электронный научно-технический журнал
ИНЖЕНЕРНЫЙ ВЕСТНИК
Издатель ФГБОУ ВПО "МГТУ им. Н.Э. Баумана". Эл No. ФС77-51036. ISSN 2307-0595

Программно-конфигурируемые сети на базе протокола OpenFlow

05, май 2014
УДК: 004.72

Коломеец А. Е., Сурков Л. В.

Россия, МГТУ им. Н.Э. Баумана
toljio@gmail.com
srk@bmstu.ru

Введение

Рост количества и разнообразия мобильных устройств, развитие различных технологий беспроводной связи привели к тому, что сегодня число их пользователей превысило число пользователей сетей с фиксированной связью. Однако рост мощности мобильных терминалов стимулирует 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 несколько лет приведет к увеличению нагрузки на несколько порядков. По прогнозам компании Cisco в ближайшие 5 лет объем трафика увеличится в 4 раза, причем, мобильный трафик будет удваиваться ежегодно.

Современные компьютерные сети состоят из множества отдельных элементов сети, выполняющих специфические функции: маршрутизаторы, коммутаторы, балансировщики нагрузки, NAT (Network Address Translation), брандмауэры. Технология SDN предлагает отказаться от такой тенденции развития компьютерных сетей, выполнив переход от отдельных сетевых элементов и сети как платформ в целом к программируемым сущностям. С помощью приложений можно оптимизировать транспортные потоки, чтобы найти кратчайший путь, как это делают современные распределенные протоколы маршрутизации, а также оптимизировать сеть для максимального использования связей, сделать мобильность устройств бесшовной или создавать различные домены для разных пользователей.

Чтобы понять, почему писать протоколы маршрутизации так трудно, рассмотрим, как работает маршрутизации в современных сетях. Сети изготовлены из конечных устройств (ПК и сервер) и промежуточных устройств, соединенных кабельной системой. Пакет поступает на один порт, маршрутизатор рассматривает его и отправляет его через

порт, который сделает пакет на один шаг ближе к месту назначения. Каждый маршрутизатор периодически опрашивает своих соседей, к каким сетям он подключен, и каждый сосед накапливает эту информацию и использует для построения дизайна всей сети. Хотя маршрутизаторы делятся топологической информацией между собой, каждый из них выполняет расчет маршрутов самостоятельно. Даже если в топологии сети два близлежащих маршрутизатора рассчитают аналогичные результаты, они не передадут друг другу результаты перекрывающихся расчетов. Так как каждому циклу процессора требуется определенная величина мощности, это дублирование является не энергоэффективным. Выполнение сложных алгоритмов маршрутизации требует большой вычислительной мощности устройств.

Каждый маршрутизатор в отдельности являются дорогим устройством, которое делает те же вычисления, что и все остальные, только чтобы получить немного другой результат. Большие сети требуют больших вычислений. При росте предприятия сеть растет и каждый маршрутизатор должен быть модернизирован для обработки дополнительных вычислений. Типы и количество портов на маршрутизаторе не изменится, но процессор больше не имеет достаточной мощности для выполнения своих алгоритмов. Иногда оказывается достаточным добавление оперативной памяти, но часто необходимо произвести замену процессорного блока на более дорогой. Это хорошая бизнес-модель для сетевых поставщиков: при покупке достаточного количества маршрутизаторов, вам нужно регулярно докупать обновления и модернизировать оборудование. При этом данные процессоры можно получить только у данных сетевых поставщиков, так как это — специализированные, проприетарные процессоры.

На сегодня число фактически (активно) используемых протоколов более 600, и эта цифра далеко не конечная. Итак, можно выделить следующие проблемы современных компьютерных сетей [1]:

- научно-технические — сегодня невозможно контролировать и надежно предвидеть поведение таких сложных объектов, как глобальные компьютерные сети;
- экономические — сети дороги, сложны и требуют для своего обслуживания высококвалифицированных специалистов;
- проблемы развития — в архитектуре современных сетей имеются существенные барьеры для экспериментирования и создания новых сервисов.

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

SDN - архитектура

Концепцию новой сетевой архитектуры программно-конфигурируемых сетей предложили в 2007 году сотрудники Стэнфордского университета [2]. С тех пор сети ПКС развивались преимущественно в научной лаборатории Стэнфорда и Беркли и в промышленно значимых масштабах их еще никто не пробовал. Зарождение данной технологии было связано с несколькими моментами.

- Сети традиционной архитектуры проприетарны, закрыты для исследований и практически любых изменений извне. Оборудование разных производителей часто между собой плохо совместимы.
- Рост трафика в геометрической прогрессии и тезис о том, что сети нынешней архитектуры не смогут с ним справиться на необходимом уровнем качества.
- Рост количества протоколов и их стеков в сети.

Исследователи из Стэнфорда и Беркли предположили, что в компьютерных сетях возможно разделить функции управления и передачи данных.

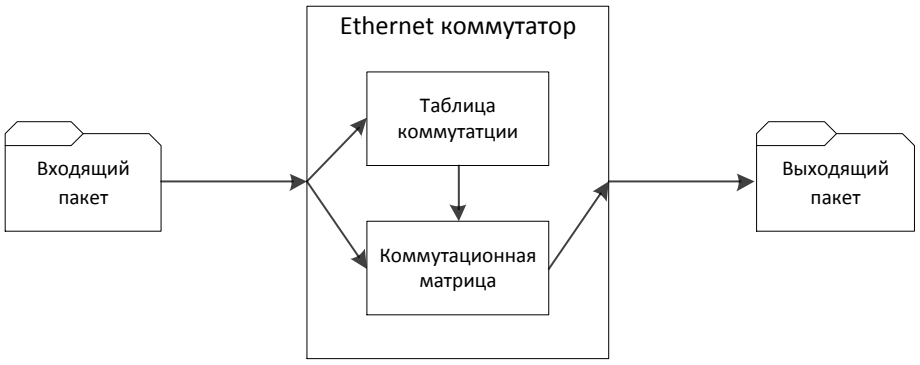

Рисунок 1 - Обработка и передача данных в коммутаторе Ethernet

При передаче данных коммутатор Ethernet подает запрос к таблице коммутации (рисунок 1). Затем на основании полученной информации коммутационная матрица осуществляет дальнейшую обработку и передачу данных на целевой исходный порт.

В обычном коммутаторе Ethernet одновременно реализуются и управление и передача данных. Уровень управления представлен встроенным контроллером, уровень передачи данных – таблицей коммутации и коммутационной матрицей. Контроллер обладает некоторыми интеллектуальными функциями, позволяющим ему самому принимать решения о передаче данных на основе информации о структуре сети. Но непосредственно управлять принятием решения нельзя – можно лишь конфигурировать контроллер, задавая определенные наборы правил и приоритетов. Это значительно ограничивает функциональность коммутатора и всей сети. Например, организацию связей «каждый с каждым» в такой сети нельзя построить без сетевого устройства третьего уровня – маршрутизатора. Проблему разделения уровня управления и передачей данных исследователи Стэнфорда и Беркли предложили решить в рамках подхода, получившего название SDN (Software Defined Networking).

В архитектуре SDN можно выделить три уровня [2]:

- *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.

Стандарт OpenFlow

В основе концепции SDN лежит стремительно развивающийся открытый стандарт OpenFlow — стандарт, определенный общественной организацией Open Networking Foundation (ONF, открытый фонд сетевых технологий) в 2011 году [3]. Данный фонд образован альянсом компаний Deutsche Telekom, Facebook, Google, Microsoft, Verizon, и Yahoo, нацеленный на продвижение и стандартизацию программно-конфигурируемых сетей. О значимости нового движения говорит быстрорастущий состав ONF, в которую вошли уже больше 40 крупнейших компаний мира, в числе которых Brocade, Cisco, Citrix, Oracle, Dell, Ericsson, HP, IBM, Juniper, Marvell, NEC, Netgear, NTT, Riverbed и ряд других.

Интерфейс OpenFlow предоставляет доступ и связь между уровнями управления и инфраструктуры архитектуры SDN как физической, так и виртуальной. OpenFlow использует термин контроллер, который основываясь на общей конфигурации и информации во всей сети, передает по защищенному каналу команды для каждого маршрутизатора (ПКС коммутатора) в отдельности для интерпретации их в индивидуальных таблиц решений. Центральный контроллер имеет точную информацию о структуре и топологии сети. Это позволяет оптимизировать продвижение пакетов данных, и, в частности, прокладывать связи «каждый с каждым» на уровне L2, не прибегая к IP-маршрутизации (рисунок 2).

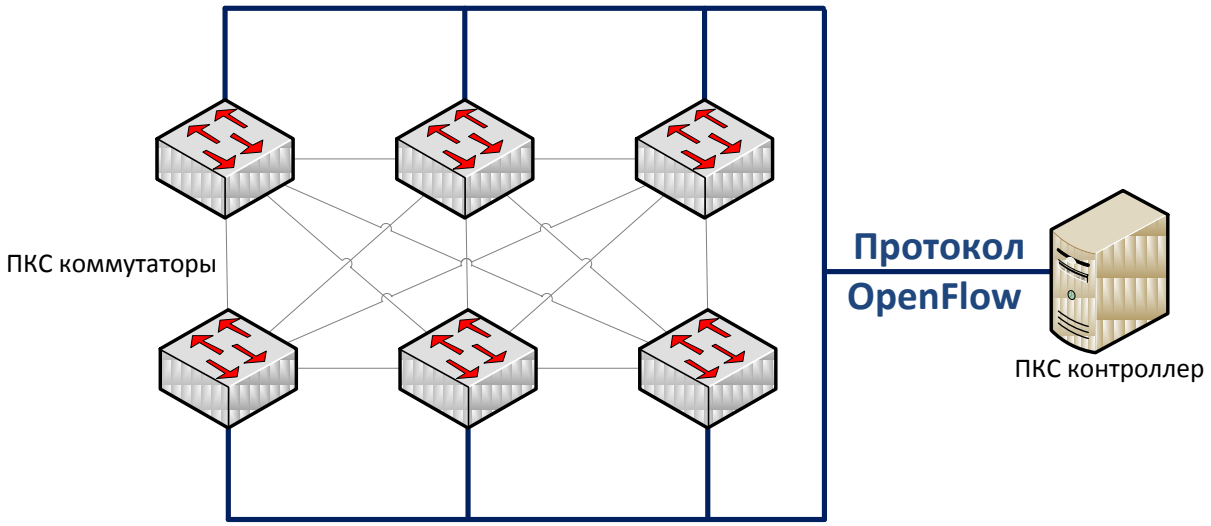

Рисунок 2 — Топология SDN-сети на основе OpenFlow

В ПКС коммутаторе с поддержкой OpenFlow реализован только уровень передачи данных. У каждого коммутатора своя уникальная таблица, которую он заполняет только на основании информации, полученной от центрального контроллера. Такая таблица коммутации получила название FlowTable (таблица потоков), так как по ПКС - сети передаются потоки данных, а не отдельные пакеты (правило в коммутаторе устанавливается только для первого пакета, а затем все остальные пакеты потока его используют). С по-

мощью таких таблиц входящие пакеты классифицируются, основываясь на порту, MAC-адресе, IP-адресе и других средствах [4].

У каждого пришедшего пакета «вырезается» заголовок (битовая строка определенной длины). Для этой битовой строки в таблицах потоков, начиная с первой, ищется правило, у которого поле признаков ближе всего соответствует (совпадает) заголовку пакета. При наличии совпадения, над пакетом и его заголовком выполняются преобразования, определяемые набором инструкций, указанных в найденном правиле. Запись о потоке может предписывать переслать пакет в определенный порт (обычный физический порт либо виртуальный, назначенный коммутатором, или зарезервированный виртуальный порт, установленный спецификацией протокола). Зарезервированные виртуальные порты определяют общие действия пересылки: отправка контроллеру, широковещательная (лавинная) рассылка, пересылка без OpenFlow. Виртуальные порты могут точно определять группы агрегирования каналов, туннели или интерфейсы с обратной связью.

Если нужного правила в первой таблице не обнаружено, то пакет инкапсулируется и отправляется контроллеру, который формирует соответствующее правило для пакетов данного типа и устанавливает его на коммутаторе (или на наборе управляемых им коммутаторов), либо пакет может быть изменен или сброшен. Инструкции конвейера обработки позволяют пересылать пакеты в последующие таблицы для дальнейшей обработки и в виде метаданных передавать информацию между таблицами. Инструкции также определяют правила модификации счетчиков, которые могут быть использованы для сбора разнообразной статистики (рисунок 3).

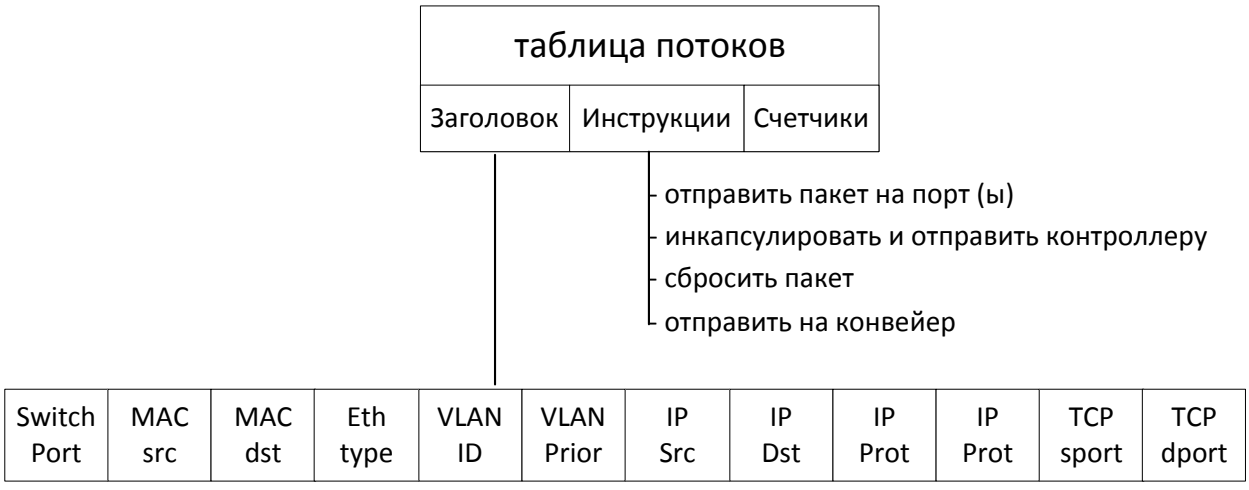

Рисунок 3 – Таблица потоков в OpenFlow – коммутаторе

Традиционные сети могут выбирать маршруты, основываясь на скорости каналов, латентности или резервных мощностях, но при этом им трудно эффективно справляться с любой реальной детализацией (даже используя MPLS TE - Multiprotocol Label Switching Traffic Engineering). OpenFlow, напротив, позволяет запрограммировать сеть для принци-

пиально различных оптимизаций на основе каждого потока. Это означает, что высокочувствительный к задержкам трафик может использовать самый быстрый путь, в то время как объемные потоки могут идти по самым «дешевым» маршрутам. Вместо того чтобы основываться на конкретных конечных устройствах, OpenFlow дает возможность использовать для классификации широкие пределы, вплоть до типа трафика, поступающего от каждой конечной точки.

Протокол OpenFlow не останавливается только на управлении трафиком. Поскольку совместимое с OpenFlow устройство способно также переписывать пакеты, оно может действовать как NAT или AFTR (Address Family Transition Router). Поскольку оно может отбрасывать пакеты, устройство способно действовать как брандмауэр. Оно также может реализовать ECMP (equal-cost multi-path) или другие алгоритмы балансировки нагрузки. Маршрутизаторы могут иметь разные правила для разных потоков, проходящих через них. Они могут распределять нагрузку для одного потока, использовать брандмауэр для другого и изменять заголовки пакетов для третьего потока. Таким образом, отдельные элементы сети становятся значительно более гибкими в условиях OpenFlow по сравнению с традиционной сетью.

OpenFlow предназначен для использования в пределах одной организации. Все маршрутизаторы в домене OpenFlow действуют как единое целое. Контроллер имеет власть над всеми маршрутизаторами, которые он контролирует. Интернет-провайдер может использовать OpenFlow, чтобы контролировать свои маршрутизаторы, но не может контролировать сети клиентов. Предприятие может использовать один OpenFlow домен для управления сетью внутри большого центра обработки данных, а так же другой домен OpenFlow для управления доступа к глобальной компьютерной сети.

Одна из идей, активно развиваемая в рамках SDN, — это виртуализация сетей с целью более эффективного использования сетевых ресурсов. Под виртуализацией сети понимается изоляция сетевого трафика — группирование (мультиплексирование) нескольких потоков данных с различными характеристиками в рамках одной логической сети, которая может разделять единую физическую сеть с другими логическими сетями или сетевыми срезами (network slices). Каждый такой срез может использовать свою адресацию, свои алгоритмы маршрутизации, управления качеством сервисов и т. д. Виртуализация сети позволяет повысить эффективность распределения сетевых ресурсов и сбалансировать нагрузку на них; изолировать потоки разных пользователей и приложений в рамках одной физической сети; администраторам разных срезов использовать свои политики маршрутизации и правила управления потоками данных; проводить эксперименты в сети, используя реальную физическую сетевую инфраструктуру; использовать в каждом срезе только те сервисы, которые необходимы конкретным приложениям.

Одним из примеров виртуализации ресурсов SDN, разделения сети на срезы и управления ими является FlowVisor — программа-посредник (proxy), действующая на уровне между OpenFlow - коммутаторами и различными контроллерами SDN. Посредством FlowVisor можно создавать логические сегменты сети, использующие разные алго-

ритмы управления потоками данных, обеспечивая изоляцию данных сетей друг от друга. Это означает, что каждый контроллер управляет только своей логической сетью и не может оказывать влияния на функционирование других. Для контроллера, взаимодействующего с оборудованием OpenFlow через FlowVisor, весь обмен сообщениями выглядит так же, как если бы контроллер взаимодействовал с обычной сетью SDN. Всю необходимую модификацию сообщений, требующуюся для поддержки различных изолированных сегментов сети, выполняет FlowVisor. То есть для контроллера логической сети не требуется модификации — это может быть любой контроллер SDN, например сетевая операционная система NOX с произвольным набором программ.

Теоретически неограниченные возможности сетей SDN к расширению позволяют строить реальные облака, масштабируемые в зависимости от решаемых задач. При этом сеть обладает требуемой «интеллектуальностью», необходимой, в частности, для оркестровки работы обширных групп коммутаторов.

Заключение

Отметим потенциальные достоинства архитектуры SDN-сетей относительно традиционных современных сетей.

- *Глубокая интеграция.* Каждый веб-сервис может направить требования к пропускной способности к контроллеру, который отвечает за удовлетворение запроса.
- *Уменьшение стоимости развертывания сетей.* Проектирование и изготовление устройства с фиксированной конфигурацией (не требующих обновлений) дешевле.
- *Использование более простых алгоритмов*. Вместо того чтобы принимать решения на основе вывода и опираться на алгоритмы взаимодействия между устройствами, могут быть использованы более непосредственные алгоритмы.
- *Возможность разработки и развития сетевых программных модули*. ПКС - контроллер имеет интерфейсы API, которые могут использоваться приложениями.
- *Глобальная оптимизация и планирование.* Контроллер ссылается на глобальное представление о сети, тем самым, использование сетевых ресурсов может стать более рациональным, а также масштабирование сети становится проще.
- *Открытость протокола OpenFlow*, позволяющая не зависеть от производителей сетевых устройств.
- *Удобство администрирования и отладки.*

Теоретически неограниченные возможности сетей SDN к расширению позволяют строить реальные облака, масштабируемые в зависимости от решаемых задач. При этом сеть обладает требуемой интеллектуальностью, необходимой, для управления работой больших групп коммутаторов.

Список литературы

1. Смелянский Р Программно-конфигурируемые сети//Открытые системы. 2013.URL: http://www.osp.ru/os/2012/09/13032491/ (дата обращения: 01.11.2013).
2. Thomas D. Nadeau, Ken Gray, SDN: Software Defined Networks, O'Reilly, 2013. pp 10-25.
3. OpenFlow Tutorial//OpenFlow.2013.URL. //http://archive.openflow.org/wk/index.php/OpenFlow_Tutorial (Дата обращения: 07.11.2013).
4. ONF Specification//Open network foundation.2013.URL: https://www.opennetworking.org/sdn-resources/onf-specifications (дата обращения: 07.11.2013).
5. Onix: A distributed control platform for large-scale production networks /T. Koponen [и др.], OSDI, 2010. 14 p.