

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Институт кибернетики

Направление подготовки 09.04.02 Информационные системы и технологии

Кафедра Информационных систем и технологий

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Тема работы
Разработка алгоритмического и программного обеспечения для передачи данных в сетях MANET

УДК 004.724.2:004.415,2:004.021

Студент

Группа	ФИО	Подпись	Дата
8ИМ5А	Никитин Илья Александрович		

Руководитель

Должность	ФИО	Ученая степень, звание	Подпись	Дата
доцент	Шерстнев В.С.	к.т.н.		

КОНСУЛЬТАНТЫ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
доцент	Попова С.Н.	к.э.н.		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
ассистент	Акулов П.А.			

ДОПУСТИТЬ К ЗАЩИТЕ:

Зав. кафедрой	ФИО	Ученая степень, звание	Подпись	Дата
Информационных систем и технологий	Мальчуков А.Н.	к.т.н.		

Томск – 2017 г.

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Код результ атов	Результат обучения (выпускник должен быть готов)
Общепрофессиональные компетенции	
P1	Воспринимать и самостоятельно приобретать, развивать и применять математические, естественнонаучные, социально-экономические и профессиональные знания для решения нестандартных задач, в том числе в новой или незнакомой среде и в междисциплинарном контексте.
P2	Владеть и применять методы и средства получения, хранения, переработки и трансляции информации посредством современных компьютерных технологий, в том числе в глобальных компьютерных сетях.
P3	Демонстрировать культуру мышления, способность выстраивать логику рассуждений и высказываний, основанных на интерпретации данных, интегрированных из разных областей науки и техники, выносить суждения на основании неполных данных, анализировать профессиональную информацию, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров с обоснованными выводами и рекомендациями.
P4	Анализировать и оценивать уровни своих компетенций в сочетании со способностью и готовностью к саморегулированию дальнейшего образования и профессиональной мобильности. Владеть, по крайней мере, одним из иностранных языков на уровне социального и профессионального общения, применять специальную лексику и профессиональную терминологию языка.
Профессиональные компетенции	
P5	Разрабатывать стратегии и цели проектирования, критерии эффективности и ограничения применимости, новые методы, средства и технологии проектирования геоинформационных систем (ГИС) или промышленного программного обеспечения.
P6	Планировать и проводить теоретические и экспериментальные исследования в области создания интеллектуальных ГИС и ГИС технологии или промышленного программного обеспечения с использованием методов системной инженерии.
P7	Осуществлять авторское сопровождение процессов проектирования, внедрения и сопровождения ГИС и ГИС технологий или промышленного программного обеспечения с использованием методов и средств системной инженерии, осуществлять подготовку и обучение персонала.
P8	Формировать новые конкурентоспособные идеи в области теории и практики ГИС и ГИС технологий или системной инженерии программного обеспечения. Разрабатывать методы решения нестандартных задач и новые методы решения традиционных задач. Организовывать взаимодействие коллективов, принимать управленческие решения, находить компромисс между различными требованиями как при долгосрочном, так и при краткосрочном планировании.
Общекультурные компетенции	
P9	Использовать на практике умения и навыки в организации исследовательских, проектных работ и профессиональной эксплуатации современного оборудования и приборов, в управлении коллективом.
P10	Свободно пользоваться русским и иностранным языками как средством делового общения.

P11	Совершенствовать и развивать свой интеллектуальный и общекультурный уровень. Проявлять инициативу, в том числе в ситуациях риска, брать на себя всю полноту ответственности.
P12	Демонстрировать способность к самостоятельному обучению новым методам исследования, к изменению научного и научно-производственного профиля своей профессиональной деятельности, способность самостоятельно приобретать с помощью информационных технологий и использовать в практической деятельности новые знания и умения, в том числе в новых областях знаний, непосредственно не связанных со сферой деятельности, способность к педагогической деятельности.

Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Институт кибернетики
Направление подготовки 09.04.02 Информационные системы и технологии
Кафедра информационных систем и технологий

УТВЕРЖДАЮ:
Зав. кафедрой
_____ Мальчуков А.Н.

ЗАДАНИЕ
на выполнение выпускной квалификационной работы

В форме:

Магистерской диссертации

Студенту:

Группа	ФИО
8ИМ5А	Никитин Илья Александрович

Тема работы:

Разработка алгоритмического и программного обеспечения для передачи данных в сетях Manet	
Утверждена приказом директора (дата, номер)	897/С от 20.02.2017 г.

Срок сдачи студентом выполненной работы:	
--	--

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе	Разработать алгоритмическое и программное обеспечение для передачи данных в сетях Manet. Исходными данными к работе является техническое задание на разработку программного продукта.
Перечень подлежащих исследованию, проектированию и разработке вопросов	<ol style="list-style-type: none">1) Изучение предметной области.2) Обзор аналогов программного продукта.3) Обзор существующих алгоритмов маршрутизации.4) Проектирование алгоритмов.5) Выбор платформы для реализации ПО.6) Программная реализация спроектированных алгоритмов.7) Тестирование мобильного приложения.

Перечень графического материала	Блок-схемы алгоритмов, макеты интерфейса, изображения разработанного интерфейса.
Консультанты по разделам выпускной квалификационной работы	
Раздел	Консультант
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Попова С.Н.
Социальная ответственность	Акулов П.А.
Раздел на иностранном языке	Горбатова Т.Н. Мирошниченко Е.А.
Названия разделов, которые должны быть написаны на русском и иностранном языках:	
Аналитический обзор	

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	
---	--

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент каф. ИСТ	Шерстнев В.С.	К.Т.Н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ИМ5А	Никитин И.А.		

Образец графика выполнения ВКР
Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Институт кибернетики
Направление подготовки 09.04.02 Информационные системы и технологии
Кафедра информационных систем и технологий
Период выполнения весенний семестр 2016/2017 учебного года)

Форма представления работы:

Магистерская диссертация

КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН
выполнения выпускной квалификационной работы

Срок сдачи студентом выполненной работы:

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
3.02.2017 г	Анализ предметной области	5
16.02.2017 г	Анализ аналогов программного продукта и алгоритмического обеспечения	15
20.02.2017 г	Проектирование алгоритмического обеспечения	20
20.03.2017 г	Разработка программного продукта	20
21.04.2017 г	Тестирование ПО	10
2.05.2017 г	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	10
12.05.2017 г	Социальная ответственность	10
18.05.2017 г	Обязательное приложение на иностранном языке	5
30.05.2017 г	Оформление пояснительной записки	5

Составил преподаватель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент каф. ИСТ	Шерстнев В.С.	к.т.н.		

СОГЛАСОВАНО:

Зав. кафедрой	ФИО	Ученая степень, звание	Подпись	Дата
ИСТ	Мальчуков А.Н.	к.т.н.		

РЕФЕРАТ

Отчет по преддипломной практике содержит 109 страниц, 25 рисунков, 32 таблицы, 28 источников, 1 публикация.

Ключевые слова: алгоритмы маршрутизации, проектирование, разработка, данные, мобильное приложение.

Объектом исследования работы являются алгоритмы маршрутизации, проектирование алгоритмов маршрутизации для сети с использованием технологии Wi-Fi Direct, программная реализация спроектированных алгоритмов маршрутизации.

Цель дипломной работы – спроектировать алгоритмы маршрутизации для сетей построенных с помощью технологии Wi-Fi Direct и реализовать спроектированные алгоритмы.

Поставленные задачи: найти и проанализировать существующие аналоги, проанализировать существующие протоколы маршрутизации в сетях Manet, спроектировать алгоритмы для передачи данных с помощью технологии Wi-Fi Direct, программно реализовать спроектированные алгоритмы, протестировать полученное программное обеспечение.

В процессе исследования проводилось: анализ аналогов программного продукта и анализ наиболее распространенные протоколов маршрутизации в беспроводных сетях.

Основные конструктивные, технические и технико-эксплуатационные характеристики: передача данных основывается на технологии Wi-Fi Direct и протоколах маршрутизации, разработанных для сетей Manet.

Область применения: передача данных между устройствами без использования точек доступа.

Экономическая эффективность/значимость работы: результат позволяет передавать данные между мобильными устройствами не подключаясь к стационарным точкам доступа, что позволяет снизить расходы на оплату «internet».

Определения

RREQ (route request) – запрос маршрута.

RREP (route reply) – ответ на запрос маршрута.

RERR (route error) – ошибка запроса.

ACK (acknowledgement) – подтверждение, в случае если связь между узлами установлена.

DSR (Dynamic Source Routing Protocol) – протокол динамической маршрутизации от источника.

AODV (Ad hoc On-Demand Distance Vector) – протокол маршрутизации для мобильных ad-hoc сетей.

OLSR (*Optimized Link-State Routing*) – *проактивный протокол маршрутизации.*

IDE (Integrated Development Environment) – интегрированная среда разработки.

P2P (Peer-To-Peer) – одноранговое соединение.

API (Application Programming Interface) – программный интерфейс приложения.

SDK (Software Development Kit) – комплект средств разработки.

AVD (Android Virtual Device) – интерактивный эмулятор мобильного устройства Android.

DSL (Domain Specific Language) – предметно-ориентированный язык.

ADT (Android Development Tools) – *инструменты разработки для Android приложений.*

WPA (Wi-Fi Protected Access) – программа сертификации беспроводных средств связи.

Оглавление

Введение.....	11
1 Аналитический обзор.....	12
1.1 Анализ аналогов программного продукта.....	12
1.2 Анализ протоколов маршрутизации	14
1.3 Сравнительный анализ протоколов маршрутизации	22
2 Проектирование алгоритмов маршрутизации.....	24
2.1 Алгоритм поиска маршрута	24
2.2 Алгоритм передачи данных	28
2.3 Алгоритм фрагментации больших пакетов.....	31
3 Программная реализация.....	32
3.1 Анализ мобильных операционных систем	32
3.1.1 Операционная система Android	33
3.1.2 Операционная система iOS	33
3.1.3 Выводы по анализу представленных мобильных ОС	34
3.2 Анализ программных средств разработки мобильных приложений	34
3.3 Разработка модуля поиска соседних узлов	38
3.4 Разработка модуля построения списка абонентов.....	46
3.5 Разработка модуля передачи данных	47
4 Тестирование приложения на мобильном устройстве	50
5 Финансовый менеджмент, ресурсоэффективность и ресурсосбережение.....	56
5.1 Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения.....	56
5.2 Организация и планирование работ	56
5.3 Расчёт сметы затрат на выполнение проекта	63
6 Социальная ответственность	74
6.1 Производственная безопасность на стадии разработки системы	74
6.2 Вредные производственные факторы	75
6.3 Опасные производственные факторы	80
6.4 Экологическая безопасность.....	82
6.5 Безопасность в чрезвычайных ситуациях.....	84

6.6 Правовые и организационные вопросы обеспечения безопасности	88
6.7 Влияние реализации системы на конечных пользователей.....	91
Список публикаций.....	95
Список использованных источников	96
Приложение А	100

Введение

В настоящее время информатизация общества достигла высоких результатов и этому в большей степени поспособствовали мобильные технологии. Одной из важнейших функций современного гаджета является передача или получение графической и текстовой информации, а так же файлов мультимедии. Данная функция делает мобильные устройства зависимыми от интернета, и в связи с этим ставится вопрос о передаче данных без наличия точек доступа подключения к глобальной сети.

Одной из относительно новых технологий для передачи данных без маршрутизаторов является Wi-Fi Direct.

Wi-Fi Direct – стандарт (набор программных протоколов), позволяющих двум и более Wi-Fi устройствам общаться друг с другом без маршрутизаторов и хот-спотов [1].

С помощью мобильного приложения и наличия в мобильном устройстве технологии Wi-Fi Direct, которая на данный момент присутствует практически на каждом смартфоне, появляется возможность передавать данные по каналам связи, которые создаются между пользователями самого приложения.

Целью дипломной работы является разработка алгоритмического программного обеспечения для передачи данных, используя технологию Wi-Fi Direct, опираясь на протоколы и алгоритмы технологии Manet.

Для достижения цели дипломной работы были поставлены следующие задачи:

1. Найти и проанализировать существующие аналоги.
2. Проанализировать существующие протоколы маршрутизации в сетях Manet.
3. Спроектировать алгоритмы для передачи данных с помощью технологии Wi-Fi Direct.
4. Программно реализовать спроектированные алгоритмы.
5. Протестировать полученное программное обеспечение.

1 Аналитический обзор

1.1 Анализ аналогов программного продукта

На данный момент на рынке мобильных приложений не так много программных продуктов, которые работают с технологией Wi-Fi Direct, и, в основном, реализуют лишь пользовательский интерфейс для передачи данных для двух мобильных устройств. Существуют приложения, которые и расширяют возможности пользователя и предлагают дополнительную информацию о Wi-Fi сигнале, позволяя выбрать точку с наилучшей пропускной способностью. Из основных аналогов программных продуктов, которые работают под операционной системой Android и являются продуктами сервиса «Google Play Market» хотелось бы выделить: Swift-Wi-Fi и SuperBeam.

1.1.1 Swift Wi-Fi

Swift Wi-Fi – приложение разработанное группой разработчиков «Swift Wi-Fi Team». В этом приложении кроме подключения устройств с помощью Wi-Fi Direct не малую роль играет аппаратная часть в виде дополнительных точек доступа, контроллеров и антенн. Так же включает в себя виджет, который показывает, как доступные сети WiFi, так и данные о них.

Виджет позволяет выполнить следующие функции:

1. Переключение Wi-Fi с помощью кнопки ВКЛ / ВЫКЛ;
2. Подключение к нужной Wi-Fi точки доступа;
3. При нажатии на точку доступа, показывает безопасность точки доступа, скорость соединения, силу сигнала и IP-адрес;
4. Управление Wi-Fi точкой доступа (Удалить неиспользуемые точки доступа);
5. Добавить новое Wi-Fi соединение (поддерживает Open, WEP, WPA/WPA2 PSK и 802.1x EAP);
6. Легкая настройка с помощью QRCode (Права Root обязательны).

1.1.2 SuperBeam | Wi-Fi Direct

SuperBeam – приложение, разработанное группой разработчиков «LiveQoS». позволяет просто передавать данные между устройствами Android, используя технологию Wi-Fi Direct. Дает возможность связать между собой два устройства, используя QR-коды (с помощью имеющегося в программе сканера QR-кодов) или NFC, если устройства её поддерживают, или передавая ключ вручную.

Основными функциями приложения SuperBeam являются:

- Связывание устройств с использованием NFC или сканирования QR-кода;
- Передача данных через web-интерфейс на устройства, на которых не установлен SuperBeam;
- Передача одного или нескольких файлов любого типа (фотографии, видео, zip-файлы, APK, контакты и др);
- История всех выполненных операций передачи;
- Автоматическое переключение в режим точки доступа при работе с устройствами, которые не поддерживают WiFi direct;
- При сбоях в работе WiFi direct может быть использовано уже имеющееся сетевое WiFi-соединение между устройствами;
- Для большей совместимости с ограниченными WiFi-сетями для всех видов передачи используется порт 8080;
- Работает со всеми сторонними файловыми менеджерами и фотогалереями.

Из вышесказанного, можно сделать вывод, что приложения, которые используют WiFi-Direct в качестве основной функции, передают данные лишь между двумя устройствами и ничем друг от друга не отличаются, за исключением различного интерфейса. Создаваемое приложение, будет основано на алгоритмах сетей Manet, с дальнейшей возможностью модификации до передачи данных между двумя и более устройствами.

1.2 Анализ протоколов маршрутизации

Wi-Fi Direct относительно новая технология для передачи данных, но у нее так же существуют и конкурент – технология MANET.

MANET (Mobile Ad hoc NETworks) сети – радиосети со случайными мобильными абонентами, реализующие полностью децентрализованное управление при отсутствии базовых станций или опорных узлов. Топология – быстро меняющаяся со случайным соединением узлов.

Сегодня подавляющее большинство наземных мобильных беспроводных сетей связи имеют фиксированную инфраструктуру и соединены между собой с помощью различных, как правило проводных или радиорелейных, каналов передачи данных. В последнее десятилетие большое внимание уделяется созданию мобильных пакетных радиосетей, которые не имеют фиксированной инфраструктуры – сети стационарных (Ad Hoc) и мобильных абонентов (MANET).

Такие сети являются самоорганизующимися, поскольку их узлы являются не только конечными пользовательскими терминалами, но и являются ретрансляторами-маршрутизаторами, ретранслируя пакеты других абонентов и участвуя в нахождении маршрутов к ним, следовательно, эти сети способны к самоорганизации. Такие сети могут состоять из десятков, сотен и даже тысяч узлов.

В противоположность сетям с иерархической структурой и централизованным управлением, одноранговые сети без инфраструктуры состоят из однотипных узлов, где каждый узел обладает комплексом программно-аппаратных средств. Это позволяет организовать передачу данных от источника к получателю напрямую при физическом наличии такого пути и тем самым распределить нагрузку на сеть и повысить суммарную пропускную способность сети. Передача данных от одного абонента к другому может происходить, даже в случае если эти узлы находятся вне зоны прямой радиовидимости. В этих случаях пакеты данных этих абонентов ретранслируются другими узлами сети, которые имеют связь с

корреспондирующими абонентами. Сети с многократной ретрансляцией называются многопролетными или многоскачковыми (multihop). При разработке таких сетей основными проблемами являются маршрутизация пакетов от узла источника к узлу получателю, масштабируемость сетей, адресация конечных устройств, поддержание связности в условиях переменной топологии.

Технология MANET базируется на большом количестве протоколов маршрутизации со сложными алгоритмами для любых ситуаций. К таким протоколам относятся: DSR, AODV, OLSR. Каждый из этих протоколов имеет своей алгоритм поиска маршрута, передачи данных, а так же собственную структуру данных.

Не смотря на то, что технология MANET ещё не используется, и все протоколы находятся в экспериментальном состоянии, они могут быть частично использованы и адаптированы под разрабатываемую систему.

1.2.1 Протокол DSR

Протокол DSR (Dynamic Source Routing Protocol) является простым и эффективным протоколом маршрутизации и предназначен для работы с беспроводными устройствами.

Протокол DSR регулярно обновляет свой кэш маршрутов, если в сети появились новые доступные простые маршруты. В случае, если были найдены новые доступные маршруты, узел направляет пакет на этот маршрут. Пакет должен знать о направлении маршрута. Для этого информация о маршруте указывается в пакете, чтобы достичь адресата от его источника. DSR имеет две основных функции для правильной работы: запрос маршрута и обслуживание маршрута. При открытии маршрута, он имеет два сообщения, запрос маршрута (RREQ) и ответ на запрос маршрута (RREP). Когда узел хочет послать сообщение в определенный пункт назначения, он транслирует RREQ пакет в сети. Соседние узлы в диапазоне вещания получают это RREQ-сообщение и добавляют свой собственный адрес, и снова ретранслируют его в

сети. В случае если RREQ-сообщение не доходит до узла-адресата, маршрут удаляется.

Каждый узел поддерживает свой кэш маршрута, который хранится в памяти для обнаруженного маршрута. Узел будет проверять свой кэш маршрута для требуемого узла назначения до ретрансляции RREQ-сообщения. Поддерживая кэш маршрута на каждом узле в сети, снижаются накладные расходы памяти, которые генерируются с помощью процедуры обнаружения маршрута. Если маршрут находится в этом кэше маршрутов узла, то он не будет ретранслировать RREQ во всей сети. RREQ-сообщение будет сразу транслироваться к узлу назначения. Первое сообщение, достигшее адресата, имеет полную информацию о маршруте. Этот узел будет посылать пакет RREP к источнику, имеющий полную информацию о маршруте. Этот маршрут будет считаться самым коротким путем, по которому пакет RREQ транслировался. Узел-источник теперь имеет полную информацию о маршруте в своем кэше маршрутов и может начинать передачу пакетов [4].

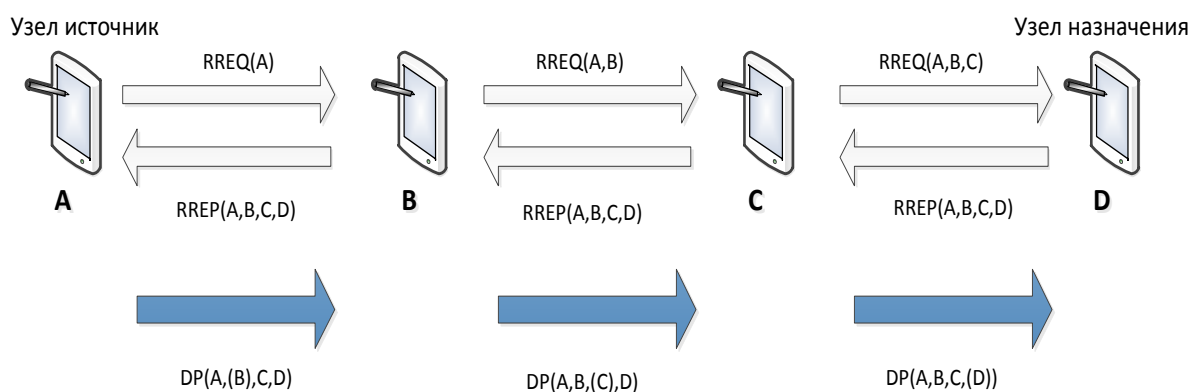


Рисунок 1 – Поиск маршрута по протоколу DSR

На рисунке 1 показана процедура поиска маршрута. Здесь мы имеем четыре узла, A, B, C и D, узел A является источником, узел D является назначения. Когда узел A желает послать пакет данных в узел D, он сначала проверит свой кэш маршрута, имеет ли он прямой маршрут к узлу D или нет. Если узел A не имеет прямой маршрут к узлу D, то он будет транслировать

сообщение RREQ в сети. Соседний узел В получит сообщение RREQ. Узел В проверяет свой кэш маршрута, есть ли прямой маршрут к узлу назначения D или нет, если он находит маршрут к узлу назначения D, он посылает RREP сообщение к узлу А.

В ответ на это сообщение узел источник начнет посылать пакеты данных (DP) по обнаруженному маршруту. Если он не обнаружил маршрут от узла В к узлу D, он передает сообщение RREQ к следующему узлу С и сохраняет маршрут А,В в кэше. Процесс продолжается до тех пор, пока сообщение RREQ не достигнет целевого узла D. Узел назначения D кэширует маршруты АВ, ВС и CD в своей памяти и посылает сообщение RREP к исходному узлу А.

Следующей функцией является обслуживание маршрута. Обслуживание маршрута использует два вида сообщений: сообщение об ошибке (RERR) и подтверждение (ACK). ACK отправляет сообщение узлу-источнику, если узел назначения был достигнут. Все промежуточные узлы так же получают информацию о новом построенном маршруте. Если в сети произошла ошибка в процессе поиска маршрута, узлу-источнику приходит сообщение об ошибке (RERR). Это происходит, если узел-источник не смог получить ACK-сообщение.

Таким образом, источник получает пакет RERR для того, чтобы вновь отправить запрос на поиск маршрута. Получив сообщение RERR узлы удаляют записи маршрута.

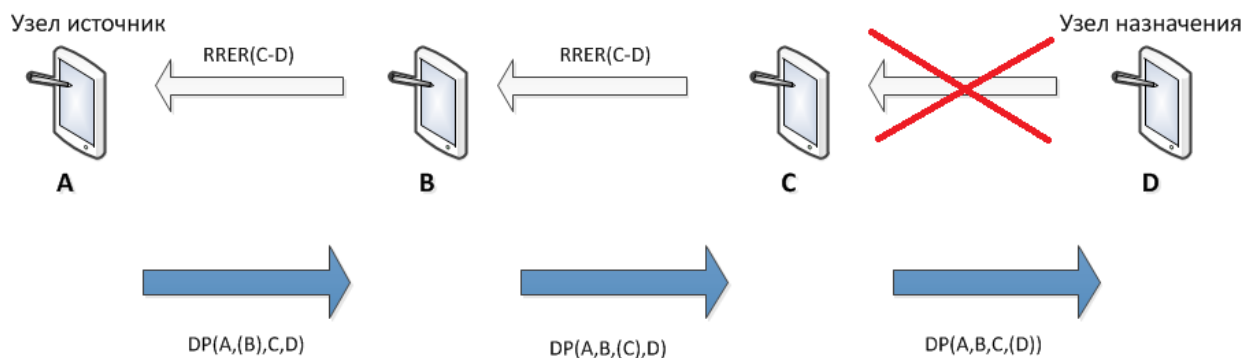


Рисунок 2 – Сообщения о подтверждении и ошибке в протоколе DSR

На рисунке 2 показаны четыре узла т.е. А, В, С и D.

Узел А посылает сообщение к узлу назначения D. сообщение проходит до узла С. Когда узел С пересылает сообщение RREQ по отношению к узлу D, и не получает сообщение о подтверждении ACK от узла D, узел С понимает, что появились проблемы при передаче.

В следствии чего, узел С посылает сообщение RRER к исходному узлу А, что позволяет узлу А начать новый поиск маршрута до узла D [5].

1.2.2 Протокол AODV

AODV (Ad hoc On-Demand Distance Vector) является протоколом маршрутизации по требованию. Алгоритм AODV дает простой способ получить информацию об изменении ситуации при отправке пакетов данных. Таким образом, если отправка пакета не удалась, выявляется пораженный узел и всем пострадавшим от этого узлам отправляется уведомление. Это уведомление отменяет все маршруты через пораженный узел. В AODV маршруты строятся однонаправленные, от источника к месту назначения, и поэтому использование сети является минимальным. Так как маршруты строятся по требованию, сетевой трафик так же будет минимальным. AODV не позволяет хранить дополнительные маршруты, которые не используются. Если два узла хотят установить соединение в одноранговой сети, то AODV позволяет им построить многоинтервальный маршрут. AODV использует порядковый номер узла-получателя (DSN), чтобы избежать подсчета до бесконечности [6].

Когда узел-источник отправляет запрос до пункта назначения, он посылает свои DSNs вместе со всей информацией о маршрутизации. Он также выбирает наиболее выгодный маршрут, основываясь на количестве DSN.

Существует три AODV-сообщения: запрос маршрута (RREQs), ответ на запрос маршрута (RREPs), и ошибка маршрута (RERRs).

Когда узел-источник хочет создать новый маршрут к месту назначения, запрашивающий узел транслирует RREQ-сообщение ближайшим соседям.

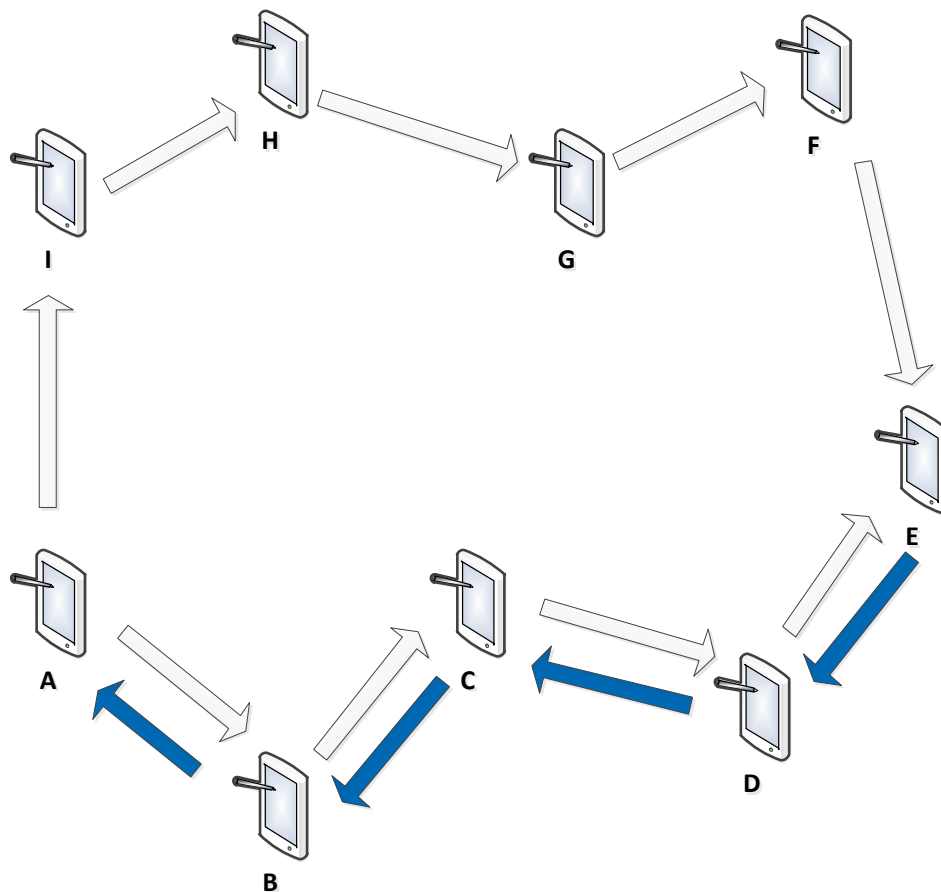


Рисунок 3 – Поиск маршрута в протоколе AODV

На рисунке 3 сообщение RREQ транслируется из узла-источника А к узлу назначения Е. Сообщение RREQ показано белой стрелкой от узла-источника А по двум направлениям. Исходный узел А транслирует сообщение RREQ соседним узлам.

Каждый добавленный узел на пути к узлу назначения увеличивается значение DSN на единицу. Каждый промежуточный узел будет проверять имеет ли он прямой маршрут к месту назначения или нет. Если у него есть маршрут, он направит RREP к источнику узла А. Если он не имеет прямого маршрута к месту назначения, он будет транслировать сообщение RREQ в сети снова, с измененным значением порядкового номера DSN.

На рисунке 3 показана процедура для нахождения узла назначения Е. сообщение RREQ транслируется в сети для нахождения узла назначения. Промежуточные узлы могут ответить на сообщение RREQ, только если они имеют число DSN, равную или больше, чем количество содержащегося в

заголовке пакета из RREQ. Промежуточные узлы пересылают сообщение RREQ к соседним узлам и записывают адреса этих узлов в кэше маршрутизации. Эта информация будет использоваться, чтобы сделать обратный путь для RREP сообщения от узла назначения, как показано на рисунке 3.

Узел назначения E отвечает сообщением RREP, обозначенным синими стрелками, который обозначает самый короткий путь от узла назначения E к узлу-источнику A. Узлы, которые получают RREP сообщение кэшируют информацию о построенном маршруте.

Если поиска маршрута не дал результата, генерируется сообщение RERR. RERR сообщение содержит информацию об узлах, которые не достижимы.

Вся информация о маршрутизации в сети хранится в таблице. Таблица маршрутизации предоставлена в таблице 1: IP-адрес назначения, порядковый номер (DSN), последовательность узлов до узла назначения, сетевой интерфейс, количество переходов, тайм аут [7].

Таблица 1 – таблица маршрутизации

IP-адрес назначения	DSN	Последовательность узлов до узла назначения	Сетевой интерфейс	Количество переходов	Тайм-аут
---------------------	-----	---	-------------------	----------------------	----------

1.2.3 Протокол OLSR

OLSR (*Optimized Link-State Routing*) это проактивный протокол маршрутизации. OLSR так же называют табличным протоколом, так как он постоянно хранит и обновляет свою таблицу маршрутизации.

OLSR отслеживает таблицы маршрутизации, для того чтобы обеспечить оптимальный маршрут, при отправке запроса на передачу данных к узлу назначения. Благодаря своей природе OLSR называется проактивным протоколом маршрутизации. Multipoint Relay (MPR) узлы показаны на рисунке 4.

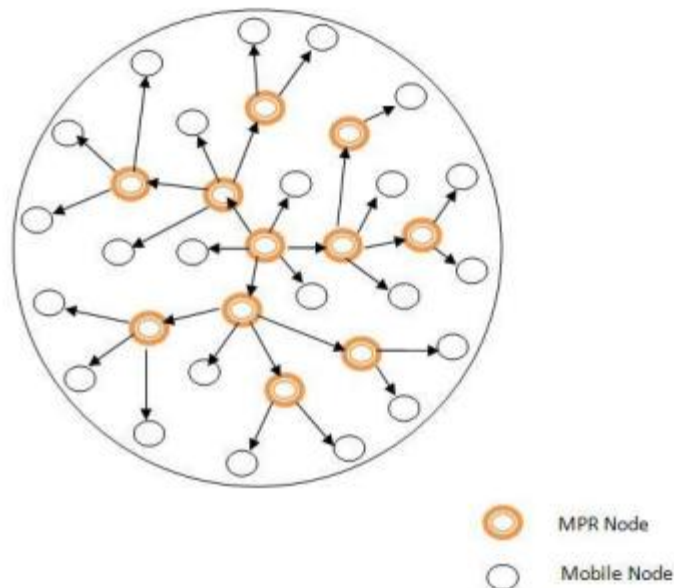


Рисунок 4 – Multipoint Relay в OLSR

В OLSR не все узлы обмениваются информацией друг с другом. Multipoint Relay (MPR) узлы предоставляют широковещательную передачу пакетов с маршрутами. Узел MPR выбирается в качестве соседа для узла-источника. Каждый узел в сети хранит список узлов MPR.

Селектор MPR получается из отправки пакетов «Hello» между соседними узлами. Эти маршруты построены до того, как исходный узел намеревается направить сообщение по указанному назначению. Каждый узел в сети хранит таблицу маршрутизации. Именно по этой причине маршрутизации накладные расходы на OLSR минимальны по сравнению с другими протоколами реактивной маршрутизации и обеспечивают кратчайший маршрут к месту назначения в сети. Нет необходимости строить новые маршруты, так как существующие маршруты имеют запасные. Это уменьшает задержку обнаружения нового маршрута. Узлы в сети отправляют сообщения «Hello» своим соседям. Эти сообщения отправляются с заданным интервалом в OLSR для определения состояния канала связи.

Если узел А и узел В являются соседями, узел А посылает сообщение «Hello» в узел В. Если узел В принимает это сообщение, мы можем сказать, что связь является асимметричной. Если узел В отправляет то же сообщение «Hello» к узлу, то это то же самое, как в первом случае, называемом

асимметричным запросом. Исходя из этого, устанавливается двухсторонняя связь и мы можем назвать эту связь симметричной. Сообщение «Hello» содержит всю информацию о соседях. Это позволяет любому узлу иметь таблицу, в которой он имеет информацию обо всех своих многочисленных соседях. Узел выбирает минимальное количество узлов MPR, когда симметричная связь установлена.

1.3 Сравнительный анализ протоколов маршрутизации

Сравнительный анализ основных характеристик реактивных протоколов представлен в таблице 2.

Таблица 2 – сравнение реактивных протоколов

Имя	Маршрутная структура	Множественные маршруты	Метод расчета маршрута	Хранение маршрутов	Стратегия
AODV	Плоская	Нет	Новый кратчайший путь	Маршрутная таблица	Полная либо частичная перезапись маршрутов
DSR	Плоская	Да	Кратчайший путь или следующий в маршрутной таблице	Кэш маршрутов	Полная перезапись маршрутов

Сравнительный анализ сложности протоколов маршрутизации представлен в таблице 3:

Таблица 3 – сложность проактивных и реактивных протоколов

Имя	Время конвергенции	Память	Управление	Преимущества \ недостатки
OLSR	$O(D \cdot I)$	$O(N^2)$	$O(N^2)$	Защита от петель на основании предшествующей информации / необходимость обмена данными
AODV	$O(2D)$	$O(2N)$	$O(2N)$	Адаптируется к высокодинамичным топологиям
DSR	$O(2D)$	$O(2N)$	$O(2N)$	Множество маршрутов, невозможно прослушать

где

$O()$ – порядок сложности;

D – диаметр сети;

I – среднее время обновления;

N – количество узлов в сети.

Сравнение преимуществ и недостатков протоколов различного типа представлено в таблице 4.

Таблица 4 – преимущества и недостатки протоколов OLSR и AODV

OLSR	AODV
Проактивный протокол	Реактивный протокол
хранит новейшие данные о всей сети	хранит данные только про активные пути, за счет чего меньше обработка служебной информации
время нахождения нового пути меньше, так как необходимо только просмотреть маршрутную таблицу, которая постоянно обновляется	время нахождения нового пути больше, так как информация имеется только об активных маршрутах
более эффективен для сетей с меньшей мобильностью	более эффективен для сетей с высокой мобильностью

В результате сравнения, наиболее оптимальным протоколом является DSR. Он включает в себя простое хранение большого количества маршрутов, может быть применен для сетей с высокой мобильностью. DSR может быть использован как основа для разработки алгоритмов для передачи данных с помощью технологии Wi-Fi Direct.

2 Проектирование алгоритмов маршрутизации

2.1 Алгоритм поиска маршрута

Алгоритм поиска маршрута позволяет собрать необходимую информацию для построения маршрута между источником и получателем.

Маршрут может быть найден двумя способами:

- может быть найден в кэше маршрутизации, если данный маршрут строится не первый раз и данные маршруте не успели устареть;
- может быть найден с помощью запроса на поиск маршрута, отправленного ближайшим соседям и который будет ретранслироваться от соседа к соседу до тех пор, пока маршрут не будет найден или пока не истечет время жизни запроса;

Если маршрут находится в кэше маршрутизации промежуточного узла, то запрос на поиск маршрута не будет ретранслирован по всей сети. Первое сообщение достигшее адресата имеет полную информацию о маршруте. Автоматически данный маршрут будет считаться самым коротким и выгодным для передачи данных.

Если маршрута нет в кэше маршрутизации, необходимо его построить с помощью рассылки запроса на поиск маршрута всем ближайшим соседям.

Если соседние узлы не могут найти в своей зоне видимости узел-получатель, то каждый узел ретранслирует запрос своим ближайшим соседям и так продолжается до тех пор, пока маршрут не будет найден, либо пока не истечет время жизни пакета.

Каждый узел отдельно обрабатывает запрос, и возвращает узлу-источнику либо информацию о пути, либо о том, что маршрут построить не удалось. В связи с этим появляется возможность нахождения нескольких маршрутов через разные узлы, что позволит записать в кэш маршрутизации несколько альтернативных узлов. Наиболее приоритетным будет маршрут, в котором наименьшее количество промежуточных узлов.

Блок схема алгоритма поиска маршрута предоставлена на рисунке 5.

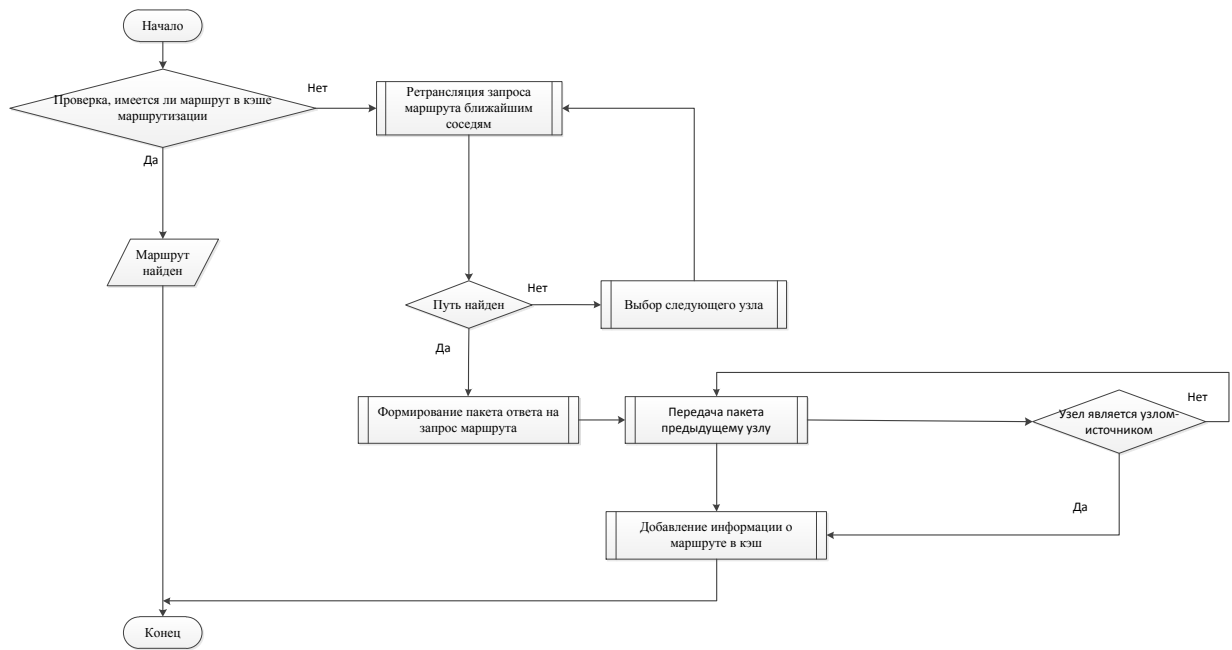


Рисунок 5 – Блок-схема алгоритма поиска маршрута

Информация, которую несет пакет запроса маршрута, показана в таблице 5.

Таблица 5 – структура данных пакета запроса маршрута

1 байт	16 байт	16 байт	16 байт	8 байт	8 байт	16*N байт
Тип запроса	UID	Узел-источник	Узел - получатель	Тайм-аут	Hops	MAC-адреса промежуточных узлов

Поле «Тип запроса» содержит бинарное число, соответствующее типу пакета. Для пакета запроса маршрута по умолчанию 0001;

Поле «UID» содержит уникальный идентификатор созданного запроса;

Поле «Узел-источник» содержит информация об узле инициировавшем запрос поиска маршрута;

Поле «Узел-получатель» содержит информацию об узле, маршрут к которому необходимо построить;

Поле «Тайм-аут» содержит время жизни пакета. При создании запроса «Тайм-аут» по умолчанию равен 30 секундам. При каждом увеличении поля «Hops», время соответственно увеличивается на 30 секунд. Если время истекло, пакет удаляется;

Поле «Hops» содержит бинарное число, по умолчанию 0000. При каждой передаче пакета на другой узел, значение увеличивается на 0001;

В поле «MAC-адреса промежуточных узлов» каждый промежуточный узел добавляет свой MAC-адрес.

По достижению узла-получателя, формируется ответ на запрос маршрута, используя конечный вариант пакета запроса маршрута.

Информация, которую несет пакет ответа на запрос маршрута, представлена в таблице 6.

Таблица 6 – структура данных пакета ответа на запрос маршрута

1 байт	16 байт	16 байт	16 байт	8 байт	8 байт	16*N байт
Тип запроса	UID	Узел-источник	Узел - получатель	Тайм-аут	Hops	MAC-адреса промежуточных узлов

Поле «Тип запроса» содержит бинарное число, соответствующее типу пакета. Для пакета ответа на запрос маршрута по умолчанию 0010;

Поле «UID» содержит уникальный идентификатор запроса;

Поле «Узел-источник» содержит информация об узле, куда необходимо доставить пакет;

Поле «Узел-получатель» содержит информацию об узле, который транслирует ответ на запрос маршрута;

Поле «Тайм-аут» содержит время жизни пакета. По умолчанию «Тайм-аут» равен 30*Hops секунд;

Поле «Hops» содержит бинарное число, которое определяет количество промежуточных узлов;

В поле «MAC-адреса промежуточных узлов» хранятся все MAC-адреса промежуточных узлов.

Когда пакет ответа на запрос маршрута достигает узел-источник, узел записывает информацию из пакета в свой кэш маршрутизации. Кроме того, каждый промежуточный узел, через который прошел пакет, так же сохраняет информацию о маршруте в свой кэш маршрутизации.

Информация в кэше маршрутизация хранится в двух таблицах:

- в первой таблице хранится вся основная информация о маршруте, кроме промежуточных узлов;

- промежуточные узлы хранятся во второй таблице, и она связана с первой первичным ключом.

Структура кэша маршрутизации представлена в таблице 7.

Таблица 7 – структура данных кэша маршрутизации

16 байт	16 байт	16 байт	8 байт	1 байт
UID	Узел-источник	Узел - получатель	Тайм-аут	Hops

Поле «UID» содержит уникальный идентификатор построенного маршрута;

Поле «Узел-источник» содержит информация об узле, который инициировал построение маршрута;

Поле «Узел-получатель» содержит информацию об узле, маршрут к которому был построен;

Поле «Тайм-аут» содержит время жизни маршрута. По умолчанию равен 300 секунд;

Поле «Hops» содержит бинарное число, равное количеству промежуточных узлов в данном маршруте.

Информация о промежуточных узлах представлена в таблице 8.

Таблица 8 – таблица хранения информации о промежуточных узлах.

16 байт	16 байт
UID	MAC-адрес промежуточного узла
1e2r31we	1.2.3.4
1e2r31we	1.2.3.5
1gd34gas	1.2.3.4

Поле «UID» содержит уникальный идентификатор построенного маршрута;

В поле «MAC-адрес промежуточного узла» содержится последовательная запись всех промежуточных узлов. Каждому промежуточному узлу присваивается UID маршрута, в котором он находится.

После того, как узел-источник получил запрашиваемую информацию, он может приступить к передаче данных [8].

2.2 Алгоритм передачи данных

Для передачи данных используется построенный маршрут между узлом-источником и узлом-получателем. Данные передаются последовательно через каждый промежуточный узел. После того, как пакет с данными достиг получателя, получатель отправляет пакет подтверждения об успешной передаче пакета с данными.

В случае, если в процессе передачи данных один из промежуточных узлов стал недоступен, узел, который не смог передать данные по построенному маршруту отправляет пакет с ошибкой, в котором указан недоступный узел. В данной ситуации узел, инициировавший передачу данных начинает строить новый маршрут до получателя, если в кэше маршрутизации не был найден альтернативный маршрут.

Блок схема алгоритма передачи данных представлена на рисунке 6.



Рисунок 6 – блок схема алгоритма передачи данных

Структура пакета передачи данных представлена в таблице 9.

Таблица 9 – структура пакета передачи данных

1 байт	1 байт	16 байт	16 байт	16*N байт	До 65535 байт	1 байт	16 байт
Тип запроса	ID пакета	Узел-источник	Узел-получатель	MAC-адреса промежуточных узлов	Данные	Последний фрагмент	Размер окна

Поле «Тип запроса» содержит бинарное число, соответствующее типу пакета. Для пакета передачи данных по умолчанию 0011;

Поле «ID пакета» содержит бинарное число, идентифицирующее номер пакета, для составления пакета подтверждения или ошибки доставки пакета;

Поле «Узел-источник» содержит информацию об узле, инициировавшем передачу данных;

Поле «Узел-получатель» содержит информацию об узле, куда данные должны быть доставлены;

В поле «MAC-адреса промежуточных узлов» последовательно записаны промежуточные узлы;

В поле «Данные» находится передаваемая информация;

В поле «Контрольная сумма» содержит контрольную сумму подсчитанную для всего пакета целиком;

В поле «Размер окна» содержится количество байт информации, которое готов принять в свои внутренние буфера инициатор передачи данных.

Первый пакет, который отправляет источник, является пустым, чтобы подтвердить то, что маршрут существует, и какое количество данных он может принять за раз: «Размер окна».

Идентификация типа пакета происходит по полю «Тип запроса». Идентифицируя тип пакета, распознается ID пакета, и узел-источник определяет, какой пакет был доставлен.

Структура пакета подтверждения о передаче данных представлена в таблице 10.

Таблица 10 – структура пакета подтверждения о передаче данных

1 байт	1 байт	16 байт	16 байт	1 байт	16*N байт	16 байт
Тип запроса	ID пакета	Узел-источник	Узел-получатель	Hops	MAC-адреса промежуточных узлов	Размер окна

Поле «Тип запроса» содержит бинарное число, соответствующее типу пакета. Для пакета подтверждения о передачи данных по умолчанию 0101;

Поле «ID пакета» содержит бинарное число, идентифицирующее номер пакета, который был доставлен;

Поле «Узел-источник» содержит информацию об узле, которому необходимо доставить пакет подтверждения;

Поле «Узел-получатель» содержит информацию об узле, куда данные были доставлены;

Поле «Hops» содержит бинарное число, равное количеству промежуточных узлов между источником и получателем;

В поле «MAC-адреса промежуточных узлов» последовательно записаны промежуточные узлы;

В поле «Размер окна» содержится количество байт информации, которое готов принять в свои внутренние буфера узел-получатель.

Структура пакет об ошибке передачи данных представлена в таблице 11.

Таблица 11 – структура пакета об ошибке передачи данных

1 байт	1 байт	16 байт	16 байт	1 байт	16*N байт	16 байт
Тип запроса	ID пакета	Узел-источник	Узел-получатель	Hops	MAC-адреса промежуточных узлов	MAC-адрес узла, на котором произошел сбой

Поле «Тип запроса» содержит бинарное число, соответствующее типу пакета. Для пакета об ошибке передачи данных по умолчанию 0110;

Поле «ID пакета» содержит бинарное число, идентифицирующее номер пакета, который не был доставлен;

Поле «Узел-источник» содержит информацию об узле, куда необходимо доставить пакет об ошибке передачи данных;

Поле «Узел-получатель» содержит информацию об узле, маршрут до которого был нарушен;

Поле «Hops» содержит бинарное число, равное количеству промежуточных узлов между источником и получателем;

В поле «MAC-адреса промежуточных узлов» последовательно записаны промежуточные узлы;

В поле «MAC-адрес узла, на котором произошел сбой» содержится информация об узле, до которого пакет передачи данных отправить не удалось;

2.3 Алгоритм фрагментации больших пакетов

Для передачи больших данных необходимо разбивать их на пакеты и передавать в виде дейтаграмм, длина которых может в итоге сможет превысить максимально возможное количество данных, которые может принять абонент. За счет фрагментирования – разбиения «большого пакета» на некоторое количество частей (фрагментов), размер каждой части удовлетворит требования абонента. После того, как все фрагменты будут переданы через промежуточную сеть, они будут собраны на узле-получателе модулем «сборки в большой пакет». Сборку пакета из фрагментов осуществляет только получатель, а не какой-либо из абонентов. Промежуточные абоненты могут только фрагментировать пакеты, но не собирать их.

Для того, чтобы не перепутать фрагменты разных пакетов, используется поле «ID пакета», значение которого должно быть одинаковым для всех фрагментов одного пакета и не повторяться для разных пакетов, пока у обоих пакетов не истекло время жизни. При делении данных пакета, размер всех фрагментов, кроме последнего, должен быть кратен 8 байтам. Это позволяет отвести меньше места в заголовке под поле Смещение фрагмента.

Второй бит поля Флаги (Последний фрагмент), если равен единице, указывает на то, что данный фрагмент – не последний в пакете. Если пакет отправляется без фрагментации, флаг «Последний фрагмент» устанавливается в 0.

В результате проектирования были получены:

- алгоритм поиска маршрута;
- алгоритм передачи данных;
- алгоритм фрагментации данных.

Кроме того, были спроектированы структуры данных:

- структура пакета данных запроса маршрута;
- структура пакета данных ответа на запрос;
- структура данных в кэше;
- структура пакета данных о промежуточных узлах;
- структура пакета передачи данных;
- структура пакета данных подтверждения о передачи данных;
- структура пакета данных об ошибке передачи данных

3 Программная реализация

3.1 Анализ мобильных операционных систем

На данный момент на рынке современных мобильных устройств существуют десятки мобильных операционных систем, но самыми популярными на данный момент являются iOS, Android, Symbian и Blackberry. На таблице 12 показано распределение мобильных операционных систем на 2013-2014 год.

Таблица 12 – распределение популярности мобильных ОС

Название операционной системы	2013 г.	2014 г.
Android	75%	48,8%
iOS	17,3%	19,1%
Blackberry	2,9%	10,5%
Прочие	4,2%	5,2%

Не смотря на то, что OS Android утратила значительную долю на рынке, она остается явным лидером мирового рынка мобильных операционных систем в целом. Далее из наиболее стабильных ОС можно выделить только iOS.

Проведем сравнительный анализ Android и iOS систем, по функциональным характеристикам, имеющим значение в разработке данной системы.

Каждая из систем имеет как ряд существенных преимуществ, так и недостатков. Далее приведены основные, явно выделенные плюсы и минусы каждой системы.

3.1.1 Операционная система Android

Достоинства операционной системы Android:

- простота и гибкость пользовательского интерфейса;
- открытость операционной системы;
- не высокая цена на мобильные устройства;
- высокая доля на рынке;
- легкая и бесплатная публикация приложений.

Недостатки операционной системы Android:

- высокое энергопотребление;
- низкая защищенность.

3.1.2 Операционная система iOS

Достоинства операционной системы iOS:

- простота пользовательского интерфейса;
- низкое энергопотребление;
- высокая защищенность;
- наиболее быстрый отклик системы при работе с внешним интерфейсом;

Недостатки операционной системы iOS:

- высокая цена на мобильные устройства.

3.1.3 Выводы по анализу представленных мобильных ОС

Проанализировав преимущества и недостатки данных операционных систем, можно сделать вывод, что оптимальным вариантом для разработки приложения будет OS Android. Android имеет значительную долю на рынке, доступную цену и лишь немного уступает iOS в плане работы с интерфейсом.

3.2 Анализ программных средств разработки мобильных приложений

3.2.1 Среда разработки Eclipse

Eclipse – свободная интегрированная среда разработки модульных кроссплатформенных приложений. Развивается и поддерживается Eclipse Foundation.

Eclipse является бесплатной программной платформой с открытым исходным кодом, контролируется организацией Eclipse Foundation. Eclipse написана на языке программирования Java и основной целью её создания является повышение продуктивности процесса разработки программного обеспечения.

Претендует на статус наиболее популярной Java IDE.

Особенности платформы Eclipse:

-кроссплатформенность – работает под операционными системами Windows, Linux, Solaris и Mac OS X;

-используя Eclipse можно программировать на множестве языков, таких как Java, C и C++, PHP, Perl, Python, Cobol и других;

-Eclipse является инструментом для создания практически любого клиентского программного обеспечения [9].

3.2.2 Среда разработки Android Studio

Android Studio – интегрированная среда разработки приложений для операционной системы android [10], призванная заменить собой Eclipse. Последняя, на данный момент является самым популярным инструментом в кругу Android разработчиков. Процесс разработки в среде Android Studio стал

ещё более гибким чем у главного конкурента. Это достигается за счет отображения всех рабочих файлов прямо в структуре проекта. Крайне полезной является возможность видеть все визуальные изменения проекта в режиме реального времени. Еще данная SDK дает возможность протестировать работу будущего приложения на разных устройствах. Во встроенном эмуляторе Android девайсов можно установить разнообразные технические конфигурации и разрешения экрана для теста. Кроме того, при использовании эмуляции есть возможность получить информацию о приблизительном уровне производительности для того или иного устройства.

Android SDK – среда разработки приложений для операционной системы Android. Она позволяет создавать и тестировать Android-приложения, использующие камеру мобильного устройства, акселерометр, компас, данные GPS, доступ по Bluetooth, Wi-Fi, EDGE и 3G. Android SDK включает в себя разнообразные библиотеки, документацию и инструменты, которые помогают разрабатывать мобильные приложения для платформы Android.

API Android SDK – API библиотеки (это сокращение от Application Programming Interface) Android, предоставляемые для разработки приложений.

Документация SDK – включает обширную справочную информацию, детализирующую, что включено в каждый пакет и класс и как это использовать при разработке приложений.

AVD (Android Virtual Device)— интерактивный эмулятор мобильного устройства Android. Используя эмулятор, можно запускать и тестировать приложения без использования реального Android устройства.

Development Tools – SDK включает несколько инструментальных средств для разработки, которые позволяют компилировать и отлаживать создаваемые приложения.

Sample Code – Android SDK предоставляет типовые приложения, которые демонстрируют некоторые из возможностей Android, и простые программы, которые показывают, как использовать индивидуальные особенности API в коде.

С другой стороны Android Studio потребляет много памяти и потому, возможны некоторые проблемы с установкой, а так же и работой. Если ПК не соответствует определенным параметрам, скорость работы за Android Studio может быть раза в 3-5 ниже, чем в других средах. Так же, эта среда часто обновляется, порой, обновления бывают через несколько дней, то есть возможно и интерфейс постоянно меняется, что может затруднять работу неопытного разработчика.

К тому же без данного пакета какие-либо разработки просто невозможны, в какой бы среде программирования они ни проводились [11].

3.2.3 Сравнительный анализ Eclipse и Android Studio

Eclipse является наиболее устоявшимся продуктом и пользуется огромной популярностью среди пользователей, которые давно программируют, особенно на языке java. Кроме того, Google предоставил ADT плагин, который превращает популярный IDE в полнофункциональную среду разработки под Android. В связи с этим, люди не спешат переходить в Android Studio.

В связи с тем, что обе среды разработки являются многообещающими и пользуются огромной популярностью, не уступая друг другу, сравним их в 5 разных областях.

3.2.3.1 Системы автоматической сборки проекта

Android Studio использует систему сборки Gradle. Он основывается на Apache Ant и Apache Maven, но он также включает в себя Groovy DSL (Domain-Specific Language), который позволяет сценарию сборки, который открывает много возможностей автоматизации, как загрузка беты .apk в TestFlight для тестирования. Eclipse, с другой стороны, использует Apache Ant в качестве основной системы сборки, которая очень надежна, и люди с ней давно знакомы.

3.2.3.2 Написание кода

Обе среды разработки имеют стандартный код Java, но в случае Android Studio, Google создал наиболее усовершенствованную систему проверки кода. Android Studio может реорганизовать свой код в тех местах, где это просто не возможно с помощью Eclipse, и ADT. Т.е., Android Studio может лучше предсказать то, что хочет сделать пользователь, чем среда Eclipse.

3.2.3.3 Интерфейс среды разработки

Одной из основной особенности созданной Google в Android Studio, является полностью переработанный пользовательский интерфейс. Интерфейс в Android Studio имеет больше вариантов настройки и производится все это с помощью специальных инструментов, что в Eclipse пользователь должен был выбирать вручную в XML.

3.2.3.4 Создание и организация проекта

Обе среды разработки работают по разному при организации проекта. Чтобы использовать Eclipse, то необходимы знания и умения работы в рабочих пространствах. При запуске Eclipse , необходимо выбрать рабочее пространство, в котором хранятся проекты, и затем загрузить его в дерево процессов. Основным минусом данного подхода является то, что для того чтобы переключиться на проект в другом рабочем пространстве, необходимо полностью перезагрузить среду разработки. Android Studio рассматривает эту ситуацию иначе, вводя понятие модулей. При работе с приложением может быть открыт любой модуль, нужная библиотека классов, которую которую загрузили с помощью SDK, а так же могут быть запущены средства интеграции. Каждый из этих модулей может иметь свои собственные Gradle сборки и иметь свои отдельные зависимости.

3.2.3.5 Производительность и стабильность

Eclipse является средой разработки основанной на java. Чтобы среда разработки работала надежно, необходимо приличное количество оперативной памяти и хорошая мощность процессора. Для Eclipse не является

необычным то, что произошел сбой при экспорте или импорте APK. Android Studio является более надежной средой разработки, в ней редко происходят непредвиденные сбои, а если таковые имеются, то пользователю будет показан соответствующий код ошибки. Тем не менее, Android Studio требует высоких вычислительных мощностей, за счет богатого встроенного инструментария [12].

3.3 Разработка модуля поиска соседних узлов

3.3.1 Описание использованного API

Wi-Fi P2P позволяет приложению быстро находить и взаимодействовать с близлежащими устройствами, находящимися за пределами возможностей Bluetooth.

API-интерфейсы Wi-Fi Peer-to-peer (P2P) позволяют приложениям подключаться к соседним устройствам без необходимости подключения к сети или точке и если приложение предназначено для работы в защищенной сети, Wi-Fi Direct является более подходящим вариантом, чем традиционные сети Wi-Fi для сетей ad-hoc по следующим причинам:

Wi-Fi Direct поддерживает шифрование WPA2. (Некоторые сети ad-hoc поддерживают только WEP-шифрование.)

Устройства могут транслировать услуги, которые они предоставляют, что помогает другим устройствам обнаруживать подходящих одноранговых узлов наиболее простым способом.

Кроме того, у технологии Wi-Fi Direct есть встроенная возможность управления питанием, пользовательским интерфейсом и сервисом каждого устройства и эту информацию можно использовать для выбора устройства, которое может наиболее эффективно работать в роли сервера.

API-интерфейсы Wi-Fi P2P состоят из следующих основных частей:

Методы, которые позволяют обнаруживать, запрашивать и подключаться к одноранговым узлам, определены в классе `WifiP2pManager`;

Методы, которые позволяют получать уведомления об успешном или неудачном вызове класса `WifiP2pManager`. При вызове `WifiP2pManager`

каждый метод может получить определенный радиосигнал, переданный в качестве параметра.

Сигналы, которые уведомляют о конкретных событиях, обнаруженных фреймворком Wi-Fi P2P, например, о сброшенном соединении или недавно обнаруженном соседнем узле.

Чаще всего используются три основных компонента API. С помощью *WifiP2pManager.ActionListener* можно вызвать метод *discoverPeers()*, который ищет ближайших соседей, и с помощью методов *ActionListener.onSuccess()* и *ActionListener.onFailure()* получать необходимые уведомления. *WIFI_P2P_PEERS_CHANGED_ACTION* – так же транслируется, если метод *discoverPeers()* обнаруживает, что список соседних узлов изменился.

Класс *WifiP2pManager* предоставляет методы, позволяющие взаимодействовать с аппаратным обеспечением Wi-Fi устройства, чтобы выполнять такие действия, как обнаружение соседних узлов и подключение к ним. Доступны следующие методы:

- initialize()* – регистрирует приложение в фреймворке Wi-Fi. Данный метод необходимо вызывать до вызова любого другого метода Wi-Fi P2P;

- connect()* – запускает одноранговое соединение с устройством с указанной конфигурацией;

- cancelConnect()* – отменяет любые текущие согласования группы одноранговых групп;

- requestConnectInfo()* – запрос информации о подключении устройства;

- createGroup()* – создает одноранговую группу с текущим устройством в качестве владельца группы;

- requestGroupInfo()* – запрос информации группы одноранговой сети;

- discoverPeers()* – инициирует одноранговое обнаружение;

- requestPeers()* – запрашивает текущий список обнаруженных одноранговых узлов.

Методы *WifiP2pManager* позволяют передавать устройству информацию о состоянии подключения, чтобы фреймворк Wi-Fi P2P мог

уведомлять пользователя о статусе вызова. Основные интерфейсы, используемые WiFiP2pManager, показаны в таблице 13:

Таблица 13 – интерфейсы используемые API WiFiP2pManager

Интерфейс	Связанные методы
<i>WifiP2pManager.ActionListener</i>	<i>connect(), cancelConnect(), createGroup(), removeGroup(), discoverPeers()</i>
<i>WifiP2pManager.ChannelListener</i>	<i>initialize()</i>
<i>WifiP2pManager.ConnectionInfoListener</i>	<i>requestConnectInfo()</i>
<i>WifiP2pManager.GroupInfoListener</i>	<i>requestGroupInfo()</i>
<i>WifiP2pManager.PeerListListener</i>	<i>requestPeers()</i>

API-интерфейсы Wi-Fi P2P определяют сигналы, которые передаются при возникновении определенных событий Wi-Fi P2P, например, при обнаружении нового узла или изменении состояния Wi-Fi устройства [13].

Сигналы и их значения описаны в таблице 14:

Таблица 14 – виды и значения сигналов

Сигнал	Описание
<i>WIFI_P2P_CONNECTION_CHANGED_ACTION</i>	Широковещательная передача при изменении состояния подключения Wi-Fi устройства
<i>WIFI_P2P_PEERS_CHANGED_ACTION</i>	Вещание, используется метод <i>discoverPeers()</i> . Обычно отправляется запрос <i>requestPeers()</i> , чтобы получить обновленный список одноранговых узлов
<i>WIFI_P2P_STATE_CHANGED_ACTION</i>	Широковещательная передача при включении или отключении Wi-Fi P2P на устройстве
<i>WIFI_P2P_THIS_DEVICE_CHANGED_ACTION</i>	Сигнал, когда детали устройства изменились, например, имя устройства

3.3.2 Создание широковещательного приемника

Широковещательный приемник позволяет получать сигналы, транслируемые системой Android, чтобы приложение могло реагировать на интересующие пользователя события. Основные шаги для создания широковещательного приемника для управления настройками Wi-Fi P2P заключаются в следующем:

Необходимо создать класс, который наследует класс *BroadcastReceiver*. Для конструктора класса, необходимы параметры класса *WifiP2pManager*, *WifiP2pManager.Channel* и уведомления о том, что широковещательный приемник был зарегистрирован. Это позволяет с помощью широковещательного канала обновлять информацию, а так же задействовать аппаратное обеспечение и канал связи, если это необходимо;

Метод *onReceive()* позволяет указать интересующие нас сигналы и какие действия необходимо выполнить, при получении того или иного сигнала. Например, если широковещательный приемник получает намерение *WIFI_P2P_PEERS_CHANGED_ACTION*, можно вызвать метод *requestPeers()*, чтобы получить список обнаруженных в данный момент узлов.

На рисунке 8 показан код, на котором можно увидеть действия приложения, при получении определенного типа сигнала:

```
public class WifiDirectBroadcastReceiver extends BroadcastReceiver {

    private WifiP2pManager mManager;
    private Channel mChannel;
    private MyWifiActivity mActivity;

    public WifiDirectBroadcastReceiver(WifiP2pManager manager, Channel channel,
        MyWifiActivity activity) {
        super();
        this.mManager = manager;
        this.mChannel = channel;
        this.mActivity = activity;
    }

    @Override
    public void onReceive(Context context, Intent intent) {
        String action = intent.getAction();

        if (WifiP2pManager.WIFI_P2P_STATE_CHANGED_ACTION.equals(action)) {
            // Check to see if Wi-Fi is enabled and notify appropriate activity
        } else if (WifiP2pManager.WIFI_P2P_PEERS_CHANGED_ACTION.equals(action)) {
            // Call WifiP2pManager.requestPeers() to get a list of current peers
        } else if (WifiP2pManager.WIFI_P2P_CONNECTION_CHANGED_ACTION.equals(action)) {
            // Respond to new connection or disconnections
        } else if (WifiP2pManager.WIFI_P2P_THIS_DEVICE_CHANGED_ACTION.equals(action)) {
            // Respond to this device's wifi state changing
        }
    }
}
```

Рисунок 8 – Описание действий приложения при получении определенного типа сигнала

На рисунке 8 показан пример реализации метода *onReceive()* с четырьмя видами сигнала: проверка возможности использования Wi-Fi, запрос списка доступных узлов, сигнал о новом подключении или отключении от устройства, ответ на изменения состояния в сети.

3.3.3 Настройка разрешений в манифесте

Для того, чтобы использовать Wi-Fi P2P, необходимо добавить *CHANGE_WIFI_STATE*, *ACCESS_WIFI_STATE* и *INTERNET* в Android Manifest. Wi-Fi P2P не требует подключения к Интернету, но использует стандартные сокет Java, для которых *INTERNET* требует разрешение. Таким образом, для использования Wi-Fi P2P необходимы разрешения, показанные на рисунке 9:

```
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
<uses-permission android:name="android.permission.CHANGE_NETWORK_STATE" />
<uses-permission android:name="android.permission.INTERNET" />
```

Рисунок 9 – Список необходимых разрешений

На рисунке 9 показан список разрешений для использования функций устройства: доступ к Wi-Fi и изменению его состояния, к сети и интернету.

3.3.4 Настройка широкополосного приемника

Чтобы использовать Wi-Fi P2P, необходимо прослушать трансляции, которые сообщают приложению, когда происходят определенные события. В приложении был создан экземпляр и установлен *IntentFilter* для получения следующей информации:

-*WIFI_P2P_STATE_CHANGED_ACTION* – указывает, включен ли Wi-Fi P2P;

-*WIFI_P2P_PEERS_CHANGED_ACTION* – указывает, что список доступных одноранговых узлов изменился;

-*WIFI_P2P_CONNECTION_CHANGED_ACTION* – показывает состояние подключения Wi-Fi P2P;

-*WIFI_P2P_THIS_DEVICE_CHANGED_ACTION* – указывает, что данные конфигурации устройства изменились [14].

Пример кода, реализующий получение уведомлений указанных выше, показан на рисунке 10.

```
private final IntentFilter intentFilter = new IntentFilter();
private Channel channel;
private BroadcastReceiver receiver = null;

/**
 * @param isWifiP2pEnabled the isWifiP2pEnabled to set
 */
public void setIsWifiP2pEnabled(boolean isWifiP2pEnabled) {
    this.isWifiP2pEnabled = isWifiP2pEnabled;
}

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);

    // add necessary intent values to be matched.

    intentFilter.addAction(WifiP2pManager.WIFI_P2P_STATE_CHANGED_ACTION);
    intentFilter.addAction(WifiP2pManager.WIFI_P2P_PEERS_CHANGED_ACTION);
    intentFilter.addAction(WifiP2pManager.WIFI_P2P_CONNECTION_CHANGED_ACTION);
    intentFilter.addAction(WifiP2pManager.WIFI_P2P_THIS_DEVICE_CHANGED_ACTION);
}
```

Рисунок 10 – Методы класса IntentFilter

В результате метода onCreate() получается экземпляр объекта WifiP2pManager и его можно вызвать методом initialize(). Метод WifiP2pManager.Channel возвращает объект, который используется для подключения приложения к фреймворку Wi-Fi P2P. Пример кода показан на рисунке 11.

```
Channel mChannel;

public void onCreate(Bundle savedInstanceState) {
    ....
    mManager = (WifiP2pManager) getSystemService(Context.WIFI_P2P_SERVICE);
    mChannel = mManager.initialize(this, getMainLooper(), null);
}
```

Рисунок 11 – Использование метода OnCreate()

Теперь создадим новый класс *BroadcastReceiver*, который будем использовать для прослушивания изменений в состоянии системы Wi-Fi P2P. В методе *onReceive()* добавим условия для обработки каждого изменения состояния P2P, указанного выше. Результат получившегося кода показан на рисунке 12.

```

@Override
public void onReceive(Context context, Intent intent) {
    String action = intent.getAction();
    if (WifiP2pManager.WIFI_P2P_STATE_CHANGED_ACTION.equals(action)) {

        // UI update to indicate wifi p2p status.
        int state = intent.getIntExtra(WifiP2pManager.EXTRA_WIFI_STATE, -1);
        if (state == WifiP2pManager.WIFI_P2P_STATE_ENABLED) {
            // Wifi Direct mode is enabled
            activity.setIsWifiP2pEnabled(true);
        } else {
            activity.setIsWifiP2pEnabled(false);
            activity.resetData();
        }

        Log.d(WifiDirectActivity.TAG, "P2P state changed - " + state);
    } else if (WifiP2pManager.WIFI_P2P_PEERS_CHANGED_ACTION.equals(action)) {

        // request available peers from the wifi p2p manager. This is an
        // asynchronous call and the calling activity is notified with a
        // callback on PeerListListener.onPeersAvailable()
        if (manager != null) {
            manager.requestPeers(channel, (PeerListListener) activity.getFragmentManager()
                .findFragmentById(R.id.frag_list));
        }
        Log.d(WifiDirectActivity.TAG, "P2P peers changed");
    } else if (WifiP2pManager.WIFI_P2P_CONNECTION_CHANGED_ACTION.equals(action)) {

        if (manager == null) {
            return;
        }

        NetworkInfo networkInfo = (NetworkInfo) intent
            .getParcelableExtra(WifiP2pManager.EXTRA_NETWORK_INFO);

        if (networkInfo.isConnected()) {

            // we are connected with the other device, request connection
            // info to find group owner IP

            DeviceDetailFragment fragment = (DeviceDetailFragment) activity
                .getFragmentManager().findFragmentById(R.id.frag_detail);
            manager.requestConnectionInfo(channel, fragment);
        } else {
            // It's a disconnect
            activity.resetData();
        }
    } else if (WifiP2pManager.WIFI_P2P_THIS_DEVICE_CHANGED_ACTION.equals(action)) {
        DeviceListFragment fragment = (DeviceListFragment) activity.getFragmentManager()
            .findFragmentById(R.id.frag_list);
        fragment.updateThisDevice((WifiP2pDevice) intent.getParcelableExtra(
            WifiP2pManager.EXTRA_WIFI_P2P_DEVICE));
    }
}

```

Рисунок 12 – Подробное описание действий в методе OnReceive()

Необходимо установить *IntentFilter* и *BroadcastReceiver* в состояние паузы, когда активен основной поток. Лучшее место, чтобы сделать это – *onResume()* и *onPause()* методы, код которых показан на рисунке 13.

```
@Override
public void onResume() {
    super.onResume();
    receiver = new WifiDirectBroadcastReceiver(mManager, mChannel, this);
    registerReceiver(receiver, intentFilter);
}

@Override
public void onPause() {
    super.onPause();
    unregisterReceiver(receiver);
}
```

Рисунок 13 – Методы OnResume() и OnPause()

Чтобы обнаружить соседние узлы, к которым можно подключиться, необходимо вызвать функцию *discoverPeers()* для обнаружения доступных одноранговых узлов, находящихся в зоне действия. Вызов этой функции является асинхронным, и успех или отказ передается в приложение с помощью методов *onSuccess()* и *onFailure()*, которые относятся к классу *WifiP2pManager.ActionListener*. Метод *onSuccess()* уведомляет о том, что процесс открытия удался и не дает никакой информации о реальных узлах, что он обнаружил, если таковые имеются. Пример кода показан на рисунке 14.

```
@Override
public void connect(WifiP2pConfig config) {
    manager.connect(channel, config, new ActionListener() {

        @Override
        public void onSuccess() {
            // WifiDirectBroadcastReceiver will notify us. Ignore for now.
        }

        @Override
        public void onFailure(int reason) {
            Toast.makeText(WifiDirectActivity.this, "Ошибка подключения",
                Toast.LENGTH_SHORT).show();
        }
    });
}
```

Рисунок 14 – Пример метода OnSuccess() и OnFailure()

3.4 Разработка модуля построения списка абонентов

Если процесс обнаружения завершается успешно и соседние узлы обнаружены, система передает сигнал WIFI_P2P_PEERS_CHANGED_ACTION, который можно использовать для получения списка соседних узлов. Когда приложение получает сигнал WIFI_P2P_PEERS_CHANGED_ACTION, можно запросить список обнаруженных одноранговых узлов с помощью requestPeers(). Пример кода показан на рисунке 15:

```
        if (manager != null) {
            manager.requestPeers(channel, (PeerListListener) activity.getFragmentManager()
                .findFragmentById(R.id.frag_list));
        }
        Log.d(WiFiDirectActivity.TAG, "P2P peers changed");
    } else if (WifiP2pManager.WIFI_P2P_CONNECTION_CHANGED_ACTION.equals(action)) {

        if (manager == null) {
            return;
        }
    }
```

Рисунок 15 – Пример реализации метода RequestPeers()

Метод requestPeers() асинхронный и может отправлять уведомления, когда список пиров доступен с onPeersAvailable(), который определен в интерфейсе WifiP2pManager.PeerListListener. Метод onPeersAvailable() предоставляет параметр WifiP2pDeviceList, который можно использовать для выбора устройства, к которому необходимо подключиться.

Когда определили устройство, к которому необходимо подключиться, из списка полученных соседних узлов, вызовем метод connect() для подключения к устройству. Для вызова этого метода WifiP2pConfig требуется объект, содержащий информацию об устройстве, к которому необходимо подключиться. С помощью WifiP2pManager.ActionListener можно получать уведомления об успешном или неудачном подключении к устройству. В коде на рисунке 16 показано подключение к нужному устройству:

```

@Override
public void connect(WifiP2pConfig config) {
    manager.connect(channel, config, new ActionListener() {

        @Override
        public void onSuccess() {

        }

        @Override
        public void onFailure(int reason) {
            Toast.makeText(WiFiDirectActivity.this, "Ошибка подключения",
                Toast.LENGTH_SHORT).show();
        }

    });
}
}

```

Рисунок 16 – Пример использования метода connect()

На рисунке 16 показано подключение к устройству из списка доступных узлов и вывод ошибки, если подключение не удалось.

3.5 Разработка модуля передачи данных

После установки соединения можно передавать данные между устройствами с помощью сокетов. Основные шаги передачи данных следующие:

Создать `ServerSocket`. Этот сокет ожидает подключения от клиента на указанном порту и блокирует его до тех пор, пока это не произойдет. Процесс работает в фоновом потоке;

Создать `ClientSocket`. Клиент использует порт серверного сокета для подключения к серверному устройству;

Отправлять данные от клиента на сервер. Когда клиентский сокет успешно подключается к серверному сокету, можно отправлять данные с клиента на сервер;

Сокет сервера ожидает подключения клиента с помощью метода `accept()`. Этот вызов блокируется до тех пор, пока клиент не подключится, поэтому этот вызов происходит в другом потоке. Когда происходит соединение, серверное устройство может получать данные от клиента. С файлами можно производить любые действия: считывать, открывать, и т.д.

```

public static class FileServerAsyncTask extends AsyncTask<Void, Void, String> {

    private Context context;
    private TextView statusText;

    /**
     * @param context
     * @param statusText
     */
    public FileServerAsyncTask(Context context, View statusText) {
        this.context = context;
        this.statusText = (TextView) statusText;
    }

    @Override
    protected String doInBackground(Void... params) {
        try {
            ServerSocket serverSocket = new ServerSocket(8988);
            Log.d(WiFiDirectActivity.TAG, "Server: Socket opened");
            Socket client = serverSocket.accept();
            Log.d(WiFiDirectActivity.TAG, "Server: connection done");
            final File f = new File(Environment.getExternalStorageDirectory() + "/"
                + context.getPackageName() + "/wifip2pshared-" + System.currentTimeMillis()
                + ".jpg");

            File dirs = new File(f.getParent());
            if (!dirs.exists())
                dirs.mkdirs();
            f.createNewFile();

            Log.d(WiFiDirectActivity.TAG, "server: copying files " + f.toString());
            InputStream inputStream = client.getInputStream();
            copyFile(inputStream, new FileOutputStream(f));
            serverSocket.close();
            return f.getAbsolutePath();
        } catch (IOException e) {
            Log.e(WiFiDirectActivity.TAG, e.getMessage());
            return null;
        }
    }

    /**
     * (non-Javadoc)
     * @see android.os.AsyncTask#onPostExecute(java.lang.Object)
     */
    @Override
    protected void onPostExecute(String result) {
        if (result != null) {
            statusText.setText("File copied - " + result);
            Intent intent = new Intent();
            intent.setAction(android.content.Intent.ACTION_VIEW);
            intent.setDataAndType(Uri.parse("file://" + result), "image/*");
            context.startActivity(intent);
        }
    }
}

```

Рисунок 17 – Пример кода для создания сокета сервера

На рисунке 17 показан код, который позволяет отправлять файлы пользователя другому пользователю, к которому на данный момент подключено приложение, передача и получение данных происходит в фоновом режиме.

На клиенте необходимо подключиться к серверному сокету для возможности передачи данных. Только когда будет установлено соединение, путем подтверждения соединения со стороны обоих устройств, откроется возможность передачи данных.

```
public class FileTransferService extends IntentService {

    private static final int SOCKET_TIMEOUT = 5000;
    public static final String ACTION_SEND_FILE = "com.example.android.wifidirect.SEND_FILE";
    public static final String EXTRAS_FILE_PATH = "file_url";
    public static final String EXTRAS_GROUP_OWNER_ADDRESS = "go_host";
    public static final String EXTRAS_GROUP_OWNER_PORT = "go_port";

    public FileTransferService(String name) { super(name); }

    public FileTransferService() { super("FileTransferService"); }

    /*
     * (non-Javadoc)
     * @see android.app.IntentService#onHandleIntent(android.content.Intent)
     */
    @Override
    protected void onHandleIntent(Intent intent) {

        Context context = getApplicationContext();
        if (intent.getAction().equals(ACTION_SEND_FILE)) {
            String fileUri = intent.getExtras().getString(EXTRAS_FILE_PATH);
            String host = intent.getExtras().getString(EXTRAS_GROUP_OWNER_ADDRESS);
            Socket socket = new Socket();
            int port = intent.getExtras().getInt(EXTRAS_GROUP_OWNER_PORT);

            try {
                Log.d(WiFiDirectActivity.TAG, "Opening client socket - ");
                socket.bind(null);
                socket.connect((new InetSocketAddress(host, port)), SOCKET_TIMEOUT);

                Log.d(WiFiDirectActivity.TAG, "Client socket - " + socket.isConnected());
                OutputStream stream = socket.getOutputStream();
                ContentResolver cr = context.getContentResolver();
                InputStream is = null;
                try {
                    is = cr.openInputStream(Uri.parse(fileUri));
                } catch (FileNotFoundException e) {
                    Log.d(WiFiDirectActivity.TAG, e.toString());
                }
                DeviceDetailFragment.copyFile(is, stream);
                Log.d(WiFiDirectActivity.TAG, "Client: Data written");
            } catch (IOException e) {
                Log.e(WiFiDirectActivity.TAG, e.getMessage());
            } finally {
                if (socket != null) {
                    if (socket.isConnected()) {
                        try {
                            socket.close();
                        } catch (IOException e) {
                            // Give up
                            e.printStackTrace();
                        }
                    }
                }
            }
        }
    }
}
```

Рисунок 18 – Пример кода для работы клиента

В примере, показанном на рисунке 18, показаны методы класса FileTransferService: отправка и выбор файла, определение источника и получателя.

В результате разработки были реализованы: алгоритм поиска маршрута, алгоритм передачи данных.

Были созданы сервисы для работы устройств в виде клиента и сервера, присваивая их когда им это необходимо.

4 Тестирование приложения на мобильном устройстве

Для тестирования приложения было использовано мобильное устройство Leagoo Elite 5. На рисунке 19 показан главный экран приложения.

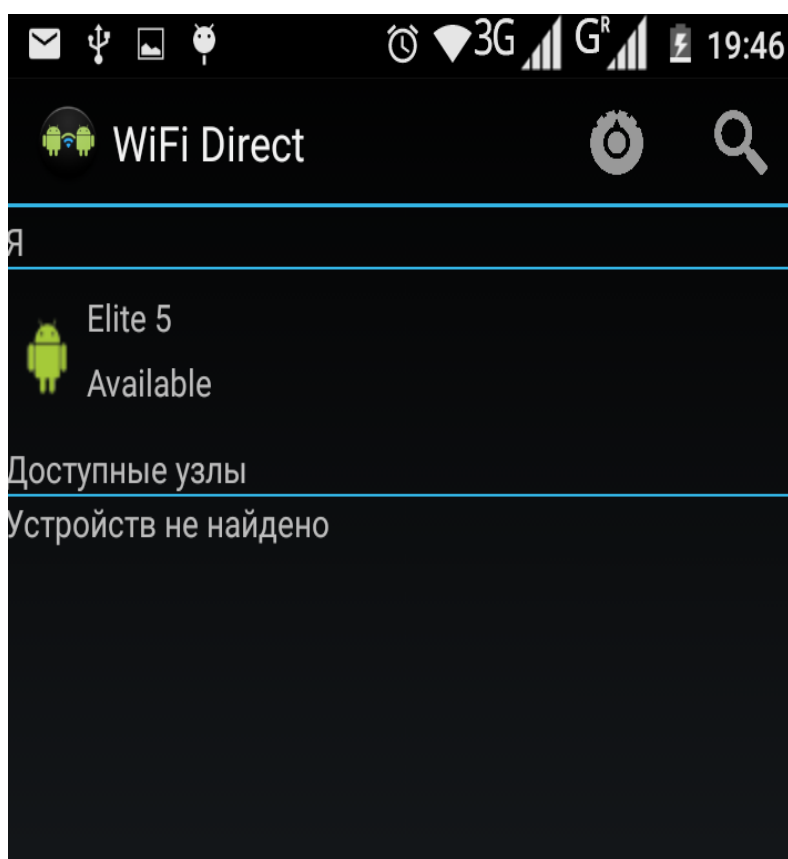


Рисунок 19 – Главный экран приложения

Во фрагменте «Я» показано наименование устройства на котором используется приложение.

Во фрагменте «Доступные узлы» показана надпись «Устройств не найдено» так как Wi-Fi Direct ещё не был активирован.

Во фрагменте «Wi-Fi Direct» есть две кнопки меню, первая открывает настройки для включения Wi-Fi, вторая для инициализации поиска доступных узлов.

Нажав на значок в виде «Лупы», мы инициируем поиск доступных узлов. На рисунке 20 показан процесс поиска доступных узлов.

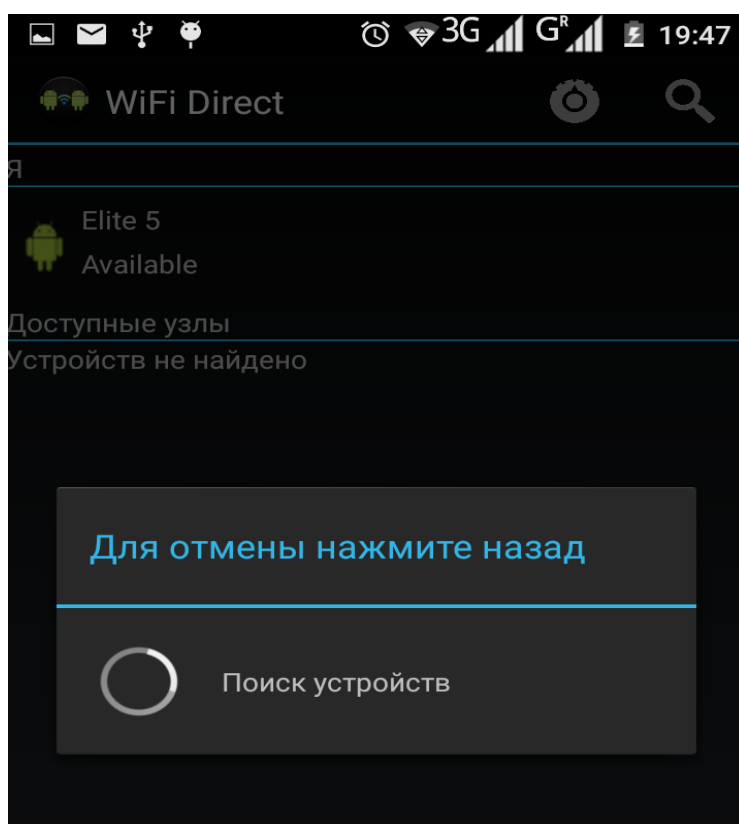


Рисунок 20 – Поиск доступных узлов

После того, как приложение обнаружит соседнее устройство, во фрагменте «Доступные узлы» отобразятся найденные устройства. Результат показан на рисунке 21.

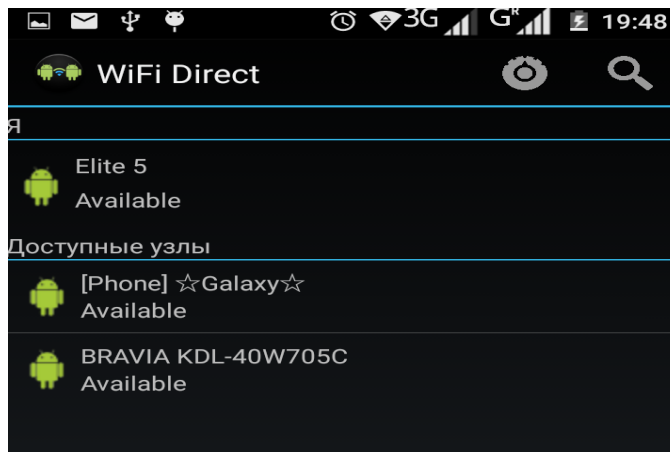


Рисунок 21 – Результат поиска доступных узлов

Во фрагменте «Доступные узлы» отображены найденные соседи, которые транслируют сигнал с помощью Wi-Fi Direct.

При одном нажатии на одно из доступных устройств, мы можем получить подробную информацию об устройстве. Информация показана на рисунке 22.

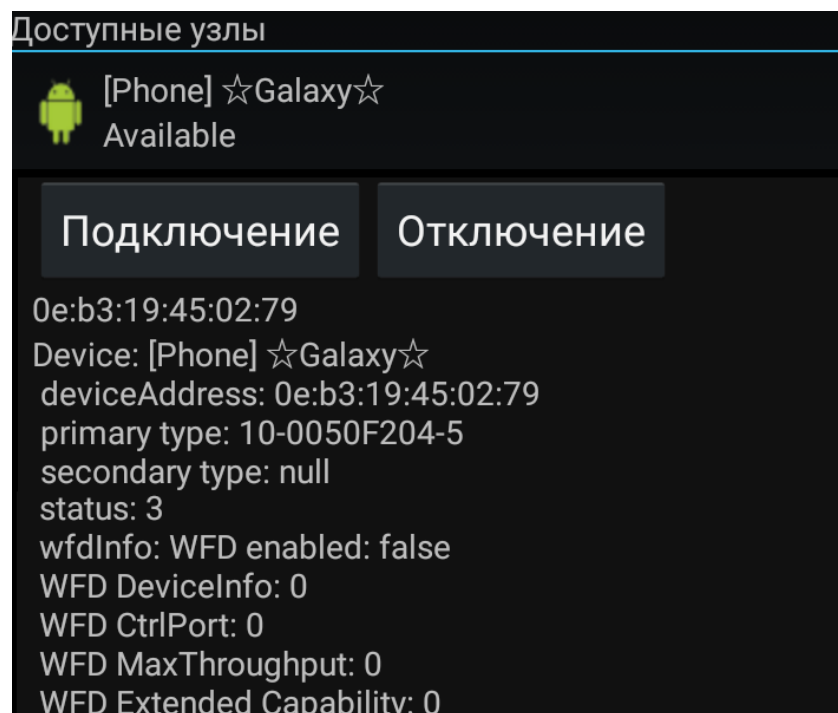


Рисунок 22 – Информация об устройстве

-deviceAddress – или MAC-адрес устройства;

-primary type – тип устройства (может быть телефон, принтер и другое);

-secondary type – дополнительный атрибут к primary type;

- status – статус подключения устройства;
- wfdInfo – подключение активно или нет;
- WFD DeviceInfo – информация об устройстве;
- WFD CtrlPort – порт;
- WFD MaxThroughput – максимальная пропускная способность;
- WFD Extended Capability – дополнительные возможности для изображений.

Нажав на кнопку «подключение» наши устройства смогут присоединиться друг к другу. На рисунке 23 показано подключение к устройству.

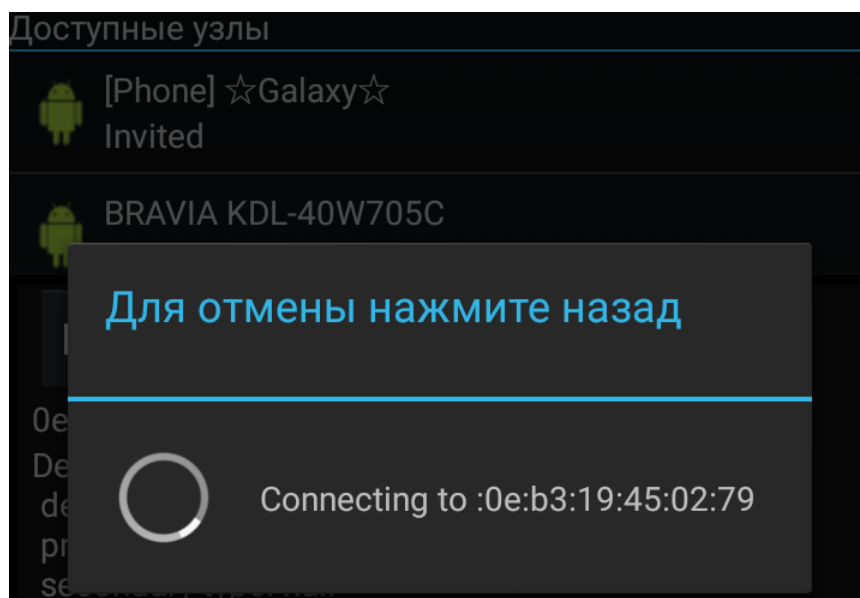


Рисунок 23 – Подключение к устройству

Когда подключение произойдет успешно, статус устройства изменится на «Connected» и появится возможность передавать информацию. Результат показан на рисунке 24.

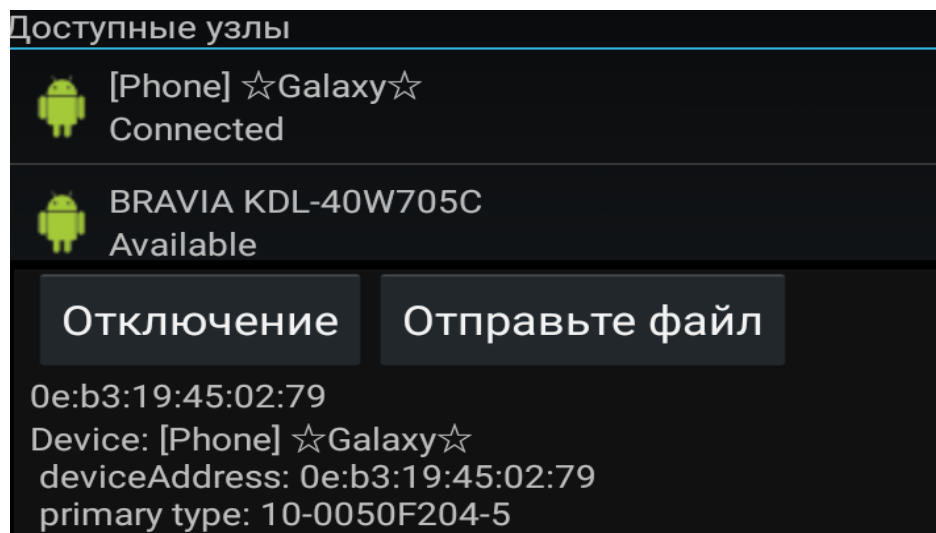


Рисунок 24 – Устройства подключены

Далее можно отправлять файлы нажав кнопку «Отправьте файл», после чего откроется память телефона и можно передать файл.

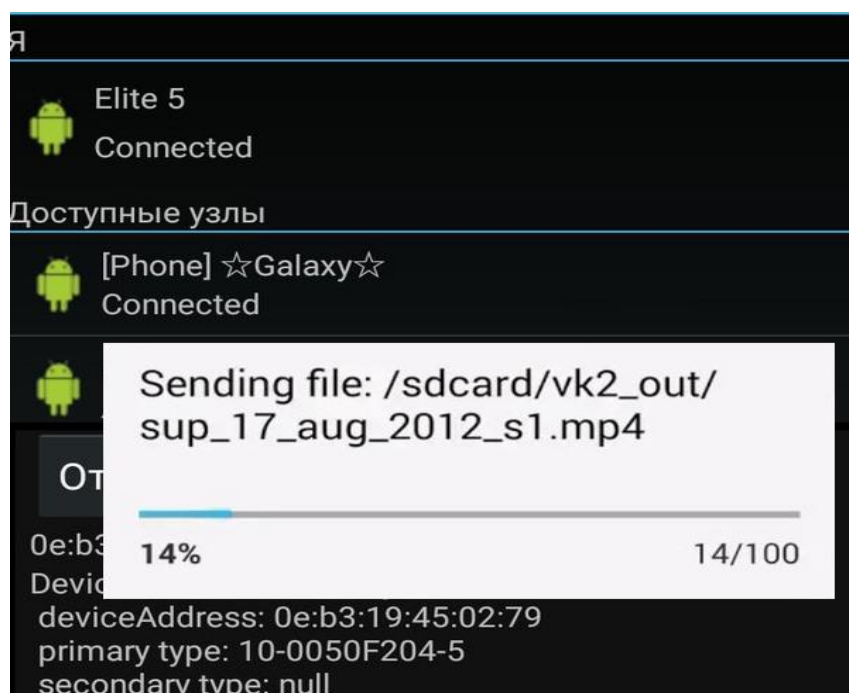


Рисунок 25 – процесс отправки файла

На рисунке 25 показан процесс отправки файла.

В результате тестирования были продемонстрированы возможности поиска соседних узлов, получение подробной информации о них, подключение к узлам и передача данных конкретному узлу. Так же был продемонстрирован пользовательский интерфейс. В качестве макетов были использованы стандартные фреймы Android Studio.

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И
РЕСУРСОСБЕРЕЖЕНИЕ»**

Студенту:

Группа	ФИО
8ИМ5А	Никитин Илья Александрович

Институт	кибернетики	Кафедра	Вычислительной техники
Уровень образования	магистр	Направление/специальность	Информационные системы и технологии

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:

1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих	Изучение информации представленной в различных материалах: – научных статьях; – научных публикация; – нормативно-правовых документах.
2. Нормы и нормативы расходования ресурсов	
3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования	

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. Оценка коммерческого и инновационного потенциала НТИ	Предпроектный анализ, описание целей исследования и возможных направлений применения полученных результатов
2. Разработка устава научно-технического проекта	
3. Планирование процесса управления НТИ: структура и график проведения, бюджет, риски и организация закупок	Планирование трудоемкости работ, построение календарного плана, построение линейного графика, расчет бюджета.
4. Определение ресурсной, финансовой, экономической эффективности	Оценка научно-технического уровня НИР.

Перечень графического материала (с точным указанием обязательных чертежей):

1. «Портрет» потребителя результатов НТИ
2. Сегментирование рынка
3. Оценка конкурентоспособности технических решений
4. Диаграмма FAST
5. Матрица SWOT
6. График проведения и бюджет НТИ
7. Оценка ресурсной, финансовой и экономической эффективности НТИ
8. Потенциальные риски

Дата выдачи задания для раздела по линейному графику	
---	--

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
доцент	Попова С.Н.	к.э.н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ИМ5А	Никитин Илья Александрович		

5 Финансовый менеджмент, ресурсоэффективность и ресурсосбережение

Целью данного раздела является определение оценки коммерческого потенциала, перспективности и альтернатив проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения, а также планирование и формирование бюджета научных исследований, определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования. Научно-исследовательская работа направлена на разработку алгоритмического и программного обеспечения для передачи данных в сетях Manet.

5.1 Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения

5.1.1. Потенциальные потребители результатов исследования

Научно-исследовательская работа направлена на создание алгоритмического и программного обеспечения для передачи данных в сетях MANET.

Основным направлением реализации разработанного продукта является применение его в качестве системы передачи данных, которая позволит передавать данные с помощью технологии Wi-Fi Direct между двумя мобильными устройствами, что позволит коммуницировать пользователям в труднодоступных зонах, или где отсутствуют средства телекоммуникации. Пользователем может быть любой человек, установивший мобильное приложение на свое мобильное устройство.

С помощью разработанного алгоритмического обеспечения, возможна модификация программы, вследствие чего возможна передача данных между двумя и более устройствами.

5.2 Организация и планирование работ

В данном разделе составляется список проводимых работ, определяются их исполнители и продолжительность. Так как число

исполнителей не превышает двух, линейный график работ является наиболее удобным и компактным способом представления данных планирования.

График выполнения научно-исследовательской работы представлен в таблице 15.

Таблица 15 – Перечень работ и продолжительность их выполнения

Этапы работы	Исполнители	Загрузка исполнителей
Постановка целей и задач, получение исходных данных	НР	НР – 100%
Составление и утверждение ТЗ	НР, И	НР – 90% И – 10%
Подбор и изучение материалов по тематике	НР, И	НР – 30% И – 70%
Разработка календарного плана	НР, И	НР – 80% И – 20%
Анализ аналогов программного продукта и алгоритмического обеспечения	НР, И	НР – 10% И – 90%
Проектирование алгоритмического обеспечения	НР, И	НР – 20% И – 80%
Разработка программного продукта	НР, И	НР – 10% И – 90%
Тестирование ПО	НР, И	НР – 15% И – 85%
Анализ результатов	НР, И	НР – 25% И – 75%
Оформление пояснительной записки	И	И – 100%
Проверка работы	НР, И	НР – 70% И – 30%

Примечание к таблице 15: НР — научный руководитель; И — инженер.

5.2.1 Продолжительность этапов работ

Расчет продолжительности этапов работ может осуществляться опытно-статистическим методом. Для расчета ожидаемого значения продолжительности работ $t_{ож}$ применяются две оценки: t_{\min} и t_{\max} (метод двух оценок).

$$t_{ож} = \frac{3t_{\min} + 2t_{\max}}{5}, \quad (5.1)$$

где t_{\min} – минимальная продолжительность работ, чел/дн;

t_{\max} – максимальная продолжительность работ, чел/дн.

Для построения линейного графика рассчитывается длительность этапов в рабочих днях, а затем осуществляется её перевод в календарные дни.

Расчёт продолжительности выполнения каждого этапа в рабочих днях ($T_{РД}$)

выполняется по формуле:

$$T_{РД} = \frac{t_{о\text{жс}}}{K_{ВН}} \cdot K_{Д}, \quad (5.2)$$

где $t_{о\text{жс}}$ – продолжительность работы, дн.;

$K_{ВН}$ – коэффициент выполнения работ ($K_{ВН} = 1$);

$K_{Д}$ – коэффициент, учитывающий дополнительное время на компенсацию непредвиденных задержек и согласование работ ($K_{Д} = 1,1$).

Расчёт продолжительности этапа в календарных днях осуществляется по формуле:

$$T_{КД} = T_{РД} \cdot T_{К}, \quad (5.3)$$

где $T_{КД}$ – продолжительность выполнения этапа в календарных днях;

$T_{РД}$ – продолжительность выполнения этапа в рабочих днях;

$T_{К}$ – коэффициент календарности.

Коэффициент календарности рассчитывается по формуле:

$$T_{К} = \frac{T_{КАЛ}}{T_{КАЛ} - T_{ВД} - T_{ПД}}, \quad (5.4)$$

где $T_{КАЛ}$ – календарные дни, $T_{КАЛ} = 365$;

$T_{ВД}$ – выходные дни, $T_{ВД} = 52$;

$T_{ПД}$ – праздничные дни, $T_{ПД} = 10$.

Подставив значения в формулу 5.4, получим следующий результат:

$$T_{К} = \frac{365}{365 - 52 - 10} = 1,205. \quad (5.5)$$

В таблице 16 приведена длительность этапов работ и число исполнителей, занятых на каждом этапе.

Таблица 16 – Временные показатели проведения научного исследования

Этап	Исполнители	Продолжительность работ, дни			Длительность работ, чел/дн.			
		t_{min}	t_{max}	$t_{ож}$	$T_{рд}$		$T_{кд}$	
					НР	И	НР	И
1. Постановка целей и задач, получение исходных данных	НР	2	4	2,8	3,08	0	3,7	0
2. Составление и утверждение технического задания	НР, И	3	6	4.2	4.158	0,462	5,01	0,34
3. Подбор и изучение материалов по тематике	НР, И	7	10	8,2	2,707	6,314	3,26	7,61
4. Разработка календарного плана	НР, И	1	3	1.8	1.584	0,396	1,91	0,48
5. Анализ аналогов программного продукта и алгоритмического обеспечения	НР, И	12	17	14	1,54	13,86	1,86	16,7
6. Проектирование алгоритмического обеспечения	НР, И	15	21	17,4	3,828	15,572	4,61	18,76
7. Разработка программного продукта	НР, И	15	21	17,4	1,914	17,226	2,31	20,76

Этап	Исполнители	Продолжительность работ, дни			Длительность работ, чел/дн.			
					T_{RD}		T_{KD}	
		t_{min}	t_{max}	$t_{ож}$	НР	И	НР	И
8. Тестирование ПО	НР, И	10	12	10,8	1,782	10,098	2,15	12,17
9. Анализ результатов	НР, И	5	9	6,6	1,815	5,445	2,19	6,56
10. Оформление пояснительной записки	И	3	5	3,8	0	4,18	0	5,04
11. Проверка работы	НР, И	2	3	2,4	1,848	0,792	2,23	0,95
Итого:		75	111	89.4	24,256	74,525	29,23	89,37

Таблица 17 – Календарный план-график проведения работ

Э	Т _{кд} НР	Т _{кд} И	Февраль			Март			Апрель			Май			Июнь		
	3,7	0															
	5,01	0,34															
	3,26	7,61															
	1,91	0,48															
	1,86	16,7															
	4,61	18,76															
	2,31	20,76															
	2,15	12,17															
	2,19	6,56															
0	0	5,04															
1	2,23	0,95															

НР –  ; И – 

5.2.2 Расчет накопления технической готовности

В данном разделе производится оценка текущих результатов работы над проектом. Величина накопления готовности работы показывает, на сколько процентов по окончании текущего этапа выполнен общий объем работ по проекту в целом.

Степень готовности определяется формулой:

$$CG_i = \frac{TP_i^H}{TP_{общ}} = \frac{\sum_{k=1}^i TP_k}{TP_{общ}} = \frac{\sum_{k=1}^i \sum_{j=1}^m TP_{km}}{\sum_{k=1}^I \sum_{j=1}^m TP_{km}}, \quad (5.5)$$

где $TP_{общ}$ – общая трудоемкость проекта;

TP_i (TP_k) – трудоемкость i -го (k -го) этапа проекта, $i = \overline{1, I}$;

TP_i^H – накопленная трудоемкость i -го этапа проекта по его завершении;

TP_{ij} (TP_{kj}) – трудоемкость работ, выполняемых j -м участником на i -м

этапе.

Таблица 18 – Нарастание технической готовности работы

Этап	ТР _i ,%	СГ _i ,%
1. Постановка целей и задач, получение исходных данных	1,47	1,47
2. Составление и утверждение технического задания	4,4	5,87
3. Подбор и изучение материалов по тематике	7,87	13,74
4. Разработка календарного плана	1,39	15,13
5. Анализ аналогов программного продукта и алгоритмического обеспечения	13,44	28,57
6. Проектирование алгоритмического обеспечения	23,09	51,66
7. Разработка программного продукта	23,09	74,75
8. Тестирование ПО	12,43	87,18
9. Анализ результатов	6,25	93,43
10. Оформление пояснительной записки	3,92	97,35
11. Проверка работы	2,65	100

5.3 Расчёт сметы затрат на выполнение проекта

Состав затрат на научно-исследовательскую работу состоит из всех расходов, необходимых для реализации комплекса работ, составляющих содержание данного исследования. Так как научно-исследовательская работа проводилась на домашнем компьютере, без аренды помещения и в программном обеспечении с бесплатной студенческой лицензией расчет сметной стоимости производится по следующим статьям затрат:

- материалы и покупные изделия;
- заработная плата;

- социальный налог;
- расходы на электроэнергию (без освещения);
- амортизационные отчисления;
- оплата услуг связи;
- прочие (накладные расходы) расходы.

5.3.1 Расчёт затрат на материалы

К данной статье расходов относится стоимость всех материалов, расходуемых непосредственно в процессе выполнения работ.

Таблица 19 – Расчёт затрат на материалы

Наименование	Единица измерения	Количество	Цена за ед., руб	Затраты на материалы, (З _м), руб.
Бумага формата А4 для принтера	Уп.	1	290	290
Картридж для принтера	Шт.	1	599	599
Итого				889

Транспортно-заготовительные расходы (ТРЗ) составляют 5% от отпускной цены материалов. Расходы на материалы с учётом ТРЗ:

$$C_{MAT} = 889 \cdot 1,05 = 933,45 \text{ руб.}$$

5.3.2 Расчёт заработной платы

Данная статья расходов включает заработную плату научного руководителя и студента, а также премии, входящие в фонд заработной платы. Расчет основной заработной платы выполняется на основе трудоёмкости выполнения каждого этапа и величины месячного оклада исполнителя.

Величина месячного оклада научного руководителя (МОНР) получена из открытых данных, размещенных на официальном сайте Национального исследовательского Томского политехнического университета. Величина месячного оклада инженеров (МОИ) берется как месячный оклад инженера кафедры.

Основной расчет фонда заработной платы выполняется по формуле:

$$ЗП_{\text{дн-т}} = MO/N, \quad (5.7)$$

где MO – месячный оклад, руб.;

N – количество рабочих дней в месяц, при шестидневной рабочей неделе – N = 24,91, а при пятидневной рабочей неделе – N = 20,58.

.Среднедневная заработная плата научного руководителя равна:

$$ЗП_{\text{дн-т}} = \frac{26\,300}{24,91} = 1\,055,8 \frac{\text{руб.}}{\text{раб. день}}$$

А среднедневная тарифная заработная плата инженеров равна

$$ЗП_{\text{дн-т}} = \frac{14874,45}{20,58} = 722,76 \frac{\text{руб.}}{\text{раб. день}}$$

Затраты времени по каждому исполнителю в рабочих днях взяты из таблицы 16. Для перехода от тарифной суммы заработка исполнителя, связанной с участием в проекте, к соответствующему полному заработку необходимо будет тарифную сумму заработка исполнителя, связанной с участием в проекте умножить на интегральный коэффициент. Интегральный коэффициент находится по формуле:

$$K_{\text{и}} = K_{\text{пр}} \cdot K_{\text{доп.ЗП}} \cdot K_{\text{р}}, \quad (5.8)$$

где $K_{\text{пр}}$ – коэффициент премий, $K_{\text{пр}} = 1,1$;

$K_{\text{доп.ЗП}}$ – коэффициент дополнительной зарплаты, при шестидневной рабочей неделе $K_{\text{доп.ЗП}} = 1,188$, а при пятидневной рабочей неделе $K_{\text{доп.ЗП}} = 1,113$;

$K_{\text{р}}$ – коэффициент районной надбавки, $K_{\text{р}} = 1,3$.

Результаты вычислений представлены в таблице 20.

Таблица 20 – Затраты на заработную плату

Исполнитель	Оклад, руб./мес	ЗП _{дн-т} , руб./раб.день	Затраты времени, раб.дни	Коэффи циент	Фонд з/платы, руб.
НР	26300	1055,8	27	1,699	48432,71
И	14874,45	722,76	99	1,59	113769,65
Итого:					162202,36

5.3.3 Расчет отчисления на социальные нужды

Взнос в социальные фонды установлен в размере 30,2% от заработной платы. Размер взноса рассчитывается по формуле:

$$C_{\text{соц}} = C_{\text{ЗП}} \cdot 0,302, \quad (5.9)$$

где $C_{\text{ЗП}}$ – размер заработной платы.

Подставив необходимые значения в формулу 5.10 получим:

$$C_{\text{соц}} = 162202,36 \cdot 0,302 = 48985,11 \text{ руб.}$$

5.3.4 Расчет затрат на электроэнергию

Затраты на электроэнергию рассчитываются по формуле:

$$C_{\text{эл.об.}} = P_{\text{об.}} \cdot t_{\text{об.}} \cdot Ц_{\text{э}}, \quad (5.10)$$

где $P_{\text{об.}}$ – мощность, потребляемая оборудованием, кВт;

$t_{\text{об.}}$ – время работы оборудования, час;

$Ц_{\text{э}}$ – тариф на 1 кВт·час. Для ТПУ, $Ц_{\text{э}} = 5,8 \text{ руб./кВт} \cdot \text{час}$.

Время работы оборудования вычисляется на основе итоговых данных таблицы 17 для инженера ($T_{\text{рд}}$) из расчета, что продолжительность рабочего дня равна 8 часов.

$$t_{\text{об.}} = T_{\text{рд}} \cdot K_t, \quad (5.11)$$

где K_t – коэффициент использования оборудования по времени, $K_t = 0,9$.

Мощность, потребляемая оборудованием, определяется по формуле:

$$P_{\text{об.}} = P_{\text{ном}} \cdot K_C, \quad (5.12)$$

где K_C – коэффициент загрузки;

$P_{\text{ном}}$ – номинальная мощность оборудования, кВт. Для технологического оборудования малой мощности $K_C = 1$.

Таблица 21 – Затраты на электроэнергию технологическую

Наименование оборудования	Время работы оборудования $t_{\text{об.}}$, час	Потребляемая мощность $P_{\text{об.}}$, кВт	Затраты $Э_{\text{об.}}$, руб.
Персональный компьютер инженера	791,6	0,09	372,89
Итого:			372,89

5.3.5 Расчет амортизационных расходов

Для расчета амортизационных расходов используется формула:

$$C_{AM} = \frac{H_A \cdot C_{OB} \cdot t_{рф} \cdot n}{F_D}, \quad (5.13)$$

где H_A – годовая норма амортизации единицы оборудования;

C_{OB} – балансовая стоимость единицы оборудования с учетом ТЗР, стоимость ПК инженера – 20 500 руб.;

$t_{рф}$ – фактическое время работы оборудования в ходе выполнения проекта, $t_{рф} = 98,95 \cdot 8 = 791,6$ часов;

n – число задействованных однотипных единиц оборудования;

F_D – действительный годовой фонд времени работы соответствующего оборудования, $F_D = 298 \cdot 8 = 2384$ часа.

H_A определяется по формуле:

$$H_A = \frac{1}{CA}, \quad (5.12)$$

где CA – срок амортизации, который можно получить из постановления правительства РФ «О классификации основных средств, включенных в амортизационные группы» Для электронно-вычислительной техники CA свыше 2 лет до 3 лет включительно. В данной работе примем $CA=2,5$ года. Тогда

$$H_A = \frac{1}{2,5} = 0,4.$$

Таким образом,

$$C_{AM}(ПК) = \frac{0,4 \cdot 20\,500 \cdot 791,6 \cdot 1}{2384} = 2722,78 \text{ руб}$$

Итого начислено амортизации 2722,78 руб.

5.3.6 Расчет расходов на услуги связи

Расходы на услуги связи определены наличием подключения к сети Интернет на компьютере, использованном в данной работе.

Ежемесячная оплата, согласно тарифу TRUnet, составляет 350 рублей. В соответствии с таблицей 5.3, трудоемкость выполняемой задачи составляет

четыре календарных месяца. Таким образом, сумма расходов на услуги связи составляет $4 \cdot 350 = 1750$ руб. Общая сумма расходов $C_{св} = 1400$

5.3.7 Расчет прочих расходов

Прочие расходы следует принять равными 10% от суммы всех предыдущих расходов. Они находятся по формуле:

$$C_{\text{проч}} = (C_{\text{мат}} + C_{\text{ЗП}} + C_{\text{соц}} + C_{\text{эл.об.}} + C_{\text{АМ}} + C_{\text{св}}) \cdot 0,1, \quad (5.13)$$

Где $C_{\text{мат}}$ – расходы на материалы, руб.;

$C_{\text{ЗП}}$ – основная заработная плата, руб.;

$C_{\text{соц}}$ – расходы на единый социальный налог, руб.;

$C_{\text{эл.об.}}$ – расходы на электроэнергию, руб.;

$C_{\text{АМ}}$ – амортизационные расходы, руб.;

$C_{\text{св}}$ – расходы на услуги связи, руб.

Подставив полученные выше результаты, получим:

$$C_{\text{проч}} = (933,45 + 162202,36 + 48985,11 + 372,89 + 2722,98 + 1400) \cdot 0,1 = 21661,68 \text{ руб.}$$

5.3.8 Расчет общей себестоимости разработки

Проведя расчет по всем статьям сметы затрат на разработку, можно определить общую себестоимость проекта.

Таблица 22 – Смета затрат на разработку проекта

Статья затрат	Условное обозначение	Сумма, руб.
Материалы и покупные изделия	$C_{\text{мат}}$	933,45
Основная заработная плата	$C_{\text{ЗП}}$	162202,36
Отчисления в социальные фонды	$C_{\text{соц}}$	48985,11
Расходы на электроэнергию	$C_{\text{эл.об.}}$	372,89
Амортизационные отчисления	$C_{\text{АМ}}$	2722,98
Расходы на услуги связи	$C_{\text{св}}$	1400
Прочие расходы	$C_{\text{проч}}$	21661,68
Итого:		238278,47

Таким образом, затраты на разработку составили $C = 238278,47$ руб

5.3.9 Расчёт прибыли

Прибыль следует принять в размере 20% от полной себестоимости разработки. Прибыль составляет:

$$238278,47 * 0,2 = 47655,69 \text{ , руб.}$$

5.3.10 Цена разработки НИР

Цена разработки научно-исследовательской работы равна сумме полной себестоимости, прибыли и НДС:

$$238278,47 + 47655,69 = 285934,16 \text{ , руб.}$$

5.4 Оценка экономической эффективности

Выполнение научно-исследовательских работ оценивается уровнями достижения экономического, научного, научно-технического и социального эффектов.

Для итоговой оценки результатов проекта в зависимости от поставленных целей в качестве критерия эффективности принимается один из видов эффекта, а остальные используются в качестве дополнительных характеристик.

На данном этапе внедрение нет возможности оценить экономический эффект в количественных показателях. Так как данная разработка является моделью для дальнейшей модификации при решении конкретно поставленной модели. Следовательно, в дальнейшем необходимо рассчитывать данный показатель исходя из заявленных параметров и условий. Поэтому в качестве критерия эффективности проекта оценим научно-технический уровень НИР.

5.4.1 Оценка научно-технического уровня НИР

Научно-технический уровень характеризует влияние проекта на уровень и динамику обеспечения научно-технического прогресса в данной области. Для оценки научной ценности, технической значимости и эффективности, планируемых и выполняемых НИР, используется метод балльных оценок. Каждому фактору по принятой шкале присваивается определенное количество баллов. Обобщенная оценка проводится по сумме баллов по всем показателям. На её основе делается вывод о целесообразности НИР.

Интегральный показатель научно технического уровня НИР определяется по формуле:

$$I_{\text{НТУ}} = \sum_{i=1}^3 R_i \cdot n_i, \quad (5.13)$$

где $I_{\text{НТУ}}$ – интегральный индекс научно-технического уровня;

R_i – весовой коэффициент i -го признака научно-технического эффекта;

n_i – количественная оценка i -го признака научно-технического эффекта,

в баллах.

Весовые коэффициенты признаков НТУ приведены в таблице 5.9.

Таблица 23 – Весовые коэффициенты признаков НТУ

Признаки научно-технического эффекта НИР	Характеристика признака НИР	R_i
Уровень новизны	Систематизируются и обобщаются сведения, определяются пути дальнейших исследований	0,40
Теоретический уровень	Разработка способа (алгоритма)	0,30
Возможность реализации	Время реализации в течение первых лет	0,30

Баллы для оценок уровня новизны, теоретического уровня и возможности реализации приведены в таблицах 24 – 26.

Таблица 24 – Баллы для оценки уровня новизны

Уровень новизны	Характеристика уровня новизны	Баллы
Принципиально новая	Новое направление в науке и технике, новые факты и закономерности, новая теория, вещество, способ	8–10
Новая	По-новому объясняются те же факты, закономерности, новые понятия дополняют ранее полученные результаты	5–7
Относительно новая	Систематизируются, обобщаются имеющиеся сведения, новые связи между известными	2–4
Не обладает новизной	Результат, который ранее был известен	0

Таблица 25 – Баллы значимости теоретических уровней

Теоретический уровень полученных результатов	Баллы
Установка закона, разработка новой теории	8
Глубокая разработка проблемы, многоспектральный анализ взаимодействия между факторами с наличием объяснений	6
Разработка способа (алгоритм, программа)	8
Элементарный анализ связей между фактами (наличие гипотезы, объяснения версии, практических рекомендаций)	2
Описание отдельных элементарных факторов, изложение наблюдений, опыта, результатов измерений	0,5

Таблица 26 – Возможность реализации результатов по времени

Время реализации	Баллы
В течение первых лет	10
От 5 до 10 лет	4
Свыше 10 лет	2

В таблице 27 указано соответствие качественных уровней НИР значениям показателя, рассчитываемого по формуле (5.11).

Таблица 27 – Оценка научно-технического уровня НИР

Фактор НТУ	Значимость	Уровень фактора	Выбранный балл	Обоснование выбранного балла
Уровень новизны	0,4	Новая	5	Облегчит разработку других структур
Теоретический уровень	0,3	Разработка способа	8	Разработка алгоритмического обеспечения
Возможность реализации	0,3	В течение первых лет	8	Быстрая разработка с помощью различных инструментальных средств

Интегральный показатель научно-технического уровня составляет:

$$I_{НТУ} = 0,4 \cdot 5 + 0,1 \cdot 8 + 0,5 \cdot 8 = 6,8.$$

Таблица 28– Оценка уровня научно-технического эффекта

Уровень НТЭ	Показатель НТЭ
Низкий	1–4
Средний	4–7
Высокий	8–10

Таким образом, научно-исследовательская работа имеет средний уровень научно-технического эффекта.

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа	ФИО
8ИМ5А	Никитин Илья Александрович

Институт	Кафедра	Уровень образования	Направление/специальность
		Магистр	09.04.02 Информационные системы и технологии

Исходные данные к разделу «Социальная ответственность»:

<p>1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения</p>	<p>Объектом исследования являются среда разработки Android Studio, язык программирования JAVA; языка разметки XML; использование эмуляторов мобильных устройств в среде Android Studio;</p> <p>Целью магистерской диссертации (предметом) является разработка распределенной автономной системы передачи данных для мобильных устройств используя технологию Wi-Fi Direct.</p>
---	--

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

<p>1 Производственная безопасность</p> <p>1.1 Анализ выявленных вредных факторов при разработке и эксплуатации проектируемого решения в следующей последовательности.</p> <p>1.2 Анализ выявленных опасных факторов при разработке и эксплуатации проектируемого решения в следующей последовательности.</p>	<p>Производственная безопасность на стадии разработки веб-системы.</p> <p>1.1 В качестве вредных факторов выделены:</p> <ul style="list-style-type: none"> – Повышенный уровень шума на рабочем месте; – Недостаточная освещенность рабочей зоны; – Умственное перенапряжение; – Монотонный режим работы. <p>1.2. В качестве опасных факторов выделены:</p> <ul style="list-style-type: none"> – Опасность поражения электрическим током; – Опасность возникновения пожара.
<p>2 Экологическая безопасность:</p> <p>2.1 Анализ воздействия объекта на окружающую среду;</p> <p>2.2 Разработать решения по обеспечению экологической безопасности со ссылками на НТД по охране окружающей среды.</p>	<p>2.1 Влияние объекта исследования на окружающую среду:</p> <ul style="list-style-type: none"> – Образование мусора. – Утилизация компьютерной техники. <p>2.2 Мероприятия по защите окружающей среды.</p>

<p>3 Безопасность в чрезвычайных ситуациях:</p> <p>3.1 Перечень возможных ЧС при разработке и эксплуатации проектируемого решения;</p> <p>3.2 Выбор наиболее типичной ЧС;</p> <p>3.3 Разработка действий в результате возникшей ЧС и мер по ликвидации её последствий.</p>	<p>3.1 Возможные чрезвычайные ситуации:</p> <ul style="list-style-type: none"> – Пожар; – Поражение электрическим током; – Социальные чрезвычайные ситуации (терроризм). <p>3.2 Типичная чрезвычайная ситуация:</p> <ul style="list-style-type: none"> – Возгорание (пожар); – Социальная чрезвычайная ситуация (терроризм). <p>3.3 Мероприятия по предотвращению наиболее типичной ЧС – пожара, согласно нормативным документам: НПБ 105-03 и ППБ 01–03.</p>
<p>4 Правовые и организационные вопросы обеспечения безопасности:</p> <p>4.1 Специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства;</p> <p>4.2 Организационные мероприятия при компоновке рабочей зоны.</p>	<p>4.1 Описание правовых норм для работ, связанных с работой за ПЭВМ согласно следующим документам:</p> <ul style="list-style-type: none"> – Трудовой кодекс Российской Федерации" от 30.12.2001 N 197-ФЗ (ред. от 30.12.2015); <p>4.2 Влияние реализации разрабатываемой системы на конечных пользователей:</p> <ul style="list-style-type: none"> – Технология Wi-Fi Direct позволяет передавать информацию, не используя точки доступа; – Возможность проводить спасательные операции в зонах, где глобальные устройства слежения могут работать со сбоем; – Эффективная передача информации для труднодоступных районов, где не проведена телекоммуникация;

Дата выдачи задания для раздела по линейному графику	22.02.2017
---	------------

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент	Акулов Петр Анатольевич			

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ИМ5А	Никитин Илья Александрович		

6 Социальная ответственность

Разработанный в рамках магистерской диссертации проект является распределенной автономной системой, которая позволит передавать данные между мобильными устройствами не подключаясь к стационарной точке доступа, используя технологию Wi-Fi Direct.

Разработка системы велась с использованием компьютерной техники. Использование средств вычислительной техники, накладывает целый ряд вредных факторов на человека, что впоследствии снижает производительность его труда и может привести к существенным проблемам со здоровьем сотрудника.

Данный раздел посвящен анализу вредных и опасных факторов производственной среды как для разработчиков, так и для пользователей.

6.1 Производственная безопасность на стадии разработки системы

Научно-исследовательская деятельность выполнялась в помещении кафедры «Вычислительной техники» десятого корпуса ТПУ в кабинете 402. Помещение оснащено видео-дисплейными терминалами (ВДТ), персональными электронно-вычислительными машинами (ПЭВМ), компьютерными столами, стульями, столом для коллективной работы, огнетушителями, кондиционером, противопожарной сигнализацией и датчиками дыма.

Для обеспечения производственной безопасности необходимо проанализировать воздействия на человека вредных и опасных производственных факторов, которые могут возникать при разработке проекта.

Производственный фактор считается вредным, если воздействие этого фактора на человека может привести к его заболеванию. Производственный фактор считается опасным, если его воздействие может привести к травме [154].

Все производственные факторы классифицируются по группам элементов: физические, химические, биологические и психофизические. Для данной работы целесообразно рассмотреть физические и психофизические вредные и опасные факторы производства, характерные для рабочей зоны

программиста, разработчика приложения, пользователя. Выявленные факторы представлены в таблице 29.

Таблица 29 – Вредные и опасные производственные факторы при выполнении работ за ПЭВМ

Источник фактора, наименование видов работ	Факторы (по ГОСТ 12.0.003-74)		Нормативные документы
	Вредные	Опасные	
1) Работа за ПК	1) Повышенный уровень шума на рабочем месте; 2) Недостаточная освещенность рабочей зоны; 3) Умственное перенапряжение; 4) Монотонный режим работы.	1) Опасность поражения электрическим током; 2) Опасность возникновения пожара.	1) СН 2.2.4/2.1.8.562-96; 2) СанПиН 2.2.4.548-96; 3) СанПиН 2.2.2/2.4.1340-03; 4) СП 52.13330.2011; 5) ГОСТ Р 12.1.019-2009 ССБТ; 6) СНиП 21-01-97.

6.2 Вредные производственные факторы

6.2.1 Повышенный уровень шума на рабочем месте

Рабочее помещение программиста имеет низкий общий уровень шума. Источником шумовых помех в данном случае могут стать сами программисты, различные устройства, такие как:

- Вентиляторы на процессорах и видеокартах;
- Жесткие диски;
- Вентиляторы блоков питания;

- Принтер;
- Источники шума вне помещения.

Повышенный уровень шума ведет к быстрой утомляемости, к снижению внимания, а также уменьшает на 5-12% производительность труда, при длительном воздействии шума с уровнем звукового давления 90 дБ снижается производительность труда на 30-40% [16].

С целью обеспечения нормальной работы уровень шума нормируется. В помещениях с ПЭВМ вибрация не должна превышать установленных допустимых норм согласно СН 2.2.4/2.1.8.562-96 «Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории жилой застройки». Нормы допустимого шума, вибрации, инфразвука приведены в таблице 6.2.

Таблица 30 – Предельно допустимые уровни звукового давления, уровни звука и эквивалентные уровни звука на рабочих местах программистов вычислительных машин [17]

Уровни звукового давления, дБ, в октавных полосах со среднегеометрическими частотами, Гц									Уровни звука и эквивалентные уровни звука (в дБА)
31,5	63	125	250	500	1000	2000	4000	8000	
86	71	61	54	49	45	42	40	38	50

В качестве мер по снижению шума можно предложить следующее:

- Облицовка потолка и стен звукопоглощающим материалом (снижает шум на 6-8 дБА);
- Экранирование рабочего места (постановка перегородок, диафрагм);
- Установка в компьютерных помещениях оборудования, производящего минимальный шум;
- Рациональная планировка помещения.

Поэтому для уменьшения шума персональные компьютеры могут комплектоваться корпусами с пониженным уровнем шума и жесткими дисками специальных «тихих» модификаций. В результате уровень звука работающей ПЭВМ не превышает 35 дБА.

6.2.2 Недостаточная освещенность рабочей зоны

Недостаточная освещенность рабочей зоны является вредным производственным фактором, возникающим при работе с ПЭВМ, уровни которого регламентируются СП 52.13330.2011.

Причиной недостаточной освещенности являются недостаточность естественного освещения, недостаточность искусственного освещения, пониженная контрастность.

Работа с компьютером подразумевает постоянный зрительный контакт с дисплеем ПЭВМ и занимает от 80 % рабочего времени. Недостаточность освещения снижает производительность труда, увеличивает утомляемость и количество допускаемых ошибок, а также может привести к появлению профессиональных болезней зрения.

Разряд зрительных работ программиста и оператора ПЭВМ относится к разряду III и подразряду Г (работы высокой точности). В таблице 6.3 представлены нормативные показатели искусственного освещения при работах заданной точности.

Таблица 31 – Требования к освещению помещений промышленных предприятий для операторов ПЭВМ [18]

Характеристика зрительной работы	Наименьший или эквивалентный размер объекта различения, мм	Разряд зрительной работы	Подразряд зрительной работы	Контраст объекта с фоном	Характеристика фона	Искусственное освещение		
						Освещённость, лк		
						При системе комбинированного освещения		При системе общего освещения
						всего	В том числе от общего	
Высокой точности	0,264	III	Г	Средний, большой	Светлый, средний	400	200	200

Для создания и поддержания благоприятных условий освещения для операторов ПЭВМ, их рабочие места должны соответствовать санитарно-эпидемиологическим правилам СанПиН 2.2.2/2.4.1340-03. Рабочее помещение должно иметь естественное и искусственное освещение, соответствующее показателям, представленным в таблице 6. Для рассеивания естественного освещения следует использовать жалюзи на окнах рабочих помещений. В качестве источников искусственного освещения должны быть использованы люминесцентные лампы, лампы накаливания – для местного освещения [18].

6.2.3 Умственное перенапряжение

Умственное перенапряжение вызывается большим объемом информации, которую надо анализировать, и чтобы избежать умственного перенапряжения необходимо устраивать небольшие перерывы в течение рабочего дня продолжительностью не более 5 минут.

При умственной работе, по сравнению с физической работой потребление кислорода мозгом увеличивается в 15-20 раз. Если для умственной работы требуется значительное нервно-эмоциональное напряжение, то возможны значительные изменения кровяного давления, пульса. Длительная работа этого характера может привести к заболеванию, в частности сердечно-сосудистым и некоторым другим заболеваниям [17].

6.2.4 Монотонный режим работы

При работе с ПЭВМ основным фактором, влияющим на нервную систему программиста или пользователя, является огромное количество информации, которое он должен воспринимать. Это является сложной задачей, которая очень сильно влияет на сознание и психофизическое состояние из-за монотонности работы. Поэтому меры, позволяющие снизить воздействие этого вредного производственного фактора, которые регулируются СанПиН 2.2.2/2.4.1340-03, являются важными в работе оператора ПЭВМ. Они позволяют увеличить производительность труда и предотвратить появление профессиональных болезней.

Организация работы с ПЭВМ осуществляется в зависимости от вида и категории трудовой деятельности. Виды трудовой деятельности разделяются на 3 группы: группа А – работа по считыванию информации с экрана с предварительным запросом; группа Б – работа по вводу информации; группа В – творческая работа в режиме диалога с ПЭВМ. Работа программиста-разработчика рассматриваемой в данной работе относится к группам А и Б, в то время, как деятельность пользователя приложения относится к группе В. Категории трудовой деятельности, различаются по степени тяжести выполняемых работ. Для снижения воздействия рассматриваемого вредного фактора предусмотрены регламентированные перерывы для каждой группы работ – таблица 6.4.

Таблица 32 – Суммарное время регламентированных перерывов в зависимости от продолжительности работы, вида категории трудовой деятельности с ПЭВМ [18]

Категория работы с ПЭВМ	Уровень нагрузки за рабочую смену при видах работ с ПЭВМ			Суммарное время регламентированных перерывов, мин.	
	группа А, количество знаков	группа Б, количество знаков	группа В, ч	при 8-часовой смене	при 12-часовой смене
I	до 20 000	до 15 000	до 2	50	80
II	до 40 000	до 30 000	до 4	70	110
III	до 60 000	до 40 000	до 6	90	140

Для предупреждения преждевременной утомляемости пользователей ПЭВМ рекомендуется организовывать рабочую смену путем чередования работ с использованием ПЭВМ и без него. В случаях, когда характер работы требует постоянного взаимодействия с компьютером (работа программиста-разработчика) с напряжением внимания и сосредоточенности, при исключении возможности периодического переключения на другие виды трудовой деятельности, не связанные с ПЭВМ, рекомендуется организация перерывов на 10–15 мин. через каждые 45–60 мин. работы. При высоком уровне

напряженности работы рекомендуется психологическая разгрузка в специально оборудованных помещениях [18].

6.3 Опасные производственные факторы

6.3.1 Опасность поражения электрическим током

Поражение электрическим током является опасным производственным фактором и, поскольку программист имеет дело с электрооборудованием, то вопросам электробезопасности на его рабочем месте должно уделяться особое внимание. Нормы электробезопасности на рабочем месте регламентируются СанПиН 2.2.2/2.4.1340-03, вопросы требований к защите от поражения электрическим током освещены в ГОСТ Р 12.1.019-2009 ССБТ.

Электробезопасность – система организационных и технических мероприятий и средств, обеспечивающих защиту людей от вредного и опасного воздействия электрического тока, электрической дуги, электромагнитного поля и статического электричества.

Опасность поражения электрическим током усугубляется тем, что человек не в состоянии без специальных приборов обнаружить напряжение дистанционно.

Помещение, где расположено рабочее место оператора ПЭВМ, относится к помещениям без повышенной опасности ввиду отсутствия следующих факторов: сырость, токопроводящая пыль, токопроводящие полы, высокая температура, возможность одновременного прикосновения человека к имеющим соединение с землей металлоконструкциям зданий, технологическим аппаратам, механизмам и металлическим корпусам электрооборудования.

Основным организационным мероприятием по обеспечению безопасности является инструктаж и обучение безопасным методам труда, а также проверка знаний правил безопасности и инструкций в соответствии с занимаемой должностью применительно к выполняемой работе.

К мероприятиям по предотвращению возможности поражения электрическим током относятся:

- С целью защиты от поражения электрическим током, возникающим между корпусом приборов и инструментом при пробое сетевого напряжения на корпус, корпуса приборов и инструментов должны быть заземлены;
- При включенном сетевом напряжении работы на задней панели корпуса приборов должны быть запрещены;
- Все работы по устранению неисправностей должен производить квалифицированный персонал;
- Необходимо постоянно следить за исправностью электропроводки [17, 18].

6.3.2 Опасность возникновения пожара

Возникновение пожара является опасным производственным фактором, т.к. пожар на предприятии наносит большой материальный ущерб, а также часто сопровождается травмами и несчастными случаями. Регулирование пожаробезопасности производится СНиП 21-01-97.

В помещениях с ПЭВМ повышен риск возникновения пожара из-за присутствия множества факторов: наличие большого количества электронных схем, устройств электропитания, устройств кондиционирования воздуха; возможные неисправности электрооборудования, освещения, или неправильная их эксплуатация может послужить причиной пожара.

Возможные виды источников воспламенения:

- Искра при разряде статического электричества;
- Искры от электрооборудования;
- Искры от удара и трения;
- Открытое пламя [20].

Для профилактики организации действий при пожаре должен проводиться следующий комплекс организационных мер: должны обеспечиваться регулярные проверки пожарной сигнализации, первичных средств пожаротушения; должен проводиться инструктаж и тренировки по действиям в случае пожара; не должны загромождаться или блокироваться пожарные выходы; должны выполняться правила техники безопасности и

технической эксплуатации электроустановок; во всех служебных помещениях должны быть установлены «Планы эвакуации людей при пожаре и других ЧС», регламентирующие действия персонала при возникновении пожара.

Для предотвращения пожара помещение с ПЭВМ должно быть оборудовано первичными средствами пожаротушения: углекислотными огнетушителями типа ОУ-2 или ОУ-5; пожарной сигнализацией, а также, в некоторых случаях, автоматической установкой объемного газового пожаротушения [20].

6.4 Экологическая безопасность

6.4.1 Влияние объекта исследования на окружающую среду

В данном разделе рассматривается воздействие на окружающую среду деятельности по разработке проекта, а также самого продукта в результате его реализации на производстве.

В ходе выполнения ВКР и дальнейшем использовании алгоритмов отсутствуют выбросы каких-либо вредных веществ в атмосферу и гидросферу, следовательно, загрязнение воздуха и воды не происходит.

Люминесцентные лампы, применяющиеся для искусственного освещения рабочих мест, также требуют особой утилизации, т.к. в них присутствует от 10 до 70 мг ртути, которая относится к чрезвычайно-опасным химическим веществам и может стать причиной отравления живых существ, а также загрязнения атмосферы, гидросферы и литосферы. Сроки службы таких ламп составляют около 5-ти лет, после чего их необходимо сдавать на переработку в специальных пунктах приема.

Во время разработки и написания ВКР образовывался мусор, такой как: канцелярские принадлежности, бумажные отходы, неисправные комплектующие персонального компьютера, люминесцентные лампы.

6.4.2 Мероприятия по защите окружающей среды

Для уменьшения вредного влияния на литосферу необходимо производить сортировку отходов и обращаться в службы по утилизации для дальнейшей переработки или захоронения [28].

В основном, организации, занимающиеся приёмом и утилизацией ртути содержащих отходов, принимают люминесцентные лампы в массовых количествах. Лампа состоит из электронного блока — выгодный компонент для реставрации и утилизации; колба и цоколь также ценное сырьё. По стране утилизацией «ртутных» ламп занимаются более 50 фирм, но единственное их условие — деньги, которые вы должны заплатить за вывоз.

Такие лампы нельзя выкидывать в мусоропровод или уличные контейнеры, а нужно отнести в свой районный ДЕЗ (Дирекция единичного заказчика) или РЭУ (Ремонтно-эксплуатационное управление), где есть специальные контейнеры. Там они принимаются бесплатно, основанием должна служить утилизация в соответствии с Управлением Федеральной службы по надзору в сфере защиты прав потребителей и благополучия человека по Томской области. Пункты приёма отработавших свой срок люминесцентных ламп по городам можно найти в интернете [29].

Переработка макулатуры представляет собой многоэтапный процесс, цель которого заключается в восстановлении бумажного волокна и, зачастую, других компонентов бумаги (таких как минеральные наполнители) и использование их в качестве сырья для производства новой бумаги.

Организации, занимающиеся покупкой сломанных компьютеров на запчасти, готовы платить за запчасти деньги, которые они сэкономят на покупке новых деталей, необходимых для ремонта. Такие организации принимают даже битую и залитую чем-то технику. Компьютерная техника (или ее компоненты) может также заинтересовать тех, кто скупает старые платы и радиодетали для получения из них после переработки драгоценных и редких металлов. Многие сетевые гипермаркеты электронной техники периодически устраивают программу утилизации. Условия такие: за старую бытовую технику вам предложат неплохую скидку на последующую покупку в этом магазине. Также можно самостоятельно отвезти сломанный компьютер в пункт приема металлолома не составит труда. Такие точки приема есть в каждом городе.

6.5 Безопасность в чрезвычайных ситуациях

6.5.1 Основные чрезвычайные ситуации в офисном помещении

Чрезвычайные ситуации бывают техногенного, природного, биологического, социального или экологического характера.

При работе в кабинете могут возникнуть следующие классификации чрезвычайных ситуаций:

- Преднамеренные/непреднамеренные;
- Техногенные: взрывы, пожары, обрушение помещений, аварии на системах жизнеобеспечения/природные – связанные с проявлением стихийных сил природы.

- Экологические – это аномальные изменения состояния природной среды, такие как загрязнения биосферы, разрушение озонового слоя, кислотные дожди/ антропогенные – являются следствием ошибочных действий людей.

- Биологические – различные эпидемии, эпизоотии, эпифитотии;

- Социальные – это обстановка на определенной территории, сложившаяся в результате опасного социального явления, которое повлекло в результате человеческие жертвы, ущерб здоровью, имуществу или окружающей среде;

- Комбинированные.

6.5.2 Типичные чрезвычайные ситуации

Пожар (возгорание)

Наиболее вероятная чрезвычайная ситуация, которая может возникнуть при работе с ПЭВМ – пожар, так как в современных ЭВМ очень высокая плотность размещения элементов электронных схем. В непосредственной близости друг от друга располагаются соединительные провода и кабели, при протекании по ним электрического тока выделяется значительное количество теплоты, при этом возможно оплавление изоляции и возникновение возгорания.

Биологические, так как программист работает в кабинете и контактирует с большим количеством людей, в том числе с другими сотрудниками, то велик риск заражения одного сотрудника от другого (чем больше народу, тем выше

риск). В связи с большим скоплением народа в одном помещении появляется необходимость в непрерывном проветривании, что приводит к образованию сквозняков, что так же может сказаться на здоровье.

Возникновение других видов ЧС – маловероятно [26].

Социальная чрезвычайная ситуация (терроризм)

Терроризм – это метод, посредством которого организованная группа или партия стремятся достичь провозглашенные ими цели через систематическое использование насилия.

Компьютерный терроризм (кибертерроризм) — использование компьютерных и телекоммуникационных технологий (прежде всего, интернета) в террористических целях.

В киберпространстве могут быть использованы различные способы для совершения кибертеракта:

- Получение несанкционированного доступа к государственным и военным секретам, банковской и личной информации;
- Нанесение ущерба отдельным физическим элементам информационного пространства, например, разрушение сетей электропитания, создание помех;
- Использование специальных программ для разрушения аппаратных средств;
- Кража или уничтожение информации, программ и технических ресурсов путем преодоления систем защиты, внедрения вирусов, программных закладок;
- Воздействие на программное обеспечение и информацию;
- Раскрытие и угроза публикации закрытой информации;
- Захват каналов средств массовой информации с целью распространения дезинформации, слухов, демонстрации мощи террористической организации и объявления своих требований;

- Уничтожение или активное подавление линий связи, неправильная адресация, перегрузка узлов коммуникации;
- Проведение информационно-психологических операций.

Использование технологии Wi-Fi Direct позволяет двум и более пользователям передавать информацию любого содержания. Данная система позволяет автоматизировать данный процесс. В связи с этим появляется возможность утечки информации, которую могут использовать террористы.

В качестве передаваемой информации могут быть личные данные, реквизиты, пин-коды банковских карт. Террористы могут использовать эти данные чтобы нанести вред пользователям.

В результате чего, физические лица могут понести финансовые потери, а в худшем случае могут так же пострадать юридические лица.

Действия в результате возникновения чрезвычайной ситуации и мер по ликвидации ее последствий

При работе компьютерной техники выделяется много тепла, что может привести к пожароопасной ситуации. Источниками зажигания так же могут служить приборы, применяемые для технического обслуживания, устройства электропитания, кондиционеры воздуха. Серьёзную опасность представляют различные электроизоляционные материалы, используемые для защиты от механических воздействий отдельных радиодеталей.

В связи с этим, участки, на которых используется компьютерная техника, по пожарной опасности относятся к категории пожароопасных «В».

Меры, соблюдение которых поможет исключить с большой вероятностью возможность возникновения пожара:

- Для понижения воспламеняемости и способности распространять пламя кабели покрывают огнезащитным покрытием;
- При ремонтно-профилактических работах строго соблюдаются правила пожарной безопасности;
- Помещения, в которых должны располагаться ПЭВМ проектируют I или II степени огнестойкости;

– Каждое из помещений, где производится эксплуатация устройств ПЭВМ, должно быть оборудовано первичными средствами пожаротушения и обеспечено инструкциями по их применению. В качестве средств пожаротушения разрешается использование углекислотного огнетушителя типа ОУ-2, ОУ-5(описание ниже), а также порошковый тип. Применение пенных огнетушителей не допускается, так как жидкость пропускает ток;

– Устройства ПЭВМ необходимо устанавливать вдали отопительных и нагревательных приборов (расстояние не менее 1 м и в местах, где не затруднена их вентиляция и нет прямых солнечных лучей);

– Разрабатываются организационные меры по обучению персонала навыкам ликвидации пожара имеющимися в наличии средствами тушения пожара до прибытия пожарного подразделения [22].

При пожаре люди должны покинуть помещение в течение минимального времени.

В помещениях с компьютерной техникой, недопустимо применение воды и пены ввиду опасности повреждения или полного выхода из строя дорогостоящего электронного оборудования.

Для тушения пожаров необходимо применять углекислотные и порошковые огнетушители, которые обладают высокой скоростью тушения, большим временем действия, возможностью тушения электроустановок, высокой эффективностью борьбы с огнем. Воду разрешено применять только во вспомогательных помещениях [21].

6.6 Правовые и организационные вопросы обеспечения безопасности

6.6.1 Описание правовых норм для работ, связанных с работой за ПЭВМ

Регулирование отношений между работником и работодателем, касающихся оплаты труда, трудового распорядка, особенности регулирования труда женщин, детей, людей с ограниченными способностями и проч., осуществляется законодательством РФ, а именно трудовым кодексом РФ.

Нормальная продолжительность рабочего времени не может превышать 40 часов в неделю.

Порядок исчисления нормы рабочего времени на определенные календарные периоды (месяц, квартал, год) в зависимости от установленной продолжительности рабочего времени в неделю определяется федеральным органом исполнительной власти, осуществляющим функции по выработке государственной политики и нормативно-правовому регулированию в сфере труда.

Продолжительность ежедневной работы (смены) не может превышать:

- Для работников в возрасте от 15 до 16 лет – 5 часов, в возрасте от 16 до 18 лет – 7 часов;
- Для учащихся общеобразовательных учреждений, образовательных учреждений начального и среднего профессионального образования, совмещающих в течение учебного года учебу с работой, в возрасте от 14 до 16 лет – 2,5 часа, в возрасте от 16 до 18 лет – 4 часов;
- Для инвалидов – в соответствии с медицинским заключением, выданным в порядке, установленном федеральными законами и иными нормативными правовыми актами российской федерации.

Для работников, занятых на работах с вредными и (или) опасными условиями труда, где установлена сокращенная продолжительность рабочего времени, максимально допустимая продолжительность ежедневной работы (смены) не может превышать:

- При 36-часовой рабочей неделе - 8 часов;

– При 30-часовой рабочей неделе и менее - 6 часов.

Продолжительность работы (смены) в ночное время сокращается на один час без последующей отработки. К работе в ночное время не допускаются: беременные женщины; работники, не достигшие возраста 18 лет, за исключением лиц, участвующих в создании и (или) исполнении художественных произведений, и других категорий работников в соответствии с настоящим Кодексом и иными федеральными законами.

В течение рабочего дня (смены) работнику должен быть предоставлен перерыв для отдыха и питания. Время предоставления перерыва и его конкретная продолжительность устанавливаются правилами внутреннего трудового распорядка или по соглашению между работником и работодателем.

Всем работникам предоставляются выходные дни (еженедельный непрерывный отдых).

Организация-работодатель выплачивает заработную плату работникам. Возможно удержание заработной платы только в случаях, установленных ТК РФ ст. 137. В случае задержки заработной платы более чем на 15 дней, работник имеет право приостановить работу, письменно уведомив работодателя.

Законодательством РФ запрещена дискриминация по любым признакам и принудительный труд [22].

Если пользователь постоянно загружен работой с ЭВМ, приемлемой является поза сидя. В положении сидя основная нагрузка падает на мышцы, поддерживающие позвоночный столб и голову. В связи с этим при длительном сидении время от времени необходимо сменять фиксированные рабочие позы.

Исходя из общих принципов организации рабочего места, в нормативно-методических документах сформулированы требования к конструкции рабочего места.

Основными элементами рабочего места программиста являются: рабочий стол, рабочий стул (кресло), дисплей, клавиатура, мышь; вспомогательными - пюпитр, подставка для ног [24].

Взаимное расположение элементов рабочего места должно обеспечивать возможность осуществления всех необходимых движений и перемещений для эксплуатации и технического обслуживания оборудования [24].

Рабочие места с ЭВМ должны располагаться на расстоянии не менее 1,5 м от стены с оконными проемами, от других стен – на расстоянии 1 м, между собой – на расстоянии не менее 1,5 м. При размещении рабочих мест необходимо исключить возможность прямой засветки экрана источником естественного освещения.

При размещении ЭВМ на рабочем месте должно обеспечиваться пространство для пользователя величиной не менее 850 м. Для стоп должно быть предусмотрено пространство по глубине и высоте не менее 150 мм, по ширине – не менее 530 мм. Располагать ЭВМ на рабочем месте необходимо так, чтобы поверхность экрана находилась на расстоянии 400 – 700 мм от глаз пользователя. Конструкция рабочего места и взаимное расположение всех его элементов (сиденье, органы управления, средства отображения информации и т.д.) должны соответствовать антропометрическим, физиологическим и психологическим требованиям, а также характеру работы [25].

Рабочее кресло обеспечивает поддержание рабочей позы в положении сидя, и чем длительнее это положение в течение рабочего дня, тем жестче должны быть требования к созданию удобных и правильных рабочих сидений.

Высота поверхности сиденья должна регулироваться в пределах 400 – 550 мм. Ширина и глубина его поверхности должна быть не менее 400 мм. Поверхность сиденья должна быть плоской, передний край – закругленным. Сиденье и спинка кресла должны быть полумягкими, с нескользящим, не электризующимся и воздухопроницаемым покрытием, материал которого обеспечивает возможность легкой очистки от загрязнения.

Опорная поверхность спинки стула должна иметь высоту 280 – 320 мм, ширину – не менее 380 мм и радиус кривизны горизонтальной плоскости – 400 мм. Расстояние спинки от переднего края сиденья должно регулироваться в пределах 260 – 400 мм.

Рабочее место должно быть оборудовано устойчивой и просто регулируемой подставкой для ног, располагающейся, по возможности, по всей ширине отводимого участка для ног. Подставка должна иметь ширину не менее 300 мм, глубину не менее 400 мм, регулировку по высоте до 150 мм и по углу наклона опорной поверхности подставки до 20. Поверхность подставки должна быть рифленой, по переднему краю иметь бортик высотой 10 мм.

При организации рабочего пространства необходимо учитывать индивидуальные антропометрические параметры пользователя с соответствующими допусками на возможные изменения рабочих поз и потребность в перемещениях.

Рациональной рабочей позой может считаться такое расположение тела, при котором ступни работника расположены на плоскости пола или на подставке для ног, бедра сориентированы в горизонтальной плоскости, верхние части рук – вертикальный угол локтевого сустава колеблется в пределах 70 – 90, запястья согнуты под углом не более чем 20, наклон головы – в пределах 15 – 20, а также исключены частые ее повороты [24].

6.7 Влияние реализации системы на конечных пользователей

Основным направлением реализации разработанного продукта является применение его в качестве системы передачи данных, которая позволит передавать данные с помощью технологии Wi-Fi Direct между двумя и более мобильных устройств, что позволит коммуницировать пользователям в труднодоступных зонах, или где отсутствуют средства телекоммуникации. Пользователем может быть любой человек, установивший мобильное приложение на свое мобильное устройство.

Wi-Fi Direct – один из способов передачи данных в одноранговых сетях, где каждый пользователь может быть и клиентом, и сервером. В связи с этим, в системе отсутствует общая точка доступа.

Технология Wi-Fi Direct благодаря своей скорости передачи данных (около 250 мб/с) практически заменяет технологию Bluetooth, которая имеет скорость передачи данных не более 24 мб/с. Bluetooth продолжает

использоваться преимущественно на тех мобильных устройствах, в которых отсутствует технология Wi-Fi Direct.

Одним из основных вариантов использования Wi-Fi Direct, является взаимодействие сотрудников МЧС при выполнении спасательных операций по ликвидации последствий стихийных бедствий. Wi-Fi Direct может быть так же использована для поиска пропавших людей, которые имеют при себе мобильное устройство с данной технологией.

Wi-Fi Direct позволяет обмениваться информацией в местах, где нет стационарных точек доступа. Это могут быть населенные пункты, в которых отсутствуют средства телекоммуникации.

Сам продукт не пригоден для массового использования в любом уголке земного шара. В городах, где содержится большое количество стационарных или беспроводных точек доступа к сети интернет, в приоритете остается использование технологии Wi-Fi.

Заключение

В настоящее время мобильные технологии являются неотъемлемой частью человеческой жизни. Они применяются во многих отраслях: медицина, социальных, экономических, и военных сферах. Применение мобильных технологий позволяет существенно снизить временные и материальные затраты.

В результате выполнения выпускной квалификационной работы поставленная цель была достигнута, путем выполнения поставленных задач:

1. Найдены и проанализированы существующие аналоги.
2. Проанализированы существующие протоколы маршрутизации в сетях Manet.
3. Спроектированы алгоритмы для передачи данных с помощью технологии Wi-Fi Direct.
4. Программно реализованы спроектированные алгоритмы.
5. Протестировано полученное программное обеспечение.

В данной работе исследовались протоколы маршрутизации для мобильных беспроводных сетей. Не смотря на то, что беспроводные сети активно развиваются, большинство протоколов находятся лишь в экспериментальном виде.

Были спроектированы следующие алгоритмы: алгоритм поиска маршрута, алгоритм передачи данных, алгоритм фрагментации данных.

Были спроектированы структуры данных для пакетов в различных ситуациях, таких как: поиск маршрута, запрос маршрута, ответ на запрос маршрута, передача данных, ошибка передачи данных, подтверждение передачи данных.

Разработано мобильное приложение, которое осуществляет передачу данных между мобильными устройствами, используя технологию Wi-Fi Direct.

В финансовой части работы была проведена оценка потенциальных потребителей результатов исследований, сформирован бюджет затрат. Также

были рассмотрены аспекты, связанные с безопасностью труда на рабочем месте, включая вредные и опасные факторы, режимы работы и защита в чрезвычайных ситуациях.

Список публикаций

1. Designing of routing algorithms in autonomous distributed datatransmission system for mobile computing devices with 'WiFi-Direct'technology // 10th International Conference on Mechanical Engineering, Automation and Control Systems 2016, MEACS 2016; Tomsk; Russian Federation; 27October 2016; Code 126826 // IOP Conference Series: Materials Science and Engineering. Vol. 177, Issue 1, 1 March 2017, Article number 012035

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. WiFi-Direct [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/Wi-Fi_Direct WiFi-Direct (дата обращения 27.06.2016)
2. The Dynamic Source Routing Protocol (DSR). [Electronic resource] URL: <https://tools.ietf.org/html/rfc4728#section-6.2> (Date of treatment 30.06.2016)
3. Performance Analysis of AODV, DSR and OLSR in MANET. Department of Electrical Engineering with emphasis on Telecommunication Blekinge Institute of Technology. – Sweden. 2009. P. 25-27
4. Ad hoc On-Demand Distance Vector (AODV) Routing [Electronic resource]. URL: <https://tools.ietf.org/html/rfc3561> (Date of treatment 01.07.2016)
5. Performance Analysis of AODV, DSR and OLSR in MANET. Department of Electrical Engineering with emphasis on Telecommunication Blekinge Institute of Technology. – Sweden. 2009. P. 24-25
6. Optimized Link State Routing Protocol (OLSR) [Electronic resource]. URL: <https://tools.ietf.org/html/rfc3626> (Date of treatment 03.07.2016)
7. Performance Analysis of AODV, DSR and OLSR in MANET. Department of Electrical Engineering with emphasis on Telecommunication Blekinge Institute of Technology. – Sweden. 2009. P. 28-31
8. Designing of routing algorithms in autonomous distributed datatransmission system for mobile computing devices with 'WiFi-Direct' technology // 10th International Conference on Mechanical Engineering, Automation and Control Systems 2016, MEACS 2016; Tomsk; Russian Federation; 27 October 2016; Code 126826 // IOP Conference Series: Materials Science and Engineering. Vol. 177, Issue 1, 1 March 2017, Article number 012035
9. Среда разработки Eclipse [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/Eclipse_ (дата обращения 03.02.2017)

10. Android Studio. URL: https://ru.wikipedia.org/wiki/Android_Studio (Date of treatment 02.07.2016)
11. Обзор средств разработки мобильных приложений под ОС Android. URL: <https://prezi.com/3xp64zmivud4/presentation/> (дата обращения 03.02.2017)
12. Android Studio Vs Eclipse. What are the main differences? URL: <http://www.avocarrot.com/blog/android-studio-vs-eclipse-main-differences/> (Date of treatment 05.02.2017)
13. Wi-Fi Peer-To-Peer. [Electronic resource] URL: <https://developer.android.com/guide/topics/connectivity/wifip2p.html#creating-br> (Date of treatment 07.02.2017)
14. Creating P2P connections with Wi-Fi [Electronic resource]. URL: <https://developer.android.com/training/connect-devices-wirelessly/wifi-direct.html#permissions> (Date of treatment 10.07.2016)
15. Охрана труда. Основы безопасности жизнедеятельности // www.Grandars.ru. URL: <http://www.grandars.ru/shkola/bezopasnost-zhiznedeyatelnosti/ohrana-truda.html> (дата обращения: 11.03.2017).
16. Воздействие шума на человека // GardenWeb. URL: <http://gardenweb.ru/vozdeistvie-shuma-na-cheloveka> (дата обращения: 10.03.2017).
17. СН 2.2.4/2.1.8.562-96. Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории жилой застройки // Библиотека гостей и нормативов. 2016. URL: http://ohranatruda.ru/ot_biblio/normativ/data_normativ/5/5212/ (дата обращения: 11.03.2017).
18. Попов В.М. Психология безопасности профессиональной деятельности: учебное пособие / В. М. Попов; Новосибирский государственный технический университет. – Новосибирск: Изд-во Новосибирского государственного технического университета, 1996 г. – 155 с.
19. СанПиН 2.2.2/2.4.1340-03. Санитарно-эпидемиологические правила и нормы. Гигиенические требования к персональным электронно-

- вычислительным машинам и организации работы // Электронный фонд правовой и нормативно-технической документации. URL: <http://docs.cntd.ru/document/901865498> (дата обращения: 10.03.2017).
- 20.ГОСТ Р 12.1.019-2009 ССБТ. Электробезопасность. Общие требования и номенклатура видов защиты // Электронный фонд правовой и нормативно-технической документации. URL: <http://docs.cntd.ru/document/1200080203> (дата обращения: 11.03.2017).
- 21.Чрезвычайные ситуации при работе с ПЭВМ // Студопедия — Ваша школопедия. URL: http://studopedia.ru/8_107307_osveshchenie-promeshcheniy-vichislitelnih-tsentrov.html (дата обращения: 10.03.2017).
- 22.Долин П.А. Справочник по технике безопасности. М.: Энергоатомиздат, 1984 г. – 824 с.
- 23.Трудовой кодекс Российской Федерации от 30.12.2001 N 197-ФЗ (ред. от 3.07.2016) // Электронный фонд правовой и нормативно-технической документации. URL: <http://docs.cntd.ru/document/901807664> (дата обращения: 11.03.2017).
- 24.ГОСТ Р 50923-96 Дисплеи. Рабочее место оператора. Общие эргономические требования и требования к производственной среде. Методы измерения // Электронный фонд правовой и нормативно-технической документации. URL: <http://docs.cntd.ru/document/1200025975> (дата обращения: 11.03.2017).
- 25.ГОСТ 22269-76 Система "Человек-машина". Рабочее место оператора. Взаимное расположение элементов рабочего места. Общие эргономические требования // Электронный фонд правовой и нормативно-технической документации. URL: <http://docs.cntd.ru/document/1200012834> (дата обращения: 11.03.2017).
- 26.ГОСТ 12.2.032-78 ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования // Электронный фонд правовой и нормативно-технической документации. URL: <http://docs.cntd.ru/document/1200003913> (дата обращения: 11.03.2017).

27. Федеральный закон от 22.07.2008 г. №123 – ФЗ. "Технический регламент о требованиях пожарной безопасности".
28. Постановление Правительства РФ от 03.09.2010 N 681 (ред. от 01.10.2013) "Об утверждении Правил обращения с отходами производства и потребления в части осветительных устройств, электрических ламп, ненадлежащие сбор, накопление, использование, обезвреживание, транспортирование и размещение которых может повлечь причинение вреда жизни, здоровью граждан, вреда животным, растениям и окружающей среде // Консультант Плюс. 2015. URL: http://www.consultant.ru/document/cons_doc_LAW_104420/e1b31c36ed1083efeb6cd9c63ed12f99e2ca77ed/#dst100007 (дата обращения: 03.04.2017).
29. Как утилизировать люминесцентную лампу?
<http://ecob3.ru/lampalum.html> (дата обращения: 03.04.2017).

Приложение А

Раздел 2 Development of algorithmic and software for data transmission in Manet networks

Студент:

Группа	ФИО	Подпись	Дата
8ИМ5А	Никитин Илья Александрович		

Консультант кафедры ИСТ:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент каф. ИСТ	Мирошниченко Е.А.	К.Т.Н.		

Консультант – лингвист кафедры ИЯИК:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель каф. ИЯИК	Горбатова Т.Н.			

1 Analytical review

1.1 Analysis of software products

There are not so many software products that work with Wi-Fi Direct technology in the mobile application market at the moment, and they only implement a user interface for transferring data between two mobile devices. Today, some applications are able to expand the user's opportunities and to offer additional information about the Wi-Fi signal, allowing you to select the point with the best throughput. Unfortunately, official applications that implement the connection between two or more devices do not exist on the world market. Apps Swift-Wi-Fi and SuperBeam are noted as the main analogues of software products that operate under the Android operating system and are products of the "Google Play Market" service.

1.1.1 Swift Wi-Fi

Swift Wi-Fi is an application which has been created by a group of developers "Swift Wi-Fi Team". In this application the hardware plays a big role in the form of additional access points, controllers and antennas. It also includes a widget that shows available Wi-Fi networks and their data. The widget allows you to perform the following functions:

1. Switch Wi-Fi using the ON / OFF button;
2. Connect to the desired Wi-Fi access point;
3. When you click on the access point, it shows the security of the access point, connection speed, signal strength and IP address;
4. Manage the Wi-Fi access point (Delete unused access points);
5. Add a new Wi-Fi connection (supports Open, WEP, WPA / WPA2 PSK and 802.1x EAP);
6. Easy configuration via QRCode (Root rights are required) [1].

1.1.2 SuperBeam | Wi-Fi Direct

SuperBeam is an application which has been created by the "LiveQoS" development team. It allows you to transfer data between Android devices using Wi-Fi Direct technology easily. It allows you to connect two devices using QR-codes

(using the QR-codes available in the program) or NFC, if the devices support it, or by handing the key manually.

The main functions of the SuperBeam application are as follows:

- Binding devices using NFC or QR-code scanning;
- Data transfer through the web-interface to devices that do not have SuperBeam installed;
- Transfer of one or several files of any type (photos, videos, zip files, APK, contacts, etc.);
- History of all has been performed transmission operations;
- Automatic switching to access point mode when working with devices that do not support WiFi direct;
- If WiFi direct fails, an existing WiFi connection between devices can be used;
- For greater compatibility with limited WiFi networks for all types of transmission, port 8080 is used;
- Works with all third-party file managers and photo galleries.

It can be concluded that the applications that use a WiFi Direct as a basic function, only transmit data between two devices and don't differ from each other, except their different interface. Based on this, it can be concluded that the main feature of the application being created is the ability to transfer data between two or more devices using it as the router of the nearest user of the application [2].

1.2 Analysis of routing protocols

Wi-Fi Direct is a relatively new technology for data transmission, but it also has a competitor - the MANET technology.

MANET (Mobile Ad hoc NETWORKS) networks are radio networks with random mobile subscribers that implement fully decentralized control in conditions when there are no base stations or reference nodes. Topology is rapidly changing with random junction of nodes.

Today the overwhelming the majority of terrestrial mobile wireless networks have a fixed infrastructure and are connected between themselves with the help of

various wired or radio relay channels of data transmission. In the last decade, much attention has been paid to the creation of mobile packet radio networks, which do not have a fixed infrastructure - fixed (Ad Hoc) networks and mobile subscribers (MANET).

These networks are self-organizing, since their nodes are not only ultimate user terminals, but also are relay-routers, relaying packets of other subscribers and participating in finding routes to them, hence they are capable of self-organization. These networks can consist of tens, hundreds and even thousands of nodes.

Peer-to-peer networks without infrastructure are the opposite of networks with a hierarchical structure and centralized management. Each node has a complex of software and hardware that allows you to transfer data from the source to the recipient directly in the physical presence of such a path. This allows you to distribute the load to the network and increase the total network bandwidth. This transfer of data from one subscriber to another can occur, even if these nodes are out of the zone of direct radio visibility. In these cases, the data packets of these subscribers are relayed by other nodes of the network, which have a connection with the corresponding subscribers. Networks with multiple retransmissions are called multi-span or multihop. The main problems in the development of such networks are the routing of packets from the source node to the recipient node, and the scalability of networks, and the addressing of terminal devices, and the maintenance of connectivity in variable topology.

The MANET technology is based on a large quantity of routing protocols with complex algorithms for any situations. These protocols include DSR, AODV, OLSR. Each protocol has its own algorithm for finding the route, data transfer, as well as its own data structure.

Despite the fact that the MANET technology is not yet in use and all the protocols are in an experimental state, they can be partially used and adapted to the system being developed [3].

1.2.1 Protocol DSR

Protocol DSR (Dynamic Source Routing Protocol) is a simple and effective routing protocol and is designed to work with wireless devices.

DSR regularly updates its route cache if new available simple routes have appeared. If any new available routes have been found, the node directs the packet to this route. The package should be aware of the direction of the route. Thus, the route information is indicated in the packet to reach the destination from its source. DSR has two main functions for proper operation, i.e. Route of request and maintenance of the route. In opening a route, it has two messages, i.e. Request Route (RREQ) and Route Response (RREP). When a node wants to send a message to a specific destination, it broadcasts the RREQ packet on the network. Neighboring nodes in the broadcast band receive this RREQ message and add their own address, and again relay it on the network. If the RREQ messages don't reach the destination node, the route is deleted.

Each node maintains its own route cache, which is stored in memory for the detected route. The node will check its route cache for the desired destination node before relaying the RREQ message. By supporting the route cache on each node on the network, the memory overhead that is generated using the route discovery procedure is reduced. If the route is in this node's cache route cache, then it will not retransmit the RREQ in the entire network. The RREQ message will be immediately broadcast for the destination node. The first message that has reached the destination has complete information about the route. This node will send the RREP packet to the source, which has complete information about the route. This route will be considered the shortest route by which the RREQ packet was broadcast. The source node now has full path information in its route cache and can begin sending packets [4].

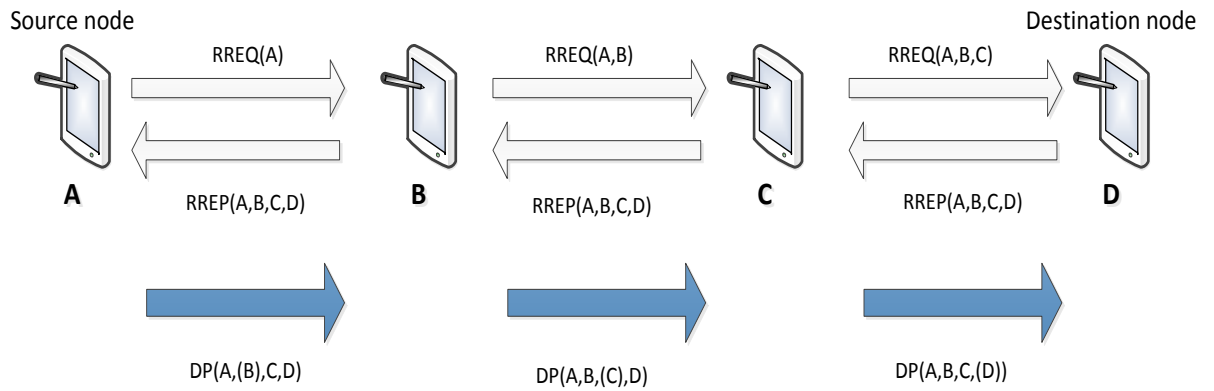


Figure 1 - Search for a route using the DSR protocol

In figure 1 shows the procedure for finding a route. Here we have four nodes, i.e. A, B, C, and D, node A is the source, node D is the destination. When node A wishes to send a data packet to node D, it first checks its route cache, whether it which has a direct route to node D or not. If node A don't have a direct route to node D, it will broadcast the RREQ message on the network. Neighboring node B will receive a RREQ message. The Node B checks its route cache whether there is a direct route to the destination node D or not, if it finds a route to the destination node D, it sends a RREP message to the node A.

In response to this message, the source node will start sending data packets (DP) over the detected route. If he did not find the route from node B to node D, he sends the RREQ message to the next node C and stores route A, B in the cache. The process continues until the RREQ message reaches the target node D. The destination node D caches the routes AB, BC and CD in its memory and sends a RREP message to the original node A.

The next function is to maintain the route. Route maintenance uses two types of messages, an error message (RERR) and an acknowledgment (ACK). The ACK sends a message to the source node if the destination node which has been reached. All intermediate nodes also receive information about the newly constructed route. If an error occurs during the route search, the source node receives an error message (RRER). This occurs if the source node was unable to receive the ACK message.

Thus, the source receives the RERR packet in order to resend the request their route search. After receiving the RERR message, the nodes delete the route entries.

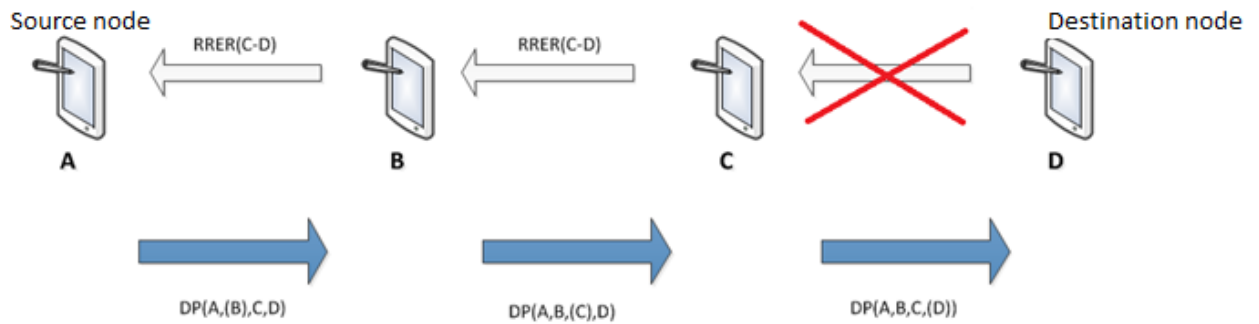


Figure 2 – Messages about confirmation and error in the DSR protocol

In figure 2 shows four nodes A, B, C and D.

The Node A sends a message to the destination node D. The message goes to node C. When node C forwards the RREQ message to node D and don't receive the ACK confirmation message from node D, node C understands that there are some transmission problems.

Thus, node C sends an RRER message to source node A, which allows node A to begin a new route search to node D [5].

1.2.2 Protocol AODV

AODV (Ad hoc On-Demand Distance Vector) is a routing protocol on demand. The AODV algorithm provides an easy way to get information about changing the situation when sending data packets. Thus, if the package was not sent successfully, the affected node is identified and a notification is sent to all the affected sites. This notification cancels all routes through the affected node. In AODV, the routes are unidirectional, from source to destination, and therefore the use of the network is minimal. Since the routes are built on demand, network traffic will also be minimal. AODV don't allow storing additional routes that are not used. If two nodes want to establish a connection in a peer-to-peer network, then AODV allows them to build a multi-hop route. AODV uses the destination node number (DSN) to avoid counting to infinity [6].

When the source node sends the request to the destination, it sends its DSNs along with all the routing information. He also chooses the most profitable route, based on the number of DSNs.

There are three AODV messages Route Request (RREQs), Route Reply (RREPs), and Route Errors (RERRs).

When the source node wants to create a new route to the destination, the requesting node translates the RREQ message to the nearest neighbors.

In Figure 3, the RREQ message is translated from the source node A to the destination node E. The RREQ message is indicated by a white arrow from source node A in two directions. The source node A translates the RREQ message to neighboring nodes.

Each added node in the path to the destination node increases the DSN value by one. Each intermediate node will check whether it which has a direct route to the destination or not. If it has a route, it will direct RREP to the source of node A. If it don't have a direct route to its destination, it will broadcast the RREQ message on the network again with the changed DSN sequence number.

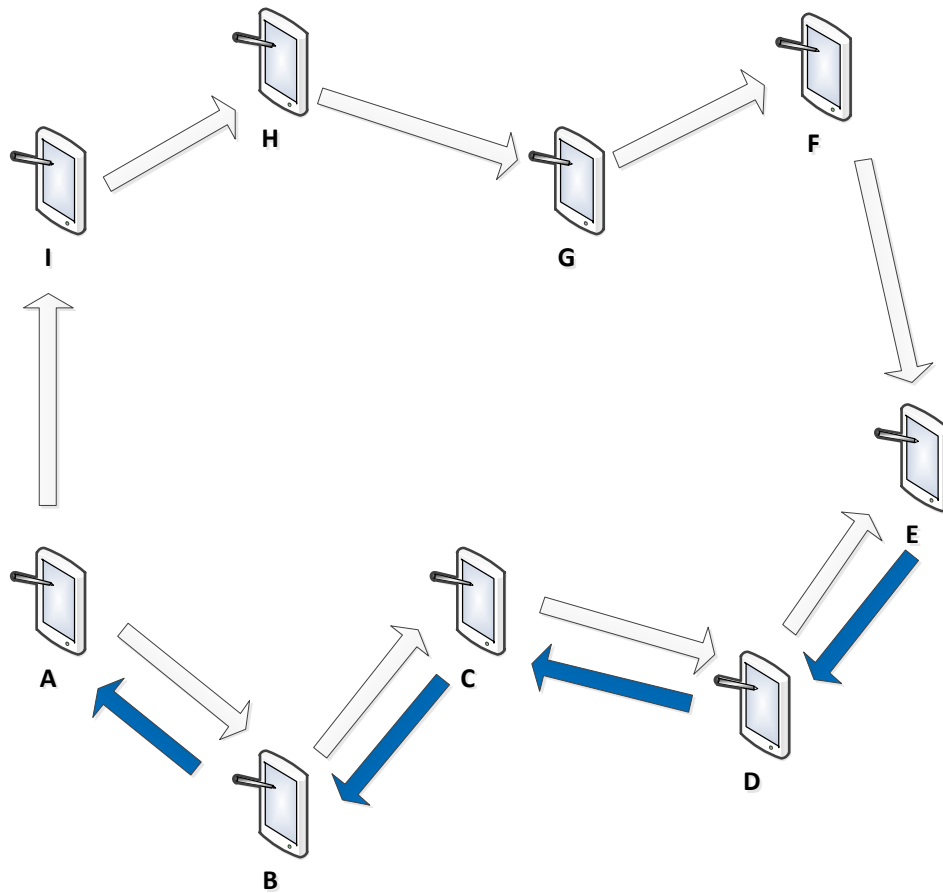


Figure 3 – Route search in AODV protocol

In figure 3 shows the procedure for finding the destination node B. The RREQ message is flooded on the network in search of finding the destination node B. Intermediate nodes can respond to the RREQ message only if they have a DSN number equal to or greater than the number of the packet contained in the header of the RREQ . Intermediate nodes forward the RREQ message to neighboring nodes and write the addresses of these nodes in the routing cache. This information will be used to make a return path for the RREP message from the destination node, as shown in Figure 3.1.

The destination node B responds with a RREP message, denoted by blue arrows, which denotes the shortest path from the destination node B to the source node A. The nodes that receive the RREP message cache information about the constructed route.

If the route search fails, a RERR message is generated. The RERR message contains information about nodes that do not reachable.

All information about routing in the network is stored in the table. Table 1 shows the routing table, which includes the IP-address of destination, the sequence number (DSN), sequence of nodes to the destination node, the network interface, the number of transitions, the timeout [7].

Table 1

IP-address of destination	D SN	Sequence of nodes to destination node	Network interface	Number of conversions	Timeout
---------------------------	------	---------------------------------------	-------------------	-----------------------	---------

1.2.3 Protocol OLSR

OLSR (Optimized Link-State Routing) is a proactive routing protocol also called a table protocol because it constantly stores and updates its routing table.

OLSR monitors the routing tables in order to provide the best route when sending a request for data transfer to the destination node. Due to its nature, OLSR is called a proactive routing protocol. In figure 4 are shown Multipoint Relay (MPR) nodes[8].

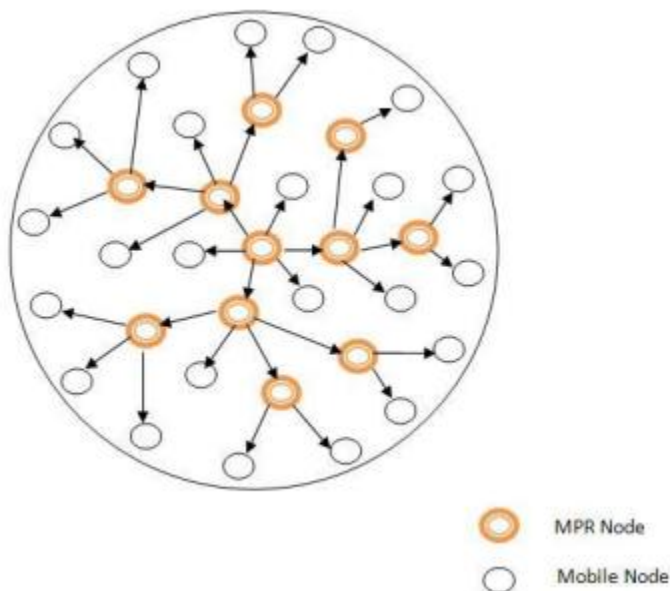


Figure 4 – Multipoint Relay in OLSR

In OLSR, not all nodes exchange information with each other. Multipoint Relay (MPR) hosts provide broadcast route packets. The MPR node is selected as a

neighbor for the source node. Each node on the network stores a list of MPR nodes[9].

This MPR selector is obtained from sending "Hello" packets between neighboring nodes. These routes are constructed before the source node intends to forward the message to the specified destination. Each node on the network stores a routing table. It is for this reason that routing overhead for OLSR is minimal compared to other reactive routing protocols and provides the shortest route to the destination on the network. There's no need to build new routes, because the existing route in use don't lead to an increase in the overhead of enough routing. This reduces the delay in detecting the route. Nodes on the network send "Hello" messages to their neighbors. These messages are sent at a predetermined interval in the OLSR to determine the status of the communication channel.

If node A and node B are neighbors, node A sends a "Hello" message to node B. If node B receives this message, we can say that the link is asymmetric. If the Node B sends the same "Hello" message to the node, it is the same as the first case, called an asymmetric request. Proceeding from this, a two-way communication is established and we can call this connection symmetric. The "Hello" message contains all the information about the neighbors. This allows any node to have a table in which it has information about all its numerous neighbors. The node selects the minimum number of MPR nodes when a symmetric link is established[8].