

УДК 004.7, 004.732

РАЗРАБОТКА ПРОТОТИПОВ СРЕДСТВ УПРАВЛЕНИЯ СЕТЕВЫМИ РЕСУРСАМИ И ПОТОКАМИ ДАННЫХ НА ОСНОВЕ ПРОГРАММНО-КОНФИГУРИРУЕМЫХ СЕТЕЙ OPENFLOW

Власов Д. В., Грудинин В. А., Каирканов А. Б., Садов О. Л., Сомс Л. Н., Титов В. Б., Хоружников С. Э., Чугреев Д. А., Шевель А. Е., Шкребец А. Е.

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики», Санкт-Петербург, Россия (197101, г. Санкт-Петербург, Кронверкский проспект, д.49), e-mail: xse@vuztc.ru

Объектом исследования являются программно-конфигурируемые сети (ПКС) OpenFlow и эффективность их применения для построения сетей хранения данных. В работе представлено: исследование принципов построения и изучение эксплуатационных характеристик программно-конфигурируемых сетей. Обсуждается создание прототипов средств управления сетевыми ресурсами и потоками данных на основе ПКС. Оценивается применимость ПКС для центров обработки данных (ЦОД) и распределенных систем хранения данных (СХД). Описываются эксперименты с различными модификациями OpenFlow контроллеров NOX и излагается специфика использования различных программных и аппаратных OpenFlow коммутаторов, а также приводятся сравнительные результаты эффективности применяемых в них специфических механизмов обеспечения методов QoS. Дается описание основных проблем существующих реализаций компонентов OpenFlow ПКС, выявленных в ходе исполнения работ, даются рекомендации по возможным путям их решения.

Ключевые слова: программно-конфигурируемая сеть (ПКС), сеть хранения данных, система хранения данных (СХД), центр обработки данных (ЦОД), система хранения данных, OpenFlow.

THE PROTOTYPE DEVELOPMENT OF THE TOOLS TO MANAGE THE NETWORK RESOURCES AND DATA FLOWS WITH OPENFLOW SOFTWARE-DEFINED NETWORK

Vlasov D. V., Grudin V. A., Kairkanov A. B., Sadov O. L., Soms L. N., Titov V. B., Khoruzhnikov S. E., Chugreev D. A., Shevel A. E., Shkrebits A. E.

National Research University of Information Technologies, Mechanics and Optics, Saint-Petersburg, Russia (197101, Saint-Petersburg, Kronverkskiy pr., 49), e-mail: xse@vuztc.ru

The purpose of the study are software-defined network (SDN) OpenFlow and effectiveness of their approach to build a network storage. Here we present the study how to build up and study of operating characteristics of software-configurable networks. Prototyping tools for managing network resources and data flows on the basis of the SDN are described. The estimation of the SDN approach for data centers (DC) and the distributed storage systems is observed. We describe experiments with various modifications of OpenFlow controller NOX and set out the specifics of the use of various software and hardware OpenFlow switches. We provide comparative performance results of specific QoS methods. A description of the main problems of existing implementations of OpenFlow SDN components identified during the research, and recommendations on possible ways to address them are provided.

Key words: Software-Defined Networks (SDN), Storage Area Network (SAN), Storages, Data Center, OpenFlow.

Введение

Выбор данного объекта исследований обусловлен тем, что архитектура компьютерных сетей (КС), применяемая в настоящее время, перестает соответствовать предъявляемым к ней требованиям [1]. В ситуации, когда происходит отказ от иерархических топологий и ориентации на полностью автономные сетевые устройства, технологии программно-конфигурируемых сетей – ПКС (Software Defined Networks – SDN) могут занять ведущее положение в ЦОД с высокой степенью виртуализации, ориентированных на предоставление

многопользовательских облачных сервисов. ПКС обладают достаточным потенциалом для коренного изменения ситуации на рынке производителей сетевого оборудования и призваны повысить роль программного обеспечения в процессах управления компьютерными сетями.

Значительный интерес к технологиям ПКС в настоящее время вызван тем, что эффект от внедрения может быть весьма существенным. Передача части функций сетевого оборудования стандартным серверам, использующим специализированное программное обеспечение, может привести к тому, что новое поколение сетевых устройств будет обладать более высокой степенью унификации, низкой стоимостью и уменьшенным энергопотреблением. Упрощение архитектуры оборудования приведет к более эффективному использованию ресурсов и снижению капитальных затрат при сохранении технических характеристик на прежнем уровне. Унификация и упрощение протоколов конфигурирования устройств повысит степень автоматизации, что в свою очередь приведет к повышению производительности и улучшению условий труда администраторов ЦОД и СХД. Централизованное управление сетями, которое является отличительной особенностью ПКС, позволит обеспечить надежную защиту систем от несанкционированного доступа. Возможность применения единых политик в масштабах всей сети приведет к внедрению более высоких стандартов качества обслуживания. Программная реализация основных функций управления потоками данных даст возможность оперативно внедрять новые сервисы, что в конечном итоге приведет к повышению качества обслуживания пользователей ЦОД.

Одним из наиболее интенсивно развивающихся направлений ПКС на настоящий момент является спецификация OpenFlow [2]. Данная технология предполагает четкое разделение функциональности между сетевыми коммутаторами, удовлетворяющим требованиям довольно простой спецификации, и внешними программными контроллерами, реализующими алгоритмы управления сетевой инфраструктурой любой степени сложности. Коммутаторы обслуживают фиксированный набор функций согласно установкам в специализированных таблицах управления потоками (flow tables). В зависимости от сообщений, генерируемых коммутаторами (например, содержащих информацию об отсутствии правил, соответствующих полученному коммутатором сетевому пакету), программные контроллеры формируют изменения в таблицах управления потоками соответствующих коммутаторов.

Постановка задачи

Целью работ было исследование принципов построения и эксплуатационных характеристик программно-конфигурируемых сетей, а также создание прототипов средств управления сетевыми ресурсами и потоками данных на основе ПКС, исследование и оценка

применимости ПКС для ЦОД и распределенных СХД. Для экспериментальных исследований были выбраны ПКС на основе протокола OpenFlow и оценивалась эффективность их использования при управлении СХД с интерфейсом iSCSI.

Были поставлены задачи по созданию прототипов средств управления сетевыми ресурсами и потоками данных, а также для методов обеспечения Quality of Service (QoS).

Состав экспериментального стенда

В качестве основных средств для прототипирования были выбраны системы программной эмуляции:

- OpenFlow коммутатора на базе разработок программного проекта CPqD/of12softswitch [3];
- контроллера сети OpenFlow CPqD/nox12oflib [4];
- системы эмуляции OpenFlow сети Mininet [5].

В ходе дальнейших исследований разработанные прототипы адаптировались для аппаратных коммутаторов, поддерживающих спецификацию OpenFlow – Pica8 и HP 3500-24G-PoE у1.

В качестве СХД использовались хранилища HP P4300 G2 7.2TB SAS Starter SAN BK716A, поддерживающие протокол iSCSI.

Специализированный модуль контроллера

Для обеспечения возможности управления сетевыми ресурсами и потоками данных, а также методами обеспечения QoS на базе модуля NOX switch был разработан специализированный модуль switchqos.

Данный модуль обеспечивает функциональность вычисления маршрутов прохождения пакетов через набор коммутаторов OpenFlow и генерации таблиц потоков (flow table) для каждого из этих коммутаторов. При изменении топологии соединений и прерывании потока данных по истечении периода ожидания производится изменение маршрута, и в таблицы потоков коммутаторов загружаются соответствующие изменения.

Для осуществления возможности управления методами обеспечения QoS в модуле задается привязка определенного набора TCP/UDP портов к соответствующим уровням обслуживания QoS. В данном случае следует принимать во внимание, что методы обеспечения QoS в OpenFlow коммутаторах могут быть реализованы либо через механизм очередей (queue) OpenFlow, либо через традиционные механизмы IP ToS и VLAN PCP. Возможность применения той или иной реализации зависит от типа коммутатора.

С учетом указанных выше моментов, а также в связи с тем, что разные коммутаторы поддерживают разные версии протокола OpenFlow, были реализованы различные варианты специализированного модуля NOX classic, NOX, nox12oflib [6].

Управление сетевыми ресурсами и потоками данных

В качестве системы управления сетевыми ресурсами и потоками данных был разработан набор программных модулей, позволяющий осуществлять подключение и отключение соединений между коммутаторами.

В режиме программной эмуляции это осуществлялось при помощи модуля на языке программирования Python, использующем программную оснастку Mininet.

Для аппаратных коммутаторов были разработаны специализированные модули на языке программирования Python, осуществляющие удаленное управление через зашифрованный SSH-канал в режиме управления CLI.

Для проведения экспериментов использовалась топология типа «петля» (loop topology), состоящая из 4-х коммутаторов (узлы s2, s3, s4, s5), хостов для генерации и приема трафика (узлы h1 и h6) в соответствии с рисунком 1:

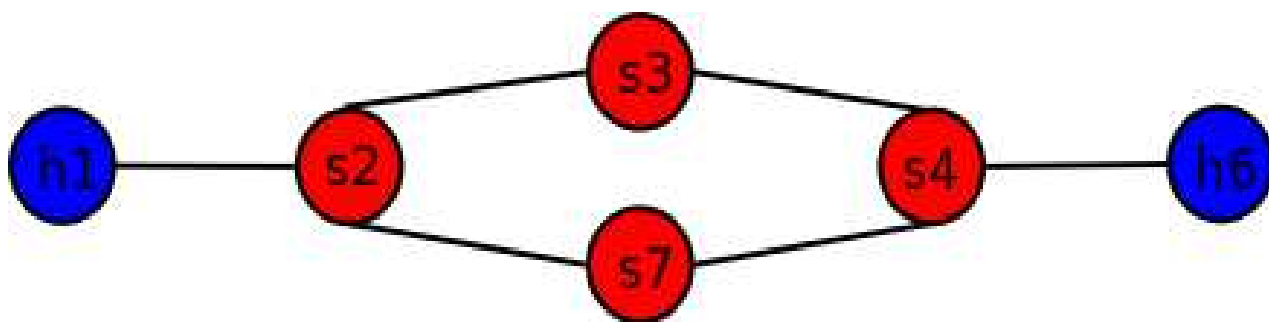


Рисунок 1. Схема экспериментальной сети с топологией «петля»

Как известно, топология «петля» представляет собой довольно серьезную проблему при организации сетей в силу неопределенности предпочтений при пересылке сетевых пакетов; при образовании такого рода конфигураций передача пакетов может и вовсе остановиться. Коммутатор, находящийся в точке ветвления, по сути дела, попадает в ситуацию Буриданова осла. Для снятия такого рода неопределенностей в традиционных сетевых решениях разрабатываются специальные средства и алгоритмы, например – Spanning Tree Protocol (STP).

Эксперименты заключались в передаче тестового трафика (ping) с хоста h1 на h6. В начальном состоянии все узлы соединены согласно приведенной схеме. Приложение контроллера находится в состоянии неопределенности, схема маршрутизации не выстраивается, и пакеты не проходят. При отключении одного из соединений маршрут выстраивается, начинается прохождение пакетов. После этого даже восстановление отключенного маршрута (и соответственно – восстановление «петли») не останавливает тестового трафика. Отключение же активного маршрута после окончания периода ожидания приводит к автоматической перестройке топологии и перенаправлению трафика по другому

маршруту.

Методы обеспечения QoS

Управление приоритетностью обслуживания сетевых потоков через методы QoS осуществляется при помощи набора программных модулей, написанных на языке программирования Python, путем установки значения пропускной способности (bandwidth) для очередей OpenFlow или уровней обслуживания ToS/PCP для коммутаторов. Для управления программным коммутатором используется специализированная программа управления dpctl. Управление аппаратными коммутаторами осуществляется через зашифрованный SSH-канал в режиме управления CLI.

Для проверки возможности использования ПКС в ЦОД была построена модель ЦОД. В состав этой модели в соответствии с рисунком 2 входят СХД и несколько виртуальных машин (далее – VM). Первая VM выполняет роль коммутатора, совместимого со стандартом OpenFlow версии 1.2, вторая служит генератором трафика для СХД по протоколу iSCSI, а остальные предназначены для генерации и приема нагрузочного трафика.

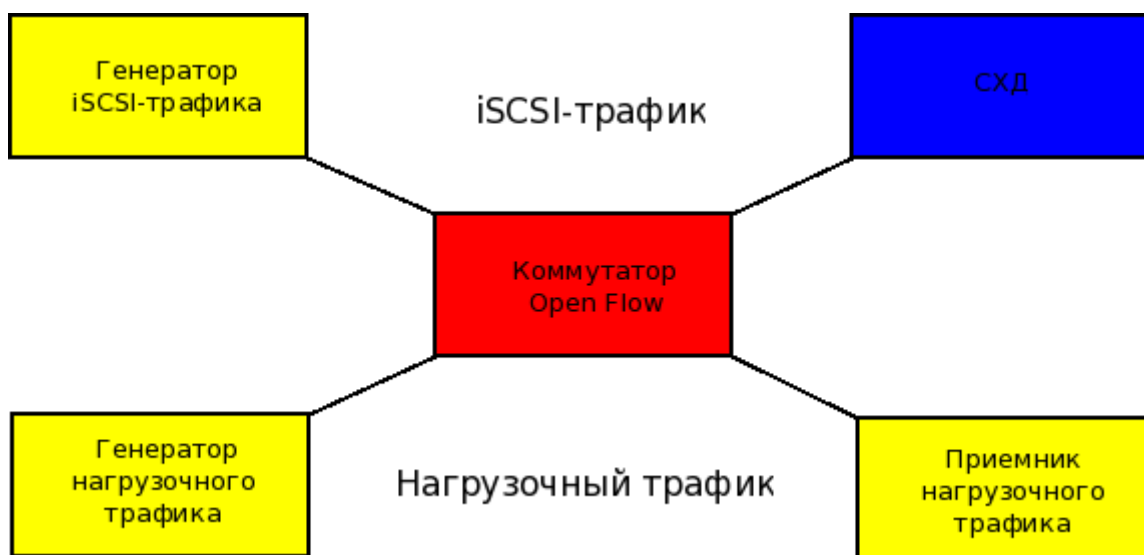


Рисунок 2. Модель ЦОД, построенная на базе ПКС

Для проведения тестирования производилось чтение данных с СХД по протоколу iSCSI при параллельной генерации нагрузочного трафика. Для создания нагрузочного трафика использовались специализированное программное средство для измерения эксплуатационных характеристик IP сетей Iperf и генератор трафика VoIP (голосовой связи) SIPp.

В ходе проводившихся экспериментов было выяснено, что стандартная модель задания диапазона для динамического изменения значения пропускной способности, применяемая Linux Traffic Control в CPqD/of12softswitch для проведения требуемых измерений, неприменима в силу стохастического характера процессов изменения пропускной способности при потере соединения по iSCSI в случае интенсивного нагрузочного трафика.

Для решения этой проблемы динамическая модель была заменена на установку фиксированной полосы пропускания. Модифицированная версия программного коммутатора размещена в репозитории [7]. С помощью утилиты `drctl` задаются доли ширины канала для разных очередей QoS в процентах относительно общей, причем сумма по всем очередям не обязательно должна равняться ста процентам. По таблице 1 видно наличие влияния управления очередями на результирующую скорость взаимодействия с СХД, однако она меняется достаточно незначительно относительно изменений пропускной способности.

Пропускная способность для очереди, в % от максимальной		Скорость взаимодействия с СХД, Кбайт/с
полезный трафик	нагрузочный трафик	
100	0	35.1
100	0.1	31.6
100	100	8.3
0.1	100	5.4
0.1	0.1	9.2

Таблица 1. Скорость взаимодействия с СХД через программный коммутатор в зависимости от заданных пропускных способностей для очередей QoS

При задании долей ширины канала для очередей на аппаратном коммутаторе HP сумма должна не превышать сто процентов. Проведение аналогичного теста с аппаратным коммутатором показало наличие корреляции между задаваемой пропускной способностью очереди и получаемой скоростью прохождения трафика (см. таблицу 2).

Пропускная способность для очереди, в % от максимальной		Скорость взаимодействия с СХД, Мбит/с
полезный трафик	нагрузочный трафик	
100	0	10.0
80	20	8.4
20	80	2.1
0	100	0

Таблица 2. Скорость взаимодействия с СХД через аппаратный коммутатор в зависимости от заданных пропускных способностей для очередей QoS

Не сравнивая абсолютные скорости прохождения данных, завязанные на априорные ограничения общей ширины канала, можно отметить преимущества механизма, реализованного на аппаратном коммутаторе, в сравнении с программным: хороший уровень точности, невозможность задания суммарной пропускной способности больше 100 %.

Выводы

Экспериментальные исследования показали, что применение OpenFlow ПКС в ЦОД эффективно, позволяет динамически конфигурировать компоненты сетевой инфраструктуры и задавать оптимальные параметры передачи данных для различных типов сетевого трафика.

Разработанные программные компоненты были размещены в программных репозиториях [6] и [7], а также оформлены в виде программных пакетов для дистрибутива ОС Linux НауЛинукс [8].

В ходе работ были выявлены следующие проблемы:

- различные механизмы управления политиками QoS в разных маршрутизаторах;
- нестабильность работы контроллеров и маршрутизаторов;
- отсутствие отработанных механизмов/методик низкоуровневой оптимизации трафика в СХД крупных ЦОД.

Возможные методы решения указанных проблем:

- создание высокоуровневых абстракций для работы с различными маршрутизаторами;
- создание высокоуровневого API, позволяющего задавать политики передачи данных и допускающего мониторинг и управление для различных уровней абстракции (IAAS/PAAS/SAAS);
- создание отказоустойчивых ПКС-конфигураций;
- адаптация/разработка алгоритмов, ориентированных на поддержание заданного уровня QoS при передаче данных в сетях OpenFlow.

Статья подготовлена на материале исследований, которые проводились при финансовой поддержке Министерства образования и науки Российской Федерации в рамках государственного контракта № 07.514.11.4152.

Список литературы

1. Операционная система НауЛинукс [Электронный ресурс]. – Режим доступа: <http://www.naulinux.ru/> (дата обращения: 29.03.2013).
2. Репозитории контроллера OpenFlow ИТМО [Электронный ресурс]. – Режим доступа: <https://github.com/itmo-infocom/> (дата обращения: 13.06.2013).
3. Репозиторий программного коммутатора ИТМО [Электронный ресурс]. – Режим доступа: <https://github.com/itmo-infocom/of12softswitch/commit/d376d8dfe9973d3b74d1b17e83ebd532c5b4fd14> (дата обращения: 26.03.2013).
4. Спецификации OpenFlow [Электронный ресурс]. – Режим доступа: <https://www.opennetworking.org/sdn-resources/onf-specifications/openflow> (дата обращения: 13.06.2013).
5. CPqD/of12softswitch [Электронный ресурс]. – Режим доступа: <https://github.com/CPqD/of12softswitch> (дата обращения: 28.03.2013).
6. CPqD/nox12oflib [Электронный ресурс]. – Режим доступа: <https://github.com/CPqD/nox12oflib> (дата обращения: 28.03.2013).

7. Mininet [Электронный ресурс]. – Режим доступа: <http://mininet.github.com/> (дата обращения: 28.03.2013).

8. Open Networking Foundation White Paper Software-Defined Networking: The New Norm for Networks [Электронный ресурс]. – Режим доступа: <https://www.opennetworking.org/images/stories/downloads/white-papers/wp-sdn-newnorm.pdf> (дата обращения: 10.11.2012).

Рецензенты:

Парфенов Владимир Глебович, д-р техн. наук, профессор, декан факультета информационных технологий и программирования Санкт-Петербургского национального исследовательского университета информационных технологий, механики и оптики (НИУ ИТМО), г. Санкт-Петербург.

Горелик Самуил Лейбович, д-р техн. наук, профессор Санкт-Петербургского национального исследовательского университета информационных технологий, механики и оптики (НИУ ИТМО), г. Санкт-Петербург.