СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Компьютерные сети как основополагающая инфраструктура — важнейший фактор развития современных информационных технологий, однако архитектура сети, основы которой закладывались еще в конце шестидесятых годов, устарела и сейчас не всегда способна адекватно и эффективно реагировать на новые задачи и потребности. Рост количества и разнообразия мобильных устройств, развитие различных облачных технологий, беспроводной связи привели к тому, что сегодня число их пользователей превысило число пользователей сетей с фиксированной связью. Однако рост мощности мобильных терминалов подталкивает увеличение вычислительной емкости приложений, что, в свою очередь, требует увеличения пропускной способности каналов связи — объем мобильного трафика растет в геометрической прогрессии, а виды трафика становятся все более разнообразными. По данным ведущих производителей сетевого оборудования, трафик удваивается примерно каждые девять месяцев, что в ближайшие годы приведет к огромному увеличению нагрузки на несколько порядков. В то же время сегодня эффективность доступного спектра частот для мобильных сетей уже близка к насыщению.

Развитие микропроцессорной техники и телекоммуникаций привело к тому, что сейчас на каждого человека приходится в среднем около 40 чипов, однако появляются все новые и новые сетевые устройства, и внесение любых изменений в их существующие конфигурации трудоемко, затратно и практически невозможно без привлечения производителя. Нельзя гарантировать, что программно-аппаратные средства производителя содержат только ту функциональность, которая описана в документации, а в сетях ситуация может быть еще сложнее — такая функциональность может быть распределенной. Средства построения сетей сегодня проприетарны, являются частной собственностью авторов или правообладателей, их основной функционал реализован аппаратно и закрыт для изменений со стороны владельцев сетей.

Рост количества и разнообразия контента, развитие сервисов и масштабов их охвата привели к изменению парадигмы организации вычислений — на место клиент-серверной архитектуры пришли центры обработки данных и облачные сервисы, а файловые системы и базы данных трансформировались в сети хранения данных. Однако объем трафика в Интернете за последние пять лет вырос втрое, а пропускная способность современных каналов связи при

существующих методах и средствах управления трафиком в сетях уже близка к исчерпанию — нынешние темпы роста пропускной способности сети не в состоянии удовлетворять растущие потребности пользователей.

Одновременно с ростом количественных показателей нагрузки на сети усложнились задачи их управления — увеличились их перечень, значимость и критичность, причем на фоне повышения требований к безопасности и надежности. Сети сейчас строятся на базе устройств, которые каждый день усложняются, поскольку вынуждены поддерживать все больше распределенных стандартных протоколов, одновременно используя закрытые интерфейсы. В таких условиях провайдеры не могут оперативно вводить новые сервисы, а производители сетевого оборудования не могут быстро и качественно модернизировать свои изделия для удовлетворения требований заказчиков. Как следствие, поддержка и управление сложной сетевой инфраструктурой стали искусством, а не инженерией, что отчасти подтверждается увеличением числа сетевых атак, вирусов и других сетевых угроз, свидетельствующих о том, что вопросы безопасности до сих пор не имеют надежных решений.

Ответом на кризис компьютерных сетей стало появление принципиально нового подхода к их построению — программно-конфигурируемых сетей.

Цель данной работы — разработать систему управления коммутатором для программно-конфигурируемой сети с функцией мониторинга трафика, в среде моделирования `MiniNet`, на основе протокола `OpenFlow`, которая сможет использоваться на физическом устройстве.

В первой главе рассматриваются теоретические основы программно-конфигурируемых сетей. Во второй главе исследуются программные средства для выполнения поставленных задач, более подробно рассматриваются `OpenFlow` контроллеры, а так же производится сравнительный анализ существующих сетевых операционных систем для программно-конфигурируемых сетей. В третьей главе описан процесс реализации системы управления коммутатором, а так же описан процесс реализации системы мониторинга сети.

1 Программно-коммутируемые сети

1.1 Архитектура программно-конфигурируемых сетей

Программно-коммутируемые сети (программно-конфигурируемые сети, Software Defined Networks, SDN) — развивающаяся архитектура сети, где функция управления сетью разделена с функцией передачи данных и полностью программируема. Идея таких сетей была сформулирована специалистами университетов Стэнфорда и Беркли еще в 2006 году, и сейчас SDN находит все большую поддержку среди IT-компаний.

Основные концепции SDN:

— разделение процессов передачи и управления данными;
— единый, унифицированный, независящий от поставщика интерфейс между уровнем управления и уровнем передачи данных;
— логически централизованное управление сетью, осуществляемое с помощью контроллера с установленной сетевой операционной системой и реализованными поверх сетевыми приложениями;
— виртуализация физических ресурсов сети. [1]

В архитектуре SDN можно выделить три уровня:

— инфраструктурный уровень, предоставляющий набор сетевых устройств (коммутаторов и каналов передачи данных);
— уровень управления, включающий в себя сетевую операционную систему, которая обеспечивает приложениям сетевые сервисы и программный интерфейс для управления сетевыми устройствами и сетью;
— уровень сетевых приложений для гибкого и эффективного управления сетью.

1.2 Преимущества SDN

Архитектура программно-коммутируемых сетей и предлагаемый централизованный подход дает следующие преимущества по сравнению с традиционными сетями с распределенным управлением передачей данных:

— программируемость и гибкость управления сетью, значительное упрощение возможности модификации управления сетью за счет создания новых приложений или модификации существующих, автоматизация управления и администрирования сетями;
— адаптивность управления сетью, то есть возможность изменять поведе-

ние и состояние сети в режиме реального времени с учетом изменяющихся условий функционирования и адаптироваться к ним, адаптироваться к меняющимся потребностям пользователей сетей за счет создания новых сетевых приложений и сервисов;

— на разработку сетевых приложений требуется значительно меньше времени по сравнению с ручным переконфигурированием всей сети;
— независимость от оборудования и проприетарного программного обеспечения производителей сетевого оборудования;
— возможность независимого развертывания уровня управления и уровень передачи данных;
— возможность независимого масштабирования уровня управления и уровень передачи данных;
— повышение надежности за счет снижения объема распределенного состояния для управления. Вместо имеющихся распределенных протоколов, которые работают на каждом узле сети, каждый из них поддерживает базу данных распределенных копий состояний каналов в каждом узле, однако такая информация может быть собрана централизованно в одном месте — на контроллере. Таким образом, такая централизованная база данных будет содержать значительно меньше несогласованной информации, и такой подход позволит уменьшить вероятность циклов в сети;
— упрощение структуры и логики сетевых устройств, поскольку теперь им не требуется обрабатывать огромное количество стандартов и протоколов, а достаточно выполнять только инструкции, полученные от контроллера;
— снижение стоимости коммутаторов и сетевой инфраструктуры в целом за счет вынесения «мозгов роутеров» в контроллер. [2]

1.3 Сетевой мост

Сетевой мост — это сетевое устройство, предназначенное для объединения сегментов сети передачи данных в единую сеть. Он работает на канальном (втором) уровне модели OSI. В отличие от концентратора, который работает на физическом уровне, сетевой мост не просто транслирует кадры полученные с одного порта устройства на другие, а анализирует заголовок и отправляет на какой-либо один порт, либо не передает никуда. Однако в отличие от маршрутизатора не имеет таблицы маршрутизации, является настраиваемым

устройством и работает по заранее заложенным в нем принципам. Сетевой мост используется в нескольких сетевых технологиях, однако наибольшее распространение нашел в Ethernet. [3]

1.4 Ситуация с SDN в России

Как показывают опросы, примерно $2/3$ российских специалистов отметили, что их интерес к SDN пока носит лишь чисто теоретический характер.

Таким образом, концепции SDN в России находятся на стадии формирования. По мнению аналитиков для ускорения готовности к коммерческому внедрению необходимы изучение и адаптация нормативно-правовых аспектов, технических требований и вопросов регулирования; создание ассоциаций научно-исследовательских университетов, лабораторий, профильных академических институтов, представителей телеком-сообщества, стартапов, российских разработчиков; привлечение в Россию ведущих зарубежных экспертов в области SDN; интеграция российских исследователей и экспертов в международные проекты, связанные с SDN. [4]

Прогнозы SDN в России носят пока не столь радужный характер, как для мирового рынка. Объем российского сегмента SDN к концу 2018 года составит 40 млн. долл. Такова оценка ведущих мировых вендоров. Основными пользователями SDN станут владельцы крупных центров обработки данными и федеральные операторы связи.

1.5 Технология OpenFlow

Одной из наиболее перспективных и развивающихся реализаций подхода программно-конфигурируемых сетей является технология OpenFlow. Основным её документом является спецификация OpenFlow, в которой описываются основные компоненты OpenFlow-сети, принципы работы и взаимодействия компонентов. Стандартизирующей организацией для спецификации является ONF — Open Networking Foundation. [5]

Согласно спецификации основными компонентами OpenFlow сети являются:

1. контроллер, содержащий:
   а) сетевую операционную систему;
   б) сетевые приложения.
2. OpenFlow коммутатор;

3. защищенный канал между контроллером и коммутатором;
4. протокол OpenFlow.

Общий принцип функционирования OpenFlow-сети заключается в том, что каждый OpenFlow коммутатор устанавливает защищенный канал с контроллером, посредством которого контроллер управляет им. Взаимодействие между коммутаторами и контроллером осуществляется посредством сообщений протокола OpenFlow. Контроллер получает информацию об изменении состояний элементов в сети, на основе которой он конфигурирует сетевое оборудование, управляет сетевой инфраструктурой и потоками данных в сети.

1.6 Протокол OpenFlow

Протокол поддерживает три типа сообщений:

— controller-to-switch — инициируются контроллером и используются для непосредственного контроля и управления состоянием коммутатора;
— асинхронные сообщения инициируются коммутатором и используются для уведомления контроллера о событиях в сети (ошибках, отказах) и изменениях состояния коммутатора.
— симметричные сообщения могут инициироваться как коммутатором, так и контроллером.

Сообщения controller-to-switch могут требовать или не требовать ответа от коммутатора. Коммутаторы посылают асинхронные сообщения контроллеру для уведомления о прибытии пакета, изменении состояния коммутатора или ошибке. Симметричные сообщения отправляются без запроса в любом направлении.

Протокол OpenFlow обеспечивает надежную доставку сообщений и их обработку, но не обеспечивает автоматические подтверждения о доставке или упорядочению обработку сообщений. Обработка сообщений обеспечивается для основного соединения и дополнительных соединений, использующих надежную передачу данных, но не поддерживается на дополнительных соединениях, использующих ненадежную передачу данных. Доставка сообщений гарантируется до тех пор, пока полностью не откажет OpenFlow канал, в этом случае контроллер не может делать какие-либо предположения о состоянии коммутатора.

1.7 Пакеты, коммутация и маршрутизация

Различают внутреннюю (по отношению к сети) и внешнюю, или межсетевую маршрутизацию. Эти две системы используют различные протоколы, например, для внутренней маршрутизации широко используется протокол OSPF, а стандартом межсетевой маршрутизации является BGP. Ключевым элементом сетевой инфраструктуры является маршрутизатор, который выполняет две функции: маршрутизацию и пересылку пакетов с одного интерфейса на другой. Хотя иногда под маршрутизацией понимают обе функции, на самом деле они существенно различаются. [6]

В алгоритмическом плане пересылка пакетов достаточно проста, и основной задачей является производительность. Отправителями и получателями пакетов являются интерфейсы маршрутизатора, а определение адресата осуществляется с помощью таблицы передачи — Forwarding Information Base (FIB).

Задачей же маршрутизации является построение собственной таблицы, таблицы маршрутизации — Routing Information Base (RIB), которая потом транслируется в таблицу FIB. Для построения этой таблицы и используются протоколы маршрутизации, которые на основе информации, полученной от соседних маршрутизаторов (например, о связности и доступности тех или иных маршрутов), и собственной конфигурации (например, статических маршрутов и ограничений, наложенных сетевой политикой маршрутизации), формируют собственное представление о сети и ее топологии. [7]

1.8 Требование к OpenFlow switch

OpenFlow маршрутизатор состоит из одной или нескольких flow-таблиц, групповой таблицы и OpenFlow канала к удаленному контроллеру. Маршрутизатор обменивается сообщениями с контроллером при помощи протокола OpenFlow. Используя данный протокол, контроллер может добавлять, обновлять и удалять flow-записи (flow-entries) во flow-таблицах. Каждая flow-таблица содержит набор flow-записей, каждая запись определяется полями сравнения (matching fields), счетчиками (counters) и набором инструкций (instructions), которые применяются к пакету с совпавшими полями. [8]

Сравнение начинается с первой flow-таблицы и может продолжаться в других таблицах. Поля сравнений flow-записей сравниваются с заголовком

пакета в порядке приоритета (priority — одно из полей сравнения). Если найдена совпадающая flow-запись, к пакету применяются инструкции ассоциированные с данной записью. Если не найдено ни одной записи, результат зависит от конфигурации маршрутизатора (пакет может быть сброшен либо передан контроллеру по OpenFlow каналу для анализа и принятия решения). [9]

1.9 Open vSwitch

Open vSwitch — это виртуальный многоуровневый маршрутизатор разрабатываемый под лицензией Apache 2.0. Open vSwitch условно можно разделить на две части — user-space и kernel-space.

User-space состоит из нескольких наиболее важных компонент: это демоны, которые реализуют коммутатор с таблицей потоков, ядро Open vSwitch, и набор утилит, которые позволяют конфигурировать коммутатор, его базу данных и напрямую посылать сообщения в ядро.

1.10 Контроллер, сетевая ОС и сетевые приложения

Ключевым элементом в программно-коммутируемых сетях является контроллер. Под контроллером понимают специальное ПО установленное на физическом сервере, осуществляющее контроль, управление состоянием сети и ее элементами, а так же управление потоками данных в сети. Контроллер состоит из сетевой операционной системы и набора сетевых приложений, функционирующих поверх нее. Сетевая ОС осуществляет непосредственное взаимодействие с элементами сети, коммутаторами, контролирует и формирует состояние сети на основе сообщений от элементов сети, предоставляет возможности взаимодействия приложений между собой, распространяет управляющие воздействия на элементы сети, то есть осуществляет управление сетевыми ресурсами сети. Сетевые приложения, написанные администратором сети, на основе информации о состоянии сети осуществляют непосредственное управление трафиком. Таким образом, на контроллере должно быть установлено хотя бы одно приложение.

1.11 Подходы к реализации сети управления в SDN сетях

Для корректной работы OpenFlow коммутатора необходимо, чтобы он мог устанавливать и поддерживать соединение с соответствующим контроллером. Существует два основных подхода к организации такого соединения:

— передавать сообщения OpenFlow по физически независимому каналу, изолированному от сети передачи данных;
— для передачи управляющей информации можно использовать непосредственно сеть передачи данных, которой управляет контроллер.

Первый подход получил название out-of-band, а второй — in-band. В случае использования out-of-band подхода в OpenFlow коммутаторе выделяется отдельный порт. Вся управляющая информация между контроллером и коммутатором передается через этот порт, для передачи пакетов в сети передачи данных используются другие порты коммутатора. [10] Таким образом, для управления коммутаторами создается отдельная сеть.

В случае in-band подхода для передачи управляющей информации используется та же сеть, что и для передачи пользовательских данных.

2 Программные средства

2.1 Среда MiniNet

MiniNet — это эмулятор компьютерной сети. Под компьютерной сетью подразумеваются простые компьютеры — хосты, коммутаторы, а так же контроллеры OpenFlow. С помощью простейшего синтаксиса в примитивном интерпретаторе команд можно разворачивать сети из произвольного количества хостов, коммутаторов в различных топологиях и все это в рамках одной виртуальной машины. На всех хостах можно изменять сетевую конфигурацию, пользоваться стандартными утилитами, например ipconfig и ping, и даже получать доступ к терминалу. [11]

2.2 OpenFlow контроллеры

Контроллер является ключевым элементом SDN архитектуры, он выступает в роли «мозга» всей сети. Производительность и возможности сети напрямую связаны с характеристиками контроллера. Сам контроллер представляет собой сетевую операционную систему, установленную на выделенном физическом сервере.

Разработано большое количество сетевых операционных систем, в настоящее время известны следующие: NOX-Classic, NOX, POX, SNAC, Beacon, Floodlight, Maestro, Mul, Trema, ONIX, Kandoo, OpenDaylight, Ryu.

2.3 Описание сетевых операционных систем

В данной главе приведено описание сетевых операционных систем, а так же выявлены основные цели их создания.

2.4 Сравнение контроллеров по общим характеристикам

В данной главе приведен сравнительный анализ сетевых операционных систем по общим характеристикам. На основе сравнительного анализа общих характеристик сетевых операционных систем выделены активно развивающиеся, наиболее перспективные и жизнеспособные проекты.

2.5 Сравнение особенностей реализации

В данной главе приведено сравнение особенностей реализации сетевых операционных систем.

3 Практическая часть

3.1 Реализация системы управления коммутатором

В процессе выполнения магистерской работы была создана система управления коммутатором для программно-коммутируемой сети. Для разработки использовалась виртуальная машина Ubuntu, с предварительно установленной средой MiniNet и графической оболочкой Lightweight X11 Desktop Environment (lxde). Был выбран контроллер Ryu. Для виртуализации использовался программный продукт VirtualBox.

Была смоделирована топология в среде MiniNet из трех хостов и одного коммутатора в системе MiniNet. После запуска всех хостов производится запуск системы управления коммутатором. С помощью программы tcpdump прослушивается трафик на каждом хосте. Отправляя пакеты с хоста $h1$ на хост $h2$ используя утилиту ping, пакеты будут переданы хостам согласно описанной выше функциональности.

3.2 Реализация системы мониторинга трафика

В ходе дипломной работы была реализована система постоянного наблюдения за трафиком — мониторинг трафика. Для демонстрации реализованной функциональности была смоделирована топология из трех хостов и одного коммутатора в среде MiniNet. После запуска топологии производится запуск системы управления коммутатором. После запуска системы управления коммутатором, отображается таблица потоков, которая в начальный момент пуста. Отправляя пакеты с хоста $h1$ на хост $h2$ используя утилиту ping, пакеты будут переданы хостам согласно описанной выше функциональности, передача пакетов зарегистрируется системой и статистическая информация поменяется.

Так как изначально таблица потоков пуста, то для определения MAC-адреса хоста будет отправлен широковещательный запрос, что так же отобразится в системе мониторинга. При повторной отправке пакетов и широковещательный запрос будет отсутствовать. Если же отправка пакетов будет производится с другого хоста, первоначально будет отправлен широковещательный запрос.

ЗАКЛЮЧЕНИЕ

В рамках выполнения магистерской работы:

— изучено функционирование SDN сетей;
— изучено функционирование и рассмотрены особенности маршрутизации протокола OpenFlow;
— выполнен сравнительный анализ существующих сетевых операционных систем для программно-коммутируемых сетей;
— смоделирована программно-коммутируемая сеть в среде MiniNet;
— разработана система управления коммутатором для SDN сети, с помощью контроллера RYU.
— разработана функция мониторинга трафика для системы управления коммутатором, поддерживающий протокол OpenFlow.

Получившиеся система может быть использована на физическом устройстве, который поддерживает протокол OpenFlow.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 *Azodolmolky, S.* Software Defined Networking with OpenFlow / S. Azodolmolky. — Packt Publishing, 2013.

2 *Marconett, D.* A Software-Defined Network Controller Architecture / D. Marconett, B. Yoo. — Journal of Network and Systems Management, 2014.

3 *Tanenbaum, A.* Computer Networks 5th Edition / A. Tanenbaum. — Prentice Hall, Indian International Ed, 2010.

4 Рынку SDN обещают двузначный рост [Электронный ресурс]. — URL: https://sk.ru/news/b/press/archive/2014/07/25/rynku-sdn-obeschayut-dvuznachnyy-rost.aspx (дата обращения: 17.02.2018). Загл. с экр. Яз. рус.

5 *Nadeau, D.* SDN - Software Defined Networks / D. Nadeau, K. Gray. — O'Reilly Media, 2013.

6 *Kreutz, D.* Software-Defined Networking: A Comprehensive Survey / D. Kreutz. — Proceedings of the IEEE, 2015.

7 Programmable Networking with Open vSwitch [Электронный ресурс]. — URL: https://events.static.linuxfound.org/sites/events/files/slides/OVS-LinuxCon%202013.pdf (дата обращения: 12.12.2017). Загл. с экр. Яз. англ.

8 *Smiler, K.* OpenFlow Cookbook / K. Smiler. — Packt Publishing, 2015.

9 OpenFlow Switch Specification [Электронный ресурс]. — URL: https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.3.0.pdf (дата обращения: 10.03.2018). Загл. с экр. Яз. англ.

10 Modeling Control Traffic in Distributed Software Defined Networks [Электронный ресурс]. — URL: http://www.diva-portal.org/smash/get/diva2:1038710/FULLTEXT01.pdf (дата обращения: 20.03.2018). Загл. с экр. Яз. англ.

11 Mininet [Электронный ресурс]. — URL: http://mininet.org/ (дата обращения: 12.12.2017). Загл. с экр. Яз. англ.

25.06.2018