

МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ,  
СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Санкт-Петербургский государственный университет телекоммуникаций  
им. проф. М.А. Бонч-Бруевича»

На правах рукописи

Кузнецов Константин Алексеевич

**ИССЛЕДОВАНИЕ И РАЗРАБОТКА МЕТОДОВ ПРЕДОСТАВЛЕНИЯ  
УСЛУГ ТЕЛЕПРИСУТСТВИЯ В СЕТЯХ СВЯЗИ ШЕСТОГО ПОКОЛЕНИЯ**

2.2.15. Системы, сети и устройства телекоммуникаций

Диссертация на соискание ученой степени  
кандидата технических наук

Научный руководитель  
доктор технических наук  
Мутханна Аммар Салех Али

Санкт-Петербург - 2024

**ОГЛАВЛЕНИЕ**

ОГЛАВЛЕНИЕ .....	2
ВВЕДЕНИЕ.....	4
ГЛАВА 1. МЕТАВСЕЛЕННЫ И МЕТОДЫ ИХ РЕАЛИЗАЦИИ В СЕТЯХ СВЯЗИ ШЕСТОГО ПОКОЛЕНИЯ .....	9
1.1    Интегрированная архитектура сети «воздух-земля».....	11
1.2    Воздушные и наземные сети в 6G-SAGIN .....	13
1.2.1 Воздушные сети .....	13
1.2.2 Наземные сети .....	17
1.3    Правовые аспекты применения метавселенных .....	32
ГЛАВА 2. ИНТЕЛЛЕКТУАЛЬНОЕ ЯДРО ИНТЕГРИРОВАННОЙ СЕТИ 6G НА ОСНОВЕ ТЕХНОЛОГИИ SDN ДЛЯ ПОДДЕРЖКИ ПРИЛОЖЕНИЙ ТЕЛЕПРИСУТСТВИЯ.....	37
2.1    Архитектура сети .....	39
2.2    Формулировка задачи.....	43
2.3    Результаты моделирования .....	47
2.4    Вывод по главе 2 .....	50
ГЛАВА 3. МЕТОДЫ РАЗМЕЩЕНИЯ МАРШРУТИЗАТОРОВ СЕТИ НА БЕСПИЛОТНЫХ ЛЕТАТЕЛЬНЫХ АППАРАТАХ (БПЛА) И ОБЕСПЕЧЕНИЯ МАРШРУТИЗАЦИИ С ПОМОЩЬЮ РОЯ БПЛА .....	51
3.1    Модель и постановка задачи.....	54
3.2    Выбор позиции маршрутизатора.....	55
3.3    Маршрутизация трафика.....	65
3.4    Эффективность .....	67

3.5 Вывод по главе 3 .....	69
ГЛАВА 4. АЛГОРИТМ МИГРАЦИИ СЕРВИСОВ ДЛЯ РАСПРЕДЕЛЕННЫХ ГРАНИЧНЫХ ВЫЧИСЛЕНИЙ В СЕТЯХ 5G/6G.....	71
4.1 Технологии развертывания и миграции сервисов.....	72
4.2 Исследование существующих технологии миграции услуг.....	77
4.2.1 Анализ средств конфигурирования оркестровки .....	77
4.2.2 Анализ процесса миграции, времени полного переноса и времени простоя.....	79
4.3 Метод автоматической миграции сервисов .....	81
4.3.1 Разработка алгоритма для автоматизации миграции сервисов.....	84
4.4. Вывод по главе 4 .....	93
ГЛАВА 5. МЕТОД РЕАЛИЗАЦИИ ИММЕРСИВНЫХ ТЕХНОЛОГИЙ В СЕТЯХ СВЯЗИ ПЯТОГО И ШЕСТОГО ПОКОЛЕНИЙ, ПОЗВОЛЯЮЩИЙ В РЕЖИМЕ РЕАЛЬНОГО ВРЕМЕНИ ПЕРЕДАВАТЬ ЖЕСТОВУЮ РЕЧЬ.....	95
5.1 Описание стенда.....	96
5.2 Противодействие задержкам .....	104
5.2.1 Эксперимент миграции функциональных элементов сети.....	104
5.2.2 Эксперимент миграции стенда .....	109
ЗАКЛЮЧЕНИЕ .....	114
СПИСОК ЛИТЕРАТУРЫ.....	116
Приложение А - исполняемый код ESP-32 .....	126
Приложение Б - исполняемый код Processing.....	133
Приложение В – акты о внедрении результатов исследования .....	138

## ВВЕДЕНИЕ

**Актуальность тема исследования.** За последние 20 лет в сфере развития сетей связи произошел значительный прогресс. Появилось множество новых технологий, которые сначала позволили предоставлять пользователям широкий спектр услуг, включая развитие мобильной телефонии. Затем они обеспечили возможность передачи данных и видео, а сейчас уже привели к изменениям в принципах построения сетей и в услугах, доступных для пользователей. При этом ключевую роль в формировании нового облика сетей связи и предоставляемых сетью услугах сыграли концепции Интернета Вещей и Тактильного Интернета. Первая способствовала появлению сетей высокой и сверхвысокой плотности, а вторая – сетей связи с ультрамалыми задержками. Все это вместе взятое потребовало комплексного использования ресурсов всевозможных сетей и преобразовало сети связи из гомогенных в гетерогенные.

В начале третьего десятилетия 21 века стало ясно, что нескоординированное развитие мобильных и фиксированных сетей связи не способствует решению проблемы интегрирования всех ресурсов всех сетей для предоставления современных услуг всем пользователям сетей связи общего пользования и на этапе формирования подходов к реализации сетей связи шестого поколения 6G появилась новая концепция развития сетей связи, в основе которой лежит понимание необходимости интеграции не только разнообразных технологий в рамках тех или иных сетей, но и интеграции сетей связи в единую сеть. Эта концепция называется интегрированные сети Космос-Воздух-Земля SAGIN (Space-Air-Ground). Представляется, что эта концепция определяет множество научных проблем и задач, по крайней мере, на ближайшее десятилетие. В связи с изложенным диссертационная работа, в которой решается научная проблема разработки и исследования комплекса моделей и методов интеграции граничных и туманных вычислений в сетях связи пятого и шестого поколений для глобального фрагмента Воздух-Земля концепции SAGIN является актуальной.

**Разработанность темы исследования.** Существует множество работ в области сетей 5G и последующих поколений, высокоплотных и сверхплотных сетей, сетей связи с ультрамалыми задержками как теоретического, так и экспериментального плана. Определяющий вклад в теоретические и экспериментальные исследования этих научных проблем внесли российские и зарубежные ученые: В. М. Вишнеvский, К.Е.Самуйлов, Ю. В. Гайдамака, Б. С. Гольдштейн, В. Г. С. Н. Степанова, Карташевский, А. И. Парамонов, А. Е. Кучеряvый, Е.А.Кучеряvый, М. С. Маколкина, Д.А.Молчанов, Р. В. Киричек, А. П. Пшеничников, В. К. Сарьян, С. Н. Степанов, М. А. Сиверс, Н. А. Соколов, В. О. Тихвинский, М. А. Шнепс-Шнеппе, M. Dohler, G. Fettweis, J. Hosek, A. A. Ateya, M. Maier, M. Z. Shafiq и другие.

Работы отмеченных выше авторов внесли весомый вклад в исследования характеристик сетевого трафика и качества обслуживания для сетей связи пятого и шестого поколений, в основном для услуг Интернета Вещей, передачи данных, видео, телеконференций и т.п. Однако до настоящего времени практически не уделялось внимания исследованию характеристик трафика и качества обслуживания для услуг телеприсутствия и методам их реализации сети связи с ультрамалыми задержками. Это и определяет цель, задачи, объект и предмет диссертационной работы. При этом решается следующая научная задача: исследование и разработка методов обеспечения услуг телеприсутствия в сетях связи шестого поколения.

**Объект и предмет исследования.** Объектом исследования являются сети связи пятого и шестого поколений. Предметом исследования являются сети связи шестого поколения на основе БПЛА и программно-конфигурируемых сетей и предоставление услуг телеприсутствия.

**Цель и задачи исследования.** Целью диссертационной работы является исследование и разработка методов обеспечения услуг телеприсутствия в сетях связи шестого поколения.

**Научная новизна.** Научная новизна работы состоит в следующем:

1. Предложен метод оптимизации числа контроллеров в мультиконтроллерных сетях SDN, который позволяет снизить среднее число контроллеров на 46% за счет динамического управления, что уменьшает энергопотребление и затраты на развертывание, а также повышает надежность сети.
2. Разработан модельно-методический аппарат, позволяющий повысить эффективность построения сетей беспроводного доступа с применением БПЛА, в частности позволяющий выбирать позиции размещения маршрутизаторов в рое БПЛА и логическую структуру сети. Разработанный модельно-методический аппарат решает задачу маршрутизации трафика с учетом качества его обслуживания.
3. Предложен метод передачи языка жестов, как элемент костюма телеприсутствия для людей с ограниченными возможностями.

**Теоретическая значимость работы.** Теоретическая значимость диссертационной работы состоит в разработке новой модели и метода, которые позволяют произвести распределение маршрутизаторов (точек доступа) сети, размещенных на БПЛА с учетом качества обслуживания и обеспечения построения связанной mesh-сети и ее связи с сетью подвижной связи, которые могут быть использованы как в современных, так и перспективных сетях связи; предложен метод реализации иммерсивных технологий в сетях связи пятого и шестого поколений.

**Практическая ценность работы.** Практическая ценность работы состоит в разработке методика планирования мультиконтроллерных сетей на основе программно-конфигурируемых сетей SDN, внедренной в ПАО «ГИПРОСВЯЗЬ», предложенной модели и методах, которые могут быть использованы при организации обслуживания в сетях 5G и последующих поколений, в частности, они позволяют обеспечить доступность связи и оперативность организации сети в случаях недостаточного покрытия, а также в случаях выхода из строя отдельных элементов сети.

**Реализация результатов работы.** Полученные в диссертационной работе результаты внедрены в ПАО «ГИПРОСВЯЗЬ» при разработке методики

планирования мультиконтроллерных программно-конфигурируемых сетей SDN, в Санкт-Петербургском государственном университете телекоммуникаций им. проф. М.А. Бонч-Бруевича при проведении работ по Мегагранту “Исследования сетевых технологий с ультрамалой задержкой и сверхвысокой плотностью на основе широкого применения искусственного интеллекта для сетей 6G” по соглашению № 075-15-2022-1137 с Министерством науки и высшего образования РФ, чтении лекций, проведении практических занятий и лабораторных работ. Акты реализации диссертационных исследований представлены в Приложении В.

**Методы исследования.** При проведении исследований были использованы методы теории телетрафика и теории массового обслуживания, теории оптимизации, теории вероятностей, а также метаэвристические алгоритмы и методы имитационного моделирования.

**Основные положения, выносимые на защиту.**

1. Метод построения интеллектуального ядра интегрированной сети 6G на основе технологии SDN для услуг телеприсутствия, снижающий среднее число контроллеров в мультиконтроллерных сетях SDN на 46% и увеличивающий при этом эффективность использования контроллеров на 53%.
2. Модель и методы размещения маршрутизаторов сети на беспилотных летательных аппаратах (БПЛА) и обеспечения маршрутизации с помощью роя БПЛА, позволяющие увеличить эффективность использования сети на 50%.
3. Метод реализации иммерсивных технологий в сетях связи пятого и шестого поколений, позволяющий в режиме реального времени передавать жестовую речь, и требования к качеству обслуживания и качеству восприятия таких услуг.

**Достоверность результатов.** Степень достоверности полученных результатов подтверждается корректным применением математического аппарата, результатами натурного и имитационного моделирования, а также спектром

публикаций и выступлений как на российских, так и международных научно-технических конференциях.

**Апробация работы.** Основные положения диссертационной работы обсуждались и были одобрены на следующих конференциях, конгрессах и семинарах: Международной конференции по проводным и беспроводным сетям и системам следующего поколения NEW2AN (Дубай, 2023); Международной конференции “Распределенные компьютерные и телекоммуникационные сети: управление, вычисления, связь” DCCN (Москва, 2023); Международной научно-технической и научно-методической конференции “Актуальные проблемы инфотелекоммуникаций в науке и образовании” АПИНО (Санкт-Петербург, 2020, 2022); Научно-технической конференции Санкт-Петербургского НТО РЭС им. А.С. Попова, посвященной Дню радио (Санкт-Петербург, 2024).

**Публикации.** Основные результаты диссертации изложены в 11 опубликованных работах, в том числе в 2 работах, опубликованных в журналах из перечня ВАК Министерства образования и науки Российской Федерации; 2 работах в изданиях, включенных в международные базы цитирования; 7 работах других научных изданий и материалов конференций.

**Соответствие паспорту специальности.** Диссертационная работа выполнена по специальности 2.2.15. Системы, сети и устройства телекоммуникаций и соответствует следующим пунктам паспорта специальности: 2, 4, 9, 10, 12.

**Личный вклад автора.** Все основные результаты диссертационной работы получены автором самостоятельно.

## ГЛАВА 1. МЕТАВСЕЛЕННЫ И МЕТОДЫ ИХ РЕАЛИЗАЦИИ В СЕТЯХ СВЯЗИ ШЕСТОГО ПОКОЛЕНИЯ

На протяжении нескольких десятилетий происходит трансформация и непрерывное развитие технологий дополненной и виртуальной реальности, активно разрабатываются способы передачи иных иммерсивных технологий. Так, еще в 1957 году для рекреационных целей была создана “Sensorama”. Этот кинематографический опыт, разработанный Мортонем Хейлигом, помещал зрителя в "сенсорный" театр, включающий динамики, вентиляторы, генераторы запахов и вибрирующее кресло для погружения зрителя в фильм.

Иммерсивные технологии — это набор различных методов и средств, позволяющих воздействовать на спектр человеческих ощущений или на его часть. В последнее время такие технологии набирают все большую популярность и стремительно входят в обыденную жизнь, так, например, VR и AR уже активно применяются в строительстве, машиностроении и других отраслях промышленности. По мере развития иммерсивных технологий логичным образом появляется спрос на передачу человеческих ощущений на расстоянии, то есть технология телеприсутствия, которая позволяет человеку «транспортироваться» из одного физического пространства в другое через телекоммуникационную сеть, управляя доступом и находясь в местах, как если бы он находился там физически, но испытывая это через виртуальное присутствие [1].

Одной из первых технологий нацеленных на передачу ощущений на расстоянии стал тактильный Интернет, который является результатом эволюции технологии Интернета Вещей, позволяющей передавать на расстоянии не только видеок cadры и речевую информацию, но и ощущения эквивалентные человеческим прикосновениям. Термин и понятие технологии появилось в 2011 году в Дрезденском техническом университете. Профессор мобильных коммуникаций - Герхардом Фетвайсом описал свою идею так: *“Таким способом, вы можете,*

*например, ловить падающий объект дистанционно или управлять подключенным автомобилем на перекрестке. Если вы предоставляете тактильную обратную связь, вы также можете почувствовать реакцию, которая кажется мгновенной. Тактильный Интернет будет использоваться в таких областях, как автоматизация, образование, развлечения, игры, сельское хозяйство, здравоохранение и промышленный транспорт. Это также позволит людям управлять роботами удаленно в режиме реального времени.”*

В то время, как тактильный Интернет позволяет нам “перемещаться” в пространстве, метавселенные представляют собой виртуальные миры, в которых цифровые объекты сливаются (взаимодействуют) с физическими вещами и событиями. Metaverse - сочетание слов «meta» (означает «за пределами») и «verse» (от слова universe, вселенная) — это парадигма взаимодействия в сети Интернет нового поколения, в которой пользователи взаимодействуют с программными приложениями и другими пользователями как аватары в трехмерном (3D) виртуальном мире, уделяя особое внимание социальным связям [2]. Термин «метавселенная» был впервые использован Нилом Стивенсоном в его научно-фантастическом романе Snow Crash в 1992 году, где люди используют технологию виртуальной реальности, чтобы попасть в метавселенную [3]. Метавселенная вновь стала популярным словом спустя более двадцати лет как «преемник мобильного интернета». С помощью виртуальной реальности (VR), дополненной реальности (AR), человеко-компьютерных интерфейсов (HCI) и тактильного интернета пользователи могут ощутить уникальное существование в виртуальном пространстве метавселенной, как в реальном мире [4]. С момента своего появления идея метавселенной описывалась по-разному, включая вторую жизнь, дополненную реальность, 3D-виртуальные миры, life-logging, цифровых людей и зеркальные миры.

На сегодняшний день уже существует множество цифровых миров, так, например, созданы копии таких городов, как: Сеул, Дубай, Абу-Даби, появляются магазины одежды и электроники, проводятся выставки искусства и техники. Разработчики компьютерных игр также не отстают в развии, к метавселенным

можно отнести такие игры, как: Decentraland, VR chat и др. Отдельным направлением развития стали “промышленны” мета-пространства, так автомобильный производитель Renault создал цифрового двойника одного из своих заводов с целью безболезненной оптимизации производственных процессов реального мира. Таким образом, метавселенная — это перспективное решение, объединяющее такие ключевые технологии, как AR, VR, MR, так и расширенную реальность (XR) в глобальном контексте [4].

### **1.1 Интегрированная архитектура сети «воздух-земля»**

Предполагается, что 6G вытеснит существующие поколения сетевых возможностей, предоставляя услуги лучшего качества. Мы можем ожидать, что в ближайшие дни 6G обеспечит супергибкость и высокую производительность. Начиная с 1981 года и по сегодняшний день в мире наблюдается постепенная эволюция мобильных и коммуникационных технологий. Начиная с 1G, 2G и 3G, в настоящее время мы наслаждаемся различными коммуникационными преимуществами сетей 4 и 5 поколений [6]. С таким ростом коммуникационных технологий, скорость передачи данных значительно увеличивается. Во время 1G мы использовали скорость передачи данных всего 2 Кбит/с, которая увеличилась до 64 Кбит/с в 2G, 8 Мбит/с в 3G, 50 Мбит/с в 4G и до 5 Гбит/с в 5G. Ожидается, что 6G обеспечит около 100 Тбит/с при реальном внедрении. Таким образом, частотный спектр плавно увеличился с 1G. Так, 5G имеет полосу пропускания около 30-300 ГГц. 6G превзойдет этот показатель и выйдет за пределы гигагерцового диапазона. Что касается аксессуаров, то IoT используется в качестве интеллектуального средства связи между устройствами уже в 4G. Мы ожидаем, что 6G будет включать в себя будущие технологии радиодоступа, OFDMcmW, OFDM-mmWave, связь на в терагерцовом диапазоне, ультра URLLC и бесшовную

интеграцию коммуникационных и вспомогательных технологий на основе SAGIN. Вместо того чтобы полагаться исключительно на узкий диапазон частот, как это было в предыдущих поколениях, 6G предполагает наземное широкополосное покрытие поверх обычных широкополосных, сверхширокополосных и беспроводных WWW-технологий [7].

6G расширит возможности многочисленных ключевых показателей эффективности (KPI), таких как скорость, надежность, низкая стоимость, емкость, связь, низкая задержка, доступность, охват, познание, зондирование, доверие и безопасность. Таким образом, 6G позволит пользователям и поставщикам сетевых услуг тесно сотрудничать для взаимной выгоды разворачиваемых приложений. В эпоху 6G могут появиться различные инновационные сценарии использования. Особенно в области применения SAGIN. Например, автоматизированные общественные и частные БПЛА с человеческим транспортом будут летать с помощью системы самонаведения и управления. Подводная связь может быть установлена с путешественником, который летит на воздушном шаре. Даже астронавт сможет играть со своими детьми в видеоигры сверхвысокой четкости (UHD) в реальном времени. Таким образом, мы можем ожидать, что объединение 6G-SAGIN преодолет все барьеры, с которыми приходится сталкиваться в настоящее время. Благодаря использованию тактильных технологий, когнитивных технологий, искусственного интеллекта и тактической обработки информации, 6G улучшит инфраструктуру SAGIN.

Однако для достижения такой ориентации 6G-SAGIN необходимо включить некоторые ключевые требования, такие как модернизация существующей связи H2M, сверхнадежные сети, связь с низкой задержкой и улучшенная мобильная широкополосная связь. Гарантированное качество обслуживания с надежностью 99,999999% будет необходимостью. Необходимо задействовать новые диапазоны спектра при использовании  $>100$  Гбит/с пиковой скорости передачи данных. Экстремальное покрытие спектра в воздухе, космосе и под водой. Потребление энергии должно быть минимизировано до такого уровня, чтобы позволить себе нановаттные затраты в mmWave/ТГц. Минимальное вмешательство должно

зависеть от схемы зарядки аккумулятора. Должна быть достигнута сверхмассивная дальность подключения устройств ( $>100 \text{ М/км}^2$ ). Необходимо обеспечить высокоточное позиционирование (порядка мм/см) с расширенными возможностями зондирования [8]. На рисунке 1.1 представлена архитектура 6G-SAGIN.

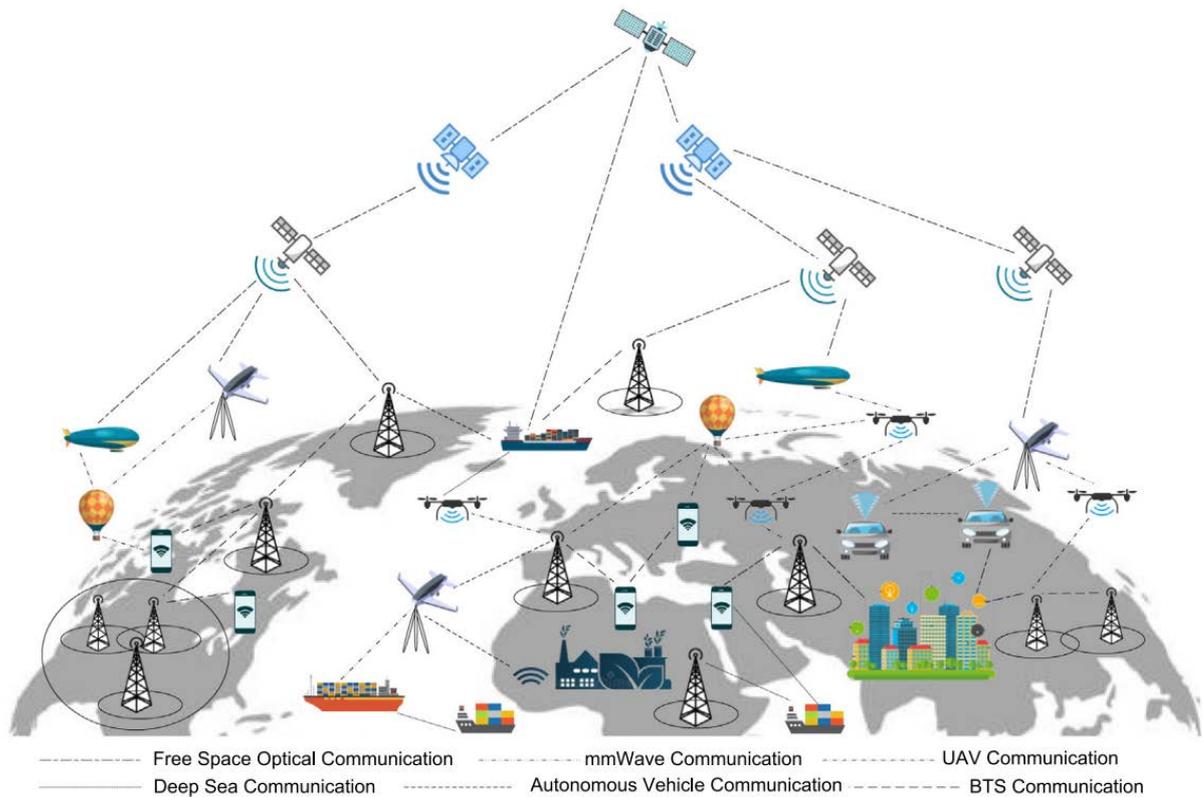


Рисунок 1.1 - Архитектура SAGIN

## 1.2 Воздушные и наземные сети в 6G-SAGIN

### 1.2.1 Воздушные сети

Сеть космического доступа (SAN) предназначена для полноценного отслеживания и ретрансляции спутниковых и пространственных устройств

передачи данных. SAN может быть использована в 6G-SAGIN для установления связи между различными коммуникационными элементами. Обычно для обеспечения доступа к сети используются геостационарные спутники, покрывающие Ku-диапазон. Скорость передачи данных составляет около 506 Мбит/с. Она может покрывать 100-6000 км пространственной области. В него могут быть интегрированы модемы и антенны, размещенные как в помещении, так и на улице. В последнее время для облегчения связи используются средние и низкоорбитальные спутники [9]. Отмечается, что низко и среднеорбитальные спутники обеспечивают скорость 64 Кбит/с и 1 Гбит/с, соответственно. Спутниковый доступ в интернет (SIA) также иногда ассоциируется с SAN, где двусторонняя спутниковая связь является нормальной.

Однако в последнее время развиваются портативные спутниковые модемы и использование интернета через спутниковый телефон. С другой стороны, односторонние приемники с наземными передатчиками используются с небольшими изменениями в режиме вещания. Такие системы могут обеспечить работу 100-4000 конечных пользователей в любой момент времени. Этот диапазон может быть расширен с помощью соответствующих изменений в программном обеспечении и схемах множественного доступа с временным разделением каналов. Эта технология сталкивается с некоторыми ключевыми проблемами, которые включают в себя задержку сигнала, размещение на геостационарных орбитах, помехи со стороны атмосферы, зоны Френеля и линии прямой видимости. Однако ожидается, что использование SAN и SIA вместе принесет пользу 6G-SAGIN.

Спутниковая связь будет органично интегрирована в экосистему 6G. Благодаря высокому радиусу охвата и чрезвычайно точным аспектам связи спутниковая связь станет неотъемлемой частью 6G. В основном в цифровой связи участвуют три типа спутников: низкая околоземная орбита (LEO), средняя околоземная орбита (MEO) и геостационарная околоземная орбита (GEO). Спутники LEO вращаются на высоте от 160 до 2000 км над Землей. Спутники MEO вращаются на высоте от 2000 до 36000 км над Землей. Спутники GEO имеют стационарную орбиту 22236 км над поверхностью Земли. Различные виды

спутниковой связи используются для обеспечения ряда приложений, например, фиксированная спутниковая служба (FSS) предназначена для трансляции сигналов телестанций, радиоканалов и вещательных сетей [10]. Услуги связи с Землей и космосом обычно предоставляются FSS. Мы видим, что диапазоны 14-14,25 ГГц и 14,25-14,3 ГГц используются для радионавигации и космических исследований. Вещание телевизионных каналов может осуществляться на частотах 12-18 ГГц с помощью небольшой антенны с радиусом  $<0,5$  м. Стандарты цифрового видеовещания (DVB), включая DVB-T, S, C и H, очень полезны для передачи данных через телевизионные приставки. Мобильная спутниковая служба (MSS) обеспечивается за счет использования различных частотных диапазонов, таких как 137-136,025 МГц, 137,025-137,175/137,825 МГц для связи «космос - земля» (S2E) и 148-149,9/150,05 МГц и 156,7625-156,8375 МГц для связи «земля - земля» (E2E). Таким образом, 6G может использовать спутники для выполнения приложений, связанных с мобильной передачей данных, ТВ-вещанием, радиовещанием, военными и широкополосными интернет-услугами. Для получения доступа к радиослугам непосредственно со спутников может быть использована технология спутниковой RAN (SARN). Связь в свободном пространстве с помощью лазера — это признанный метод создания сетей. Оптические телескопы могут использоваться в качестве связующего звена между наземными и космическими элементами для обеспечения дальней связи. В настоящее время ведется несколько исследовательских проектов, направленных на реализацию лазерной космической связи с использованием различных сред. Низкая околоземная орбита, средняя околоземная орбиты и геостационарной орбиты лазерная связь является наиболее распространенной. Однако в настоящее время изучаются схемы, ориентированные на стратосферу и тропосферу. Обычно тестируются четыре типа сценариев: «космос - земля» (S2G), «космос - космос» (S2S), «воздух - воздух» (A2A) и «воздух - земля» (A2G). В среднем при передаче сигнала достигается скорость передачи данных 1 Гбит/с. Однако она может варьироваться и снижаться до минимальных 0,15 Гбит/с с максимальным пределом до 10 Гбит/с. Цели, стоящие за такими реализациями, следующие: создание глобальной телекоммуникационной

магистрала, спутниковая мегаконstellляция, проектирование гибридной сети ретрансляции данных с поддержкой радиочастот, покрытие сетью сельских и удаленных районов и связь в полете [11].

Другой вид космической связи, а именно оптическая связь в глубоком космосе (DSOC), может рассматриваться как ключевой фактор развития 6G-SAGIN. DSOC может улучшить существующие характеристики связи до 100 раз. Предполагается, что DSOC позволит использовать очень высокую пропускную способность для операций нисходящей связи. Разработанная в лаборатории NASA JPL, система DSOC, как ожидается, будет использовать ближний инфракрасный диапазон (NIR), т.е. длину волны 1,55 мкм при передаче оптических сигналов. Предполагается, что метод DSOC будет интегрирован в миссию Psyche NASA в 2022 году. Система будет состоять из двух блоков бортового лазерного излучателя и наземной станции. Лазерный блок спутника будет фокусировать лазер мощностью 4 Вт от телескопа с апертурой 22 см. При общей мощности модуля передатчика менее 100 Вт операции восходящей и нисходящей линии связи будут согласованы на наземной станции [12]. В режиме восходящей линии связи наземная станция будет использовать 1,064 мкм. Ожидается, что средняя скорость передачи данных составит 292 Кбит/с, таким образом 6G-SAGIN может быть улучшена благодаря вмешательству DSOC.

С целью поддержания оптической связи между космосом и наземной станцией была задумана оптическая система Payload for space communication (OPASC). Однако мы считаем ее подходящей для возможного использования в сценарии 6G-SAGIN. OPASC способен обрабатывать искаженные сигналы в космической связи. Это очень полезно для разработки процедурных порций с заданными оптическими каналами. Обычная скорость передачи данных составляет около 50 Мбит/с. 6GSAGIN будет использовать OPASC для использования в работе миссии и прикладных задач. С помощью систем отслеживания данных и ретрансляции (TDRSS) OPASC начнет взаимодействовать между спутником и наземной станцией. В этом аспекте 6GSAGIN может интегрировать подходы на

основе околосетевой сети (NEN) или наземной сети (GN) для формулирования более эффективных способов передачи данных.

### 1.2.2 Наземные сети

Под мобильной транзитной связью (МВН) понимается соединение сотовых станций будущего поколения с проводными сетями через воздушные интерфейсы [13]. Цель МВН - предоставить мобильным пользователям сети услуги, ориентированные на передачу данных и контента. Поскольку потребность в пропускной способности растет с каждым днем, МВН должна соединять сети радиодоступа (RAN) для интеграции малых и макросотовых станций с остальной проводной транзитной инфраструктурой. МВН должна обладать ключевыми характеристиками для включения в эру 6G, а именно: увеличение пропускной способности для обслуживания, увеличение емкости для обслуживания быстрых пользователей, эксплуатация и администрирование для предприятий, техническое обслуживание и предоставление услуг мобильной связи, управление синхронизацией и синхронизацией сети, и обеспечение высокой доступности сети для мобильных пользователей [14]. Для достижения этой цели МВН следует сосредоточиться на интеграции виртуализации программных функций и модернизации программно-определяемых сетей в существующую инфраструктуру. Автоматизация услуг гетерогенных сетей (HetNet) является наиболее важным аспектом МВН, который направлен на объединение ряда микросот, пикосот, фемтосот и малых сот с макросотами с помощью беспроводных средств связи.

RAN реализует технологию радиодоступа в беспроводных сетях, обеспечивает связь между мобильными устройствами и опорной сетью (CN). Технология кремниевых чипов развертывается в виде RAN как в CN, так и в пользовательском оборудовании (UE) или мобильной станции (MS). Последние достижения RAN могут взаимодействовать с облачной инфраструктурой, т.е. Cloud RAN (CRAN). CRAN может облегчить соединение на больших расстояниях

дальность связи, например, 40-80 км для 3G и >20 км для 4G. Эффективная скорость передачи данных составляет около 10 Гбит/с при очень низкой задержке (10 мс). Будучи связанной с облаком, CRAN может обеспечить возможность виртуализации в реальном времени для платформ HetNet. CRAN на основе интернет-протокола (IP-CRAN) может быть интегрирована с МВН для снижения стоимости услуг мобильных пользователей в области технологии 6G в будущем.

Спектр mmWave лежит в пределах 30-300 ГГц. Частотные диапазоны mmWave полезны для обеспечения мгновенных действий, коммуникации с учетом всех особенностей и иммерсивного опыта в области 6G. mmWave позволяет использовать короткий путь передачи и снижать потери при распространении, ограничивая потери между соседними сотами [15]. Антенны очень малого размера могут быть оснащены сфокусированным лучом для усиления сигнала. Диапазоны мм-волн могут быть использованы тремя способами: 57-64 ГГц без лицензии, 28/38 ГГц, недоиспользуемые 7 ГГц, и 71/81/92 ГГц с 12,9 ГГц и 3 ГГц по лицензии. Такие диапазоны полезны для разработки бессотовой архитектуры 6G, виртуализации сети 6G, дезагрегации структуры 6G, методов сбора энергии в 6G и усовершенствованной конструкции каркаса сети. Кроме того, на основе мм-волны в 6G могут быть созданы точки доступа в помещениях, развернуты критически важные приложения и стратегии фиксированного беспроводного доступа в Интернет. Мм-волны в 6G-SAGIN могут быть использованы в восьми различных диапазонах, включая 24,25-27,5 ГГц, 31,8-33,4 ГГц, 37,40,5 ГГц, 40,5-42,5 ГГц, 45,5-50,2 ГГц, 50,4-52,6 ГГц, 66-76 ГГц и 81-86 ГГц.

ТГц связь лежит в пределах 0,3-10 ТГц. Это может быть очень полезно для раскрытия перспектив 6G-SAGIN с использованием неиспользуемой полосы пропускания. Недавнее испытание, проведенное в торговом центре

Университетом Аалто, продемонстрировано использование распространения сигнала в D-диапазоне на частоте 140 ГГц. Аналогичным образом, в рамках программы беспроводных исследований Нью-Йоркского университета было изучено использование систем связи с каналами 140 ГГц и выше для измерения и распространения сигнала на большие расстояния. При минимальных потерях на

пути сигнал успешно передается. Разработка прототипов для развертывания в реальном времени, по воздуху. Таким образом, 6G-SAGIN может воспользоваться несколькими ключевыми аспектами ТГц связи, включая гибридные ТГц-оптические каналы беспроводной связи, ТГц для высокочастотных приложений, улучшение доступности услуг центров обработки данных, возможность развертывания мобильной сети HetNet и ТГц-3D формирование сигнала. ТГц-связь может быть использована для создания тера-вай-фай, тера-ИоТ, тера-интегрированных сетей доступа и тера-космической связи, что делает ее лучшим направлением для развертывания 6G-SAGIN. Несмотря на ряд ключевых преимуществ, необходимо провести дополнительные исследования, чтобы увидеть, как массивные антенные решетки могут быть использованы в SAGIN, сформулировать дизайн двунаправленных антенн и смягчить критичные к задержкам приложения.

Далее представлена архитектура 6G-SAGIN, основанную на оптике свободного пространства (FSO) и подразумевающую гибридную оптическую беспроводную сеть (HOWF). Предполагается, что HOWF будет работать в перегруженном спектре 6G. Использование FSO в HOWF повышает безопасность экосистемы 6GSAGIN. HOWF сочетает в себе связь на основе FSO и парадигму связи в видимом свете (VLC) для создания сценариев обслуживания, чувствительных к радиочастотам. Для создания приложений 6G-SAGIN потребуются большие расстояния и высокая пропускная способность [16]. Таким образом, HOWF получит аналогичные преимущества благодаря включению схемы отслеживания наведения (APT). Поскольку кабины на базе VLC находятся в динамическом состоянии, они должны быть интегрированы с модулями APT для распространения бесперебойной связи с вычислительными объектами в кабине. Каждая из кабин на базе VLC оснащена высокоскоростными протоколами беспроводной связи, которые лучше всего работают на коротких расстояниях и в условиях прямой видимости. Использование световых светодиодов (LED) в кабинах VLC позволяет различным гетерогенным сетевым терминалам, таким как смартфоны, роботы, транспортные средства и компьютеры, получать доступ к 6G-

SAGIN в полностью двунаправленно. Встраиваемые контроллеры на базе VLC могут быть размещены на потолке, а терминалы - внутри кабин для связи. Все передаваемые данные транслируются внутри каждой кабины с помощью используя схемы мультиплексирования с временным разделением каналов (TDM) и множественного доступа с временным разделением каналов (TDMA). Для связи между всеми терминалами внутри кабины можно использовать каналы «устройство-устройство» на основе VLC. Второй тип соединения с большей пропускной способностью - между терминалами одной кабины VLC. В разных ячейках кабины могут использоваться два двунаправленных модуля восходящей и нисходящей связи. Контроллер гетерогенной сети (HetNet) позволит разгрузить спектр связи между терминалами VLC внутри кабины. Необходимо подумать об организации связи между терминалами нескольких кабин VLC с помощью модуля FSO-VLC HetNet. Это позволит двум двунаправленным FSL-каналам общаться с одним двунаправленным FSO-каналом. Контроллер HetNet является наиболее важным уровнем в этой структуре, который выполняет взаимодействие сигналов, взаимодействие сигналов, распределение сигналов, планирование и оптимизацию ресурсов. На рисунке 1.2 представлена архитектура HOWF.

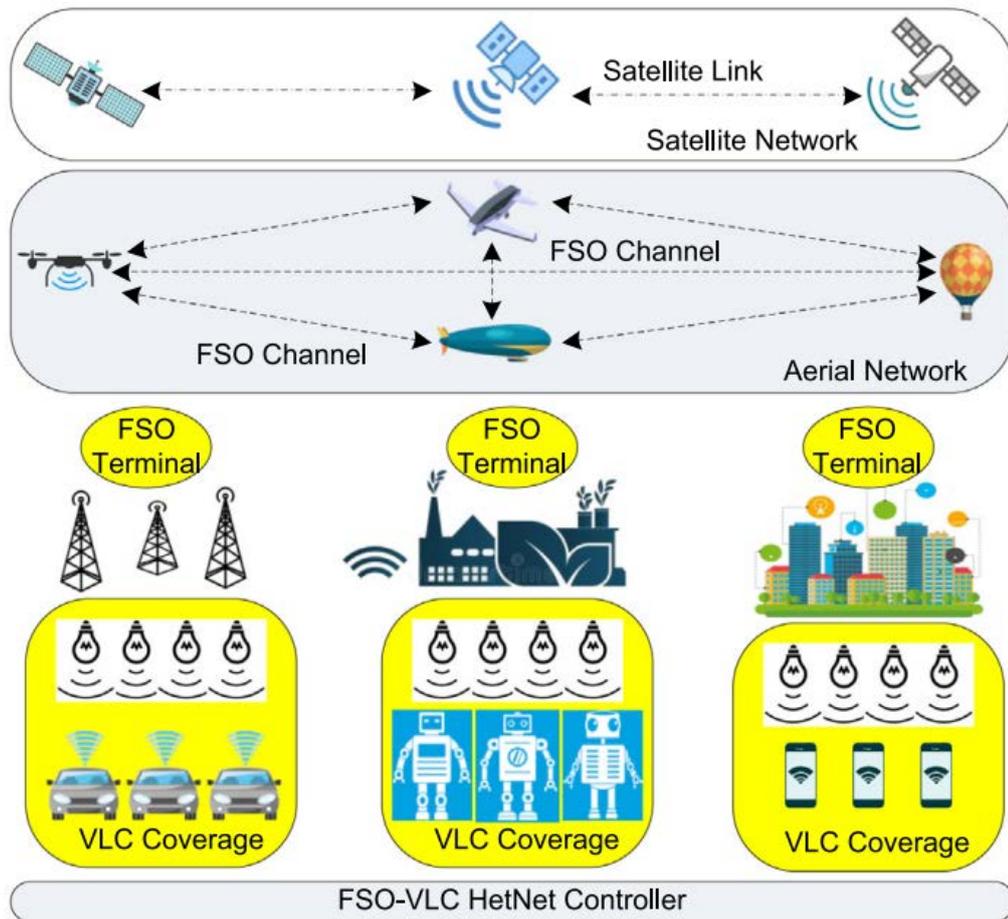


Рисунок 1.2 - HOWF: архитектура оптической сети 6G-SAGIN

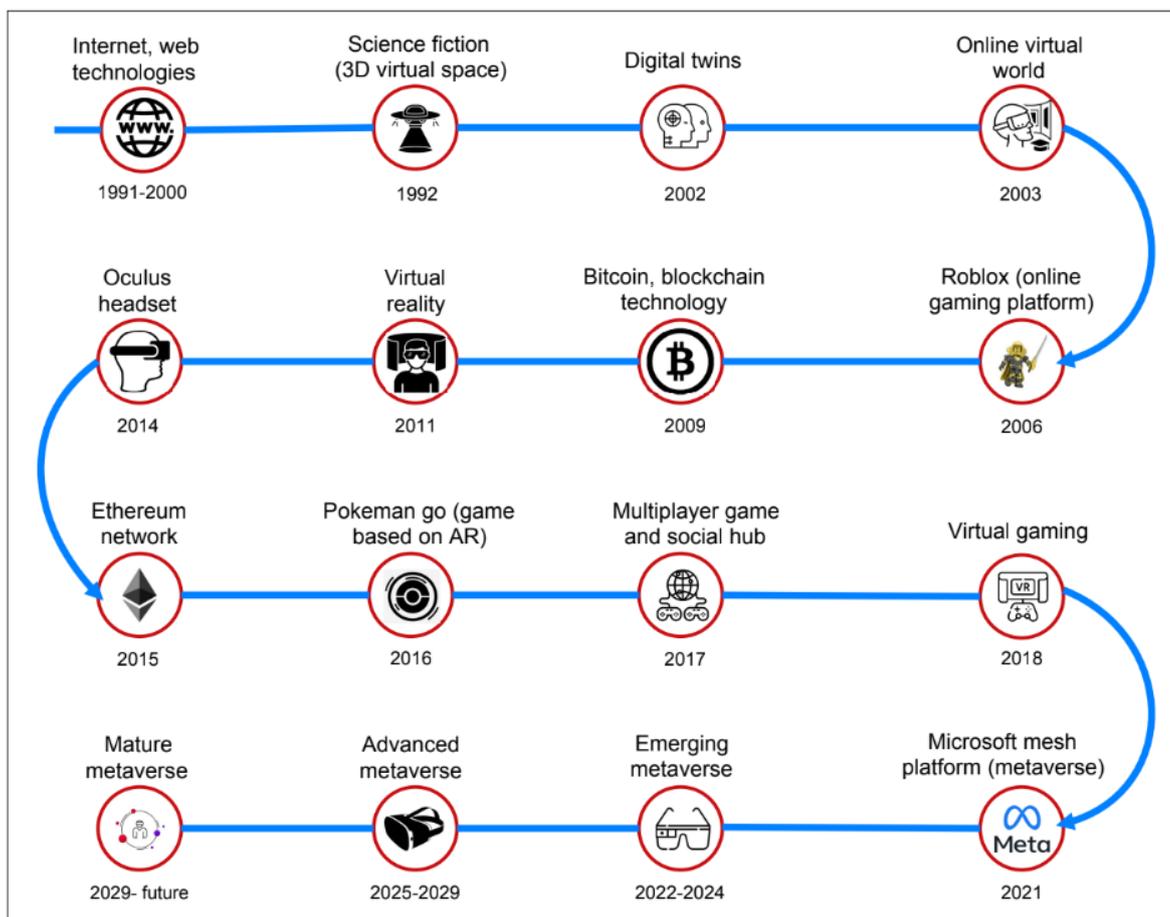


Рисунок 1.3 - Путь развития метавселенной

Концепция метавселенной существует уже несколько десятилетий, но популярность приобрела недавно. На рисунке 1.3 показана временная шкала развития метавселенной, которая включает в себя несколько значимых событий, в том числе изобретение Интернета в 1991 году, создание первого виртуального мира Second Life в 2003 году, а также недавние инициативы по созданию метавселенной от крупных технологических компаний, таких как Microsoft и Facebook. Таким образом, в ближайшие несколько десятилетий метавселенная будет продолжать развиваться и расти. Для обеспечения единообразия виртуальных миров были разработаны стандарты, обеспечивающие совместимость и взаимодействие. Наиболее распространенными стандартами являются OpenMetaverse, язык моделирования виртуальной реальности (VRML), расширяемый 3D (X3D) и совместная проектная деятельность (COLLADA) [17]. OpenMetaverse — это фреймворк с открытым исходным кодом, основанный на протоколе OpenSim,

который предоставляет набор интерфейсов прикладного программирования (API), библиотек и инструментов для создания и взаимодействия с виртуальными мирами. VRML - открытый стандарт 3D-графики для создания и управления интерактивными 3D-мирами. X3D - открытый, основанный на XML стандарт 3D-графики для создания интерактивных 3D-миров [18]. COLLADA - открытый, основанный на XML формат 3D файлов, который используется для обмена 3D активами между различными приложениями, а также для создания 3D моделей и анимации. Так, протокол OpenSim используется для передачи данных между различными виртуальными мирами, а VRML и X3D - для создания 3D-сцен и объектов [19]. COLLADA используется для обмена 3D-активами, а OpenMetaverse - для создания целых виртуальных миров. В результате, понимая различные роли каждого стандарта, можно разработать более захватывающий и интерактивный опыт метавселенной [20].

Однако для того, чтобы сценарии использования metaverse в полной мере реализовали возможности огромной виртуальной вселенной, они должны отвечать определенным требованиям, включая относительно низкую задержку, высокую потребность в ресурсах, совместимость приложений и безопасность. Для удовлетворения потребностей в более высоких скоростях передачи данных, сверхнадежной связи с низкой задержкой (URLLC) и услугах Интернета всего (IoE) уже развернуты сети 5G, которые будут способствовать появлению множества услуг. К ним относятся сверхнадежная связь с низкой задержкой, расширенная мобильная широкополосная связь (eMBB) и массовая машинная связь (mMTC) с пиковой скоростью передачи данных 20 Гбит/с. Несмотря на то что в 5G задействованы частотные диапазоны mmWave, они не могут поддерживать скорости передачи данных на уровне Тбит/с и задержки на уровне микросекунд, необходимые для следующего поколения виртуальной дополненной реальности (VAR), сервисов IoE, интерфейсов «мозг-компьютер» и подключенных автономных систем.

Для преодоления этих проблем 6G рассматривается как новая революционная технология беспроводной связи и передовая сетевая архитектура для тактильных и

осязательных приложений, обеспечивающая высокую пропускную способность, надежность, низкую задержку, энергоэффективность и интеллектуальные услуги. Системы 6G позволят повысить производительность и качество обслуживания пользователей (QoS) в большей степени, чем 5G. Также ожидается, что это будет глобальное средство связи с пользовательской скоростью передачи данных около 1 Тбит/с и задержкой менее 1 мкс. В этом контексте мобильные граничные вычисления (MEC) также рассматриваются как фундаментальная технология для перехода к сетям 5G и 6G. Это связано с тем, что они объединяют облачные вычисления и среду интернет-услуг на границе сети, предлагая повышая эффективность сети доступа.

Благодаря тому, что MEC фокусируется на близости к пользователю, сокращается время работы сети и задержки в предоставлении услуг, повышается удобство работы пользователей. Кроме того, значительное внимание исследователей привлекли метавселенные для развивающихся беспроводных систем следующего поколения. В [3] авторы представили обширный обзор, описывающий ключевые роли и применения блокчейна в метавселенной. Выявлено, что блокчейн обладает большим потенциалом для сохранения цифровых данных и поддержания целостности в распределенной среде. В [4] авторы представили обзор решений на основе метавселенной для образовательных сценариев использования и роль IoE и XR в метавселенной для образования, обучения и развития навыков. Несмотря на прогресс в мире метавселенной, риски безопасности и конфиденциальности остаются главной проблемой для ее дальнейшего развития [5]. В связи с этим необходимо использовать ИИ и технологию блокчейн для развития возможностей самоуправления сообществ метавселенных, чтобы сообщать о нарушениях [17, 18].

В постпандемическую эпоху произошли значительные изменения в сфере здравоохранения, которые привели к более широкому использованию виртуальных систем здравоохранения и связанных с ними цифровых достижений [19]. Таким образом, включение метавселенной в виртуальную систему здравоохранения предлагает новые методы предоставления улучшенного лечения при меньших

затратах [20, 121]. В работах [22, 23] авторы обсуждают ключевые приложения и связанные с ними технологические достижения, которые позволят создать системы 6G, и предлагают комплексное перспективное видение сети 6G. Кроме того, авторы работ [24, 25] указывают на потенциал метавселенной в городском планировании для пересмотра дизайна города и предоставления услуг с целью повышения эффективности городского хозяйства. Таким образом, прогнозируется, что метавселенная повлияет на качество жизни. В работе [26] автор изучил применение технологии VR с использованием видеопотоков для управления удаленными сервисами. Автор оценил задержку прямых потоков с разным битрейтом и протоколами и обнаружил, что передача потока по протоколу обмена сообщениями в реальном времени (RTMP) на сервер дает наименьшую задержку. В работе [27] автор представляет два сценария использования, требующих услуг потокового видео в реальном времени, а именно удаленное вождение и взвод. Автор также реализовал и оценил решение для потоковой передачи видео в реальном времени, которое обеспечивает низкую сквозную задержку во время операций передачи в облачных и туманных системах. Однако в существующих исследованиях не рассматривались проблемы развертывания и вопросы интеграции 6G в приложения метавселенных [28].

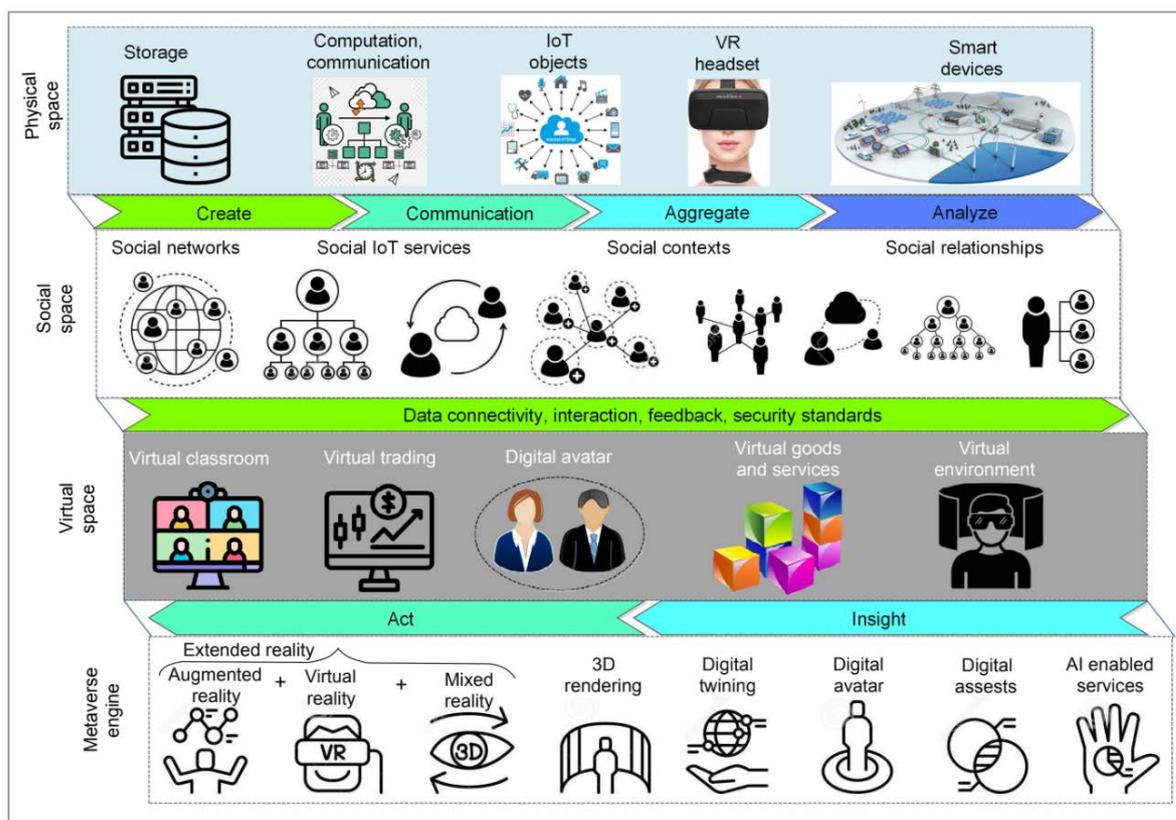


Рисунок 1.4 - Многоуровневая модель метавселенной

Метавселенная - одно из наиболее значимых и перспективных интеллектуальных приложений в развивающихся беспроводных системах шестого поколения (6G). Поэтому мы предлагаем высокоуровневую многоуровневую архитектуру метавселенной, как показано на рисунке 1.4. Эта архитектура описывает отношения между различными компонентами метавселенной, которые поддерживают взаимодействие пользователей в реальном времени между физическим и виртуальным пространством.

Физическое пространство: Физический мир имеет фундаментальные требования, такие как конечные устройства и облачные/краевые серверы для поддержки операций виртуальной среды в метавселенной. Он облегчает взаимодействие между человеческим и цифровым мирами, объединяя восприятие мультисенсорных данных, коммуникацию, обработку, хранение и физическое управление. Интеллектуальные устройства, такие как головные дисплеи VR (HMD), голографические дисплеи, умные очки и тактильные датчики, могут

обеспечить полное погружение и интерактивный опыт пользователя. В физическом пространстве IoT-устройства подключаются к метавселенным приложениям, чтобы извлекать данные из реального мира и внедрять их в виртуальный. Коммуникационная инфраструктура обеспечивает сетевое взаимодействие, состоящее из различных гетерогенных проводных или беспроводных сетей. Кроме того, вычислительные мощности и мощности хранения данных распределяются по вычислительной инфраструктуре и инфраструктуре хранения данных с помощью облачных/краевых вычислений.

Социальное пространство: Метавселенная представляет собой новую эру в развитии человеческих социальных систем, где люди могут удовлетворять потребности более высокого уровня, чем в реальном мире, такие как социальное взаимодействие, виртуальная рабочая среда и развлечения. В дополнение к традиционным социальным сетям она создает социальную сеть и интегрируется с социальным контекстом пользователя, предоставляемым социальными IoT-сервисами, где метавселенная может предложить опыт «третьего места» для погружения в социальную жизнь для социальных взаимодействий, игр, покупок, фильмов, киберспорта и развлечений. В результате метавселенная изменит способ общения с социальными сетями, устранив единственную точку доступа и используя технологию VR для создания 3-х мерной реальности.

Виртуальное пространство: Физические и социальные объекты в метавселенной отображаются в виртуальной среде с помощью API, которые позволяют пользователям взаимодействовать между физическим и виртуальным мирами. Физический уровень обеспечивает сетевую инфраструктуру и вычислительные ресурсы, а социальный уровень - необходимые связи между пользователями в метавселенной. Виртуальное пространство метавселенной включает в себя виртуальную среду для образования, торговли, товаров и услуг, а также цифровые представления пользователей в виде аватаров. Эта среда создается и поддерживается с помощью технологий рендеринга и симуляции высокой четкости, что обеспечивает пользователям максимальное погружение. Подобно походу за покупками в реальной среде, здесь пользователи также могут

приобретать виртуальные товары и одежду у поставщиков для получения интерактивного опыта.

Ядро метавселенной: ядро метавселенной получает данные от заинтересованных сторон, которые изначально создаются и поддерживаются субъектами в реальном и виртуальном мирах, для выполнения вычислений 3D-модели и других аспектов виртуальной вселенной. Он также облегчает взаимодействие между цифровыми двойниками, аватарами и технологиями интерактивного опыта, которые реализуются либо на границе, либо в облаке.

Расширенная реальность (XR): Расширенная реальность — это зонтичный термин, объединяющий виртуальную, смешанную и дополненную реальность для получения революционных впечатлений от мультисенсорного крупномасштабного 3D-рендеринга с помощью устройств XR. VR обеспечивает визуальный опыт в действительно погруженном цифровом мире, а AR предлагает графический, аудиовизуальный и голографический опыт, наложенный на реальный мир, MR обеспечивает переход между ними [18]. Таким образом, AR и VR являются необходимыми компонентами для достижения цели 3D-иммерсивного метавселенной как в физическом, так и в цифровом мире.

Цифровой двойник: близкая к реальному времени копия физических объектов в цифровых или виртуальных моделях посредством моделирования и слияния данных. Цифровые двойники могут использовать ИИ и аналитику данных для внедрения интеллекта в метавселенную для улучшения пользовательского опыта, например, когнитивные аватары и 3D-рендеринг объектов. Например, NVIDIA выпустила GANverse3D, фреймворк на базе ИИ, который может захватывать изображения реальных объектов и мгновенно создавать их цифровые копии [18]. Слои в многоуровневой архитектуре metaverse взаимодействуют друг с другом через коммуникационную инфраструктуру, обеспечивая взаимодействие пользователей в реальном времени между физическим и виртуальным пространством. Они связаны между собой через уровень обмена данными, который облегчает обмен данными между физическим и виртуальным мирами. Протоколы безопасности и конфиденциальности обеспечивают безопасный обмен данными и

их целостность. Применяя аналитику к данным, хранящимся в облаке/краевых серверах, мы получаем дополнительные сведения о том, как эффективно персонализировать услуги для пользователей. Таким образом, технология «цифрового двойника» позволит оптимизировать работу бизнеса, используя данные IoT в режиме реального времени, и будет способствовать появлению новых услуг и социальных взаимодействий.

Метавселенные генерируют огромные объемы данных, когда миллионы пользователей общаются друг с другом, а для передачи 3D-анимации высокого разрешения также требуется значительная пропускная способность сети. Однако в метавселенных вышеупомянутые технологии чувствительны к задержкам и требуют крайне низкого времени ожидания. Поэтому сети 6G и последующие могут предоставить возможности для повсеместного использования интеллектуальных сервисов для крупномасштабных метавселенных устройств.

Кроме того, для обеспечения безопасности связи могут использоваться такие протоколы аутентификации, как Kerberos, безопасность транспортного уровня (TLS) и инфраструктура открытых ключей (PKI). Кроме того, внедрение технологии blockchain обеспечит защиту данных, повысит интеллектуальность сети при обработке больших данных и обеспечит безопасность сетей IoT и XR.

Терагерцовый (THz) частотный диапазон: Частотный диапазон ТГц-коммуникаций будет поддерживать терабитную скорость передачи данных, широкие возможности подключения, безопасную передачу данных, сверхнизкую задержку и высокую спектральную эффективность для удовлетворения требований приложений 6G. Однако ТГц-связь является жизненно важной технологией для сетей 6G, несмотря на некоторые ограничения, включая значительные потери при распространении, высокие отражения, быстрые колебания канала и ограниченную дальность связи.

Решение: Усилители мощности и УКВ-трансиверы должны быть разработаны с учетом поддержки много-гигагерцового диапазона частот. Междиапазонные помехи должны быть правильно смоделированы, исследованы и подавлены в многодиапазонной системе, чтобы полностью использовать полосу

пропускания и удовлетворить требования QoS. Метод многоцелевой ретрансляции может расширить диапазон передачи ТГц-связи и компенсировать высокие потери на отражение.

Формирование сигнала: Формирование сигнала в терагерцовом диапазоне поддерживает высокоскоростную передачу данных с помощью технологии Massive MIMO. Однако из-за потерь при распространении и проникновении в ТГц-диапазоне формирование сигнала для субмиллиметровых волн является сложной задачей. Кроме того, для достижения высокой эффективности использования спектра и энергоэффективности разрабатываются гибридные архитектуры формирования сигнала (НВА). Однако развертывание таких НВА в системах THzCom затруднено из-за низкого коэффициента усиления мультиплексирования, низкой эффективности и эффекта кривого канала.

Решение: Потенциальным решением этой проблемы является развертывание распределенных гибридных архитектур формирования сигнала (ДНВА) для систем связи в диапазоне ТГц. ДНВА могут улучшить усиление мультиплексирования, повысить эффективность и устранить эффект смещение луча. Кроме того, для повышения точности формирования луча в ТГц можно использовать методы оценки и отслеживания канала на основе глубокого обучения.

Высокое энергопотребление: Эра 6G станет свидетелем широкого использования маломощных устройств, датчиков и смартфонов без батарей для реализации IoT. Значительное количество энергии также будет расходоваться на питание сверхбольших антенн для широкополосной обработки сигналов, отслеживания мобильных устройств и обеспечения адекватного покрытия для устройств со сверхузкими лучами. Таким образом, энергопотребление будет существенно препятствовать успеху 6G, что потребует внедрения оптимальных энергоэффективных технологий.

Решение: Метавселенная требует значительного количества энергии из-за непрерывного зондирования и передачи данных в сетях 6G. Использование передовых энергоэффективных технологий, таких как усовершенствованные антенные технологии, динамическая частота, управление питанием и маломощные

приемопередатчики, сыграет ключевую роль в снижении энергопотребления сетей 6G. Использование интеллектуальных технологий сбора энергии и беспроводной передачи энергии также будет способствовать дальнейшему снижению общего энергопотребления. Кроме того, использование пограничных и облачных вычислений для снятия нагрузки с базовой станции также будет способствовать снижению энергопотребления.

Совместимость: ожидается, что сеть 6G будет предоставлять широкий спектр дополнительных функций, которые должны поддерживаться другими устройствами. Очень сложно обеспечить скорость передачи данных в 1 Тбит/с, искусственный интеллект, XR, интегрированное зондирование и функции связи с помощью отдельных устройств. Поэтому возникает важнейшая проблема совместимости устройств при переходе от одной технологии к другой.

Решение: В эпоху 6G должны быть решены проблемы с возросшей сложностью метавселенной, которая будет включать в себя большое количество разнородных устройств и сетей. Чтобы решить эту проблему, необходимо разработать решения для междоменной совместимости, которые позволят устройствам и сетям взаимодействовать друг с другом независимо от их базовых платформ и протоколов. Внедрение 6G должно быть поддержано разработкой стандартизированной платформы, такой как Metaverse Ecosystem, которая позволит различным типам устройств взаимодействовать друг с другом.

Неэффективное использование ресурсов и задержка: Иммерсивный опыт, обеспечиваемый метавселенной с помощью AR/VR, предъявляет значительные требования к инфраструктуре мобильной граничной сети с точки зрения скорости передачи данных, надежности и задержки. Для сети 6G сверхнизкая задержка является ключевым требованием для различных приложений (например, CRAS, XR, VCI). Соблюдение компромисса между скоростью, надежностью и задержкой является одной из проблем при разработке архитектур поддержки сверхмалой задержки для обеспечения спектральной и энергетической эффективности.

Решение: Решения на основе машинного обучения и глубокого обучения могут помочь в прогнозировании запросов пользователей и изменений в состоянии

канала для снижения задержки передачи данных. Для обеспечения связи с низкой задержкой, мобильности и управления ресурсами для улучшения сквозной производительности можно использовать нарезку сети.

Межканальные помехи: между спутниками и наземными сетями связи, вызванные совместным использованием спектра, являются одной из основных проблем для сверхплотных сетей 6G. Например, ограничения по дальности связи и вычислительной мощности в спутниковой связи могут вызвать проблемы с обратной связью по информации о состоянии канала (CSI). В результате для сверхплотных сетей необходимо разработать соответствующий подход к управлению помехами.

Решение: Функция плотности вероятности или кумулятивный расчет могут быть использованы в будущих сверхплотных беспроводных сетях 6G для предотвращения межканальных помех.

### **1.3 Правовые аспекты применения метавселенных**

Метавселенная привнесет новые аспекты в сферу защиты данных и конфиденциальности. Если до сих пор правила защиты данных касались физических данных о пользователях/людях и их перемещения между странами, то в метавселенной помимо первоначальных пользователей появятся совершенно новые участники (аватары) с большим количеством данных, полученных из новых источников, таких как данные, собранные из выражения лица и глаз, перемещающиеся между различными метавселенными. Эта общая идея несет в себе множество сложностей, проблем и политических вопросов, которые необходимо рассмотреть с точки зрения конфиденциальности и защиты данных.

Метавселенная будет связывать человека с его «аватаром» или другими цифровыми представлениями. Поэтому страны, скорее всего, сочтут собранную

информацию о действиях пользователя metaverse персональными данными, на которые распространяются существующие законы о защите частной жизни и данных. В связи с этим возникают сложные вопросы такие как юрисдикционная ответственность, а также переносимость и совместимость. Метавселенная создает проблемы совместимости и перемещения пользователей внутри и между различными метавселенными вместе с их данными и активами. Это создает дублирование с перемещением данных в реальном мире. Определение юрисдикции в метавселенной становится очень сложной задачей в результате добавления нового важного игрока «аватара». Будет ли юрисдикция применяться в зависимости от местонахождения пользователя реального мира или аватара? Во многих странах действуют дополнительные договорные требования, а также некоторые требования к локализации, и пока неясно, как это будет регулироваться в метавселенной. Также неясно, как концепции «экстерриториального охвата», присутствующие в GDPR и других нормативных актах, будут применяться в метавселенной. Дублирование, возникающее при производстве данных в результате присутствия нового субъекта, должно регулироваться.

Пользователи, скорее всего, будут предоставлять больше информации о себе, чем сегодня в результате разнообразия источников сбора данных. Вместо того чтобы иметь дело с явными источниками сбора данных, метавселенная введет новые источники сбора данных, которые, например, глазные трекеры, которые могут предоставлять данные и сведения об эмоциях посредством интерпретации выражения лица и мозговых волн. Некоторые эксперты в области права рекомендуют разработать правила использования метавселенных таким образом, чтобы ограничить сферу применения рекламы, реагирующей на эмоции. Кроме того, режимы, используемые в метавселенных, могут представлять собой высокие риски, которые могут быть использованы для нарушения неприкосновенности частной жизни. В метавселенной игроки перемещают свои аватары по локациям, а игрок наблюдает за происходящим, принимая либо перспективу от первого лица и глядя глазами своего аватара, либо перспективу от третьего лица, когда камера не прикреплена к аватару, что позволяет игроку наблюдать как за своим аватаром, так

и за окружающей его средой. В перспективе от третьего лица, которая иногда используется по умолчанию, камера может перемещаться независимо от аватара, и ее можно переносить в места, отличные от местоположения аватара. Это практически позволяет игроку использовать камеру в качестве шпионского устройства. Кроме того, камера может быть прикреплена к другому аватару без ведома этого аватара. Одним из примеров источников данных, нарушающих приватность в метавселенной, являются некоторые устройства, используемые в таких играх, как Second Life Game. В игре бесплатно предоставляются наручные часы, которые сообщают о местонахождении владельца часов, а также других аватаров, находящихся поблизости, затем эти данные передаются в базу данных за пределами виртуального пространства. Поведение друзей владельца часов, находящихся поблизости от аватара, отслеживается. Друзья не знают о работе часов. Поскольку база данных размещена на сайте за пределами виртуального пространства, она доступна для поисковых систем реального мира. Такое реально-виртуальное взаимодействие создает проблемы с конфиденциальностью и возможными нарушениями защиты данных.

С учетом вышеизложенных опасений метавселенная создает риски массового профилирования, которое может использоваться для рекламы, контроля за решениями людей в социальной, политической и экономической сферах, а также риски усиления государственной слежки за счет доступа к таким чувствительным данным, как эмоциональные реакции и биометрические данные. Ожидается, что метавселенная усугубит нынешнюю ситуацию со сбором данных о гражданах и массовым профилированием благодаря более широким источникам сбора данных, которые помогают правительствам собирать данные не только о личной информации и поведении людей, но и о внутренней реакции людей на различные увиденные или пережитые вещи и соотноситься с тем, как они реагируют. Это позволит с высокой точностью предсказывать поведение и, следовательно, создавать более четкие массовые профили и легче манипулировать людьми. Это может представлять непосредственный риск для национальной безопасности стран.

Пока неясно, нужно ли наделять аватаров правосубъектностью, чтобы они несли ответственность за свои действия. Будет ли это отдельно от правосубъектности оригинальных пользователей или они связаны? Создает ли это риски для защиты данных и идентификации пользователей? Поскольку специальных законов, регулирующих аватары, не существует, контент, который пользователи раскрывают с помощью аватаров, может нарушить защиту персональных данных пользователей и сделать их идентифицируемыми. Кроме того, в метавселенной разрешено создавать альтернативные учетные записи, называемые алтами, что позволяет пользователям участвовать в метавселенной под разными именами. Такие алты обеспечивают некую анонимность, которая может быть использована для незаконных действий и поведения.

Гарантировать права на интеллектуальную собственность в метавселенной, где контент распространяется и тиражируется с помощью Web 3.0 и платформ на основе блокчейна, довольно сложно. В качестве технического решения были представлены NFT, однако это может вызвать вопросы, связанные с применимым законодательством и юрисдикциями. В метавселенной существует несколько видов интеллектуальной собственности: защита авторских прав на виртуальные объекты, такие как аватары и предметы, защита товарных знаков на логотипы и бренды, а также патентная защита на технологические достижения в метавселенной. В метавселенной пользователи могут создавать виртуальные репрезентации объектов реального мира, которые могут включать работы, защищенные авторским правом, или товарные знаки, которые им не принадлежат. Также спорным является вопрос о том, будут ли аватары подпадать под защиту авторских прав в случае, если человек создаст аватар реально существующего человека.

Метавселенная создает большие риски в отношении цифрового суверенитета стран: как страны будут осуществлять суверенитет над своей землей и гражданами во всех смыслах; будут ли аватары гражданами; и как будет осуществляться суверенитет над землей в метавселенной. Кроме того, многие правительства ввели цифровые банки в метавселенной; как они будут осуществлять суверенитет в реальном мире и как они будут осуществлять суверенитет над своими экономиками

и валютами в метавселенной? Под угрозой могут оказаться и некоторые национальные ценности, например, как можно гарантировать свободу слова и защиту прав и достоинства человека.

Концепция метавселенной несет в себе множество рисков и угроз для суверенитета данных стран в целом. Осуществление экономической деятельности в метавселенной сталкивается со всеми вышеупомянутыми проблемами защиты данных, а также с другими, связанными с товарными знаками. Поэтому до фактического внедрения метавселенной страны должны предпринять активные действия, чтобы минимизировать риски, связанные с метавселенной, и в то же время получить от нее пользу. Один из способов сделать это - оценить уровень угроз, которые представляет собой метавселенная. Это может помочь странам оценить силу действующих норм защиты данных и их способность противостоять возможным угрозам, которые несет в себе *metaverse*. Это также поможет определить приоритеты различных угроз и разработать необходимые защитные механизмы перед внедрением проектов метавселенных.

## ГЛАВА 2. ИНТЕЛЛЕКТУАЛЬНОЕ ЯДРО ИНТЕГРИРОВАННОЙ СЕТИ 6G НА ОСНОВЕ ТЕХНОЛОГИИ SDN ДЛЯ ПОДДЕРЖКИ ПРИЛОЖЕНИЙ ТЕЛЕПРИСУТСТВИЯ

Проблемы, возникающие при проектировании систем сотовой связи пятого и шестого поколений (5G/6G), особенно необходимость обеспечения субмиллисекундных задержек, сверхвысокой доступности и надежности, имеют подходы к их решению. Удастся улучшать качество передачи данных благодаря технологиям программно-определяемых сетей (software-defined network, SDN), виртуализации сетевых функций (network functions virtualization, NFV), мобильных граничных вычислений (mobile edge computing, MEC). Как следствие, и академическое сообщество, и бизнес уделяют большое внимание прогрессу в этих областях. В современных сотовых сетях множество сетевых функций отвечают за обработку сетевых сервисов через статические соединения. Поэтому добавление дополнительных сервисов не является простой задачей из-за стоимости, энергопотребления и проблем, связанных с интеграцией с необходимым оборудованием. Цель развития NFV и SDN – решить эти проблемы.

Технология NFV позволяет обеспечивать традиционные (фиксированные) сетевые функции с помощью программного обеспечения, выполняемого в виртуализированной среде, что позволяет более гибко управлять сетью. NFV использует технологии виртуализации на программируемом оборудовании, таком как устройства хранения данных, серверы общего назначения и коммутаторы, чтобы отделить программную реализацию сетевых функций от специального, аппаратного оборудования [29]. Сейчас разрабатываются многочисленные инициативы по созданию коммерческих решений и стандартов для NFV с открытым исходным кодом. Среди наиболее важных из них можно назвать два проекта: управление и оркестровка MANO (MANagement and Orchestration), который является инициативой Европейского института телекоммуникационных

стандартов (ETSI) с открытым исходным кодом [30] и программирование и оркестровка сервисов для виртуализированных программных сетей (SONATA). Среда разработки SONATA призвана повысить программируемость и адаптивность сотовой сети 5G. Она направлена на упрощение и организацию модульных сервисных платформ, чтобы удовлетворить потребности различных поставщиков услуг. Кроме того, система поддерживает оркестрацию сетевых функций и предлагает собственную парадигму разработки.

Программно-определяемые сети SDN – это технология, которая, физически разделяя плоскость пересылки данных и плоскость управления, предлагает гибкую и экономичную сетевую архитектуру. Часть сети, которая принимает решения о том, как обрабатывать трафик, называется плоскостью управления [31]. В ответ на команды плоскости управления информация пересылается по плоскости данных. Для передачи управляющего сигнала от плоскости управления к плоскости данных используются интерфейсные протоколы открытого стандарта, такие как OpenFlow и ForCES. Интерфейсы прикладного программирования (API) позволяют оператору сети настраивать, контролировать и администрировать сеть SDN. Программно-определяемые сети создаются в первую очередь для снижения сложности аппаратного обеспечения и расширения возможностей адаптации сети.

Последние достижения в технологиях NFV и SDN способствуют наступлению эры сетей на основе программного обеспечения. Хотя ни NFV, ни SDN не являются необходимыми для работы, эти две технологии хорошо дополняют друг друга [32]. Оба метода способны сэкономить средства при внедрении сложного компьютерного оборудования. Первые итерации SDN использовались в основном в университетских системах, центрах обработки данных, сетях колледжей и частных локальных сетях. Конечная цель NFV – заменить промежуточные устройства в сотовых сетях на программное обеспечение, работающее на недорогих готовых компьютерах, т.е. менее дорогое серверное оборудование общего назначения [33].

С помощью SDN и NFV может быть реализована так называемая нарезка сети (network slicing). Сетевой фрагмент (slice) – это набор сетевых функций и

технологий радиодоступа, которые работают вместе для выполнения определенного сценария использования сети. Другими словами, сетевой фрагмент – это специализированное звено, обеспечивающее предоставление услуги связи. Используя нарезку сети, оператор может создать множество логических сетей (каждая из которых предназначена для разных целей) на одной и той же базовой физической сети. В данной работе рассматривается разработка оптимизированного алгоритма балансировки нагрузки для мультиконтроллерных сетей SDN, которые могут быть развернуты для систем 5G/6G. Предложенная модель отвечает заявленным требованиям сетей 6G.

## 2.1 Архитектура сети

В современном мире появляется все больше технологий иммерсивного взаимодействия посредством интернет-пространства, многие уже сталкивались с концепцией тактильного интернета, которая позволяет нам «перемещаться» в пространстве. Мультивселенные в свою очередь представляют собой виртуальные миры, в которых цифровые объекты сливаются (взаимодействуют) с физическими вещами и событиями. Это образ интернет-пространства нового поколения, в котором пользователи взаимодействуют с программными приложениями и другими пользователями в качестве аватаров в трехмерном (3D) виртуальном мире, ориентированном на социальные связи.

Так как услуги телеприсутствия подразумевают взаимодействие машин и людей в режиме реального времени, крайне важно обеспечить высокую надежность обмена информацией и гарантированную доставку пакетов данных. Для этого необходимо соблюдать строгие требования к среде передачи. В рекомендации ITU-R M.2083-0 [34] основополагающими принципами проектирования сетей связи 5G были обозначены гибкость и разнообразие для обслуживания множества

различных сценариев использования. С точки зрения передачи полного спектра человеческих чувств основными являются пункты по задержке и надежности. Круговая задержка должна быть неощутимой для человека и составлять максимум 1 мс, так как приложения, которые входят в группу URLLC (Ultra-Reliable Low-Latency Communications, сверхнадежная связь с малой задержкой), очень чувствительны ко времени [34, 35]. Например, телемедицина предполагает, что любое движение врача на другом конце канала связи, обязательно должно передаваться по сети и воспроизводиться на стороне получателя практически мгновенно, что позволит проводить удаленные хирургические операции, например, в тех ситуациях, когда пациент и врач находятся в разных странах и невозможно организовать их личную встречу. Кроме того, сети связи с ультрамалыми задержками требуют надежности передачи данных выше 99,9999% [36].

Только при реализации этих требований появляется возможность обеспечить высокую надежность и точность передачи данных в режиме реального времени, что является необходимым для решения задач в области автономной техники, телемедицины, индустриальной автоматизации и т.д. Для построения сетевой инфраструктуры подходящего уровня необходимо использовать передовые технологии. Кроме того, очень важно также внедрять новые технологии аналитики и обработки данных и модели искусственного интеллекта для обработки больших объемов информации и принятия решений на основе полученных в реальном времени или уже имеющихся данных [34, 37].

Архитектура должна быть гибкой и масштабируемой и состоять из слоев, взаимодействующих друг с другом по отдельным сетевым интерфейсам [37]. Должен обеспечиваться контроль и эффективное распределение ресурсов сети. В целом, реализация услуг URLLC требует комплексного подхода к развитию сетевой инфраструктуры и применению передовых технологий.

В качестве одного из элементов достижения поставленных требований к качеству передачи данных предлагается рассмотреть и протестировать мультиконтроллерную программно-определяемую архитектуру сети (рисунок 2.1), которая позволит рационально использовать имеющиеся вычислительные ресурсы,

а также сравнить ее эффективность с традиционными архитектурами [38].



Рисунок 2.1 – Общая архитектура

Набор развернутых контроллеров SDN в плоскости управления называется  $C$  и определяется следующим образом:

$$C = \{C_1, C_2, \dots, C_N\}, N \in R, \quad (2.1)$$

где  $N$  – общее количество развернутых SDN-контроллеров. В плоскости данных развернуто  $K$  коммутаторов OpenFlow, распределенных между кластерами контроллеров. Каждый коммутатор имеет подключение к SDN-контроллеру. Для решения задачи оптимизации в статье применен алгоритм хаотической оптимизации salp swarm [39].

Набор развернутых OpenFlow-коммутаторов определяется следующим образом:

$$S = \{S_1, S_2, \dots, S_K\}. \quad (2.2)$$

Связи между коммутаторами и контроллерами SDN отображаются в матрице коммутации, где строки указывают на номер контроллера SDN, а столбцы – на номер коммутатора OpenFlow. Символ  $T$  обозначает общее количество коммутаторов OpenFlow, связанных с контроллерами SDN. Пример матрицы коммутации:

$$\begin{bmatrix} 0 & \dots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \dots & 0 \end{bmatrix}, T = \begin{bmatrix} 2 \\ \vdots \\ M \end{bmatrix}. \quad (2.3)$$

Одним из методов оценки производительности контроллера является измерение времени его отклика, на которое в первую очередь влияет задержка в очереди. Развернутые контроллеры SDN смоделированы с помощью модели очередей M/M/s, предполагающей, что каждый контроллер имеет  $s$  ядер. Передаваемые пакеты поступают на контроллер со скоростью, определяемой распределением Пуассона, формируя единичную очередь на контроллере. Среднее время отклика ( $T_i$ )  $i$ -го контроллера вычисляется по формуле Эрланга следующим образом:

$$T_i(\lambda) = \frac{O\left(s, \frac{\lambda_i}{\mu}\right)}{s\mu_i - \lambda_i} + \frac{1}{\mu},$$

$$O\left(s, \frac{\lambda}{\mu}\right) = \frac{\left(\frac{(s\rho)^0}{s!}\right)\left(\frac{1}{1-\rho}\right)}{\sum_{k=0}^{s-1} \frac{(s\rho)^k}{k!} + \left(\frac{(s\rho)^0}{s!}\right)\left(\frac{1}{1-\rho}\right)} = \frac{1}{1 + \left(\frac{1}{1-\rho}\right)\left(\frac{s!}{(s\rho)^0}\right)\sum_{k=0}^{s-1} \frac{(s\rho)^k}{k!}}, \quad (2.4)$$

$$\rho = \frac{\lambda_i}{s\mu},$$

где  $\lambda$  – скорость прихода пакетов;

$\lambda_i$  – средняя скорость поступления компьютерных задач на все коммутаторы, подключенные к  $i$ -му контроллеру;

$O(s, \lambda/\mu)$  – вероятность постановки прибывших пакетов в очередь, когда серверы системы заняты;

$\mu$  – скорость обслуживания;

$\mu_i$  – скорость обслуживания  $i$ -го граничного устройства;  $\rho$  – уровень использования серверов, который измеряет надежность системы.

Когда контроллер достигнет своей максимальной пропускной способности, переход будет продолжаться с  $(s\mu)$  и не более. Скорость прибытия контроллера  $\lambda_i$  – это совокупность средних скоростей прибытия всех подключенных коммутаторов:

$$\lambda_i = \sum_{s=1}^{k_i} \lambda_s. \quad (2.5)$$

Средняя загрузка контроллера  $L_i$  определяется как среднее количество запросов, ожидающих в очереди перед обслуживанием, и формулируется следующим образом:

$$L_i(\lambda) = s\rho + \frac{\rho}{1-\rho} O\left(s, \frac{\lambda_i}{\mu}\right). \quad (2.6)$$

## 2.2 Формулировка задачи

Задача эксперимента – динамический поиск оптимального количества SDN-контроллеров, при котором достигается минимальная задержка связи между распределенными OpenFlow-коммутаторами и SDN-контроллерами и минимальная стоимость сети. Это оптимальное число динамически изменяется при изменении

нагрузки на сеть. Кроме того, задача направлена на определение оптимальных связей между коммутаторами и контроллерами SDN таким образом, чтобы достичь целей минимальной задержки и стоимости. Также задача направлена на получение оптимального распределения коммутаторов относительно SDN-контроллеров, которое минимизирует нагрузку на каждый контроллер.

Задача состоит в том, чтобы распределить выключенные (т.е. находящиеся в спящем режиме) контроллеры или удалить текущие работающие контроллеры в соответствии с динамическим изменением трафика сети. Задача оптимизации сформулирована таким образом, чтобы найти оптимальное число контроллеров SDN и оптимальное число этих контроллеров на кластер, при котором достигается минимальная задержка, стоимость, максимальное использование и эффективность балансировки нагрузки. Задача оптимизации представляет собой функцию минимизации, которая направлена на уменьшение общего количества SDN-контроллеров в сети ( $N_T$ ), общего числа SDN-контроллеров на кластер ( $N_C$ ), общей стоимости SDN-контроллеров ( $C$ ) и средней задержки от каждого SDN-контроллера до подключенных OpenFlow-коммутаторов ( $D$ ). Целевая функция выглядит при этом следующим образом:

$$\text{Min } f(N_T, N_C, C, D), \quad (2.7)$$

Ограничения:

$$T_j \leq T_{\text{thr}}, \quad \forall C_i \in C,$$

$$U_{\text{lb}} \leq U_i \leq U_{\text{ub}}, \quad \forall C_i \in C,$$

$$U_{C-\text{lb}} \leq U_C \leq U_{C-\text{ub}},$$

где  $T_i$  –  $i$ -ое время отклика контроллера;

$T_{\text{thr}}$  – пороговое время отклика;

$C_i$  –  $i$ -ый контроллер;

$U_{\text{lb}}$  – нижний предел утилизации SDN контроллера;

$U_{\text{ub}}$  – верхний предел утилизации SDN контроллера;

$U_{C-lb}$  – нижний предел утилизации контроллеров кластера SDN;

$U_{C-ub}$  – верхний предел утилизации контроллеров кластера SDN;

$U_C$  – средняя загрузка кластера контроллеров SDN.

Эта задача представляет собой схему многоцелевой оптимизации с множеством ограничений, которую может решить соответствующий метаэвристический алгоритм. На поставленную задачу накладываются три ограничения, первое из которых заключается в том, что среднее время отклика ( $T_i$ ) контроллера, принадлежащего кластеру  $j$ , должно быть меньше, чем заранее определенное время отклика  $T_{thr}$ . Пороговое время отклика  $T_{thr}$  задается таким образом, чтобы обеспечить требуемое качество обслуживания (QoS).

Второе ограничение вводится для поддержания уровня использования контроллера SDN. Индекс использования ( $U_i$ ) должен быть не ниже минимального значения использования контроллера SDN  $U_{lb}$ , ниже которого использование контроллера становится экономически неэффективным, и не выше максимального значения использования контроллера SDN  $U_{ub}$ , при превышении которого контроллер может быть перегружен. Оператор сети заранее определяет верхний и нижний пределы использования для удовлетворения требований QoS. Этот показатель утилизации отображает использование вычислительных ресурсов, включая мощность, хранение и обработку.

Третье ограничение учитывает поддержание общего индекса использования кластера  $j$  между максимальным и минимальным пределами  $U_{C-ub}$  и  $U_{C-lb}$ . Это ограничение вводится для балансировки нагрузки между кластерами и предотвращения сбоев в работе сетевых разделов.

Рассмотрим далее функцию полезности, которая сравнивает различные решения и указывает на оптимальное решение путем отображения переменных или событий в вещественные числа:

$$U : V \rightarrow R. \quad (2.8)$$

Время утилизации – это первая полезность, введенная для отображения временного отклика контроллеров SDN. Функция потерь хорошо подходит для такой функции использования. Таким образом, можно ввести любую общую модель функции потерь. Рассматривается квадратичная функция симметрии для моделирования функции использования времени, поскольку она поддается математическому анализу. Временная полезность  $i$ -го контроллера, принадлежащего  $j$ -му кластеру, равна  $U_{T-C_i}^j$  и определяется уравнением:

$$U_{T-C_i}^j = \begin{cases} \alpha + \delta(T_{thr} - T_i^j(\lambda))^2, & T_i^j(\lambda) \leq T_{thr} \\ 0, & T_i^j(\lambda) > T_{thr} \end{cases} \quad \forall i \in \gamma \wedge j \in \beta. \quad (2.9)$$

Функция использования времени  $j$ -го кластера –  $U_T^j$ . Она может быть рассчитана как нормализованное среднее значение использования времени каждого члена кластера:

$$U_T^j = \sum_{N_c} \frac{U_{T-C_i}^j}{N_c} = \left( \sum_{N_c} \begin{cases} \alpha + \delta(T_{thr} - T_i^j(\lambda))^2, & T_i^j(\lambda) \leq T_{thr} \\ 0, & T_i^j(\lambda) > T_{thr} \end{cases} \right) / N_c, \quad (2.10)$$

где  $\alpha$  и  $\delta$  – коэффициенты полезного использования времени. Коэффициент  $\alpha$  вводится для того, чтобы избежать уменьшения коэффициента использования времени. Он может быть установлен на наименьшее требуемое значение, при котором коэффициент использования времени контроллера должен быть равен  $U_{T-thr}$ . Таким образом,  $\alpha$  и  $\delta$  могут быть представлены следующим образом:

$$\alpha = U_{T-thr} \quad \forall U_T \in [0,1]; \quad (2.11)$$

$$\delta = \frac{(1 - U_{T-thr})}{T_{thr}^2}, \quad \forall U_T \in [0,1]. \quad (2.12)$$

Сложность задачи оптимизации снижена за счет того, что рассматривается только оптимизация числа развернутых контроллеров, а оптимизация числа контроллеров внутри кластеров исключена. Это может привести к дисбалансу нагрузки между кластерами, но снижает вычислительную сложность на 57%, что экономит ресурсы и уменьшает время динамики, необходимое для нахождения оптимального решения. Это хорошо подходит для голографического типа связи, однако учет баланса нагрузки между кластерами может потребоваться для плотного развертывания, которое редко используется в голографическом типе связи в настоящее время.

### 2.3 Результаты моделирования

Проведена оптимизация числа контроллеров и их распределения в мультиконтроллерных сетях SDN на основании методики, базирующейся на параметрической оптимизации. Этот подход учитывает динамические изменения сетевого трафика и позволяет достичь следующих целей:

- минимизация задержек;
- снижение энергопотребления;
- уменьшение затрат на развертывание;
- повышение надежности сети;
- снижение вычислительной сложности.

На рисунке 2.2 показано среднее число контроллеров для традиционного метода и метода, предлагаемого в данной статье. Здесь рассматривается десять значений каждого параметра, а каждое значение представляет собой вариант моделирования. Рассматриваемые параметры моделирования представлены в таблице.

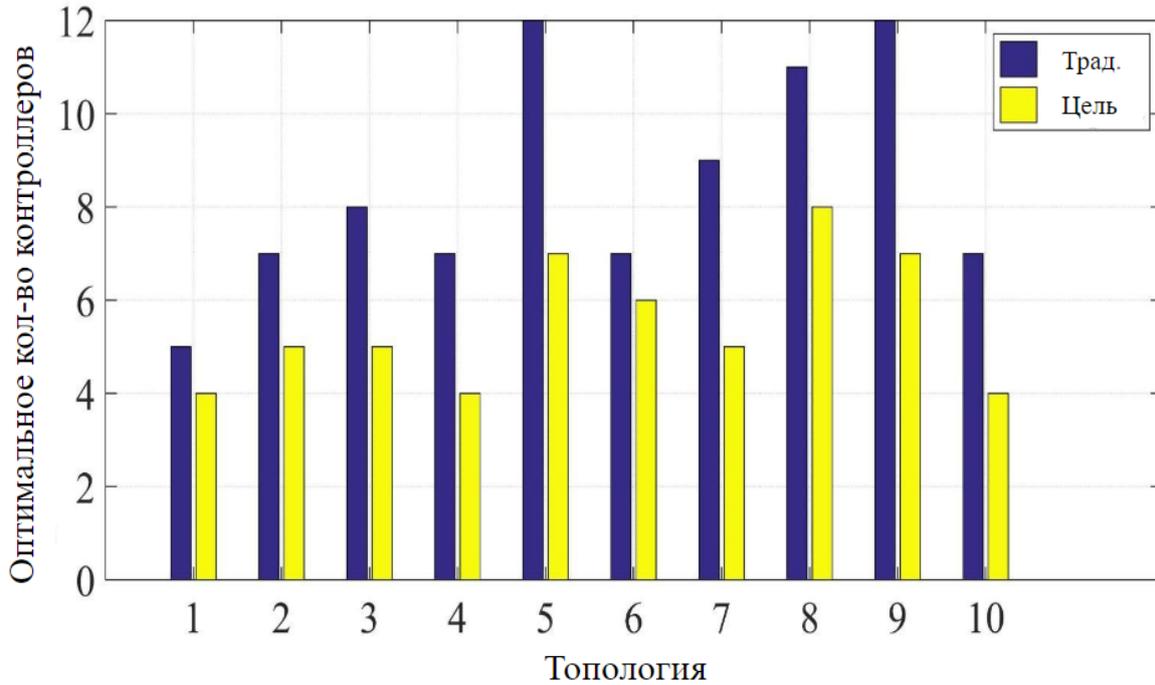


Рисунок 2.2 - Графическое представление оптимального числа контроллеров

Таблица 2.1 - Топологии сети, рассмотренные для моделирования

Номер топологии	Наименование	Количество контроллеров
1	IBM	18
2	Oxford	20
3	SCCN	23
4	AGIS	25
5	Viatel	83
6	GEANT	27
7	TATA	169
8	DFN	47
9	RNP	28
10	Carnet	43

Из рисунка 2.2 видно, что:

- в традиционном методе заранее определено фиксированное число

контроллеров независимо от нагрузки;

- предложенный метод снижает среднее число контроллеров в сети на 46% за счет динамического управления, что значительно уменьшает эксплуатационные затраты.

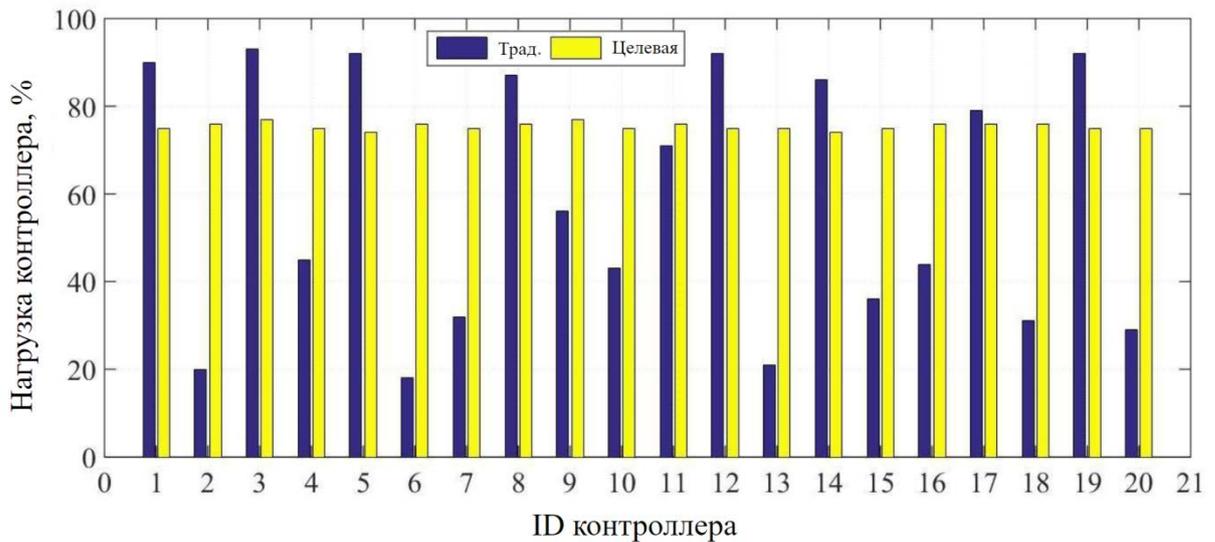


Рисунок 2.3 - Загрузка контроллеров

Из рисунка 2.3 видно, что:

- традиционный метод может демонстрировать эффект простаивания контроллеров из-за фиксированного распределения ресурсов;
- для предложенного метода загрузка контроллеров в среднем выше, что объясняется оптимизацией построения сети и избавлением от эффекта простаивания. Контроллеры работают более эффективно, обрабатывая большее количество запросов, что улучшает общую производительность сети;
- предложенный метод позволяет минимизировать эффект простаивания контроллеров из-за неэффективного распределения ресурсов, а также равномерно распределить нагрузку между контроллерами, при этом средняя нагрузка на контроллер возрастает на 53%, но не превышает оптимального значения загрузки вычислительных ресурсов в 80%.

## 2.4 Вывод по главе 2

Разработан метод повышения эффективности использования сетевых ресурсов интегрированной сети 6G, учитывающий особенности трафика при реализации услуг телеприсутствия. Основные результаты работы:

1. Предложенный метод оптимизации числа контроллеров в мультиконтроллерных сетях SDN демонстрирует значительные преимущества по сравнению с традиционным методом. Он снижает среднее число контроллеров на 46% за счет динамического управления, что уменьшает энергопотребление и затраты на развертывание, а также повышает надежность сети.

2. Предложенный метод на 53% повышает эффективность использования контроллеров, за счет динамического распределения нагрузки между ними в мультиконтроллерных сетях SDN, тем самым значительно сокращая расходы на развертывание сети.

### **ГЛАВА 3. МЕТОДЫ РАЗМЕЩЕНИЯ МАРШРУТИЗАТОРОВ СЕТИ НА БЕСПИЛОТНЫХ ЛЕТАТЕЛЬНЫХ АППАРАТАХ (БПЛА) И ОБЕСПЕЧЕНИЯ МАРШРУТИЗАЦИИ С ПОМОЩЬЮ РОЯ БПЛА**

Развитие сетей связи направлено на повышение основных показателей качества, таких как достижимая скорость передачи данных, задержка доставки, емкость сети [40-44]. Сети 5G и последующих поколений ориентированы на предоставления услуг, чувствительных к задержкам и скорости передачи данных. Рост проникновения услуг виртуальной и дополненной реальности, а в последствии и услуг телеприсутствия [45] приводит к необходимости обеспечения высокого качества обслуживания трафика в сети.

В частности, возникает необходимость передачи трафика голографических изображений [46], а также трафика, производимого костюмами телеприсутствия, обеспечивающими интерфейс взаимодействия с метавселенной [47]. Метавселенная направлена на создание виртуального мира для всех видов деятельности, включая образование, торговлю и игры, и считается следующим поколением Интернета. При поддержке AR/VR-приложений онлайн-пользователям предоставляются услуги, схожие с очными. Для поддержки приложений метавселенной синхронизация данных и широкое покрытие беспроводной сети - две практические проблемы, которые необходимо решить, поскольку в сервисах телеприсутствия обычно задействованы носимые беспроводные устройства. Ввиду повышения частотного диапазона, используемого в сетях 5G и выделяемого для 6G, стремительно возрастает затухание сигнала. Таким образом, развертывание базовых станций в пригородах с низкой плотностью населения становится экономически неэффективно [48]. Беспилотные летательные аппараты являются более дешевым решением для организации покрытия сети для синхронизации данных метавселенных в пригородной зоне.

Применение костюмов телеприсутствия или подобных им устройств, обеспечивающих получение информации о динамике движения людей или иных объектов, существенно отличается от получения информации с помощью изображений и в значительной степени дополняет «картину» состояния и изменения окружающего нас мира. Трафик, производимый упомянутыми услугами, должен обслуживаться с тем уровнем качества, который требуется для их реализации. Одним из методов решения данной задачи является применение выгрузки трафика или выделения дополнительных ресурсов для его обслуживания.

Повышение основных показателей качества достигается внедрением новых технических решений и новых моделей и методов организации сетей связи. Одним из «узких мест» для решения задачи повышения показателей качества являются сети доступа. В сетях подвижной связи они ограничены из-за ограничения используемого радиочастотного ресурса. Наиболее эффективным методом является повышение достижимой скорости передачи данных. В процессе эволюции сетей подвижной связи можно наблюдать процесс повышения полосы частот, занимаемых каналом связи и повышения рабочих частот. Это естественный процесс, который происходит из-за необходимости повышения пропускной способности.

Согласно известному выражению К. Шеннона [49], достижимая скорость передачи  $b = w \log_2(1 + SNR)$  бит/с, где  $w$  – полоса пропускания канала,  $SNR$  – отношение сигнал/шум. Несмотря на некоторое уменьшение  $SNR$  с увеличением полосы пропускания, это является наиболее «простым» методом увеличения скорости. Слово «простым» взято в кавычки потому, что увеличение полосы пропускания связано с повышением рабочей частоты и и необходимостью решения многих задач, связанных с этим.

Однако, каждое последующее поколение сетей связи, предполагает повышение полосы частот и увеличение рабочих частот. Так в сетях 6G уже предполагается использование субмиллиметрового диапазона и рабочих частот вплоть до 1 ТГц.

Особенности распространения сигналов в этом диапазоне волн таковы, что эффективно они могут быть использованы только на относительно малых расстояниях между приемником и передатчиком и в зоне прямой видимости между ними. Свойства сигналов в этом диапазоне волн приближаются к свойствам распространения света.

В настоящее время являются открытыми задачи создания электронных приборов, позволяющих реализовать устройства для генерации, модуляции, приема и обработки таких сигналов. Также открыты вопросы создания антенных устройств в этом диапазоне частот. Однако, исследования [50,51] позволяют ожидать, что такие элементы будут созданы и они позволят использовать особенности данного диапазона для повышения эффективности сети связи.

Однако, задачи, связанные с особенностями распространения сигналов в субмиллиметровом диапазоне волн требуют решения на уровне организации сети доступа. В частности, «затенение» окружающими объектами, в том числе человеческим телом может приводить к полной потере связи. Это требует использование большего числа базовых станций (антенн) и соответствующего управления ими.

Одним из подходов к решению задачи организации покрытия в сетях 5G и 6G, может быть, использование мобильных маршрутизаторов. Уже сейчас производители оборудования выпускают 5G маршрутизаторы с автономным питанием [52], которые позволяют организовать локальную сеть стандарта IEEE 802.11. Основной задачей применения такого устройства является обеспечение достаточно высокой скорости передачи на уровне подключения к сети 5G.

При работе в перспективных сетях на частотах субмиллиметрового диапазона нужно обеспечить достаточно малое расстояние и прямую видимость между маршрутизатором и антенной базовой станции. Одним из способов достижения этого является применение беспилотных летательных аппаратов (БПЛА) [53-55]. В данной работе предлагается метод применения БПЛА или

группы БПЛА для размещения маршрутизаторов и организации обслуживания пользователей, находящихся в сложных условиях для приема сигналов сети.

### 3.1 Модель и постановка задачи

В общем случае, при размещении маршрутизатора на БПЛА необходимо решить задачу выбора точки его размещения. При этом возможны варианты, когда БПЛА является неподвижным, например, привязным [56] и когда БПЛА может перемещаться как относительно пользователей, так и относительно других БПЛА. Эти варианты отличаются характером движения БПЛА, если для первого варианта структуру сети изменять не требуется из-за изменения позиций узлов сети, то во втором варианте такая потребность может иметь место. Однако, и первый и второй варианты требуют решения двух основных задач:

1. выбора позиций БПЛА относительно пользователей и базовых станций сети связи;
2. выбора логической структуры сети, т.е. маршрутов пропуска трафика в mesh сети.

Условная модель сети приведена на рисунке 3.1. На рисунке показаны одиночный БПЛА и группа (рой) БПЛА. Маршрутизатор, размещенный на одиночном БПЛА, выполняет роль точки доступа, обслуживая пользователей в своей зоне связи. Маршрутизаторы, размещенные на БПЛА роя организованы в mesh (ячеистую) сеть. Один или несколько маршрутизаторов из этой группы могут взаимодействовать с базовыми станциями сети подвижной связи. Маршрутизаторы в «рое», в зависимости от условий, могут выполнять ряд функций: связь с базовой станцией сети подвижной связи, связь с другими маршрутизаторами группы, связь с пользователями [57].

Применение группы БПЛА позволяет покрыть связью пользователей, распределенных на некоторой территории или в некотором объеме и организовать пропуск трафика между ними и сетью подвижной связи.

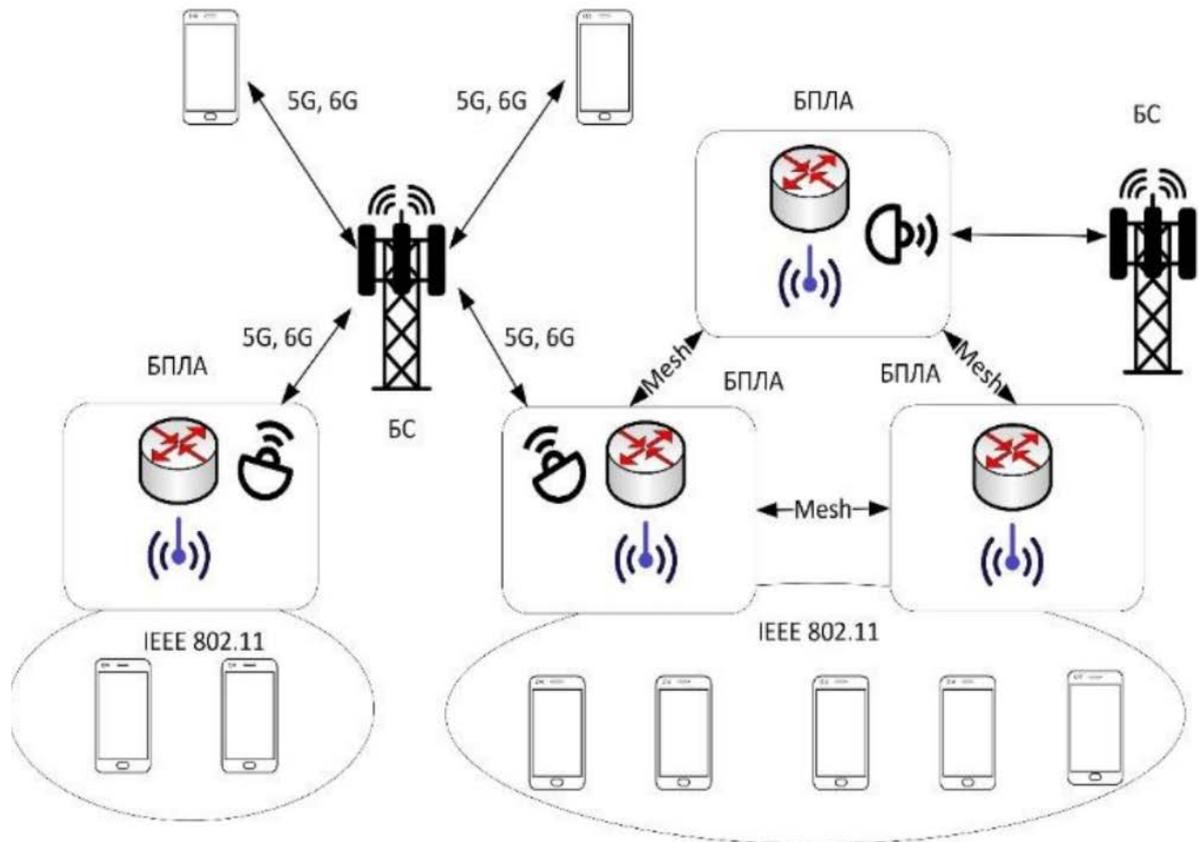


Рисунок 3.1 – Модель сети с использованием маршрутизаторов на БПЛА

### 3.2 Выбор позиции маршрутизатора

Рассмотрим случай, когда имеется группа из  $n$  маршрутизаторов. В общем случае  $n \in \mathbb{N}$  и может быть задано или ограничено  $n \leq n_{\max}$ .

Для выбора позиций размещения точек доступа чаще всего применяют различные методы кластеризации [58]. При этом объектами кластеризации являются пользователи, а за позицию размещения точек доступа принимаются центры найденных кластеров.

Задача кластеризации, как правило, решается с помощью таких алгоритмов, как  $k$ -средних, DBSCAN, FOREL [59], в работе [60] построен алгоритм с использованием фрактальных свойств сети. Каждый из этих алгоритмов имеет свои достоинства и недостатки. Например, алгоритм  $k$ -средних целесообразно применять при равномерном распределении пользователей и заранее известном числе кластеров (форма кластера при росте количества объектов стремится к форме многогранника), DBSCAN при выраженных скоплениях пользователей, причем плотность пользователей в зоне кластеров должна быть примерно одинаковой (кластер может иметь произвольную форму), FOREL позволяет найти решения как при выраженных скоплениях пользователей, так и при их равномерном распределении (форма кластера стремится к форме круга), алгоритм их работы применим при выраженных скоплениях пользователей.

В данной задаче алгоритм кластеризации должен, наряду с выделением групп пользователей (кластеров), учитывать их взаимное расположение, т.к. маршрутизаторы в центрах кластеров должны образовывать связную сеть, а сама эта сеть должна быть связана с сетью подвижной связи.

Далее предлагается алгоритм, построенный на основе алгоритма кластеризации FOREL [61]. Идея данного алгоритма предпочтительна по ряду причин. Во-первых, в данном методе задается размер кластера  $R$ , который фактически является максимально возможным размером выделяемого кластера. В задаче построения сети это позволяет определить размер кластера через достижимую скорость передачи данных  $b$  на участке пользователь-точка доступа. Если определена зависимость  $b = f(R)$ , то

$$R = \arg \{b_{\min} = f(R)\} \quad (3.1)$$

где  $b_{\min}$  – минимально допустимая скорость передачи данных, задаваемая в качестве параметра при построении сети.

Функциональная зависимость  $f(R)$  – специфична для применяемого стандарта или группы стандартов и может быть определена на их основе, например [62].

Для большей общности эту функцию можно аппроксимировать с помощью известного выражения К. Шеннона, с учетом корректирующего коэффициента  $\alpha$

$$b(R) = \alpha w \log_2(1 + \gamma(R)) \quad (3.2)$$

где  $\alpha$  – корректирующий коэффициент, учитывающий отличие используемой технологии от аналитической модели (теоретически достижимой скорости),

$\gamma$  – отношение сигнал/шум,

$w$  – полоса пропускания (Гц).

В общем случае  $0 < \alpha$ . В случае, когда не применяются методы пространственно-временного уплотнения (MIMO)  $0 < \alpha < 1$ . При применении таких методов коэффициент может превышать единицу.

Отношение сигнал/шум, как функция расстояния может быть определено как:

$$\gamma(R) = \frac{S_{RX}}{P_n} = \frac{1}{P_n} a(R) S_{TX} \quad (3.3)$$

где  $a(R)$  – зависимость затухания сигнала от расстояния, в качестве ее может быть принята одна из известных моделей [63].

Зависимость скорости передачи данных от расстояния, согласно стандарту IEEE 802.11ac, при  $w=20$  МГц, приведена на рисунке 1 (синяя линия). Красная кривая получена согласно выражению (2)  $w=20$  МГц,  $\alpha=0,5$ .

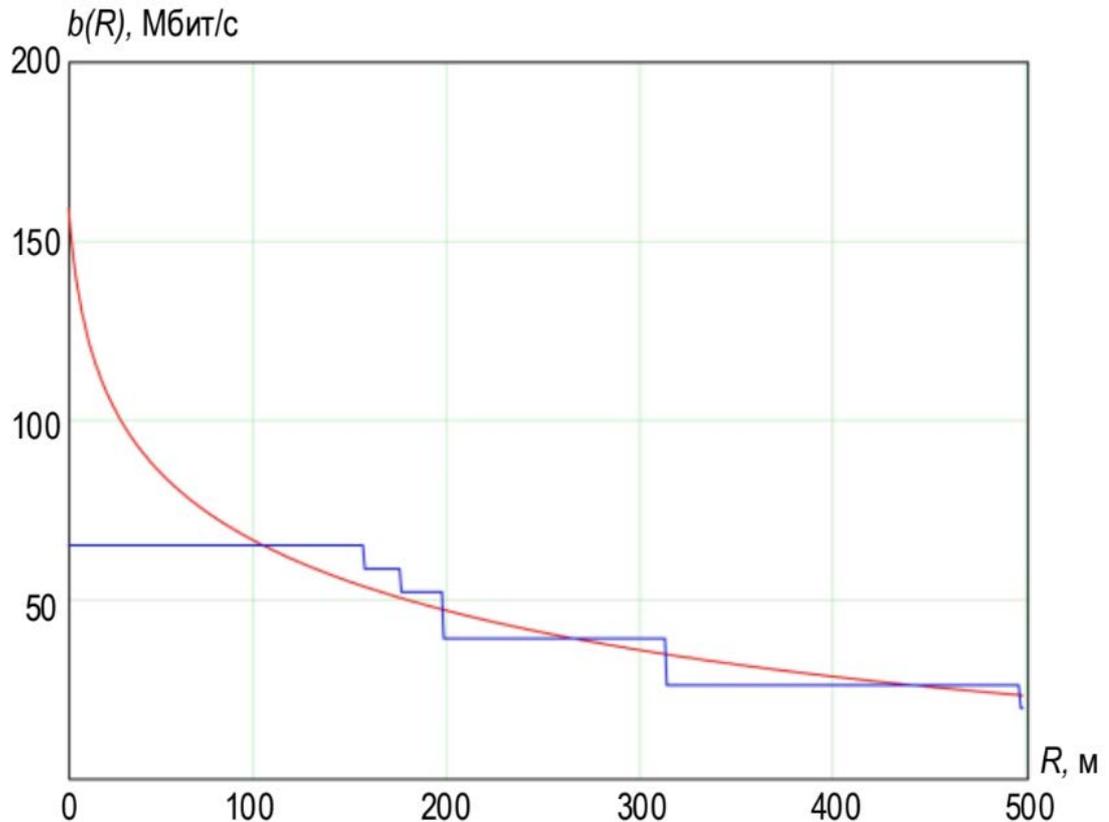


Рисунок 3.2 - Зависимость скорости передачи данных от расстояния между оборудованием пользователя и точкой доступа

Приведенный график показывает достаточно высокую близость модели (2) к зависимости, полученной по данным стандарта. Наибольшая ошибка имеет место при малых расстояниях, т.е. при высоких значениях отношения сигнал/шум. Это можно объяснить технологическими возможностями, которые учитываются в данном стандарте.

Применение выбранного подхода кластеризации решает задачу минимизации суммы расстояний между элементами и центрами кластеров. При описании системы приведенными выше выражениями целевая функция кластеризации будет выражаться как:

$$O = \max_{c_j, j=1 \dots n} \sum_{j=1}^n \sum_{i=1}^{m_j} b(c_j, u_{ji}) \quad (3.4)$$

где  $b(c_j, u_{ji}) = b(d(c_j, u_{ji}))$  достижимая скорость передачи между центром  $j$ -го кластера и  $i$ -м элементом  $j$ -го кластера,

$d(c_j, u_{ji})$  расстояние между центром  $j$ -го кластера и  $i$ -м элементом  $j$ -го кластера.

Таким образом, применение данного метода позволяет выбрать позиции центров кластеров (позиций точек доступа) так, чтобы суммарная скорость передачи в сети была максимальна, при этом условие (1) обеспечивает минимально допустимую величину скорости передачи данных.

Решение (4) дает близкие к оптимальным позиции центров кластеров, по отношению к элементам, которыми являются пользователи. Однако, в данном случае необходимо учитывать положение центров кластеров относительно базовых станций сети подвижной связи и относительно друг друга.

Будем полагать, что  $R_C$  – радиус зоны связи кластера,  $R_M$  – радиус зоны связи с маршрутизатора с базовой станцией сети подвижной связи. Эти параметры должны быть введены в виде ограничений в используемый алгоритм кластеризации.

Особенность данного алгоритма состоит в том, что выбор позиции центра кластера производится с учетом упомянутых ограничений, а именно

$$\begin{aligned} \exists j \in \{1 \dots n\}, \quad d(c_j, B) \leq R_M \\ \exists (j, i) \in \{1 \dots n\}, \quad j \neq i, \quad d(c_j, c_i) \leq R_C \end{aligned} \quad (3.5)$$

Эти ограничения, фактически, определяют то, что как минимум один из маршрутизаторов должен быть в зоне связи базовой станции сети подвижной связи, и каждый из маршрутизаторов должен быть в зоне связи хотя бы одного из соседних маршрутизаторов. Эти ограничения обеспечивают связность между маршрутизаторами, расположенными на БПЛА и связность с сетью подвижной связи.

Эти ограничения могут быть усилены заданием минимального количества связей между маршрутизаторами  $r_{\min}$  и с базовыми станциями сети подвижной связи  $M_{\min}$

$$\begin{aligned} \exists J \in \{1 \dots n\}, \quad |J| \geq M_{\min}, \quad d(c_j \in J, B) \leq R_M \\ \exists L \in \{1 \dots n\}, \quad |L| \geq r_{\min}, \quad d(c_j \in L, c_i \in L) \leq R_C, \quad j \neq i. \end{aligned} \quad (3.6)$$

Псевдокод, предлагаемого алгоритма приведен ниже.

Исходные данные:

-множество объектов кластеризации (пользователей)  $U = \{u_i\}, \quad i = 1 \dots m$ .

Каждому из объектов соответствует точка с заданными координатами  $u_i \boxtimes p(u_i) = (x_i^{(u)}, y_i^{(u)}, z_i^{(u)}), \quad i = 1 \dots m$ ;

-множество базовых станций сети подвижной связи  $V = \{v_i\}, \quad i = 1 \dots k$ .

Каждому из объектов соответствует точка с заданными координатами  $v_i \boxtimes p(v_i) = (x_i^{(k)}, y_i^{(k)}, z_i^{(k)}), \quad i = 1 \dots M$ ;

-множество центров кластеров  $C = \{c_i\}, \quad i = 1 \dots n$

Каждому из центров кластеров  $c_j \quad j=1 \dots n$  соответствует точка с заданными координатами

$$c_i \boxtimes p(c_i) = (x_i^{(c)}, y_i^{(c)}, z_i^{(c)}), \quad i = 1 \dots n$$

Алгоритм

$U_0 = \{u_i\}, \quad i = 1 \dots m$ ; // Множество некластеризованных объектов

$C = \emptyset$  // Множество кластеров

$i=1$ ;

//Случайная точка в зоне обслуживания

$$p(c_i) = \{randomPos(x, y, z) \in S, \quad d(c_i, v_q) \leq R_M, \quad d(c_i, c_j) \leq R_C, \quad v_q \in V\}$$

$C_0 = \{u_j \in U_0; d(c_i, u_j) \leq R\}$ ; //Образовать кластер  $C_0$

While  $U_0 \neq \emptyset$

{

//Вычислить центр кластера

$cm = S(U, V, R)$

//Если центр масс не совпадает с центром кластера и он находится на удалении не превышающим  $R_C$  от центра другого кластера

if  $cm \neq p(c_i)$

if  $\exists c_j \in C, d(cm, c_j) \leq R$

$p(c_i) = cm$  //Перейти в центр масс

else

$\omega = \omega + \Delta\omega$ ;

else

{

$U_0 = U_0 \setminus C_0$  //Исключить элементы кластера из множества

некластеризованных элементов

$C = C \cup cm$

}

} //End while

ReDistributeNones(*cond*)//Опционально

Центр кластера (центр масс) вычисляется согласно следующему выражению:

$$S_m(C_0, V, R) = \left\{ \begin{array}{l} x_0 = \frac{1}{K} \left( \sum_{i=1}^K x_i \eta_i + x_{C_n} \omega \right), y_0 = \frac{1}{K} \left( \sum_{i=1}^K y_i \eta_i + y_{C_n} \omega \right), \\ z_0 = \frac{1}{K} \left( \sum_{i=1}^K z_i \eta_i + z_{C_n} \omega \right) \end{array} \right\},$$

$$K = |C_0|, \quad (x_i, y_i, z_i) = p(u_i), \quad u_i \in C_0, \quad (x_{C_n}, y_{C_n}, z_{C_n}) = p(c_n), \quad c_n \in C, \quad \eta_i, \omega \geq 0 \quad (3.7)$$

где  $K$  – число элементов кластера,

$x_0, y_0, z_0$  – координаты центра кластера,

$\eta_i$  – весовой коэффициент элемента (пользователя),

$x_{C_n}, y_{C_n}, z_{C_n}$  – координаты центра ближайшего соседнего кластера,

$\omega$  – весовой коэффициент.

Особенность вычисления центра кластера состоит в том, что при этом учитывается расстояние до центра ближайшего кластера. Изменяя величину  $\omega$  можно управлять «тяготением» центра искомого кластера к ближайшему соседнему кластеру.

Приведенный выше алгоритм описывает общий трехмерный случай, когда кластер представляет собой сферу. Ввиду недостаточной наглядности представления результатов кластеризации в пространстве, на рисунке 3.3 приведен пример кластеризации на плоскости. В данном примере моделируется кластеризация 1000 пользователей, распределенных на плоскости. Для моделирования используется точечный процесс Гиббса [64].

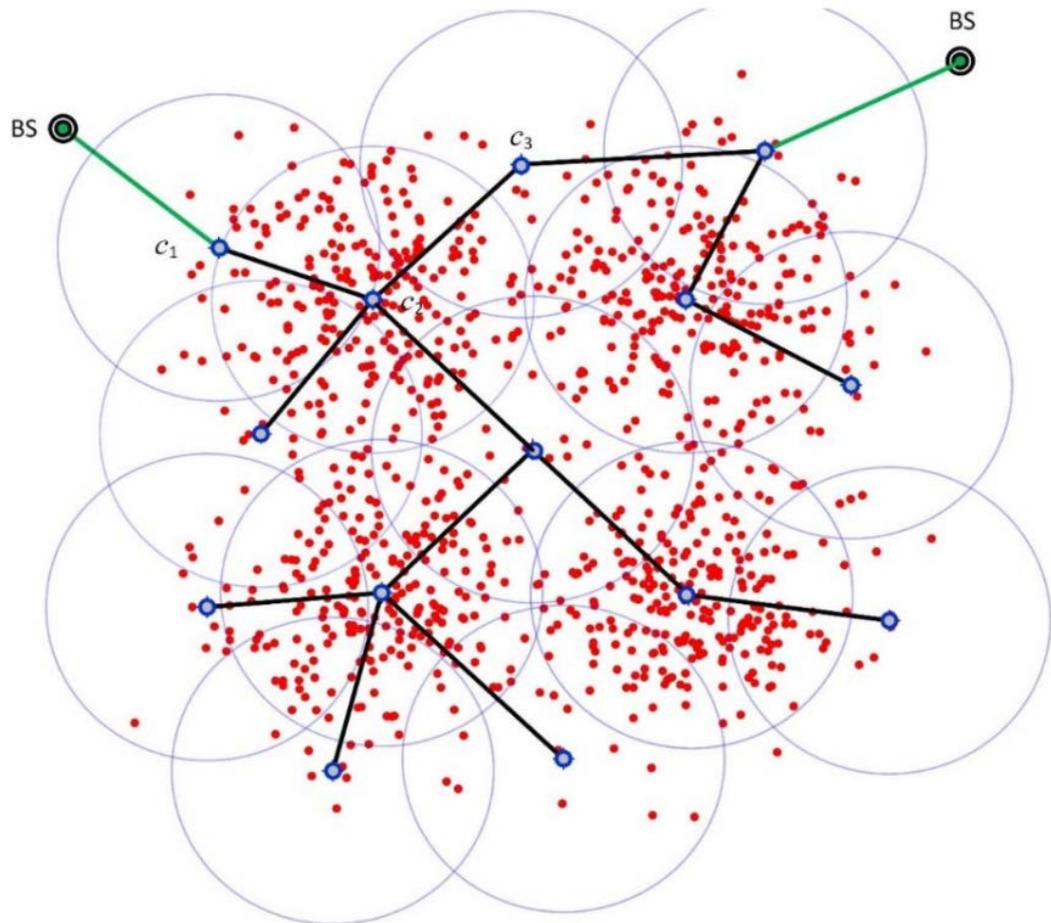


Рисунок 3.3 – Результат кластеризации и выбора структуры сети

После выполнения, приведенного выше алгоритма, будут найдены центры кластеров. На этом алгоритм может быть завершен, а элементы могут быть отнесены к кластерам в том порядке, в котором выделялись кластеры в ходе выполнения алгоритма. Однако, такой подход дает хорошие решения в случае наличия скоплений элементов (пользователей), причем размер этих скоплений не превышает размера кластера  $R$ . В противном случае размеры кластеров могут сильно отличаться.

В таком случае целесообразно по завершении алгоритма выполнить перераспределение элементов путем отнесения их к ближайшим центрам кластеров. В приведенном выше алгоритме эта операция выполняется процедурой `ReDistributeNones(y/n)`, аргумент имеет смысл выполнять операцию перераспределения или нет.

После операции перераспределения элементы кластеры будут иметь форму многогранников, стороны которых определяются диаграммой Вороного [65], построенной относительно найденных центров кластеров, пример на рисунке 3.4.

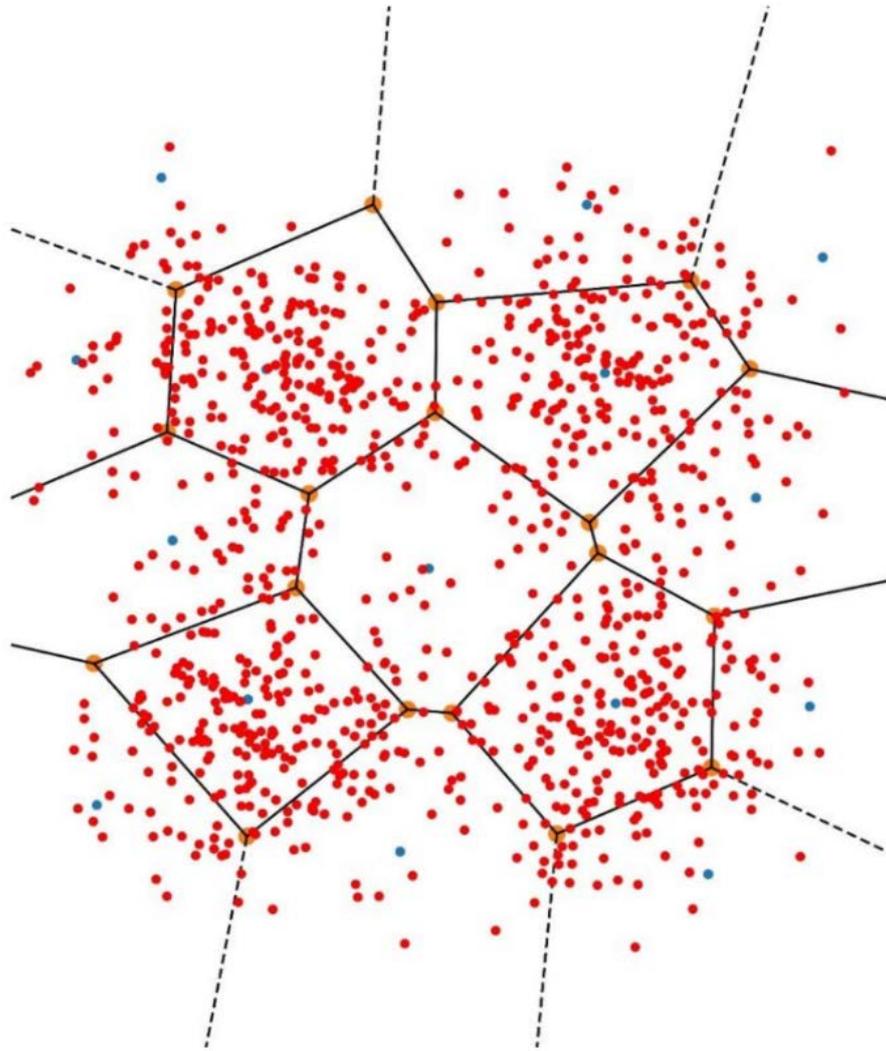


Рисунок 3.4 – Результат модификации формы кластеров

В качестве критерия необходимости перераспределения предлагается использовать дисперсию количества элементов в кластерах, определяемую как:

$$cond = \begin{cases} true & D_{II}(K) < D_I(K) \\ false & D_{II}(K) \geq D_I(K) \end{cases}, \quad (3.8)$$

где  $D_I$ ,  $D_{II}$  – дисперсии числа элементов кластеров для алгоритма без перераспределения элементов и с их перераспределением, соответственно.

Дисперсии для обоих случаев могут быть найдены по выражению:

$$D(K) = \frac{1}{n} \sum_{i=1}^n (K_i - E(K))^2, \quad E(K) = \frac{1}{n} \sum_{i=1}^n K_i, \quad (3.9)$$

где  $n$  – число кластеров,

$K_i$  – число элементов в  $i$ -м кластере.

Предложенный метод и алгоритм позволяют произвести кластеризацию пользователей в заданной зоне обслуживания с учетом допустимых расстояний между центрами кластеров и допустимых расстояний до базовых станций сети подвижной связи. Благодаря этим качествам данный метод позволяет построить связную mesh сеть на основе маршрутизаторов, размещаемых в центрах кластеров, имеющую связь с базовыми станциями сети подвижной связи. Алгоритм кластеризации обеспечивает выделение кластеров по максимуму качества обслуживания.

### 3.3 Маршрутизация трафика

Найденные позиции маршрутизаторов (центры кластеров) объединены в mesh сеть, структура которой может быть определена кратчайшими маршрутами между отправителем и получателем. Под кратчайшим маршрутом будем понимать маршрут, обеспечивающий наибольшее качество обслуживания трафика. Будем полагать, что качество обслуживания характеризуется такими показателями как: достижимая скорость передачи данных, задержка доставки данных, вероятность потери данных.

Структура mesh сети определяется матрицей:

$$ST = \begin{bmatrix} l_{11} & \dots & l_{1n} \\ \vdots & & \vdots \\ l_{n1} & \dots & l_{nn} \end{bmatrix}, \quad l_{ij} = \begin{cases} q_{ij} & l_{ij} \leq R_c \\ \infty & l_{ij} > R_c \end{cases} \quad (3.10)$$

где  $q_{ij}$  показатель качества линии связи между узлами сети,

$l_{ij}$  – характеристика линии связи между узлами сети.

В качестве показателя качества предлагается интегральный показатель, учитывающий скорость передачи данных, нагрузку и вероятность потерь:

$$q_{ij} = -\omega_1 b_{ij} + \omega_2 \rho_{ij} + \omega_3 p_{ij} \quad (3.11)$$

где  $\omega_1, \omega_2, \omega_3$  – коэффициенты условной стоимости единицы соответствующего показателя.

Интегральный показатель имеет смысл условного выигрыша или расходов, в зависимости от знака. Знак слагаемого в выражении (11) положителен для показателей, рост которых ведет к росту условных расходов, в данном случае к уменьшению качества обслуживания.

В таком случае использование любого метода поиска кратчайшего пути для сети, описываемой матрицей (10) позволяет выбрать маршрут, имеющий минимальное значение суммы коэффициентов для каждого из ребер, входящих в него, формально можно записать как:

$$\Omega_{st} = \arg \left( \Omega_{st} = \Omega_{st} \cup q_{ij} \mid \min_{q_{ij} \in ST} \sum_{r=1}^{L_{st}} q_{ij}, \quad i, j = 1 \dots n, \quad i \neq j \right) \quad (3.12)$$

где  $\Omega_{st}$  множество вершин (узлов), входящих в кратчайший путь между вершинами  $s$  и  $t$ .

Для нахождения всех кратчайших путей может быть использован, например, алгоритм Данцига [66].

### 3.4 Эффективность

Для оценки эффективности предложенных методов, в качестве альтернативы, выбран не модифицированный алгоритм FOREL. При его применении распределение маршрутизаторов в зоне обслуживания не гарантирует связности mesh сети и доступности сети подвижной связи. Положения маршрутизаторов в таком случае случайны и зависят от распределения пользователей, но никак не ограничены в части взаимного расположения. В таком случае, связность образуемой ими сети также является случайной величиной.

Если маршрутизаторы, размещенные в выбранных позициях (центрах кластеров), не образуют связной сети, то пользователи, включенные в эти маршрутизаторы, не имеют доступа к другой части сети и выхода во внешнюю сеть. Таким образом, для сравнения методов наиболее показательной характеристикой является связность сети, образованной при размещении маршрутизаторов.

Связность сети можно описать вероятностью связности, которую можно оценить долей возможных маршрутов в сети:

$$con = \frac{n_e}{(n+1)^2} \quad (3.13)$$

где  $n$  – количество кластеров (маршрутизаторов),  $n_e$  – количество доступных маршрутов.

Величина  $(n+1)^2$  это максимально возможное количество маршрутов между всеми  $n$  узлами сети и одним узлом, описывающим базовую станцию сети подвижной связи, которое имеет место при полной связности сети.

Вероятность связности (13) в случае не модифицированного алгоритма не превышает единицы, а в случае модифицированного алгоритма она всегда равна единице. Величину  $1-con$  будем рассматривать как выигрыш, полученный при использовании предложенных методов:

$$e = 1 - con \quad (3.14)$$

Эта величина зависит от радиуса зоны связи маршрутизаторов  $R_C$  и  $R_M$ , полагая, что эти величины равны  $R_C = R_M$ , можно оценить зависимость  $e(R_C/W)$ , где  $W$  – размер зоны обслуживания.

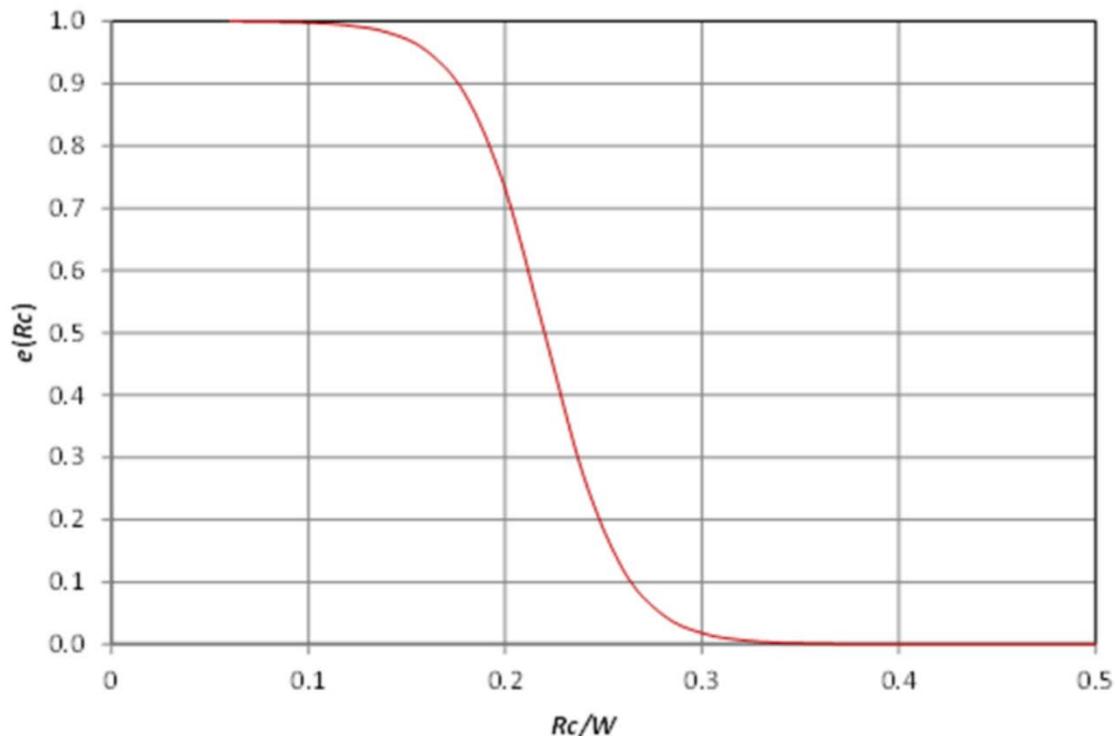


Рисунок 3.5 – Зависимость эффективности метода от радиуса зоны связи между кластерами  $R_C$

Можно заметить, что применение метода целесообразно, когда радиус зоны связи маршрутизаторов в mesh сети меньше размера зоны обслуживания. Это естественно, т.к. в противном случае, любое расположение маршрутизатора равнозначно, с точки зрения обеспечения связности.

В приведенном выше примере, при размере зоны обслуживания 500x500 м. и заданном распределении пользователей, при размере зоны связи маршрутизатора около 0,23 размера зоны обслуживания эффективность предложенного метода составляет около 50%.

Эффективность метода тем выше, чем меньше отношения радиуса зоны связи маршрутизатора к размеру зоны обслуживания. Таким образом его применение целесообразно при необходимости обслуживания относительно больших площадей или объемов.

### **3.5 Вывод по главе 3**

1. Моделирование линии связи между маршрутизатором и пользователем с помощью модели К. Шеннона с учетом технологических отличий его реализации в действующем стандарте сети беспроводного доступа позволяет решать задачу кластеризации пользователей как задачу максимизации суммарной достижимой скорости передачи данных.

2. Предложенный алгоритм кластеризации, являющийся модификацией алгоритма FOREL, обеспечивает выбор позиций маршрутизаторов для обслуживания пользователей и организации связной mesh сети и связи этой сети с базовой станцией подвижной связи.

3. Модификация алгоритма кластеризации, обеспечивающая перераспределение элементов кластеров, позволяет повысить качество кластеризации путем выравнивания количества элементов выделяемых кластеров.

4. Метод маршрутизации трафика в mesh сети обеспечивает выбор маршрутов на основе минимизации интегрального показателя качества маршрута, учитывающего достижимую скорость передачи данных, величину нагрузки и вероятность потерь.

5. Предложенные методы распределения маршрутизаторов в рое БПЛА или на единичных или привязных БПЛА эффективен в случае, когда размер зоны связи маршрутизатора меньше размера зоны обслуживания.

## **ГЛАВА 4. АЛГОРИТМ МИГРАЦИИ СЕРВИСОВ ДЛЯ РАСПРЕДЕЛЕННЫХ ГРАНИЧНЫХ ВЫЧИСЛЕНИЙ В СЕТЯХ 5G/6G**

Статистика «Data Age 2025» от международной исследовательской компании IDC, которая анализирует тенденции развития информационных технологий и дает прогнозы на основе собственных данных, свидетельствует о том, что к 2025 году к Интернету будет подключено более 100 млрд устройств, а объем данных об устройствах достигнет 300 Зб [67, 68]. При традиционных схемах обработки все полученные данные необходимо передавать в облако в главный центр обработки данных для анализа. Из-за такого огромного объема данных облачная вычислительная платформа сталкивается с проблемами - высокая задержка в сети, большое количество подключенных устройств, сложная обработка больших объемов данных, недостаточная пропускная способность и высокое энергопотребление. В результате возникла необходимость в технологии, которая сегодня называется Edge Computing [69, 70]. Эта технология представляет собой концепцию распределенных вычислений, осуществляемых в пределах досягаемости конечных устройств. Такой тип вычислений используется для сокращения времени отклика сети и более эффективного использования ее пропускной способности. Решения, обеспечивающие отказоустойчивость сети при обработке больших объемов данных, включают поддержку подходов миграции сервисов в различных системах [71]. Эта технология предполагает перенос сервисов с одного ресурса на другой для обеспечения отказоустойчивости системы в целом. В данной работе будет описано применение граничных вычислений в виде переноса процессов на граничный вычислительный кластер, а также представлены результаты исследования существующих процессов миграции, полного времени переноса и времени простоя, выявления оптимального процесса миграции для дальнейшей реализации алгоритма автоматизации миграции сервисов.

## 4.1 Технологии развертывания и миграции сервисов

### **Kubernetes**

Kubernetes — это платформа с открытым исходным кодом, автоматизирующая контейнерные операции в Linux. Она устраняет многие ручные процессы, связанные с развертыванием и масштабированием контейнерных приложений. Другими словами, вы можете объединить в кластер группы узлов, на которых запущены контейнеры Linux, а Kubernetes поможет вам легко и эффективно управлять этими кластерами [72]. Слово «Kubernetes» происходит от древнегреческого κυβερνήτης, что означает капитан, рулевой, пилот; тот, кто управляет. В аббревиатуре «K8s» число 8 — это восемь букв между K и S.

### **Программный интерфейс Kubernetes**

Kubernetes API — это набор объектов и функций, используемых для взаимодействия с платформой Kubernetes. Эти объекты можно использовать для управления ресурсами, такими как капсулы и узлы, а также для выполнения таких задач, как планирование и масштабирование. API также облегчает взаимодействие между компонентами платформы, позволяя эффективно распределять и управлять ресурсами. Одной из ключевых особенностей K8s API является его расширяемость, позволяющая разработчикам добавлять пользовательскую функциональность с помощью пользовательских определений ресурсов. В целом, Kubernetes API играет важнейшую роль в управлении и использовании ресурсов кластера Kubernetes. Являясь частью проекта с открытым исходным кодом, Kubernetes API постоянно развивается, предлагая новые возможности и обновления. Доступ к нему можно получить различными способами, включая использование инструментов командной строки, таких как kubectl, или взаимодействие с графической панелью управления. Кроме того, разработчики могут взаимодействовать с API напрямую через HTTP-запросы или с помощью клиентских библиотек на таких языках

программирования, как Java и Go. В целом, Kubernetes API предоставляет широкие возможности для управления и использования кластера Kubernetes.

В кластере K8s все ресурсы представлены в виде объектов API, часто описанных на языке разметки в формате YAML. Все объекты кластера имеют метаданные, которые их описывают, причем структура такого описания одинакова для всех типов объектов. Имя, используемое для объекта, должно быть уникальным среди всех объектов этого типа в данном пространстве имен (комбинация полей имени и пространства имен). Пространства имен Author Proof позволяют разделить один кластер K8s на несколько виртуальных с целью изоляции ресурсов друг от друга. В свою очередь, сервер API присваивает каждому объекту уникальный идентификатор (UID) для однозначной идентификации во всем кластере.

### **Pods**

Pod — это наименьшая единица выполнения в Kubernetes. Pod состоит из группы одного или нескольких контейнеров с общими ресурсами хранения и сети, а также спецификацией того, как запускать контейнеры. Например, предположим, что у нас есть 5 контейнеров, готовых к развертыванию в Kubernetes. Мы можем поместить все 5 контейнеров в один под, если все они должны работать с одинаковой конфигурацией, или разделить контейнеры в зависимости от их назначения и управления нагрузкой. Капсулы также могут поддерживать тома, как и docker, и их можно легко настроить в зависимости от требований. Каждая нода привязывается к узлу, на котором он запланирован, и остается там до завершения (в соответствии с политикой перезапуска) или удаления. В случае отказа узла идентичные поды планируются на других доступных узлах кластера [73].

### **Балансировка сети и нагрузки**

В кластере K8s каждая нода имеет свой собственный IP-адрес, который разделяется между контейнерами этой ноды. Контейнеры внутри ноды могут взаимодействовать друг с другом. Однако эти сетевые возможности не реализованы в самом K8s, для их использования необходимо установить сетевой плагин,

который через сетевой интерфейс (Container Network Interface, CNI) будет управлять сетью в кластере [74]. Ресурсы типа Service создаются для реализации сервисных функций, таких как балансировка нагрузки. Используя метки нод, сервис определяет, какие ноды должны быть доступны через него, затем создает конечные точки (Endpoint) для каждой ноды. При необходимости доступа к внешним сервисам можно вручную создать конечную точку. Помимо описанного способа, доступ к сервису можно получить через переменные окружения и записи DNS, а также, при необходимости, вручную указать IP-адрес кластера или порт узла [75].

### **Архитектура кластера kubernetes**

Традиционно кластер K8s состоит из двух плоскостей: плоскости управления (control plane) и плоскости данных (data plane) (рисунок 4.1). Плоскость управления может использоваться для управления тысячами рабочих узлов (node) благодаря возможностям масштабирования. Она включает в себя API-сервер, который хранит информацию об объектах в распределенном высокопроизводительном хранилище типа ключ-значение - etcd. Контроллеры, упомянутые выше в контексте заданий, также являются частью плоскости управления, работая через kube-controller-manager (c-m). Этот менеджер упрощает развертывание, объединяя различные контроллеры в один модуль. Еще одна обязательная часть плоскости управления - планировщик (kube-scheduler, sched). Его задача - заботиться о том, чтобы оптимальным образом планировать размещение капсул на узлах [76]. Также стоит упомянуть опциональный модуль для взаимодействия с облачным провайдером, который расширяет возможности кластера. Плоскость данных включает в себя узлы - серверы кластера, на которых выполняется работа.

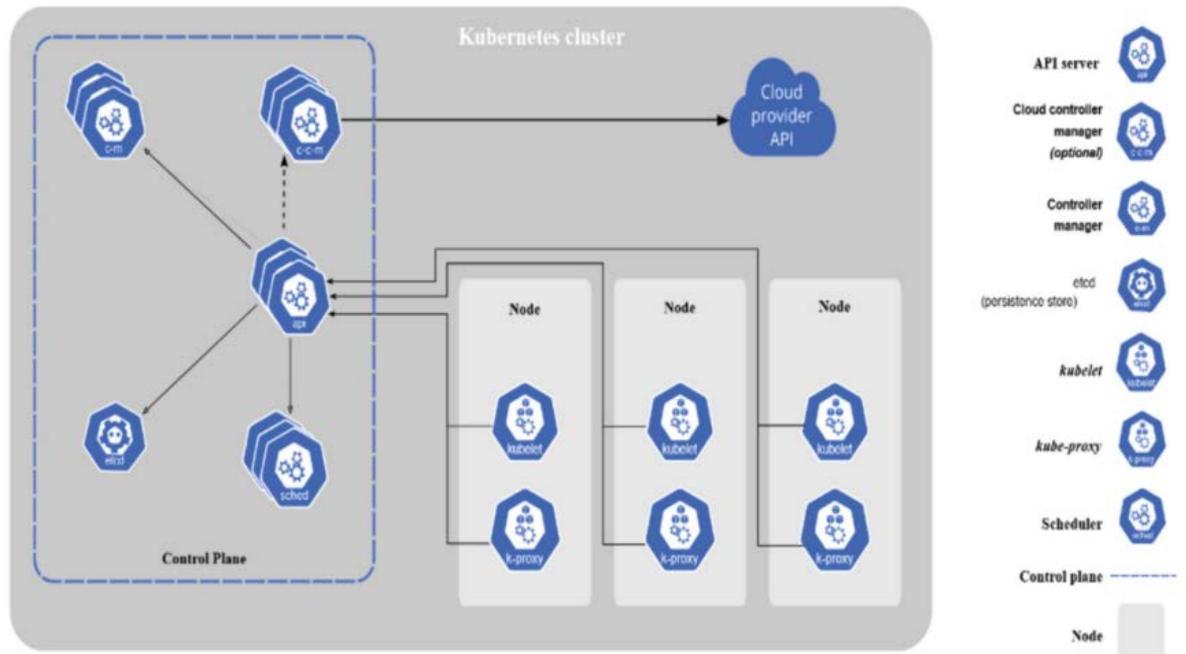


Рисунок 4.1 – Схема кластера кubernetes

На каждом таком узле запускается kubelet-агент, который взаимодействует с API-сервером и запускает поды, предназначенные для этого узла. Агент следит за состоянием подов и запущенных в них приложений, обновляет статусы в API. Кроме того, на узлах запускается компонент kube-proxy, который обрабатывает трафик для данного узла [77].

Для управления контейнерами агент kubelet взаимодействует со средой выполнения контейнеров через специальный интерфейс CRI (Container Runtime Interface). Этот интерфейс определяет возможности для создания, изменения и удаления контейнеров совместимого с K8s типа. Это позволяет менять платформу контейнеризации, на которой работает K8s, даже без явной поддержки со стороны последней.

### Основные технологии миграции

Для миграции службы необходимо перенести контейнеры, на которых работает служба. Поскольку контейнеры — это процессы, выполняющиеся в изолированной среде, вопрос переноса сводится к миграции процессов. Для системы Linux существует технология создания контрольных точек и

восстановления процессов (Checkpoint-Restore), которая появилась еще в 1970-х годах и с тех пор значительно модернизировалась. Изначально для копирования области памяти, используемой процессом, необходимо было получить привилегированный доступ в системе, что требовало соответствующей поддержки на уровне ядра операционной системы. Однако этот метод позволял осуществлять прозрачную миграцию, то есть для переноса не требовалось вносить изменения в саму программу.

Дальнейшее развитие технологии Checkpoint-Restore (CR) с возможностью хранения всех данных памяти процесса в пользовательском пространстве позволило избавиться от ограничений, связанных с доступом к памяти. В начале 2010-х годов в открытом доступе появилось программное решение - CRIU (Checkpoint-Restore In Userspace) [78]. Это решение использует стандартную технику переноса процесса:

- замораживание процесса;
- сохранение состояния на диск;
- перенос сохраненного состояния в место восстановления;
- возобновление процесса.

Основным отличием CRIU от CR была возможность работать с памятью процесса без излишних привилегий. Изоляция процессов, обеспечиваемая контейнеризацией, делает восстановление процесса (контейнера) на новом месте еще проще, поскольку технология CRIU старается выполнить перенос максимально прозрачно, копируя, в том числе, идентификаторы процессов (PID), а поскольку среды выполнения изолированы, вероятность столкновения этих идентификаторов при восстановлении крайне мала.

## 4.2 Исследование существующих технологии миграции услуг

### 4.2.1 Анализ средств конфигурирования оркестровки

В данной работе в качестве оркестратора используется K8s. Для решения задач по определению оптимальной конфигурации для развертывания и запуска используются следующие инструменты:

- Сопоставление свойства nodeSelector ноды с метками узлов;
- Affinity и Anti-affinity;
- Прямое указание ноды на имя узла;
- Ограничения на топологию размещения нод;

#### **Сопоставление свойства nodeSelector ноды с метками узлов**

Этот метод наиболее прост и популярен благодаря своей понятной логике и легкости применения. В Kubernetes есть инструменты для маркировки большинства объектов, поэтому вы можете вручную добавить метку к нужному узлу. Для этого достаточно добавить поле nodeSelector в декларативное описание конфигурации ноды. Кроме того, существует инструмент автоматической маркировки стандартных ролей в кластере, со списком которых вы можете ознакомиться на официальном сайте. Добавление меток к узлам позволяет планировать запуск подкастов на конкретном узле или группе узлов. Это позволяет быть уверенным в том, что определенные поды будут запущены только на узлах с заданными свойствами, изоляцией и уровнем безопасности. В случае использования меток с целью избежать запуска на определенном узле, следует использовать такие ключи меток, которые kubelet не сможет изменить. Таким образом, безопасность кластера повышается, поскольку даже при обращении к узлу и попытке установить метку для размещения интересующего злоумышленника произойдет ошибка и рабочая нагрузка не будет скомпрометирована.

### **Affinity и Anti-affinity**

Этот метод, по сравнению с `nodeSelector`, расширяет возможности выбора и ограничений, которые можно задать для капсулы. Так, в случае нескольких меток для размещения в случае `nodeSelector` все они должны совпадать, а метод `affinity` дает больше возможностей и гибкости для определения логики. Предусмотрена возможность задать не строгую, а предпочтительную метку, тогда капсула будет запущена, даже если узел с указанной меткой не будет найден. Отдельно при размещении можно ориентироваться на метки других под, запущенных на узле, тем самым задавая ограничения на одновременную работу под на одном узле.

### **Прямое указание поду на имя узла**

Это правило жестко фиксирует имя узла для размещения на нем ноды, приоритет такого указания (`nodeName`) выше, чем у правил `nodeSelector` и `Affinity`. Прямое указание - одно из самых надежных правил, позволяющее получить ожидаемое поведение кластера и гарантированное размещение приложений и сервисов, но при его использовании следует обратить внимание на следующее:

- если в кластере нет узла с таким именем, то нода не будет размещен и запущен;
- если на узле с таким именем недостаточно ресурсов для запуска этой ноды, то нода не будет запущена, и появится журнал ошибок с указанием причины (нехватка памяти или процессорного ресурса);
- имена узлов при развертывании в облаке часто непредсказуемы и могут меняться в процессе работы.

### **Ограничения топологии размещения подов**

Этот инструмент должен использоваться для управления распределением нод по кластеру с учетом используемых меток и свойств, таких как зоны размещения и регионы, заданные при создании кластера. Такие правила используются для повышения производительности, уменьшения времени ожидания ответа сервиса и улучшения его доступности.

Инструмент для использования ограничений возник из-за неоптимального распределения нод по узлам при использовании географически разнесенных центров обработки данных. При увеличении количества сервисов они начинают генерировать большой служебный трафик между собой по мере роста нагрузки. Для минимизации трафика управления был разработан и реализован механизм, позволяющий разместить группу нод, выполняющих общую задачу, в одной зоне, благодаря чему, помимо увеличения пропускной способности и снижения нагрузки на каналы связи, уменьшается общий отклик системы на запросы пользователей.

### **Taints и Tolerations**

Этот инструмент удобно рассматривать как противоположность Affinity для узлов. Таким образом, правила taint позволяют узлу отказаться от размещения определенных капсул. Допуски, в свою очередь, применяются к нодам, позволяя планировщику сообщать, что, несмотря на запятнанные правила узла, можно разместить ноду или группу нод. Совместное использование Taints и Tolerations гарантирует, что ноды не будут запущены на неподходящем узле. К узлу может быть применено более одного правила запятнанности, что укажет планировщику на недопустимость размещения на узле нод, неустойчивых к ограничениям.

#### 4.2.2 Анализ процесса миграции, времени полного переноса и времени простоя

Чтобы выбрать оптимальную стратегию миграции, необходимо протестировать все описанные методы миграции. В ходе эксперимента будет использовано программное обеспечение CRIU. В качестве нагрузки используется приложение для стресс-тестирования узла K8s - kube-stresscheck [79]. Тестирование проводится в двух режимах нагрузки на оперативную память, низком, когда используется 32 МБ, и высоком - 512 МБ. Для управления этим параметром при

сборке образа необходимо указать соответствующее значение stressDefaultMemoryMB (рисунок 4.2).

```

1 package main
2
3 import (
4     "fmt"
5     "log"
6     "os"
7     "os/exec"
8     "runtime"
9 )
10
11 // #include <unistd.h>
12 import "C"
13
14 const (
15     listenSocket = ":6666"
16
17     stressBinary = "stress"
18     stressTimeout = 60
19     stressIterations = 5
20     stressDefaultMemoryMB = 256
21     stressMemoryHangSec = 2
22 )
23
24 // Common variables
25 var (
26     description = "Simple stress checker for Kubernetes nodes."
27     gitCommit = "n/a"
28     name = "kube-stresscheck"
29     source = "https://github.com/giantswarm/kube-stresscheck"

```

Рисунок 4.2 - Настройки тестового приложения

Для получения статистически значимых результатов выполним миграцию этой ноды четырьмя способами, описанными ранее: холодным, с предварительным и последующим копированием, а также гибридным, по 100 раз каждый. Полное время миграции определяется по данным мониторинга о местоположении соответствующей ноды, а время простоя - по падению и росту нагрузки на ресурсы узла. Полученные данные будут изображены на гистограммах, по уровню синего цвета, которых можно судить о минимальном времени миграции, по красному уровню - о наиболее характерном времени миграции (медиане), а по зеленому - о максимальном времени (рисунок 4.3, рисунок 4.4) [33].

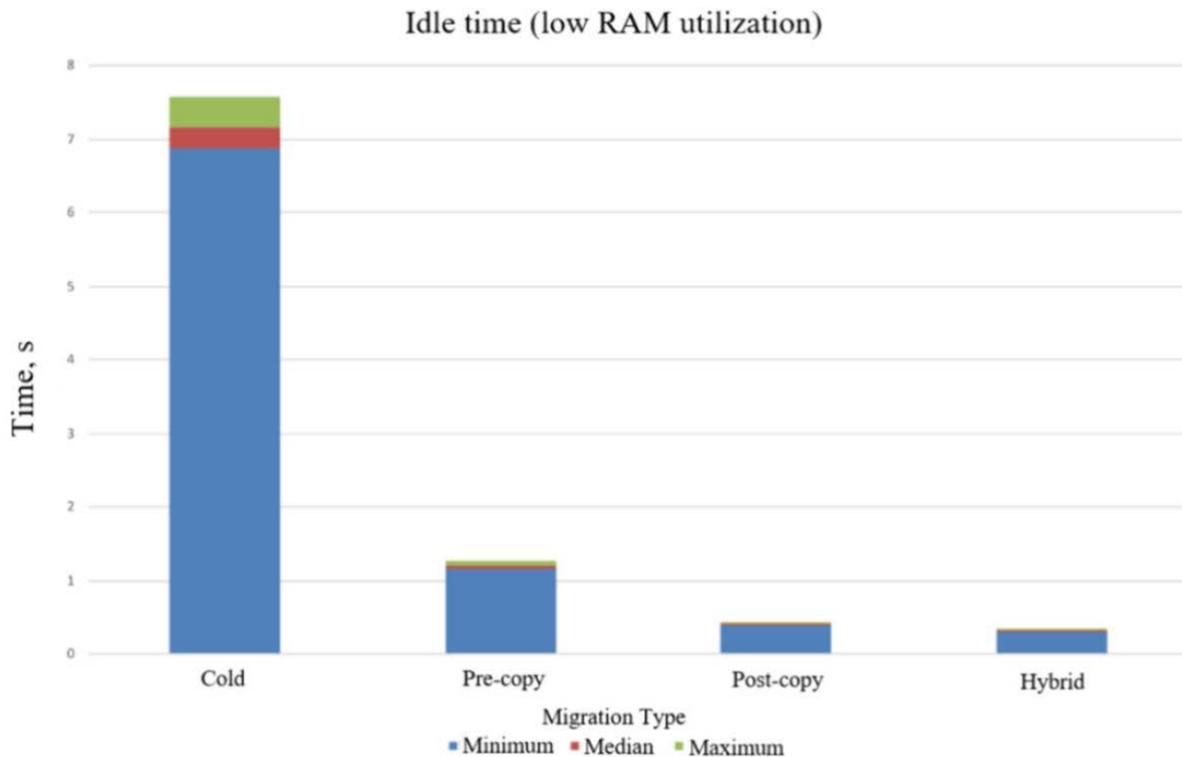


Рисунок 4.3 - График зависимости времени простоя от типа миграции при низкой загрузке памяти

Поскольку некоторые показатели очень близки, для удобства сравнения данные сведены в таблицу 4.1.

### 4.3 Метод автоматической миграции сервисов

Экспериментальная часть будет проводиться на оборудовании, имитирующем облачный кластер и оборудование для пограничных вычислений. В качестве физического оборудования для облачного кластера был использован компьютер на базе процессора x86-64 с установленной системой виртуализации Proxmox. На нем запущены пять виртуальных машин, одна из которых используется в качестве системы развертывания кластера, вторая - в качестве контроллера кластера, а оставшиеся три - в качестве вычислительных узлов. В

качестве базовой ОС на каждой ВМ кластера использована Ubuntu 22.04 LTS, а на вспомогательной - Alpine v3.16. В качестве вычислительного оборудования используется одноплатный компьютер Raspberry Pi на базе процессора архитектуры ARM64. Базовой ОС на Raspberry Pi является Ubuntu 22.04 LTS, а для создания кластера используется программное обеспечение minikube. Полная схема тестового стенда для проведения тестов с сетевыми соединениями между компонентами кластера и кластерами приведена ниже (рисунок 5).

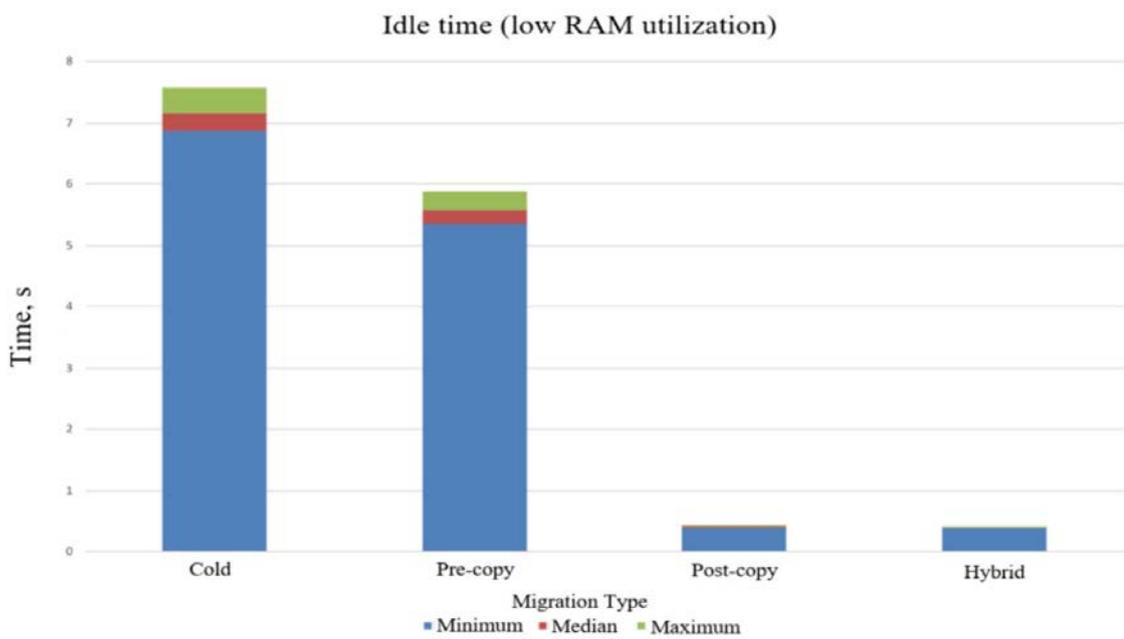


Рисунок 4.4 - График зависимости времени простоя от типа миграции при высокой загрузке памяти

Таблица 4.1 - Временные характеристики процесса миграции различных типов при низком и высоком уровнях использования оперативной памяти

Тип миграции	Cold	Pre-copy	Post-copy	Hybrid
Параметр	Время полной миграции (низкая загрузка оперативной памяти), с			
Минимальный	6,0998	7,2541	8,5962	8,9440
Медианный	6,3557	7,8941	8,7723	10,1920
Максимальный	6,7183	8,8009	9,0216	11,9600
Параметр	Время полной миграции (высокая загрузка оперативной памяти), с			
Минимальный	6,1021	12,2027	8,3870	13,7280
Медианный	6,3581	12,7147	8,7536	14,3040
Максимальный	6,7208	13,4400	9,2375	15,1200
Параметр	Время простоя (низкая загрузка оперативной памяти), с			
Минимальный	6,8663	1,1440	0,3661	0,3127
Медианный	7,1544	1,1920	0,3821	0,3258
Максимальный	7,5625	1,2600	0,4032	0,3444
Параметр	Время простоя (высокая загрузка оперативной памяти), с			
Минимальный	6,8594	5,3387	0,3890	0,3737
Медианный	7,1472	5,5627	0,4060	0,3894
Максимальный	7,5550	5,8800	0,4284	0,4116

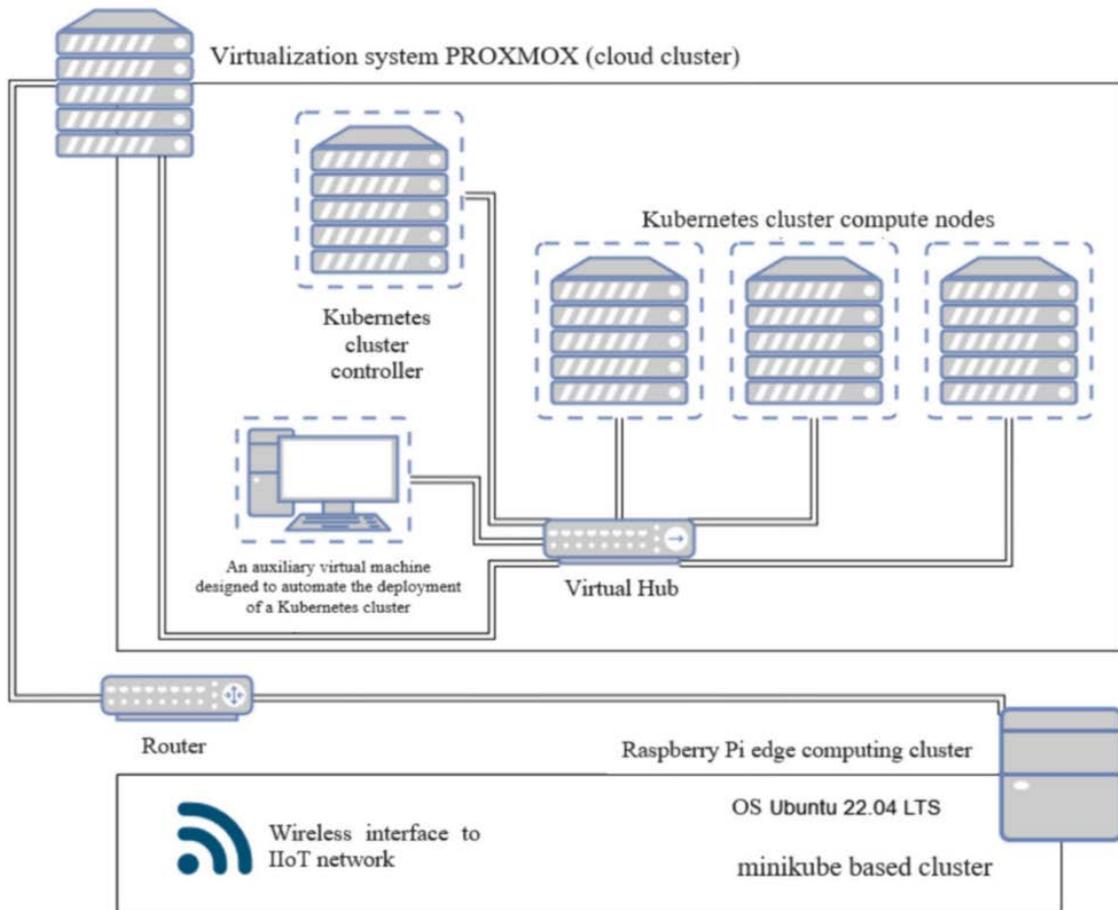


Рисунок 4.5 - Схема тестового ландшафта и сетевых соединений

По результатам проведенных тестов можно утверждать, что для сервисов, требующих быстрого реагирования и минимального времени простоя, оптимальным стандартным методом миграции является гибридный. Несмотря на то, что он имеет самое длительное время полной миграции, время простоя сервиса при использовании этого метода минимально. Это подтверждается исследованиями [80]. В экспериментальной части будет использован гибридный метод.

#### 4.3.1 Разработка алгоритма для автоматизации миграции сервисов

Для автоматизации процесса миграции сервисов предлагается использовать демон, который отслеживает состояние очереди запросов, скорость их выполнения, имеет доступ к информации о нагрузке на процессор и память на узлах, а также о

состоянии коммуникационной сети между кластерами для оперативной оценки времени, затрачиваемого на операции миграции.

При разработке алгоритма, учитывающего состояние системы для принятия решения о миграции, необходимо в первую очередь рассмотреть, какие типы задач будут решать сервисы, определить необходимую скорость реакции системы для каждого типа. Для тестирования алгоритма будут использоваться два типа задач, которые необходимо решать в процессе эксплуатации:

- Тип 1 - оперативные задачи, требующие высокой скорости отклика системы, теряющие актуальность в течение 1 с, критические;
- Тип 2 - второстепенные задачи, требующие больше вычислительных ресурсов, теряют актуальность в течение 10 с, некритичные.

```

1 import time
2
3 start_time = time.time()
4 sum = 0
5 for i in range(1000000):
6     sum += 16**(-i) * (4/(8*i+1) - 2/(8*i+4) - 1/(8*i+5) - 1/(8*i+6))
7
8 print("%s" % (time.time() - start_time))

```

Рисунок 4.6 - Расчетная часть программы испытаний

В качестве тестовой нагрузки будут выступать стандартные Docker-контейнеры для запуска python-приложений. Внутри контейнера будет запущен сервис для вычисления числа Пи с заданной точностью, до 50000 цифр для задач первого типа и до 1000000 - для второго.

Вычисление Пи будет производиться способом, описанным в [81]. Для подсчета используется формула 1:

$$\pi = \sum_{i=0}^N \frac{1}{16^i} \left( \frac{4}{8i+1} - \frac{2}{8i+4} - \frac{1}{8i+5} - \frac{1}{8i+6} \right) \quad (4.1)$$

Чтобы понять, сколько ресурсов требуется для решения задач обоих типов, разместим программное обеспечение на кластере каждого типа. Проведем

статистическое исследование, чтобы получить данные о зависимости времени выполнения кода от объема предоставляемых вычислительных мощностей.

Время актуальности для задач первого типа составляет 1 с, задачи второго типа могут быть обслужены за 10 с. Для определения времени, затрачиваемого на выполнение операций на каждом из кластеров, была запущена тестовая программа (рис. 6), отличающаяся для задач первого и второго типов только диапазоном подстановок - 50000 и 1000000 соответственно.

В результате запуска этого теста получен набор данных, указывающий минимальное, медианное и максимальное время решения задач на каждом из кластеров. Таким образом, можно планировать нагрузку, подаваемую на вход для имитационного моделирования, для качественного сравнения подхода автоматизированного обслуживания и работы планировщика. Графики с указанием значений в исследуемых точках представлены на рис. 7, 8, 9, 10.

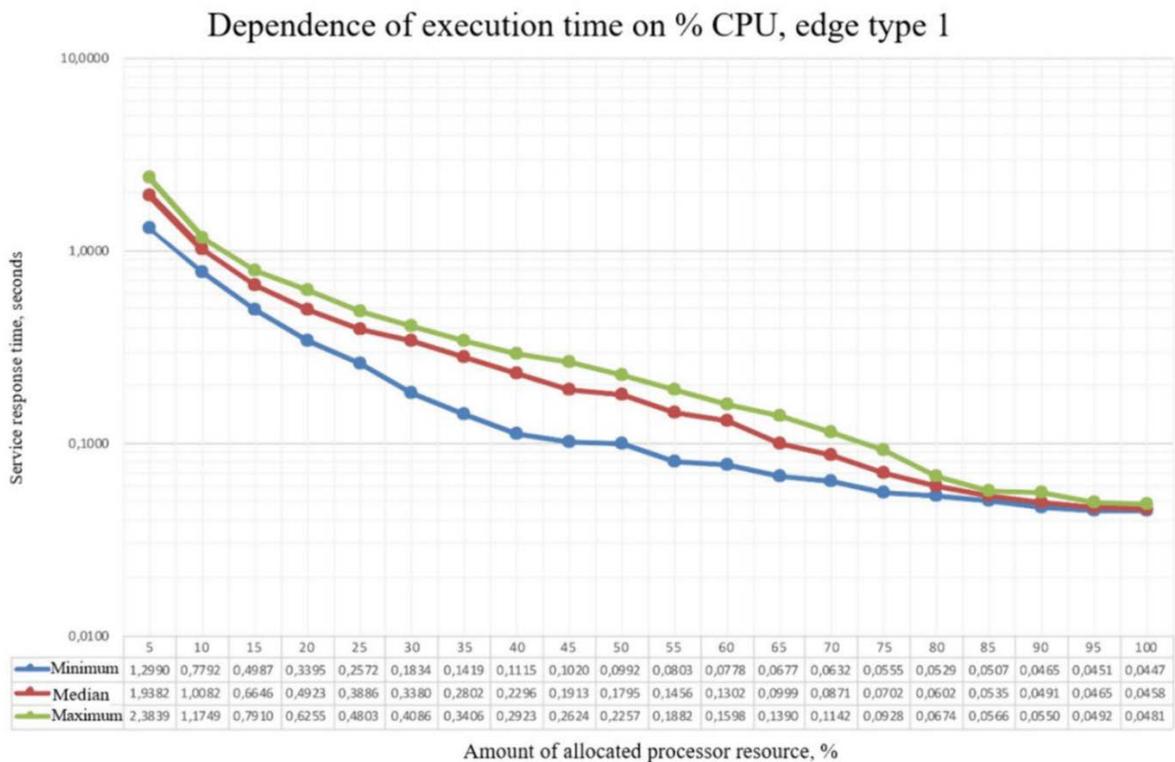


Рисунок 4.7 - График зависимости времени отклика сервиса, расположенного на граничном кластере, от выделенного ресурса процессора для задачи типа 1

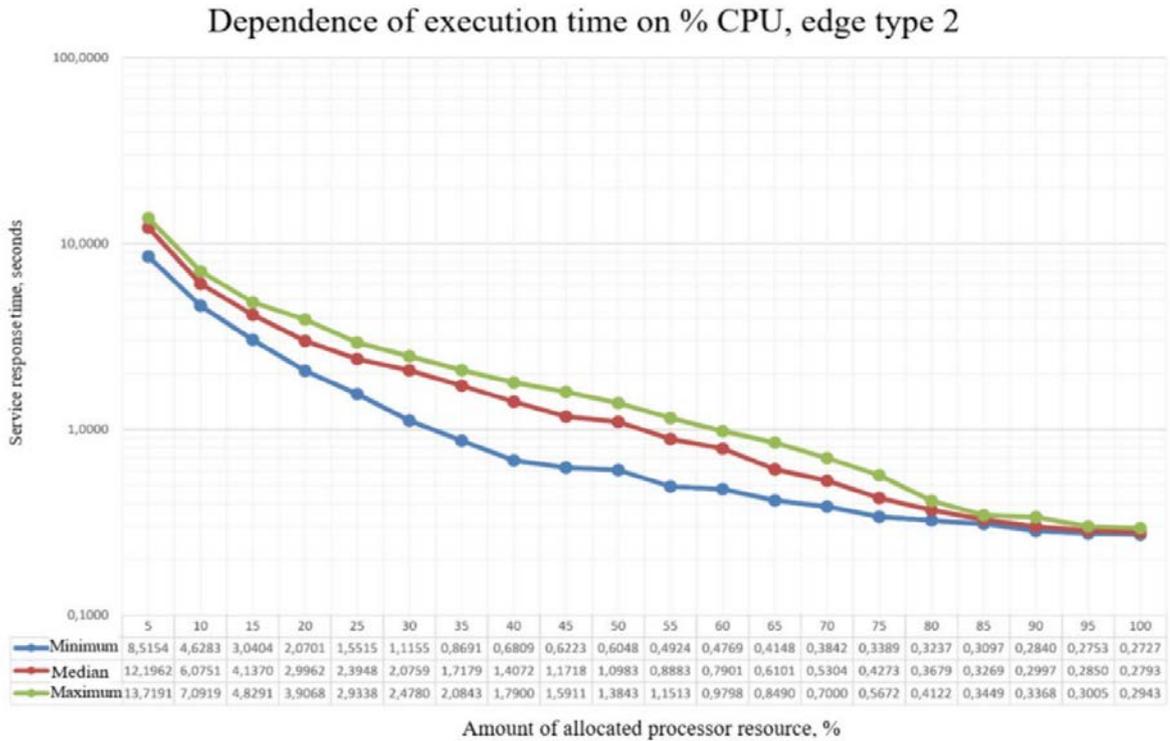


Рисунок 4.8 - График зависимости времени отклика сервиса, расположенного на граничном кластере, от выделенного ресурса процессора для задачи типа 2

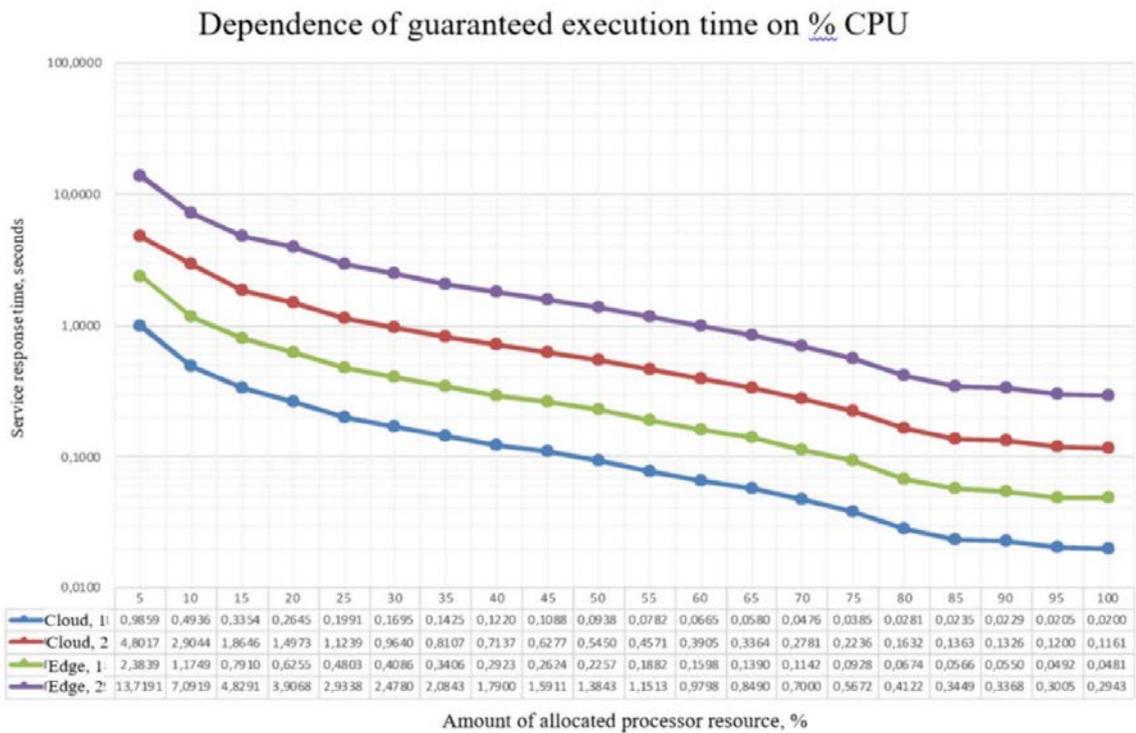


Рисунок 4.9 - График зависимости времени отклика сервиса, расположенного на облачном кластере, от выделенного ресурса процессора для задачи типа 1

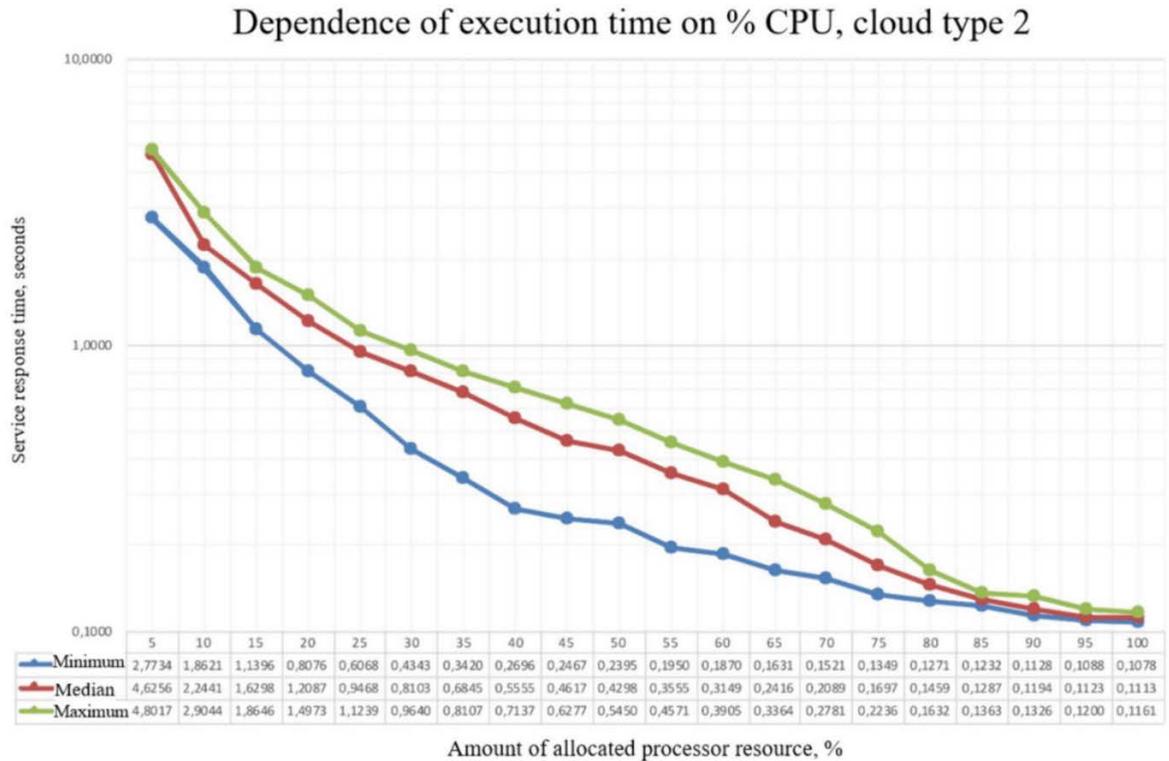


Рисунок 4.10 - График зависимости времени отклика сервиса, расположенного на облачном кластере, от выделенного процессорного ресурса для задачи типа 2

Для определения гарантированного времени выполнения мы используем значения максимального времени

затраченного каждым из кластеров на выполнение задач обоих типов (рисунок 11).

Следует отметить, что увеличение скорости обработки задач в зависимости от предоставленных контейнеру ресурсов происходит экспоненциально (аналогичный результат был получен группой исследователей в [82]), поэтому имеет смысл увеличивать выделение ресурсов в ограниченном диапазоне, так как выигрыш во времени становится меньше, чем при параллельном запуске контейнера. Для определения наиболее удачного выбора доступного процессорного ресурса необходимо произвести расчет для максимизации количества ответов от кластера по формуле:

$$\max N = \text{floor}\left(\frac{T_{z\_max}}{T_{z\_cpu}}\right) \cdot \text{floor}\left(\frac{100\%}{Q_{cpu}}\right) \quad (4.2)$$

где  $T_{z\_max}$  - максимально допустимое время, отведенное на выполнение задачи;

$T_{z\_cpu}$  - время, затраченное кластером на решение задачи;

$Q_{cpu}$  - процент доступного ресурса процессора.

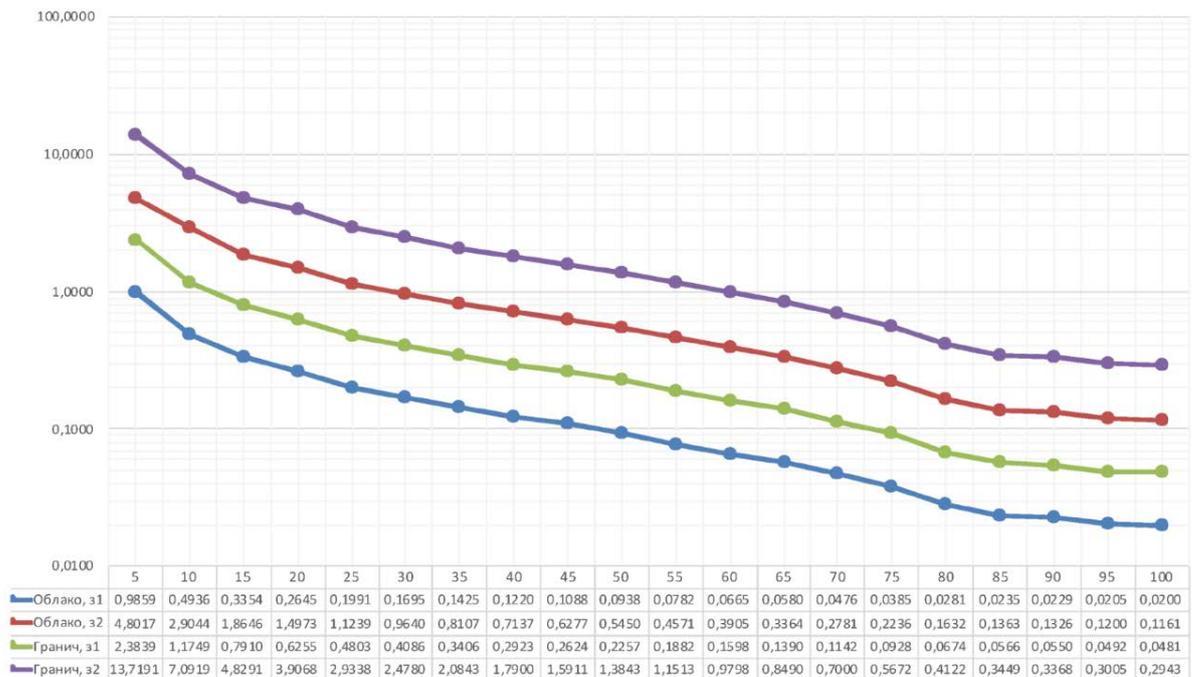


Рисунок 4.11 - График зависимости времени отклика сервиса от выделенного вычислительного ресурса CPU

Левая часть этой формулы показывает, сколько последовательных запусков задач можно выполнить, чтобы уложиться во время актуальности задачи, а правая часть показывает, сколько задач можно запустить параллельно.

Результаты расчетов для случая полностью свободного кластера представлены ниже в графическом виде (рисунок 12). Как видно из графика, наибольшее количество запросов может быть обработано при последовательном запуске контейнеров и выделении каждому из них максимальных ресурсов

процессора. Также имеются локальные максимумы, показывающие, что при выделении 50 % и 25 % CPU производительность всей системы будет выше, чем в случае предоставления для решения задачи 65 % и 45 % вычислительной мощности CPU соответственно. Во время работы службы автоматической миграции этот расчет будет выполняться «на лету» для текущих данных о доступных системных ресурсах.

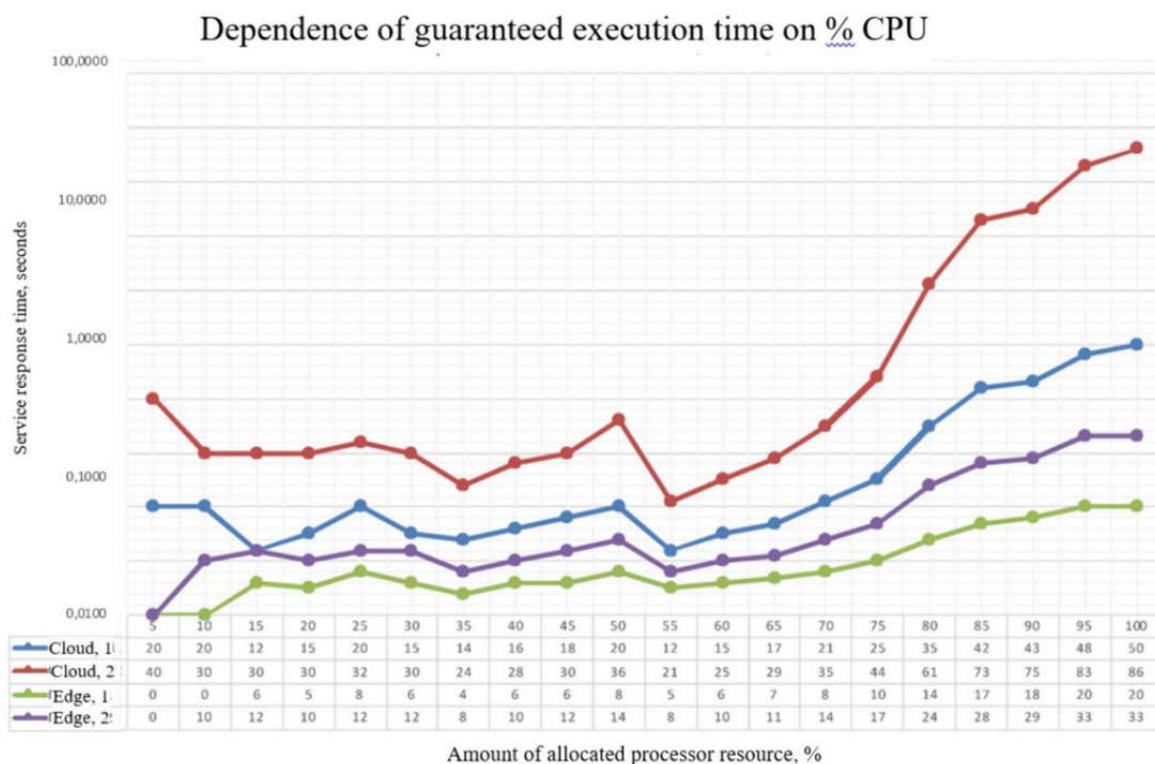


Рисунок 4.12 - График зависимости количества ответов, возвращаемых сервисом, от выделенного вычислительного ресурса CPU

Таким образом, можно рассчитать максимальную нагрузку, которую следует применить при тестировании производительности. Мы учитываем, что задачи имеют разное время актуальности, а на облачном кластере есть три вычислительных узла (по 2 процессорных ядра на каждый), предназначенных для запуска контейнеров. Raspberry Pi имеет 4 вычислительных ядра. Тогда за 10-секундный интервал на граничном кластере может быть решено 800 задач первого типа или 132 задачи второго типа. А на облачном кластере за 10 с можно решить

3000 задач первого типа или 516 второго типа. В тестовых условиях задачи будут комбинироваться, что позволит имитировать условия работы, близкие к реальным.

Метод автоматического размещения нод, используемый для сравнения с планировщиком, работает по простому алгоритму - размещаем здесь, если есть доступный ресурс, если нет - передаем запрос дальше или отбрасываем. Проверка скорости работы системы с использованием автоматической миграции по сравнению с планировщиком. Для решения задач будет использоваться шаблон оператора, который позволяет отслеживать загрузку узлов, перенаправление запросов осуществляется путем размещения проксирующего контейнера, который отправляет запрос в облачный кластер. При увеличении нагрузки будет автоматически осуществляться горизонтальное масштабирование количества нод на узле, количество определяется максимальной производительностью при параллельном выполнении запросов.

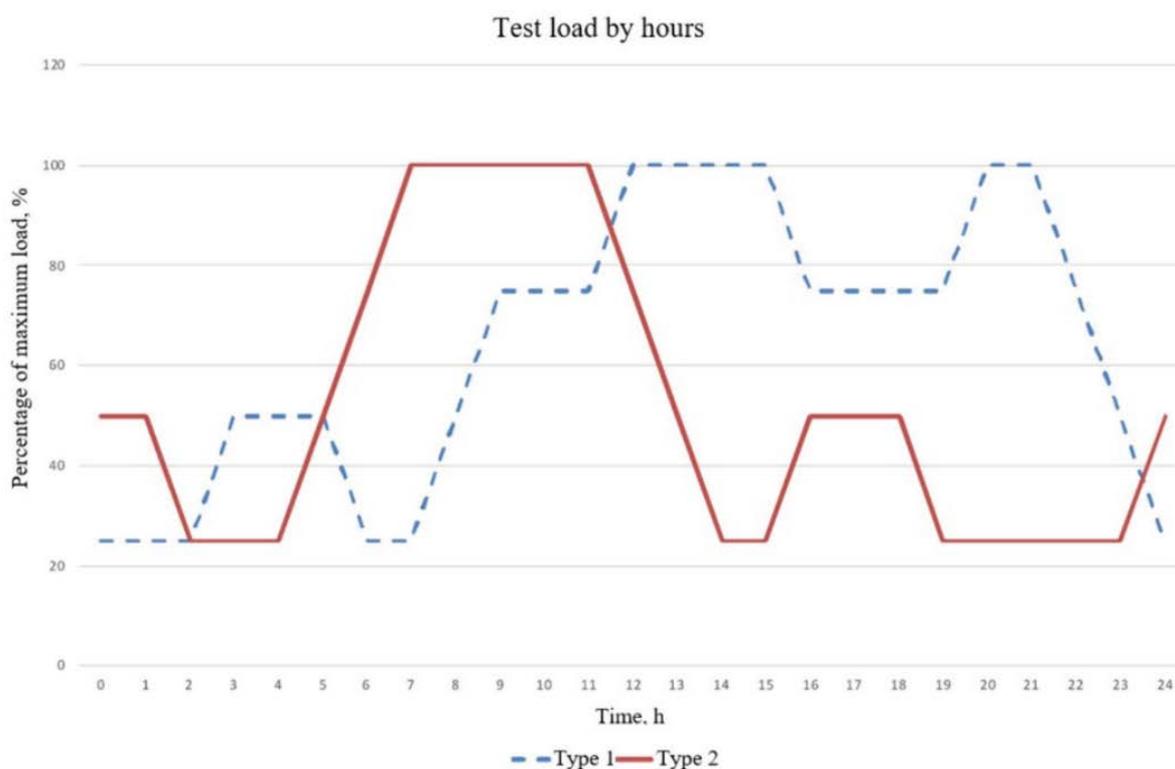


Рисунок 4.13 - График максимального уровня нагрузки на инфраструктуру по часам

Нагрузка, подаваемая на вход системы, является синтетической, генерируется перед испытанием и ограничивается условиями, представленными на графике (рисунок 4.13). Таким образом, в каждом типе теста (предлагаемый и сравниваемый алгоритм) на вход подается одна и та же нагрузка, превышающая физические возможности кластеров. Предлагаемый алгоритм использует возможности миграции нод при необходимости, учитывая существующую потребность в скорости обработки поступающей нагрузки. То есть миграция используется только в тех случаях, когда необходимо освободить имеющийся вычислительный ресурс для решения приоритетных задач. Следует отметить, что миграция менее приоритетных задач не приводит к многократному увеличению производительности системы. Однако за счет высвобождения ресурсов можно сократить количество потерянных задач в целом и, в частности, практически гарантировать выполнение критических задач и повысить стабильность обработки при высокой степени использования ресурсов.

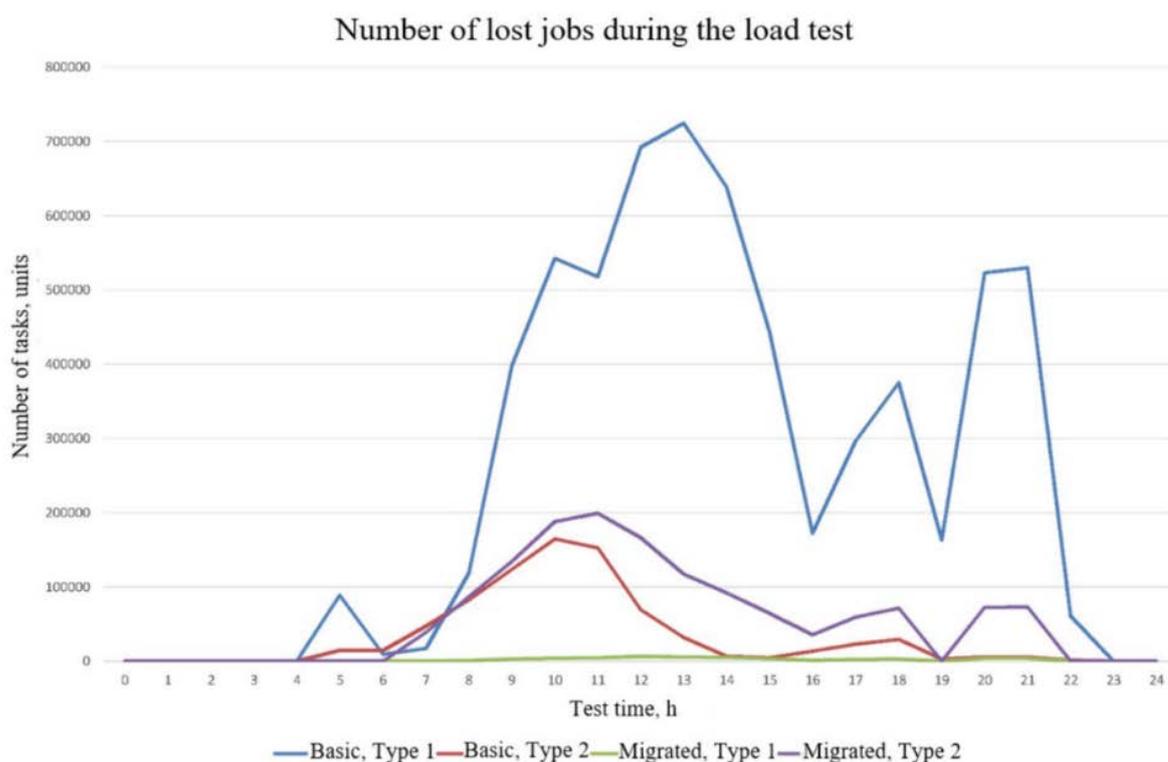


Рисунок 4.14 - График количества невыполненных задач по часам

#### 4.4. Вывод по главе 4

Изучены принципы работы системы Kubernetes с масштабированием и распределением задач, схемы и методы, используемые при развертывании и автоматизации инфраструктуры автоматизации. Kubernetes имеет развитую систему планирования и распределения задач по кластеру. Такие свойства объектов K8s, как Affinity и Anti-affinity, Taints и Tolera - обеспечивают широчайшие возможности для качественного размещения сервисов и приложений в кластере. Однако использовать эти свойства для межкластерного размещения капсул невозможно. Что касается решений, предлагаемых сообществом разработчиков, то предлагаются варианты с объединением нескольких кластеров в отказоустойчивую систему с выделением одного ведущего контроллера и распределением ролей резервных контроллеров между остальными. Таким образом, действительно можно использовать перечисленные свойства для балансировки нагрузки внутри объединенного кластера, но при разрыве сетевого соединения между связанными таким образом кластерами возникает задержка в обработке поступающих запросов, а время восстановления значительно превышает время миграции сервисов между кластерами.

Большой интерес вызвали технологии, связанные с миграцией, сокращением времени простоя сервисов и улучшением пользовательского опыта при их использовании. Например, отличным вариантом миграции является метод с созданием дополнительной ноды для переключения между исходным и целевым нодами. Благодаря тому, что целевая нода создается как копия рабочей ноды с работающим сервисом, удастся практически свести к нулю время простоя.

Эксперимент показал, что при наличии систем автоматического отслеживания необходимости проведения процедур миграции сервисов общая способность системы справляться с повышенными нагрузками возрастает по

сравнению с тем, если бы перенос и планирование размещения нод с сервисами осуществлялись вручную или полностью отсутствовали.

## **ГЛАВА 5. МЕТОД РЕАЛИЗАЦИИ ИММЕРСИВНЫХ ТЕХНОЛОГИЙ В СЕТЯХ СВЯЗИ ПЯТОГО И ШЕСТОГО ПОКОЛЕНИЙ, ПОЗВОЛЯЮЩИЙ В РЕЖИМЕ РЕАЛЬНОГО ВРЕМЕНИ ПЕРЕДАВАТЬ ЖЕСТОВУЮ РЕЧЬ**

Иммерсивные технологии стремятся переносить в пространстве наиболее полный объем человеческий чувств. Но что, если количеству чувств ограничено у самого человека? К сожалению, в мире множество людей с отклонениями по слуху, речи, осязания (тактильные ощущения). Внедряемые все более новые технологии стремятся к уменьшению влияния таких отклонений на полноценную жизнь человек [83].

Целью эксперимента является создание веб-приложения с микросервисной архитектурой, позволяющего управлять рукой робота и считывать показания “перчатки” в режиме реального времени для воспроизведения и считывания позиций руки соответствующей букве из алфавита языка жестов.

Рука должна позволить показывать, а перчатка считывать буквы на языке жестов, как в режиме передачи тактильных ощущений, так и в режиме аватара, повторяя движения при наступлении определенного события.

Алфавит языка жестов состоит из 32 букв - 23 из них символизируются статичными жестами, 9 - динамичными.

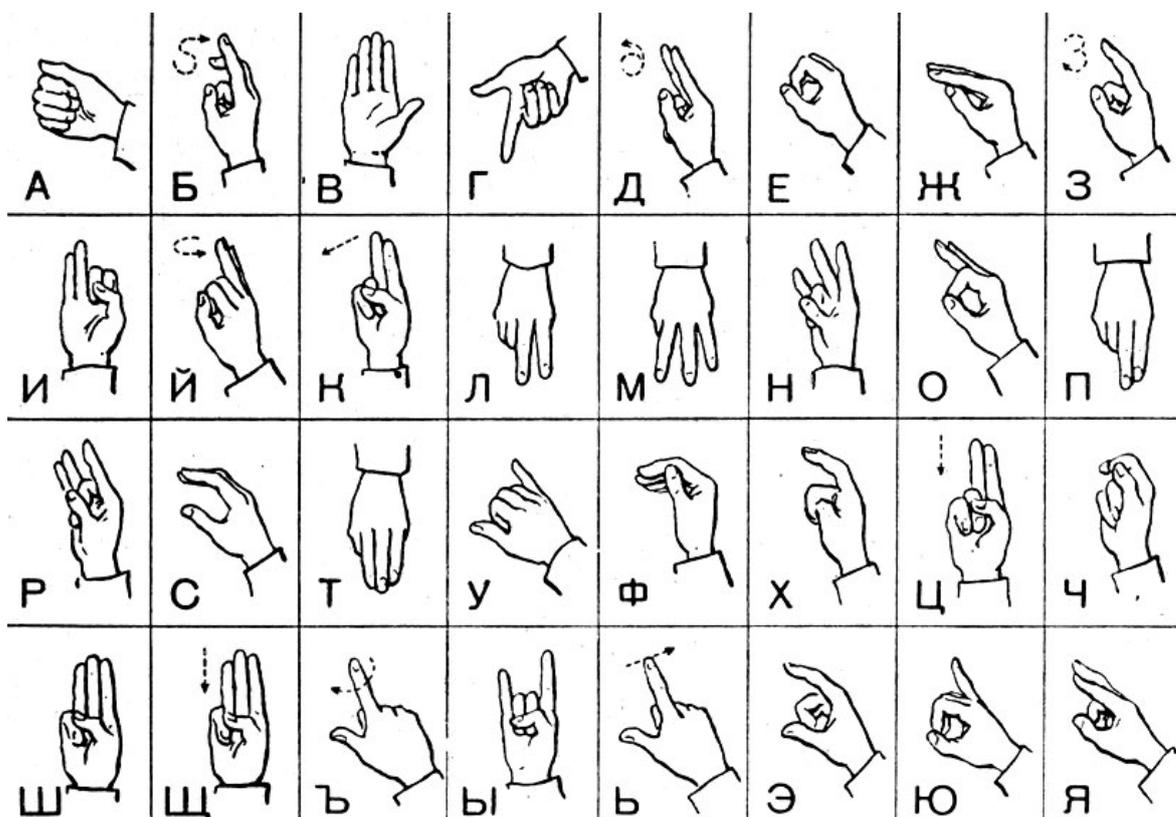


Рисунок 5.1 - Алфавит языка жестов

## 5.1 Описание стенда

### ESP-32

ESP-32 - современный микроконтроллер производства Espressif Systems, являющийся последователем всемирно популярной платы ESP8266 (Рисунок 5.2). Контроллер часто используется при создании устройств Интернета вещей или иных устройств, взаимодействующих по стандарту Wi-Fi, BLE, LoRa и др.



Рисунок 5.2 - Микроконтроллер ESP-32

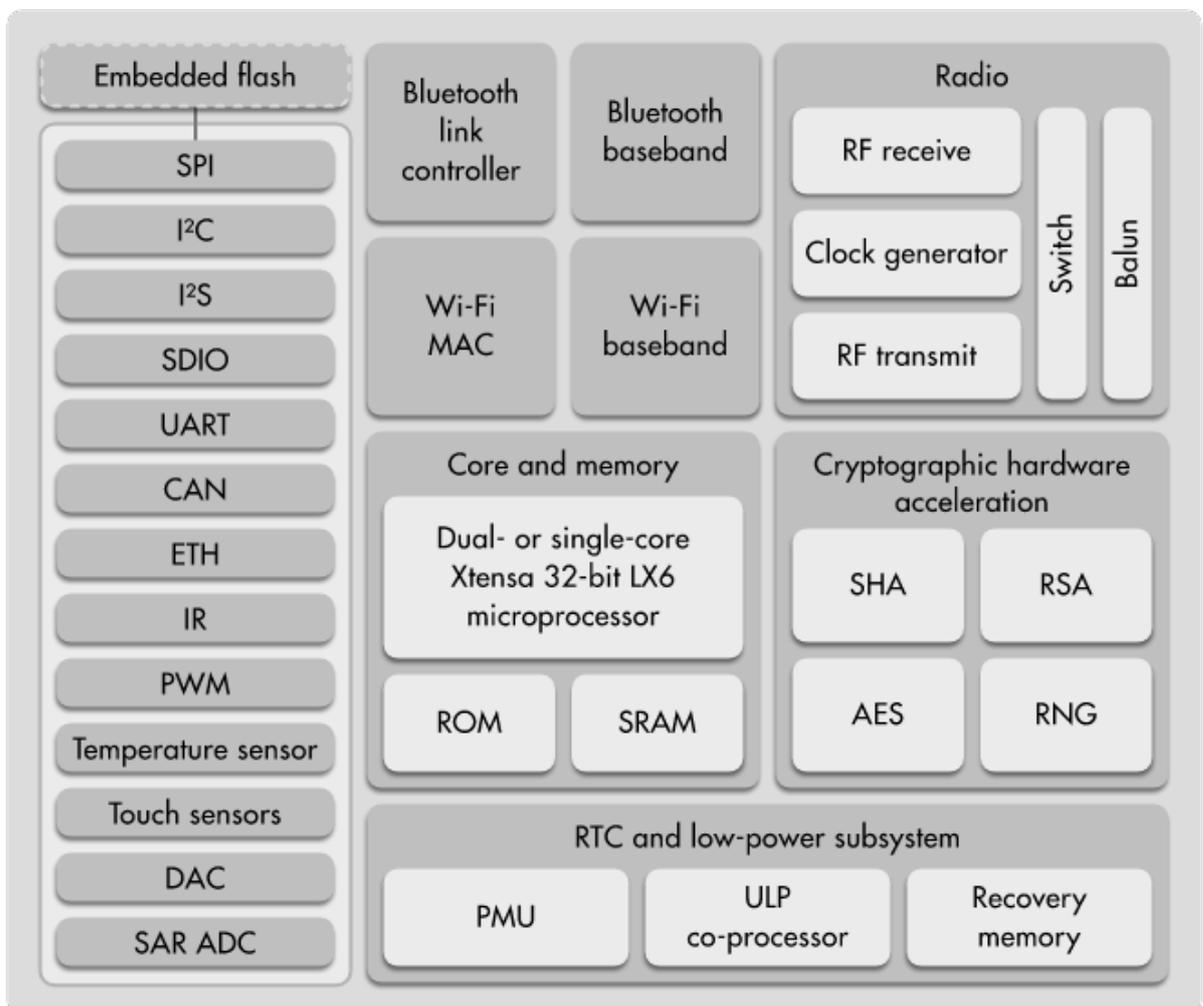


Рисунок 5.3 - Функциональная блок схема ESP-32

### Основные характеристики [84]:

- Процессор: 32-битный микропроцессор Tensilica Xtensa LX6 2 ядра
- Тактовая частота: до 240 МГц
- Производительность: до 600 DMIPS (сопроцессор с ультранизким энергопотреблением: позволяет выполнять преобразования АЦП, вычисления и пороговые значения уровней в состоянии глубокого сна).
- Беспроводное подключение:
  - Wi-Fi: 802.11 b/g/n/e/i (802.11n @ 2,4 ГГц до 150 Мбит/с)
  - Bluetooth: v4.2 BR/EDR и Bluetooth Low Energy (BLE).
- Память:
  - Внутренняя память ПЗУ: 448 Кбит
  - Для загрузки и основных функций: SRAM: 520 Кбит
    - Для данных и инструкций RTC fast SRAM: 8 Кбит
    - Для хранения данных и основного процессора во время загрузки RTC из режима глубокого сна RTC slow SRAM: 8 Кбит
    - Для доступа к сопроцессору в режиме глубокого сна eFuse: 1 кибит
  - Встроенная флэш-память:
    - Флэш-память, подключенная внутренне через IO16, IO17, SD\_CMD, SD\_CLK, SD\_DATA\_0 и SD\_DATA\_1 на ESP32-D2WD и ESP32-PICO-D4 0 Мб (микросхемы ESP32-D0WDQ6, ESP32-D0WD и ESP32-S0WD) 2 Мб (микросхема ESP32-D2WD) 4 Мб (SiP-модуль ESP32-PICO-D4)
  - Внешняя флэш-память и SRAM: ESP32 поддерживает до четырех внешних QSPI флэш-памятей и SRAM объемом 16 Мбайт с аппаратным шифрованием на основе AES для защиты программ

и данных разработчиков. ESP32 может обращаться к внешней флэш-памяти QSPI и SRAM через высокоскоростной кэш.

- До 16 Мбайт внешней флэш-памяти отображаются на кодовое пространство процессора, поддерживая 8-, 16- и 32-битный доступ. Поддерживается выполнение кода.
- До 8 Мбайт внешней флэш-памяти/SRAM отображаются на пространство данных процессора, поддерживая 8-, 16- и 32-битный доступ. Поддерживается чтение данных из флэш-памяти и SRAM. Запись данных поддерживается в SRAM.
- Периферийный ввод/вывод: Богатый периферийный интерфейс с DMA, включающий емкостное сенсорное управление, АЦП (аналого-цифровой преобразователь), ЦАП (цифро-аналоговый преобразователь), I<sup>2</sup>C (межсетевой интегральный интерфейс), UART (универсальный асинхронный приемник/передатчик), CAN 2.0 (сеть контроллеров), SPI (последовательный периферийный интерфейс), I<sup>2</sup>S (интегрированный межсетевой интерфейс), RMI (сокращенный медиа-независимый интерфейс), ШИМ (широтно-импульсная модуляция) и многое другое.
- Безопасность: Поддерживаются все функции безопасности стандарта IEEE 802.11, включая WPA, WPA/WPA2 и WAPI. Безопасная загрузка, флэш-шифрование, 1024-битное OTP, до 768-битного для клиентов. Криптографическое аппаратное ускорение: AES, SHA-2, RSA, криптография эллиптической кривой (ECC), генератор случайных чисел (RNG).

Контроллер обладает встроенными функциями управления ядрами процессора, что позволяет изолировать и существенно ускорять вычислительные процессы. Аналогично своему предшественнику (ESP8266) контроллер в первую очередь является модулем беспроводной связи, но также позволяет производить дополнительные вычисления, что делает его популярной, универсальной SoC-платформой для IoT. Исполняемый код представлен в Приложении А.

## **MPU-6050**

MPU-6050 - трехосевой гироскоп и акселерометр для отслеживания движения, разработанный с учетом требований к низкому энергопотреблению, низкой стоимости и высокой производительности смартфонов, планшетов и носимых датчиков.

В модуль встроено программное обеспечение InvenSense MotionFusion и калибровка во время выполнения, что позволяет производителям отказаться от дорогостоящего и сложного выбора, квалификации и интеграции дискретных устройств на уровне системы в продукты с поддержкой движения, гарантируя, что алгоритмы слияния датчиков и процедуры калибровки обеспечивают оптимальную производительность для потребителей.

Устройства MPU-6050 объединяют 3-осевой гироскоп и 3-осевой акселерометр на одной кремниевой матрице вместе со встроенным цифровым процессором движения (DMP), который обрабатывает сложные алгоритмы 6-осевого MotionFusion. Устройство может получать доступ к внешним магнитометрам или другим датчикам через вспомогательную ведущую шину I<sup>2</sup>C, что позволяет устройствам собирать полный набор данных датчиков без вмешательства системного процессора. Устройства предлагаются в корпусе QFN размером 4 мм x 4 мм x 0,9 мм.

Для точного отслеживания как быстрых, так и медленных движений, модуль оснащен программируемым пользователем гироскопом с полным диапазоном  $\pm 250$ ,  $\pm 500$ ,  $\pm 1000$  и  $\pm 2000$  °/сек (dps) и программируемым пользователем акселерометром с полным диапазоном  $\pm 2$ г,  $\pm 4$ г,  $\pm 8$ г и  $\pm 16$ г. Дополнительные функции включают встроенный датчик температуры и встроенный в микросхему осциллятор с отклонением  $\pm 1\%$  в диапазоне рабочих температур.

В целях упрощения взаимодействия с микроконтроллером, в проекте используется отладочная плата NodeMCU.

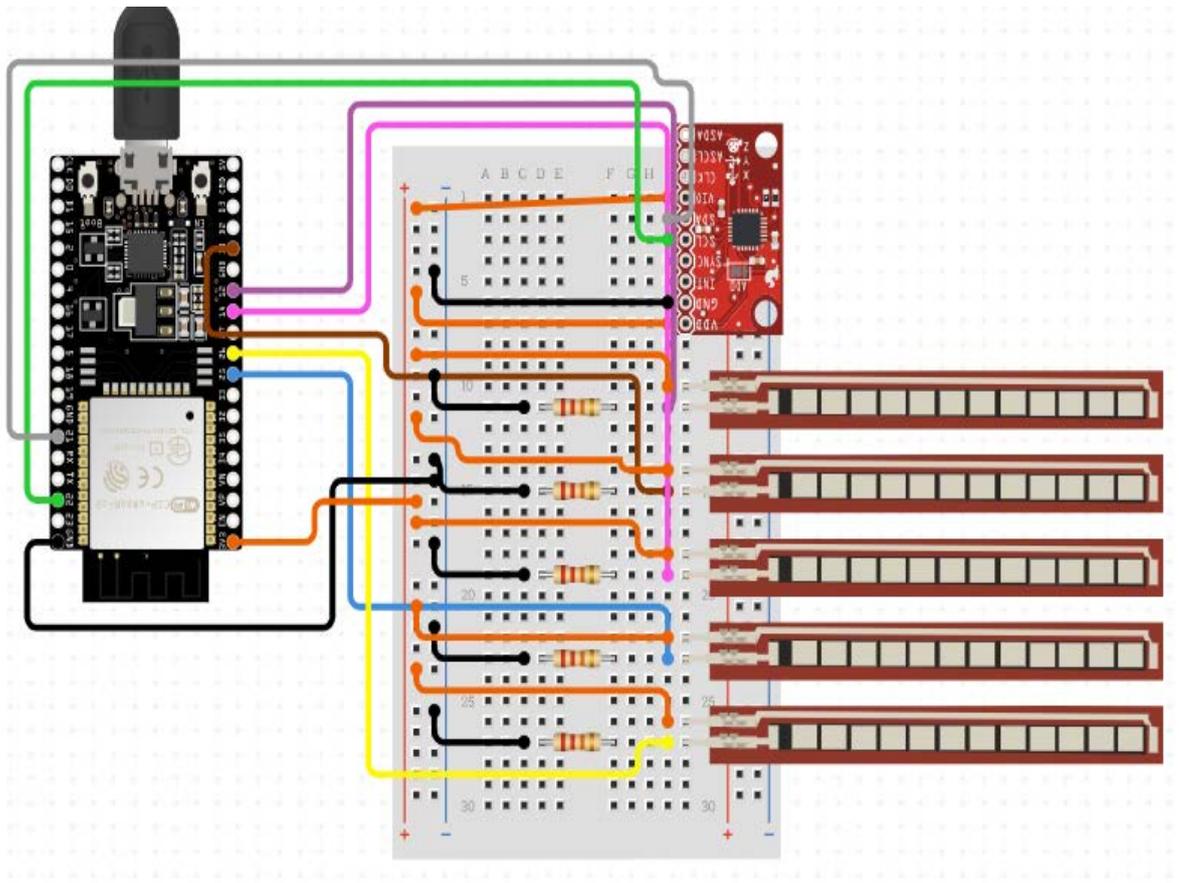


Рисунок 5.4 - Схема подключения



Рисунок 5.5 - Стенд с робой перчаткой

## **Processing**

Processing - открытый язык программирования, основанный на Java. которые хотят программировать изображения, анимацию и интерфейсы. Программное обеспечение Processing — это бесплатное, открытое, кроссплатформенное ПО. Исходный архив включает в себя java-машину, сам интерпретатор, мини-IDE, и несколько десятков примеров. ПО включает инструменты для построения графических примитивов, 3D-объектов, работу со светом, текстом, инструментами трансформации., позволяет импортировать и экспортировать файлы аудио/видео/звуковых форматов, обрабатывать события мыши/клавиатуры, работать со сторонними библиотеками (OpenGL, PDF, DXF).

В рамках ВКР Processing применяется в целях визуализации движений эскиза механизированной руки-робота на языке программирования Java. Разработанная программа является универсальной и позволяет проводить любые эксперименты с использованием трехмерной визуализации, в том числе может быть использована для обучения студентов работе с устройствами IoT и создания новых кейсов применения робо-перчатки. Код проекта представлен в Приложении Б.

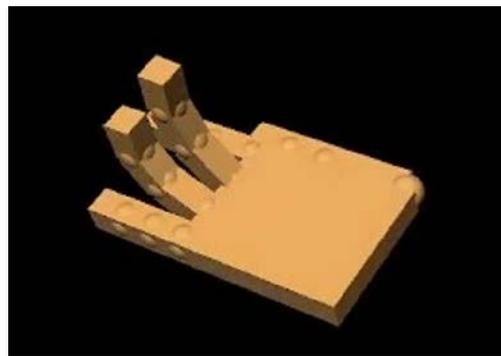
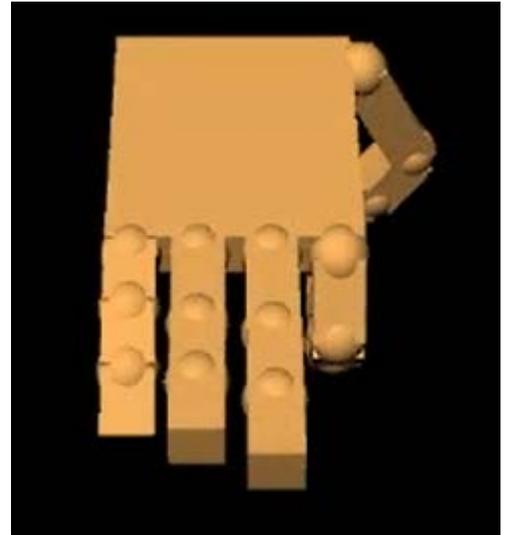
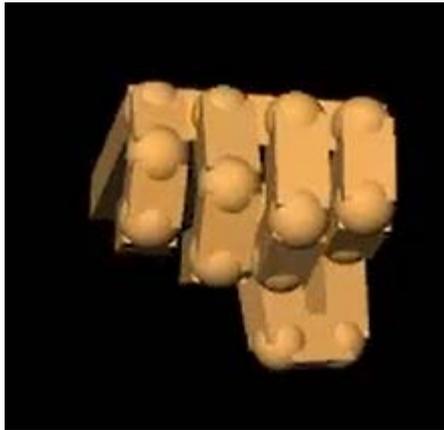


Рисунок 5.6 - Processing. 3D - визуализация

### **MQTT Брокер**

EMQX — это брокер сообщений IoT MQTT с открытым исходным кодом, основанный на платформе Erlang/OTP (платформа для разработки в режиме Soft-Realtime, с низкой задержкой и распределением).

Брокер предназначен для массового доступа клиентов и реализует быструю и маршрутизацию сообщений между массивными физическими сетевыми устройствами с минимальными задержками. Это одна из самых масштабируемых и надежных платформ обмена сообщениями MQTT для подключения, перемещения и обработки данных в критически важных для бизнеса сценариях эпохи IoT. Позволяет подключать практически любое устройство с помощью открытых стандартных IoT-протоколов MQTT, CoAP и LwM2M. Легко масштабируется до

десятков миллионов одновременных MQTT-соединений с помощью кластера EMQX Enterprise [85].

## 5.2 Противодействие задержкам

### 5.2.1 Эксперимент миграции функциональных элементов сети

N. Kato с соавторами в журнале IEEE предлагает актуальные входные и выходные характеристики гетерогенных сетевых трафиков и систем контролируемых глубоких нейронных сетей. В этой работе авторы описывают, как работает предложенная система и как это отличается от традиционных нейронных сетей. Кроме того, предварительные результаты демонстрируют многообещающие показатели предложенной системы глубокого машинного обучения, по сравнению с эталонной стратегией направления (Open Shortest Path First), с точки зрения передачи служебного сигнала, пропускной способности и задержки [86]. В [87], авторами была предложена практическая платформа и процессы прогнозирования трафиков, на основе больших массивов данных, машинного обучения и сетевых Key Performance Indicators (KPI), достаточно гибких, для обеспечения точной оценки параметров потоков данных различных типов сот (GSM, 3G, 4G) для долгосрочного и краткосрочного прогнозирования. Исполнение предложенной модели было оценено применимо к реальному набору данных, снятых с KPI – более чем 6 000 сот реальной сети в течение 2016 и 2017 годов. В то время как, авторами статьи “Ant colony optimization algorithm with Internet of Vehicles for intelligent traffic control system” была предложена система Internet of Vehicle (IoV)[88], где карты города сегментированы в небольшое количество уникальных карт. Они применяют алгоритм Колонии муравьев к каждой карте, чтобы найти оптимальный маршрут. Кроме того, предложена функция расчета интенсивности потоков данных для

модели интенсивных потоков данных, основана на принципах нечеткой логики. Предложенный метод выбора маршрута, основанный на IoV был сравнен с существующими алгоритмами выбора кратчайшего пути такими, как алгоритм Дейкстра, алгоритм Краскала и алгоритм Прима. Результаты эксперимента показали хорошее исполнение предложенного метода выбора маршрута, основанного на IoV [89].

Для реализации предложенной методологии динамического распределения сервиса мы собрали на базе лаборатории программно-конфигурируемых сетей стенд 5G NSA показанный на рисунке 4.6. Каждый элемент сети был запакован в отдельный контейнер со своими зависимостями, и запущен в одной виртуальной плоскости. В качестве RAN мы имплементировали работающий на FPGA. Со следующими характеристиками:

- Частотный диапазон: 10 МГц до 6 ГГц;
- Мгновенная пропускная способность: до 160 МГц;
- Мощность: 50-100 мВт (17-20 dBm)
- Диапазон шума: 0-31.5 dBm
- Разрядность ЦАП: 16 бит;
- Количество каналов: 2

Таблица 5.1 - Перечень элементов тестовой сети

HSS+UDM	Home Subscriber Server + Unified Data Management
PCF+PCRF	Policy Control Function + Policy and Charging Rules Function
PGW-C + SMF	Packet Data Network Gateway - Control + Session Management Function
UPF + PGW-U	User Plane Function + Packet Data Network Gateway - User
MME	Mobility Management Entity
AMF	Access and Mobility Management Function

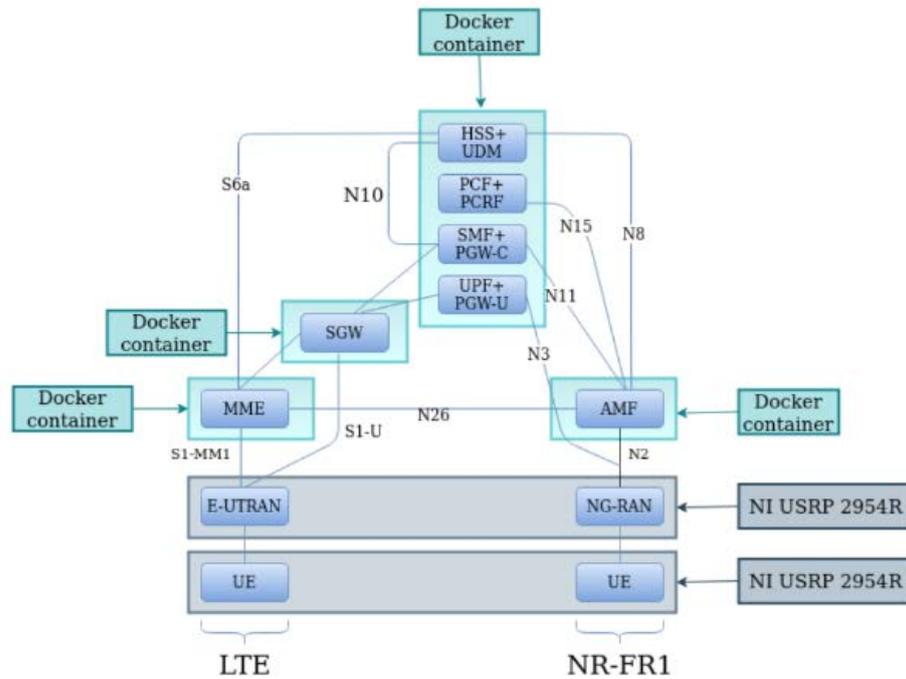


Рисунок 5.7 - Схема взаимодействия тестируемой сети

Поставленной задачей стало распределение элементов сети NSA таким образом, чтобы уменьшить ресурсные расходы при сохранении необходимого качества QoS. Для этого предложен механизм бесшовного переноса сервиса на другие nodes. Sequence diagram представлен в рисунке 5.7. В нём описано, как при помощи SDN происходит перенос сервиса с наименьшими потерями при активных пользователях. Для данной процедуры вводится дополнительный элемент, который будет отвечать за выделения и разворачивания ресурса в nodes. Этот элемент имеет доступ к контроллеру сети и не посредственно к сервисам EPC методом REST API.

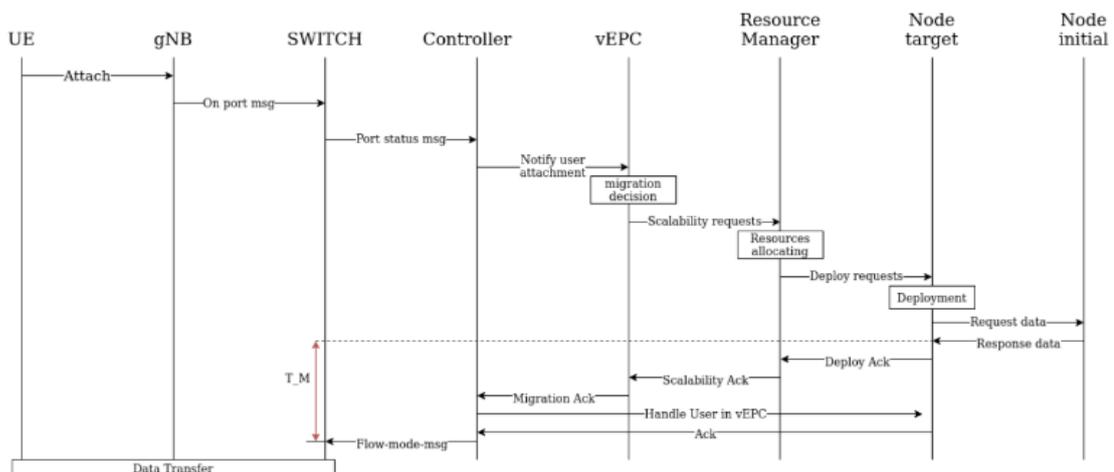


Рисунок 5.8 - Диаграмма работы алгоритма

Такая система также необходима для гибкого масштабирования сети и быстрой локализации проблемных областей сети. В эксперименте было использовано разное количество сессии активных пользователей, чтобы изучить возможные влияния на процесс миграции. Так как активные сессии имеют непосредственное подключение к HSS+UDM. Именно эти элементы мы подвергали к процессу миграции, чтобы получить самый худший вариант событий.

## Результаты

Как говорили ранее, к процессу миграции мы подвергли сам EPC в составе HSS, UDM, PCF, PCRF, PGW-C, SMF, UPF, PGW-U одним контейнером. И наблюдали за длительностью миграции в зависимости от сессий и потерей пакетов. Длительность миграции считать время  $T_M$  показанный на рисунке 5.7. Далее получили следующие результаты. На рисунке 5.8 показано, что при 640 активных сессий миграция происходит за 4 минуты. А при малых количествах сессии за меньше одной минуты. И видно, что далее происходит резкое увеличение времени миграции. Это связано с тем, что система для сохранения сети пытается по блочно переносить сессию из-за этого такие показатели длительности миграции.

В рисунке 5.9 показано потери пакетов при миграции с тем же количеством сессий. В итоге получаем, что при большем переносе пользователей увеличивается время миграции, но уменьшается количество потерянных пакетов на пользователя.

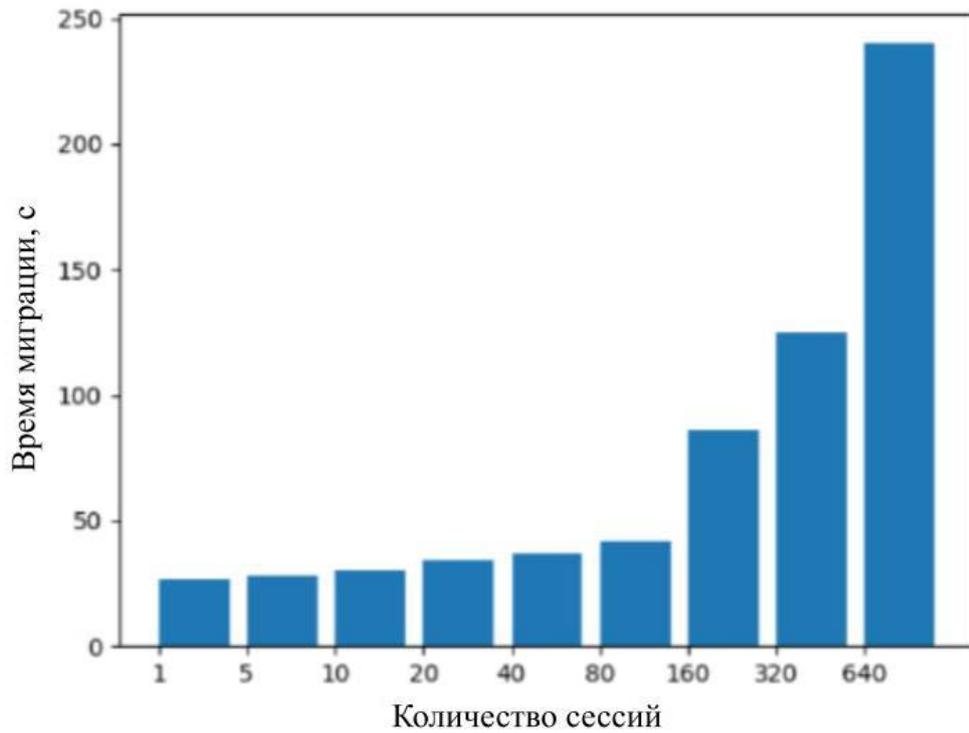


Рисунок 5.9 - Зависимость времени миграции от количества сессий

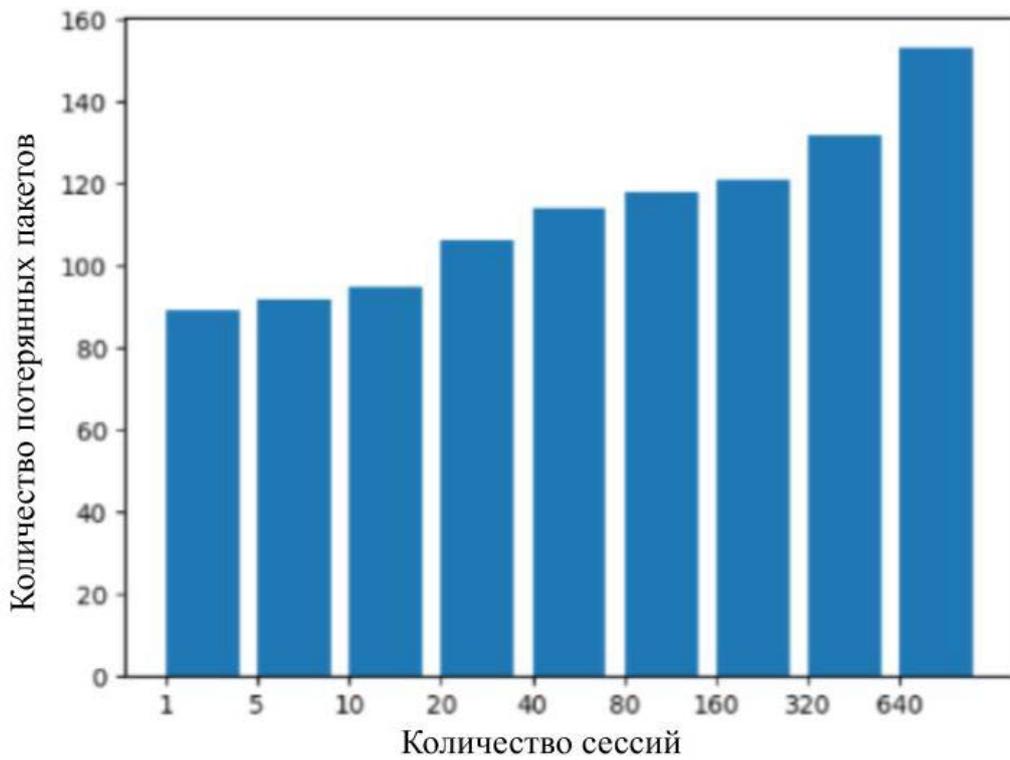


Рисунок 5.10 - Зависимость потери пакетов от количества сессий

### 5.2.2 Эксперимент миграции стенда

В целях проведения анализа уровня задержек, а также исследования методов их сокращения предлагается рассмотреть IoT систему взаимодействия робо-перчатки и ее виртуального аватар, которые описаны в пункте 5.1. В ходе эксперимента предлагается произвести сравнительный анализ трех схем взаимодействия (Рисунок 5.11) [90]:

1. Все составные части системы располагаются в локальной сети.
2. Робо-перчатка и приложение 3D визуализации располагается локально, а MQTT брокер на удаленной машине в сети Интернет. Взаимодействие производится по Wi-Fi.
3. Робо-перчатка находится в локальной сети, приложение и брокер подвергаются миграции, с использованием Docker контейнеров.

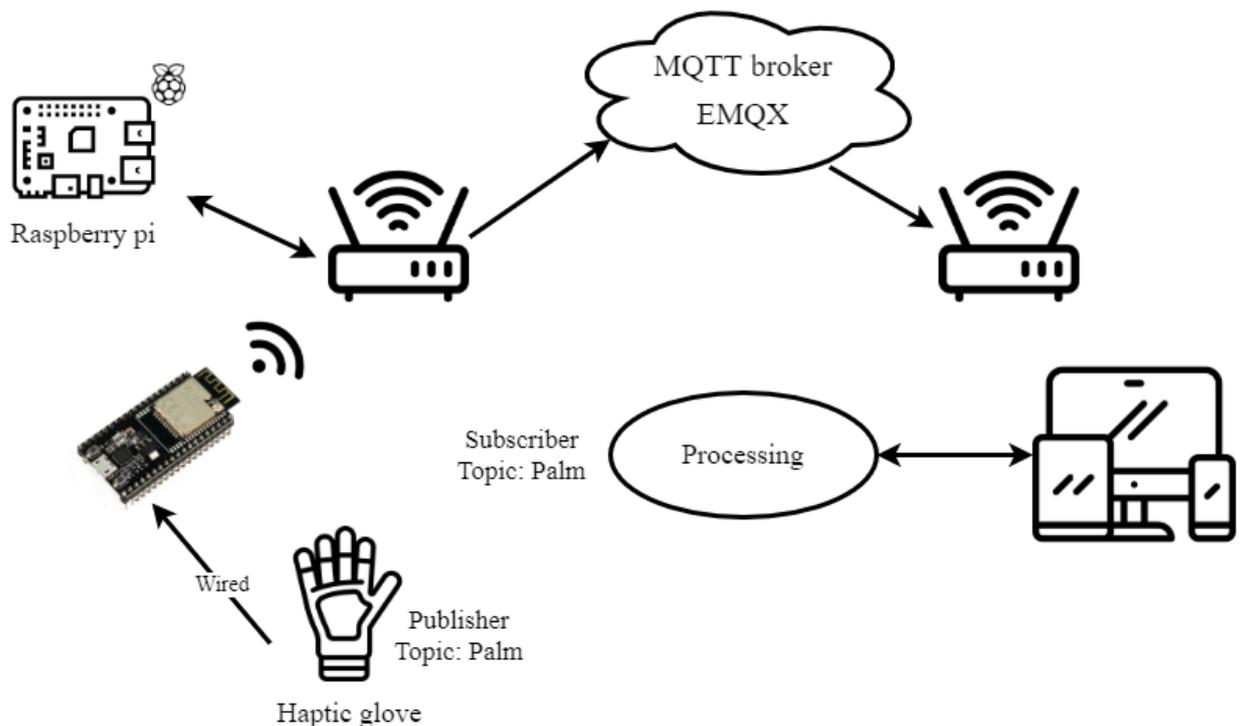


Рисунок 5.11 - Общая схема взаимодействия устройств

### Вариант № 1

Рассмотрим использование стенда в локальной сети. Для придания большей реалистичности эксперимент проводился в домашней сети Wi-Fi, с подключенными клиентами, не участвующими в эксперименте.

Робо-перчатка подключена к ESP-32, которая в свою очередь осуществляется взаимодействие с MQTT брокером на Wi-Fi. Брокер и приложение, визуализирующее руку-аватара, располагают на одной локальной машине.

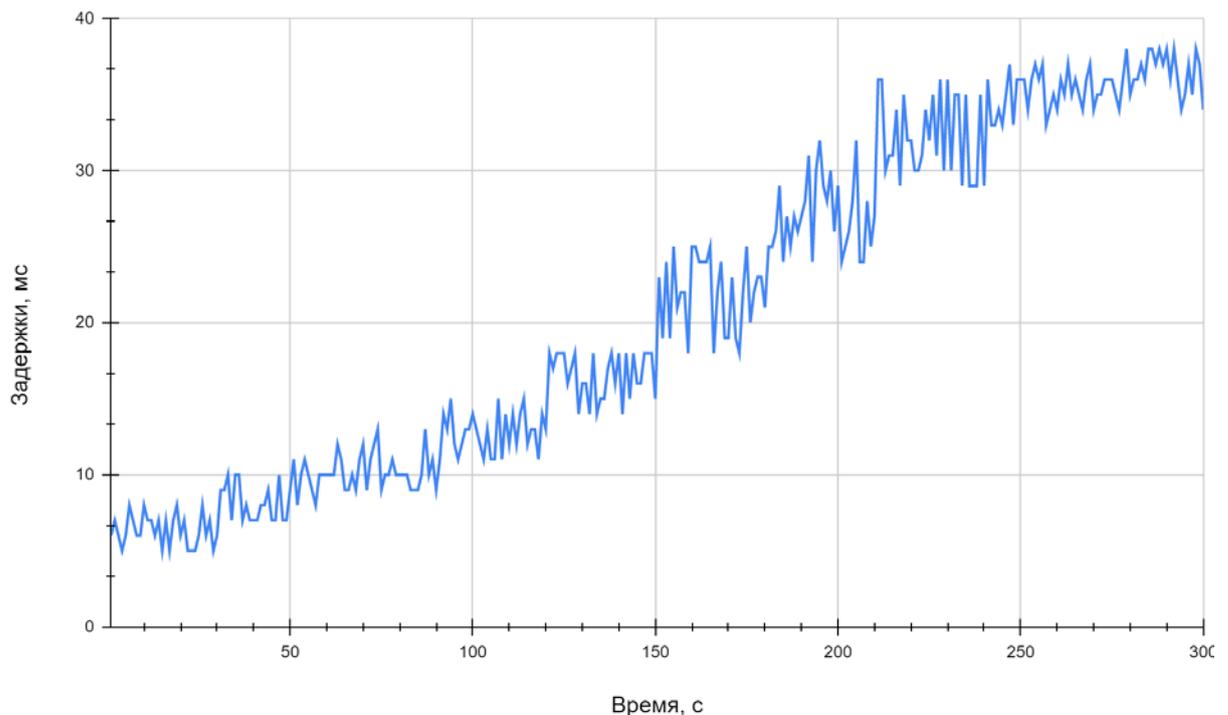


Рисунок 5.12 - График распределения задержек для варианта №1

Таким образом, наблюдается негативное влияние возрастающей нагрузки на MQTT брокер по мере его наполнения (Рисунок 5.12). В среднем “пороговое” значение количества пакетов достигает 300 000 шт. После перезагрузки сервера (сброса накопленного объема данных) качество работы восстанавливается.

## Вариант № 2

Эксперимент проводится при удаленном расположении MQTT Брокера в сети Интернет. Брокер развернут на виртуальных облачных сервера (VPS) с использованием контейнера Docker. Остальные устройства стенда располагаются аналогично варианту №1.

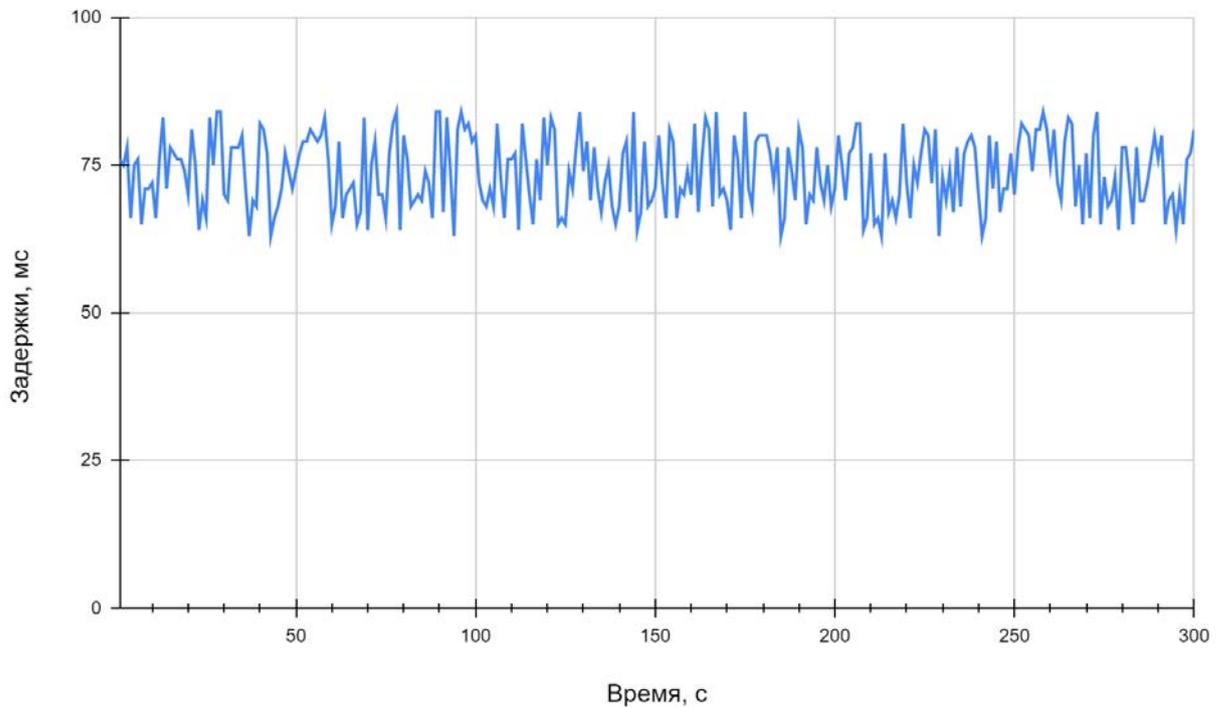


Рисунок 5.13 - График распределения задержек для варианта №2

Учитывая задержки прохождения ICMP пакетов до описанного сервера в размере 50-70 мс, можно сделать вывод о стабилизации работы стенда, за счет разворачивания брокера на отдельной, более производительной ЭВМ (Рисунок 5.13).

Таким образом, наиболее существенное влияние оказывают задержки на пути следования пакетов по сети Интернет. В целях их сокращения предлагается мигрировать брокер ближе к абоненту.

### Вариант № 3

Рассмотрим способ снижения уровня задержек, на основании данных, полученных в ходе предыдущих экспериментов. Предлагается расположить MQTT брокер в Docker контейнер и произвести его миграцию на оборудования расположенное рядом с конечным абонентом. Для проведения эксперимента мигрируем брокер, развернутый ранее на удаленных VPS серверах, на виртуальную машину, находящуюся на расстоянии 2 хопов от конечного оборудования. Данный метод предполагает использование только встроенного функционала Docker'a. Управление будет осуществляться с помощью Docker CLI, удаленным облачным хранилищем выступает Docker Hub, за организацию сети, скачивание образа и развертывание контейнера отвечает Docker daemon. Миграция была произведена 10 раз для установления времени переноса сервиса и его стабильности. В среднем время развертывания сервиса составило 1,56 при условии отсутствии предзагруженного образа [91].

Далее проведен анализ задержек при работе стенда в новой конфигурации. Стоит отметить, что производительность оборудования существенно увеличилась по сравнению с вариантом №1 также, как и пропускная способность брокера (Рисунок 5.14).

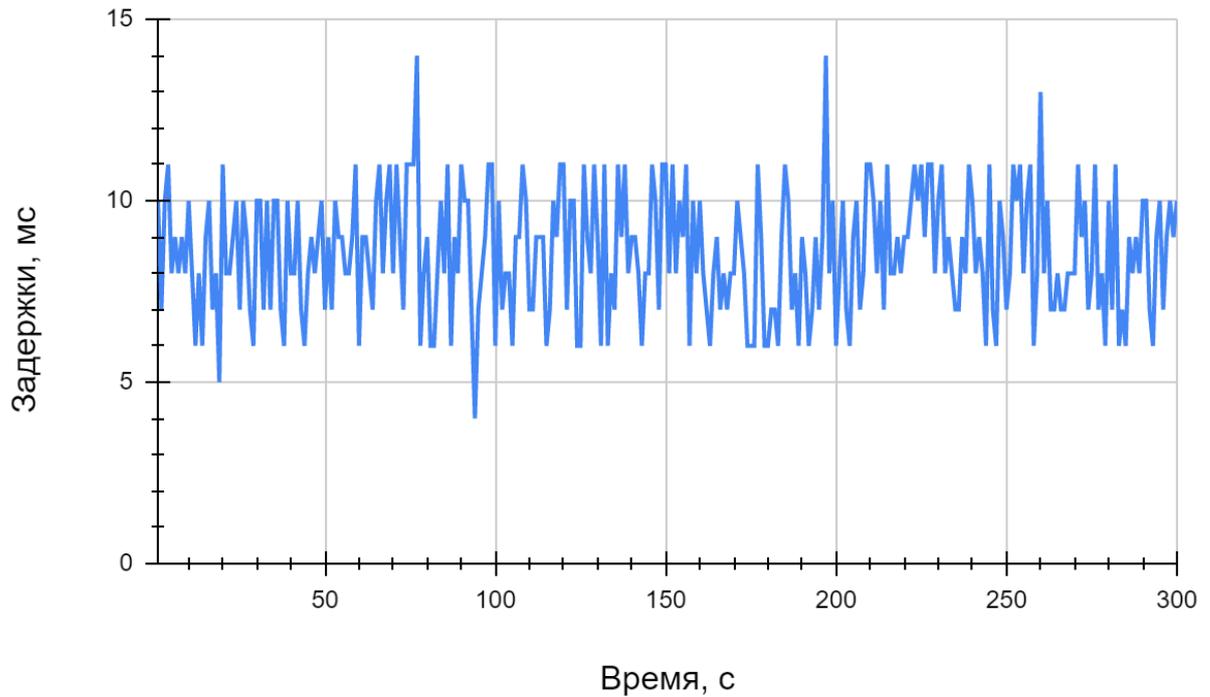


Рисунок 5.14 - График распределения задержек для варианта №3

Задержки прохождения ICMP пакетов между конечным оборудованием станда сократилась до 2-4 мс. Таким образом общая работа станда укладывается в пределах 10-11 мс задержки, что не позволяет передать тактильные ощущения, но вполне достаточно для выполнения поставленных задач. В сравнение с вариантом №1 станда начал функционировать в стабильном состоянии. Дополнительно был произведен эксперимент по работе сервиса в течение 2 часов, качество работы станда осталось неизменным.

## ЗАКЛЮЧЕНИЕ

В диссертационной работе получены следующие основные результаты:

1. Предложенный метод оптимизации числа контроллеров в мультиконтроллерных сетях SDN демонстрирует значительные преимущества по сравнению с традиционным методом. Он снижает среднее число контроллеров на 46% за счет динамического управления, что уменьшает энергопотребление и затраты на развертывание, а также повышает надежность сети.

2. Предложенный метод на 53% повышает эффективность использования контроллеров, за счет динамического распределения нагрузки между ними в мультиконтроллерных сетях SDN, тем самым значительно сокращая расходы на развертывание сети.

3. Предложенный алгоритм кластеризации, являющийся модификацией алгоритма FOREL, обеспечивает выбор позиций маршрутизаторов для обслуживания пользователей и организации связной mesh сети и связи этой сети с базовой станцией подвижной связи.

4. Модификация алгоритма кластеризации, обеспечивающая перераспределение элементов кластеров, позволяет повысить качество кластеризации путем выравнивания количества элементов выделяемых кластеров.

5. Метод маршрутизации трафика в mesh сети обеспечивает выбор маршрутов на основе минимизации интегрального показателя качества маршрута, учитывающего достижимую скорость передачи данных, величину нагрузки и вероятность потерь.

6. Предложенный метод распределения маршрутизаторов в рое БПЛА или на единичных или привязных БПЛА эффективен в случае, когда размер зоны связи маршрутизатора меньше размера зоны обслуживания.

Представленные в диссертации модели и методы построения интеллектуального ядра сети связи шестого позволяют повысить эффективность использования существующих подходов к построению сетей связи, также

подготовленная модельно-методическая база позволяет проводить дальнейшие исследования в области передачи услуг телеприсутствия на расстоянии. При дальнейшей разработке темы исследования рекомендуется изучить особенности трафика генерируемыми костюмами телеприсутствия, а также рассмотреть возможность применения искусственного интеллекта для сокращения задержек.

**СПИСОК ЛИТЕРАТУРЫ**

1. Кузнецов К.А. Метавселенные и методы их реализации в сетях связи 2030 / Кучерявый А.Е., Мутханна А.С.А. // 79-я Научно-техническая конференция Санкт-Петербургского НТО РЭС им. А.С. Попова, посвященная Дню радио: сборник докладов / СПбГЭТУ «ЛЭТИ».2024. С. 200-201
2. Леонова М.Д. Анализ технологии для совместной работы приложений и сервисов метавселенных / Мутханна А.С.А., Кузнецов К.А. // 79-я Научно-техническая конференция Санкт-Петербургского НТО РЭС им. А.С. Попова, посвященная Дню радио: сборник докладов / СПбГЭТУ «ЛЭТИ».2024. С. 216-219
3. Y. Wang et al., “A Survey on Metaverse: Fundamentals, Security, and Privacy,” IEEE Commun. Surveys & Tutorials, DOI: 10.1109/COMST.2022.3202047, Sept. 2022.
4. M. Xu et al., “Wireless Edge-Empowered Metaverse: A Learning-Based Incentive Mechanism for Virtual Reality,” ICC 2022 — IEEE Int’l. Conf. Commun., Seoul, Korea, pp. 5220–25, DOI: 10.1109/ICC45855.2022.9838736, May 2022.
5. R. Cheng et al., “Will Metaverse Be Nextg Internet? Vision, Hype, and Reality,” arXiv preprint arXiv:2201.12894, Jan. 2022.
6. Y. J. Moon et al., “OFDM-based 25Gbps Wireless Backhaul System for 5G Convergence Service,” 2020 International Conference on Information Networking (ICOIN), Barcelona, Spain, 2020, pp. 814-817, 10.1109/ICOIN48656.2020.9016621.
7. M. J. Piran and D. Y. Suh, “Learning-Driven Wireless Communications, towards 6G,” 2019 International Conference on Computing, Electronics & Communications Engineering (iCCECE), 2019, pp. 219-224, 10.1109/iCCECE46942.2019.8941882.
8. K. Popoola, D. Grace and T. Clarke, “Capacity and Coverage Analysis of High-Altitude Platform (HAP) Antenna Arrays for Rural Vehicular Broadband

- Services,” 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), Antwerp, Belgium, 2020, pp. 1-5, 10.1109/VTC2020-Spring48590.2020.9129557.
9. David, K., Berndt, H., 2018. 6G vision and requirements: Is there any need for beyond 5G? *IEEE Veh. Technol. Mag.* 13 (3), 72–80. <https://doi.org/10.1109/MVT.2018.2848498>.
  10. Li, F., Lam, K., Liu, X., Wang, L., 2020. Resource Allocation in Satellite-Based Internet of Things Using Pattern Search Method. *IEEE Access* 8, 110908–110914. <https://doi.org/10.1109/ACCESS.2020.3002834>.
  11. Chen, S., Liang, Y., Sun, S., Kang, S., Cheng, W., Peng, M., 2020. Vision, Requirements, and Technology Trend of 6G: How to Tackle the Challenges of System Coverage, Capacity, User Data-Rate and Movement Speed. *IEEE Wireless Communications* 27 (2), 218–228 <https://doi.org/10.1109/MWC.001.1900333>.
  12. Z. Zhang et al., “User Activity Detection and Channel Estimation for Grant-Free Random Access in LEO Satellite-Enabled Internet-of-Things,” in *IEEE Internet of Things Journal*, 10.1109/JIOT.2020.2997336
  13. Nawaz, S.J., Sharma, S.K., Wyne, S., Patwary, M.N., Asaduzzaman, M., 2019. Quantum Machine Learning for 6G Communication Networks: State-of-the-Art and Vision for the Future. *IEEE Access* 7, 46317–46350. <https://doi.org/10.1109/ACCESS.2019.2909490>
  14. M. Fadlullah and N. Kato, “HCP: Heterogeneous Computing Platform for Federated Learning Based Collaborative Content Caching Towards 6G Networks,” in *IEEE Transactions on Emerging Topics in Computing*, 10.1109/TETC.2020.2986238
  15. Kato, N. et al., 2019. Optimizing Space-Air-Ground Integrated Networks by Artificial Intelligence. *IEEE Wireless Commun.* 26 (4), 140–147. <https://doi.org/10.1109/MWC.2018.1800365>.
  16. “White Paper: Key Drivers and Research Challenges for 6G Ubiquitous Wireless Intelligence.” <http://jultika.oulu.fi/files/isbn9789526223544.pdf>, accessed in Dec. 2019.

17. T. R. Gadekallu et al., “Blockchain for the Metaverse: A Review,” arXiv preprint arXiv:2203.09738, Mar. 2022.
18. S. K. Jagatheesaperumal et al., “Advancing Education Through Extended Reality and Internet of Everything Enabled metaverses: Applications, Challenges, and Open Issues,” arXiv preprint arXiv:2207.01512, June 2022.
19. P. Bhattacharya et al., “Coalition of 6G and Blockchain in AR/VR Space: Challenges and Future Directions,” *IEEE Access*, vol. 9, Dec. 2021, pp. 168,455–84.
20. T. Huynh-The et al., “Artificial Intelligence for the metaverse: A Survey,” *Engineering Applications of Artificial Intelligence*, vol. 117, Jan. 2023.
21. R. Chengoden et al., “Metaverse for Healthcare: A Survey on Potential Applications, Challenges and Future Directions,” arXiv preprint arXiv:2209.04160, Sept. 2022.
22. W. Saad, M. Bennis, and Mingzhe Chen, “A Vision of 6G Wireless Systems: Applications, Trends, Technologies, and Open Research Problems,” *IEEE Network*, vol. 34, no. 3, June 2020, pp. 134–42.
23. A. I. Salameh and M. El Tarhuni, “From 5G to 6G — Challenges, Technologies, and Applications,” *Future Internet*, vol. 14, no. 4, Apr. 2022, pp. 117.
24. Y. Wu, G. Lin, and J. Ge, “Knowledge-Powered Explainable Artificial Intelligence for Network Automation toward 6G,” *IEEE Network*, vol 36, no. 3, June 2022, pp. 16–23.
25. Z. Allam et al., “The Metaverse as A Virtual Form of Smart Cities: Opportunities and Challenges for Environmental, Economic, and Social Sustainability in Urban Futures” *Smart Cities*, vol. 5, no. 3, July2022, pp. 771–801.
26. El Marai, Oussama and Taleb, Tarik and Song, JaeSeung, “AR-based Remote Command and Control Service: Self-driving Vehicles Use Case” *IEEE Network*, DOI: 10.1109/MNET.119.2200058, Oct 2022.
27. El Marai, Oussama, and Tarik Taleb, “Smooth and Low Latency Video Streaming for Autonomous Cars During Handover,” *IEEE Network*, vol. 34, no.6, Dec. 2020, pp. 302–09.

28. D. Yang et al., “Expert Consensus on the Metaverse in Medicine,” *Clinical eHealth*, vol. 5, Dec 2022, pp. 1–9.
29. ETSI GS NFV 002 V1.2.1 (2014-12). Network Functions Virtualisation (NFV); Architectural Framework.
30. Faisal, V.A. “NFV MANO-Management & Orchestration”, *TelcoCloud Bridge*, 2020.
31. Атея, А.А. Интеллектуальное ядро для сетей связи 5G и тактильного интернета на базе программно-конфигурируемых сетей / А.А. Атея, А.С. Мутханна, А.Е. Кучерявый // *Электросвязь*. – 2019. – № 3. – С. 34–40.
32. Sahoo, K.S. Metaheuristic solutions for solving controller placement problem in SDN-based WAN architecture / K.S. Sahoo, A. Sarkar, S.K. Mishra et al. // *8th International Conference on Data Communication Networking (DCNET)*, Madrid, Spain, 2017. – P. 15-23.
33. Kuznetsov K. Service migration algorithm for distributed edge computing in 5G/6G networks / Kuzmina E., Lapteva T., Volkov A., Muthanna A., Aziz A. // *International Conference on Next Generation Wired/Wireless Networking. Lecture Notes in Computer Science (LNCS)*. Cham: Springer Nature Switzerland. Vol 14542. pp. 320-337.
34. Recommendation ITU-R M.2083-0 (09/2015). IMT Vision – Framework and overall objectives of the future development of IMT for 2020 and beyond.
35. Paramonov, A. An efficient method for choosing digital cluster size in ultralow latency networks / A. Paramonov, M. Khayyat, N. Chistova et al. // *Wireless Communications and Mobile Computing*. – Vol. 2021. – Article 9188658.
36. Кучерявый, А.Е. Тактильный интернет. Сети связи со сверхмалыми задержками / А.Е. Кучерявый, М.А. Маколкина, Р.В. Киричек // *Электросвязь*. – 2016. – № 1. – С. 44-46.
37. Recommendation ITU-T Y.3104 (12/2018). Architecture of the IMT-2020 network.

38. Кузнецов К.А. Интеллектуальное ядро сети для интегрированной сети 6G на основе технологии SDN для поддержки приложений телеприсутствия // Электросвязь. 2024. № 8. С. 26-31.
39. Мутханна, А.С.А. Интегральное решение проблемы размещения контроллеров и балансировки нагрузки / А.С.А. Мутханна // Труды учебных заведений связи. – 2023. – Т. 9, № 2. – С. 81-93.
40. Taleb, T. 6G System Architecture: A Service of Services Vision / Tarik Taleb, Chaika Benzaïd, Miguel Bordallo Lopez and etc. //ITU Journal on Future and Evolving Technologies, Volume 3, Issue 3, December 2022.
41. ITU-T Technical Report. Network 2030 - Additional Representative Use Cases and Key Network Requirements for Network 2030. Telecommunication standardization sector of ITU, June 2020.
42. ITU-T Deliverable. New Services and Capabilities for Network 2030: Description, Technical Gap and Performance Target Analysis. Telecommunication standardization sector of ITU, 10/2019.
43. Network 2030. A Blueprint of Technology, Applications and Market Drivers Towards the Year 2030 and Beyond. Written by FG-NET-2030.
44. ITU-T Technical Specification. Network 2030 Architecture Framework. Telecommunication standardization sector of ITU, June 2020.
45. Волков А.Н., Перспективные исследования сетей и услуг 2030 в лаборатории 6G Meganetlab СПбГУТ. / Волков А.Н., Мутханна А.С.А., Кучерявый А.Е., и др. // Электросвязь. 2023. № 6. С. 5-14.
46. Демидов Н.А., Исследование трафика 3d-видеопотока на имитационной модели. / Демидов Н.А. // Электросвязь. 2024. № 3. С. 44-48.
47. ITU Focus Group Technical Specification. Definition of metaverse. 12/2023.
48. P. V. Mane-Deshmukh, “Designing of Wireless Sensor Network to Protect Agricultural Farm from Wild Animals,” i-Manager’s Journal on Information Technology, vol. 7, no. 4, p. 30, 2018.
49. Claude E. Shannon and Warren Weaver. The Mathematical Theory of Communication. // The University of Illinois press. Urbana·1964.

50. Akyildiz, F. Terahertz Band Communication: An Old Problem Revisited and Research Directions for the Next Decade. / Ian F. Akyildiz, Life Fellow, Chong Han, Zhifeng Hu, Shuai Nie, Josep M. Jornet // arXiv:2112.13187v3 [eess.SP] 27 Apr 2022.
51. Vitaly, Petrov., Alexander, Pyattaev., Dmitri, Moltchanov., Yevgeni, Koucheryavy. (2016). Terahertz band communications: Applications, research challenges, and standardization activities. 183-190. doi: 10.1109/ICUMT.2016.7765354.
52. <https://www.amazon.com/5g-router/s?k=5g+router>
53. Дорохова А.А., Исследование трафика и качества обслуживания в самоорганизующихся сетях на базе БПЛА. / Дорохова А.А., Парамонов А.И. // Информационные технологии и телекоммуникации. 2016. Т. 4. № 2. С. 12-25.
54. Варельджян К.С., Оптимизация траектории движения бпла в летающих сенсорных сетях. / Варельджян К.С., Парамонов А.И., Киричек Р.В. // Электросвязь. 2015. № 7. С. 20-25.
55. Захаров М.В., Задача распределения ресурсов в группах БПЛА. / Захаров М.В., Киричек Р.В., Парамонов А.И. // Информационные технологии и телекоммуникации. 2015. Т. 3. № 1. С. 62-70.
56. Вишневский В.М., Методы и алгоритмы проектирования и реализации привязных высотных беспилотных телекоммуникационных платформ. / Вишневский В.М. // В сборнике: XIII Всероссийское совещание по проблемам управления ВСПУ-2019. Сборник трудов XIII Всероссийского совещания по проблемам управления ВСПУ-2019. Институт проблем управления им. В.А. Трапезникова РАН. 2019. С. 40-42.
57. Кузнецов К.А. Методы размещения маршрутизаторов сети на беспилотных летательных аппаратах (БПЛА) и обеспечения маршрутизации с помощью роя БПЛА / Парамонов А.И., Мутханна А.С.А., Кучерявый А.Е. // Труды учебных заведений связи 2024. Т. 10. № 4. С. 24-34.

58. Дж.-О. Ким, Факторный, дискриминантный и кластерный анализ / Дж.-О. Ким, Ч. У. Мьюллер, У. Р. Клекка и др. // М.: Финансы и статистика, 1989.— 215 с: ил. ISBN 5-279-00247-X.
59. <https://scikit-learn.org/stable/modules/clustering.html>
60. Марочкина А.В., Моделирование и кластеризация трехмерной сети интернета вещей с применением метода оценки фрактальной размерности. / Марочкина А.В. // Электросвязь. 2023. № 6. С. 60-66.
61. Загоруйко Н. Г., Ёлкина В. Н., Лбов Г. С. Алгоритмы обнаружения эмпирических закономерностей. — Новосибирск: Наука, 1985. — 999 с.
62. Викулов А.С., Модель канала OFDM в задаче оценки эффективности сети IEEE 802.11. / Викулов А.С., Парамонов А.И. // Инфокоммуникационные технологии. 2018. Т. 16. № 3. С. 290-297.
63. Рекомендация МСЭ-R P.1238-9 (ITU-R P.1238-9), Данные о распространении радиоволн и методы прогнозирования для планирования систем радиосвязи внутри помещений и локальных зонных радиосетей в частотном диапазоне 300 МГц – 100 ГГц. Серия Р. Распространение радиоволн. – МСЭ-R, Июнь 2017. – 28 с.
64. An Introduction to the Theory of Point Processes: Volume I: Elementary Theory and Methods, Second Edition. / D.J. Daley, D. Vere-Jones // Springer Science & Business Media, 10 apr. 2006 г. - PP: 471.
65. Препарата Ф., Шеймос М. Вычислительная геометрия. Введение 1989.
66. Майника Э. Алгоритмы оптимизации на сетях и графах. М. Мир. 1981. С.323.
67. Reinsel, D., Gantz, J., Rydning, J.: Data Age 2025: the digitization of the world from edge to core/Framingham: An IDC White Paper, Sponsored by Seagate, pp. 1–25 (2018)
68. Alsbouí, T., Hammoudeh, M., Bandar, Z., Nisbet, A.: August. an overview and classification of approaches to information extraction in wireless sensor networks. In: Proceedings of the 5th International Conference on Sensor Technologies and Applications (SENSORCOMM 2011), vol. 255 (2011)

69. Muthanna, M.S.A., et al.: Deep reinforcement learning based transmission policy enforcement and multi-hop routing in QoS aware LoRa IoT networks. *Comput. Commun.. Commun.* 183, 33–50 (2022)
70. Ateya, A.A., Muthanna, A., Vybornova, A., Pyatkina, D., Koucheryavy, A.: Energy-aware offloading algorithm for multi-level cloud based 5G system. In: *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*, pp. 355–370 (2018)
71. Hammoudeh, M., Newman, R.: Information extraction from sensor networks using the Watershed transform algorithm. *Inf. Fusion* 22, 39–49 (2015)
72. Dame, M.: *The Kubernetes Operator Framework Book: Overcome complex Kubernetes cluster management challenges with automation toolkits* (2022)
73. Kubernetes Pods [website]. Text: electronic (2023). <https://www.knowledgehut.com/blog/devops/kubernetes-pods>. Accessed 22 Aug 2023
74. Muschko, B.: *Certified Kubernetes Application Developer (CKAD) Study Guide: In-Depth Guidance and Practice*. O'Reilly Media, Sebastopol (2021)
75. Rosso, J., Lander, R., Brand, A., Harris, J.: *Production Kubernetes*. O'Reilly Media, Sebastopol (2021)
76. Kubernetes Documentation [website]. Text: electronic (2023). <https://kubernetes.io/docs/home/>. Accessed 22 Aug 2023
77. Ma, L., Yi, S., Carter, N., Li, Q.: Efficient live migration of edge services leveraging container layered storage. *IEEE Trans. Mob. Comput.* (2018)
78. Welcome to CRIU, a project [website]. Text: electronic (2023). [https://criu.org/Main\\_Page](https://criu.org/Main_Page). Accessed 23 Aug 2023
79. Kube-stresscheck Script to check Kubernetes nodes on stress (CPU/RAM) resistance. [website]. Text: electronic (2018). <https://github.com/giantswarm/kube-stresscheck>
80. Puliafito, C., Vallati, C., Mingozzi, E., Merlino, G., Longo, F., Puliafito, A.: Container migration in the fog: a performance evaluation. *Sensors* 19(7), 1–22 (2019)

81. Bailey, D.H.: The BBP Algorithm for Pi [website]. Text: electronic (2006). <https://www.experimentalmath.info/bbp-codes/bbp-alg.pdf>. Accessed 25 Aug 2023.
82. Mohan, A., Yezalaleul, J., Chen, A., Enkhjargal, T.: Seamless container migration between cloud and edge. CA, Santa Clara: Santa Clara University (2021)
83. Кузнецов К.А. Тактильный интернет и его приложения / Мутханна А.С.А., Кучерявый А.Е. // Информационные технологии и телекоммуникации.2019. Т. 7. №2. С. 12-20.
84. ESP32 Series Datasheet // Espressif systems, 2022.
85. An Open-Source, Cloud-Native, Distributed MQTT Broker for IoT. Дата обращения 02.05.2022 emqx.io.
86. L. V. Le, D. Sinh, L. P. Tung, and B. S. Paul Lin, "A practical model for traffic forecasting based on big data, machine learning, and network KPIs" // Consumer Communications Networking Conference, 15th IEEE Annual, pp. 1-4. IEEE, 2018.
87. P. M. Kumar, U. Devi, G. Manogaran, R. Sundarasekar, N. Chilamkurti, and R. Varatharajan, "Ant colony optimization algorithm with Internet of Vehicles for intelligent traffic control system" // Computer Networks, Vol. 144, p. 154-162, 2018.
88. Shi-Wan Lin, Bradford Miller, Jacques Durand, Graham Bleakley "The Industrial Internet of Things Volume G1: Reference Architecture" // Industrial Internet Consortium. 2016.
89. Kuznetsov K. Revolutionizing H2M interaction: telepresence system enabling sign language expansion for individuals with disabilities / Muthanna A., Ateya A. A., Koucheryavy A. // Lecture Notes in Computer Science (LNCS). Cham: Springer Nature Switzerland. 2023. Vol. 14123. pp. 21-33.
90. Кузнецов К.А. Методы реализации иммерсивных технологий в сетях связи пятого поколения / Мутханна А.С.А. // XI Международная научно-техническая и научно-методическая конференция: сборник научных статей. 2022. Т. 1. С. 627-631.

91. Кузнецов К.А. Комплекс средств натурального моделирования услуг иммерсивных технологий для людей с ограниченными возможностями / Леонова М.Д., Мутханна А.С.А. // Информационные технологии и телекоммуникации.2022. Т. 10. №4. С. 13-26.

## ПРИЛОЖЕНИЕ А

### Исполняемый код ESP-32

```
#include "I2Cdev.h"
#include "MPU6050_6Axis_MotionApps20.h"
#include <WiFi.h>
#include <PubSubClient.h>
#if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
#include "Wire.h"
#endif
#define OUTPUT_READABLE_YAWPITCHROLL
#define INTERRUPT_PIN 18
#define FILO_BEFORE_LEN 2
#define FILO_BEFORE_COUNT 100

MPU6050 mpu;
bool dmpReady = false;
uint8_t mpuIntStatus;
uint8_t devStatus;
uint16_t packetSize;
uint16_t fifoCount;
uint8_t fifoBuffer[64];

Quaternion q;
VectorFloat gravity;
float ypr[3];

const char *ssid = "Bonch";
const char *password = "*****";
const char *mqtt_broker = "192.168.1.71";
const int mqtt_port = 1883;
const char *topic = "Bonch";
String client_id = "esp32_client";

WiFiClient espClient;
```

```

PubSubClient client(espClient);

int          Flex[5][2],          analogRead_data[5],
FILO_before[5][FILO_BEFORE_COUNT][FILO_BEFORE_LEN];
int VG[5][2] = {
    {0, 255}, {255, 0}, {255, 0}, {0, 255}, {0, 255}
};
String STR_out = "";
int TXbuff[5];

TaskHandle_t Task1;
TaskHandle_t Task2;

volatile bool mpuInterrupt = false;    // indicates whether MPU
interrupt pin has gone high
void dmpDataReady() {
    mpuInterrupt = true;
}

void setup() {

    #if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
        Wire.begin();
        Wire.setClock(400000);
    #elif I2CDEV_IMPLEMENTATION == I2CDEV_BUILTIN_FASTWIRE
        Fastwire::setup(400, true);
    #endif

    Serial.begin(115200);
    while (!Serial);

    //-----PALM-----
    delay(1000);
    int timer = millis();
    for(int i=0; i<5; i++){

```

```

Flex[i][0] = 1024;
Flex[i][1] = 0;
}
Serial.println("calibration 1 + 5 sec");
delay(1000);
while(millis()-timer < 5000){
  for(int i = 0; i<5; i++){
    analogRead_data[i] = myAnalogRead(i);
    if(analogRead_data[i] > Flex[i][1]) Flex[i][1] =
analogRead_data[i];
    if(analogRead_data[i] < Flex[i][0]) Flex[i][0] =
analogRead_data[i];
    Serial.print(String(analogRead_data[i]) + "\t");
  }
  Serial.println();
}
Serial.println("Result:");
for(int i=0; i<5; i++){
  Serial.println(String(Flex[i][0]) + "\t" +
String(Flex[i][1]));
}
delay(2000);

//-----WiFi-----
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.println("Connecting to WiFi..");
}
Serial.println("Connected to the WiFi network");
//connecting to a mqtt broker
client.setServer(mqtt_broker, mqtt_port);
//client.setCallback(callback);
while (!client.connected()) {
  client_id += String(WiFi.macAddress());

```

```

        Serial.printf("The client %s connects to the public
mqtt broker\n", client_id.c_str());
        //          if          (client.connect(client_id.c_str(),
mqtt_username, mqtt_password))
        if (client.connect(client_id.c_str(), mqtt_username,
mqtt_password)) {
            Serial.println("Emqx mqtt broker connected");
        } else {
            Serial.print("failed with state ");
            Serial.print(client.state());
            delay(2000);
        }
    }
    delay(1000);

//-----MPU-----
    Serial.println(F("Initializing I2C devices..."));
    mpu.initialize();
    pinMode(INTERRUPT_PIN, INPUT);

    Serial.println(F("Testing device connections..."));
    Serial.println(mpu.testConnection() ? F("MPU6050 connection
successful") : F("MPU6050 connection failed"));

    Serial.println(F("Initializing DMP..."));
    devStatus = mpu.dmpInitialize();

    mpu.setXGyroOffset(220);
    mpu.setYGyroOffset(76);
    mpu.setZGyroOffset(-85);
    mpu.setZAccelOffset(1788);

    if (devStatus == 0) {
        mpu.CalibrateAccel(6);
        mpu.CalibrateGyro(6);
    }

```

```

    mpu.PrintActiveOffsets();
    Serial.println(F("Enabling DMP..."));
    mpu.setDMPEnabled(true);
    Serial.print(F("Enabling interrupt detection (Arduino
external interrupt "));
    Serial.print(digitalPinToInterrupt(INTERRUPT_PIN));
    Serial.println(F(")..."));
    attachInterrupt(digitalPinToInterrupt(INTERRUPT_PIN),
dmpDataReady, RISING);
    mpuIntStatus = mpu.getIntStatus();
    Serial.println(F("DMP ready! Waiting for first
interrupt..."));
    dmpReady = true;
    packetSize = mpu.dmpGetFIFOPageSize();
} else {
    // ERROR!
    // 1 = initial memory load failed
    // 2 = DMP configuration updates failed
    // (if it's going to break, usually the code will be 1)
    Serial.print(F("DMP Initialization failed (code "));
    Serial.print(devStatus);
    Serial.println(F(")"));
}
xTaskCreatePinnedToCore(Task1code, "Task1", 10000, NULL, 5, &Task1, 1
);

delay(500);

xTaskCreatePinnedToCore(Task2code, "Task2", 10000, NULL, 10, &Task1, 0);
delay(500);
}
void Task2code( void * parameter) { //void Task2code( void *
parameter)
    while(true) {
        for(int i=0; i<5; i++) {
            analogRead_data[i] = myAnalogRead(i);

```

```

        if(analogRead_data[i] < Flex[i][0]) analogRead_data[i] =
Flex[i][0];
        if(analogRead_data[i] > Flex[i][1]) analogRead_data[i] =
Flex[i][1];

        for(int j=FILO_BEFORE_LEN-1; j>0; j--)
            FILO_before[i][0][j] = FILO_before[i][0][j-1];
        FILO_before[i][0][0] = analogRead_data[i]
        for(int k=1; k<FILO_BEFORE_COUNT; k++){
            for(int j=FILO_BEFORE_LEN-1; j>0; j--)
                FILO_before[i][k][j] = FILO_before[i][k][j-1];
            for(int m=0; m<FILO_BEFORE_LEN-1; m++)
                FILO_before[i][k-1][FILO_BEFORE_LEN-1] +=
FILO_before[i][k-1][m];
            FILO_before[i][k][0] = FILO_before[i][k-
1][FILO_BEFORE_LEN-1]/FILO_BEFORE_LEN;
        }
        for(int n=0; n<FILO_BEFORE_LEN-1; n++)
            FILO_before[i][FILO_BEFORE_COUNT-
1][FILO_BEFORE_LEN-1] += FILO_before[i][FILO_BEFORE_COUNT-1][n];
        TXbuff[i] = FILO_before[i][FILO_BEFORE_COUNT-
1][FILO_BEFORE_LEN-1]/FILO_BEFORE_LEN;
        STR_out += String(map(TXbuff[i], Flex[i][0],
Flex[i][1], VG[i][0], VG[i][1])) + ";");
        //Serial.print(String(map(TXbuff[i], Flex[i][0],
Flex[i][1], VG[i][0], VG[i][1]))+"\t");
    }
    String ypr0 = String(ypr[0]*180/M_PI, DEC);
    String ypr1 = String(ypr[1]*180/M_PI, DEC);
    String ypr2 = String(ypr[2]*180/M_PI, DEC);
    String full = String(STR_out + ypr0 + ";" + ypr1 + ";" +
ypr2);
    client.publish(topic1, full.c_str());
    STR_out = "";
    delay(5);
}

```

```

}

void Task1code( void * parameter) {
    while(true){
        if (!dmpReady) return;
        if (mpu.dmpGetCurrentFIFOPacket(fifoBuffer)) {
            #ifdef OUTPUT_READABLE_YAWPITCHROLL
                // display Euler angles in degrees
                mpu.dmpGetQuaternion(&q, fifoBuffer);
                mpu.dmpGetGravity(&gravity, &q);
                mpu.dmpGetYawPitchRoll(ypr, &q, &gravity);
                /*Serial.print("ypr\t");
                Serial.print(ypr[0] * 180/M_PI);
                Serial.print("\t");
                Serial.print(ypr[1] * 180/M_PI);
                Serial.print("\t");
                Serial.println(ypr[2] * 180/M_PI);*/
            #endif
        }
        }
        delay(5);
    }

void loop() {
}

int myAnalogRead(int i) {
    if (i == 0) return analogRead(32); //
    else if (i == 1) return analogRead(33); //
    else if (i == 2) return analogRead(34); //
    else if (i == 3) return analogRead(35); //
    else if (i == 4) return analogRead(36); //sp
    else if (i == 5) return analogRead(39); //sn
    else return 0;
}

```

**ПРИЛОЖЕНИЕ Б****Исполняемый код Processing**

```
import processing.serial.*;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
import mqtt.*;

String REGEXP = "(.*);(.*);(.*);(.*);(.*);(.*);(.*);(.*));";
Pattern pattern = Pattern.compile(REGEXP);
Serial port;
int interval = 0;

float[] fingerJoynt1 = new float[5];
float[] fingerJoynt1speed = new float[5];
float[] fingerJoynt2and3 = new float[5];
float[] fingerJoynt2and3speed = new float[5];
float gRotateX = 0.83;
float gRotateY = 0;
float gRotateZ = 3.48;
float gRotateXspeed = 0.0;
float gRotateYspeed = 0;
float gRotateZspeed = 0;
float[] finger = new float[5];
float[] fingerMin = new float[5];
float[] fingerMax = new float[5];

StringBuilder value = new StringBuilder();
float[] ypr = new float[3];
MQTTClient client;

void setup() {
  fullScreen(P3D);
  noStroke();
```

```

for (int i = 0; i<5; i++) {
    fingerJoynt1[i] = 0.0;
    fingerJoynt2and3[i] = 0.0;
    fingerJoynt1speed[i] = 0.0;
    fingerJoynt2and3speed[i] = 0.0;
}
client = new MQTTClient(this);
client.connect("mqtt://192.168.1.71", "processing");
client.subscribe("Bonch");
}

void draw() {
    if (millis() - interval > 1000) {
        interval = millis();
    }
    background(0);
    ambientLight(50, 50, 50);
    directionalLight(255, 255, 255, .1, .17, -0.5);
    calculateJoynt();
    translate(width / 2, height / 2, 550);
    drawHand();
}

void drawHand() {
    if (millis() - interval > 1000) {
        interval = millis();
    }

    //Base
    fill(#D69A50);

    rotateX(ypr[1]/50);
    rotateY(-ypr[0]/50);
    rotateZ(-ypr[2]/50);

```

```

box(60, 10, 50);

//Lillefinger
translate(30, 0, 20);
drawFinger(fingerJoynt1[0], fingerJoynt2and3[0], 13, 12);

//Ringfinger
translate(0, 0, -13);
drawFinger(fingerJoynt1[1], fingerJoynt2and3[1], 18, 15);

//Langemann
translate(0, 0, -14);
drawFinger(fingerJoynt1[2], fingerJoynt2and3[2], 20, 16);

//Pekeringer
translate(0, 0, -13);
drawFinger(fingerJoynt1[3], fingerJoynt2and3[3], 17, 14);

//Tommel
translate(-55, 0, -7);
rotateY(1.83);
drawFinger(fingerJoynt1[4], fingerJoynt2and3[4], 30, 15);
}

void drawFinger(float Joynt1, float Joynt2and3, int
fingerLength1, int fingerLength2) {
    pushMatrix();
    rotateZ(Joynt1);

    fill(#D69A50);
    sphere(6);

    translate(fingerLength1/2, 0, 0);
    fill(#D69A50);

```

```

    box(fingerLength1, 10, 10);

    translate(fingerLength1/2, 0, 0);
    rotateZ(Joynt2and3);

    fill(#D69A50);
    sphere(6);

    translate(fingerLength2/2, 0, 0);
    fill(#D69A50);
    box(fingerLength2, 10, 10);

    translate(fingerLength2/2, 0, 0);
    rotateZ(Joynt2and3);

    fill(#D69A50);
    sphere(6);

    translate(fingerLength2/2, 0, 0);
    fill(#D69A50);
    box(fingerLength2, 10, 10);

    popMatrix();
}

void calculateJoynt() {
    gRotateX += gRotateXspeed;
    gRotateZ += gRotateZspeed;

    for (int i = 0; i<5; i++) {
        fingerJoynt1[i] += fingerJoynt1speed[i];

        if(fingerJoynt1[i] < 0) {
            fingerJoynt1[i] = 0;
        }
    }
}

```

```

    if(fingerJoynt1[i]>1.5) {
        fingerJoynt1[i] = 1.5;
    }

    fingerJoynt2and3[i] += fingerJoynt2and3speed[i];

    if(fingerJoynt2and3[i] < 0) {
        fingerJoynt2and3[i] = 0;
    }
    if(fingerJoynt2and3[i]>1.5) {
        fingerJoynt2and3[i] = 1.5;
    }
}
}

void messageReceived(String topic, byte[] payload) {
    String full = new String(payload);
    Matcher m = pattern.matcher(full);
    if (m.find( )) {
        ypr[0] = Float.parseFloat(m.group(6));
        ypr[1] = Float.parseFloat(m.group(7));
        ypr[2] = Float.parseFloat(m.group(8));

        fingerJoynt1[0] = Float.parseFloat(m.group(1)) / 100.;
        fingerJoynt2and3[0] = Float.parseFloat(m.group(1)) / 100.;
        fingerJoynt1[1] = Float.parseFloat(m.group(2)) / 100.;
        fingerJoynt2and3[1] = Float.parseFloat(m.group(2)) / 100.;
        fingerJoynt1[2] = Float.parseFloat(m.group(3)) / 100.;
        fingerJoynt2and3[2] = Float.parseFloat(m.group(3)) / 100.;
        fingerJoynt1[3] = Float.parseFloat(m.group(4)) / 100.;
        fingerJoynt2and3[3] = Float.parseFloat(m.group(4)) / 100.;
        fingerJoynt1[4] = Float.parseFloat(m.group(5)) / 100.;
        fingerJoynt2and3[4] = Float.parseFloat(m.group(5)) / 100.;
    }
}
}

```

## ПРИЛОЖЕНИЕ В

### Акты о внедрении результатов исследования

МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ,  
СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ  
УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ  
ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»  
(СПбГУТ)

Юридический адрес: набережная реки Мойки,  
д. 61, литера А, Санкт-Петербург, 191186

Почтовый адрес: пр. Большевиков, д. 22, корп. 1,  
Санкт-Петербург, 193232  
Тел.(812) 3263156, Факс: (812) 3263159  
<http://sut.ru>  
E-mail: [rector@sut.ru](mailto:rector@sut.ru)  
ОКПО 01179934 ОГРН 1027809197635  
ИНН 7808004760 КПП 784001001  
ОКТМО 40909000

12.09.2024 № 06/154  
на № \_\_\_\_\_ от \_\_\_\_\_

УТВЕРЖДАЮ

И.о. проректора по научной работе,  
д-р. техн. наук, доцент

А.В. Рабин



Акт

о внедрении научных результатов,  
полученных Кузнецовым Константином Алексеевичем

Комиссия в составе Елагина В.С., и.о. декана факультета инфокоммуникационных сетей и систем, Дунайцева Р.А., доцента кафедры сетей связи и передачи данных и Ворожейкиной О.И., заведующей лабораторией кафедры сетей связи и передачи данных составила настоящий акт о том, что научные результаты, полученные Кузнецовым Константином Алексеевичем, использованы в учебном процессе СПбГУТ.

При чтении лекций и проведении практических занятий по курсу «Интернет вещей и самоорганизующиеся сети» (Рабочая Программа № 23.05/105-Д, утв. 23.06.2023), разделы программы:

- Сети связи 2030. Децентрализация сетей связи. Услуги телеприсутствия.
- Требования к качеству обслуживания и качеству восприятия, модели трафика для самоорганизующихся сетей.

При чтении лекций и проведении практических занятий по курсу «Современные проблемы науки в области инфокоммуникаций» (Рабочая Программа № 23.05/344-Д, утв. 23.06.2023), раздел программы:

- Концепции развития сетей связи. Текущее состояние развития сетей. Прогнозы развития сетей связи.

При чтении лекций и проведении практических занятий по курсу «Системы, сети и устройства телекоммуникаций» (Рабочая Программа № 21.05/780-Д, утв. 03.07.2021), раздел Программы:

- Методы исследования процессов, явлений и объектов.

- Эффективность функционирования (использования) систем, сетей и устройств связи.

В указанных дисциплинах используются следующие новые научные результаты, полученные Кузнецовым Константином Алексеевичем в диссертационной работе:

Метод построения интеллектуального ядра интегрированной сети 6G на основе технологии SDN для услуг телеприсутствия, снижающий среднее число контроллеров в мультиконтроллерных сетях SDN на 46% и увеличивающий при этом эффективность использования контроллеров на 53%.

Модель и методы размещения маршрутизаторов сети на беспилотных летательных аппаратах (БПЛА) и обеспечения маршрутизации с помощью роя БПЛА, позволяющие увеличить эффективность использования сети на 50%.

Метод реализации иммерсивных технологий в сетях связи пятого и шестого поколений, позволяющий в режиме реального времени передавать жестовую речь, и требования к качеству обслуживания и качеству восприятия таких услуг.

Полученные научные результаты использованы при выполнении Соглашения о предоставлении из федерального бюджета гранта в форме субсидий, выделяемого для государственной поддержки научных исследований, проводимых под руководством ведущих ученых в российских образовательных организациях высшего образования, научных учреждениях и государственных научных центрах Российской Федерации от "06" июля 2022г. № 075-15-2022-1137 по приоритетному направлению научно-технологического развития Российской Федерации 20а - Переход к передовым цифровым, интеллектуальным производственным технологиям, роботизированным системам, новым материалам и способам конструирования, создание систем обработки больших объемов данных, машинного обучения и искусственного интеллекта.

И.о. декана факультета ИКСС,  
канд. техн. наук, доцент

Доцент кафедры ССиПД,  
канд. техн. наук

Зав. лабораторией кафедры ССиПД

В.С. Елагин

Р.А. Дунайцев

О.И. Ворожейкина

**УТВЕРЖДАЮ**

Заместитель генерального директора  
ПАО «ГИПРОСВЯЗЬ»  
А.Б. Васильев

АКТ

о внедрении научных результатов, полученных  
Кузнецовым Константином Алексеевичем в диссертационной работе  
«Исследование и разработка методов предоставления услуг  
телеприсутствия в сетях связи шестого поколения»

Настоящим актом подтверждаем, что научные результаты диссертационной работы Кузнецова Константина Алексеевича «Исследование и разработка методов предоставления услуг телеприсутствия в сетях связи шестого поколения», представленной на соискание ученой степени кандидата технических наук, внедрены в ПАО «ГИПРОСВЯЗЬ» при разработке «Методики планирования мультиконтроллерных программно-конфигурируемых сетей».

При разработке были использованы следующие новые научные результаты из диссертации К.А. Кузнецова:

- Метод построения интеллектуального ядра интегрированной сети 6G на основе технологии SDN для услуг телеприсутствия, снижающий среднее число контроллеров в мультиконтроллерных сетях SDN на 46% и увеличивающий при этом эффективность использования контроллеров на 53%.

**Председатель комиссии:**

Директор департамента  
по системно-сетевым разработкам,  
экспертизе и консалтингу

А.А. Иванов

**Члены комиссии:**

Начальник информационно-  
аналитического отдела

Ю.А. Нопина