

ИССЛЕДОВАНИЕ СПОСОБОВ УПРАВЛЕНИЯ QoS В ПРОГРАММНО-КОНФИГУРИРУЕМЫХ СЕТЯХ С ПОМОЩЬЮ КОНТРОЛЛЕРА RYU

Садов О.Л.¹, Тивиков Н.В.¹

¹ Санкт-Петербургский национальный исследовательский университет Информационных технологий, механики и оптики, Санкт-Петербург, Россия (197101, Санкт-Петербург, Кронверкский проспект, д. 49), e-mail: n.tivikov@gmail.com

В данной работе рассмотрены методы обеспечения QoS с помощью протокола OpenFlow в программно-конфигурируемых сетях. Целью работы является предложение метода управления качеством обслуживания в SDN с учетом проблем неполной поддержки протоколов со стороны производителей оборудования и специфичности его настройки. Описаны механизмы реализации QoS в OpenFlow-сетях. Выделены проблемы при работе с оборудованием, имеющим различные сценарии конфигурирования, определяемые производителями. Рассмотрен контроллер SDN Ryu, описан REST API, позволяющий управлять сетевой инфраструктурой на уровне приложений. Предложено решение проблемы настройки оборудования на основе расширяемого модуля для контроллера и драйверов для соответствующих сетевых устройств, работа с которым осуществляется с помощью REST API. Определены основные команды для управления оборудованием. Выделены достоинства и недостатки подхода.

Ключевые слова: QoS, SDN, OpenFlow, Ryu, контроллер.

RESEARCH OF WAYS OF MANAGEMENT OF QOS IN THE PROGRAM CONFIGURED NETWORKS BY MEANS OF THE RYU CONTROLLER

Sadov O.L.¹, Tivikov N.V.¹

¹ Saint Petersburg National Research University of Information Technologies, Mechanics and Optics, Saint Petersburg, Russia (197101, Saint Petersburg, Kronverksky Pr., 49), e-mail: n.tivikov@gmail.com

In this article are considered methods of providing QoS by means of the OpenFlow protocol in the software defined networks. The purpose of work is the offer of a method of management quality of service in SDN taking into account problems of incomplete support of protocols from producers of the equipment and specificity of its control. Mechanisms of realization of QoS in OpenFlow-based networks are described. Problems are allocated during the work with the equipment having various scenarios of a configuration determined by vendors. The SDN Ryu controller is considered, REST API allowing to operate network infrastructure at the level of appendices is described. The solution of the problem of control of the equipment on the basis of the expanded module for the controller and drivers for the corresponding network devices work with which is carried out with the help of REST API is proposed. The main teams for management of the equipment are defined. Approach merits and demerits are allocated.

Keywords: QoS, SDN, OpenFlow, Ryu, controller.

Концепция SDN (Software-defined Networking, программно-конфигурируемая сеть) заключается в разделении плоскостей данных и управления. SDN ориентирован на взаимодействие с пользователем через открытый API. Управление данными передается контроллеру. Это позволяет упростить коммутаторы, сняв с них вычислительную нагрузку. Пересылка пакетов осуществляется по таблице маршрутизации, получаемой с контроллера. Таблица создается не для конкретного пакета, а для типа пакетов — потока[6]. Таким образом, SDN видится как решение проблем сетей с оборудованием разных производителей, когда конфигурирование сети затруднено специфичностью оборудования и разным подходом к управлению.

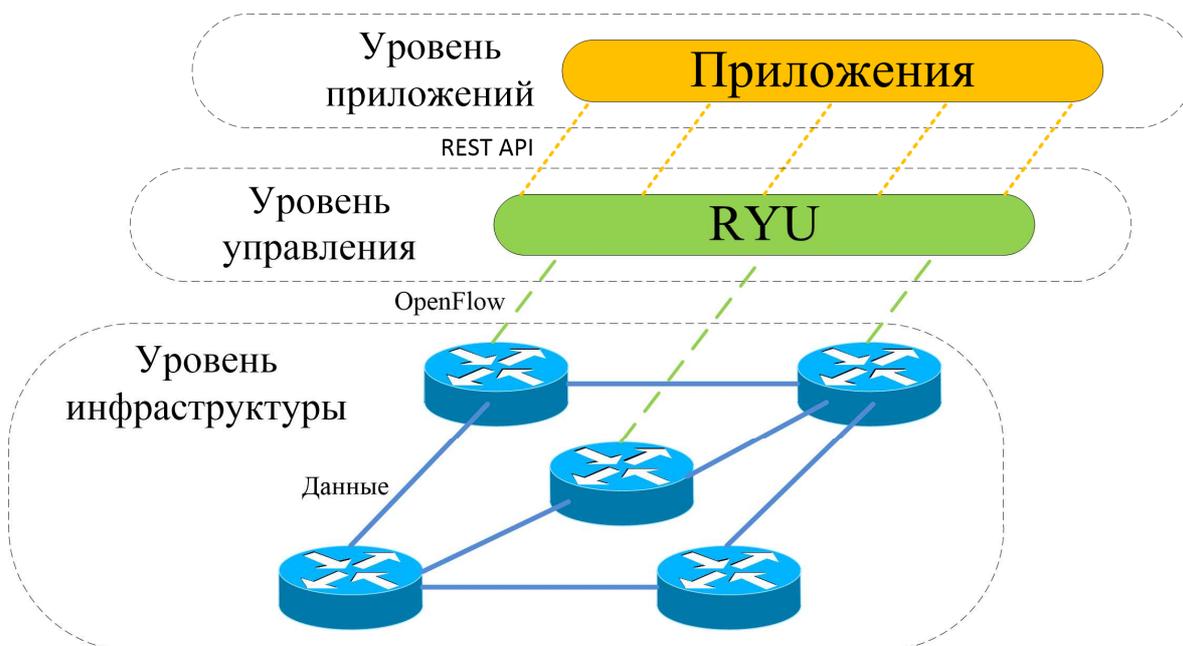


Рис.1. Архитектура программно-конфигурируемой сети

Протокол OpenFlow на данный момент является популярным решением для построения SDN. Но, к сожалению, его недостаточно своевременная поддержка со стороны производителей не позволяет в полной мере использовать все функции по управлению QoS в сети. Для решения проблем специфичности настройки оборудования был создан протокол OF-CONFIG, но на данный момент он не получил достаточного распространения

В статье предложен один из способов решения проблемы специфичности управления оборудованием различных производителей с помощью абстрактного модуля для контроллера OpenFlow Ryu. Модуль предоставляет возможность добавлять различные конфигурации нового оборудования, благодаря разделению на framework и драйвера. Также модуль не зависит от версии OpenFlow и позволяет работать с оборудованием, не поддерживающим его вовсе. Управление модулем осуществляется через REST API, как и любым другим модулем контроллера Ryu.

Работа QoS в SDN с использованием протокола OpenFlow и контроллера Ryu

Сейчас управление QoS в OpenFlow возможно двумя базовыми механизмами:

1. Обеспечение QoS через очереди выходного порта.

У OpenFlow есть протокол-расширение OF-CONFIG, позволяющее управлять основными настройками коммутаторов, такими как очереди и состояния портов. Управление производится централизованно, например, через контроллер[3]. В зависимости от наличия или отсутствия поддержки протокола OF-CONFIG у коммутаторов и контроллеров, возможно два способа работы с очередями.

- Без поддержки OF-CONFIG.

Требуется предварительная настройка стандартными методами (например, CLI) очередей на интерфейсах коммутатора с указанием пропускных способностей и привязки пакетов к ним. Привязка осуществляется в зависимости от параметров, содержащихся в полях TOS/PCP/DSCP. Эти поля являются частью множества RFC (Request for Comments, рабочих предложений) ISOC (Internet Society, Общество Интернета), которые описывают стандарты Интернет. Они широко используются в классических сетях и предназначены для классификации трафика [1]. Значение этих полей применяется к целому потоку на этапе его создания на стороне контроллера. В соответствии с настройками, коммутатор направляет трафик потока в ту или иную очередь.

- С поддержкой OF-CONFIG.

Через контроллер, поддерживающий OF-CONFIG, создаются и настраиваются очереди выходных портов. Коммутатор также должен поддерживать OF-CONFIG той же версии. При создании потока указывается номер очереди, в которую коммутатор будет направлять соответствующий трафик.

2. Обеспечение QoS через метрики потоков.

Данный механизм был добавлен в OpenFlow 1.3[4]. Метрика позволяет измерять и контролировать интенсивность трафика потока, который к ней привязан. В каждой метрике может быть одна или более полосы пропускания, образуя диапазоны, на которых они применяются и происходит ограничение трафика по определенным правилам. В качестве необязательных метрик доступны только отброс пакета, для ограничения полосы пропускания, и перемаркировка DSCP, как реализация простейших политик DiffServ.

Эти механизмы несут в себе несколько проблем. Протокол OF-CONFIG пока не получил широкого распространения среди производителей OpenFlow оборудования. В случае отсутствия его поддержки, управление очередями полностью зависит от коммутаторов, и стандартного метода для этого нет. Поэтому, при работе с оборудованием разных производителей необходимо использовать разные механизмы и производить настройку децентрализованно, что требует значительных временных затрат и не позволяет менять настройки очередей «на лету». Поддержка протокола OpenFlow бывает ограничена ранними версиями, и поэтому механизм метрик потоков может быть недоступен.

Ryu представляет из себя контроллер SDN, написанный на языке Python, распространяющийся по свободной лицензии, поддерживающий OpenFlow версий 1.0, 1.1, 1.2, 1.3, 1.4, OF-CONFIG, NetConf и некоторые другие протоколы [5]. Для работы приложений с сетевой инфраструктурой он предлагает REST API, что в совокупности с модульной архитектурой позволяет добавлять новый функционал по мере необходимости.

REST API представляет из себя способ взаимодействия сетевых устройств в виде концепции клиент-сервер. Возможно взаимодействие с одним или несколькими промежуточными компонентами, а не непосредственно с сервером. Формат запросов к серверу соответствует URI (Uniform Resource Identifier, единообразный идентификатор ресурса), а методы обращения ограничены стандартными для HTTP GET, POST, PUT и DELETE. В запросе можно в явном виде задать формат ответа, как например JSON и XML. В ответе содержится его описание по MIME (Multipurpose Internet Mail Extensions — многоцелевые расширения интернет-почты) для правильной обработки и сами данные. Интерфейс для взаимодействия обеспечивается через web сервер WSGI. Использование REST API позволяет расширять функционал контроллера, обеспечивает стандартный интерфейс для доступа и простоту в использовании.

Предлагаемое решение

Одним из способов решения проблемы специфичности управления оборудованием различных производителей является разработка абстрактного модуля для контроллера SDN Ryu. Исходя из концепции, модуль представляет из себя framework, обрабатывающий вызовы REST API и направляющий их в соответствующий конкретному устройству драйвер. Драйвер, должным образом обработав команду, через выбранный разработчиком способ (SSH, Telnet, RS232 и др.), минуя протокол OpenFlow, направляет команду устройству в понятном для него виде. Вывод результатов команды, в случае его наличия, такой же, как и других команд REST API контроллера Ryu. В случае возникновения ошибок они направляются в журнал через соответствующий модуль.

Таблица 1

Список вызовов REST API, необходимых для управления очередями

Метод	Вызов	Данные	Описание
GET	/qos/queue/{switch} -switch: ["all" Switch ID]	*имя порта, *номер очереди	Получить настройки очередей на порту
POST	/qos/queue/{switch} -switch: ["all" Switch ID]	имя порта, номер очереди; пропускная способность;	Создать очередь
DELETE	/qos/queue/{switch} -switch: ["all" Switch ID]	*имя порта, *номер очереди	Удалить очередь

В таблице 1 описан список вызовов REST API, с помощью которых можно управлять очередями и правилами, классифицирующими трафик. Звездочкой помечены необязательные данные. В случае отсутствия в поле данных имени порта и/или номера очереди, действие применяется ко всем.

Достоинством такой схемы является расширяемость. Возможно добавление поддержки новых устройств, путем добавления драйвера и внесения редакций в соответствующие Python-скрипты. Таким же образом возможно расширение списка

поддерживаемых команд, с условием поддержки в драйвере и устройстве. Для пользователя или программы данный модуль представляется абстракцией, не зависящей от производителя сетевого оборудования и его особенностей конфигурации.

Недостатком можно считать необходимость составления таблицы, в которую будет занесена необходимая информация для подключения к устройству, а так же параметры для авторизации. При подключении новых устройств в сеть или отключении существующих таблицу тоже необходимо обновлять.

Заключение

В данной работе рассмотрены методы обеспечения QoS с помощью протокола OpenFlow, выделены проблемы при работе с оборудованием, имеющим различные сценарии конфигурирования. Рассмотрен контроллер SDN Ryu и REST API.

Предложено решение на основе модуля контроллера и драйверов для соответствующих сетевых устройств, работа с которым осуществляется с помощью REST API. Выделены достоинства и недостатки подхода.

Список литературы

1. В. Олифер, Н. Олифер Компьютерные сети. Принципы, технологии, протоколы: Учебник для ВУЗов. 4-е изд. — СПб.: Питер, 2010. — 944 е.: ил.
2. Mission — Network Technologies in Distributed Computing Systems [Электронный ресурс]. — Режим доступа: <http://sdn.ifmo.ru/hardware-switch/presentation/tyestirovaniie-qos-na-eksperimentalnom-stiendie>
3. OF-CONFIG 1.2 [Электронный ресурс]. — Режим доступа: <https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow-config/of-config-1.2.pdf>
4. OpenFlow Switch Specification Version 1.3.5 [Электронный ресурс]. — Режим доступа: <https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-switch-v1.3.5.pdf>
5. RYU SDN Framework - Ryubook 1.0 documentation [Электронный ресурс]. — Режим доступа: <http://osrg.github.io/ryu-book/en/html/>
6. 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>

Рецензенты:

Горелик С.Л., д.т.н., профессор, Санкт-Петербургского национального исследовательского университет информационных технологий, механики и оптики (НИУ ИТМО), Минобрнауки РФ, г. Санкт-Петербург;

Федоров А.В., д.ф.-м.н., профессор, зав. кафедрой ОФ и СЕ Санкт-Петербургского национального исследовательского университет информационных технологий, механики и оптики (НИУ ИТМО), Минобрнауки РФ, г. Санкт-Петербург.