

Труды МАИ. 2023. № 128
Trudy MAI, 2023, no. 128

Научная статья
УДК 621.396
DOI: [10.34759/trd-2023-128-11](https://doi.org/10.34759/trd-2023-128-11)

АЛГОРИТМ БАЛАНСИРОВКИ НАГРУЗКИ СЛУЖЕБНОГО ТРАФИКА ВИРТУАЛИЗИРОВАННОЙ ИНФРАСТРУКТУРЫ МОБИЛЬНЫХ СЕТЕЙ БУДУЩЕГО ПОКОЛЕНИЯ

**Игорь Геннадьевич Бужин¹✉, Вероника Михайловна Антонова²,
Юрий Борисович Миронов³, Эльдар Альбертович Гайфутдинов⁴**

^{1,2,3,4}Московский технический университет связи и информатики, МТУСИ,
Москва, Россия

¹i.g.buzhin@mtuci.ru✉

²xarti@mail.ru

³i.b.mironov@mtuci.ru

⁴e.a.gaifutdinov@mtuci.ru

Аннотация. В статье предлагается алгоритм балансировки нагрузки служебного трафика виртуализированной инфраструктуры между управляющими устройствами. В результате работы алгоритма выделяются группы узлов виртуализированной инфраструктуры, общая нагрузка служебного трафика от которых не превышает максимальную производительность узлов контура управления. Приведена программная реализация алгоритм балансировки нагрузки управляющих сообщений, приведен пример работы алгоритма.

Ключевые слова: мобильные сети будущего поколения, балансировка нагрузки, сетевые слои, сетевая связность, виртуализированная инфраструктуры, сети передачи данных

Для цитирования: Бужин И.Г., Антонова В.М., Миронов Ю.Б., Гайфутдинов Э.А.

Алгоритм балансировки нагрузки служебного трафика виртуализированной инфраструктуры мобильных сетей будущего поколения // Труды МАИ. 2023. № 128.

DOI: [10.34759/trd-2023-128-11](https://doi.org/10.34759/trd-2023-128-11)

Original article

LOAD BALANCING ALGORITHM FOR SERVICE TRAFFIC IN THE VIRTUALIZED INFRASTRUCTURE OF NEXT GENERATION MOBILE NETWORKS

Igor G. Buzhin^{1✉}, Veronika M. Antonova²,

Yuri B. Mironov³, Eldar A. Gaifutdinov⁴

^{1,2,3,4}Moscow technical university of communications and informatics,

Moscow, Russia

i.g.buzhin@mtuci.ru[✉]

xarti@mail.ru²

i.b.mironov@mtuci.ru³

e.a.gaifutdinov@mtuci.ru⁴

Abstract. One of the key features that distinguishes future generation mobile networks (5G and 6G) from the previous generations is an increase in the number of services provided, a significant increase in data transmission speeds and a highly reliable control loop. The

architecture of such networks is being built employing Software Defined Networking (SDN) and Network Functions Virtualization (NFV) technologies, as well as Network Slicing technology.

It is necessary to develop messages load balancing algorithm of the virtualized infrastructure between the control devices to ensure the high fault tolerance level of control, load balancing, network connectivity of mobile network architecture of future generation mobile network by the SDN/NFV technology employing.

All SDN switches should be necessarily divided into the groups depending on their output loading so that the common loading from one group of switches does not exceed the SDN controller productive capacity for the load distribution (Packet-In messages received by the controller per second) from the virtualized infrastructure based on the SDN/NFV controllers.

This requires permanent load measuring on each of the controllers. The main challenge is the load-balancing task, i.e. the SDN switches allocation over controllers. The network topology can be plotted as a graph, where the nodes represent the SDN switches (virtualized infrastructure nodes) with the rated load about network states and the controllers with the rated data flow processing capacity, while the graph edges represent the data links.

Let us decompose the task into two subtasks; each of them herewith is being solved by a different algorithm. The first algorithm has to divide the switches into groups based on the information on the current loads and flows in the system, so that the groups can be assigned to the controllers employed in the system. The second algorithm should allocate the groups resulting from the first algorithm running to the employed controllers so that the minimum number of migrations would be required to convert the considered system. The

algorithm for Packet-In messages allocating to controllers is new and has no equivalents for comparison.

The algorithm results in groups of virtualized infrastructure nodes with the total service traffic load not exceeding the maximum performance of the control loop nodes.

Keywords: future generation mobile networks, load balancing, network layers, network connectivity, virtualised infrastructure, data networks

For citation: Buzhin I.G., Antonova V.M., Mironov Yu.B., Gaifutdinov E.A. Load balancing algorithm for service traffic in the virtualized infrastructure of next generation mobile networks. *Trudy MAI*, 2023, no.128. DOI: [10.34759/trd-2023-128-11](https://doi.org/10.34759/trd-2023-128-11)

Введение

В архитектуре мобильных сетей будущего поколения (5G и 6G) сетевые услуги реализуются с помощью контура управления, в основе которого лежат технологии программно-конфигурируемых сетей (SDN) и виртуализации сетевых функций (NFV) [1]. Таким образом, в мобильных сетях будущего поколения (5G и 6G) вместо сетевых топологий с использованием традиционных аппаратных сетевых узлов применяются распределенные программно-управляемые сетевые архитектуры [3].

Одной из главных особенностей мобильных сетей будущего поколения является наличие технологии разделения сетевой инфраструктуры на слои (Network Slicing). С помощью данной технологии вычислительные и сетевые ресурсы мобильных сетей разделяются на виртуальные слои (Slices) [3]. Данная процедура возможна за счет применения технологий SDN/NFV, в результате чего каждый сетевой слой представляет собой отдельную изолированную сеть (в том числе сеть

радиодоступа 5G), в которой реализуется определенная услуга мобильной сети со своим уровнем QoS. Применяя технологию Network Slicing в мобильных сетях будущего поколения, сетевые услуги необходимо рассматривать как набор сетевых составляющих (правил, платформ, виртуальных сущностей), которые соединены в единую услугу. Данное свойство повышает гибкость и масштабируемость сети связи. Помимо этого, разные сетевые слои, которые реализуются с помощью SDN/NFV, могут функционировать на одних и тех же аппаратных ресурсах. Таким образом, управление и конфигурирование такими виртуальными сетевыми сущностями может производиться с единого контура управления, например, системой управления и взаимодействия (MANO).

В архитектуре на основе SDN/NFV особое место занимают сетевой контроллер и прикладной уровень. Логически централизованный контроллер обеспечивает функции консолидированного управления виртуализированной инфраструктурой. Контроллер использует протокол управления для установки правил потока в каждом экземпляре виртуализированной инфраструктуры для маршрутизации трафика по уровню передачи данных мобильной сети. Прикладной уровень состоит из всех приложений мобильной сети. Традиционные элементы мобильной сети, такие как функция политик и правил тарификации (PCRF), домашний абонентский сервер (HSS), объект управления мобильностью (MME) и аутентификация, авторизация и учет (AAA), теперь являются программными приложениями.

Для обеспечения отказоустойчивости, балансировки нагрузки, обеспечения сетевой связности частей архитектуры мобильной сети будущего поколения с использованием технологии SDN/NFV необходимо разработать алгоритм

балансировки нагрузки сообщений виртуализированной инфраструктуры между управляющими устройствами.

Алгоритм балансировки нагрузки управляющих сообщений

виртуализированной инфраструктуры мобильных сетей будущего поколения

Входными данными методики расчета являются граф $G = (S, E)$, который описывает сети связи с применением ПКС. Здесь $|S|$ – число коммутаторов ПКС (узлов виртуализированной инфраструктуры) в сети, $E = \{E_{i,j}: E_{i,j} = (S_i, S_j)\}$ – множество соединений между узлами. Также входными данными является $L = \{l_{i,j}, i, j = 0, \dots, |S| - 1\}$ – множество длин кратчайших путей в графе между узлами i и j . Длина пути определяется как сумма соединений между узлами сети, входящих в этот путь. Также дано $C = \{c_i, i = 0, \dots, |C| - 1\}$ – множество контроллеров ПКС, $|C|$ – число контроллеров ПКС в сети, $C_{\text{отк}}$ - число контроллеров ПКС, при отказе которых сеть связи будет устойчива, $D_i = \{S_i, S_i \in S\}$ – множество коммутаторов ПКС, входящих в сегмент i -го контроллера ПКС, S_{max} – максимально возможное число коммутаторов ПКС в сегменте контроллера, T_{max} – максимально возможная задержка между контроллером и коммутатором ПКС одного сегмента, P_i – максимальная производительность (пакетов Packet-In в секунду) контроллера c_i , I_i – среднее число новых потоков в секунду, приходящих в коммутатор S_i , I_{ij} – среднее число потоков в секунду между коммутаторами S_i и S_j .

Необходимо разбить граф $G = (S, E)$ на такие группы связности, чтобы сумма чисел в узлах каждой группы связности не превосходила производительность P_i контроллера. Предлагаемый к рассмотрению алгоритм выполняет функции

балансировки нагрузки контроллеров так, что группа коммутаторов, которая не подлежит перераспределению управляющего устройства, будет иметь суммарную нагрузку такую, что контроллер не останется в недогруженном режиме.

Произведем декомпозицию задачи на две подзадачи, каждую из которых будет решать свой алгоритм. Первый должен на основе информации о текущих нагрузках и потоках в системе разделить коммутаторы на группы так, чтобы в дальнейшем получившиеся группы можно было назначить существующим в системе контроллерам. Этот алгоритм будем называть «алгоритмом разделения коммутаторов на группы». Второй алгоритм должен получившиеся в результате работы первого алгоритма группы распределить по существующим контроллерам так, чтобы для преобразования существующей системы потребовалось минимальное количество миграций. Этот алгоритм будем называть «алгоритмом распределения групп коммутаторов по контроллерам».

По векторам L_c построим ненаправленный граф G , узлы которого будут соответствовать существующим в системе коммутаторам, а ребра - существующим между коммутаторами каналам связи. Для каждого узла зафиксируем число, которое будет обозначать номинальную нагрузку на соответствующий коммутатор - нагрузку с конечных устройств. Эта нагрузка не зависит от распределения коммутаторов по группам. Для каждого ребра также зафиксируем его вес - сумму по двум портам на концах соответствующего канала связи.

В графе сети аналогом группы коммутаторов будет являться компонента связности графа. То есть фаза разделения коммутаторов на группы — это задача разбиения графа на такие компоненты связности, чтобы каждая компонента

связности удовлетворяла условию, что сумма чисел в узлах этой компоненты не превосходит мощности контроллера.

Фаза состоит из повтора шагов. На каждом шаге выполняется набор действий:

1.1. Вычисляется $I_i + I_{ij}$ на узлах каждой компоненты связности. Если $I_i + I_{ij} \leq P_i$, то алгоритм завершает работу.

1.2. Если $I_i + I_{ij} > P_i$, необходимо:

1.2.1. Создать новый граф с вершинами S без соединений: $G^1 = (S, E^{\text{нов}})$.

1.2.2. В графе G выбрать соединение $E_{i,j}$ с максимальным I_{ij} , удалить его.

Добавить в граф G^1 . На узлах графа G пересчитываются нагрузки: из существующих значений нагрузки вычитаются нагрузки, соответствующие концам ребер, которые были удалены из G . Получилось разделение на группы.

1.3. Если $I_i + I_{ij} \leq P_i$ на каждой компоненте связности нового графа G , то объединение прошло успешно. Если $I_i + I_{ij} > P_i$, то объединение считается неуспешным. Производится удаление следующего (по нагрузке) ребра графа G .

В результате работы второй фазы выделяется набор групп, сумма на узлах каждой из которых не превосходит мощности контроллера. При этом в этом наборе могут существовать такие компоненты связности, при объединении которых также не будет нарушаться условие на сумму чисел в узлах. Задача второй фазы состоит в том, чтобы произвести такое объединение.

На каждом шаге второй фазы выполняется набор действий:

1) Добавляется ребро с наибольшим весом из тех, которые были удалены во второй фазе.

2) На узлах пересчитываются нагрузки в соответствии с ребрами, добавленными в действиях 1 и 2. То есть из значений на узлах вычитаются значения, соответствующие портам на концах тех ребер, которые были добавлены.

3) Вычисляется нагрузка получившейся объединенной группы:

а) Если нагрузка меньше, чем мощность коммутаторов, то объединение считается успешным.

б) Если нагрузка больше, чем мощность коммутаторов, то объединение считается неуспешным, добавленные в действиях 1 и 2 ребра удаляются.

Далее происходит распределение полученных компонент связности по управляющим узлам C за минимальное количество действий. Выделяются отсортированные множества групп связности (в порядке убывания величины нагрузки служебного трафика) и сравниваются с текущими связностями графа (сетевые узлы, которые подчинены одному узлу управления, считаются в одной группе). На основании сравнения (в случае несоответствия двух множеств) осуществляется переключение сетевых узлов между компонентами управления. После этого получаем новые компоненты связности.

В результате работы алгоритма выделяются группы узлов виртуализированной инфраструктуры, общая нагрузка служебного трафика от которых не превышает максимальную производительность узлов контура управления. Данные группы распределены в порядке убывания по контроллерам SDN.

Программная реализация алгоритма балансировки нагрузки управляющих сообщений виртуализированной инфраструктуры мобильных сетей будущего поколения

Программная реализация алгоритма балансировки нагрузки управляющих сообщений виртуализированной инфраструктуры мобильных сетей будущего поколения осуществлена на высокоуровневом языке программирования Python с использованием библиотеки networkx (библиотека для изучения графиков и сетей) [16]. Логика программы построена преимущественно с применением словарей, то есть данные хэшируются, что приводит к увеличению быстродействия алгоритма.

На рисунке 1 представлена блок-схема программной реализации алгоритма. Пример выполнения программным кодом предложенного алгоритма будет осуществляться на примере модели SDN сети с шестью коммутаторами SDN (switch1, switch2, ..., switch6) и тремя контроллерами (controller1 – основной, controller2 и controller3 – резервные). Коммутаторы соединены между собой в следующем порядке: switch1 – switch2, switch2 – switch3, switch2 – switch6, switch3 – switch5, switch4 – switch5, switch5 – switch6.

Максимальная производительность контроллеров – 8. Номинальные нагрузки на коммутаторы и веса ребер перед началом работы алгоритма изображены на рисунке 2. Первая, вторая и третья фазы представлены на рисунках 3, 4 и 5 соответственно. В ходе работы алгоритма получилось две группы коммутаторов: switch1-3, switch4-6. Суммы узлов в компонентах связности для switch1-3 составляет 7,9, а для switch4-6 - 6,8. Третья фаза алгоритма выполняет перераспределение групп по контроллерам. Первая группа, имеющая наибольшее значение суммы узлов,

остается подключенной к основному контроллеру, а вторая группа переключилась на резервный контроллер controller2. Причем первый контроллер не находится в недогруженном состоянии.

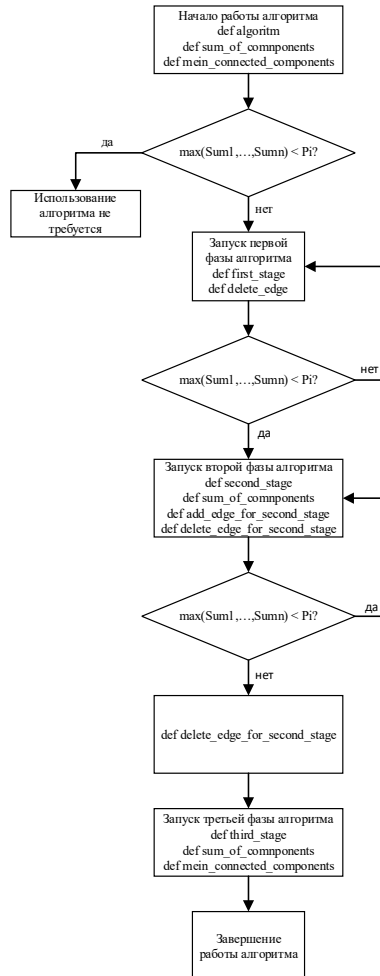


Рисунок 1 - Блок-схема программной реализации алгоритма балансировки нагрузки

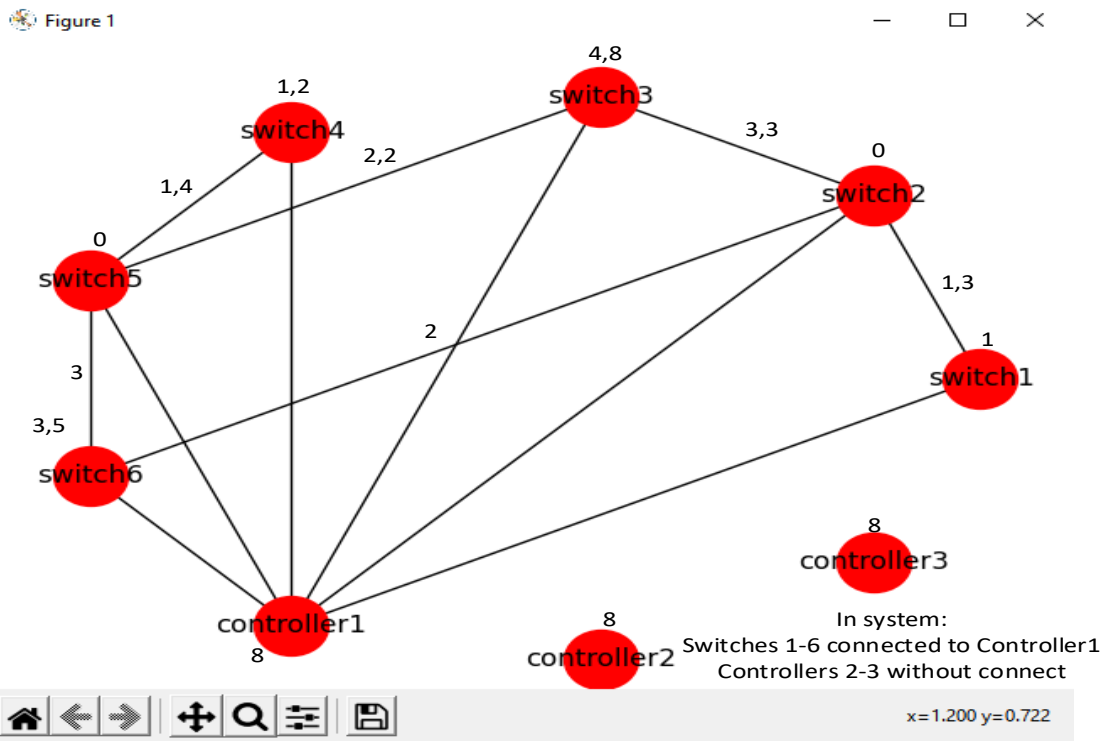


Рисунок 2 - Пример топологии сети

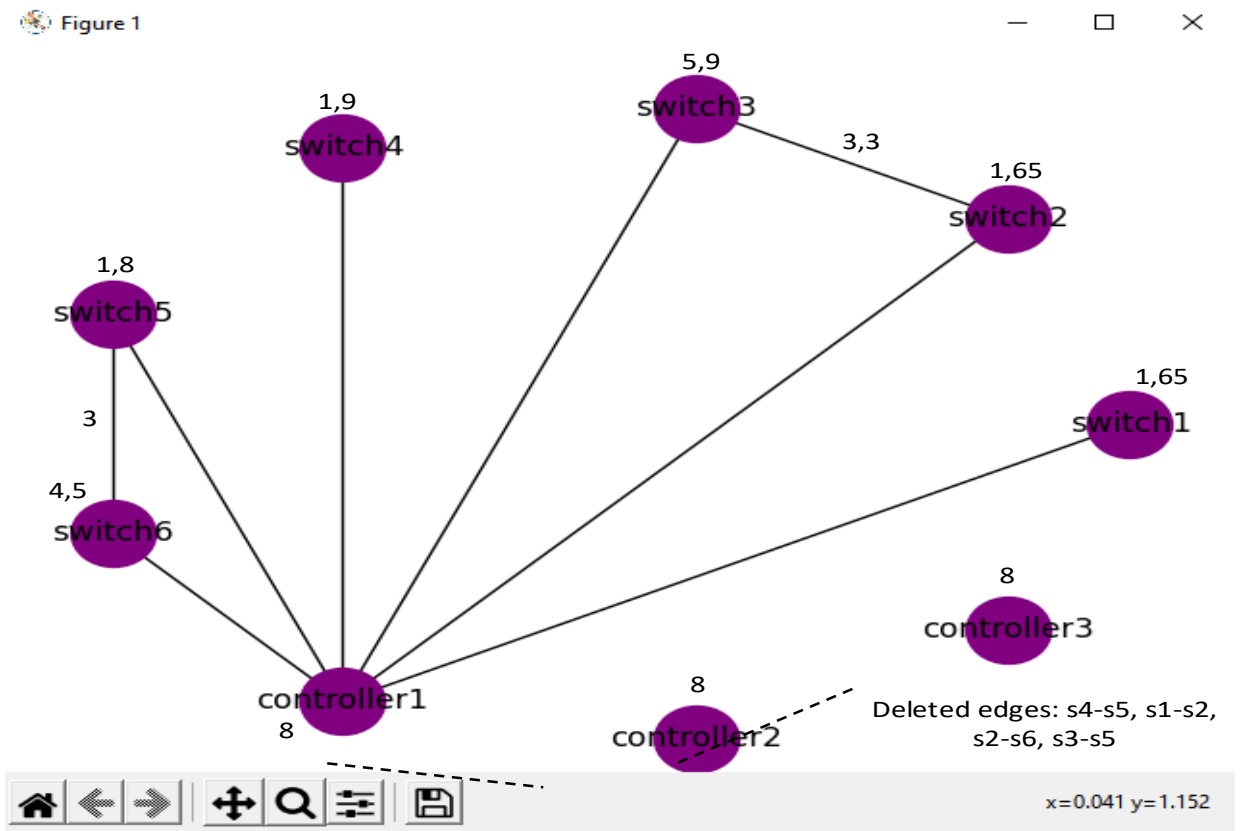


Рисунок 3 - Первая фаза алгоритма

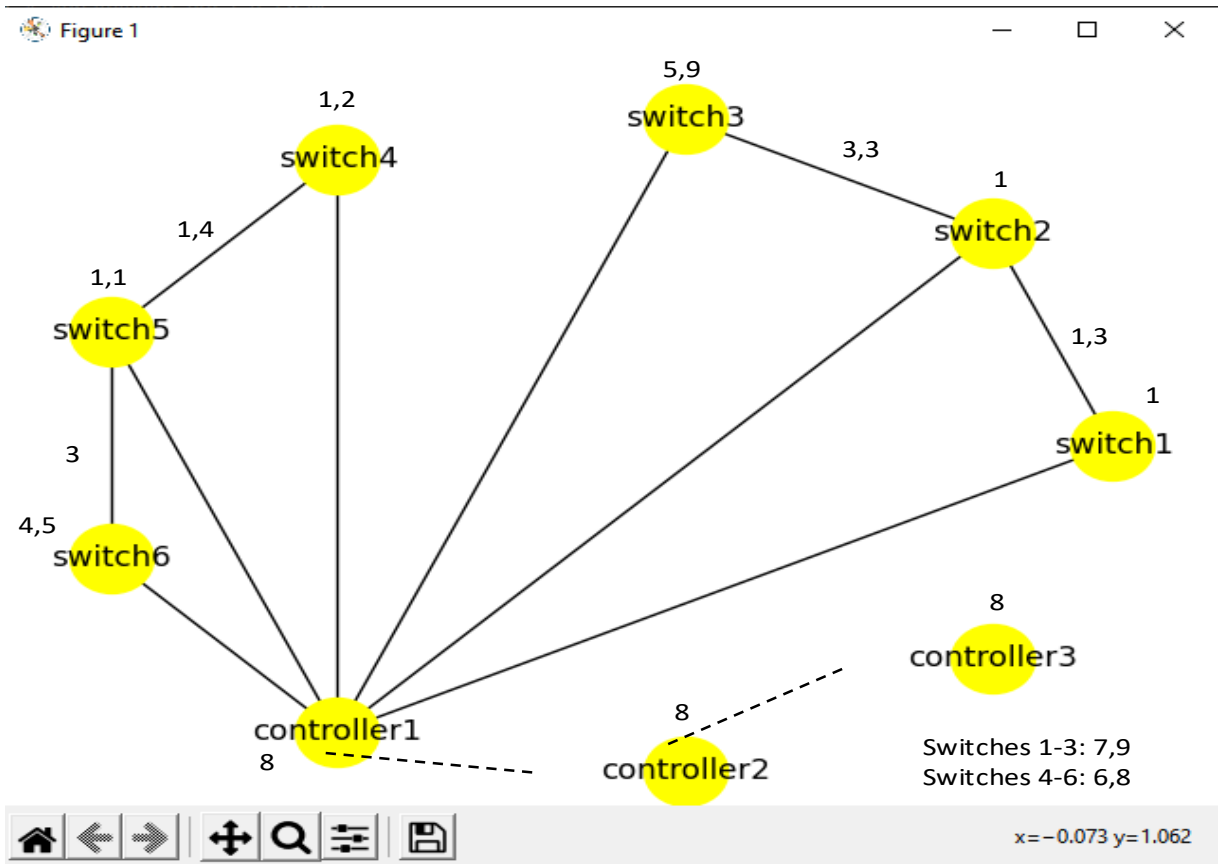


Рисунок 4 - Вторая фаза алгоритма

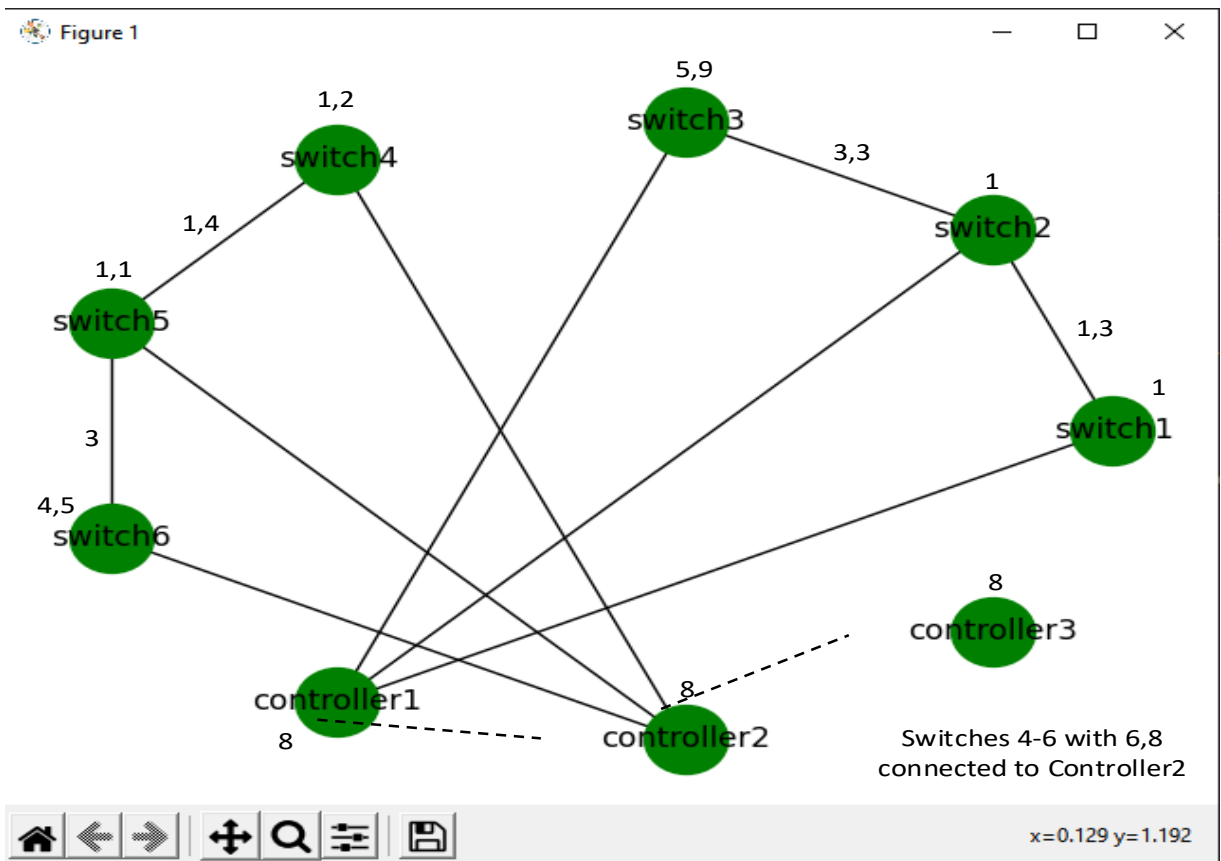


Рисунок 5 – Третья фаза алгоритма

Заключение

Мобильные сети будущего поколения должны предоставлять большое число сервисов с высоким качеством обслуживания. Для этого необходимо наличие отказоустойчивого уровня управления, балансировки нагрузки, обеспечения сетевой связности частей архитектуры мобильной сети будущего поколения с использованием технологии SDN/NFV.

Для обеспечения отказоустойчивости и распределенности контура управления мобильных сетей будущего поколения предложен алгоритм балансировки нагрузки, поступающих от коммутаторов SDN (узлов виртуализированной инфраструктуры), между контроллерами SDN. В результате работы алгоритма выделяются группы узлов виртуализированной инфраструктуры, общая нагрузка служебного трафика от которых не превышает максимальную производительность узлов контура управления.

Список источников

1. Lin Y.B., Tseng C.C., Wang M.H. Effects of Transport Network Slicing on 5G Applications // Future Internet, 2021, vol. 13, no. 69. URL: <https://doi.org/10.3390/fi13030069>
2. Tsai C.C., Lin F.J., Tanaka H. Evaluation of 5G Core Slicing on User Plane Function // Communications and Network, 2021, vol. 13, pp. 79-92. URL: <https://doi.org/10.4236/cn.2021.133007>.
3. P. Camps-Aragó, S. Delaere, P. Ballon. 5G business models: Evolving mobile network operator roles in new ecosystems // Conference: Smart Cities & Information and

Communication Technology (CTTE-FITCE), 2019. DOI:[10.1109/CTTE-FITCE.2019.8894822](https://doi.org/10.1109/CTTE-FITCE.2019.8894822)

4. Z. Li, X. Wang, T. Zhang. System architecture and technological basics of 5G, Chap. 4 in 5G+ How 5G Change the Society, Springer, 2021.

5. S. Wijethilaka, M. Liyanage. Survey on network slicing for Internet of things realization in 5G networks // IEEE Communications Surveys & Tutorials, 2021, vol. 23, no. 2, pp. 957–994. DOI:[10.1109/COMST.2021.3067807](https://doi.org/10.1109/COMST.2021.3067807)

6. Borcoci E. et al. An Overview of 5G Slicing Operational Business Models for Internet of Vehicles, Maritime IoT Applications and Connectivity Solutions // IEEE Access, 2021. DOI:[10.1109/ACCESS.2021.3128496](https://doi.org/10.1109/ACCESS.2021.3128496)

7. Xue H., Kim K.T., Youn H.Y. Dynamic Load Balancing of Software-Defined Networking Based on Genetic-Ant Colony Optimization // Sensors, 2019, vol. 19, pp. 311. DOI:[10.3390/s19020311](https://doi.org/10.3390/s19020311)

8. Cui X., Huang X., Ma Y., Meng Q. A load balancing routing mechanism based on SDWSN in smart city // Electronics, 2019, vol. 8, no. 273. DOI:[10.3390/electronics8030273](https://doi.org/10.3390/electronics8030273)

9. Semong T., Maupong T., Anokye S., Kehulakae K., Dimakatso S., Boipelo G., Sarefo S. Intelligent load balancing techniques in software defined networks: A survey // Electronics, 2020, vol. 9 (7), pp. 1091.

10. Бахтин А.А., Волков А.С., Солодков А.В., Баскаков А.Е. Разработка модели сегмента сети SDN для стандарта 5G // Труды МАИ. 2021. № 117. URL: <https://trudymai.ru/published.php?ID=122307>. DOI: [10.34759/trd-2021-117-07](https://doi.org/10.34759/trd-2021-117-07)

11. Волков А.С., Баскаков А.Е. Разработка процедуры двунаправленного поиска для решения задачи маршрутизации в транспортных программно-конфигурируемых

- сетей // Труды МАИ. 2021. № 118. URL: <https://trudymai.ru/published.php?ID=158240>. DOI: [10.34759/trd-2021-118-07](https://doi.org/10.34759/trd-2021-118-07)
12. Principles and Practices for Securing Software-Defined Networks. ONF TR-511 Open Networking Foundation, 2015. URL: <https://pdfslide.net/documents/principles-and-practices-for-securing-software-defined-networks.html>
13. Salman O. et al. Multi-level security for the 5G/IoT ubiquitous network // 2017 Second International Conference on Fog and Mobile Edge Computing (FMEC), IEEE, 2017, pp. 188-193. DOI:[10.1109/FMEC.2017.7946429](https://doi.org/10.1109/FMEC.2017.7946429).
14. Бужин И.Г., Антонова В.М., Миронов Ю.Б., Антонова В.А., Корчагина А.С, Канищева М.Г. Особенности классификации и фильтрации трафика сети передачи данных 6G // Труды МАИ. 2021. № 121. URL: <https://trudymai.ru/published.php?ID=162659>. DOI: 10.34759/trd-2021-121-12.
15. Масюков И.И. Метод и устройство расположения задач в реконфигурируемых вычислительных системах // Труды МАИ. 2021. № 120. URL: <https://trudymai.ru/published.php?ID=161427>. DOI: 10.34759/trd-2021-120-13.
16. Антонова В.М., Захир Б.М., Кузнецов Н.А. Моделирование графов с различными видами достижимости с помощью языка Python // Информационные процессы. 2019. Т. 19. № 2. С. 159-169.
17. Буранова М.А., Карташевский В.Г., Мутханна А.С. Анализ параметров функционирования программно-конфигурируемой сети на основе протокола OpenFlow // Электросвязь. 2022. № 4. С. 2-7. DOI [10.34832/ELSV.2022.29.4.001](https://doi.org/10.34832/ELSV.2022.29.4.001)

18. Перепелкин Д.А., Нгуен В.Т. Исследование процессов балансировки нагрузки в программно-конфигурируемых сетях на основе генетического алгоритма // Вестник Рязанского государственного радиотехнического университета. 2021. № 77. С. 43-57.
19. Калмыков Н.С., Докучаев В.А. Сегментная маршрутизация как основа программно-конфигурируемой сети // Т-Comm: Телекоммуникации и транспорт. 2021. Т. 15. № 7. С. 50-54.
20. Волков А.С., Баскаков А.Е. Разработка алгоритма многопутевой маршрутизации в программно-конфигурируемых сетях связи // Т-Comm-Телекоммуникации и транспорт. 2021. Т. 15. № 9. С. 17-23.

References

1. Lin Y.B., Tseng C.C., Wang M.H. Effects of Transport Network Slicing on 5G Applications, *Future Internet*, 2021, vol. 13, no. 69. URL: <https://doi.org/10.3390/fi13030069>
2. Tsai C.C., Lin F.J., Tanaka H. Evaluation of 5G Core Slicing on User Plane Function, *Communications and Network*, 2021, vol. 13, pp. 79-92. URL: <https://doi.org/10.4236/cn.2021.133007>.
3. P. Camps-Aragó, S. Delaere, P. Ballon. 5G business models: Evolving mobile network operator roles in new ecosystems, *Conference: Smart Cities & Information and Communication Technology (CTTE-FITCE)*, 2019. DOI:[10.1109/CTTE-FITCE.2019.8894822](https://doi.org/10.1109/CTTE-FITCE.2019.8894822)
4. Z. Li, X. Wang, T. Zhang. *System architecture and technological basics of 5G*, Chap. 4 in *5G+ How 5G Change the Society*, Springer, 2021.

5. S. Wijethilaka, M. Liyanage. Survey on network slicing for Internet of things realization in 5G networks, *IEEE Communications Surveys & Tutorials*, 2021, vol. 23, no. 2, pp. 957–994. DOI:[10.1109/COMST.2021.3067807](https://doi.org/10.1109/COMST.2021.3067807)
6. Borcoci E. et al. An Overview of 5G Slicing Operational Business Models for Internet of Vehicles, Maritime IoT Applications and Connectivity Solutions, *IEEE Access*, 2021. DOI:[10.1109/ACCESS.2021.3128496](https://doi.org/10.1109/ACCESS.2021.3128496)
7. Xue H., Kim K.T., Youn H.Y. Dynamic Load Balancing of Software-Defined Networking Based on Genetic-Ant Colony Optimization, *Sensors*, 2019, vol. 19, pp. 311. DOI:[10.3390/s19020311](https://doi.org/10.3390/s19020311)
8. Cui X., Huang X., Ma Y., Meng Q. A load balancing routing mechanism based on SDWSN in smart city, *Electronics*, 2019, vol. 8, no. 273. DOI:[10.3390/electronics8030273](https://doi.org/10.3390/electronics8030273)
9. Semong T., Maupong T., Anokye S., Kehulakae K., Dimakatso S., Boipelo G., Sarefo S. Intelligent load balancing techniques in software defined networks: A survey, *Electronics*, 2020, vol. 9 (7), pp. 1091.
10. Bakhtin A.A., Volkov A.S., Solodkov A.V., Baskakov A.E. *Trudy MAI*, 2021, no. 117. URL: <https://trudymai.ru/eng/published.php?ID=122307>. DOI: 10.34759/trd-2021-117-07
11. Volkov A.S., Baskakov A.E. *Trudy MAI*, 2021, no. 118. URL: <https://trudymai.ru/eng/published.php?ID=158240>. DOI: 10.34759/trd-2021-118-07
12. Principles and Practices for Securing Software-Defined Networks. ONF TR-511 Open Networking Foundation, 2015. URL: <https://pdfslide.net/documents/principles-and-practices-for-securing-software-defined-networks.html>

13. Salman O. et al. Multi-level security for the 5G/IoT ubiquitous network, *2017 Second International Conference on Fog and Mobile Edge Computing (FMEC)*, IEEE, 2017, pp. 188-193. DOI: [10.1109/FMEC.2017.7946429](https://doi.org/10.1109/FMEC.2017.7946429).
14. Buzhin I.G., Antonova V.M., Mironov Yu.B., Antonova V.A., Korchagina A.S., Kanishcheva M.G. *Trudy MAI*, 2021, no. 121. URL: <https://trudymai.ru/eng/published.php?ID=162659>. DOI: 10.34759/trd-2021-121-12.
15. Masyukov I.I. *Trudy MAI*, 2021, no. 120. URL: <https://trudymai.ru/eng/published.php?ID=161427>. DOI: 10.34759/trd-2021-120-13.
16. Antonova V.M., Zakhir B.M., Kuznetsov N.A. *Informatsionnye protsessy*, 2019, vol. 19, no. 2, pp. 159-169.
17. Buranova M.A., Kartashevskii V.G., Mutkhanna A.S. *Elektrosvyaz'*, 2022, no. 4, pp. 2-7. DOI: [10.34832/ELSV.2022.29.4.001](https://doi.org/10.34832/ELSV.2022.29.4.001)
18. Perepelkin D.A., Nguen V.T. *Vestnik Ryazanskogo gosudarstvennogo radiotekhnicheskogo universiteta*, 2021, no. 77, pp. 43-57.
19. Kalmykov N.S., Dokuchaev V.A. *T-Comm-Telekommunikatsii i transport*, 2021, vol. 15, no. 7, pp. 50-54.
20. Volkov A.S., Baskakov A.E. *T-Comm-Telekommunikatsii i transport*, 2021, vol. 15, no. 9, pp. 17-23.

Статья поступила в редакцию 18.12.2022

Одобрена после рецензирования 26.12.2022

Принята к публикации 27.02.2023

The article was submitted on 18.12.2022; approved after reviewing on 26.12.2022; accepted for publication on 27.02.2023