

УДК 004.732

АВТОМАТИЗАЦИЯ И МАСШТАБИРОВАНИЕ ЛОКАЛЬНЫХ ВЫЧИСЛИТЕЛЬНЫХ СЕТЕЙ С ИСПОЛЬЗОВАНИЕМ SDN

С.Ю. ПАРХОМИК, А.Д. СЕМАК, И.Л. СЕЛЕЗНЕВ

Белорусский государственный университет информатики и радиоэлектроники
П. Бровка, 6, Минск, 220013, Беларусь

Поступила в редакцию 10 ноября 2017

В данной статье рассмотрена основная проблема традиционного подхода к построению сетей передачи данных. Проанализированы принцип организации SDN сети и ее основные компоненты. Проведено сравнение подхода с использованием SDN сетей и традиционного подхода.

Ключевые слова: сеть передачи данных, программно-конфигурируемая сеть, информационный поток, контроллер.

Введение

В современном мире бизнес в сфере информационных технологий предъявляет большие требования к гибкости и масштабируемости компьютерных сетей. С ростом объемов сетевого трафика и количества подключенных к сети устройств конфигурирование крупномасштабных сетей превращается в сложную задачу. В традиционных коммутаторах и маршрутизаторах процессы передачи трафика и управления им неотделимы друг от друга и реализованы в одной «коробке»: специальные микросхемы обеспечивают пересылку пакетов с одного порта на другой, а вышележащее программное обеспечение (ПО) определяет правила такой пересылки, выполняет необходимый анализ пакетов, производит изменение содержащейся в них служебной информации (рис. 1).

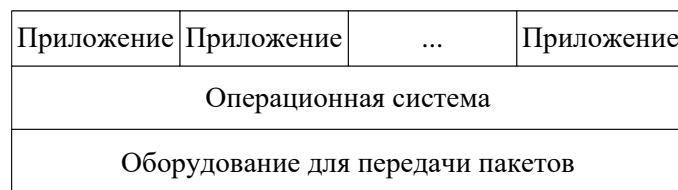


Рис. 1. Архитектура типичного коммутатора или маршрутизатора

Главная проблема данного подхода заключается в том, что основанная на нем сеть статична и потому не соответствуют динамике, свойственной современной сетевой инфраструктуре. Наиболее очевидным решением данной проблемы является использование виртуализации: инфраструктура должна работать как обычная физическая сеть, но выбор модели и месторасположение оборудования становятся менее критичным. Соответственно, настраивая виртуальную сеть, можно не знать ее состава и архитектуры. В таком случае на одном комплекте оборудования может быть развернуто несколько независимых и никак не пересекающихся сетей. Одной из наиболее перспективных разработок в данной области считается технология Software-Defined Network (SDN) или программно-конфигурируемая сеть (ПКС).

Основные компоненты SDN и их взаимодействие

Суть концепции SDN заключается в разделении функций управления и пересылки данных, передаче функций маршрутизации контроллеру сети и реализации на основе этого принципа легко масштабируемой, быстро и гибко настраиваемой виртуальной сети. Каждое SDN приложение фактически представляет собой интерфейс оптимизации сети под конкретные цели. Основная его роль – изменение сети в реальном времени под текущие требования обслуживаемой программы [1].

В информационных потоках архитектуры SDN можно выделить два основных направления обмена: первый («северный мост») – между SDN приложениями, второй («южный мост») – для управления физическими сетевыми устройствами (рис. 2).

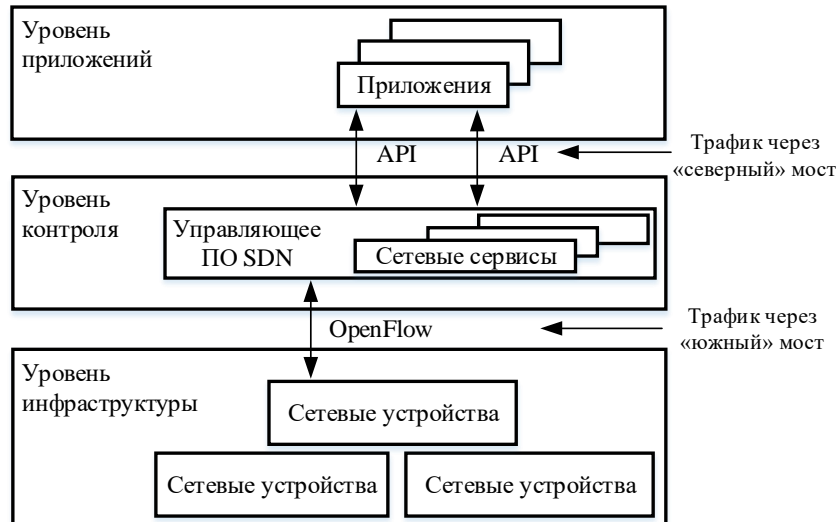


Рис. 2. Архитектура технологии SDN

Данный подход позволяет решить проблему разделения уровня управления и передачи данных, которая значительно ограничивает функциональность коммутатора и всей сети.

Организация обмена сообщениями протоколом OpenFlow

Идея OpenFlow основана на том, что все коммутаторы и маршрутизаторы содержат таблицу потоков, определяющую базовую функцию передачи данных – для каждого входящего пакета определить определенный исходящий интерфейс. Формат этих таблиц различен, но можно идентифицировать стандартный набор функций.

1. Каждая запись абстрактной таблицы передачи OpenFlow является «правилом» и связана с так называемым «поток» данных (flow). Поток определяется заголовком пакета – например, комбинацией адресов MAC, IP и номеров портов источника и получателя данных.

2. Другим элементом записи таблицы является «действие» (action), определяющее требуемую обработку пакетов потока. Различают четыре вида действий:

- передать пакет на определенный порт (порты) коммутатора;
- передать пакет контроллеру через «защищенный» канал: как правило, первый пакет неизвестного потока отправляется контроллеру для определения правила и создания новой записи таблицы передачи;
- отбросить пакет: это действие может быть использовано в борьбе с сетевыми атаками;
- пакет может быть направлен на «стандартную» обработку, это позволяет разделить потоки данных на потоки, управляемые OpenFlow, и потоки, управляемые другими механизмами.

3. Последний элемент содержит статистику – продолжительность потока, число полученных и переданных пакетов и другую дополнительную информацию [2].

Совокупность этих элементов образует правило, структура которого представлена на рис. 3.

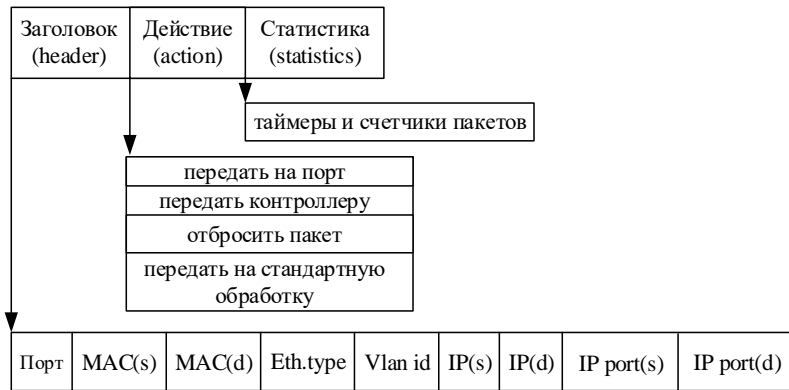


Рис. 3. Упрощенная структура таблицы передачи OpenFlow

Все правила формируются на центральном контроллере и затем передаются в таблицу потоков коммутатора (рис. 4).

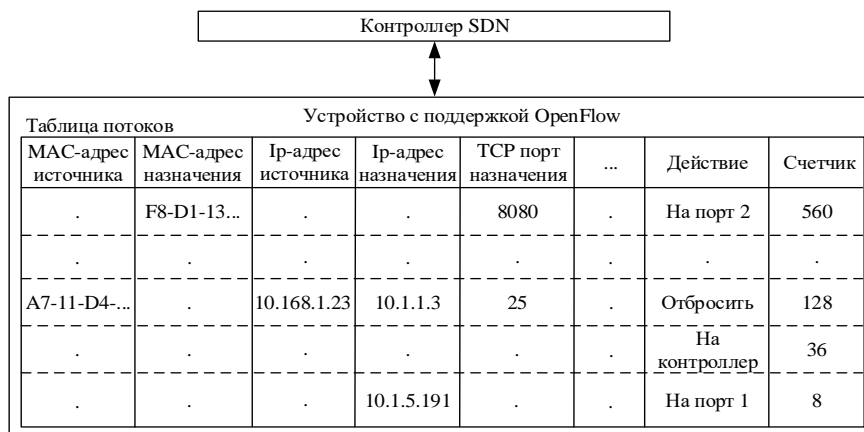


Рис. 4. Таблица потоков OpenFlow

На основании этой таблицы коммутатор определяет, что необходимо сделать с поступившим пакетом. Алгоритм поиска правила прост: у каждого пришедшего пакета «вырезается» заголовок (битовая строка определенной длины), и для него ищется правило в таблице потоков, у которого поле признаков ближе всего соответствует заголовку пакета. Если нужного правила в таблице не обнаружено, то пакет инкапсулируется и отправляется контроллеру, который формирует соответствующее правило для пакетов данного типа и устанавливает его на коммутаторе/коммутаторах. Также пакет может быть изменен или отброшен. Вместо того, чтобы основываться на конкретных конечных устройствах, OpenFlow дает возможность использовать более широкие пределы для классификации, вплоть до типа трафика, поступающего от каждой конечной точки [3].

Данный протокол не ограничивается только управлением трафиком. Совместимые с ним устройства могут выполнять и другие операции: переписывать либо отбрасывать пакеты, реализовывать балансировку нагрузки, выступать в качестве межсетевого экрана. Таким образом, отдельные элементы сети, использующие OpenFlow, становятся значительно более гибкими по сравнению с традиционной сетью.

SDN и традиционная архитектура сети передачи данных

Сеть передачи данных (СПД) – совокупность трех и более оконечных устройств (терминалов) связи, объединенных каналами передачи данных и коммутирующими устройствами (узлами сети), обеспечивающими обмен сообщениями между всеми оконечными устройствами. К основным параметрам сети передачи данных относятся следующие:

1. Время реакции: определяет время между возникновением запроса и получением ответа на этот запрос (рис. 5).

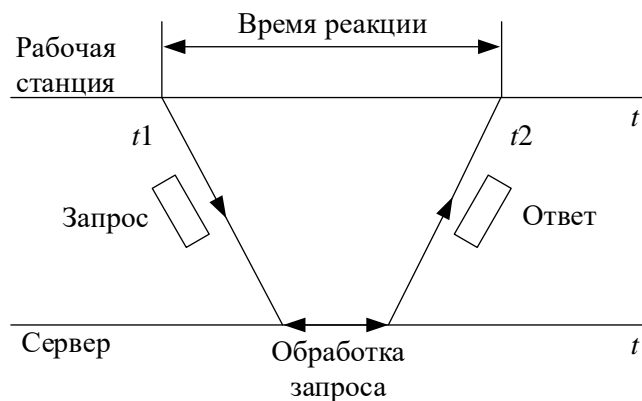


Рис. 5. Время реакции обработки запроса

2. Полоса пропускания – мера емкости соединения, часто определяемая как скорость передачи данных по каналу связи.

3. Отказоустойчивость – способность сети скрыть от пользователя отказ отдельных элементов: определяется количеством любых последовательных единичных отказов компонентов, после которого сохраняется работоспособность системы в целом.

4. Задержка передачи – задержка между моментом поступления данных на вход сетевого устройства и появления их на выходе, характеризует сетевые этапы обработки данных, не учитывая задержки обработки конечными узлами сети.

Эти параметры необходимо учитывать при построении сети, т.к. они оказывают непосредственное влияние на качество передачи данных.

Основные подходы построения сетей передачи данных представлены на рис. 6.

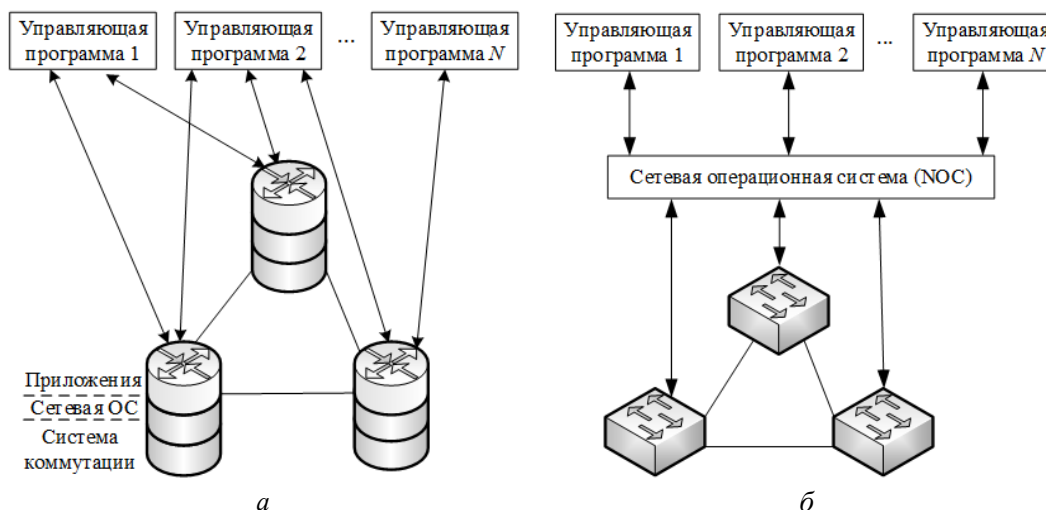


Рис. 6. Архитектура сети: *а* – традиционная архитектура с автономными сетевыми элементами; *б* – архитектура SDN с централизованной управляющей функцией

В традиционных коммутаторах и маршрутизаторах функции передачи трафика и управления реализованы на одном устройстве и неотделимы друг от друга (рис. 6, *а*). Такие сети обычно строятся с использованием протоколов динамической маршрутизации, где наиболее распространенным является протокол Open Shortest Path First (OSPF), который основан на технологии отслеживания состояния каналов. Отличительной особенностью подобных протоколов является быстрое реагирование на изменения сети. Кроме того, в отличие от дистанционно-векторных протоколов динамической маршрутизации, рассылка обновлений происходит либо в случае появления изменений, либо каждые 30 мин. В случае изменения состояния канала, устройство, обнаружившее такое изменение, формирует извещение о состоянии канала для соседних устройств [4].

В основе OSPF лежит алгоритм поиска кратчайшего пути – Short Path First (SPF), который также называют алгоритмом Дейкстры. Выбор оптимального маршрута происходит на основании метрики, рассчитываемой по формуле

$$\text{Стоимость маршрута} = \frac{\text{Эталонная полоса пропускания}}{\text{Полоса пропускания интерфейса}},$$

где Эталонная полоса пропускания – это пропускная способность, относительно которой высчитывается стоимость интерфейса: по умолчанию ее значение 100 Мб.

Например, если пропускная способность интерфейса 10 Мб, то значение его стоимости будет получено по формуле

$$\text{Стоимость маршрута} = \frac{100 \text{ Мб}}{10 \text{ Мб}} = 10.$$

Таким образом, чем выше пропускная способность интерфейса, тем ниже стоимость маршрута.

Достоинства протокола OSPF следующие.

1. Отсутствие ограничения на размер сети.
2. Разделение системы на области маршрутизации. Изменения топологии сети коснутся только той области, в которой они произошли, все остальные области не будут затронуты при перерасчете маршрутов.
3. Высокая скорость построения маршрута. Данный параметр выше, чем у дистанционно-векторных протоколов динамической маршрутизации.
4. Использование аутентификации. При установке соседства маршрутизаторов можно установить аутентификацию устройств.

К основным недостаткам OSPF относится:

1. Сложность настройки.
2. Дороговизна оборудования. При построении крупных сетей нужны высокопроизводительные маршрутизаторы для быстрого расчета новых маршрутов, в случае отказа одного из каналов связи.

Благодаря своим преимуществам перед остальными протоколами динамической маршрутизации, в настоящее время протокол OSPF широко применяется при построении средних и крупных сетей передачи данных.

В сравнении с традиционной архитектурой, концепция SDN предоставляет более гибкую инфраструктуру и предусматривает передачу управляющих функций центральному серверу-контроллеру, таким образом, заменяя традиционную распределенную модель маршрутизации на централизованную (рис. 6, б). Построение маршрута осуществляется на основе глобального представления сети (состояния всех сетевых элементов), которое должно поддерживаться в актуальном состоянии. Такой централизованный подход позволяет более функционально, быстро и эффективно управлять потоками данных в сети, т.к. создание новой топологии сети – это вычислительный процесс [2].

Важным элементом функционирования сети SDN является установка правил управления потоками в коммутаторе. Выделяют два основных подхода.

1. Реактивный подход. На запрос установки нового потока контроллер формирует одно правило и устанавливает его на коммутатор, инициировавший запрос. Он требует значительных вычислительных ресурсов контроллера, поскольку на каждом шаге маршрута запрашивается новое правило, однако не требуется информация о состоянии всей сети.

2. Проактивный подход. На запрос установки нового потока контроллер вычисляет маршрут (или его часть) для потока и устанавливает соответствующие правила на все коммутаторы этого маршрута. Он позволяет вычислить весь маршрут на основании информации о состоянии всей сети и быстро пропустить поток, тем самым значительно увеличивая пропускную способность.

Для разного типа трафика возможно использование одного из наиболее приемлемых подходов или при необходимости одновременно обоих.

Основными производителями оборудования для сетей SDN являются Huawei, Cisco, ZTE, Nokia, Run OS (Russian networks operation systems), HP. Типичные характеристики контроллера SDN (на примере контроллера RUN OS) представлены в таблице.

Характеристики контроллера RUN OS

Характеристика	Значение
Скорость обработки	30 млн. потоков/сек
Время установки нового соединения	45 мкс
Количество поддерживаемых коммутаторов	1000
Поддерживаемые технологии	Многопоточковая маршрутизация, работа с сетевыми протоколами (ARP, DNS, DHCP, BGP), трансляция адресов (NAT), балансировка нагрузки, виртуализация сетей, анти-DDoS, верификация сети, интеграция с системой управления ЦОД

Как видно из таблицы, контроллер поддерживает весь набор стандартных сетевых протоколов и при этом все необходимые расчеты производятся централизованно на нем, что позволяет значительно сократить нагрузку на промежуточное оборудование (коммутаторы).

Основные достоинства технологии SDN следующие.

1. Упрощение процесса вычисления маршрута. Расчет оптимального маршрута производится в контроллере, после чего в таблицу потоков коммутатора вносится изменение. Так же в контроллере можно настроить готовый сценарий выбора маршрутов при отказе одного из коммутаторов.

2. Увеличение возможности для модернизации сети, т.к. модернизация сети SDN – это задача написания нового приложения.

3. Выполнение всех сложных вычислений на контроллере. Вместо дорогостоящих маршрутизаторов можно использовать более простые и дешевые устройства.

К недостаткам SDN следует отнести:

1. Единая точка отказа. В случае выхода из строя контроллера ко всем коммутаторам будет применена базовая конфигурация (по умолчанию) и весь функционал, ранее установленный контроллером, будет утерян.

2. Критичность к ошибкам в настройке. Ошибка в настройке контроллера может привести к неработоспособности всей сети, а не только ее части; для современных программных средств конфигурирования контроллеров вероятность данной ошибки крайне низкая [1].

Теоретически неограниченные возможности сетей SDN к расширению позволяют строить крупные сети, масштабируемые в зависимости от решаемых задач. При этом сеть обладает требуемой «интеллектуальностью», необходимой для организации совместной работы больших групп коммутаторов.

Заключение

По итогам проведенного аналитического сравнения методов и средств управления SDN с традиционными методами и средствами управления сетевыми ресурсами и потоками данных в коммутируемых сетях можно отметить, что подход SDN позволяет выполнять функцию маршрутизации и доставки данных значительно лучше, чем в традиционных сетях. Подход SDN предоставляет более гибкие возможности по управлению трафиком, перегрузками, управлению сетевой инфраструктурой с помощью сетевых приложений, запущенных на логически централизованном контроллере. В распределенной среде традиционных коммутируемых сетей эти функции также реализованы, но в гибкости управления они существенно проигрывают возможностям, предоставляемым SDN сетями.

AUTOMATION AND SCALING OF LOCAL NETWORKS WITH USING SDN

S.U. PARKHOMIK, A.D. SIAMAK, I.L. SELEZNEV

Abstract

This article shows the main problem of the traditional approach to building data networks. A principle of SDN network organization and its main components are analyzed. Comparison of approach using SDN networks and traditional approach was carried out.

Keywords: data network, software-defined network, information flow, controller.

Список литературы

1. *Смелянский Р.Л.* Компьютерные сети. М., 2011.
2. Архитектура SDN. Анализ основных проблем на пути развития. [Электронный ресурс]. – Режим доступа: www.hups.mil.gov.ua/periodic-app/article/12126/soi_2015_3_21.pdf.
3. Software-Defined Networking: The New Norm for Networks. [Электронный ресурс]. – Режим доступа: <https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf>.
4. Автоматизация масштабирования локально-вычислительных сетей с использованием SDN. [Электронный ресурс]. – Режим доступа: <https://libeloc.bsuir.by/handle/123456789/13313>.