

УДК 004.72

doi: 10.21685/2072-3059-2024-3-2

Моделирование коммутатора программно-конфигурируемой сети с поддержкой протокола OpenFlow в среде AnyLogic

И. А. Филиппов¹, В. А. Волков², К. И. Никишин³, Е. И. Гурин⁴

^{1,2,3,4}Пензенский государственный университет, Пенза, Россия

¹ilya_filippov_2002@inbox.ru, ²valek.volkov.2001@mail.ru,

³nkipnz@mail.ru, ⁴gurin2@yandex.ru

Аннотация. *Актуальность и цели.* Малая гибкость и управляемость программного управления сетевым оборудованием – одна из проблем современных сетей. С целью решения этой проблемы прибегают к использованию программно-конфигурируемых сетей (ПКС), основным компонентом которых является коммутатор, поддерживающий протокол *OpenFlow*. Цель исследования заключается в моделировании коммутатора ПКС с поддержкой протокола *OpenFlow* в среде моделирования *AnyLogic*, для того чтобы получить характеристики передачи трафика в ПКС, экспериментально их оценить. *Материалы и методы.* Рассмотрена имитационная модель коммутатора ПКС в среде моделирования *AnyLogic*. Кратко описаны основные компоненты коммутатора, такие как таблицы потоков, правила действий над пакетом на входном и выходном портах коммутатора, управляющий процессор, очереди коммутатора. Отлажена работа схемы с коммутатором ПКС и поддержкой протокола *OpenFlow*. Произведены тесты, замерены данные, описаны преимущества представленной модели. *Результаты.* В рамках исследования проведен анализ процессов управления трафиком и конфигурации сетевого оборудования с использованием протокола *OpenFlow*, построена модель и выявлены потенциальные области применения и перспективы развития данной модели в ПКС. Полученные результаты могут быть использованы для оптимизации работы ПКС и разработки эффективных методов управления сетевым трафиком. *Выводы.* Была рассмотрена модель коммутатора ПКС с поддержкой протокола *OpenFlow*. Выявлены положительные и отрицательные моменты работы модели с использованием протокола *OpenFlow* в ПКС, а также продемонстрировано решение об увеличении пропускной способности сети для повышения эффективности работы модели.

Ключевые слова: протокол *OpenFlow*, ПКС, архитектура ПКС, *AnyLogic*, автоматизация сети, принцип разделения сети

Для цитирования: Филиппов И. А., Волков В. А., Никишин К. И., Гурин Е. И. Моделирование коммутатора программно-конфигурируемой сети с поддержкой протокола OpenFlow в среде AnyLogic // Известия высших учебных заведений. Поволжский регион. Технические науки. 2024. № 3. С. 14–24. doi: 10.21685/2072-3059-2024-3-2

Modelling of SDN switch with OpenFlow protocol in AnyLogic

I.A. Filippov¹, V.A. Volkov², K.I. Nikishin³, E.I. Gurin⁴

^{1,2,3,4}Penza State University, Penza, Russia

¹ilya_filippov_2002@inbox.ru, ²valek.volkov.2001@mail.ru,

³nkipnz@mail.ru, ⁴gurin2@yandex.ru

© Филиппов И. А., Волков В. А., Никишин К. И., Гурин Е. И., 2024. Контент доступен по лицензии Creative Commons Attribution 4.0 License / This work is licensed under a Creative Commons Attribution 4.0 License.

Abstract. *Background.* Low flexibility and manageability of software control of network equipment is one of the problems of modern networks. In order to solve this problem, software-defined networks (SDN) are used, the main component of which is switch with OpenFlow protocol. The purpose of the study is to model a SDN switch with OpenFlow protocol in AnyLogic, in order to obtain the transmission characteristics of SDN, experimentally evaluate these characteristics. *Materials and methods.* The article considers the simulation model of SDN switch in AnyLogic. The main components of the switch, such as flow tables, rules of packet actions on the input and output ports of switch, control processor, switch queues are briefly described. The work of the scheme with SDN switch and OpenFlow protocol is debugged. Tests are performed, data are measured, and the advantages of the presented model are described. *Results.* The research analyses the processes of traffic management and network equipment configuration using OpenFlow protocol, builds a model and identifies potential areas of application and prospects for the development of this model in SDN. The results obtained can be used for optimization of SDN operation and development of effective methods of network traffic management. *Conclusion.* The model of SDN switch with OpenFlow protocol was considered. Positive and negative aspects of the model using OpenFlow protocol in SDN were revealed, and the decision to increase the network bandwidth to improve the efficiency of the model was demonstrated.

Keywords: OpenFlow protocol, SDN, SDN architecture, AnyLogic, network automation, network partitioning principle

For citation: Filippov I.A., Volkov V.A., Nikishin K.I., Gurin E.I. Modelling of SDN switch with OpenFlow protocol in AnyLogic. *Izvestiya vysshikh uchebnykh zavedeniy. Volzhskiy region. Tekhnicheskie nauki* = *University proceedings. Volga region. Engineering sciences*. 2024;(3):14–24. (In Russ.). doi: 10.21685/2072-3059-2024-3-2

Введение

В настоящее время существует большой интерес к технологиям программно-конфигурируемых сетей (ПКС), которые предоставляют возможности для гибкого управления сетевым трафиком [1]. Одной из ключевых технологий, способствующих развитию ПКС, является протокол *OpenFlow*, который используется для управления сетевой инфраструктурой.

Основная идея *OpenFlow* заключается в том, что принятие решений о передаче трафика в сети происходит централизованно в рамках контроллера, который управляет работой коммутаторов посредством управляющих сообщений. Это отличается от классического подхода, где интеллектуальные решения о маршрутизации и управлении потоками принимаются распределенно на уровне каждого отдельного коммутатора.

Применение коммутаторов *OpenFlow* в контексте современных сетевых сред приносит ряд значительных преимуществ, таких как повышение гибкости сетевой инфраструктуры, улучшение масштабируемости, оптимизация управления трафиком и более эффективное реагирование на динамические требования приложений и услуг.

Таким образом, внедрение коммутаторов с поддержкой протокола *OpenFlow* в современные сети представляет собой актуальное направление для сетевых инженеров и специалистов по управлению сетями, требующего комплексного изучения и проектирования для достижения оптимальных параметров в различных конфигурациях сетей. В связи с этим в статье рассматриваются коммутаторы с поддержкой протокола *OpenFlow* в ПКС, их основные принципы работы, преимущества и возможные области применения.

Основные компоненты коммутатора ПКС с поддержкой протокола *OpenFlow*

В ходе исследования необходимо было изучить поведение сложного телекоммуникационного оборудования и получить параметры передачи трафика. Это становится возможным благодаря только имитационному моделированию. Моделирование позволяет предсказывать поведение системы в различных условиях и оценивать потенциальные риски и возможности для улучшения работоспособности системы.

Коммутатор с поддержкой протокола *OpenFlow* представляет собой ключевой элемент ПКС, который используется для управления трафиком в сетевых устройствах [2–4]. Достоинства коммутаторов с поддержкой протокола *OpenFlow* включают гибкость и возможность централизованного управления, что позволяет администраторам эффективно контролировать сетевые ресурсы и применять политики трафика. Также благодаря открытому протоколу *OpenFlow* коммутаторы становятся совместимыми с различными устройствами и вендорами, обеспечивая свободу выбора оборудования. Важным преимуществом является возможность программного управления трафиком, что упрощает внедрение новых сервисов и изменения в сети [5–7].

Однако у коммутаторов, поддерживающих протокол *OpenFlow*, имеются и недостатки. Например, централизованное управление может привести к уязвимостям и задержкам в обработке трафика. Кроме того, для эффективной работы *OpenFlow* требуется высокий уровень навыков управления сетями и обеспечения безопасности, что может создать дополнительные сложности для администраторов.

В связи с этими особенностями необходимо было разработать модель коммутатора ПКС с поддержкой *OpenFlow* в среде *AnyLogic*. Данная среда предлагает широкий спектр возможностей для создания мультипарадигменных моделей, включая дискретно-событийное моделирование, системную динамику и агентное моделирование. *AnyLogic* также обладает интуитивно понятным интерфейсом, имеет обширную библиотеку компонентов и предоставляет возможности интеграции с различными источниками данных, что делает ее привлекательным выбором для моделирования сложных систем и процессов [8, 9].

В составе коммутатора ПКС можно выделить 4 основных узла: таблицы потоков (нахождение и сверка правил поступившего пакета), узел анализа прописанных в правилах действий по отношению к пакету на входе (сопоставление найденного правила соответствующему действию, выполнение прописанного действия), управляющий процессор (анализ поступившего пакета, обнаружение новых устройств для отправки пакета), узел анализа прописанных в правилах действий по отношению к пакету на выходе (подготовка и распределение пакетов перед отправкой) [10, 11]. Основными функциями коммутатора ПКС являются обработка, проверка и передача данных в сети (рис. 1).

Построение входных узлов коммутатора ПКС с поддержкой протокола *OpenFlow*

Работа коммутатора с поддержкой протокола *OpenFlow* исследована на основе модели ПКС сети в среде *AnyLogic*. Представленная модель показыва-

ет работу основных узлов, которые будут описываться далее, и динамику передачи пакетов от входного порта до его отправки. Рассмотрим первый узел модели, который представлен на рис. 2.

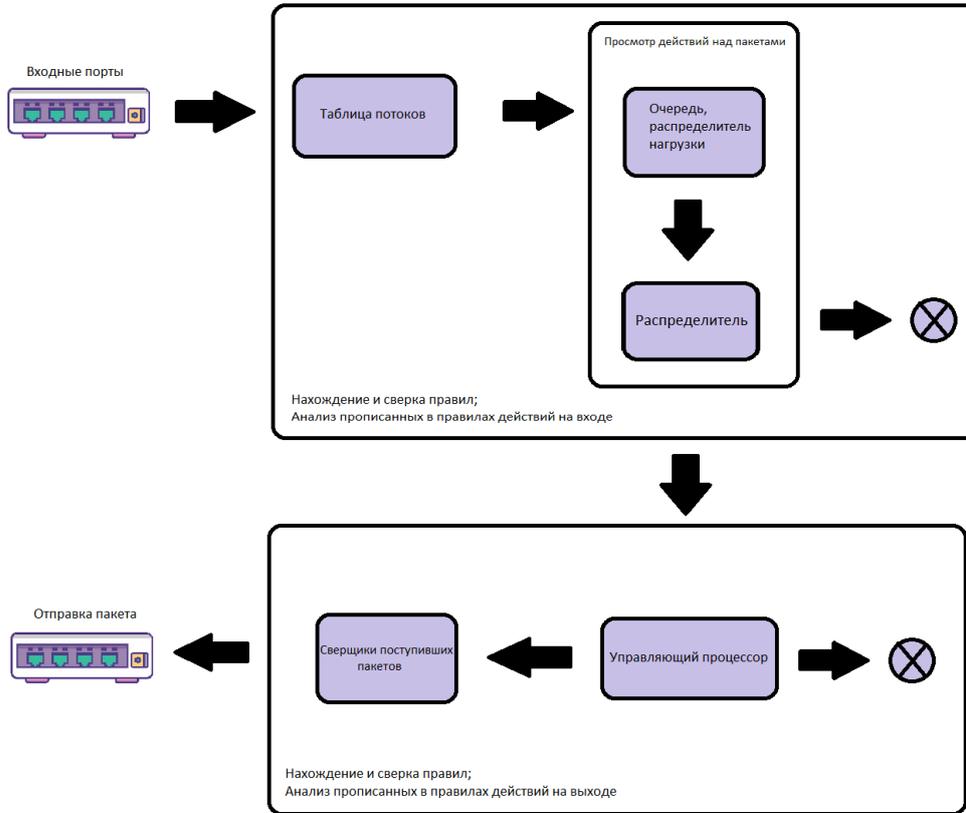


Рис. 1. Компоненты коммутатора, поддерживающего протокол *OpenFlow*

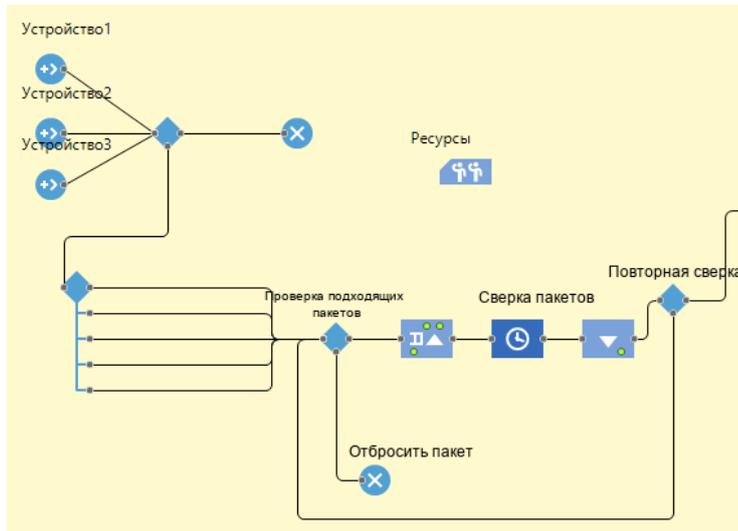


Рис. 2. Узел таблицы потоков

Когда пакет данных поступает на вход коммутатора или маршрутизатора, он проходит через правила обработки трафика. Коммутатор ищет правило, соответствующее пакету, на основе полей заголовка пакета, таких как исходный и целевой IP-адреса, порты и т.д. В начале узла стоит распределитель потока для предотвращения переполнения фрагмента сверки правил. Далее идет блок сверки правил, который состоит из узла, отбрасывающего пакеты, не подходящие под эти правила. После чего идет сверка указанных правил, и если в одном из них указано возвращение на повторную сверку, то данный пакет проходит этот этап повторно.

Далее идет узел просмотра действий над пакетом. Основной принцип данного узла заключается в следующем: если правило найдено, коммутатор применяет действия, указанные в правиле. Сам узел представлен на рис. 3. На нем представлены очередь и распределитель нагрузки, необходимые для предотвращения переполнения на участке сверки действий. Далее установлен распределитель, если в правиле считано действие передачи пакета на выход, то он проходит дальше, иначе отбрасывается.

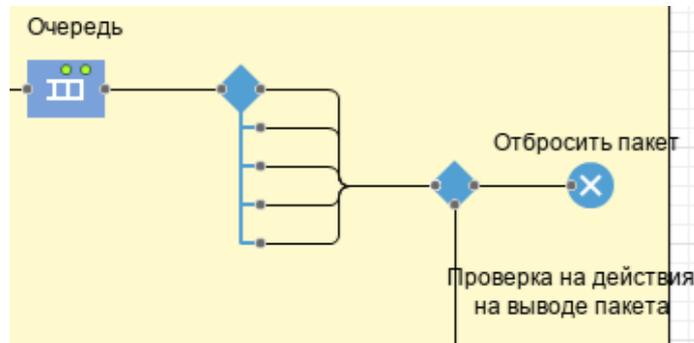


Рис. 3. Узел проверки действий над пакетами на входе коммутатора

После прохождения участка обработки на входе коммутатора пакеты проходят дальше, подготавливаясь к передаче через выходной порт, описание моделирования представлено дальше.

Построение выходных узлов коммутатора ПКС с поддержкой протокола *OpenFlow*

Пакету после прохождения предписанных действий больше нет необходимости находиться в предыдущем узле, и он перенаправляется в узел управляющего процессора. Участок данного узла представлен на рис. 4.

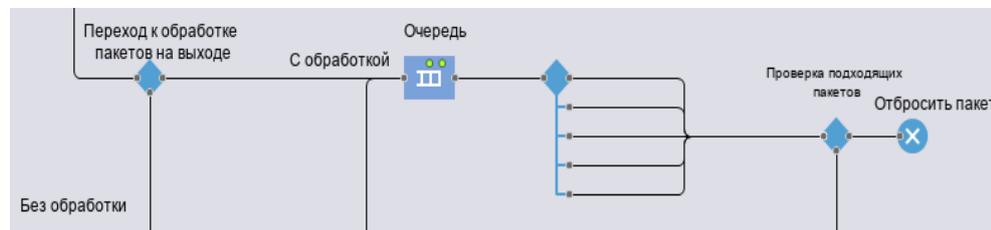


Рис. 4. Узел управляющего процессора

Когда управляющий процессор находит соответствующее правило, он выполняет определенные действия, которые установлены в правиле. Если правило для пакета отсутствует или в правиле указано отправить пакет на контроллер *OpenFlow*, такой пакет перенаправляется на централизованный контроллер, где происходит принятие решения об обработке пакета.

Сам узел, представленный на рис. 4, состоит из сверщика пакета, который определяет необходимость обработки пакета. Если в правиле пакета установлена отправка без обработки, то такой пакет передается непосредственно на отправку. Иначе пакет отправляется на дальнейшую обработку, в которой предусмотрена защита от переполнения на участке сверки с помощью очередей и распределителя нагрузки. Подходящий пакет передается в узел проверки действий над пакетами. Данный узел представлен на рис. 5.



Рис. 5. Узел проверки действий над пакетами

Узел проверки действий над пакетом может определить направление передачи в выходной порт коммутатора. На рис. 5 установлены два сверщика. Первый проверяет пакет на наличие действия в правиле, в случае отсутствия действия отправляет его на повторную сверку правил. Второй сверщик проверяет конкретно действие и доводит его на отправку.

Концептуальная модель коммутатора, поддерживающего протокол *OpenFlow*, построена с целью передачи пакетов в ПКС (рис. 6). Сетевая модель, основанная на *OpenFlow* (часть архитектуры ПКС), отделяет управление от пересылки трафика. В этой модели коммутаторы выполняют инструкции, которые были заданы контроллером *OpenFlow*. Контроллер представляет централизованное управление сетью, позволяя администраторам программно определять и контролировать потоки данных, управлять пропускной способностью, изменять маршрутизацию и применять политики без необходимости вручную настраивать каждое сетевое устройство в сети.

Результаты

В процессе моделирования пакеты поступают на входные порты коммутатора. На момент тестирования было зафиксировано поступление 40 пакетов/с при пропускной способности 65 пакетов/с. Интенсивность поступления пакетов была подобрана таким образом, чтобы обеспечить умеренную плотность потока и отобразить работу всех узлов под нагрузкой. В случае необходимости увеличения количества поступающих пакетов требуется увеличить пропускную способность на узле обработки правил пакета.

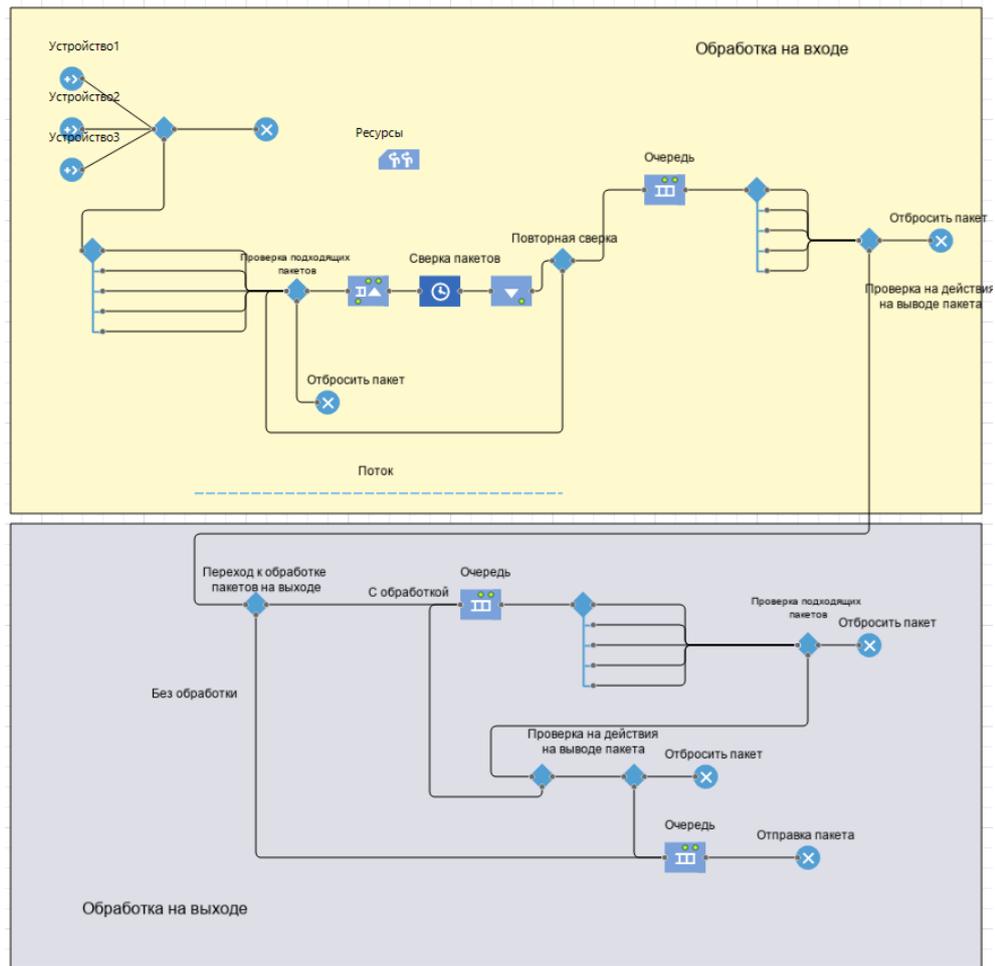


Рис. 6. Модель коммутатора ПКС с поддержкой *OpenFlow*

В первом случае передавался поток данных, равный 40 пакетам/с, что соответствовало корректной работе модели. Во втором случае пропускная способность была уменьшена без изменения входящего потока, что привело к тому, что узел обработки не успевал сверять правила, вызывая переполнение. Графики обработки пакетов показаны на рис. 7, 8.

На рис. 7 показан полностью прошедший поток данных через коммутатор, не вызывающий переполнения. По графику в конце работы модели видно, что пакеты проходят своевременную обработку в течение 0,05 с не перегружая работу модели. На следующем эксперименте было показано переполнение в коммутаторе за счет уменьшения ресурсов (рис. 8).

На рис. 8 демонстрируется, что поток не успевает обрабатываться и вызывает переполнение. Путем исследования было выяснено, что решением данной проблемы является увеличение ресурсов. Также в процессе исследования видно, как идет обработка пакетов в разных узлах смоделированной схемы (рис. 9). Из рис. 9 видно, что пакеты идут через каждый узел, проходя полную обработку.

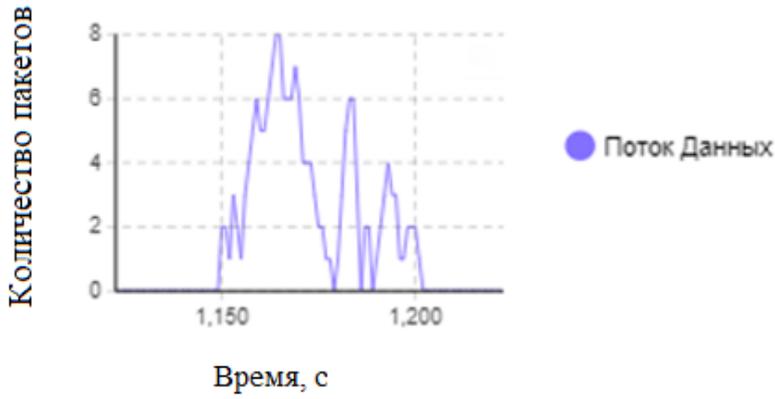


Рис. 7. График обработки пакетов в начале работы модели

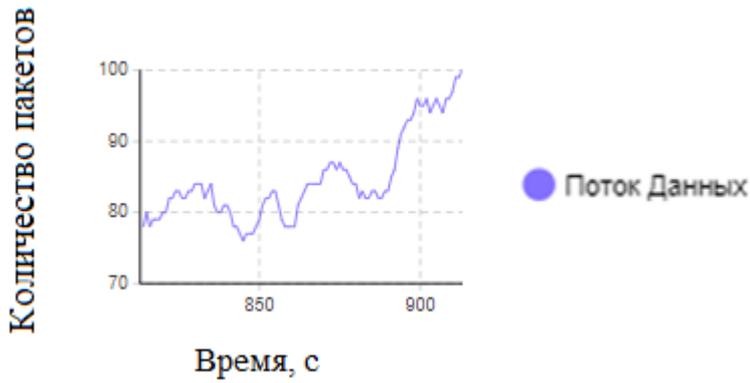


Рис. 8. График обработки пакетов в конце работы модели с переполнением

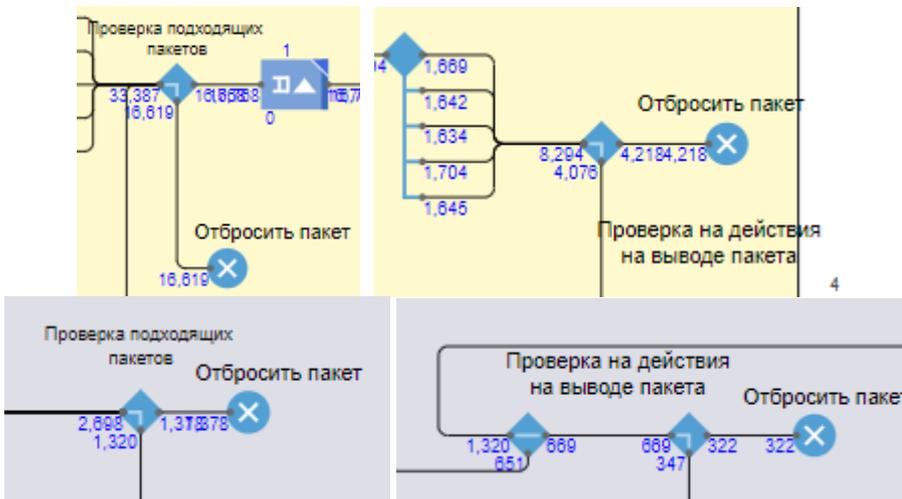


Рис. 9. Обработка пакетов

Заключение

Исследованы возможности и ограничения коммутатора с поддержкой протокола *OpenFlow* в ПКС. Представленная модель демонстрирует значи-

тельные преимущества, такие как гибкость, централизованное управление, возможность реализации политик безопасности и управления трафиком. Она показала, что сетевые администраторы могут эффективно управлять потоками данных, улучшая производительность сети и гибко настраивая ее в соответствии с требованиями приложений и бизнес-процессов.

Преимущества модели коммутатора *OpenFlow* становятся особенно ярко выраженными в средах с большим количеством виртуальных машин, облачных вычислений и технологий ПКС. Она может быть успешно использована в центральных и участвующих в пересылке узлах сети, в центрах обработки данных, дата-центрах, а также для управления трафиком в корпоративных и кампусных сетях.

На протяжении всего анализа данная модель показала, что количество тестируемых пакетов составило максимально допустимое значение, которое позволяет моделировать среда *AnyLogic*, а показатель обработанных пакетов поднялся с 35 000 до 50 000. Средняя длина пакета составляла 1,3 Кб (длина варьировалась от 1200 до 2500 байт).

Список литературы

1. Мухизи С., Шамшин Г. И., Мутханна А. С., Киричек Р. В. Исследование и разработка имитационной модели функционирования фрагмента программно-конфигурируемой сети как системы массового обслуживания // Информационные технологии и телекоммуникации. 2016. Т. 4, № 4. С. 49–57.
2. Олифер В. Г., Олифер Н. А. Компьютерные сети. Принципы, технологии, протоколы. 4-е изд. СПб. : Питер, 2010. 943 с.
3. Shalimov A., Zuikov D., Zimarina D., Pashkov V., Smeliansky R. [et al.]. Advanced study of SDN/OpenFlow controllers // Proceedings of the 9th Central & Eastern European Software Engineering Conference in Russia. ACM. 2013. P. 1–6.
4. Никульчев Е. В., Паяин С. В., Плужник Е. В. Динамическое управление трафиком программно-конфигурируемых сетей в облачной инфраструктуре // Вестник Рязанского государственного радиотехнического университета. 2013. № 3 (45). С. 54–57.
5. Никишин К. И. Моделирование и верификация топологий программно-конфигурируемых сетей // Вестник Рязанского государственного радиотехнического университета. 2022. № 80. С. 67–74.
6. Антонова В. М., Бужин И. Г., Гайфутдинов Э. А., Гнездилов В. С., Кузнецов Н. А., Миронов Ю. Б. Система управления трафиком в перспективных мобильных сетях на основе технологий SDN/NFV // Информационные процессы. 2023. Т. 23, № 1. С. 113–125.
7. Buzhin I. G., Antonova V. M., Gaifutdinov E. A., Mironov Yu. B. Methodology for a comprehensive assessment of the telecommunication services quality of transport networks using SDN/NFV technologies // T-Comm. 2022. Vol. 16, № 12. P. 40–45.
8. Никишин К. И. Метод ранней диагностики потерь трафика реального времени с контролем таймаутов в программно-конфигурируемых сетях // Известия Юго-Западного государственного университета. 2022. Т. 26, № 2. С. 142–158.
9. Jensen K., Kristensen L. M. Coloured Petri Nets. Modelling and Validation of Concurrent Systems. Berlin : Springer, 2009. 384 p.
10. McKeown N., Anderson T., Balakrishnan H. [et al.]. Openflow: enabling innovation in campus networks // ACM SIGCOMM Computer Communication Review. 2008. Vol. 38, № 2. P. 69–74.
11. Kobayashi M., Seetharaman S., Parulkar G., Appenzeller G., Little J., Van Reijendam J., McKeown N. Maturing of OpenFlow and Software-Defined Networking Through Deployments // Computer Networks. 2014. Vol. 61. P. 151–175.

References

1. Mukhizi S., Shamshin G.I., Mutkhanna A.S., Kirichek R.V. Research and development of a simulation model of the functioning of a fragment of a software-defined network as a queuing system. *Informatsionnye tekhnologii i telekommunikatsii = Information technology and telecommunications*. 2016;4(4):49–57. (In Russ.)
2. Olifer V.G., Olifer N.A. *Komp'yuternye seti. Printsipy, tekhnologii, protokoly. 4-e izd. = Computer networks. Principles, technologies, protocols. The 4th edition*. Saint Petersburg: Piter, 2010:943. (In Russ.)
3. Shalimov A., Zuikov D., Zimarina D., Pashkov V., Smeliansky R. et al. Advanced study of SDN/OpenFlow controllers. *Proceedings of the 9th Central & Eastern European Software Engineering Conference in Russia. ACM*. 2013:1–6.
4. Nikul'chev E.V., Payain S.V., Pluzhnik E.V. Dynamic traffic management of software-defined networks in cloud infrastructure. *Vestnik Ryazanskogo gosudarstvennogo radio-tekhnicheskogo universiteta = Bulletin of Ryazan State Radio Engineering University*. 2013;(3):54–57. (In Russ.)
5. Nikishin K.I. Modeling and verification of software-defined network topology. *Vestnik Ryazanskogo gosudarstvennogo radiotekhnicheskogo universiteta = Bulletin of Ryazan State Radio Engineering University*. 2022;(80):67–74. (In Russ.)
6. Antonova V.M., Buzhin I.G., Gayfutdinov E.A., Gnezdilov V.S., Kuznetsov N.A., Mironov Yu.B. Traffic management system in advanced mobile networks based on technologies of SDN/NFV. *Informatsionnye protsessy = Information processes*. 2023;23(1):113–125. (In Russ.)
7. Buzhin I.G., Antonova V.M., Gaifutdinov E.A., Mironov Yu.B. Methodology for a comprehensive assessment of the telecommunication services quality of transport networks using SDN/NFV technologies. *T-Comm*. 2022;16(12):40–45. (In Russ.)
8. Nikishin K.I. Method for early diagnostics of real-time traffic losses with timeout monitoring in software-defined networks. *Izvestiya Yugo-Zapadnogo gosudarstvennogo universiteta = Proceedings of the South-West State University*. 2022;26(2):142–158. (In Russ.)
9. Jensen K., Kristensen L.M. Coloured Petri Nets. *Modelling and Validation of Concurrent Systems*. Berlin: Springer, 2009:384.
10. McKeown N., Anderson T., Balakrishnan H. et al. Openflow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*. 2008;38(2):69–74.
11. Kobayashi M., Seetharaman S., Parulkar G., Appenzeller G., Little J., Van Reijendam J., McKeown N. Maturing of OpenFlow and Software-Defined Networking Through Deployments. *Computer Networks*. 2014;61:151–175.

Информация об авторах / Information about the authors

Илья Александрович Филиппов

студент, Пензенский государственный университет (Россия, г. Пенза, ул. Красная, 40)

E-mail: ilya_filippov_2002@inbox.ru

Цуа А. Филиппов

Student, Penza State University
(40 Krasnaya street, Penza, Russia)

Валентин Алексеевич Волков

студент, Пензенский государственный университет (Россия, г. Пенза, ул. Красная, 40)

E-mail: valek.volkov.2001@mail.ru

Valentin A. Volkov

Student, Penza State University
(40 Krasnaya street, Penza, Russia)

Кирилл Игоревич Никишин

кандидат технических наук, доцент
кафедры вычислительной техники,
Пензенский государственный
университет (Россия, г. Пенза,
ул. Красная, 40)

E-mail: nkipnz@mail.ru

Kirill I. Nikishin

Candidate of engineering sciences, associate
professor of the sub-department of computer
engineering, Penza State University
(40 Krasnaya street, Penza, Russia)

Евгений Иванович Гурин

доктор технических наук, профессор,
профессор кафедры вычислительной
техники, Пензенский государственный
университет (Россия, г. Пенза,
ул. Красная, 40)

E-mail: gurin2@yandex.ru

Evgeny I. Gurin

Doctor of engineering sciences,
professor, professor of the sub-department
of computer engineering, Penza State
University (40 Krasnaya street,
Penza, Russia)

Авторы заявляют об отсутствии конфликта интересов / The authors declare no conflicts of interests.

Поступила в редакцию / Received 12.03.2024

Поступила после рецензирования и доработки / Revised 02.04.2024

Принята к публикации / Accepted 11.05.2024