

УДК 004.72

Научная статья

<https://doi.org/10.32603/2071-8985-2023-16-3-15-24>**Исследование и моделирование 5G сети
и характеристик передачи в среде AnyLogic****К. И. Никишин**

Пензенский государственный университет, Пенза, Россия

nkipnz@mail.ru

Аннотация. Беспроводные сети – Wi-Fi, 4G, 5G и т. п., играют большую роль в современном мире и жизни общества. Минимальная скорость передачи сигнала 5G в 10 раз превышает скорость сигнала 4G. Цель исследования – анализ данных и разработка модели распределенной сети 5G в среде AnyLogic, его задачи заключаются в построении модели, состоящей из одной базовой станции 5G с различными характеристиками сети, оценке эффективности передачи данных в сети и анализе влияния характеристик вышек 5G (задержки, нагрузки и приоритетов) на качество обслуживания пакетов данных, а именно на скорость обработки и количество потерь пакетов. Разработана модель с одной базовой станцией и шестью вышками, которые, в свою очередь, имеют разные параметры (нагрузка, количество поступающих пакетов, приоритет) в среде AnyLogic. Были проведены экспериментальное исследование и оценка эффективности передачи данных в сети, а также проанализировано влияние характеристик на вышки (задержки, нагрузки и приоритетов), т. е. на качество обслуживания пакетов данных.

Ключевые слова: распределенные сети, 5G, базовая станция 5G, очереди, задержка пакетов, нагрузка сети, потеря пакетов, имитационное моделирование, AnyLogic

Для цитирования: Никишин К. И. Исследование и моделирование 5G сети и характеристик передачи в среде AnyLogic // Изв. СПбГЭТУ «ЛЭТИ». 2023. Т. 16, № 3. С. 15–24. doi: 10.32603/2071-8985-2023-16-3-15-24.

Original article

**Research and Simulation of a 5G Network
and Transmission Characteristics in AnyLogic****K. I. Nikishin**

Penza State University, Penza, Russia

nkipnz@mail.ru

Abstract. Wireless networks, such as Wi-Fi, 4G, and 5G, play an important role in the development of modern technologies. The minimum transmission speed of a 5G signal is 10 times higher than that of a 4G signal. The author of the present paper aims to construct and investigate a model of distributed 5G network in the AnyLogic environment. The main research objectives included construction of a model consisting of one 5G base station and different network characteristics, evaluation of the efficiency of data transmission in the network, and analysis of the impact of 5G tower characteristics (delays, loads, and priorities) on the service quality of data packets, namely, the speed of packet processing and the number of lost packets. A model with one base station and six towers having different parameters (load, number of incoming packets, priority) was developed in AnyLogic. An experimental evaluation of data transmission efficiency in the constructed network was carried out; the effect of tower characteristics (delays, loads, and priorities) on the service quality of data packets, namely, the speed of packet processing and the number of lost packets, was estimated.

Keywords: distributed networks, 5G, 5G base station, queues, packet delay, network load, lost packets, simulation, AnyLogic

For citation: Nikishin K. I. Research and Simulation of a 5G Network and Transmission Characteristics in AnyLogic // LETI Transactions on Electrical Engineering & Computer Science. 2023. Vol. 16, no. 3. P. 15–24. doi: 10.32603/2071-8985-2023-16-3-15-24.

Введение. Современные информационные технологии, программные продукты развиваются стремительно. Это относится и к компьютерным сетям [1]. Классической компьютерной сетью является Ethernet с поддержкой «качества обслуживания» (Quality of Service QoS) [2]. QoS позволяет передавать разнородный трафик в соответствии с его приоритетом, тем самым в коммутаторе Ethernet происходит классификация трафика [3], [4].

Следующим поколением сетей стало появление распределенных компьютерных сетей – Time-Triggered Ethernet [5], программно-конфигурируемые сети [6]–[10], облачные вычисления [11]–[13] и т. п. Особое место в этом ряду занимают беспроводные сети – Wi-Fi, 4G и 5G.

На сегодняшний день современные компьютерные сети предусматривают обмен сообщениями по средствам беспроводной связи (Wi-Fi, 4G, 5G). При этом сети 5G вытесняют собой сети 4G вследствие утраты последними необходимых характеристик передачи данных для удовлетворения запросов людей.

Также существуют аргументы в пользу того, что сети 5G имеют полноценные возможности для замены собой Wi-Fi. Сети 5G используют такое же широкополосное соединение, как и сети 4G, но работают на более высоких частотах.

Точность передачи сигнала 5G выше, чем у 4G, а задержки при передаче этого сигнала сводятся к минимуму. Еще одно значительное отличие двух сигналов: 5G имеет адаптацию к передаваемому контенту, а именно он может переключаться в режим пониженного энергопотребления и наоборот.

Базовая станция 5G может вмещать большее количество направленных антенн, т. е. увеличиваются затраты на оборудование, однако большее количество антенн обеспечит большее количество подключенных устройств. Главное достоинство заключается в том, что минимальная скорость передачи сигнала 5G в 10 раз превышает скорость сигнала 4G.

Цель статьи состоит в исследовании и разработке модели распределенной сети 5G сети в среде AnyLogic.

Задачами исследования являются построение модели, состоящей из одной базовых станций 5G и различными характеристиками сети, оценка эффективности передачи данных в сети и анализ влияния характеристик 5G вышек (задержки, нагрузки и приоритетов) на качество обслуживания пакетов данных, т. е. на скорость обработки пакетов и количество потерь пакетов.

Исследование и моделирование сети 5G. Имитационное моделирование компьютерных сетей играет важную роль для верификации протоколов, алгоритмов в этих сетях. Наиболее частый инструмент моделирования сетей – сети Петри с пакетом CPN Tools [14]–[15]. Существуют специализированные прикладные продукты для исследования сетей – Cisco Packet Tracer, OMNET++ [16] и т. п. Однако для исследования распределенной сети 5G подошел программный продукт AnyLogic, поскольку в среде имеется необходимое оборудование сети, кроме этого на прикладном пользовательском уровне можно настраивать сеть (задержку пакетов, приоритеты и нагрузку сети).

Была разработана модель распределенной сети 5G в программе AnyLogic, которая показывает принцип распределения устройств, подключаемых к вышкам 5G. Модель распределения подключаемых устройств к станциям 5G представлена на рис. 1.

Модель демонстрирует, как устройства (2 – телефоны, планшеты, компьютеры, ноутбуки и т. д.), которые для наглядности перемещаются по конвейеру 1, точно подключаются к вышкам 5G (на рисунке – линии). Вышки в свою очередь объединены коммутаторами (по 6 вышек на 1 коммутатор), образуя станции 5G – 4. Подобные станции распределены по континентам земного шара. Для наглядности на модели вышка, к которой подключается устройство, выделяется шестигульником 3.

Программа для функционирования модели состоит из трех классов: Main (основной), Sub и Base. Класс Main включает в себя ссылки на классы Sub и Base (subs[..] и base[..] соответственно). Класс Main содержит логическую цепь,

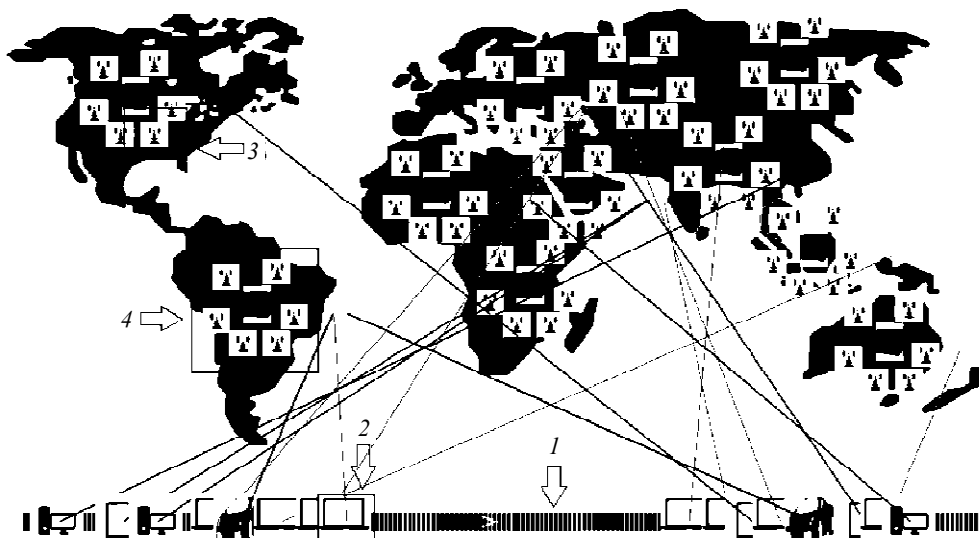


Рис. 1. Модель распределения подключаемых устройств к станциям 5G
 Fig. 1. A model of distribution of devices connected to 5G stations

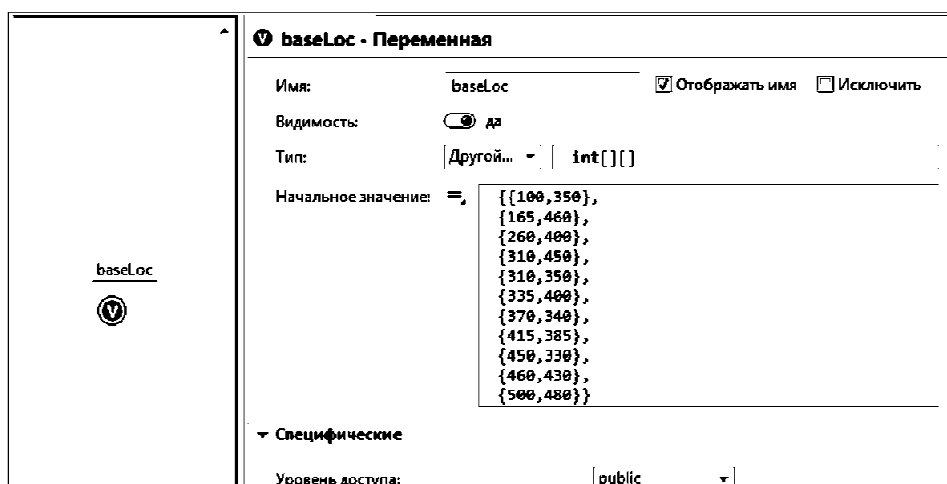


Рис. 2. Установка переменной baseLoc в программе AnyLogic
 Fig. 2. Setting the baseLoc variable in AnyLogic

состоящую из генератора «sourceSub», очереди «queue», конвейера «convey» и завершающего блока «sink». Также класс Main имеет переменную baseLoc.

Переменная baseLoc задает координаты для станций 5G на поле (на него для наглядности заранее была добавлена карта мира). Содержимое переменной создано вручную и представлено на рис. 2.

Логическая цепь необходима для генерирования абонентов в случайном порядке и подачи их на конвейер. Информация об абонентах всеми элементами логической цепи берется из класса Sub.

Генератор «sourceSub» использует переменные из класса Sub (абонентов), он в случайном порядке с определенной интенсивностью (0.5 с) добавляет абонентов в следующий блок логической цепи – очередь «queue». Данный элемент обрабатывает информацию о приоритете абонентов и выпускает на конвейер абонентов согласно приоритету.

В данном случае приоритета у абонентов нет, и они поступают на конвейер по принципу FIFO (First In First Out, «первым пришел – первым ушел»). Третий элемент логической цепи, т. е. сам конвейер, двигает поступивших абонентов по полю. Последний элемент цепи («sink») убирает абонентов с поля.

Логическая цепь устройств представлена на рис. 3 и состоит из элементов: генератора «sourceSub», очереди «queue» (рис. 4), конвейера «convey» (рис. 5) и завершающего блока «sink».

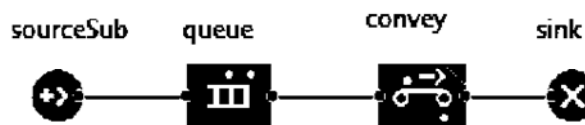


Рис. 3. Логическая цепь устройств в программе AnyLogic
 Fig. 3. Logic circuit of equipment in AnyLogic

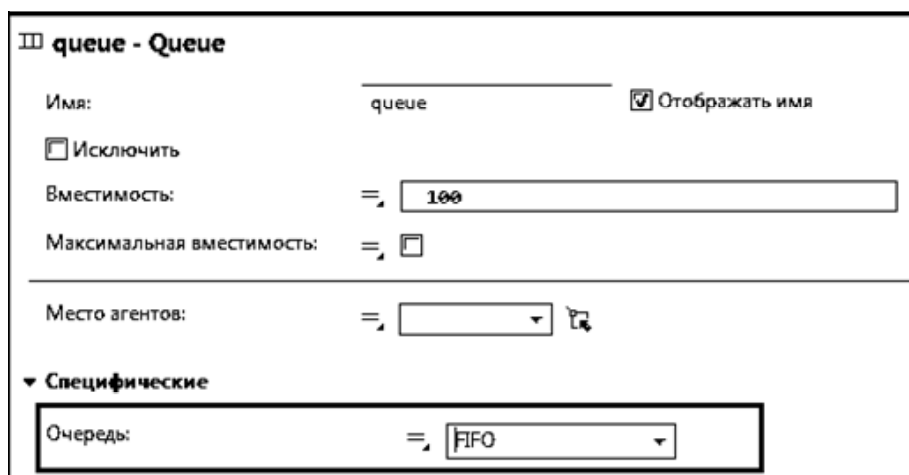


Рис. 4. Настройка параметров и алгоритма обработки очереди в программе AnyLogic
Fig. 4. Configuration of queue processing parameters and algorithm in AnyLogic

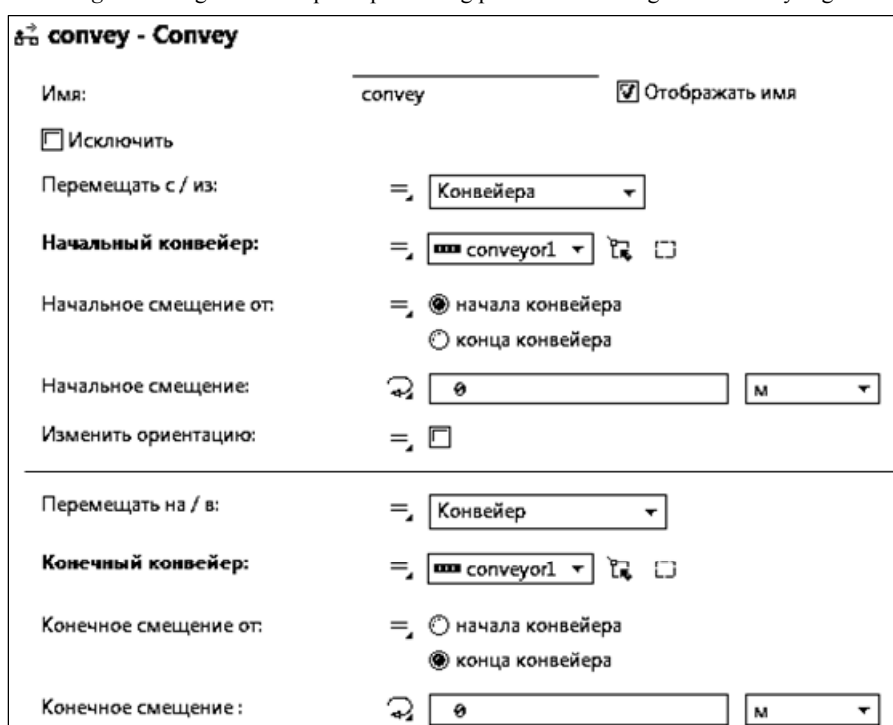


Рис. 5. Настройка параметров конвейера в программе AnyLogic
Fig. 5. Configuring pipeline of parameters in AnyLogic

С помощью ссылки `subs[.]` класс `Main` использует данные класса `Sub`, который служит для управления абонентами и подключениями их к вышкам. В ссылке переменным класса `Sub` устанавливаются координаты для начального расположения на поле класса `Main` и генерируется номер абонента от 1 до 6 для дальнейшего случайного выбора на конвейере.

С помощью ссылки `base[.]` класс `Main` использует данные класса `Base`, который служит для управления вышками. В `base[.]` задается начальное количество станций и с помощью вспомогательной переменной `index` станции располагают-

ся в соответствии с заранее объявленными в переменной `baseLoc` координатами.

Класс `Sub` включает в себя 4 параметра:

- 1) `type` – для определения типа генерируемого абонента;
- 2) `x` ($= -15$) – задает начальное положение абонента по оси $0x$;
- 3) `y` ($= -15$) – задает начальное положение абонента по оси $0y$;
- 4) `corner` – задает угол поворота абонента.

Также в классе `Sub` есть функция `DrawLine`, которая рисует линии соединения абонента и вышки. Рисование происходит посредством кода

```
ShapeLine newLine = new ShapeLine(SHAPE_DRAW_2D3D, true, 0, 0, 0.0, colorList.get(type-1), dx, dy, 0.0, 1.0, 10.0, LINE_STYLE_SOLID);
```

```
presentation.add(newLine);
line=newLine;
```

В классе Sub имеется коллекция colorList, в которой хранятся цвета вырисовываемых линий. В классе Sub расположена переменная line для хранения линий.

Еще в классе Sub присутствует диаграмма состояний с состоянием State и переходом transition. Переход необходим для удержания линии на конкретной вышке, с которой соединен абонент. Подобное действие выполняется благодаря коду

```
if(line != null) line.setDx(line.getDx()-1.9499070602);
```

Класс Sub содержит в себе код, который определяет номер станции через переменную index случайным образом; далее в станции отыскивается случайная вышка, обводится шестиугольником (что демонстрирует соединение абонента именно с ней); для коллекции colorList задает цвета линий, вызывает функцию DrawLine.

Класс Base состоит из одной переменной ceil, которая служит для обращения к определенной вышке.

Общая модель (см. рис. 1) представляет собой несколько станций, распределенных по миру. Далее будет рассмотрена частная модель (рис. 6), которая представляет собой одну станцию с шестью вышками, которые в свою очередь имеют

разные параметры – нагрузка, количество поступающих пакетов, приоритет и т. д.

На вышки поступает несколько видов данных, сгруппированных по потреблению трафика – это FTP-протоколы, VoIP (телефония), видеофайлы и 2 вида сенсоров: Time Driven (с регулярной отчетностью) и Event Driven (событийные) (1 на рис. 6).

Данные виды сигналов поступают на три вышки согласно заданной нагрузке и их приоритету (на рис. 6 вышки 2 приоритетами обозначены как Slice1–Slice3 с 1, 5, 10 соответственно). Узел RN соединяет все вышки и служит счетчиком пакетов 3, проходящих через него от одних вышек к другим.

Таким образом, с трех вышек через ретранслирующий узел пакеты данных поступают на три другие вышки (4 на рис. 6), на которых пакеты обрабатываются в соответствии с приоритетом. На последних вышках проверяется качество обслуживания – потеря пакетов (5 на рис. 6) и задержки в передаче информации. Модель станции 5G сети представлена на рис. 6.

Модель содержит один класс Main, который располагает в себе группы элементов. На рис. 6 расположены пять «начальных блоков» Source 1, которые создают агенты: Eventdrive, Regular, Video, VoIP, FTP. Эти блоки показывают подаваемые на вышки типы данных. Для каждого блока установлена интенсивность прибытия, а также с помощью кода uniform_discr(*,*) задано количество агентов, прибывающих в секунду.

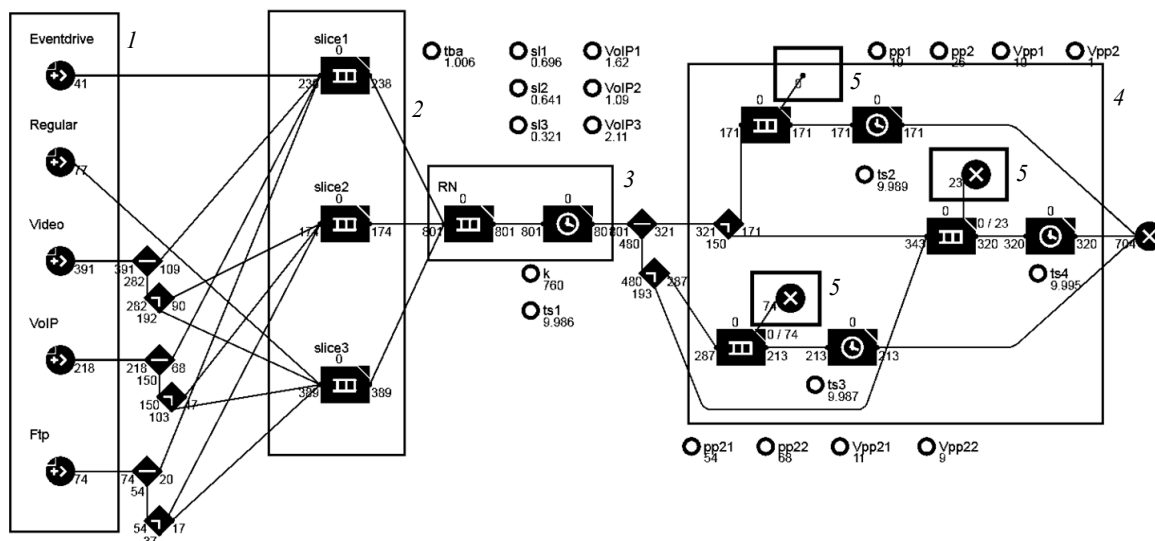





Рис. 6. Модель станции 5G в программе AnyLogic
Fig. 6. Model of 5G station in AnyLogic

На рис. 6 расположено 9 блоков Select Output, они отмечены символами  и . Их задача – направлять входящие агенты в один из двух выходных портов согласно заданным вероятностям или в зависимости от того, выполняется ли заданное условие. Данные блоки используются для перенаправления потоков данных на разные вышки в зависимости от параметров этих вышек (нагрузки, приоритета).


Для шести блоков Select Output (находящихся между областями 1 и 2 на рис. 6) установлена одинаковая вероятность, равная 0.3. Для других трех блоков (между областями 3 и 5) установлены вероятности 0.4–0.6. Подобная вероятность устанавливается для обеспечения необходимой нагрузки на вышках.


На рис. 6 расположено 7 блоков Queue, они отмечены символами  и моделируют очередь из поступающих на них агентов, которые ждут дальнейшей обработки [17]. Данные блоки и представляют собой вышки. Первые три блока (область 2 рис. 6) принимают и передают информацию в определенном порядке, т. е. согласно приоритету (= 1, 5, 10 соответственно).

Четвертый блок (рис. 6, область 3) объединяет в себе все поступающие в него данные. Последние три блока (рис. 6, область 4) принимают и передают данные, основываясь не только на при-

оритете, но и на таймауте (рис. 7), т. е. задержке (= 1, 4, 15 соответственно).

Также для блоков Queue установлен параметр «вместимость», который показывает, сколько агентов может вмещать в себя конкретный блок.

На рис. 6 расположены 4 блока Delay, которые задерживают агентов на заданный период времени (отмечены символами ). Первый блок устанавливает время задержки для входов с вышек, он находится в области 3 рис. 6. Задержка устанавливается в виде кода `triangular(1, 4, 15)` (рис. 8). В трех последних блоках установлено время задержки (1, 4, 15 соответственно) для вышек из сегмента 4.

На рис. 6 расположены 4 блока Sink (отмечены символами ), которые подсчитывают поступающие в них пакеты данных. Три из них подсчитывают необработанные пакеты данных с трех вышек соответственно, последний подсчитывает суммарное количество обработанных пакетов с трех вышек.

На поле находится семь переменных:

- 1) `s11`, `s12`, `s13` хранят данные о нагрузке;
- 2) `k`, `ts1`, `ts2`, `ts3`, `ts4` служат для построения графиков.

Переменная `k` (для графика на рис. 9) считает данные с выходов Queue-блоков с помощью кода `k = queue4.size() + k`. Переменные `ts1`, `ts2`, `ts3`, `ts4`

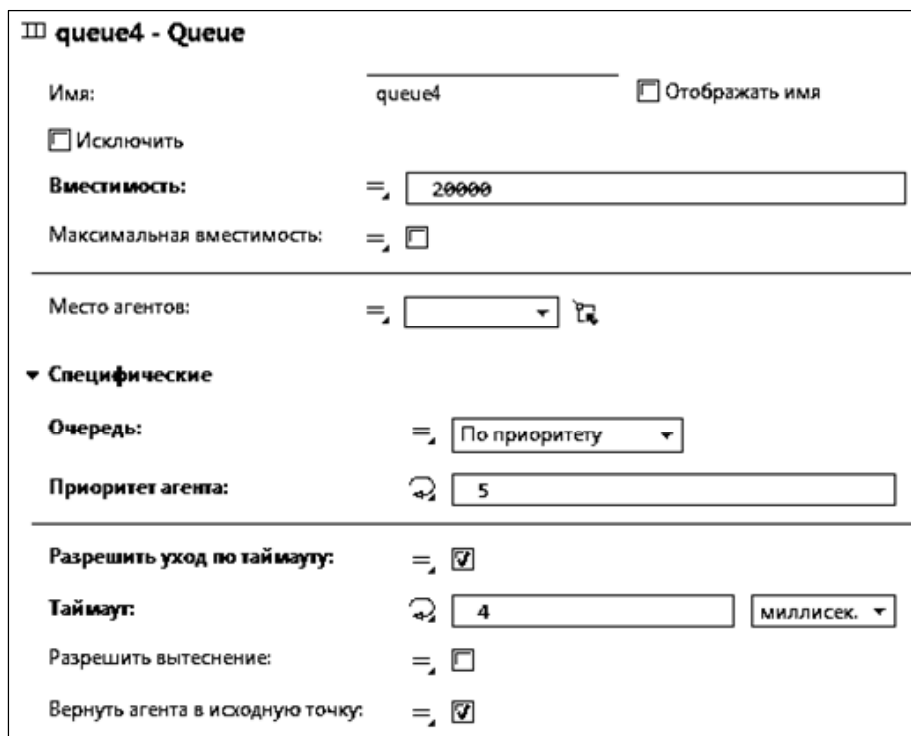


Рис. 7. Пример Queue-блока в программе AnyLogic
Fig. 7. Example of Queue-block in AnyLogic

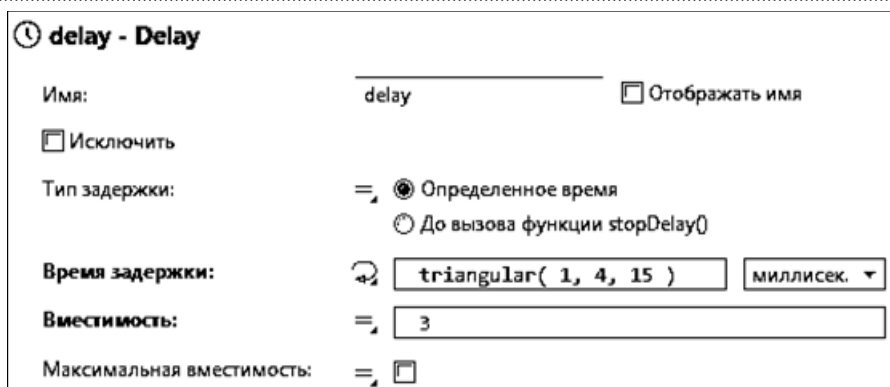


Рис. 8. Настройка задержки в программе AnyLogic
 Fig. 8. Delay setting in AnyLogic

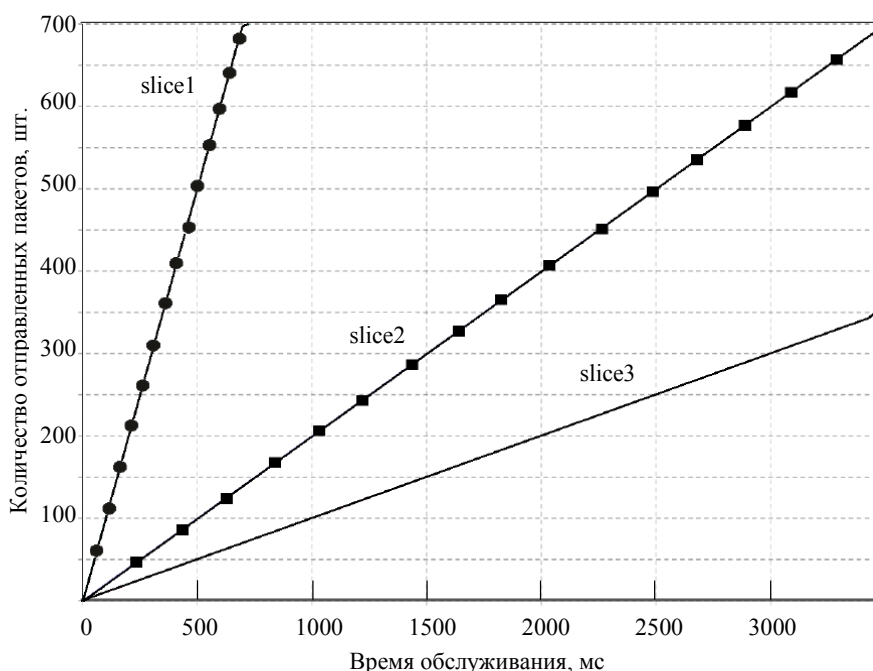


Рис. 9. График времени обслуживания
 Fig. 9. Graph of time service

Результаты моделирования
 Simulation results

Характеристика	Вышки 5G		
	slice1	slice2	slice3
Настраиваемая нагрузка, числовой коэффициент	0.7	0.6	0.3
Настраиваемый приоритет, числовой коэффициент	1	5	10
Настраиваемая максимальная задержка, мс	1	4	15
Количество отправленных пакетов, шт.	238	174	389
Количество потерянных пакетов, шт.	0	23	74
Время обслуживания, мс	0.003	0.004	7.712

считают данные при уходе по таймауту с помощью кода $ts1 = queue4.size() + ts1$.

На рис. 6 расположены 6 наборов данных, которые необходимы для построения графиков: VoIP1, VoIP2, VoIP3; sl1, sl2, sl3. В этих наборах задаются данные для осей $0x$ и $0y$.

Результаты моделирования. Основными параметрами исследования были время обслуживания пакетов и количество потерянных пакетов данных. Время обслуживания рассчитывается из данных о нагрузке на конкретном сегменте модели (вышке) и количества отправленных пакетов (которое рассматривается исходя из приоритета).

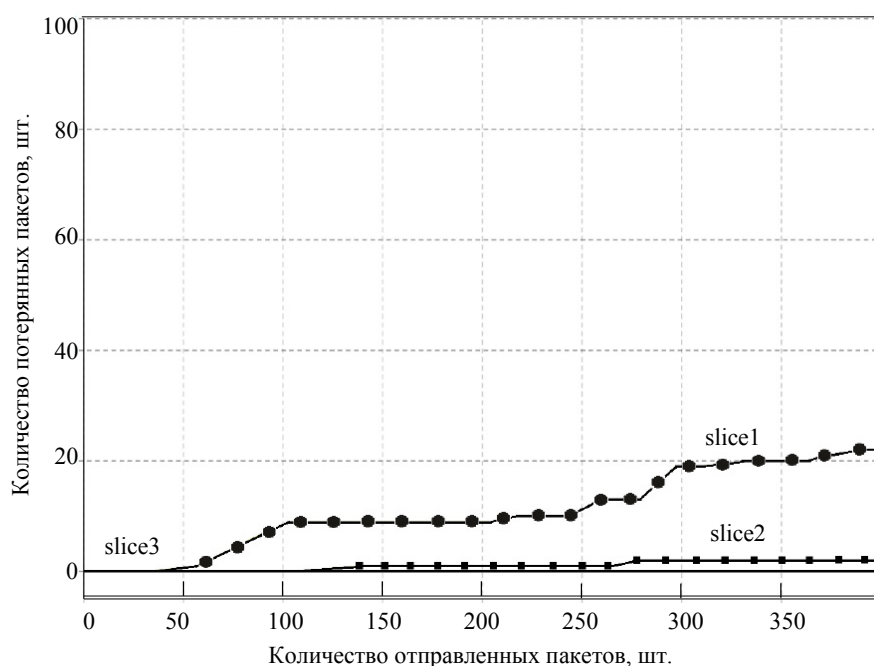


Рис. 10. График зависимости потерянных пакетов от обслуженных
Fig. 10. Graph of the dependence of lost packets on serviced

В данном случае расчет происходит за 40 с моделирования модели. Получаем формулу: $T_o = N/K_{po}$, где T_o – время обслуживания, N – нагрузка, K_{po} – количество отправленных пакетов.

Основываясь на данных о приоритете и времени обслуживания на конкретной вышке, можно объяснить количество потерянной информации. Пакеты теряются, когда при поступлении на определенную вышку и во время ожидания обработки установленного времени задержки не хватает на обработку предыдущих пакетов. Результаты моделирования представлены в таблице.

Таким образом, на модели можно видеть, что у первой вышки (рис. 6, область 4) самый высокий приоритет и большое время обслуживания по таблице и как следствие, на нее поступает мало пакетов, и они быстро обрабатываются, из этого видно, что потерь нет или они минимальны.

У второй вышки средний приоритет и среднее время обслуживания, т. е. пакеты успевают обрабатываться, но небольшая часть все равно теряется. У третьей вышки очень низкий приоритет и низкое

время обслуживания, в результате чего большое количество пакетов обрабатываются долго, при этом многие пакеты «не дожидаются» и теряются.

График времени обслуживания, создаваемый при работе модели, представлен на рис. 9; график, на котором видна зависимость потерянных пакетов от обслуженных на конкретных вышках, представлен на рис. 10.

Заключение. В статье были рассмотрены и исследованы принципы работы распределенных сетей 5G, проведено сравнение с другими сетями. С помощью данных принципов работы сети, а также изучения программного продукта AnyLogic были созданы различные модели сети станций 5G.

Была разработана модель, состоящая из одной базовой станции 5G и различными характеристиками. Проведена оценка эффективности передачи данных в сети и проанализировано влияние характеристик на вышки (задержки, нагрузки и приоритетов), на качество обслуживания пакетов данных, т. е. на скорость обработки пакетов и количество потерь пакетов.

Список литературы

1. Олифер В. Г., Олифер Н. А. Компьютерные сети. Принципы, технологии, протоколы. 4-е изд. СПб.: Питер, 2010. 943 с.
2. Никишин К. И. Механизм управления трафиком реального времени в коммутаторе Ethernet // Вестн. компьютерных и информационных технологий. 2015. № 10. С. 32–37.
3. Перепелкин Д. А., Бышов В. С. Балансировка потоков данных в программно-конфигурируемых сетях с обеспечением качества обслуживания сетевых сервисов // Радиотехника. 2016. № 11. С. 111–119.
4. Scheduling queues in the Ethernet switch, considering the waiting time of frames / E. Kizilov, N. Konnov,

K. Nikishin, D. Pashchenko, D. Trokoz // MATEC Web of Conferences. 2016. Vol. 44. P. 01011.

5. Nikishin K., Konnov N. Schedule time-triggered Ethernet // Intern. Conf. on Engineering Management of Communication and Technology, EMCTECH. Vienna: IEEE, 2020. P. 1–5. doi: 10.1109/EMCTECH49634.2020.9261540.

6. Advanced study of SDN/OpenFlow controllers / A. Shalimov, D. Zuikov, D. Zimarina, V. Pashkov, R. Smeiliansky // Proc. of the 9th Central & Eastern Europ. Software Engin. Conf. in Russia. ACM. Moscow, 2013. P. 1–6. doi: 10.1145/2556610.2556621.

7. Перепелкин Д. А. Концептуальный подход динамического формирования трафика программно-конфигурируемых телекоммуникационных сетей с балансировкой нагрузки // Информационные технологии. 2015. Т. 21, № 8. С. 602–610.

8. McKeown N., Anderson T., Balakrishnan H. Openflow: enabling innovation in campus networks // Comp. Communication Rev. 2008. Vol. 38, no. 2. P. 69–74.

9. Maturing of OpenFlow and Software-Defined networking through deployments / M. Kobayashi, S. Seetharaman, G. Parulkar, G. Appenzeller, J. Little, J. Van Reijendam, N. McKeown // Comp. Networks. 2014. Vol. 61. P. 151–175.

10. Никишин К. И. Моделирование контроллера и верификация процесса передачи данных в программно-конфигурируемых сетях // Вестн. Рязанского гос. радиотехнического ун-та. 2022. № 80. С. 75–83.

11. Ушакова М. В., Ушаков Ю. А. Исследование сети виртуальной инфраструктуры центра обработки данных с гибридной программно-конфигурируемой

коммутацией // Вестн. Рязанского гос. радиотехнического ун-та. 2021. № 75. С. 34–43. doi: 10.21667/1995-4565-2021-75-34-43.

12. Никульчев Е. В., Паяин С. В., Плужник Е. В. Динамическое управление трафиком программно-конфигурируемых сетей в облачной инфраструктуре // Вестн. Рязанского гос. радиотехнического ун-та. 2013. № 3 (45). С. 54–57.

13. Леохин Ю. Л., Фатхулин Т. Д. Оценка возможности предоставления гарантированной скорости передачи данных в программно-конфигурируемой оптической сети // Вестн. Рязанского гос. радиотехнического ун-та. 2020. № 71. С. 45–59. doi: 10.21667/1995-4565-2020-71-45-59.

14. Никишин К. И. Моделирование и верификация топологии программно-конфигурируемых сетей // Вестн. Рязанского гос. радиотехнического ун-та. 2022. № 80. С. 67–74.

15. Никишин К. И., Коннов Н. Н. Генератор трафика Ethernet на основе цветных сетей Петри // Модели, системы, сети в экономике, технике, природе и обществе. 2016. № 1 (17). С. 299–307.

16. Никишин К. И. Моделирование беспроводной сенсорной сети с использованием OMNET++ // Вестн. Рязанского гос. радиотехнического ун-та. 2021. № 78. С. 46–54.

17. Xiaolong J., Geyong M. Modelling and analysis of priority queueing systems with multi-class self-similar network traffic: a novel and efficient queue-decomposition approach // IEEE Transactions on Communications, 2007. Vol. 57. P. 1444–1452.

Информация об авторе

Никишин Кирилл Игоревич – канд. техн. наук, доцент, Пензенский государственный университет, ул. Красная, 40, Пенза, 440026, Россия.

E-mail: nkipnz@mail.ru

<https://orcid.org/0000-0001-7966-7833>

References

1. Olifer V. G., Olifer N. A. Komp'yuternye seti. Principy, tehnologii, protokoly. 4-e izd. SPb.: Piter, 2010. 943 s. (In Russ.).

2. Nikishin K. I. Mehanizm upravlenija trafikom real'nogo vremeni v kommutatore Ethernet // Vestn. komp'yuternyh i informacionnyh tehnologij. 2015. № 10. S. 32–37. (In Russ.).

3. Perepelkin D. A., Byshov V. S. Balansirovka potokov dannyh v programmno-konfiguriruemyh setjah s obespecheniem kachestva obsluzhivaniya setevyh servisoov // Radiotekhnika. 2016. № 11. С. 111–119. (In Russ.).

4. Scheduling queues in the Ethernet switch, considering the waiting time of frames / E. Kizilov, N. Konnov, K. Nikishin, D. Pashchenko, D. Trokoz // MATEC Web of Conferences. 2016. Vol. 44. P. 01011.

5. Nikishin K., Konnov N. Schedule time-triggered Ethernet // Intern. Conf. on Engineering Management of

Communication and Technology, EMCTECH. Vienna: IEEE, 2020. P. 1–5. doi: 10.1109/EMCTECH49634.2020.9261540. (In Russ.).

6. Advanced study of SDN/OpenFlow controllers / A. Shalimov, D. Zuikov, D. Zimarina, V. Pashkov, R. Smeiliansky // Proc. of the 9th Central & Eastern Europ. Software Engin. Conf. in Russia. ACM. Moscow, 2013. P. 1–6. doi: 10.1145/2556610.2556621. (In Russ.).

7. Perepelkin D. A. Konceptual'nyj podhod dinamicheskogo formirovaniya trafika programmno-konfiguriruemyh telekommunikacionnyh setej s balansirovkoj nagruzki // Informacionnye tehnologii. 2015. Т. 21, № 8. С. 602–610. (In Russ.).

8. McKeown N., Anderson T., Balakrishnan H. Openflow: enabling innovation in campus networks // Comp. Communication Rev. 2008. Vol. 38, no. 2. P. 69–74.

9. Maturing of OpenFlow and Software-Defined networking through deployments / M. Kobayashi, S. Seetharaman, G. Parulkar, G. Appenzeller, J. Little, J. Van Reijendam, N. McKeown // *Comp. Networks*. 2014. Vol. 61. P. 151–175.

10. Nikishin K. I. Modelirovanie kontrollera i verifikacija processa peredachi dannyh v programmno-konfiguriruemym setjah // *Vestn. Rjazanskogo gos. radiotekhnicheskogo un-ta*. 2022. № 80. S. 75–83. (In Russ.).

11. Ushakova M. V., Ushakov Ju. A. Issledovanie seti virtual'noj infrastruktury centra obrabotki dannyh s gibridnoj programmno-konfiguriruemoj kommutaciej // *Vestn. Rjazanskogo gos. radiotekhnicheskogo un-ta*. 2021. № 75. S. 34–43. doi: 10.21667/1995-4565-2021-75-34-43. (In Russ.).

12. Nikul'chev E. V., Pajain S. V., Pluzhnik E. V. Dinamicheskoe upravlenie trafikom programmno-konfiguriruemym setej v oblachnoj infrastrukture // *Vestn. Rjazanskogo gos. radiotekhnicheskogo un-ta*. 2013. № 3 (45). S. 54–57. (In Russ.).

13. Leohin Ju. L., Fathulin T. D. Ocenka vozmozhnosti predostavlenija garantirovannoj skorosti peredachi

dannyh v programmno-konfiguriruemoj opticheskoj seti // *Vestn. Rjazanskogo gos. radiotekhnicheskogo un-ta*. 2020. № 71. S. 45–59. doi: 10.21667/1995-4565-2020-71-45-59. (In Russ.).

14. Nikishin K. I. Modelirovanie i verifikacija topologij programmno-konfiguriruemym setej // *Vestn. Rjazanskogo gos. radiotekhnicheskogo un-ta*. 2022. № 80. S. 67–74. (In Russ.).

15. Nikishin K. I., Konnov N. N. Generator trafika Ethernet na osnove cvetnyh setej Petri // *Modeli, sistemy, seti v jekonomike, tehnike, prirode i obshhestve*. 2016. № 1 (17). С. 299–307. (In Russ.).

16. Nikishin K. I. Modelirovanie besprovodnoj sensornoj seti s ispol'zovaniem OMNET++ // *Vestn. Rjazanskogo gos. radiotekhnicheskogo un-ta*. 2021. № 78. S. 46–54. (In Russ.).

17. Xiaolong J., Geyong M. Modelling and analysis of priority queueing systems with multiclass self-similar network traffic: a novel and efficient queue-decomposition approach // *IEEE Transactions on Communications*, 2007. Vol. 57. P. 1444–1452.

Information about the author

Kirill I. Nikishin – Cand. Sci. (Eng.), Associate Professor of Penza State University, Krasnaya str., 40, Penza, 440026, Russia.

E-mail: nkipnz@mail.ru

<http://orcid.org/0000-0001-7966-7833>

Статья поступила в редакцию 27.12.2022; принята к публикации после рецензирования 24.01.2023; опубликована онлайн 25.03.2023.

Submitted 27.12.2022; accepted 24.01.2023; published online 25.03.2023.
