Программно-конфигурируемые сети

Компьютерные сети переживают кризис, преодолеть который взялись технологии Software-Defined Networks. Постепенно завоевывая ИТ-рынок, они докатились и до России. В чем достоинства и возможности этого подхода к построению и эксплуатации сетей?

Руслан Смелянский

Программно-конфигурируемые сети (Software Defined Networks, SDN) — одна из самых «горячих» сегодня технологий, вставших на пути коллапса Сети, однако, несмотря на то что тема еще относительно нова, вокруг нее уже сформировалось несколько полярных мнений: от полного восторга до скепсиса и клише «маркетинговый бантик»[1].

Компьютерные сети как основополагающая инфраструктура — стратегический фактор развития современных ИТ, однако архитектура Сети, основы которой закладывались еще в конце 60-х годов, устарела и уже не всегда способна адекватно и эффективно реагировать на новые потребности. 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. В то же время сегодня эффективность доступного спектра частот для мобильных сетей уже близка к насыщению.

Для того чтобы справиться со значительным ростом трафика, беспроводные сети должны иметь более плотное покрытие, и если сделать соту небольшой, приблизив мобильного клиента к базовой станции, то это увеличит пропускную способность соты и уменьшит количество пользователей в ней. По оценке экспертов, для этого потребуется в 20 раз увеличить плотность размещения базовых станций. Однако современная сетевая архитектура плохо приспособлена для поддержки такого плотного трафика. Во-первых, невозможно равномерно увеличить плотность покрытия — базовые станции придется развертывать везде, где это возможно, то есть хаотично. Во-вторых, такой инфраструктурой будет очень сложно управлять, она будет испытывать неравномерные нагрузки, взаимные влияния сот и действие других факторов. В-третьих, плотная инфраструктура очень дорога в развертывании и эксплуатации.

Развитие микропроцессорной техники и телекоммуникаций привело к тому, что сейчас на каждого человека приходится в среднем около 40 чипов, однако появляются все новые сетевые устройства, 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, их основной функционал реализован аппаратно и закрыт для изменений со стороны владельцев сетей.

Рост количества и разнородности контента, развитие сервисов и масштабов их охвата привели к изменению парадигмы организации вычислений — на место клиент-серверной архитектуры пришли ЦОД и облака, 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. Начиная с 2007 года ежегодные темпы роста пропускной способности сетей во всем мире составляли около 60%, однако исследования специалистов IEEE показывают, что пропускную способность каналов связи требуется увеличивать вдвое раз в два года.

Одновременно с ростом количественных показателей нагрузки на сети усложнились задачи управления сетями — увеличились их перечень, значимость и критичность, причем на фоне повышения требований к безопасности и надежности. Сети строятся на базе устройств, которые постоянно усложняются, поскольку вынуждены поддерживать все больше распределенных

стандартных протоколов (сегодня число активно используемых протоколов и их версий превысило 600), одновременно используя закрытые (проприетарные) интерфейсы. В таких условиях провайдеры не могут оперативно вводить новые сервисы, а производители сетевого оборудования не могут быстро 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.

В 70–80-е годы в СССР велись работы над своими стандартами и средствами построения сетей. Чаще всего они были несовместимы со стандартами, принимавшимися тем, что позднее стало Интернет Сообществом (www.icos.org), в частности, это привело к тому, что в стране не возникло своего производства сетевого оборудования. В результате отечественные телекоммуникационные инфраструктуры сегодня строятся на основе зарубежных средств, а значит, управление инфокоммуникационными сетями в России возможно лишь настолько, насколько это позволяют изделия зарубежных производителей.

Итак, можно выделить следующие проблемы современных компьютерных сетей:

- 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.

Что такое SDN?

В SDN уровни управления сетью и передачи данных разделяются за счет переноса функций управления (маршрутизаторами, коммутаторами и т. п.) в приложения, работающие на отдельном сервере (контроллере). Идея таких сетей была сформулирована специалистами университетов Стэнфорда и Беркли еще в 2006 году [1,2], а инициированные ими исследования нашли поддержку не только в академических кругах, но и были активно восприняты ведущими производителями сетевого оборудования, образовавшими в марте 2011 года консорциум Open Networking Foundation (ONF). Его учредителями выступили Google, Deutsche Telekom, Facebook, Microsoft, Verizon и Yahoo. Состав ONF быстро расширяется, в нее уже вошли такие компании, как Brocade, Citrix, Oracle, Dell, Ericsson, HP, IBM, Marvell, NEC и ряд других. Одной из первых практическую реализацию SDN предложила компания Nicira, вошедшая недавно в состав VMware.

Заинтересованность ИТ-компаний в SDN вызвана тем, что такие технологии позволяют повысить эффективность сетевого оборудования на 25–30%, снизить на 30% затраты на эксплуатацию сетей, превратить управление сетями из искусства в инженерию, повысить безопасность и предоставить пользователям возможность программно создавать новые сервисы и оперативно загружать их в сетевое оборудование.

В части исследований и разработок ключевые заделы в SDN связаны с развиваемой в США программой GENI (Global Environment for Network Innovations) исследования будущего Интернета, объединяющей около 40 ведущих университетов США; деятельностью объединенного центра Стэнфорда и Беркли (Open Network Research Center), выполняющего исследования и разработки в области Internet2; а также с Седьмой рамочной программой исследований Европейского СоюзаOfelia и проектом FEDERICA.

Основные идеи 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.

Архитектура

В архитектуре SDN можно выделить три уровня (рис. 1):

- *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.

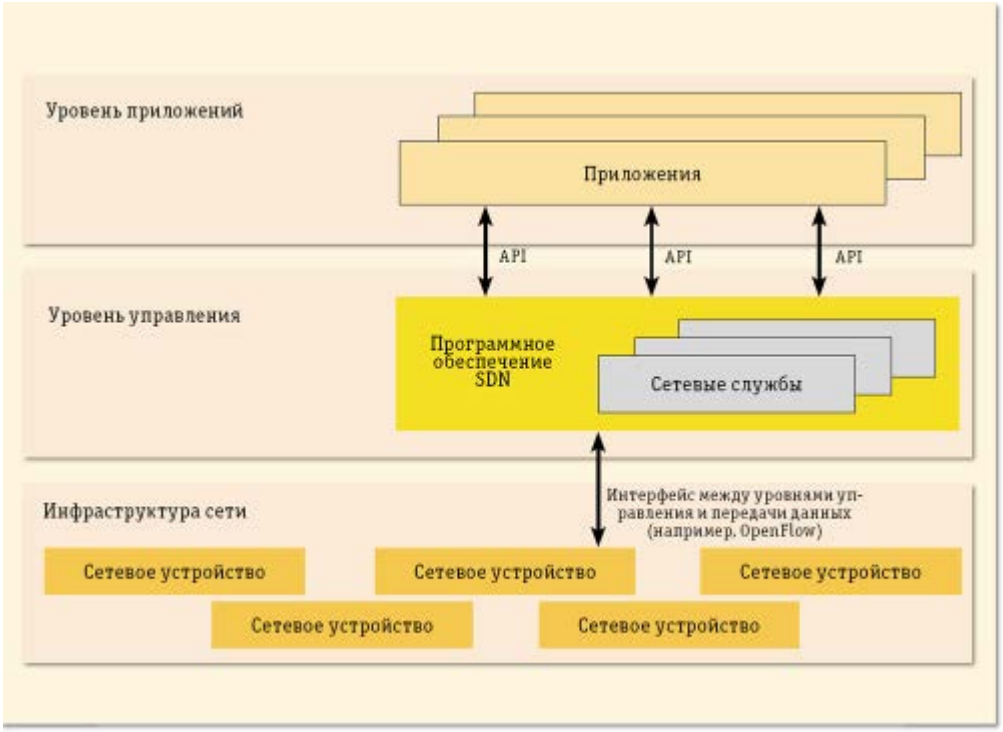

Рис. 1. Архитектура программно-конфигурируемых сетей

Наиболее перспективным и активно развивающимся стандартом для SDN является OpenFlow (OpenFlow версия 1.3) — открытый стандарт, в котором описываются требования, предъявляемые к коммутатору, поддерживающему протокол OpenFlow для удаленного управления.
С помощью современных маршрутизаторов обычно решаются две основные задачи: *передача данных* (forwarding) — продвижение пакета от входного порта на определенный выходной порт и *управление данными* — обработка пакета и принятие решения о том, куда его передавать дальше, на основе текущего состояния маршрутизатора. Это соответствует уровню передачи данных, на котором собраны средства передачи (линии связи, каналообразующее оборудование, маршрутизаторы, коммутаторы), и уровню управления состояниями средств передачи данных (рис. 2). Развитие маршрутизаторов до сих пор шло по пути сближения этих уровней, однако с уклоном на передачу (аппаратное ускорение, совершенствование ПО и внедрение новых функциональных возможностей для увеличения скорости принятия решения по маршрутизации каждого пакета), тогда как уровень управления оставался достаточно примитивным и опирался на сложные распределенные алгоритмы маршрутизации и замысловатые инструкции по конфигурированию и настройке сети. Разумеется, ПО маршрутизаторов, реализующее уровень управления, было проприетарным и закрытым.

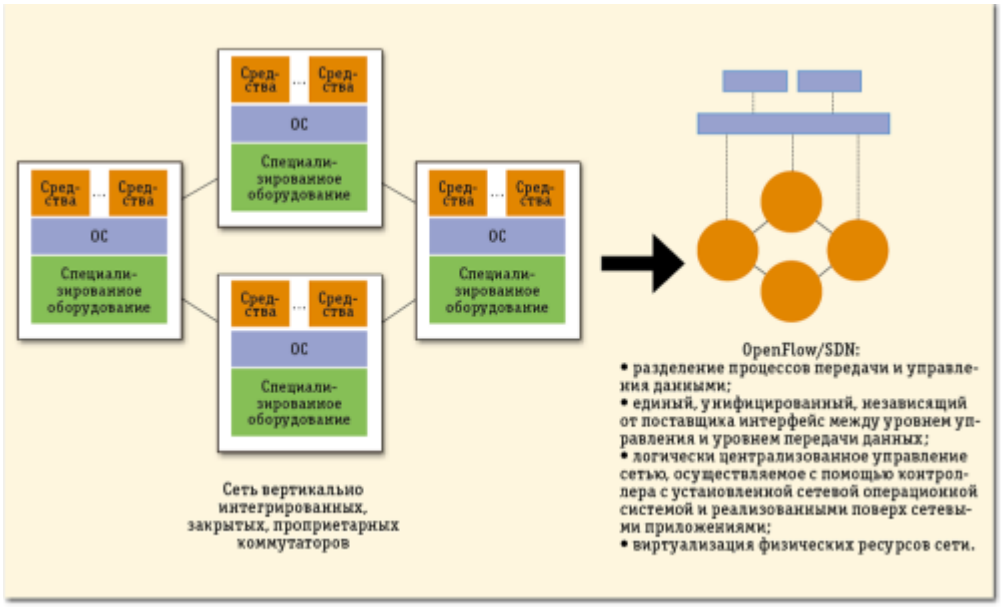

Рис. 2. Традиционные сети и SDN

Согласно спецификации 1.3 стандарта OpenFlow, взаимодействие контроллера с коммутатором осуществляется посредством протокола OpenFlow — каждый коммутатор должен содержать одну или более таблиц потоков (flow tables), групповую таблицу (group table) и поддерживать канал (OpenFlow channel) для связи с удаленным контроллером — сервером. Спецификация не регламентирует архитектуру контроллера и API для его приложений. Каждая таблица потоков в коммутаторе содержит набор записей (flow entries) о потоках или правила. Каждая такая запись состоит из полей-признаков (match fields), счетчиков (counters) и набора инструкций (instructions).

Механизм работы коммутатора OpenFlow достаточно прост. 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. Инструкции, ассоциированные с каждой записью таблицы, описывают действия, связанные с пересылкой пакета, модификацией его заголовка, обработкой в таблице групп, обработкой в конвейере и пересылкой пакета на определенный порт коммутатора. 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.

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 сброшен (в зависимости от конфигурации коммутатора).

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 могут определять общие действия пересылки: отправка контроллеру, широковещательная (лавинная) рассылка, пересылка без OpenFlow. Виртуальные порты, определенные коммутатором, могут точно определять группы агрегирования каналов, туннели или интерфейсы с обратной связью.

Записи о потоках могут также указывать на группы, в которых определяется дополнительная обработка. Группы представляют собой наборы действий для широковещательной рассылки, а также наборы действий пересылки с более сложной семантикой, например быстрое изменение маршрута или агрегирование каналов. Механизм групп позволяет эффективно изменять общие выходные действия для потоков. Таблица групп содержит записи о группах, содержащие список контейнеров действий со специальной семантикой, зависящей от типа группы. Действия в одном или нескольких контейнерах действий применяются к пакетам, отправляемым в группу.

Разработчики коммутаторов могут быть свободны в реализации их внутренней начинки, однако процедура просмотра пакетов и семантика инструкций должны быть для всех одинаковы. Например, в то время как поток может использовать все группы для пересылки в некоторое множество портов, разработчик коммутатора может выбрать для реализации этого единую битовую маску внутри аппаратной таблицы маршрутизации. Другой пример — это процедура просмотра таблиц: конвейер физически может быть реализован с помощью различного количества аппаратных таблиц. Установка, обновление и удаление правил в таблицах потоков коммутатора осуществляются контроллером. Правила могут устанавливаться реактивно (в ответ на пришедшие пакеты) или проактивно (заранее, до прихода пакетов).
Управление данными в OpenFlow осуществляется не на уровне отдельных пакетов, а на уровне их потоков. Правило в коммутаторе OpenFlow устанавливается с участием контроллера только для первого пакета, а затем все остальные пакеты потока его используют.
Имеющиеся на сегодняшний день физические коммутаторы SDN соответствуют пока спецификации OpenFlow 1.0 и содержат только одну таблицу потоков.

Протокол OpenFlow

Идея SDN о создании унифицированного, независимого от производителя сетевого оборудования, программно-управляемого интерфейса между контроллером и транспортной средой сети нашла отражение в протоколе OpenFlow, позволяющем пользователям самим определять и контролировать, кто с кем, при каких условиях и с каким качеством может взаимодействовать в Сети. Протокол поддерживает три типа сообщений: контроллер-коммутатор, асинхронные и симметричные.
Сообщения типа контроллер-коммутатор инициируются контроллером и используются для непосредственного управления и слежения за состоянием коммутатора. Сообщения данного типа могут использоваться контроллером для установки параметров конфигурации коммутатора, для сбора статистики, для добавления, удаления и модификации записей в таблицах потоков.
Асинхронные сообщения инициируются коммутатором для оповещения контроллера о сетевых событиях (прибытии пакетов или удалении записи из таблицы по тайм-ауту) и изменениях состояния коммутатора или ошибках.
Симметричные сообщения могут инициироваться коммутатором или контроллером без запроса и используются при установлении соединения, а также при измерении задержек, пропускной способности соединения контроллер-коммутатор или для проверки живучести соединения.

Сетевая ОС

Логически-централизованное управление данными в сети предполагает вынесение всех функций управления сетью на отдельный физический сервер, называемый контроллером, который находится в ведении администратора сети. Контроллер может управлять как одним, так и несколькими OpenFlow-коммутаторами и содержит сетевую операционную систему, предоставляющую сетевые сервисы по низкоуровневому управлению сетью, сегментами сети и состоянием сетевых элементов, а также приложения, осуществляющие высокоуровневое управление сетью и потоками данных.
Сетевая ОС (СОС) обеспечивает приложениям доступ к управлению сетью и постоянно отслеживает конфигурацию средств сети. В отличие от традиционного толкования термина ОС, под СОС понимается программная система, обеспечивающая мониторинг, доступ и управление ресурсами всей сети, а не ее конкретного узла.
Подобно традиционной операционной системе, СОС обеспечивает программный интерфейс для приложений управления сетью и реализует механизмы управления таблицами коммутаторов: добавление, удаление, модификацию правил и сбор разнообразной статистики. Таким образом, фактически решение задач управления сетью выполняется с помощью приложений, реализованных на основе API сетевой операционной системы, позволяющих создавать приложения в терминах высокоуровневых абстракций (например, имя пользователя и имя хоста), а не низкоуровневых параметров конфигурации (например, IP- и MAC-адресов). Это позволяет выполнять управляющие команды независимо от базовой топологии сети, однако требует, чтобы СОС поддерживала отображения между высокоуровневыми абстракциями и низкоуровневыми конфигурациями.
В каждом контроллере имеется хотя бы одно приложение, которое управляет коммутаторами, соединенными с этим контроллером, и формирует представление о топологии физической сети, находящейся под управлением контроллера, тем самым централизуя управление. Пред-

ставление топологии сети включает в себя топологию коммутаторов, расположение пользователей и хостов и других элементов и сервисов сети. Представление также включает в себя привязку между именами и адресами, поэтому одной из важнейших задач, решаемых СОС, является постоянный мониторинг сети. Таким образом, СОС позволяет создавать приложения в виде централизованных программ, использующих высокоуровневые имена, на основе таких алгоритмов, как, например, алгоритм Дейкстры поиска кратчайшего пути в графе, вместо сложных распределенных алгоритмов вроде алгоритма Беллмана – Форда, в терминах низкоуровневых адресов, которые используются в современных маршрутизаторах.
На данный момент имеется 28 реализаций сетевых ОС для программно-определяемых сетей: NOX, POX, Beacon, Maestro, Trema, BigSwitch, FloodLight и др.
Для контроллеров в SDN очень важным является требование того, что все приложения одного контроллера в каждый момент времени должны иметь одинаковое представление о топологии сети. Однако переход от распределенного управления сетью к централизованному имеет и ряд недостатков. Например, снижение надежности, отказоустойчивости, масштабируемости.
Сегодня получили развитие несколько подходов к построению распределенного масштабируемого контроллера: HyperFlow [3], Onix [4] и Kandoo [5]. Однако, согласно результатам исследований ЦПИ КС, наиболее перспективен альтернативный подход (рис. 3). Поскольку каждый контроллер может быть соединен с несколькими коммутаторами, а каждый коммутатор — с несколькими контроллерами, то контроллеры, управляющие одним и тем же коммутатором, можно объединить в групповой контроллер (ГК). Все контроллеры одного и того же ГК должны иметь согласованное представление о топологии той части сети, к которой они обеспечивают доступ. Как видно из рис. 3, C1 – C3 — контроллеры, S1 – S4 — коммутаторы, а V1 – V3 — фрагменты сети, к которым обеспечивает доступ коммутатор S1, S2, S3 соответственно. Тогда ГК1 образуют контроллеры C1 и C2, ГК2 — C2 и C3, а все приложения в ГК1 должны иметь согласованное представление о топологии V1 и V2, все приложения в ГК2 — о топологии V2 и V3. В случае выхода из строя, например, контроллера C1 его может заменить C2, взяв на себя управление V1. Представление о состоянии соответствующей части сети контроллеры могут согласовывать либо через коммутатор S4, либо через S1, S2 и S3.

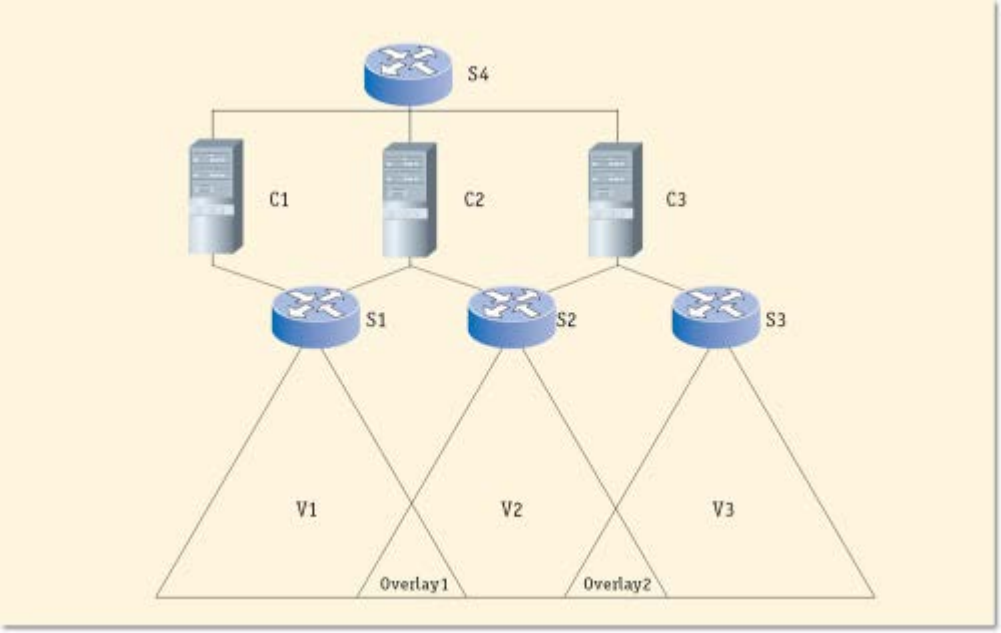

Рис. 3. Альтернативный подход к построению распределенного масштабируемого контроллера

Такой подход к построению распределенного контроллера решает проблему масштабируемости и повышает отказоустойчивость SDN.

Виртуализация в SDN

Одна из идей, активно развиваемая в рамках SDN, — это виртуализация сетей с целью более эффективного использования сетевых ресурсов (рис. 4). 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.

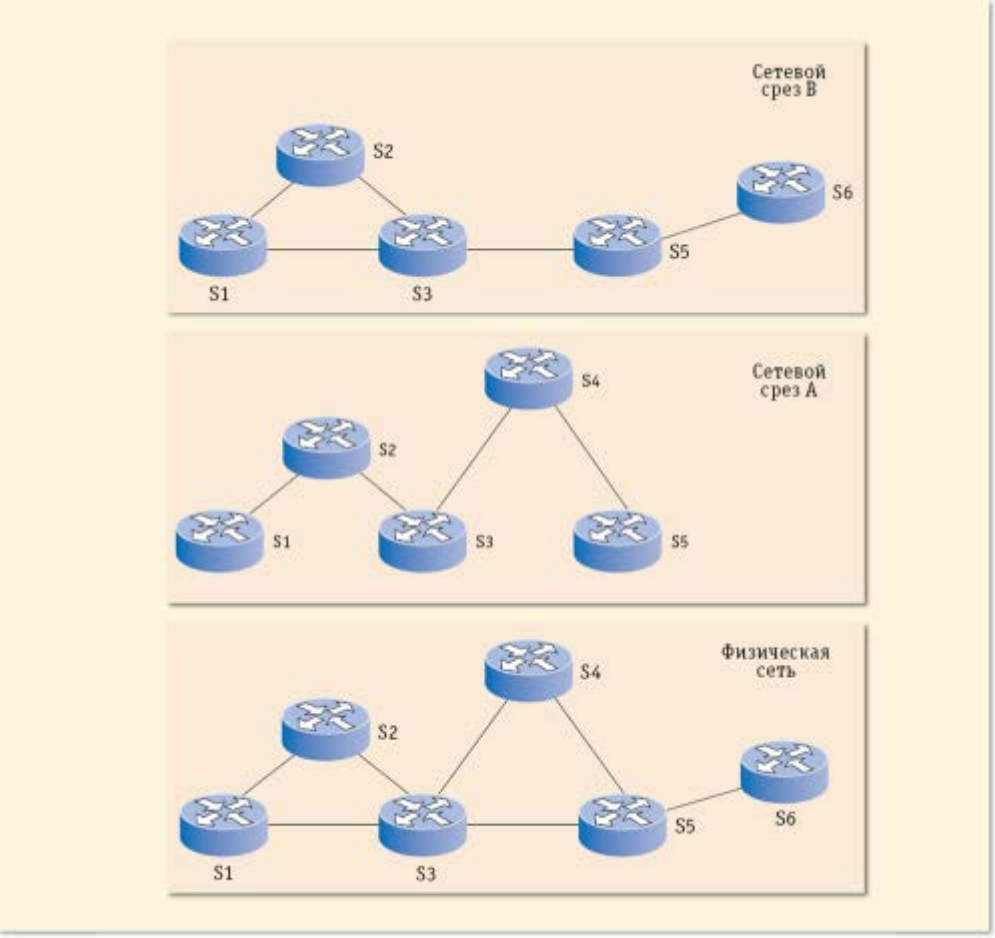

Рис. 4. Виртуализация в SDN

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 [6] — программа-посредник (proxy), действующая на уровне между OpenFlow- коммутаторами и различными контроллерами SDN. Посредством FlowVisor можно создавать логические сегменты сети, использующие разные алгоритмы 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.

Преимущества SDN

Благодаря снятию с коммутаторов нагрузки по обработке тракта управления, SDN позволяет этим устройствам направить все свои ресурсы на ускорение перемещения трафика, что существенно повышает производительность. При этом за счет виртуализации управления сетью снижаются расходы на их построение и сопровождение. По результатам тестов, проведенных на базе крупнейших провайдеров США, использование SDN позволяет на 20–30% увеличить утилизацию ресурсов ЦОД и в несколько раз снизить эксплуатационные расходы.

Программные средства SDN позволяют администраторам добавлять новую функциональность к уже имеющейся сетевой архитектуре. При этом новые функции будут работать на многих платформах — их не придется реализовывать заново во встроенном программном обеспечении коммутаторов каждого поставщика.
На централизованном контроллере SDN системный администратор может наблюдать всю сеть в едином представлении, за счет чего повышаются удобство управления, безопасность и упрощается выполнение ряда других задач. Действительно, поскольку администратор видит все потоки трафика, то ему легче замечать вторжения, назначать приоритеты различным типам трафика и разрабатывать правила реагирования сети при заторах и проблемах с оборудованием.
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.

Перспективы SDN

Сегодня на рынке решений SDN заметны две тенденции: наблюдается активное появление перспективных стартапов и происходит ориентация лидеров рынка ИКТ на SDN, что выражается в открытии собственных научно-исследовательских подразделений, работающих по данной тематике, и отдельных линеек продуктов, основанных на новом подходе.
Первый коммерческий проект в области SDN выполнила в 2007 году компания Nicira, основанная Ником Маккеоном, Мартином Касадой и Скоттом Шенкером. В Nicira разработали собственную *платформу виртуализации сетей* (Network Virtualization Platform, NVP), которой очень быстро заинтересовались клиенты AT&T и NTT, а затем и такие компании, как eBay, DreamHost, Fidelity Investments и Rackspace. В результате нескольких раундов инвестиций, в июле 2012 года компания была куплена VMware, что положило начало формированию рынка решений SDN.
Другая компания — BigSwitch — была основана профессором Стэнфордского университета Гуидо Аппенцеллером и бывшим сотрудником Cisco Кайлом Форстером, причем в рамках первого раунда инвестиций в компанию вложил свои средства венчурный фонд Khosla Ventures, образованный Винодом Кошла, сооснователем Sun Microsystems. Аналитики полагают, что компания BigSwitch, так же, как и Nicira, в скором времени получит предложение о покупке от лидеров ИТ-рынка. Кроме того, в конце июля 2012 года было объявлено о том, что Oracle достигла соглашения о покупке компании Xsigo Systems, занимающейся разработкой ПО для SDN.
Говоря о второй тенденции, необходимо отметить, что ряд производителей уже имеют готовые к продаже собственные решения в области SDN. Например, Cisco Systems, помимо запуска линейки коммутаторов Nexus и Catalyst 35XX, способных работать в традиционных сетях и в SDN, анонсировала платформу Open Network Environment (ONE), специально предназначенную для поддержки решений SDN. Кроме этого, компания объявила о разработке пилотной версии программного обеспечения для контроллеров, а также пилотной версии агента OpenFlow для сбора сведений о работе сетевых инфраструктур SDN. Компания Juniper Networks добавила опцию OpenFlow в операционную состему JunOS SDK, а в июне объявила о реализации этой технологии в линейке коммутаторов серий EX и MX. Компании NEC, Pronto и Marvell предлагают коммутаторы, реализующие только протокол OpenFlow, а IBM выпустила контроллер IBM System Networking Programmable Network Controller как программное приложение на Linux-платформе на основе OpenFlow. В HP реализуется стратегия HP Virtual Application Networks, предусматривающая выпуск контроллера, приложения, а также услуг и решения на основе SDN, а Brocade представила первые продукты с поддержкой SDN, в частности коммутатор Brocade VDX 8770. К гонке присоединилась и компания Intel, продемонстрировавшая на IDF свое решение для коммутатора SDN и ПО с поддержкой протокола OpenFlow на базе Linux.
В апреле 2012 года Урс Хольце, старший вице-президент по технической инфраструктуре Google, заявил, что компания перевела всю внутреннюю сеть G-Scale для обмена трафиком между ЦОД Google по всему миру на SDN, самостоятельно изготовив коммутаторы OpenFlow, поскольку существующие аналоги на рынке были в тот момент для компании недоступны. Коммутаторы Google OpenFlow способны масштабироваться до сотен неблокирующихся портов 10-Gigabit Ethernet. Для Google использование SDN позволило выбирать оборудование, строго соответствующее необходимому ПО; осуществлять централизованное управление сетью и потоками данных; оптимизировать процессы тестирования и мониторинга.
Вместе с тем говорить о формировании полноценного рынка решений SDN пока еще преждевременно, однако, по оценкам аналитиков, к 2017 году он сформируется и его объем может

вырасти до 2,1 млрд долл. против 198 млн долл. в 2012 году. Основными движущими силами этого рынка названы такие факторы, как растущая потребность в мобильности, потребность в новой сетевой архитектуре при переходе на облачные услуги и использовании разного вида трафика. Главными локомотивами рынка SDN будут пока телекоммуникационные компании, которым эта технология дает гибкость в предоставлении новых услуг и достижении требуемой производительности.

OpenFlow и SDN в России

Согласно прогнозам экспертов, объем мирового рынка сетевого оборудования к 2015 году превысит 184 млрд долл. при ежегодном росте в 10%. Российский рынок сетевого оборудования оценивается в 3–4 млрд долл., при этом на продукцию зарубежных производителей приходится свыше 90%, поэтому задачу импортозамещения можно рассматривать как критически важную. В этом смысле успех проектов и решений SDN может позволить стране стать полноправным партнером лидеров данного сегмента ИТ-рынка. Однако для этого требуется: создать ассоциации научно-исследовательских университетов, лабораторий и профильных академических институтов, занимающихся тематикой сетевых технологий; привлечь в Россию ведущих зарубежных ученых и инженеров-исследователей, работающих в области SDN; интегрировать российских ученых в международные проекты, связанные с SDN.

В феврале 2012 года на базе лаборатории вычислительных комплексов факультета ВМК МГУ был создан Центр прикладных исследований компьютерных сетей (ЦПИ КС, резидент ИТ-кластера Фонда «Сколково»), в задачи которого входит проведение научных исследований в области сетевых технологий, в том числе SDN. В июле того же года ОАО «Ростелеком» заключило контракт с ЦПИ КС на проектирование и создание опытного сегмента облачной платформы для ЦОД на основе SDN. Результатом работ станет создание тестового сегмента облачной платформы на базе ЦОД «Ростелекома», разработка прототипа системы управления сетевой инфраструктурой ЦОД, системы показателей эффективности управления новой сетевой инфраструктурой ЦОД, а также методики встраивания в существующую инфраструктуру ЦОД.

Андрей Аксенов, директор департамента исследований и разработок ОАО «Ростелеком», отмечает: «Мы решили заранее предусмотреть ситуацию и не дожидаться, пока SDN сама всех настигнет. Мы прежде всего хотим сократить затраты на телекоммуникационное оборудование. У нас достаточно амбициозные планы — трансформация компании в сервис-провайдера, а также построение облачной платформы на распределенной по всей стране сети ЦОД, создание новых сервисов. В этом смысле SDN очень перспективна».

В рамках данного проекта «Ростелеком» выбирает приложение, функционирующее внутри ЦОД, а ЦПИ КС обеспечивает сетевое взаимодействие этого приложения с помощью SDN. Таким образом, становится возможной гибкая, динамическая настройка используемого сетевого оборудования, позволяющая избежать ограничения на количество создаваемых виртуальных ЦОД (сейчас их может быть не больше 4096). Нынешние ЦОД «Ростелекома» работают на связке традиционных протоколов TCP/IP с серверами виртуализации, и в рамках реализации проекта часть ресурсов будет отдана для тестирования технологий SDN. Традиционная и новая архитектуры будут работать параллельно, не взаимоисключая друг друга. Критериями для сравнения двух архитектур станут загрузка и задержки каналов связи, задержки сетевого трафика и число одновременно поддерживаемых виртуальных ЦОД. Кроме того, новая архитектура должна уменьшить накладные расходы на перенастройку сетевого оборудования — ее просто не потребуется.

ЦПИ КС выполняет проект совместно с компанией «Русьтелетех», разрабатывающей телекоммуникационное оборудование. Эта компания предложит дизайн типового сегмента ЦОД облачной системы, включающий в себя средства виртуализации вычислительных ресурсов и систем хранения данных. Для оптимизации использования аппаратных и виртуальных серверов сетевая инфраструктура типового сегмента строится на базе оборудования, поддерживающего технологии SDN. Например, в качестве коммутатора применяется изделие серии РТТ-А410 с плотностью портов 48x10G и производительностью 960 Гбит/с, функционирующее как в стандартном режиме коммутации и маршрутизации данных, так и в режиме OpenFlow. Это позволяет обеспечить «безболезненное» разворачивание программно-конфигурируемых сетей на базе традиционных сетевых архитектур. Для централизованного управления системой стоечных решений РТТ-А410, работающих в режиме поддержки протокола OpenFlow, они объединяются через коммутатор уровня Layer 2+ серии РТТ-А311 и подключаются к контроллеру сети.

Проект тестовой опытно-конструкторской зоны облачной платформы на базе SDN завершится в декабре 2012 года, и впоследствии будет создана зона на площадке «Ростелекома» с использованием коммутаторов «Русьтелетех», сертифицированных ФСТЭК на соответствие третьему уровню контроля отсутствия НДВ и четвертому классу защищенности МЭ. По словам вице-президента по инновационному развитию компании «Ростелеком» Алексея Нащекина, технология SDN позволит более эффективно решать задачи виртуализации сетей и разработки новых комплексных сервисов для клиентов «Ростелекома», даст возможность в ближайшем будущем сократить капитальные и операционные затраты на телекоммуникационное оборудование облачных ЦОД компании.

***

Программно-конфигурируемые сети открывают большие возможности для промышленности и бизнеса, позволяя решать задачи повышения пропускной способности каналов, упрощения управления сетью, перераспределения нагрузки, повышения масштабируемости сети. Каждая компания, в зависимости от своих потребностей, может внедрить решение, соответствующее конкретно ее задачам. Данной технологией могут заинтересоваться хостеры и провайдеры, владельцы дата-центров и операторы связи, финансовые и банковские структуры, телекоммуникационные компании, которым внедрение SDN позволит повысить эффективность их работы. Программно-конфигурируемые сети — одна из самых «горячих» сегодня технологий в мире компьютерных сетей, однако пока это лишь айсберг, у которого видна только верхушка возможностей, ведь многие технологии все еще не стандартизированы. А это значит, что и Россия может повлиять на то, как завтра будут работать компьютерные сети.

Литература

1. Martin Casado, Tal Garfinkel, Aditya Akella, Michael J. Freedman Dan Boneh, Nick McKeown, Scott Shenker. SANE: A Protection Architecture for Enterprise Networks, 15-th Usenix Security Symposium, Vancouver, Canada, August 2006.
2. N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, J. Turner. Openflow: Enabling innovation in campus networks. SIGCOMM Computer Communication Review, vol. 38, no. 2, pp. 69–74, 2008.
3. A. Tootoonchian, Y. Ganjali. HyperFlow: A Distributed Control Plane for OpenFlow. In Proc. INM/WREN, San Jose, CA, April 2010.
4. T. Koponen, M. Casado, N. Gude, J. Stribling, L. Poutievski, M. Zhu, R. Ramanathan, Y. Iwata, H. Inoue, T. Hama, S. Shenker. Onix: A distributed control platform for large-scale production networks, in OSDI, Oct 2010.
5. S. Hassas Yeganeh, Y. Ganjali. Kandoo: a framework for efficient and scalable offloading of control applications. Proceedings of the first workshop on Hot topics in software defined networks (HotSDN '12), pp. 19–24.
6. Rob Sherwood, Glen Gibby, Kok-Kiong Yapy, Guido Appenzellery, Martin Casado, Nick McKeowny, Guru Parulkary. FlowVisor: A Network Virtualization Layer, 2009.

Руслан Смелянский *(rsmeliansky@arccn.ru) — профессор МГУ им. М. В. Ломоносова, директор по науке и образованию Центра прикладных исследований компьютерных сетей (Москва).*

[1] В отечественной литературе также получил распространение термин «программно-определяемые сети». — *Прим. ред.*