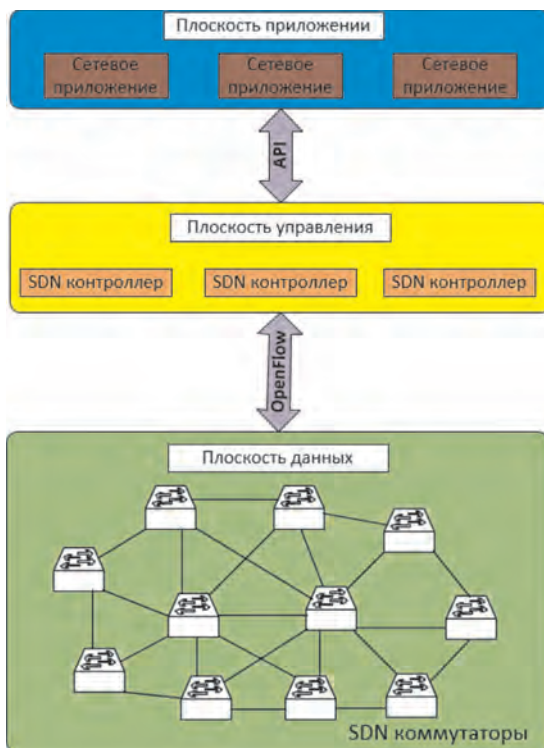


Рисунок 1 – Архитектура программно-определяемых сетей

На первом уровне расположен инфраструктурный слой, который состоит из различных сетевых устройств, таких как коммутаторы и маршрутизаторы, совместимых с протоколом OpenFlow. Они отвечают за передачу пакетов данных между различными устройствами в сети, используя информацию о маршрутизации, полученной из пакетов плоскости данных.

На втором уровне SDN находится сетевая операционная система, которая конфигурирует сетевые устройства. Она содержит один или несколько контроллеров, таких как ONOS, Maestro [6], Ryu, OpenDayLight и OpenFloodLight, которые играют роль мозгов сетей SDN. Контроллеры на плоскости управления принимают решение о пересылке новых потоков и настраивают сетевые устройства. SDN работает в соответствии со стандартом OpenFlow, который является первым стандартом для южного протокола. Эти спецификации были разработаны Stanford's Clean Slate Initiative и сейчас поддерживаются Open Network Foundation (ONF).

Третий уровень SDN – это прикладной уровень, который содержит приложения, взаимодействующие с контроллером SDN для поддержки процесса пересылки. Приложения могут агрегировать информацию от контроллера для создания абстрактного представления сети и принимать решения на основе этой информации.

SDN предлагает множество функций, позволяющих сетевым администраторам эффективнее управлять и оптимизировать свою сетевую инфраструктуру. Эти функции помогают организациям снизить эксплуатационные расходы, повысить производительность и надежность сети, а также более эффективно предоставлять услуги. Благодаря гибкой, масштабируемой и программируемой сетевой инфраструктуре, SDN позволяет организациям более эффективно управлять своими сетевыми ресурсами и предоставлять услуги.

Для оценки производительности и функциональности контроллеров SDN были проведены серии экспериментов в лабораторных условиях. Был использован эмулятор сети Mininet [7], чтобы создать топологию сети, состоящей из нескольких коммутаторов и хостов. После этого, мы подключили коммутаторы к контроллерам SDN и применили различные инструменты бенчмаркинга для измерения производительности и функциональности каждого контроллера.

Для оценки производительности контроллеров SDN мы провели серию экспериментов в лабораторных условиях. Мы измерили несколько параметров, включая скорость пересылки пакетов, задержки в плоскости управления и использование памяти каждым контроллером. Для измерения скорости пересылки пакетов мы использовали генератор трафика Ostinato [8], создавая различные нагрузки и измеряя количество пересылаемых пакетов каждым контроллером. Для измерения задержек в плоскости управления мы использовали утилиту ping для измерения времени, необходимого для передачи пакетов между контроллером и коммутаторами. Кроме того, мы использовали команду Linux top для измерения использования памяти каждым контроллером.

Для оценки функциональности каждого контроллера мы рассмотрели его возможности в области виртуализации сети, обеспечения безопасности и масштабируемости. В частности, для оценки возможностей виртуализации мы изучили способность каждого контроллера создавать и управлять виртуальными сетями, используя технологию Open Virtual Network (OVN) [9]. Для оценки безопасности мы рассмотрели поддержку каждым контроллером различных функций безопасности, таких как контроль

доступа, брандмауэр и обнаружение вторжений. В рамках оценки масштабируемости мы проверили способность каждого контроллера обрабатывать большое количество сетевых устройств и потоков.

Для обеспечения надежности результатов мы повторили каждый эксперимент несколько раз. Кроме того, мы провели статистический анализ для сравнения производительности и функциональности каждого контроллера и определения значимых различий между ними.

Благодаря применению различных методов бенчмаркинга и тестированию разных характеристик каждого контроллера, мы смогли провести всестороннее сравнение производительности и функциональности нескольких известных контроллеров SDN в строго контролируемой лабораторной среде. Это позволило нам получить точные и достоверные результаты, которые позволяют сделать выводы о том, какой контроллер лучше всего соответствует конкретным потребностям и требованиям пользователя.

Контроллеры SDN можно классифицировать по типу архитектуры и функциональным возможностям, которые они предоставляют. На рисунке 2 показаны три основных типа архитектур контроллеров SDN и их специализации. С учетом функциональных возможностей, контроллеры SDN могут быть настроены в соответствии с определенными требованиями сети, такими как управление сетевыми сервисами, определение сетевых политик, обеспечение сетевой безопасности и анализ сетевой активности.

Строгая и систематическая методология для сравнения контроллеров SDN является необходимой. Ее разработка должна учитывать множество факторов, таких как характеристики сети, тестовые сценарии, метрики производительности, инструменты тестирования, а также критерии оценки результатов. Это позволит получить точные и надежные результаты, которые могут быть использованы для принятия обоснованных решений в контексте конкретной сети. Кроме того, методология должна быть прозрачной и доступной для повторного использования. Она должна быть документирована с четкими инструкциями о том, как проводить тесты, какие данные собирать и как анализировать результаты. Это поможет обеспечить согласованность результатов и обеспечит возможность повторения тестов другими исследователями. Также, методология должна быть динамичной и готовой к адаптации к изменяющимся условиям сети и новым технологиям.

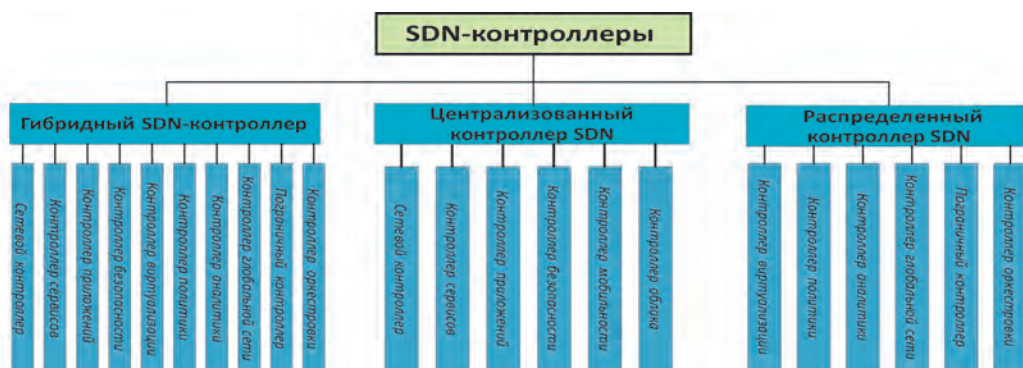


Рисунок 2 – Классификации и специализации контроллеров

Сетевые технологии постоянно развиваются, и методология должна быть способной адаптироваться к этим изменениям, чтобы оставаться актуальной и соответствовать требованиям пользователей. Здесь представлена более подробная информация о каждом этапе методики сравнения контроллеров SDN [10]:

Выбор контроллеров SDN. Первым этапом методологии является выбор контроллеров SDN, которые будут сравниваться. Для этого могут быть использованы различные критерии, такие как популярность, зрелость и поддержка сообщества разработчиков с открытым исходным кодом. Предпочтительными являются контроллеры, которые широко используются и имеют проверенную историю производительности и надежности. Также, они должны быть совместимы с сетевыми устройствами, используемыми в исследовании, и иметь активное сообщество пользователей, которое может предоставить поддержку и помощь.

Топология сети. Для сравнительного исследования необходимо выбрать топологию сети, которая наиболее точно отображает реальные условия. Она должна включать различные типы сетевых устройств, таких как маршрутизаторы, коммутаторы и брандмауэры, а также различные виды трафика, от различных приложений и сервисов. Топология также должна быть масштабируемой, позволяющей изменять количество и типы сетевых устройств и протоколов трафика. Это позволит провести сравнение контроллеров SDN в более реалистичных условиях и получить более точные результаты.

Тестовые сценарии. Для оценки производительности контроллеров SDN необходимо разработать разнообразный набор тестовых сценариев, охватывающих различные функции сети, такие как управление трафиком, обеспечение безопасности и виртуализация сети. Сценарии должны отражать реальные случаи использования и быть разработаны для проведения стресс-тестирования контроллеров в различных сетевых условиях.

Метрики. Дополнительно к метрикам, которые вы указали, можно также измерять масштабируемость и устойчивость контроллеров SDN. Масштабируемость относится к способности контроллера обрабатывать большие объемы трафика и управлять большим количеством сетевых устройств. Устойчивость относится к способности контроллера обрабатывать трафик даже при возникновении сбоев в сети или самих устройств контроллера. Метрики масштабируемости и устойчивости также должны быть определены и измерены в тестовых сценариях.

Инструменты тестирования. При выборе инструментов для тестирования производительности контроллеров SDN, необходимо учитывать их возможности по автоматизации тестирования. Автоматизация позволит значительно ускорить процесс тестирования и сделать его более точным, а также уменьшить возможность ошибок человеческого фактора. Некоторые инструменты для тестирования производительности контроллеров SDN, такие как Mininet и OVS, могут предоставить среду виртуальной сети для тестирования, что также может упростить процесс тестирования и уменьшить затраты на аппаратное обеспечение.

Выполнение тестов. Проведение тестирования несколько раз в различных условиях поможет убедиться в стабильности и последовательности результатов, а также в доказательстве того, что результаты не являются случайными или аномальными.

Документация тестов и их результатов очень важна для обеспечения повторяемости и воспроизводимости тестирования другими исследователями, а также для обеспечения прозрачности и доверия к результатам.

Анализ данных. Для анализа результатов тестирования контроллеров SDN можно использовать различные методы статистического анализа, такие как сравнение средних значений, анализ дисперсии, анализ чувствительности и другие. Важно также учитывать, что результаты тестирования могут быть зависимы от конфигурации сети, используемых устройств и настроек контроллеров. Поэтому необходимо проводить анализ результатов с учетом этих факторов и обеспечивать их повторяемость для достижения точных и надежных результатов.

Для того чтобы корректно сравнивать контроллеры SDN, требуется применять тщательно разработанную и систематическую методологию. Эта методология должна гарантировать правильность, надежность и соответствие результатов с реальными условиями эксплуатации. Благодаря четко определенной методологии, сетевые администраторы смогут принимать обоснованные решения о том, какой контроллер SDN наилучшим образом соответствует их конкретным сетевым потребностям.

Количественная оценка и сравнительный анализ контроллеров. Было проведено несколько исследований, в которых сравнивалась производительность и функциональность различных контроллеров SDN. Например, авторы в [11] сравнили производительность четырех контроллеров SDN, включая Floodlight, OpenDaylight, Ryu и POX [28], и оценили их масштабируемость и устойчивость к сетевым сбоям. Аналогично авторы в [12] оценили производительность и функциональность пяти контроллеров SDN, включая Beacon, Floodlight, NOX [27], OpenDaylight и Ryu, и сравнили их поддержку виртуализации и безопасности сети.

В [13] авторы оценили производительность OpenDaylight, Floodlight и Ryu в среде с несколькими контроллерами и сравнили их масштабируемость и возможности обхода отказа. Они также сравнили производительность этих контроллеров с устаревшей сетевой архитектурой. В [14] авторы оценили производительность и масштабируемость OpenDaylight и Floodlight в среде облачного центра обработки данных и сравнили их производительность с традиционной сетевой архитектурой.

Другие исследования были сосредоточены на оценке производительности конкретных SDN- контроллеров. Например, в [15] авторы оценили производительность контроллеров OpenFlow, включая Beacon, Floodlight и NOX, и сравнили их поддержку виртуализации и безопасности сети. В [16] авторы оценили производительность и масштабируемость Ryu в среде облачного центра обработки данных и сравнили ее с традиционными сетевыми архитектурами.

Эти исследования позволили получить ценные сведения о производительности и функциональности различных контроллеров SDN. Однако по-прежнему существует необходимость в дальнейших исследованиях для оценки производительности и функциональности новых контроллеров SDN, а также для определения лучших практик выбора и развертывания контроллера SDN в сетевой среде.

Возможности контроллера SDN:

1. Northbound API: северный API позволяет интегрировать контроллер SDN с другими приложениями и системами управления, такими как системы управления

облачной инфраструктурой или системы мониторинга. Это позволяет расширить возможности управления сетью и интегрировать ее в общую инфраструктуру организации. Северный API может быть основан на различных протоколах, таких как REST, NETCONF, или протоколе управления приложениями (Application Management Protocol, AMP).

2. Southbound API – это интерфейс, который позволяет контроллеру SDN управлять сетевыми устройствами, работающими на уровне передачи данных. Южный API обычно использует протокол OpenFlow для управления коммутаторами и маршрутизаторами, хотя существуют и другие протоколы для взаимодействия между контроллером и сетевыми устройствами. Через южный API контроллер может направлять потоки трафика и настраивать параметры сетевых устройств для оптимизации работы сети.

3. Обнаружение топологии сети: контроллер SDN способен обнаруживать топологию сети, используя протоколы, такие как OpenFlow и Link Layer Discovery Protocol (LLDP). Это позволяет контроллеру построить карту сети и определить связи между устройствами в ней [19].

4. Управление сетью: контроллер SDN является централизованной платформой управления сетевой инфраструктурой [20], обеспечивая возможность конфигурации сетевых устройств, мониторинга поведения сети и выявления неполадок в работе сети.

5. Инженерия трафика: с помощью контроллера SDN можно определять пути передачи трафика на основе требований приложений. Это позволяет оптимизировать производительность сети и эффективно использовать ее ресурсы. Контроллер может направлять трафик через определенные устройства и каналы связи, учитывая требования к задержке, пропускной способности и качеству обслуживания (QoS), установленные для конкретного приложения. Это также может помочь избежать перегрузки определенных узлов и обеспечить более надежную и предсказуемую работу сети.

6. Политики безопасности: контроллер SDN может управлять безопасностью в сети, определяя политики безопасности и направляя потоки трафика через соответствующие устройства безопасности, такие как брандмауэры и системы обнаружения вторжений. Это обеспечивает более эффективную и удобную защиту сети от угроз безопасности. [21].

7. Виртуализация сети: контроллер SDN обладает возможностью виртуализации сети, которая позволяет создавать несколько виртуальных сетей на основе физической сети. Это позволяет более эффективно использовать ресурсы сети и обеспечивает изоляцию виртуальной сети. Благодаря этому можно повысить безопасность и производительность сети, а также упростить управление ее настройками [22].

Контроллер SDN представляет собой централизованную платформу управления, которая позволяет администраторам сетей настраивать и управлять сетевой инфраструктурой в режиме реального времени. Использование контроллера может помочь организациям улучшить производительность и надежность сети, а также снизить эксплуатационные расходы и увеличить эффективность предоставления услуг. Таблица 1 содержит полный обзор свойств различных контроллеров, но из-за ограниченности пространства и того, что не все контроллеры обеспечивают полную информацию о своих внутренних процессах, мы не можем рассмотреть каждый контроллер отдельно. Вместо этого мы представляем общие свойства и дизайн контроллеров.

Инструментов сравнения контроллеров на сегодняшний день очень много. Мы представили сравнительный анализ трех основных инструментов, часто используемых для оценки в большинстве работ.

CBench [29] – это инструмент бенчмаркинга с открытой лицензией, который используется для оценки производительности контроллеров OF SDN, поддерживающих OF 1.0 и 1.3. Однако из-за ограничений совместимости контроллеры, поддерживающие OF 1.3, могут столкнуться с проблемами производительности. CBench имеет две основные метрики оценки: задержку и пропускную способность. Чтобы измерить задержку, vSwitch отправляет одно сообщение packet_in в сторону контроллера и ждет ответа, повторяя тест несколько раз для получения средней производительности. Общее количество подтверждений, полученных за период тестирования, используется для вычисления средней задержки. Чтобы измерить пропускную способность, каждый vSwitch непрерывно отправляет максимально возможное количество сообщений packet_in для оценки возможностей контроллера.

HCprobe [30] является расширением CBench с открытым исходным кодом, которое использует комбинацию Python и сценариев Shell для расширения возможностей оценки производительности, таких как надежность и масштабируемость. Эмулируемый коммутатор может отправлять уязвимые OF-сообщения контроллерам для проверки их отказоустойчивости и надежности. Тестовый механизм использует ядро Linux для настройки и масштабирования потоков процессора, что позволяет получить более точную статистику производительности контроллера SDN. В итоге, HCprobe предоставляет дополнительные возможности для оценки производительности контроллера SDN.

Hvbench [31, 32] является бесплатным инструментом для бенчмаркинга гипервизоров, созданным на базе SDN и доступным в открытом доступе. Он может использоваться в качестве контейнеров для нескольких контроллеров SDN и устройств плоскости данных, а также предоставляет различные сценарии рабочей нагрузки, позволяющие проводить масштабные оценки производительности гипервизоров SDN. Тем не менее, стоит учитывать, что инструмент имеет ограничения. Во-первых, он работает только с инициализированным гипервизором, поэтому не может использоваться для физической инфраструктуры SDN. Во-вторых, для оценки производительности необходимо предварительно настроить распределение межприходного OF-трафика, что делает невозможным использование других типов пакетов для потоков трафика.

Результаты сравнительного исследования могут показать, что определенные контроллеры SDN работают лучше других в определенных сетевых сценариях. Например, одни контроллеры могут лучше подходить для крупномасштабных сетей с высокой нагрузкой на трафик, а другие – для небольших сетей с более простой структурой трафика. Результаты могут также выявить области, в которых определенные контроллеры превосходят другие, например, политики безопасности, виртуализация сети или организация трафика.

Кроме того, результаты сравнительного исследования могут выявить области, в которых определенные контроллеры имеют ограничения или недостатки. Например, некоторые контроллеры могут иметь ограниченную поддержку определенных сетевых устройств или протоколов, или могут иметь более сложную кривую обучения для

сетевых администраторов. Выявив эти ограничения, сетевые администраторы могут принять обоснованное решение о том, какой контроллер SDN использовать и как его настроить для удовлетворения требований своей сети.

Заключение. Контроллеры SDN предлагают ряд преимуществ для сетевых администраторов, включая повышенную гибкость, масштабируемость и производительность. Однако при наличии на рынке целого ряда контроллеров SDN сетевым администраторам может быть сложно выбрать подходящий контроллер для своих конкретных сетевых требований.

Сравнительное исследование контроллеров SDN может дать ценную информацию о производительности, масштабируемости и надежности различных контроллеров в различных условиях сети. Учитывая такие факторы, как стоимость, простота использования, совместимость, открытость и поддержка сообщества, сетевые администраторы могут принимать обоснованные решения о том, какой SDN- контроллер использовать и как его настроить для удовлетворения требований своей сети.

Кроме того, результаты сравнительного исследования могут выявить области, в которых определенные контроллеры превосходят или имеют ограничения, что поможет сетевым администраторам оптимизировать производительность и функциональность сети. В конечном итоге, выбор контроллера SDN зависит от ряда факторов, специфичных для организации и ее сетевых требований. Тщательное сравнительное исследование может помочь обеспечить выбор и настройку правильного SDN-контроллера для обеспечения оптимальной производительности и функциональности сети.

ЛИТЕРАТУРА

1 Open Networking Foundation. Software-Defined Networking: The New Norm for Networks. Сан-Франциско, 2012.

2 Kreutz, D. и др. Software-defined networking: A comprehensive survey. Proceedings of the IEEE, 2015, том 103, №1, С. 14-76.

3 Ong, L., & Ranganathan, S. Software-defined networking (SDN) and IP traffic engineering (TE): Optimal routing algorithms for dynamic traffic engineering in SDN-based IP networks // IEEE Communications Magazine. – 2014. – Т. 52, № 7. – С. 52-60. – doi: 10.1109/MCOM.2014.6852114.

4 Gember, A., Tootoonchian, A., Casado, M., Sherwood, R., и McKeown, N. (2014). OpenNF: Enabling innovation in network function control [OpenNF: Обеспечение инноваций в управлении функциями сетевого уровня]. ACM SIGCOMM Computer Communication Review, 44(2), 68-73. doi: 10.1145/2602204.2602218.

5 Bianco, A., Gallo, P., & Salsano, S. (2013). Открытые вопросы программно-определяемых сетей. Computer Networks, 57(3), 708-728. doi:10.1016/j.comnet.2012.08.027.

6 Чжэн Цай, Алан Л. Кокс, и Т. С. Юджин Нг. 2010. Maestro: Система масштабируемого управления OpenFlow. Технический отчет TR10-08. Райс Университет.

7 Команда Mininet. (2018). Mininet: мгновенная виртуальная сеть на вашем ноутбуке (или другом ПК). Mininet. Получено 22 сентября 2022 года с сайта <http://mininet.org>.

8 Ostinato. (2022). Network traffic generator tool. Получено 22 сентября 2022 года с сайта <https://ostinato.org/>.

9 Kumar, S., & Parashar, M. (2018). Towards scalable virtual network function chaining in Open Virtual Network. Proceedings of the IEEE Conference on Network Function Virtualiza-

tion and Software Defined Networks (NFV-SDN'18), С. 1-6. URL: <https://ieeexplore.ieee.org/document/8725771>.

10 Rodrigues, L., Sousa, J. и Granville, L. A methodology for comparing OpenFlow controllers. *Journal of Network and Computer Applications*, 2015, vol. 57, С. 49-61.

11 Halepovic, E., Vajric, S. и Klobucar, M. Comparative analysis of SDN controllers based on performance and scalability. In: *IEEE 13th International Symposium on Intelligent Systems and Informatics (SISY)*, Subotica, 2015, С. 249-254.

12 Huang, M.C. и Huang, C.Y., "Comparison of OpenFlow controllers," *Proceedings of the 2014 International Symposium on Computer, Consumer and Control (IS3C)*, Taichung, 2014, С. 457-460.

13 Ha, S.S., Kim, S.Y., and Lee, Y.H., "Performance evaluation of SDN controllers in a multi-controller environment," *Proceedings of the 2016 International Conference on Information Networking (ICOIN)*, Kota Kinabalu, 2016, С. 465-470.

14 Sherif, M.A., Kim, H., and Kim, Y., "Performance evaluation of OpenDaylight and Floodlight controllers for cloud data center networks," *Journal of Network and Computer Applications*, том 82, С. 1-15, 2017.

15 Kumar, S. and Gopalan, J.P., "Comparative performance analysis of OpenFlow controllers for virtualized networks," *Journal of Network and Computer Applications*, том 59, С. 169-182, 2016.

16 Dinh, D.D., Liu, A., Kumar, A., Liu, K., and Zhang, B., "Ryu: A component-based software-defined networking framework," *Journal of Network and Computer Applications*, том 67, С. 165-175, 2016.

17 Kreutz, D., Ramos, F.M.V., Verissimo, P., Rothenberg, C.E., Azodolmolky, S., and Uhlig, S., "Software-Defined Networking: A Comprehensive Survey," *IEEE Communications Surveys & Tutorials*, том 17, №1, С. 27-51, 2015.

18 Goransson, P. and Black, C., *Software Defined Networks: A Comprehensive Approach*, O'Reilly Media, Inc., 2016.

19 Bombal, D., *SDN and OpenFlow for beginners with hands-on labs*, Udemey Course, 2021.

20 Subramanian, S., *Software Defined Networking (SDN) with OpenStack*, Packt Publishing, 2016.

21 Das, S., *Mastering OpenDaylight*, Packt Publishing, 2016.

22 Nadeau, T.D. and Gray, K., *SDN: Software Defined Networks*, O'Reilly Media, Inc., 2013.

23 Linux Foundation. (2018). OpenDaylight Project. Получено 9 июня 2022 года с сайта <https://wiki.opendaylight.org>

24 Berde, P., Gerola, M., Hart, J., Higuchi, Y., Kobayashi, M., Koide, T., Lantz, B., et al. (2014). ONOS. In *Proceedings of the 3rd Workshop on Hot Topics in Software Defined Networking (HotSDN'14)* (С. 1–6). ACM, New York, NY.

25 Atlassian. (2016). Floodlight Controller. Получено 15 декабря 2022 года с <https://floodlight.atlassian.net/wiki/spaces/floodlightcontroller/pages/1343542/Getting+Started>.

26 Ryu SDN Framework Community. (2017). Ryu SDN Framework. Получено 12 декабря 2022 года с сайта <https://github.com/faucetsdn/ryu>

27 Gude, N., Koponen, T., Pettit, J., Pfaff, B., Casado, M., McKeown, N., & Shenker, S. (2008). NOX: Towards an operating system for networks. *ACM SIGCOMM Computer Communication Review*, 38(3), 105–110.

28 POX. (2018). POX Controller Manual. Получено 6 января 2023 года с сайта <https://noxrepo.github.io/pox-doc/html/>

29 Sherwood, R., & Yap, K. K. (2011). Cbench Controller Benchmark. Получено 6 января 2023 года с сайта <https://github.com/andi-bigswitch/oflops/tree/master/cbench>.

30 Shalimov, A., Zuikov, D., Zimarina, D., Pashkov, V., & Smeliansky, R. (2013). Advanced study of SDN/OpenFlow controllers. In *Proceedings of the 9th Central and Eastern European Software Engineering Conference* (pp. 1–6). ACM, New York, NY.

31 Sieber, C. (2018). hvbench: An open and Scalable SDN Hypervisor Benchmark. Получено 21 января 2023 года с сайта <https://github.com/tum-lkn/perfbench>.

32 Sieber, C., Blenk, A., Basta, A., & Kellerer, W. (2016). hvbench: An open and scalable SDN network hypervisor benchmark. In Proceedings of the IEEE NetSoft Conference and Workshops (C. 403–406). IEEE, Los Alamitos, CA.

REFERENCES

1 Open Networking Foundation. Software-Defined Networking: The New Norm for Networks. San Francisco, 2012.

2 Kreutz, D. et al. Software-defined networking: A comprehensive survey. Proceedings of the IEEE, 2015, vol. 103, no. 1, pp. 14-76.

3 Ong, L., & Ranganathan, S. Software-defined networking (SDN) and IP traffic engineering (TE): Optimal routing algorithms for dynamic traffic engineering in SDN-based IP networks // IEEE Communications Magazine. – 2014. – Т. 52, № 7. – pp. 52-60. – doi: 10.1109/MCOM.2014.6852114.

4 Gember, A., Tootoonchian, A., Casado, M., Sherwood, R., & McKeown, N. (2014). OpenNF: Enabling innovation in network function control. ACM SIGCOMM Computer Communication Review, 44(2), 68-73. doi: 10.1145/2602204.2602218.

5 Bianco, A., Gallo, P., & Salsano, S. (2013). Open issues in software defined networking. Computer Networks, 57(3), 708-728. doi:10.1016/j.comnet.2012.08.027.

6 Zheng Cai, Alan L. Cox, and T. S. Eugene Ng. 2010. Maestro: A System for Scalable OpenFlow Control. Technical Report TR10-08. Rice University.

7 Mininet Team. 2018. Mininet: An Instant Virtual Network on Your Laptop (or other PC) Mininet. Retrieved September 22, 2022 from <http://mininet.org/>.

8 Ostinato. (2022). Network traffic generator tool. Retrieved September 22, 2022 from <https://ostinato.org/>.

9 Kumar, S., & Parashar, M. (2018). Towards scalable virtual network function chaining in Open Virtual Network. Proceedings of the IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN'18), pp. 1-6. URL: <https://ieeexplore.ieee.org/document/8725771>.

10 Rodrigues, L., Sousa, J. and Granville, L. A methodology for comparing OpenFlow controllers. Journal of Network and Computer Applications, 2015, vol. 57, pp. 49-61.

11 Halepovic, E., Bajric, S. and Klobucar, M. Comparative analysis of SDN controllers based on performance and scalability. In: IEEE 13th International Symposium on Intelligent Systems and Informatics (SISY), Subotica, 2015, pp. 249-254.

12 Huang, M.C. and Huang, C.Y., "Comparison of OpenFlow controllers," Proceedings of the 2014 International Symposium on Computer, Consumer and Control (IS3C), Taichung, 2014, pp. 457-460.

13 Ha, S.S., Kim, S.Y., and Lee, Y.H., "Performance evaluation of SDN controllers in a multi-controller environment," Proceedings of the 2016 International Conference on Information Networking (ICOIN), Kota Kinabalu, 2016, pp. 465-470.

14 Sherif, M.A., Kim, H., and Kim, Y., "Performance evaluation of OpenDaylight and Floodlight controllers for cloud data center networks," Journal of Network and Computer Applications, vol. 82, pp. 1- 15, 2017.

15 Kumar, S. and Gopalan, J.P., "Comparative performance analysis of OpenFlow controllers for virtualized networks," Journal of Network and Computer Applications, vol. 59, pp. 169-182, 2016.

16 Dinh, D.D., Liu, A., Kumar, A., Liu, K., and Zhang, B., "Ryu: A component-based software-defined networking framework," Journal of Network and Computer Applications, vol. 67, pp. 165-175, 2016.

17 Kreutz, D., Ramos, F.M.V., Verissimo, P., Rothenberg, C.E., Azodolmolky, S., and Uhlig, S., "Software-Defined Networking: A Comprehensive Survey," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 27-51, 2015.

18 Goransson, P. and Black, C., *Software Defined Networks: A Comprehensive Approach*, O'Reilly Media, Inc., 2016.

19 Bombal, D., *SDN and OpenFlow for beginners with hands-on labs*, Udemy Course, 2021.

20 Subramanian, S., *Software Defined Networking (SDN) with OpenStack*, Packt Publishing, 2016.

21 Das, S., *Mastering OpenDaylight*, Packt Publishing, 2016.

22 Nadeau, T.D. and Gray, K., *SDN: Software Defined Networks*, O'Reilly Media, Inc., 2013.

23 Linux Foundation. (2018). OpenDaylight Project. Retrieved June 9, 2022 from <https://wiki.opendaylight.org>

24 Berde, P., Gerola, M., Hart, J., Higuchi, Y., Kobayashi, M., Koide, T., Lantz, B., et al. (2014). ONOS. In *Proceedings of the 3rd Workshop on Hot Topics in Software Defined Networking (HotSDN'14)* (pp. 1–6). ACM, New York, NY.

25 Atlassian. (2016). Floodlight Controller. Retrieved December 15, 2022 from <https://floodlight.atlassian.net/wiki/spaces/floodlightcontroller/pages/1343542/Getting+Started>.

26 Ryu SDN Framework Community. (2017). Ryu SDN Framework. Retrieved December 12, 2022 from <https://github.com/faucetsdn/ryu>.

27 Gude, N., Koponen, T., Pettit, J., Pfaff, B., Casado, M., McKeown, N., & Shenker, S. (2008). NOX: Towards an operating system for networks. *ACM SIGCOMM Computer Communication Review*, 38(3), 105–110.

28 POX. (2018). POX Controller Manual. Retrieved January 6, 2023 from <https://noxrepo.github.io/pox-doc/html/>

29 Sherwood, R., & Yap, K. K. (2011). Cbench Controller Benchmark. Retrieved January 6, 2023 from <https://github.com/andi-bigswitch/oflops/tree/master/cbench>.

30 Shalimov, A., Zuikov, D., Zimarina, D., Pashkov, V., & Smeliansky, R. (2013). Advanced study of SDN/OpenFlow controllers. In *Proceedings of the 9th Central and Eastern European Software Engineering Conference* (pp. 1–6). ACM, New York, NY.

31 Sieber, C. (2018). hvbench: An open and Scalable SDN Hypervisor Benchmark. Retrieved January 21, 2023 from <https://github.com/tum-lkn/perfbench>.

32 Sieber, C., Blenk, A., Basta, A., & Kellerer, W. (2016). hvbench: An open and scalable SDN network hypervisor benchmark. In *Proceedings of the IEEE NetSoft Conference and Workshops* (pp. 403–406). IEEE, Los Alamitos, CA.