

ИССЛЕДОВАНИЕ ДИНАМИЧЕСКОЙ КОНФИГУРАЦИИ БУФЕРОВ В OPENFLOW КОММУТАТОРАХ

А. А. Хакимов¹, А. С. А. Мутханна¹, Р. В. Киричек^{1*}

¹ СПбГУТ, Санкт-Петербург, 193232, Российская Федерация

* Адрес для переписки: kirichek@sut.ru

Аннотация

Предмет исследования. В программно-конфигурируемых сетях, как известно, главным основополагающим признаком является разделение функций, которые отвечают за передачу данных, и вынос приложений управления на отдельный сервер (контроллер). Такие технологии позволяют повысить качество обслуживания и снизить затраты на эксплуатацию сетей. Так же преимуществом является упрощение управления сетью, удешевление оборудования и разработка ранее недоступных сервисов. **Метод.** В данной работе предлагается метод для конфигурирования буфера OpenFlow коммутатора. **Основной результат.** В результате рассматривается возможность уменьшения скорости обработки пакета в узле, также на базе инструмента GNS3 тестировали сеть SDN и сегмент с использованием нашего скрипта.

Ключевые слова

SDN, openflow, буфер, скрипт, моделирование, GNS3.

Информация о статье

УДК 004.7:621.39

Язык статьи – русский.

Поступила в редакцию 17.05.17, принята к печати 02.06.17.

Ссылка для цитирования: Хакимов А. А., Мутханна А. С. А., Киричек Р. В. Исследование динамической конфигурации буферов в OpenFlow коммутаторах // Информационные технологии и телекоммуникации. 2017. Том 5. № 2. С. 34–44.

A STUDY OF THE DYNAMIC CONFIGURATION OF BUFFERS IN OPENFLOW SWITCHES

A. Khakimov¹, A. Muthanna¹, R. Kirichek^{1*}

¹ SPbSUT, St. Petersburg, 193232, Russian Federation

* Corresponding author: kirichek@sut.ru

Abstract—In software-defined networks, as is known, the main fundamental feature is the division of functions that are responsible for data transmission, and the removal of management applications

to a separate server (controller). Such technologies allow to improve the quality of service and reduce the costs of network operation. The same advantage is the simplification of network management, cheaper equipment and the development of previously inaccessible services. In this paper, we propose a method for configuring the switch OpenFlow buffer. As a result, we are considering the possibility of reducing the processing speed of the packet in the node, also using the GNS3 tool, we tested the SDN network and the segment using our script.

Keywords—SDN, openflow, buffer, script, modeling, GNS3.

Article info

Article in Russian.

Received 17.05.17, accepted 02.06.17.

For citation: Khakimov A., Muthanna A., Kirichek R.: A Study of the Dynamic Configuration of Buffers in OpenFlow Switches // Telecom IT. 2017. Vol. 5. Iss. 2. pp. 34–44 (in Russian).

Введение

Современные сетевые технологии эволюционируют уже около 40 лет. Задачи же все эти годы быстро усложнялись. Это связано со всё более глубоким проникновением информационных процессов во все сферы человеческой жизни. Особые требования возникают при необходимости обеспечить требуемый уровень QoS не только со стороны провайдера услуг, но и в локальной сети. Мало-помалу, всё чаще ставятся задачи динамического перераспределения ресурсов сети (пропускная способность, задержки и т. д.). Решить все эти проблемы в рамках традиционных сетевых технологий уже невозможно. Ответить на эти вызовы призвана технология Software Defined Network (SDN) [1, 2] и один из протоколов – OpenFlow¹ [3].

Известно, что системные администраторы используют тонкую настройку сетевых узлов, назначая им статический размер буфера. Обычно такие настройки хороши для 70 % всего трафика. концепция SDN позволяет динамически менять размер буфера во всех узлах в зависимости от потоков трафика, что позволит уменьшить время передачи пакета и потери пакетов [4, 5]. Для данной работы мы ввели понятие «виртуальный буфер», это некий виртуальный участок оперативной памяти, который будет использоваться только для одного потока.

Время, которое необходимо для доставки пакета (кадра) в сети связи включает в себя три основные составляющие: время распространения сигнала, время передачи кадра по линии связи и время ожидания начала передачи (ожидание в очереди). К этим трём параметрам, чаще всего, приходится добавлять четвёртый параметр, характеризующий производительность коммутатора, а именно время обработки кадра процессором коммутатора. Время распространения сигнала и время передачи по линии связи определяются такими параметрами сети как протяженность линий связи, битовая скорость передачи по линии и длина кадра. Эти параметры в сети проводной связи достаточно стабильны, чтобы считать их константами. Стоит отметить, что в сети беспроводной и тем более подвижной связи их чаще всего нельзя считать константами, а следует рассматривать как случайные величины. В данной работе мы будем рассматривать сети проводной связи, поэтому, далее будем рассматривать время распространения

¹ OpenFlow. URL: <https://www.opennetworking.org/sdn-resources/openflow>

и время передачи как постоянные величины. В такой постановке задачи на эти величины невозможно повлиять, поэтому далее будем рассматривать время ожидания в очереди и время обработки кадра.

Время ожидания в очереди зависит от интенсивности поступления кадров (интенсивности трафика), интенсивности обслуживания и размера буфера. Следует отметить, что при конечном размере буфера (очереди) появляется вероятность потери кадра (превышения установленного размера буфера). Интенсивность обслуживания определяется ресурсами системы, а именно скоростью передачи кадров по линии связи и временем обслуживания кадра процессором коммутатора. Поскольку скорость передачи является в данной задаче постоянной величиной, то будем рассматривать производительность процессора как основной ресурс, использование которого влияет на интенсивность обслуживания, следовательно, и на время ожидания в буфере.

При обработке кадра большая часть процессорного времени затрачивается на поиск в буфере кадра соответствующего потока. Чем больше размер буфера, тем больше времени уходит на поиск. Уменьшение этой величины возможно за счёт использования эффективных способов хранения и поиска данных. В качестве одного из возможных решений этой задачи рассмотрим структурирование данных в виде нескольких областей (буферов), количество которых равно количеству обслуживаемых потоков. В таком случае поиск кадров одного потока сводится к выбору соответствующей области памяти, т. е. поиск по количеству кадров сводится к поиску по количеству потоков. Таким образом, при линейном поиске это время уменьшается в n/r раз, где n – количество кадров, а r – количество потоков.

В данной работе предлагаем создавать виртуальные буфера для каждого потока трафика. Размер буфера будет меняться в зависимости от среднего размера пакета в потоке и значение QoS. Так как в традиционных узлах для поиска кадра из общей очереди по значениям QoS используют перебор всех кадров (*frame*) по очереди. И это занимает относительно больше время. В предложенной системе предлагается делить эти очереди, чтобы некий процессор, отвечающий за очередь, контролировал размеры и адреса этих буферов. Это позволит уменьшить время обработки и ретрансляции, за счёт другого алгоритма.

Протокол OpenFlow имеет все права на использование ресурсов коммутаторов, поэтому он идеально подходит для решения этой задачи. Теоретическая модель показала хороший показатель по круговой задержке и по потерям пакетов.

Протокол OpenFlow

Протокол OpenFlow предполагает, что OpenFlow обмен данными (например, коммутатор Ethernet, который поддерживает протокол OpenFlow) был сконфигурирован с учётом необходимых параметров [9], таких как IP-адреса контроллера, для установления соединения и дальнейшей работы с контроллером сети. Целью протокола конфигурации OpenFlow (OF-CONFIG) является предоставление возможности удалённого конфигурирования коммутаторов для дальнейшего обмена данными. Примером работы протокола OF-CONFIG является формирование таблицы переадресации и решений относительно действий, которые протокол OpenFlow должен выполнять.

OF-CONFIG определяет схему взаимодействия с логическим переключателем OpenFlow (абстракция переключателя). Протокол OF-CONFIG допускает задание характеристик логического переключателя, так чтобы контроллер мог взаимодействовать с переключателем и управлять им посредством протокола OpenFlow. Версия OF-CONFIG 1.1 определяет операционный контекст одного или более обменов OpenFlow (рис. 1).

Коммутатор, совместимый с OpenFlow, является эквивалентом реального физического или виртуального сетевого коммутатора (например, коммутатора Ethernet). Во время работы коммутатора происходит разделение таких ресурсов, как: порты, очереди (ОЗУ коммутатора) и т. д. Протокол OF-CONFIG позволяет производить динамическое распределение ресурсов коммутатора, совместимого с OpenFlow [9]. OF-CONFIG не специфицирует того, как получают ресурсы коммутатора, совместимого с OpenFlow. При этом, существует возможность логического разделения коммутатора, соответственно и его ресурсов, на несколько частей, тем самым формируя из одного коммутатора два и более других. OF-CONFIG предполагает, что такие ресурсы как порты и очереди используются совместно несколькими логическими коммутаторами (*OpenFlowLogicalSwitches*), так что каждый такой коммутатор может иметь полный контроль над ресурсами, которые ему предоставлены.

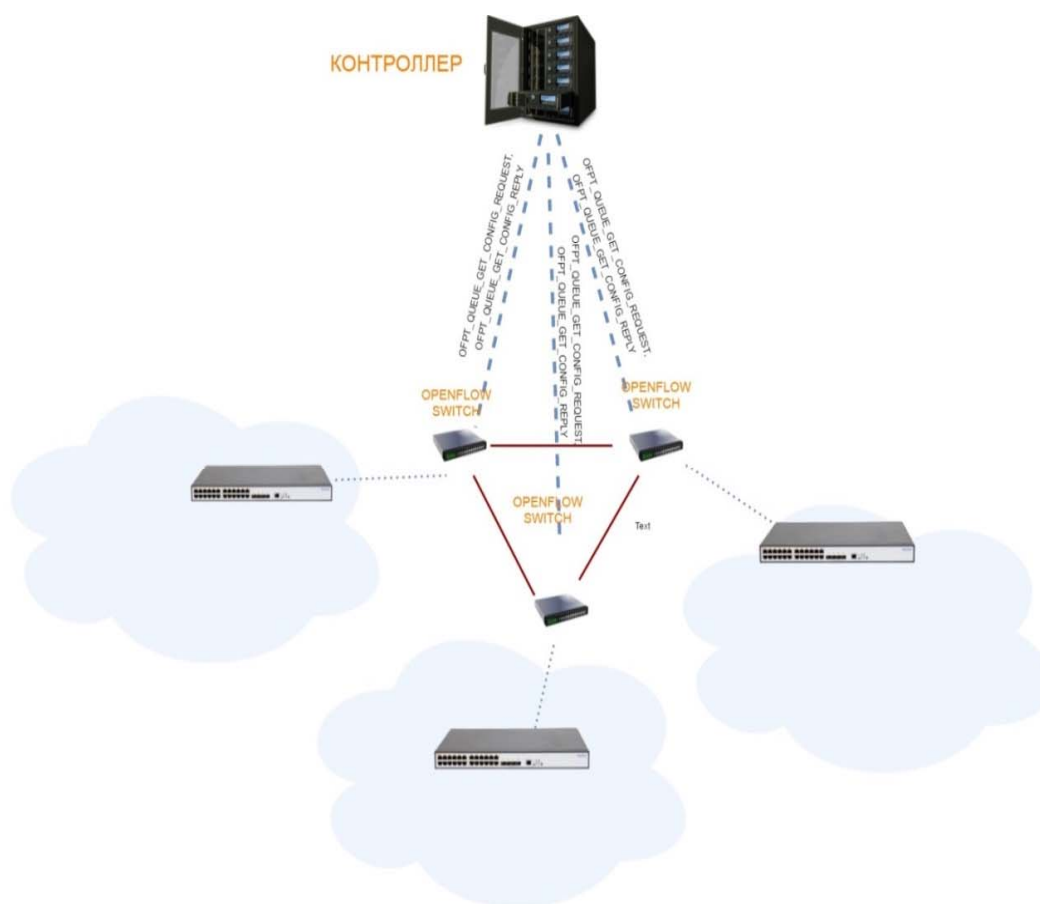


Рис. 1. Принцип работы контроллера и коммутаторов

Конфигурирование буфера OpenFlow коммутатора

В нашей модели мы предлагаем алгоритм расчёта буфера реализовать, как плагин контролера SDN. для этого мы ввели понятие «виртуальный буфер». Принцип нашей модели состоит в том что, контролер каждые промежутки времени ΔT будет менять размер буфера для всех сетевых узлов, чтобы привести показатель потери пакетов к нулевому значению. Так как OpenFlow коммутаторы отправляют пакеты неизвестного формата в контролер пакетом OpenFlow `rasket_in`, контролер сможет среагировать на этот вид пакетов, и рассчитать для этого вида трафика свой виртуальный буфер. А дальше контролер рассылает flow по всем коммутаторам, в котором также будут указаны оптимальный объём виртуального буфера.

Это позволит уменьшить величину RTT, часть которой вносится openflow коммутаторами. И распределять физическую память между потоками.

Разделяя очереди потоков, мы ускоряем процесс передачи всех потоков (*flows*). Теоретически, этот способ позволит уменьшить RTT до минимума (в пределах физики) и производить экономию физической памяти, т. е. не выделять много физической памяти на поток, где оно не нужно. Таким образом, данный метод позволит распределить физическую память между потоками.



Рис. 2. Зависимости полученных пакетов от отправленных

График зависимости полученных пакетов от отправленных (рис. 2). В случае если размер буфера оставлять статическим (*const*) то увеличению от количества пакетов наступает момент, где начинается потеря пакетов (перегрузка).

В нашей модели мы предлагаем динамически изменять размер буфера в случае если на предыдущем узле было отправлено большой поток пакетов для определенного адресата.

Суммарный размер всех виртуальных очередей равняется общему размеру буфера OpenFlow коммутатора.

В гистограмме (рис. 3) показан пример для очереди из 20 000 потоков трафика, с пропускной способностью интерфейса 2,5 Gbit/sec и оно составляет 2 000 пакетов суммарно.

Предлагается брать статистику с входных потоков и, создавая нормальное распределение, вычислять размер буфера для каждого потока и используя такие типы сообщений OpenFlow, как: OFPT_QUEUE_GET_CONFIG_REQUEST, OFPT_QUEUE_GET_CONFIG_REPLY, распределять размеры очередей.

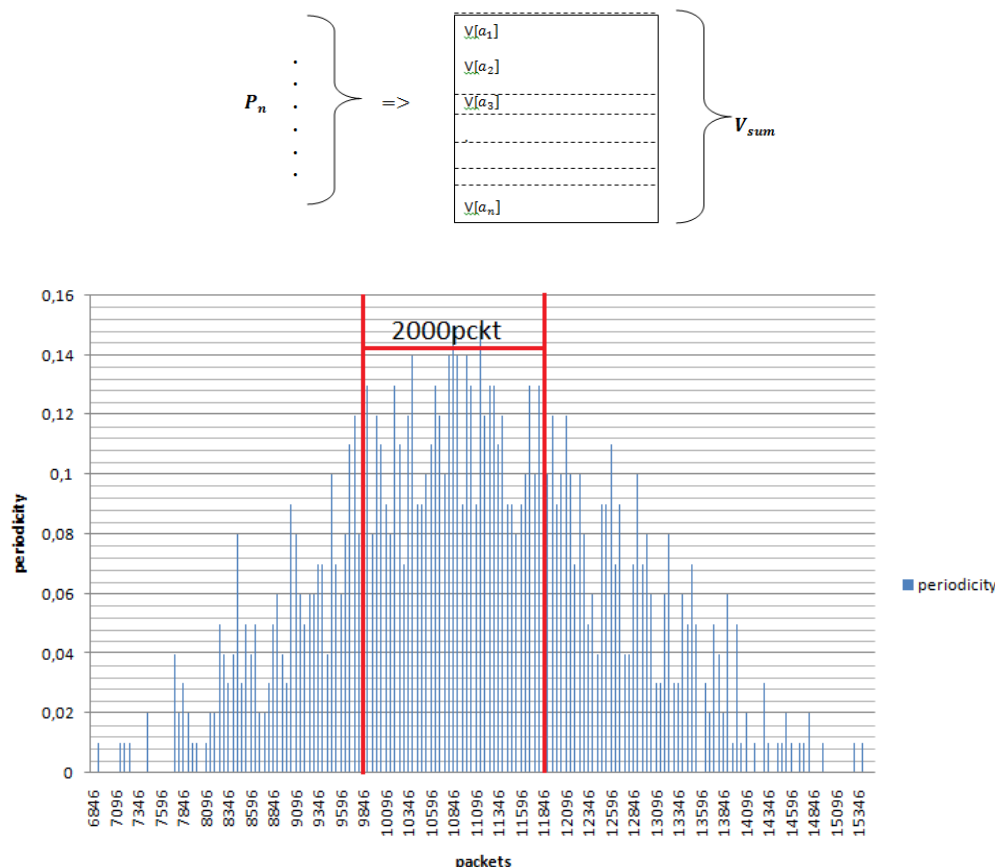


Рис. 3. Пример очереди для 20 000 потоков трафика

Разработка скрипта для конфигурирования буфера OpenFlow

В случае когда приходит пакет, скрипт проверяет данный пакет на порты назначения и адреса назначения и если он совпадает уже с существующим потоком, он помещает его в очередь того же потока. В случае если пакет не соответствует уже существующим потокам, он создает для него отдельный временный буфер, если это требуется. Сама концепция SDN и протокол OpenFlow позволит узлам заранее узнавать о массивных потоках от предыдущего узла, это происходит следующим образом, в случае если открывается сессия TCP узел А передаёт контролеру уведомление о новой сессии, тот в свою очередь назначает виртуальные буфера для узлов В, С... создавая своеобразный канал и также для узла А. При передаче полезных данных они будут использовать только те буфера, кото-

Тестирование разработанного метода

Мы сравнили передачу пакетов в условиях с общим буфером и с отдельными виртуальными, и получили следующие результаты по круговым задержкам для потока с пакетами, размером 1 500 Б.

Эксперимент проводили в имитационной среде GNS3 (рис. 6–8). В качестве хостов использовали генератор трафика HTTP. В качестве маршрутизатора с отдельными буферами использовали скрипт, разработанный нами.

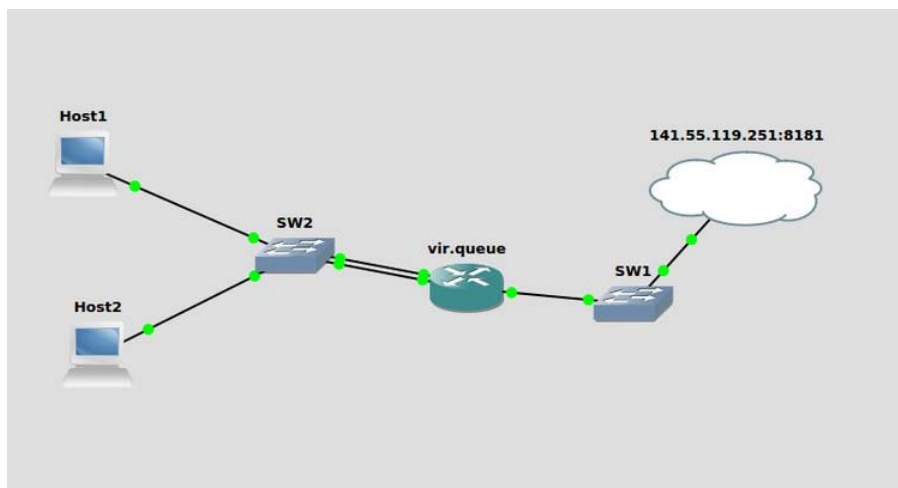


Рис. 6. Структура модели в симуляторе GNS3

Chart Title

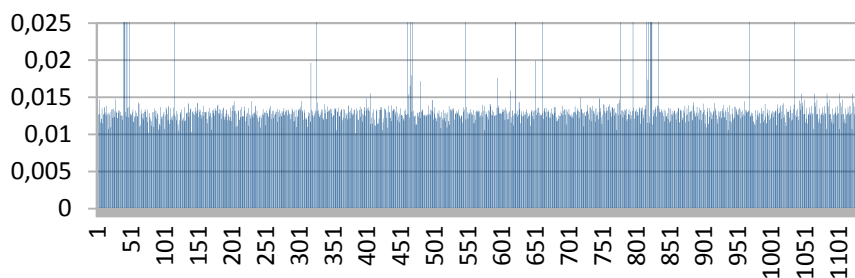


Рис. 7. RTT с размерами пакета 1500 Б с общим буфером

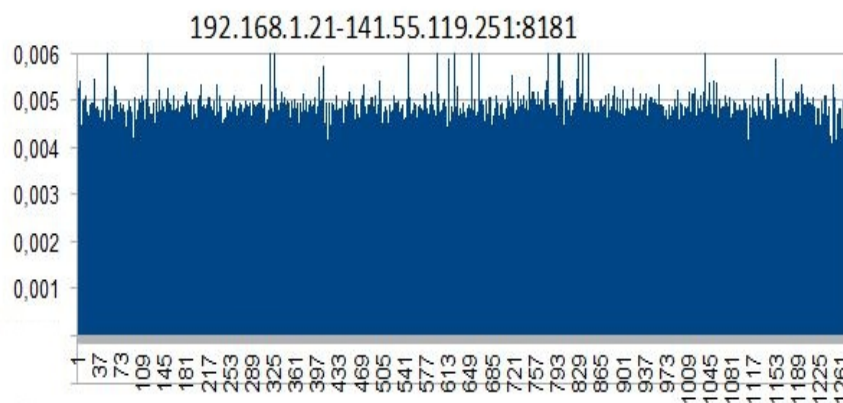


Рис. 8. RTT для пакетов с размерами 1 500 Б с виртуальными буферами

Задачи же все эти годы быстро усложнялись. Это связано со всё более глубоким проникновением информационных процессов во все сферы.

Принцип подбора размера очереди определяется по следующей формуле: $B = RTT * P$, где P – пропускная способность, RTT – круговая задержка.

Но для исследовательской работы, мы решили прибегнуть к методу подбора оптимального размера по следующим критериям: потери пакетов и круговая задержка. Для этого скрипт задает максимальное значение для буфера, и потом начинает уменьшать до тех пор, пока не начнутся потери. При отметке 5 % потери он запоминает размер буфера для данного типа потока (размер пакета и скорость обработки) как критическое значение, потом начинает увеличивать до оптимального уровня. В случае повторной подачи такого рода трафика, контролер автоматический задаст размер виртуального буфера для этого потока.

Результаты моделирования

По графикам видно (рис. 7, 8), что разница между простыми и виртуальными буферами составляет примерно 10 мс. Из этого следует, что скорость обработки пакета в узле увеличивается. А также производится экономное использование ресурсов памяти узла.

Также мы протестировали наш скрипт для трафика Интернета вещей, ведь по прогнозам ITU к 2020 г. количество Интернета вещей перевалит за 50 млрд [6, 7, 8]. В итоге, мы получили следующие результаты на графике 4 мы протестировали для пакетов размером 200 Байтов и получили круговую задержку равную 1,2 мс, что почти соответствует концепции тактильного интернета.

Chart Title

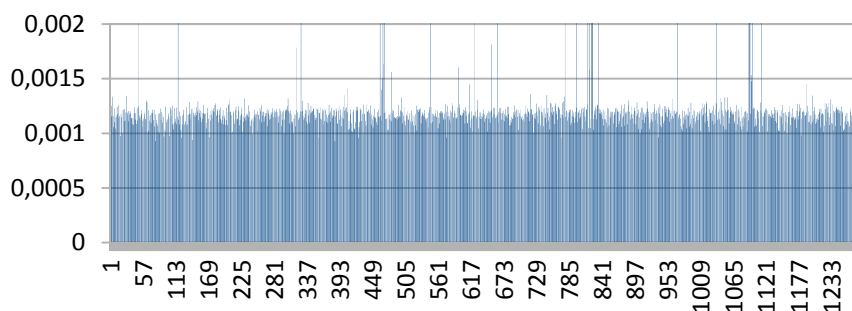


Рис. 9. RTT для пакетов размером 200 Б с отдельными буферами

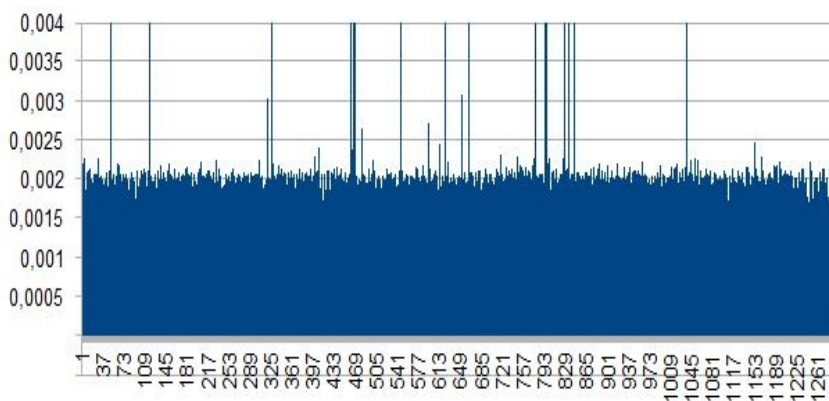


Рис. 10. RTT для пакетов размером 800 Б с отдельными буферами

Заключение

Экспоненциальный рост устройств, подключенных к Интернету и управление сетью, стали одной из самых сложных задач. Подход программно-конфигурируемых сетей позволяют значительно автоматизировать и облегчить управление сетями за счёт возможности их программирования. Существует возможность логического разделения коммутатора, соответственно и его ресурсов, на несколько частей, тем самым формируя из одного коммутатора два и более других. OF-CONFIG предполагает, что такие ресурсы как порты и очереди используются совместно несколькими логическими коммутаторами (OpenFlowLogicalSwitches), так что каждый такой коммутатор может иметь полный контроль над ресурсами, которые ему предоставлены.

В статье предлагается новый метод, где авторы разработали скрипт, который показывает размер очереди, номер условного потока, адреса и порты. В результате было доказано при применении виртуальных буферов, круговая задержка передачи пакета в узле уменьшается. А также производится экономное использование ресурсов памяти узла.

В дальнейшем, планируем написать плагин для контролера OpenDaylight, который будет задавать размер виртуальных буферов для каждого потока.

Литература / References

1. Vladyko A., Muthanna A., Kirichek R. Comprehensive SDN Testing Based on Model Network // Lecture Notes in Computer Science. 2016. Vol. 9870, pp. 539–549
2. Xia W., Wen Y., Foh C., Niyato D., Xie H. A Survey on Software-Defined Networking // IEEE Communications Surveys & Tutorials. 2015. Vol. 17. Iss. 1. pp. 27–51.
3. Liu L., Zhang D., Tsuritani T., et al. Field Trial of an OpenFlow-Based Unified Control Plane for Multilayer Multigranularity Optical Switching Networks // Journal of Lightwave Technology. 2013. Vol. 31. Iss. 4, pp. 506–514.
4. Alsmadi I. M., AlAzzam I., Akour M. A Systematic Literature Review on Software-Defined Networking // Studies in Computational Intelligence. 2017. Vol. 691. pp. 333–369.
5. Xiong B., Yang K., Zhao J., Li W., Li K. Performance Evaluation of OpenFlow-based Software-Defined Networks based on Queueing Model // Computer Networks. 2016. Vol. 102. pp. 174–183.
6. Jararweh Y., Al-Ayyoub M., Darabseh A., Benkhelifa E., Vouk M., Rindos A. SDIoT: A Software Defined Based Internet of Things Framework // Journal of Ambient Intelligence and Humanized Computing. 2015. Vol. 6. Iss. 4. pp. 453–461.
7. Kirichek R., Vladyko A., Paramonov A., Koucheryavy A. Software-Defined Architecture for Flying Ubiquitous Sensor Networking // 19th International Conference on Advanced Communication Technology (ICACT). 2017. pp. 158–162.
8. Kirichek R., Vladyko A., Zakharov M., Koucheryavy A. Model Networks for Internet of Things and SDN // 18th International Conference on Advanced Communication Technology (ICACT). 2016. pp. 76–79.
9. Vladyko A., Letenko I., Lezhepekov A., Buinevich M. Fuzzy Model of Dynamic Traffic Management in Software-Defined Mobile Networks // Lecture Notes in Computer Science. 2016. Vol. 9870. pp. 561–570
10. Xiong B., Peng X., Zhao J. A Concise Queueing Model for Controller Performance in Software-Defined Networks // Journal of Computers. 2016. Vol. 11. Iss. 3. pp. 232–237.
11. Ansell J., Seah W., Ng B., Marshall S. Making Queueing Theory More Palatable to SDN/OpenFlow-based Network Practitioners // IEEE/IFIP Network Operations and Management Symposium (NOMS). 2016. pp. 1119–1124.
12. Villy B. Iversen Teletraffic Engineering Handbook. COM Center Technical University of Denmark Building 343, DK-2800 Lyngby Tlf.: 4525 3648. URL: www.tele.dtu.dk/teletra

- Хакимов Абдукодир
Абдукаримович*** – магистр, СПбГУТ, Санкт-Петербург, 193232,
Российская Федерация, abdukadir94@mail.ru
- Мутханна Аммар Салех Али*** – кандидат технических наук, ассистент, СПбГУТ,
Санкт-Петербург, 193232, Российская Федерация,
ammarexpress@gmail.com
- Киричек Руслан Валентинович*** – кандидат технических наук, доцент, СПбГУТ,
Санкт-Петербург, 193232, Российская Федерация,
kirichek@sut.ru
- Khakimov Abdukadir*** – Undergraduate, SPbSUT, St. Petersburg, 193232,
Russian Federation, abdukadir94@mail.ru
- Muthanna Ammar*** – Candidate of Engineering Sciences, Assistant,
SPbSUT, St. Petersburg, 193232, Russian Federation,
ammarexpress@gmail.com
- Kirichek Ruslan*** – Candidate of Engineering Sciences, Associate
Professor, SPbSUT, St. Petersburg, 193232,
Russian Federation, kirichek@sut.ru