УДК 621.391

М.Б. Лурджан, А.А. Воропаева (канд. техн. наук), Г.В. Ступак
ГВУЗ «Донецкий национальный технический университет», г. Красноармейск
кафедра автоматики и телекоммуникаций
E-mail: massi.lourdjane@gmail.com; voropaeva_anna@meta.ua; s2p@list.ru

ИССЛЕДОВАНИЕ ВОЗМОЖНОСТЕЙ ВНЕДРЕНИЯ АЛГОРИТМОВ МАРШРУТИЗАЦИИ В ПРОГРАММНО-КОНФИГУРИРУЕМЫХ СЕТЯХ (SDN)

Рассмотрены возможности и механизмы тестирования и отладки, а также внедрения новых алгоритмов маршрутизации в сети, построенные согласно концепции Software Defined Networking. Разработан и внедрен новый алгоритм маршрутизации, который позволяет выбирать маршруты в зависимости от нескольких параметров и реализует свою метрику, основываясь на функциях полезности. Проведено экспериментальное сравнение нового протокола с одним из наиболее распространенных протоколов – OSPF. Получены результаты, показывающие превосходство нового протокола по ряду ключевых параметров. Модель SDN сети показала возможности эффективного управления инфраструктурой, открываемые данной концепцией.
***Ключевые слова**: протокол, внедрение, эффективность, программно-конфигурируемые сети, моделирование.*

Введение в концепцию SDN. Концепция программно-конфигурируемых сетей (SDN) призвана изменить представление и существенно повлиять на подход к построению сетей связи. Пристальное внимание к данной концепции в среде производителей оборудования и крупных игроков телекоммуникационного рынка вызвано целым рядом новых задач, требующих комплексного решения. Ежегодное увеличение объемов трафика, повышение цен а также усложнение сетевого оборудования, сложность имплементации, отладки и в конце концов создания новых алгоритмов управления сетью – все это приводит к замедлению развития бизнеса компаний, а также значительному увеличению затрат – как капитальных, так и операционных. Эффективность использования сетевых ресурсов и их стоимость приобретают все более весомое значение и выходят на первый план при проектировании и модернизации существующей инфраструктуры.

Основной идеей SDN является лишение маршрутизаторов возможности самостоятельно принимать решения по пересылке пакетов и передача этих полномочий контроллеру [1, 2]. Сегодня в зависимости от масштаба сети контроллер представляет собой сервер или группу серверов, на которых установлено специализированное программное обеспечение. На контроллере работает сетевое приложение, позволяющее с легкостью выражать требования к правилам и политике передачи данных, и создающее в зависимости от них логические топологии. Далее сетевая ОС переводит эти требования на язык, понятный SDN-коммутаторам и через специальный интерфейс связи между контроллером и сетевым устройством по протоколу (сейчас наиболее распространенный – OpenFlow) происходят необходимые записи в таблицы маршрутизации. Таким образом, произошло разделение двух плоскостей – данных и контроля.

На данный момент в сетях используется большое количество протоколов маршрутизации, которые отличаются своим типом, принципом работы, алгоритмом, на основе которого определяется оптимальный маршрут, сложностью и т.д. Их внедрение в сеть представляет собой достаточно длительный процесс настройки сетевого оборудования [3]. Далее каждое устройство на основе собранной информации от своих соседей о существующих подсетях

© Лурджан М.Б., Воропаева А.А., Ступак Г.В., 2015

составляет своего рода топологию сети и отдельно принимает решение об оптимальном маршруте для пересылки того или иного пакета, основываясь на алгоритме выбранного протокола.

SDN позволяет избавиться от этого барьера за счет переноса функций управления (маршрутизаторами, коммутаторами и т.п.) в сетевые приложения, которые работают в контроллере. Поэтому потребность введения нового протокола маршрутизации или модернизация существующего сводится к написанию определенного приложения – шаблона сети. Работая на контроллере, это приложение позволяет устанавливать правила пересылки в зависимости от любого количества переменных, не только стандартных задержки, джиттера и т.д., но и от нагрузки на конкретный участок сети, от времени суток и многих других параметров.

Протокол QoS маршрутизации. Для иллюстрации возможностей оптимизации параметров сети с использованием концепции SDN, авторами предлагается создать новый протокол маршрутизации, который будет отвечать вызовам телекоммуникационного провайдера и сравнить результаты работы этого нового протокола с одним из самых широко распространенных на сегодняшний день – OSPF, использующий алгоритм Дейкстры применительно к пропускной способности каналов как метрике [4]. Данная метрика учитывает только один параметр, что не достаточно для некоторых операторов связи.

Разработанный протокол работает на основе алгоритма Дейкстры, однако использует комплексную метрику на основе функций полезности. Данный подход позволяет учитывать тип трафика (VoIP, Video, Internet) и, в зависимости от него, рассчитывать собственно функцию полезности (таблица 1), зная показатели задержки передачи, пропускной способности и судьбы потерь пакетов. В таблице 1 использованы следующие обозначения:

$B_{\min}$ – минимально допустимая пропускная способность;

$B_{\max}$ – максимальная требуемая пропускная способность;

$\tau_{\max}$ – максимальная задержка по рекомендации ITU-T;

$\tau_{\min}$ – значение задержки, в пределах которой сохраняется наилучшее качество обслуживания – значение функции полезности близко к максимальному;

$L_{\max}$ – максимальное значение потер пакетов, ниже которого сохраняется допустимый уровень качества роботы кодека;

L_c – критическое значение потер пакетов, при достижении которого передача данных становится невозможной.

Таблица 1 – Функции полезности

	VoIP	Internet	Video
Проп спос.	$u_{VoIP}(b)=\frac{\operatorname{sgn}(b-B_{\min})+1}{2}$	$u_{FT}(b)=\frac{\log(b+1)}{\log(B_{\max}+1)}$	$u_{video}(b)=\frac{1}{1+\left(\frac{1}{\varepsilon}-1\right)e^{-rb}}$
За-держ-ка	$u_{VoIP}(\tau)=1-\frac{1}{1+\left(\frac{\tau_{\max}}{\tau_{\min}}-1\right)\cdot e^{-r\tau}}$	$u_{FT}(\tau)=1$	$u_{video}(\tau)=1-\frac{1}{1+\left(\frac{\tau_{\max}}{\tau_{mim}}-1\right)e^{-r\tau}}$
Доля по-терь	$u_{VoIP}(L)=(L_{\max})^{3L}$	$u_{FT}(L)=\begin{cases}1, \text{ если } L\le L_{\max}\\ \frac{L-L_c}{L_{\max}-L_c}, \text{если } L_{\max}<L<L_c\\ 0, \text{ если } L>L_c\end{cases}$	$u_{video}(L)=(L_{\max})^{3L}$

Каждый тип трафика характеризуется своими требованиями к параметрам каналов связи. Если голосовой трафик требователен к задержке и ее отклонениям, данные - к искажени-

ям и потерям пакетов [5]. Поэтому для каждого типа передаваемых данных проведен анализ характеристик и требований к каналу связи. Для количественной оценки параметров передачи данных целесообразно ввести функции полезности. Данные функции - это зависимость количественной нормируемой оценки от определенного параметра канала связи. Другими словами, значение функции характеризуют приемлемость канала связи для передачи этого типа трафика в данный момент. Таким образом, данная метрика позволяет не только строить маршруты в зависимости от требований конкретного типа трафик, но и учитывать качество на стороне каждого абонента, выделяя им в зависимости от типа трафика необходимую пропускную способность. При этом:

$$r(x) = \frac{2\ln\left(\frac{x_{\max}}{x_{\min}} - 1\right)}{x_{\max}}.$$

На основе приведенных формул рассчитывается метрика для каждой существующей дуги заданной топологии:

$$K_x = 1 - u_x(b)\,u_x(\tau)\,u_x(L), \tag{1}$$

где x – тип передаваемого трафика (VoIP, video, FT).

Таким образом, новая предложенная метрика потенциально значительно улучшает показатели эффективности использования сети и позволяет осуществлять QoS маршрутизацию.

Данный алгоритм маршрутизации был реализован в виде приложения для контроллера, написанного на языке C#. Данное приложение согласно введенной топологии сети и новой метрике, в зависимости от вида трафика выстраивает маршруты, которые в последствии передаются OpenFlow контроллеру [6]. Выражая, таким образом, требования к политике передачи данных сети, контроллер передает их коммутаторам, реализуя тем самым концепцию выноса плоскости контроля из каждого конкретного устройства (см. рис. 1).

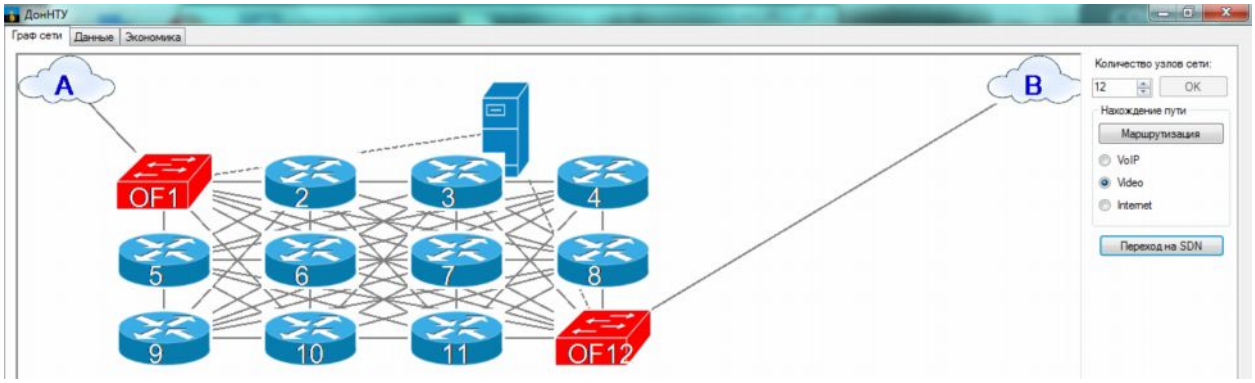

Рисунок 1 – Разработаный программный пакет

Модель сети. Эмуляция заданного участка сети производилась в виртуальной среде Mininet, где в качестве узлов используются программные коммутаторы Open vSwitch, которые работают под управлением контроллера OpenDaylight [7].

Mininet - это эмулятор компьютерной сети. Под компьютерной сетью подразумеваются простые компьютеры - хосты, коммутаторы, а так же OpenFlow-контроллеры. С помощью простейшего синтаксиса в интерпретаторе команд можно разворачивать сети из произвольного количества хостов, коммутаторов в различных топологиях в рамках одной машины [8].

Принцип работы Mininet основан на том, что начиная с версии 2.6.24, ядром Linux под-

держиваются механизмы виртуализации и изоляции – Cgroups, которые позволяют обеспечить сетевыми интерфейсами, таблицами маршрутизации и ARP-таблиами процессы в рамках одной операционной системы. Это - один из видов виртуализации на уровне ОС, позволяющий запустить множество однотипных процессов в изолированном и ограниченном по ресурсам окружении. Подобные техники позволяют Mininet создавать в пространстве ядра или пользователя коммутаторы, OpenFlow-контроллеры и хосты, и взаимодействовать в рамках моделируемой сети. В качестве виртуальных коммутаторов используется адаптированная реализация Open vSwitch. Основная функциональность Mininet реализована на языке Python, за исключением некоторых утилит написанных на языке C. Практически любая произвольная топология может быть описана с помощью специального синтаксиса на Python.

Open Switch – программный многоуровневый коммутатор с открытым исходным текстом, предназначенный для работы в гипервизорах и на компьютерах с виртуальными машинами. Для управления сети использовался контроллер OpenDaylight. Он является проектом с открытым исходным кодом с модульной, заменяемой и гибкой платформой контроллера [9]. Этот контроллер реализован строго программно и расположен в виртуальной машине Java (JVM). Таким образом, он может быть развернут на любой системе, которая поддерживает Java. Сам контроллер состоит из набора динамически подключаемых модулей для обеспечения требуемых сетевых потребностей. Модули, при необходимости, могут быть разработаны самостоятельно. Нужно отметить, что в связке с Mininet может быть использован любой другой контроллер.

Для нагрузки и оценки виртуальной сети использовалась D-ITG (Distributed Internet Traffic Generator) – программная платформа, способная генерировать IPv4 и IPv6 трафик за счет точного тиражирования нагрузки интернет-приложений [10]. В то же время, D-ITG является сетевым измерительным приложением, которое в состоянии измерить наиболее распространённые показатели производительности (например, пропускная способность, задержка, джиттер, потери пакетов) на пакетном уровне.

D-ITG может генерировать трафик следуя стохастической модели для размера пакетов (PS) и межпакетным интервалом (IDT), что позволяет имитировать поведение протоколов прикладного уровня. Используя характеристику и результаты моделирования, D-ITG способен повторить статистические свойства трафика различных известных приложений (например, Telnet, VoIP - G.711, G.723, G.729, Voice Activity Detection, сжатого RTP - DNS, сетевые игры). На транспортном уровне, D-ITG поддерживает TCP (Transmission Control Protocol), UDP (User Datagram Protocol), SCTP1 (Stream Control Transmission Protocol) и DCCP1 (Datagram Congestion Control Protocol). Также имеется поддержка ICMP (Internet Protocol Control Message). Имеется воможность установки полей TOS (DS) и TTL IP заголовка.

Для проверки показателей виртуализированной сети, с помощью D-ITG был сгенерирован трафик такого рода: 50потоковVoIP G.711.1 (~ 3.2Mbps); IPTV(~ 93.7Mbps); Internet-трафик (~ 82.5Mbps) (см. рис. 2). Объем трафика был ограничен производительностью тестового стенда.

Результаты моделирования показали работоспособность разработанных средств. Благодаря реализованным средствам происходит более эффективное перераспределение ресурсов транспортной сети с учетом нелинейной зависимости удовлетворения абонентов сети от параметров качества обслуживания мультисервисного трафика. Благодаря предложенной метрике были учтены все параметры QoS, в отличие от существующих алгоритмов маршрутизации. Для определения эффективности предлагаемого решения был проведен сравнительный анализ нашего алгоритма и одного из классических алгоритмов (OSPF), который осуществляет маршрутизацию на основе только одного параметра – пропускной способности. Результаты приведены на рисунке 3.

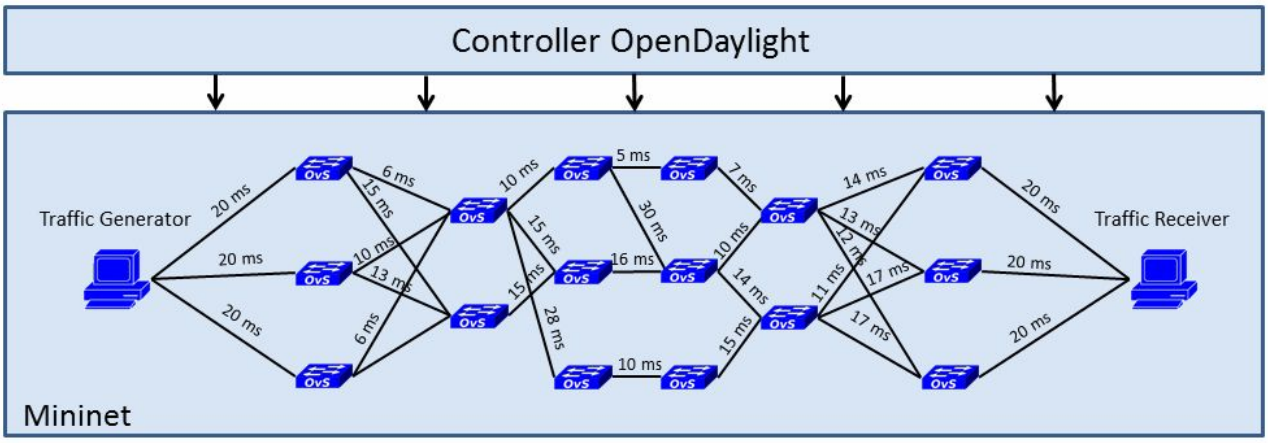

Рисунок 2 – Имитационная модель сегмента SDN-сети

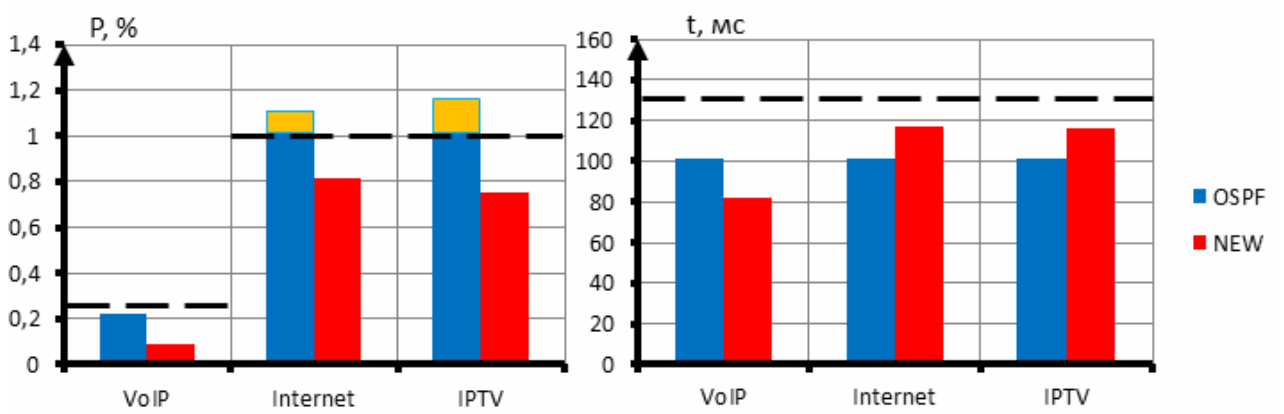

Рисунок 3 – Результаты имитационного моделирования

Диаграмма характеристик задержки для каждого вида трафика показывает уменьшение значения задержки для чувствительного к ней VoIP, и наблюдается незначительное увеличение задержек в других, менее чувствительных к ней, видах трафика. В то же время, незначительно увеличив таким образом задержку, но учтя в новой метрике другие параметры, можно получить значительное уменьшение вероятности потери пакета, которые при использовании протокола OSPF, предполагающего что весь трафик будет проходить через один путь, превышают норму.

Выводы.

Новые типы данных выдвигают новые требования к методам маршрутизации и управления трафиком в современных сетях. В традиционных сетях реализация концепции раздельной маршрутизации невозможна, но с внедрением технологии SDN эта проблема перестает существовать как таковая, при этом появляются дополнительные возможности для управления нагрузкой и ее балансировкой. В данной статье были рассмотрены и наглядно показаны возможности управления трафиком в сети SDN, создан новый алгоритм QoS маршрутизации, создан программный пакет и собрана модель сети, позволяющая реализовать данный алгоритм, проверить его работоспособность и сравнить его с уже существующим. Результаты моделирования показали улучшение основных показателей QoS для каждого вида трафика.

Список использованной литературы

1. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2013–2018:

Cisco® February 5, 2014 [Електронний ресурс] // Режим доступу: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html / Дата доступу 12.02.2015.
2. The Road to SDN: An Intellectual Historyof Programmable Networks: Nick Feamster, Jennifer Rexford, Ellen Zegura [Електронний ресурс] // Режим доступу: http://www.sigcomm.org/node/3488 / Дата доступу 15.02.2015.
3. OpenVirteX – A Network Hypervisor : Open Networking Lab [Електронний ресурс] // Режим доступу: http://www.slideshare.net/ON_LAB/demo-slides-v05-slideshare-29372103 / Дата доступу 21.02.2015.
4. The Controller Placement Problem: Brandon Heller, Rob Sherwood, Nick McKeown [Електронний ресурс] // Режим доступу: http://yuba.stanford.edu/~nickm/papers/hot21-heller.pdf / Дата доступу 21.02.2015.
5. OpenFlow Switch Specication: Open Networking Foundation June 25, 2012 [Електронний ресурс] // Режим доступу: http://archive.openflow.org/documents/openflow-spec-v1.1.0.pdf / Дата доступу 15.02.2015.
6. DISCO: Distributed Multi-domain SDN Controllers: Kevin Phemius, Mathieu Bouet and Jeremie Leguay [Електронний ресурс] // Режим доступу: http://arxiv.org/pdf/1308.6138.pdf / Дата доступу 02.03.2015.
7. FlowVisor: A Network Virtualization Layer: Deutsche Telekom Inc. R&D Lab, y Stanford University, _ Nicira Networks October 14, 2009 /Rob Sherwood, Glen Gibb, Kok-Kiong Yap [Електронний ресурс] // Режим доступу: https://sb.tmit.bme.hu/mediawiki/images/c/c0/FlowVisor.pdf / Дата доступу 15.02.2015.
8. Analysis of CAPEX and OPEX Benefits of Wireless Access Virtualization: *the 4th Workshop on E2Nets, Budapest, Hungary, June 9, 2013 /* M.M.Rahman, Charles Despins. [Електронний ресурс] // Режим доступу: http://e2nets.org/docs/Rahman-Presentation.pdf / Дата доступу 15.02.2015.
9. Pac C. A Unified Control Crchitecture for Packet and Circuit Network Convergence: Saurav Das June 2012 [Електронний ресурс] // Режим доступу: http://yuba.stanford.edu/~sd2/Thesis_Saurav_Das.pdf / Дата доступу 13.02.2015.
10. LTE Wireless Virtualization and Spectrum Management / Yasir Zaki, Liang Zhao, Carmelita Goerg [Електронний ресурс] // Режим доступу: http://www.cse.wustl.edu/~jain/cse574-14/ftp/virwn/index.html / Дата доступу 19.02.2015.

References

1. Cisco Visual Networking Index (2013). "Global Mobile Data Traffic Forecast Update, 2013–2018: Cisco®", available at: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html (Accessed February 5, 2014).
2. The Road to SDN. "An Intellectual Historyof Programmable Networks: Nick Feamster, Jennifer Rexford, Ellen Zegura", available at: http://www.sigcomm.org/node/3488 (Accessed February 15 2015).
3. OpenVirteX – A Network Hypervisor (2012). "Open Networking Lab", available at: http://www.slideshare.net/ON_LAB/demo-slides-v05-slideshare-29372103 (Accessed February 21 2015).
4. The McKeon Group (2012). "The Controller Placement Problem Brandon Heller, Rob Sherwood, Nick McKeown", available at: http://yuba.stanford.edu/~nickm/papers/hot21-heller.pdf (Accessed February 21 2015).
5. OpenFlow Switch Specication (2012). "Open Networking Foundation June 25, 2012", available at: http://archive.openflow.org/documents/openflow-spec-v1.1.0.pdf (Accessed February 15 2015).
6. DISCO (2013). "Distributed Multi-domain SDN Controllers: Kevin Phemius, Mathieu Bouet and Jeremie Leguay", available at: http://arxiv.org/pdf/1308.6138.pdf (Accessed March 2 2015).

7. FlowVisor (2009). "A Network Virtualization Layer: Deutsche Telekom Inc. R&D Lab, y Stanford University, _ Nicira Networks October 14, 2009 /Rob Sherwood, Glen Gibb, Kok-Kiong Yap", available at: https://sb.tmit.bme.hu/mediawiki/images/c/c0/FlowVisor.pdf (Accessed February 15 2015).
8. Energy to Nets (E2Nets) (2013) "Analysis of CAPEX and OPEX Benefits of Wireless Access Virtualization: *the 4th Workshop on E2Nets, Budapest, Hungary, June 9, 2013 /* M.M.Rahman, Charles Despins", available at: http://e2nets.org/docs/Rahman-Presentation.pdf (Accessed February 15 2015).
9. The McKeon Group (2012) "Pac C. A Unified Control Crchitecture for Packet and Circuit Network Convergence: Saurav Das June 2012", available at: http://yuba.stanford.edu/~sd2/Thesis_Saurav_Das.pdf (Accessed March 13 2015).
10. Virtualization inWireless Networks (2011) "LTE Wireless Virtualization and Spectrum Management / Yasir Zaki, Liang Zhao, Carmelita Goerg", available at: http://www.cse.wustl.edu/~jain/cse574-14/ftp/virwn/index.html (Accessed March 10 2015).

Поступила в редакцию:
30.03.2015

Рецензент:
д-р техн. наук, проф. С.В. Подкопаєв

М.Б. Лурджан, А.О. Воропаєва, Г.В. Ступак
ДВНЗ «Донецький національний технічний університет»

Дослідження можливостей впровадження алгоритмів маршрутизації у програмно-конфігурованих мережах (SDN). *Розглянуто можливості та механізми тестування і налагодження, а також впровадження нових алгоритмів маршрутизації в мережі, побудовані відповідно до концепції Software Defined Networking. Розроблено та впроваджено новий алгоритм маршрутизації, який дозволяє вибирати маршрути в залежності від декількох параметрів і реалізує свою метрику грунтуючись на функціях корисності. Проведено експериментальне порівняння нового протоколу з одним з найбільш поширених протоколів - OSPF. Отримано результати, що показують перевагу нового протоколу по ряду ключових параметрів. Зібрана модель SDN мережі показала можливості ефективного управління інфраструктурою, що відкриваються даною концепцією.*

Ключові слова: *протокол, впровадження ефективності, програмно-конфігуровні мережі, моделювання.*

M. Lourdjane, A. Voropaeva, G. Stupak
Donetsk National Technical University

Introduction of New Routing Algorithms in Software Defined Networks (SDN). *Software defined networking concept is already changing the idea and has a significant impact on the process of building and managing networks. The efficiency of usage of network resources and their price are getting more and more important which makes those issues one of the largest challenges, when design and modernization takes place. There is a large amount of protocols that are being currently used across the infrastructure and that have some difference in their routing algorithm, type, complexity, etc. But something always remained the same – they collect the information about topology and separately take routing decisions. SDN take approach to distinguish two in fact very different planes – data plane and control plane, making the network more programmable and easy to use, reducing by the way the cost of network equipment. The software controller would run applications, that should have some kind of user friendly interface, giving the engineers abilities of policy and rules implementation. Using the OpenFlow (or any other SDN-based protocol) interface, this controller connects to network devices, creates virtual topology and, based on policies, gives those devices the exact match for routing tables forwarding actions. Introducing and managing a*

new routing protocol becomes really fast and efficient with SDN. In this article we want to present the results of our work: there has been developed and implemented a NEW routing algorithm, that allows choosing the best route based on the utility functions and that has a complex multy-parameters metrics. It gives the provider a way to take approach to QoS based routing, keeping in mind that each type of traffic(in this particular case – WEB, Video and VoIP) has its own bandwith, delay and jitter and loss requirments. The model of SDN network has been configured using Mininet running on Linux core, OpenDaylight controller, D-ITG traffic generator and our developed application, that implemented the routing rules using our algorithm. Based on this model and chosen topology, we have made comparison between QoS parameters of our protocol and well known OSPF. Furthermore, our model exactly matches the real network requirements, meaning that all algorithms, developed applications and controller configurations could be used in the real networks without making any change. Shown results prove efficiency of our protocol, as well as ability to fast implementation of new features in SDN networks.
***Keywords**: SDN, Openflow, Protocol, Mininet, Modeling.*

Лурджан Масиниса Брахимович, бакалавр телекоммуникационных систем и сетей, студент магистр Донецкого национального технического университета, кафедры автоматики и телекоммуникаций, ГВУЗ «Донецкий национальный технический университет» (пл. Шибанкова, 2, г. Красноармейск, 85300, Украина). Основное направление научной деятельности – исследование и модернизация существующих вариаций создания компьютерных сетей, SDN, мобильные сети, протоколы маршрутизации, балансировка нагрузки.

Воропаева Анна Александровна, Украина, окончила Донецкий национальный технический университет, канд. техн. наук, доцент кафедры автоматики и телекоммуникаций, ГВУЗ «Донецкий национальный технический университет» (пл. Шибанкова, 2, г. Красноармейск, 85300, Украина). Основное направление научной деятельности – методы исследования и оптимизации беспроводных телекоммуникационных сетей.

Ступак Глеб Владимирович, Украина, окончил Донецкий национальный технический университет, старший преподаватель кафедры автоматики и телекоммуникаций, ГВУЗ «Донецкий национальный технический университет» (пл. Шибанкова, 2, г. Красноармейск, 85300, Украина). Основное направление научной деятельности – разработка алгоритмов обслуживания вызовов.