Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
институт

Вычислительная техника
кафедра

УТВЕРЖДАЮ
Заведующий кафедрой

_____ О.В. Непомнящий
подпись инициалы, фамилия
«___» _____ 2018 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.01 Информатика и вычислительная техника
код и наименование направления

Моделирование протокола маршрутизации для стандарта IEEE 802.15.4/ZigBee
тема

Руководитель _____ доцент, канд. техн. наук Ф.А. Казаков
подпись, дата должность, ученая степень инициалы, фамилия

Выпускник _____ А.В. Шнайдер
подпись, дата инициалы, фамилия

Нормоконтролер _____ доцент, канд. техн. наук В.И. Иванов
подпись, дата должность, ученая степень инициалы, фамилия

Красноярск 2018

СОДЕРЖАНИЕ

ВВЕДЕНИЕ 3
1 Подходы к построению беспроводных сенсорных сетей 4
1.1 Описание протокола IEEE 802.15.4/ZigBee 4
1.2 Основные характеристики протокола IEEE 802.15.4/ZigBee 8
1.3 Алгоритмы работы IEEE 802.15.4/ZigBee 9
1.4 Выводы 10
2 Описание модели поведения протокола 12
2.1 Выводы 15
3 Моделирование протокола 16
3.1 Система имитационного моделирования OMNet++ 16
3.2 Библиотека моделирования беспроводных сенсорных сетей 19
3.3 Построение модели сенсорной сети IEEE 802.15.4/ZigBee 23
3.3.1 Описание моделей сети и компонентов 23
3.3.2 Используемая модель протокола 26
3.4 Реализация предложенных алгоритмов 27
3.4.1 Реализация класса взаимодействия устройств 27
3.4.2 Реализация класса маршрутизации 29
3.5 Выводы 31
4 Результаты моделирования 32
4.1 Описание процесса моделирования 32
4.1 Моделирование сети стандарта IEEE 802.15.4/ZigBee 35
ЗАКЛЮЧЕНИЕ 40
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 41

ВВЕДЕНИЕ

Беспроводная сенсорная сеть (БСС) – это класс беспроводных сетей передачи данных, представляющих собой распределенную самоорганизующуюся и устойчивую к отказу отдельных элементов сеть миниатюрных электронных устройств с автономным источником питания. Область покрытия подобной сети может составлять от нескольких метров до нескольких километров за счет способности ретрансляции сообщений от одного устройства к другому [1].

Беспроводные сенсорные сети являются одним из самых перспективных направлений развития современных телекоммуникационных технологий. Перспективы их использования связаны как с заменой кабельных систем на радиоэфир, так и с новыми функциональными возможностями. Благодаря таким характеристикам БСС, как миниатюрность узлов, низкое энергопотребление, сравнительно невысокая стоимость, стало возможным их широкое использование во многих сферах человеческой деятельности, таких, как:

- мониторинг телекоммуникационной инфраструктуры;
- мониторинг на производстве (вредные выбросы и т.д.);
- экологический, биологический и медицинский мониторинг;
- система «Умный дом»;
- выявление чрезвычайных ситуаций (паводки, землетрясения, лесные пожары и т.д.);
- выявление и предупреждение чрезвычайных ситуаций техногенного характера;
- бортовые управляющие сети и другие.

1 Подходы к построению беспроводных сенсорных сетей

При построении различных технических систем требуются беспроводные сети связи, не обладающие высокой скоростью передачи, но надежные, живучие, способные к самовосстановлению, простые в развертывании и эксплуатации. Для выполнения данных требований были созданы стандарты IEEE 802.15.4 и Zigbee, описывающие устойчивые масштабируемые многошаговые беспроводные сети, простые в развертывании и поддерживающие самые разные приложения.

1.1 Описание протокола IEEE 802.15.4/ZigBee

Стандарт IEEE 802.15.4 задает протокол, который определяет физический слой и управление доступом к среде для беспроводных сенсорных сетей. Является основой для протокола ZigBee, который, в свою очередь, предлагает решение для построения сетей посредством постройки верхних слоёв, которые не регламентируются стандартом.

Сети IEEE 802.15.4/Zigbee, полностью удовлетворяют следующим существенным требованиям [2]:

- благодаря ячеистой (mesh) топологии сети и использованию специальных алгоритмов маршрутизации сеть IEEE 802.15.4/Zigbee обеспечивает самовосстановление и гарантированную доставку пакетов в случаях обрыва связи между отдельными узлами (появления препятствия), перегрузки или отказа какого-то элемента;
- спецификация IEEE 802.15.4/Zigbee предусматривает криптографическую защиту данных, передаваемых по беспроводным каналам, и гибкую политику безопасности;
- устройства IEEE 802.15.4/Zigbee отличаются низким электропотреблением, в особенности конечные устройства, для которых предусмотрен режим «сна», что позволяет этим устройствам работать до трех лет от одной обычной батарейки AA или AAA;

- сеть IEEE 802.15.4/Zigbee – самоорганизующаяся, ее структура задается параметрами профиля стека конфигуратора и формируется автоматически путем присоединения (повторного присоединения) к сети образующих ее устройств, что обеспечивает простоту развертывания и легкость масштабирования путем простого присоединения дополнительных устройств;

- устройства IEEE 802.15.4/Zigbee компактны и имеют низкую стоимость;

- связь в сети IEEE 802.15.4/Zigbee осуществляется путем последовательной ретрансляции пакетов от узла источника до узла адресата;

- сети IEEE 802.15.4/Zigbee предусмотрено несколько альтернативных алгоритмов маршрутизации, выбор которых происходит автоматически.

Стандарт предусматривает возможность использования каналов в нескольких частотных диапазонах. Наибольшая скорость передачи и наилучшая помехоустойчивость достигается в диапазоне от 2,4 до 2,48 ГГц. В этом диапазоне предусмотрено 16 каналов по 5 МГц. Цена, которую пришлось заплатить в сетях IEEE 802.15.4/Zigbee за минимизацию энергопотребления, компактность и дешевизну – относительно низкая скорость передачи данных. Базовая скорость (включая служебную информацию) составляет 250 кбит/с. Средняя скорость передачи полезных данных, в зависимости от загрузки сети и числа ретрансляций, составляет от 5 до 40 кбит/с.

В основе сети IEEE 802.15.4/Zigbee лежит ячеистая топология (mesh-топология). В такой сети, каждое устройство может связываться с любым другим устройством как напрямую, так и через промежуточные узлы сети. Ячеистая топология предлагает альтернативные варианты выбора маршрута между узлами. Сообщения поступают от узла к узлу, пока не достигнут конечного получателя. Возможны различные пути прохождения сообщений, что повышает доступность сети в случае выхода из строя того или иного звена [3].

Сетевой уровень ZigBee (NWK) поддерживает топологии звезда, дерево и сеть. В топологии звезды сеть управляется одним устройством, называемым координатором ZigBee. Координатор ZigBee отвечает за запуск и обслуживание устройств в сети. Все остальные устройства, известные как конечные устройства, напрямую взаимодействуют с координатором ZigBee. В топологии сетки и дерева координатор ZigBee отвечает за запуск сети и выбор определенных ключевых параметров сети, но сеть может быть расширена за счет использования маршрутизаторов ZigBee. В древовидных сетях маршрутизаторы перемещают данные и управляют сообщениями через сеть, используя стратегию иерархической маршрутизации. Древовидные сети могут использовать ориентированную на маяк связь, как описано в спецификации IEEE 802.15.4.

Маршрутизаторы ZigBee в сетчатых сетях в настоящее время не выпускают обычные маяки IEEE 802.15.4. Эта спецификация описывает только сети внутри PAN, то есть сети, в которых связь начинается и заканчивается в пределах одной сети.

Технология IEEE 802.15.4/Zigbee разработана для приложений, одним из ключевых требований которых является низкое энергопотребление [4]. Периоды активности устройств, выполненных по технологии Zigbee/IEEE 802.15.4, могут быть крайне малы, что обеспечивает продолжительный срок службы батарей. Кроме того, стандарт IEEE 802.15.4/Zigbee позволяет разрабатывать беспроводные интерфейсы с минимальными затратами, минимальным количеством внешних пассивных элементов, программным обеспечением стека, использующим отведенный ему объем памяти с высокой эффективностью. Стандарт позволяет создавать сети с многоячейковой топологией, таким образом обслуживая очень большое число узлов и увеличивая дальность связи без дополнительных затрат на усилители мощности.

Технология IEEE 802.15.4/Zigbee не предназначена для передачи больших объемов информации. Однако для передачи, например, показаний

датчиков, объем которых редко превышает десятков байт, не требуется высоких скоростей – в этом случае обязательны высокие показатели по энергопотреблению, цене и надежности. Большинство устройств IEEE Zigbee работают по следующему алгоритму: устройство находится в «спящем» состоянии практически все время, обеспечивая оптимальный режим энергосбережения. При поступлении новой информации либо во время очередного сеанса связи устройство активизируется, быстро передает данные и снова переходит в режим пониженного энергопотребления. Так удается продлить срок службы батарей до 10 лет и более в зависимости от типа приложения и длительности рабочего цикла, причем ток при передаче может составлять порядка 15...30 мА, а в «спящем» режиме – менее 2 мкА [5].

1.2 Основные характеристики протокола IEEE 802.15.4/ZigBee

IEEE 802.15.4 устанавливает два механизма доступа к каналу CSMA/CA, в зависимости от типа конфигурации сети: с использованием маячков и без использования. В сети без маячков используется обычный (бесслотовый) механизм доступа CSMA/CA. Каждый раз, когда устройство собирается начать передачу, оно должно выдержать паузу случайной величины после того, как канал освободится. Случайная задержка нужна потому, что очень вероятно, что многие устройства сети ждут освобождения канала и поэтому после его освобождения могут начать передачу одновременно. Если канал занят, то устройство может повторить попытку после повторной случайной задержки. Фреймы подтверждения о получении посылаются сразу, без использования описанного алгоритма.

В сети с маячками используется слотовый (тактированный) механизм доступа CSMA/CA, в котором начало временного слота должно совпадать с границей суперфрейма сетевого координатора, т.е. начало слота для каждого устройства должно быть синхронизировано с началом передачи маячка сетевым координатором. Поскольку устройство не может начать передачу, пока не найдет маячок, а маячки рассылаются только сетевым координатором, то сетевой координатор с помощью маячков выполняет тактирование актов обмена во всей сети. При этом физический уровень должен обеспечить, чтобы все передачи в сети начинались одновременно с началом слотов. Введение описанной синхронизации позволяет уменьшить вероятность одновременной передачи сообщений несколькими узлами сети.

Для устройств, которые требуют срочной доставки или большой пропускной способности канала, сетевой координатор может зарезервировать часть суперфрейма, в котором будет отсутствовать конкуренция за канал (рисунок 1), поскольку в это время сетевой координатор запрещает любую передачу всем другим устройствам [6]. Эта часть слотов суперфрейма

называется гарантированными временными слотами (Guaranteed Time Slots – GTS).

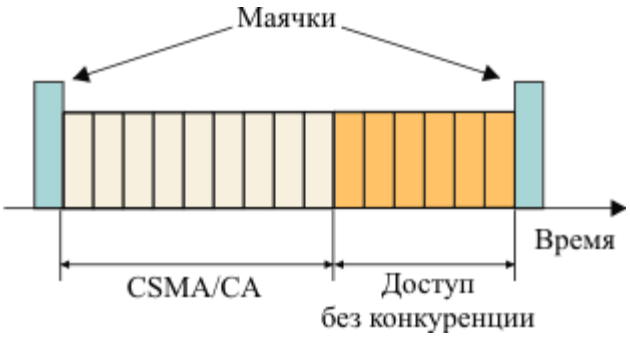

Рисунок 1 – Структура суперфрейма с гарантированными временными слотами

1.3 Алгоритмы работы IEEE 802.15.4/ZigBee

В IEEE 802.15.4 существует три типа обменных процессов:

- передача от устройства к сетевому координатору;
- передача от сетевого координатора к устройству;
- передача между двумя одноранговыми устройствами.

В топологии «звезда» используются только два первых варианта, поскольку в ней не существует обменов между одноранговыми устройствами.

Когда устройство собирается передать данные координатору в сети с маячками, оно сначала пытается обнаружить маячок. Когда маячок найден, устройство подстраивается к структуре суперфрейма. Устройство передает данные координатору, используя слотовый механизм CSMA/CA. В ответ координатор отсылает фрейм уведомления о получении. На этом цикл обмена заканчивается. Если устройство собирается передать данные в сети без маячков, оно передает данные, используя бесслотовый метод CSMA/CA.

Когда координатор желает передать данные устройству в сети с маячками, он помещает в маячок информацию о том, что имеются данные, готовые к передаче. Устройство периодически анализирует содержание маячка и, если в нем имеется информация о наличии сообщения, готового к передаче, устройство передает команду запроса данных, используя слотовый

метод CSMA/CA. Координатор подтверждает прием запроса данных с помощью фрейма уведомления. Вслед за этим координатор отсылает данные, используя слотовый метод доступа CSMA/CA. Устройство подтверждает прием данных отправкой уведомления.

Если координатор собирается передать данные без использования маячка, он запоминает данные и ждет запроса от устройства. Устройство может передать команду запроса данных координатору, используя бесслотовый метод CSMA/CA. Координатор сначала посылает уведомление о получении (в том же цикле обмена), затем, используя CSMA/CA, отсылает данные и получает уведомление о получении от устройства [7].

В период гарантированного доступа к среде передача данных происходит в заранее зарегистрированных слотах – GTS. Количество данных слотов регламентировано стандартом и равно семи. Порядок передачи данных в данном интервале определяет координатор сети.

Любой узел, на основании своих настроек, может запросить у координатора доступ в режиме GTS. Координатор, получив запрос, проверяет наличие свободного слота и в случае возможности выделения подтверждает ответным пакетом. При наличии распределенных гарантированных слотов в конце активного периода координатор резервирует время для доступа без конкуренции. Координатор в маячок помещает список узлов, для которых выделен гарантированный временной слот, с указанием порядкового номера слота GTS. Узел, на основании полученного номера GTS, и времени получения маячка определяет момент времени, в который он может монопольно использовать полосу пропускания.

1.4 Выводы

IEEE 802.15.4/Zigbee является стандартизированной беспроводной технологией, изначально нацеленной на самые широкие приложения

мониторинга и контроля, распределенные сети датчиков, на развертывание беспроводных информационных сетей для недорогих малопотребляющих систем, используемых в промышленной, коммерческой, и бытовой автоматике. Одним из основных преимуществ стандарта IEEE 802.15.4/Zigbee является простота установки и обслуживания подобных устройств.

2 Описание модели поведения протокола

Стандарт IEEE 802.15.4/ZigBee включает в себя описание многих механизмов работы беспроводной сети, основанных на сетевой модели OSI [8].

В данной работе бала проанализирована спецификация ZigBee, от 5 августа 2015 года. Спецификация ZigBee описывает инфраструктуру и сервисы, доступные для приложений, работающих на платформе ZigBee. Передачу данных, механизм доступа к среде, определения типа узлов, построенные топологии, инкапсуляцию, надежность и безопасность. Для достижения поставленной цели рассмотрены способы передача данных и построенные топологии сети, с использованием механизмов маршрутизации.

На основе этого была построена диаграммы взаимодействия подуровней (рисунок 2).

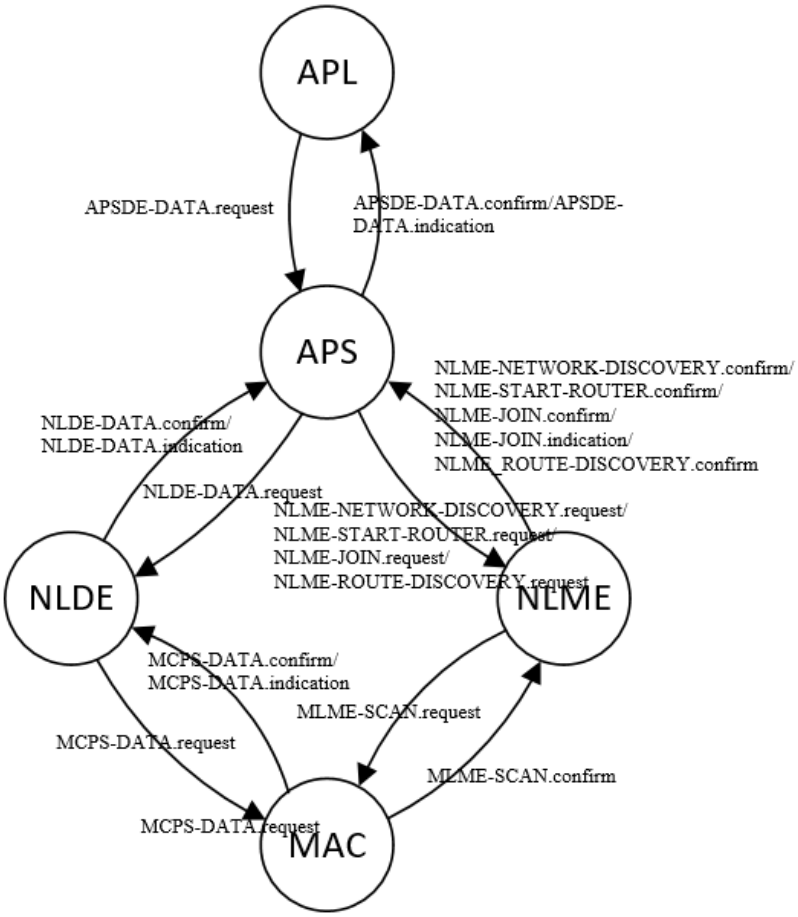

Рисунок 2 – Диаграмма взаимодействия подуровней

В таблице 1 указаны основный атрибуты и описание событий, для взаимодействия подуровней.

Таблица 1 – Описание событий для взаимодействия подуровней

Название	Атрибуты	Описание
APSDE-DATA.request	DstAddrMode, DstAddress, DstEndpoint, ProfileId, SrcEndpoint, RadiusCounter	Этот примитив генерируется локальным NHLE (объект более высокого уровня) всякий раз, когда PDU (блок данных протокола) должен быть передан.
APSDE-DATA.confirm	DstAddrMode, DstAddress, DstEndpoint, SrcEndpoint, Status, TxTime	Этот примитив генерируется локальным объектом APS, в ответ на примитив APSDE-DATA.request. Примитив возвращает статус SUCCESS, указывая, что запрос на передачу был успешным, либо код ошибки NO_SHORT_ADDRESS, NO_BOUND_DEVICE, SECURITY_FAIL, ASDU_TOO_LONG или значения, возвращаемые примитивом NLDE-DATA.confirm.
APSDE-DATA.indication	DstAddrMode, DstAddress, DstEndpoint, SrcAddrMode, SrcAddress, SrcEndpoint, Status,	Этот примитив генерируется подслоем APS, и выдается на более высокий уровень при получении соответствующего адресного кадра данных из уровня NWK, или после получения APSDE-DATA.request, в котором параметр DstAddrMode был установлен в 0x00 и запись таблицы привязки направила кадр самому устройству.
NLDE-DATA.request	DstAddrMode, DstAddr, NsduHandle, DiscoverRoute, SecurityEnable	Этот примитив генерируется локальным объектом APS, всякий раз, когда PDU должны быть переданы объекту APS
NLDE-DATA.confirm	Status. NsduHandle, TxTime	Этот примитив генерируется локальным NLDE, в ответ на прием примитива NLDE-DATA.request. Поле Status будет отображать статус соответствующего запроса.
NLDE-DATA.indication	DstAddrMode, DstAddr, SrcAddr, SecurityUse	Этот примитив генерируется NLDE и выдается подслою APS при получении данных из объекта локального подуровня MAC.
NLME-NETWORK-DISCOVERY.request	ScanChannels, ScanDuration	Этот примитив позволяет следующему более высокому уровню запрашивать, чтобы слой NWK обнаруживал сеть.

Продолжение таблицы 1

Название	Атрибуты	Описание
NLME-NETWORK-DISCOVERY.confirm	Status NetworkCount, NetworkDescriptor,	Этот примитив генерируется NLME и выдается на следующий более высокий уровень по завершении обнаружения NLME-NETWORK-DISCOVERY.request
NLME-START-ROUTER.request	BeaconOrder, SuperframeOrder, BatteryLifeExtension	Этот примитив позволяет более высокому уровню маршрутизатора ZigBee инициировать действия, маршрутизатора, включая маршрутизацию кадров данных, обнаружение маршрута и прием запросов на присоединение к сети с других устройств
NLME-START-ROUTER.confirm	Status	Этот примитив генерируется NLME и выдается на следующий более высокий уровень в ответ на примитив NLME-START-ROUTER.request. Этот примитив возвращает значение статуса INVALID_REQUEST или любое значение статуса, возвращаемое из примитива MLME-START.confirm.
NLME-JOIN.request	ExtendedPANId, RejoinNetwork, ScanChannels,	Этот примитив позволяет следующему более высокому уровню запрашивать присоединение к сети или изменить работу канал.
NLME-JOIN.confirm	Status, NetworkAddress, ExtendedPANID, ActiveChannel	Этот примитив генерируется инициирующим NLME и отправляет его следующему более высокому уровню в ответ на примитив NLME-JOIN.request. Если запрос был успешным, параметр Status будет иметь значение SUCCESS. В противном случае параметр Status указывает код ошибки или любое значение статуса, возвращаемое из примитива MLME-ASSOCIATE.confirm, примитива MLME-SCAN.confirm или примитива PLME-SET.confirm.
NLME-JOIN.indication	NetworkAddress, ExtendedAddress, CapabilityInformation, RejoinNetwork SecureRejoin	Этот примитив позволяет уведомлять следующий более высокий уровень координатора ZigBee или маршрутизатора ZigBee, когда новое устройство успешно присоединилось к своей сети путем объединения или восстановления с использованием процедуры повторного подключения NWK
NLME-ROUTE-DISCOVERY.request	DstAddrMode, DstAddr, Radius NoRouteCache	Этот примитив генерируется следующим более высоким уровнем координатора или маршрутизатора ZigBee и выдается его NLME для запроса инициирования обнаружения маршрута.
NLME_ROUTE-DISCOVERY.confirm	Status, NetworkStatusCode	Этот примитив генерируется NLME и передается на следующий более высокий уровень в результате попытки инициировать обнаружение маршрута

На рисунке 3 показан сценарий для передачи одного кадра данных от отправителя к приемнику без необходимости подтверждения

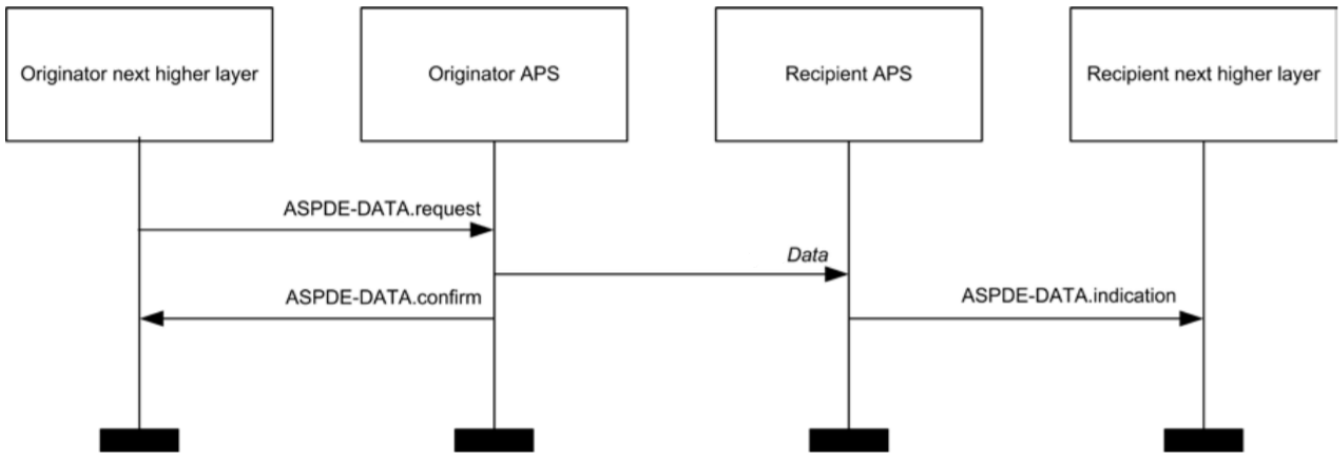

Рисунок 3 – Успешная передача данных без подтверждения

2.1 Выводы

На основе анализа протокола IEEE 802.15.4 можно сделать вывод о том, что протокол содержит большое количество механизмов, как для построения и передачу информации в сети, так и для контроля, управления, взаимодействия, анализа, безопасности, уведомления.

Для достижения поставленной цели рассмотрены способы передача данных и построенные топологии сети, с использованием механизмов маршрутизации.

3 Моделирование протокола

Наиболее эффективным средством для оценки показателей качества инфокоммуникационных систем является имитационное моделирование. Для этой цели в настоящее время разработано большое количество сетевых симуляторов

3.1 Система имитационного моделирования OMNet++

Наиболее эффективным средством для оценки показателей качества инфокоммуникационных систем является имитационное моделирование. Для этой цели в настоящее время разработано большое количество сетевых симуляторов.

Симулятор OMNet++ представляет собой систему моделирования на основе дискретных событий которая может быть использована для таких задач как:

- моделирование проводных и беспроводных коммуникационных систем;
- протоколов моделирования;
- моделирование сетей массового обслуживания.

OMNeT++ – расширяемая, модульная C++ библиотека моделирования на основе компонентов, предназначенная для создания сетевых симуляторов. Это система имитационного моделирования с событийным управлением и с открытым исходным кодом. Она является бесплатной только для академического и некоммерческого использования [9].

OMNeT++ предоставляет IDE на основе Eclipse, графическую среду моделирования, системы сбора и обработки результатов, а также множество других инструментов. Есть расширения для моделирования в реальном времени, эмуляции сети, интеграции с базами данных, с SystemC и ряд других функций.

OMNeT++ не является только сетевым симулятором, эта система используется так же в качестве платформы сетевого моделирования в научном сообществе, а также в промышленных целях. Система OMNeT++ основана на компонентной архитектуре модели. Алгоритмы функционирования базовых компонентов (простейшие модули) реализованы на языке C++, затем могут быть собраны в более крупные составные компоненты и модели с использованием языка высокого уровня (NED).

OMNeT++ предоставляет эффективные инструменты для пользователя для реализации алгоритмов поведения реальных систем любой сложности. Некоторые из главных особенностей:

- иерархически вложенные модули;
- модули являются экземплярами типов модулей;
- модули связываются сообщениями по каналам;
- гибкие параметры модуля;
- язык описания топологии.

Модель OMNeT++ состоит из следующих частей:

- описания топологий на языке NED (файлы с расширением. ned), которые описывают структуру модуля с параметрами, шлюзами и т. д. Такие файлы могут быть созданы с помощью любого текстового редактора, но интерфейс OMNeT++ обеспечивает поддержку графического и текстового редактирования;

- определения сообщений (.msg файлы). Можно определить различные типы сообщений и добавить поля данных в них. Система OMNeT++ автоматически переведет определения сообщений в полноценные классы C++;

- исходные коды простейших модулей представляют собой классы на языке C++.

Программа моделирования может быть скомпилирована как отдельный исполняемый файл так, что она может быть запущена на других компьютерах, где OMNeT++ не установлена.

Процесс построения и запуска программ моделирования представлен на рисунке 4.

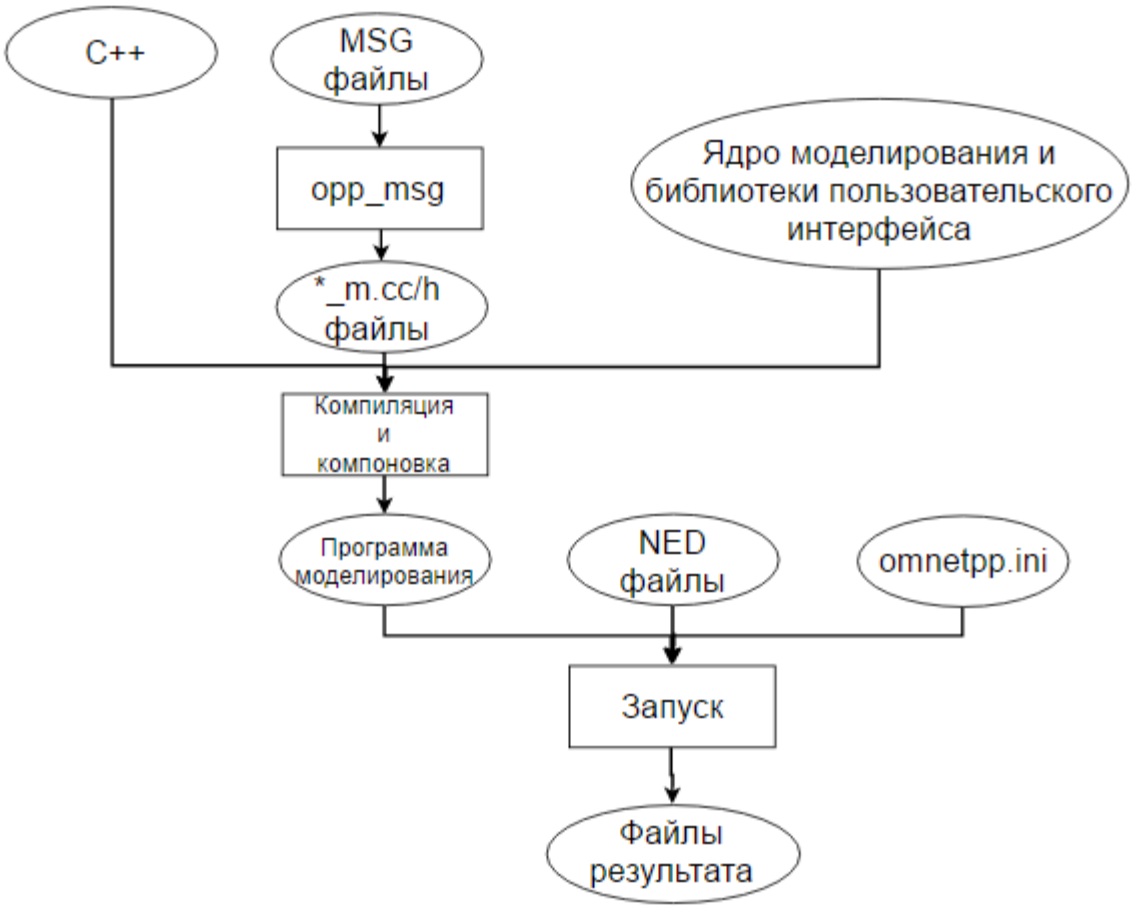

Рисунок 4 – Схема процесса моделирования

Результаты моделирования записываются в файлы результатов: выходные файлы векторов (запись данных во время выполнения моделирования), выходные файлы скаляров (итоговые результаты, вычисленные во время моделирования и записанные после завершения моделирования). Система OMNeT++ содержит богатые возможности для анализа этих файлов, построения различных временных зависимостей и гистограмм. Выходные файлы – это линейно-ориентированные текстовые файлы, что дает возможность обрабатывать их с помощью различных инструментов и языков программирования (в т. ч. MATLAB, GNU R, Perl, Python и программы для работы с электронными таблицами).

Как правило, OMNeT++ используется совместно со специализированными библиотеками моделирования. Наиболее известной и распространенной является библиотека INET. В состав данной библиотеки входят основные протоколы, агенты и другие модели для исследователей и студентов, работающих с сетями передачи данных на базе протоколов tcp/ip. Библиотека INET особенно полезна при разработке и утверждении новых протоколов или изучения новых экзотических сценариев.

3.2 Библиотека моделирования беспроводных сенсорных сетей

Одной из наиболее часто используемых библиотек для моделирования беспроводных сенсорных сетей является библиотека Castalia [10]. Данная библиотека была разработана Австралийским национальным исследовательским центром информационно-телекоммуникационных технологий (Australia's Information and Communications Technology (ICT) Research Centre) и выпущена под Academic Public License.

Используемая в библиотеки модель радиопередающих устройств основана на реальных радиостанциях (с использованием набора микросхем CC2420) для связи с низким энергопотреблением, рассматривает различные типы модуляции и передачи.

Для моделирования беспроводного канала в библиотеки Castalia учитывается мобильность узлов, и расчет опирается на логнормальную (log-normal) модель затенения. Как было показано [11] данная модель обеспечивают точные оценки средних потерь при движении узлов. Castalia также позволяет моделировать в качестве источников данных различные физические процессы.

В библиотеки реализованы функции маршрутизации и протоколов доступа к среде, в том числе IEEE 802.15.4. Кроме того, библиотека Castalia обладает большим числом настраиваемых параметров и предназначена для оценки различных характеристик конкретных приложений.

Библиотека Castalia, не являясь моделью реальных устройств, моделирует поведение обобщённых узлов, и направлена для предварительной оценки технических решений, перед внедрением приложений и алгоритмов для конкретных платформ.

Архитектура связей модулей модели показана на рисунке 5. Библиотека Castalia воспринимает сенсорную сеть как набор узлов, каждый узел опирается на ряд устройств зондирования, которые взаимодействуют с некоторыми физическими процессами с целью извлечения данных из среды.

Каждое устройство считывания может быть связано с одним физическим процессом. Затем, сенсорные узлы обмениваются данными друг с другом с помощью общего канала беспроводной связи.

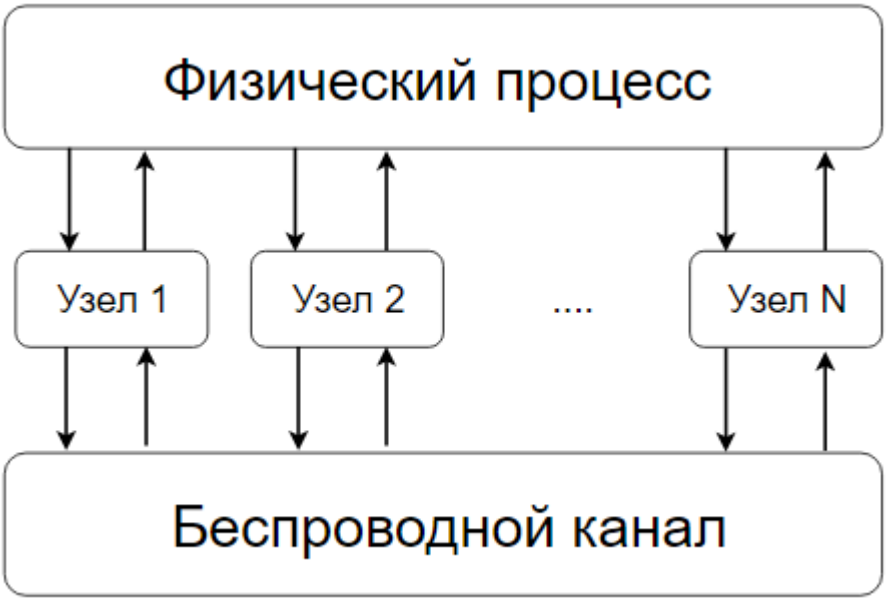

Рисунок 5 – Архитектура связей модели Castalia

Для работы датчиков, Castalia позволяет использовать либо статический процесс с широким набором параметров, либо полагаться на физическую модель процесса. В библиотеки предусмотрено две модели физического процесса: настраиваемый физический процесс и автомобильный физический процесс.

Физический процесс позволяет определить модель процесса в виде формулы определяющий значение измеряемых параметров процесса как функции от времени и места измерения. При моделировании автомобильного физического процесса рассматривается вариант движения источников сигнала по заданному пути.

Результаты моделирования беспроводного канала очень близки к реальным показателям, так как он делает попытку учитывать различные важные особенности реального беспроводного канала.

Основным аспектом моделирования беспроводного канала является оценка средних потерь в тракте передачи между любыми двумя узлами. С этой целью, используется алгоритм логнормального затенения. Это обеспечивает более точное оценку потерь на трассе в децибелах.

Модель беспроводного канала принимает во внимание наличие мобильных узлов датчиков внутри сети. Таким образом, информация о потерях пути между точками в пространстве должны быть сохранены, а не рассчитываться фактически при необходимости. Для того чтобы сделать это, Castalia организует модель пространства как набора дискретных элементов. Затем, рассчитываются потери от каждой ячейки друг с другом. Для учета статических узлов, каждый из них рассматривается как особая клетка.

Каждый узел состоит из ряда различных подмодулей. Рисунок 6 показывает общую архитектуру одного узла, и как его подмодули соединены друг с другом.

Модуль Менеджер сенсоров реализует фактические сенсорные устройства и отвечает за сбор данных из среды через доступные физические процессы. Затем он изменяет данные для того, чтобы ввести искажения из-за реалистичных устройств зондирования. Данные так же могут быть переданы в приложение, работающее на узле датчика.

Модуль Менеджер ресурсов контролирует энергию, затраченную узлом датчика.

Для того чтобы сделать это, модули, реализующие аппаратные устройства (т.е. Радио и Менеджер ресурсов) передают ему, сколько энергии они использовали для выполнения своих функций. Кроме того, модуль рассчитывает общую информацию узла, такую как дрейф часов и минимальный уровень мощности, который потребляет узел.

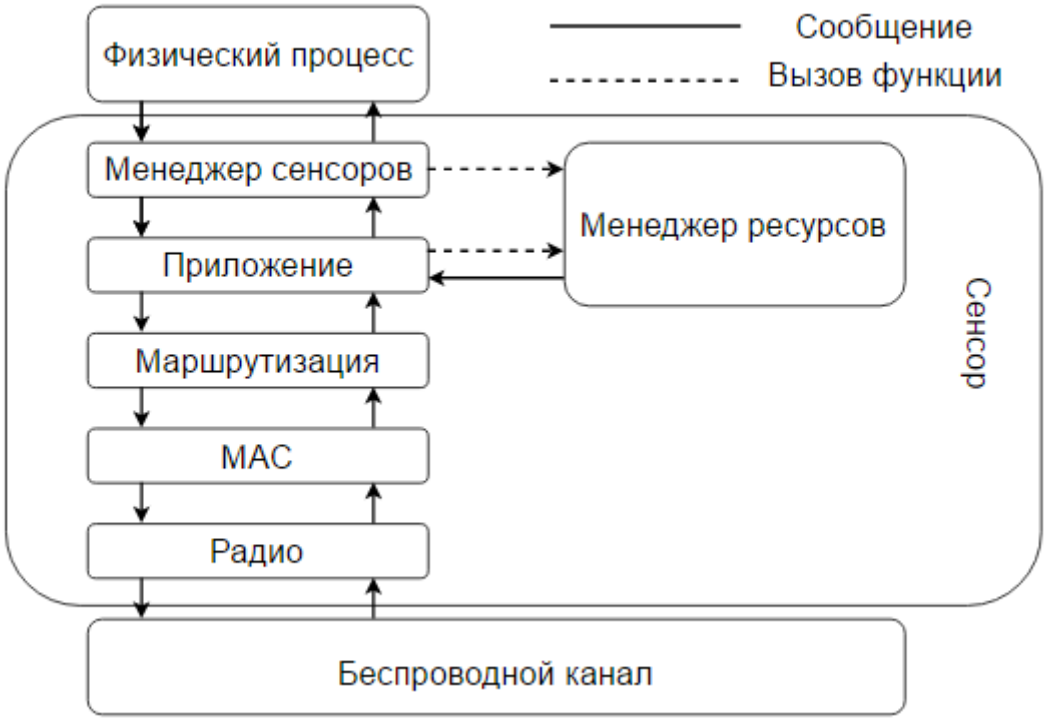

Рисунок 6 – Архитектура модели узла

Модуль Приложение представляет собой программное обеспечение, работающее на узле. Этот слой отвечает за прием измерений от Sensors manager, обработку данных в соответствии со спецификациями приложений, и с формированием пакетов пересылки и приема. В библиотеки Castalia реализованы несколько простых, хотя эффективных приложений, направленных на исследование основных характеристик построенной модели.

Модули Маршрутизация, MAC, и Радио представляет собой фактическую связь стека узлов датчиков. Что касается сетевого уровня, Castalia обеспечивает два протокола маршрутизации: т.е. bypassRouting и multipathRingsRouting. Как подсказывает название, первый не выполняет

каких-либо фактических политик маршрутизации. Второй, multipathRingsRouting реализует основы одного из доступных протоколов маршрутизации.

Протокол определяет пограничный узел, а также организует сетевые узлы в разные уровни, но никто из них не имеет конкретного родителя. Каждый узел получает номер уровня (или номер кольца) во время фазы настройки. Более конкретно, пограничный узел начинает фазу установки путем отправки пакета настройки с уровнем 0. После приема такого пакета, узел добавляет 1 к уровню и передает его на другие узлы. Этот процесс продолжается до тех пор, все подключенные узлы не получат номер уровня. Для того, чтобы отправить пакет к пограничному узлу, сетевой узел фактически передает пакет, который содержит его собственный номер уровня. Затем, любой узел с номером меньшим уровнем ретранслирует пакет. Этот процесс заканчивается, когда пограничный узел принимает пакет.

Что касается уровня MAC, то в библиотеки Castalia реализовано четыре модуля MAC, т.е. TunableMAC, TMAC, IEEE 802.15.4 и BaselineBANMac.

Библиотека моделирования Castalia позволяет пользователям тестировать и проверять алгоритмы, протоколы и приложения для БСС. Предлагаемый для скачивания вариант библиотеки ориентирован для работы с OMNeT++ версии 4.3, однако, в настоящее время для среды моделирование актуальна версия 5.0.

В рамках данной работы была произведена адаптация исходного кода библиотеки для возможности функционирования в актуальной версии системы моделирования.

3.3 Построение модели сенсорной сети IEEE 802.15.4/ZigBee

3.3.1 Описание моделей сети и компонентов

Для моделирования поведения БСС с использованием библиотеки Castalia на базе системы OMNet++ используется типовая сеть, описание

которой приведено в предыдущем разделе. Данная модель предполагает использование узлов с одинаковым набором параметров.

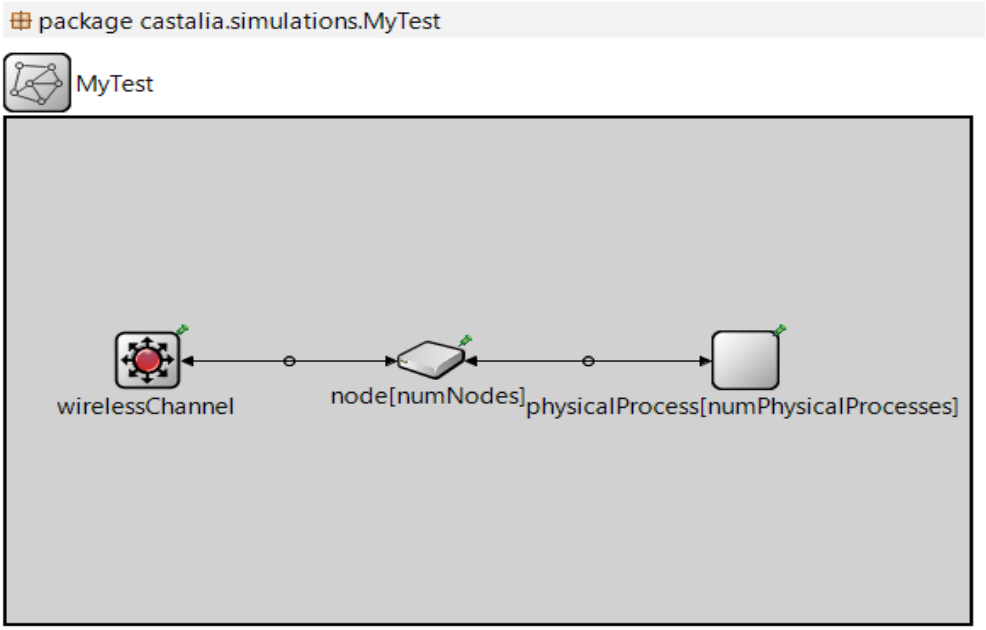

Рисунок 7 – Модель сети

Для моделирования базовая модель была изменена согласно рисунку 7.

Модель узла (Node) представлена как составной объект системы OMNet++, схема узла приведена на рисунке 8. В составе объекта входят компоненты реализующие функции приложения (Application), функции датчиков для считывания состояния процесса (SensorManager), функции учета расходуемых ресурсов – в первую очередь электроэнергии (ResourseManager), функции передачи данных через беспроводную сеть (Communcation) и блок моделирования перемещения узла (MobilityMabager).

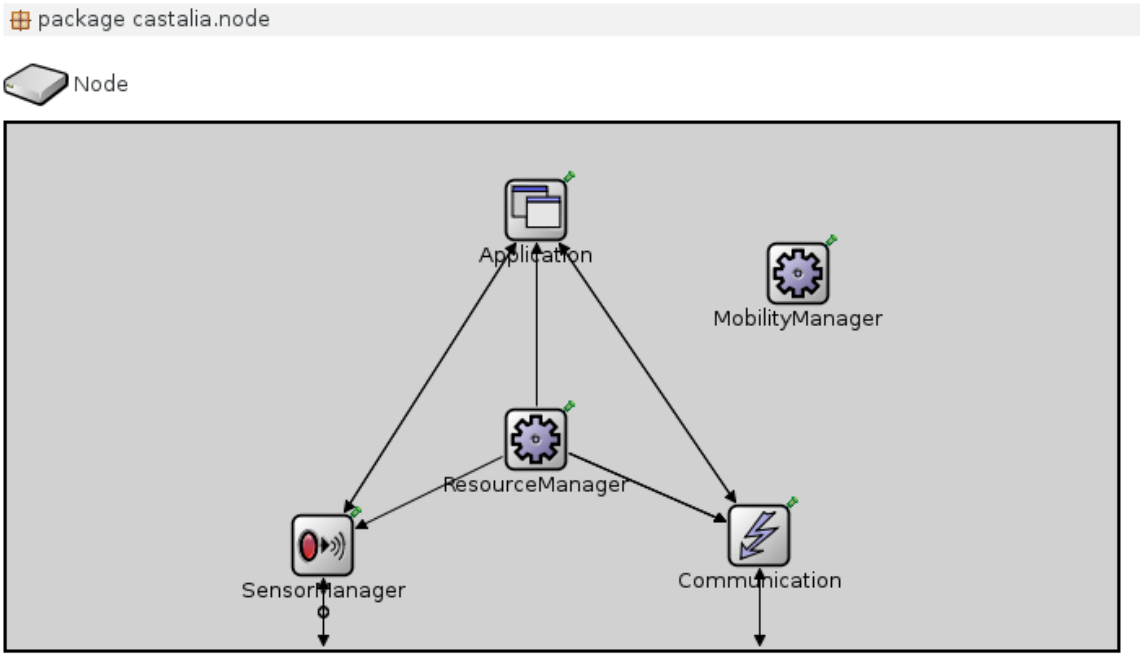

Рисунок 8 – Модель узла

Компонент Communication так же является составным (рисунок 9). Компоненты объекта реализуют функции управления доступом (MAC), доступа к радиоканалу (Radio) и модуль маршрутизации (Routing).

В дальнейшем компоненты модели уже являются простыми и реализуются на языке C++.

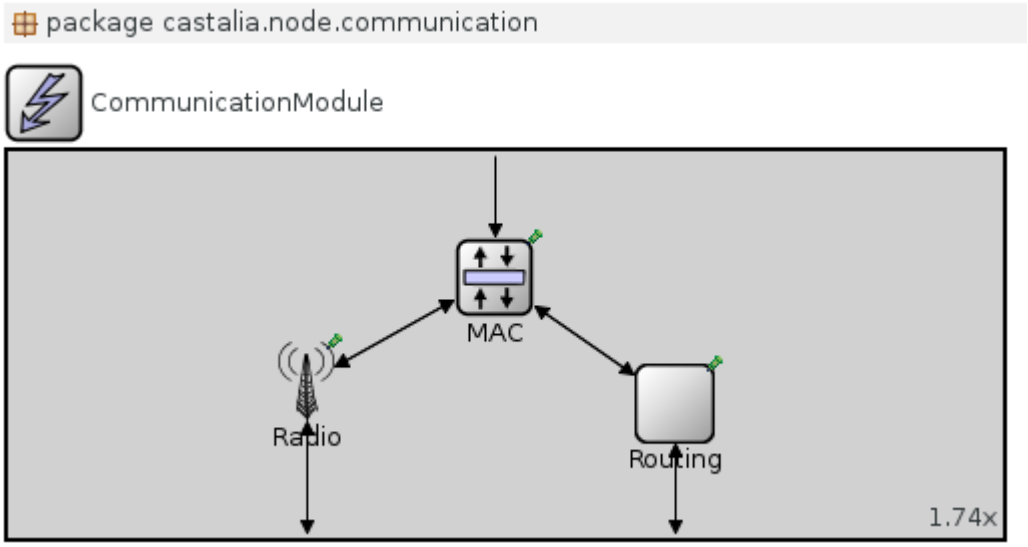

Рисунок 9 – Модель сетевого модуля

3.3.2 Используемая модель протокола

Для обеспечения возможности реализации различных алгоритмов поведения компонентов при визуальном описании составного компонента используются ссылки на простые объекты, которые являются интерфейсными классами.

Используемый объект определяется в настройках модели, при запуске системы моделирования. Пример файла настройки, в котором выбирается тип используемого протокола доступа к среде передачи, приведен на рисунке 10.

```
omnetpp.ini   *omnetpp.ini

include ../Parameters/SensorDevice/Accelerometer.ini

SN.physicalProcessName = "CarsPhysicalProcess"
SN.physicalProcess[*].car_interarrival = 5

SN.node[*].Communication.Radio.RadioParametersFile = "../Parameters/Radio/CC2420.txt"

SN.node[*].ApplicationName = "BridgeTest"
SN.node[0].Application.isSink = true
SN.node[*].Application.reportDestination = "SINK"

MyTest.node[0].Communication.MAC.isFFD = true
MyTest.node[0].Communication.MAC.isPANCoordinator = true

SN.node[*].Communication.RoutingProtocolName = "MultipathRingsRouting"

[Config l200mBridge]
SN.field_x = 200
SN.field_y = 20
SN.deployment = "[1..34]->11x2"
SN.numNodes = 11

SN.physicalProcess[0].point1_x_coord = 0
SN.physicalProcess[0].point1_y_coord = 10
SN.physicalProcess[0].point2_x_coord = 200
SN.physicalProcess[0].point2_y_coord = 10
```

Рисунок 10 – Настройка сетевого модуля

Указание параметра RoutingProtocolName = "MultipathRingsRouting" как раз указывает на необходимость при моделировании использовать объект класса MultipathRingsRouting.

Диаграмма классов, реализующих протокол маршрутизации, представлена на рисунке 11.

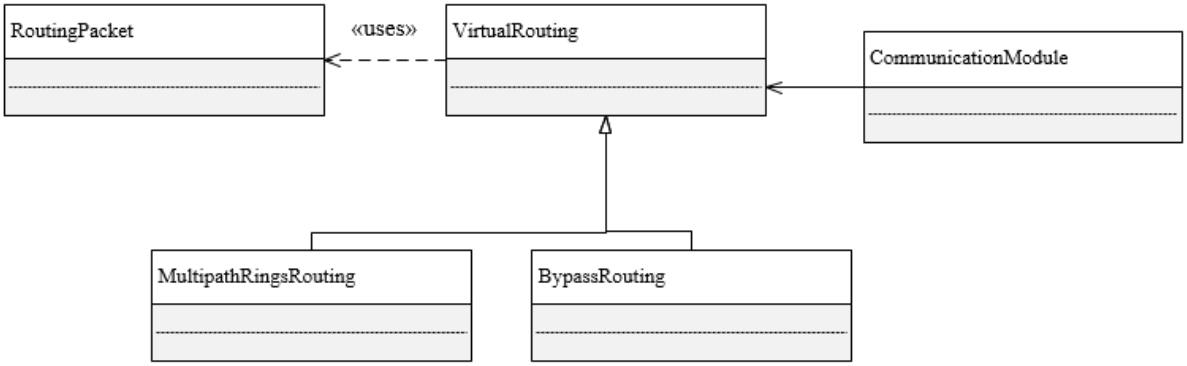

Рисунок 11 – Диаграмма классов модели

3.4 Реализация предложенных алгоритмов

3.4.1 Реализация класса взаимодействия устройств

Взаимодействие в системе моделирования основана на событийном принципе взаимодействия объектов. Для всех объектов модели обязательно должен быть реализован метод ***void handleMessage(cMessage * msg),*** в котором реализуется основной цикл обработки запросов.

Ниже приведено описание базового класса, на основании которого определяются все остальные классы реализации отдельных протоколов маршрутизации.

```
using namespace std;

class VirtualRouting: public CastaliaModule, public TimerService {
 protected:

        int maxNetFrameSize;
        int netDataFrameOverhead;
        int netBufferSize;
        unsigned int currentSequenceNumber;
        double radioDataRate;
        ResourceManager *resMgrModule;

        queue< cPacket* > TXBuffer;
        vector< list< unsigned int> > pktHistory;

        double cpuClockDrift;
        bool disabled;
```

```
        Radio *radioModule;
        string selfAddress;
        int self;

        virtual void initialize();
        virtual void startup() { }
        virtual void handleMessage(cMessage * msg);
        virtual void finish();

        virtual void fromApplicationLayer(cPacket *, const char *) = 0;
        virtual void fromMacLayer(cPacket *, int, double, double) = 0;

        int bufferPacket(cPacket *);

        void toApplicationLayer(cMessage *);
        void toMacLayer(cMessage *);
        void toMacLayer(cPacket *, int);
        bool isNotDuplicatePacket(cPacket *);

        void encapsulatePacket(cPacket *, cPacket *);
        cPacket *decapsulatePacket(cPacket *);
        int resolveNetworkAddress(const char *);

        virtual void handleMacControlMessage(cMessage *);
        virtual void handleRadioControlMessage(cMessage *);
        virtual void handleNetworkControlCommand(cMessage *) { }
        virtual void handleApplicationControlMessage(cMessage *);
};
```

В дальнейшем при реализации модифицированного протокола маршрутизации, реализующий предложенные в работе алгоритмы был создан свой класс ***MultipathRingsRouting.*** Фрагмент описания данного класса приведено ниже.

```
class MultipathRingsRouting: public VirtualRouting {
 private:
***
        int currentSinkID;
        int currentLevel;
        int tmpSinkID;
        int tmpLevel;
        bool isSink;
***
};
```

В указном фрагменте приведены основные атрибуты, отвечающие за функционирование модифицированных алгоритмов. Эти атрибуты имеют

соответствующие параметры в файле описание простого компонента *MultipathRingsRouting* на языке ned и доступны пользователям для настройки модели при запуске.

Описание параметров:

currentSinkID - текущий идентификатор шлюза маршрутизации;

currentLeve - текущий уровень кольца маршрутизации;

tmpSinkID – временный идентификатор шлюза маршрутизации;

tmpLevel – временный уровень кольца маршрутизации;

isSink - логический параметр, указывающий на то, что устройство является шлюзом маршрутизации.

3.4.2 Реализация класса маршрутизации

В рамках реализации поставленных задач не использовались возможности моделирования физических процессов, возможности перемещения узлов, а также возможности по моделированию реального радиоканала. Для оценки возможностей протокола в части распределения пакетов гарантированного доступа достаточно было использование идеальной модели радиоканала, без учета вероятностных параметров затухания и ошибок.

Для того, чтобы информация от узлов шла к шлюзу был разработан модуль маршрутизации ***MultipathRingsRouting***.

Структура пакета для маршрутизации представлена на рисунке 12.

source – адрес отправителя;

destination – адрес получателя;

sequence Number – номер последовательности;

multipath Rings Routing Packet Kind - тип пакета маршрутизации;

sink ID – ID центра маршрутизации;

sender Level – уровень кольца маршрутизации, с которого отправлен пакет.

```
encapsulatedPacket = (MultipathRingsRoutingPacket) Multipath rings routing setup packet (cPacket)
  controlInfo = NULL (cObject)
  encapsulatedPacket = NULL (cPacket)
  netMacInfoExchange (NetMacInfoExchange_type)
  source = '1' [...] (string)
  destination = '-1' [...] (string)
  sequenceNumber = 0 [...] (unsigned int)
  multipathRingsRoutingPacketKind = 2 (MPRINGS_TOPOLOGY_SETUP_PACKET) [...] (int)
  sinkID = 0 [...] (int)
  senderLevel = 1 [...] (int)
```

Рисунок 12 – Структура пакета маршрутизации

Алгоритм передачи информации состоит из двух фаз, которые разделяются типом пакета.

Первая фаза. Установление уровней. Для этого шлюз шлет в канал пакет с типом MPRINGS_TOPOLOGY_SETUP_PACKET. Узел получая такой пакет, если его собственный уровень не установлен, ставит уровень на единицу больше, запоминает как свой уровень и передает пакет дальше. Аналогично остальные узлы. Таким образом вокруг шлюза формируются «кольца уровней».

Вторая фаза. Обмен данными. При отправке сообщения от узла в пакете записывается уровень этого узла. При получении пакета с типом NETWORK_LAYER_PACKET если уровень узла меньше уровня пакета, то пакет передается далее, пока не достигнет шлюза. Иначе если уровень узла больше или равен уровню пакета, то такой пакет игнорируется (как неверное направление). Данный алгоритм помогает повысить эффективность доставки данных к шлюзу и снизить нагрузку на сеть.

3.5 Выводы

В главе были рассмотрены система моделирования OMNet++ и библиотека моделирования беспроводных сенсорных сетей Castalia. Показаны основные правила построения моделей сенсорных сетей. Приведены описания составных и простейших компонентов, используемых в работе.

Для оценки поведения сенсорной сети были разработаны модели узлов. В ходе моделирования был доработан алгоритм маршрутизации.

В заключительной части главы приведены описание класса, который реализуют предлагаемый алгоритм маршрутизации трафика.

4 Результаты моделирования

4.1 Описание процесса моделирования

После создания сети, для которой производится расчет, необходимо задать параметры модели в файле omnet.ini, фрагмент файла приведен ниже.

```
[General]

include ../Parameters/Castalia.ini

sim-time-limit = 100s

include ../Parameters/SensorDevice/Accelerometer.ini

SN.physicalProcessName = "CarsPhysicalProcess"
SN.physicalProcess[*].car_interarrival = 5

SN.node[*].Communication.Radio.RadioParametersFile = "../Radio/CC2420.txt"

SN.node[*].ApplicationName = "BridgeTest"
SN.node[0].Application.isSink = true
SN.node[*].Application.reportDestination = "SINK"

MyTest.node[0].Communication.MAC.isFFD = true
MyTest.node[0].Communication.MAC.isPANCoordinator = true

SN.node[*].Communication.RoutingProtocolName = "MultipathRingsRouting"

[Config Routing]
SN.field_x = 200
SN.field_y = 20
SN.deployment = "[1..34]->11x2"
SN.numNodes = 11

SN.physicalProcess[0].point1_x_coord = 0
SN.physicalProcess[0].point1_y_coord = 10
SN.physicalProcess[0].point2_x_coord = 200
SN.physicalProcess[0].point2_y_coord = 10
```

Файл инициализации модели состоит из секций, каждая из которых определяет набор параметров для отдельного эксперимента. Секция [General] общая для всех экспериментов. При запуске в диалоговом режиме можно выбрать, тот набор параметров, который будем использовать.

При отладке модели и в режиме однократного прогона удобно использовать оконный интерфейс моделирования. Вид окна при моделировании в этом режиме приведен на рисунке 13.

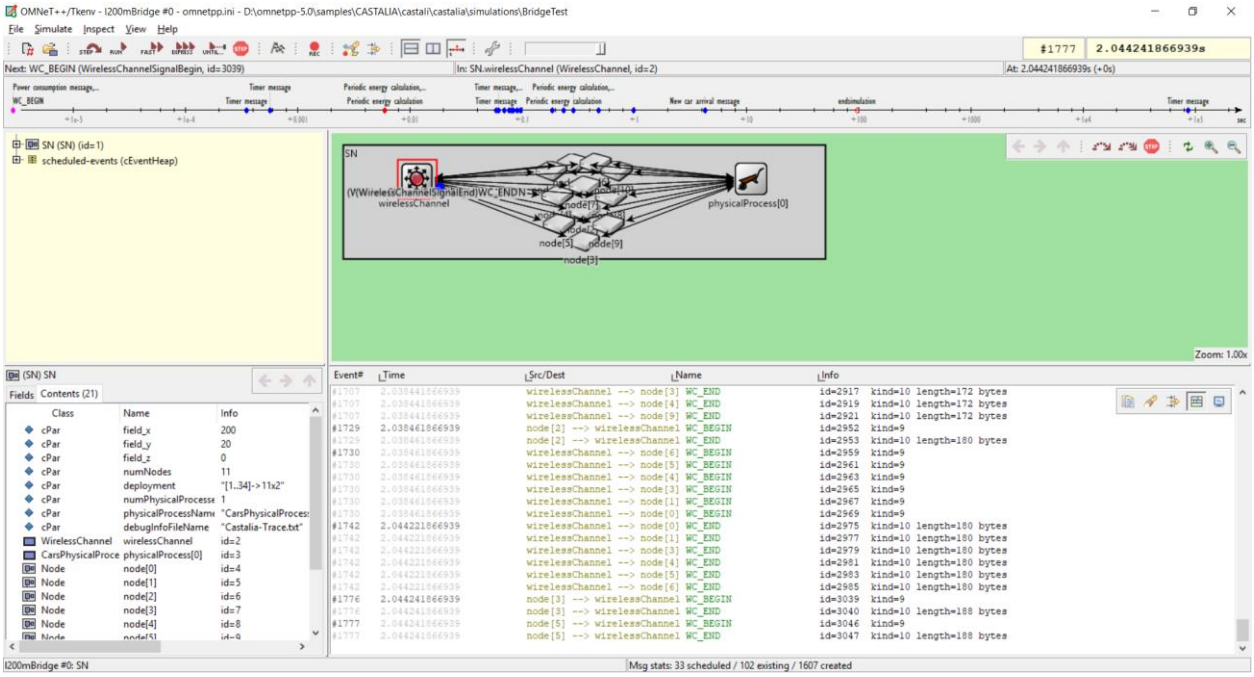

Рисунок 13 – Диалоговое окно моделирования

В таком режиме в верхней части окна отражается временная ось с отмеченными моментами приход тех или иных событий (очередь событий), в левой части окна – состав объектов и текущие значения параметров объектов, в центральной части – схема сети и в режиме мультипликации передача сообщений между объектами, в нижней части – список обработанных сообщений.

При проведении большого набора экспериментов часто бывает неудобно вручную выбирать используемые параметры и наблюдать за процессом моделирования, тогда удобно использовать пакетный режим. Вид окна настройки пакетного режима приведен на рисунке 14.

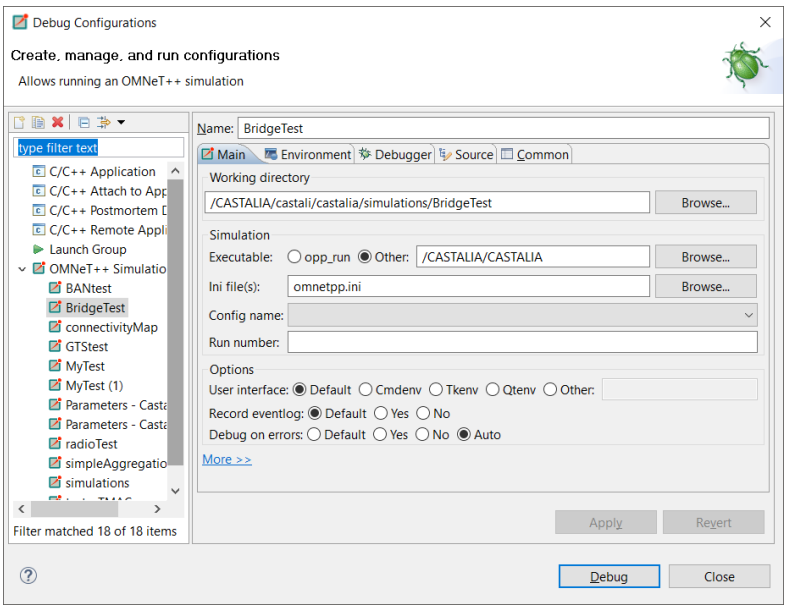

Рисунок 14 – Окно настройки пакетного режима

Результаты моделирования записываются в файл. В учебных или отладочных целях можно записать временную диаграмму взаимодействия объектов. Пример такой диаграммы приведен на рисунке 15. Она демонстрирует процесс работы модели, а именно процесс передачи сообщений (пакетов), управляющих сигналов между объектами.

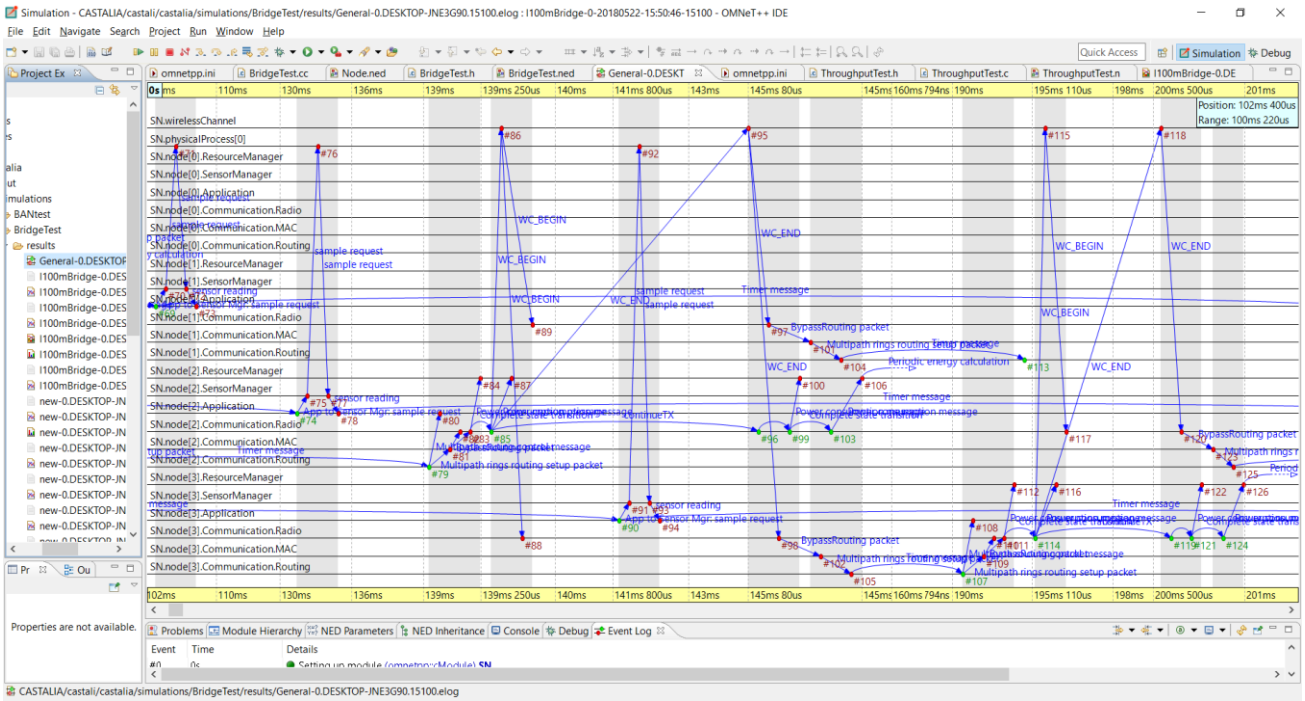

Рисунок 15 – Временная диаграмма взаимодействия объектов

4.1 Моделирование сети стандарта IEEE 802.15.4/ZigBee

Для моделирования сети была составлена диаграмма автомата для передача данных и построенные топологии сети, с использованием механизмов маршрутизации (рисунок 16).

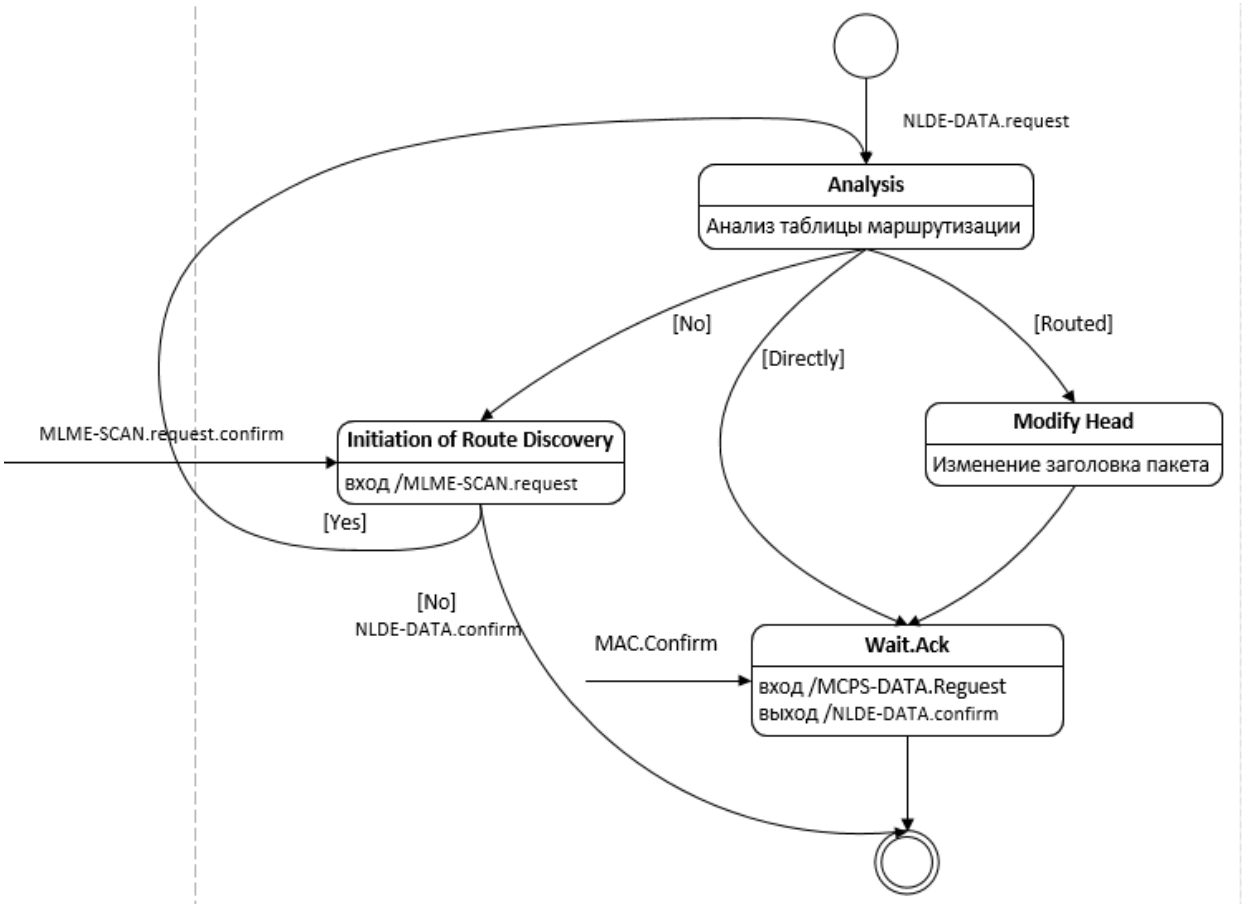

Рисунок 16 – Диаграмма состояний

Для оценки качества полученной модели сети, был проведен анализ экспериментов с использованием набора определенных характеристик, а именно:

- успешная передача с первой попытки – количество пакетов данных, которые были переданы в первый цикл передачи;

- количество потерянных пакетов – количество пакетов данных, которые были потеряны при передаче;

- latency - Время, между тем как данные поступили на устройство и временем выхода из него;

- RTT - Время, затрачиваемое на движение трафика в обоих направлениях;

- delay - Время, необходимое для того, чтобы трафик покинул отправителя и прибыл в пункт назначения.

Данные характеристики были выбраны, так как они являются определяющими при определении такого параметра как QoS (quality of service «качество обслуживания»).

Latency, время нахождения пакета данных на узле, для дальнейшей его пересылки другим узлам.

На рисунке 16, схематично показано отправка пакета, от уровня приложения с анализом заголовка пакета, изменения заголовка в случае необходимости, а также поиск маршрута до узла назначения.

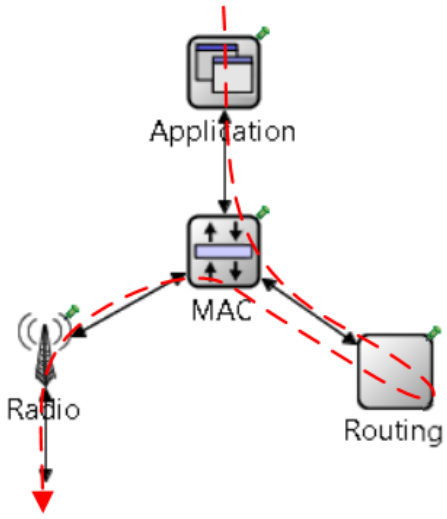

Рисунок 16 – Направления движения пакета по узлу, от уровня приложения

На рисунке 17, пакет поступил на радио уровень, аналогично происходит его анализ и пересылка, как и в случае его отправки с уровня приложения.

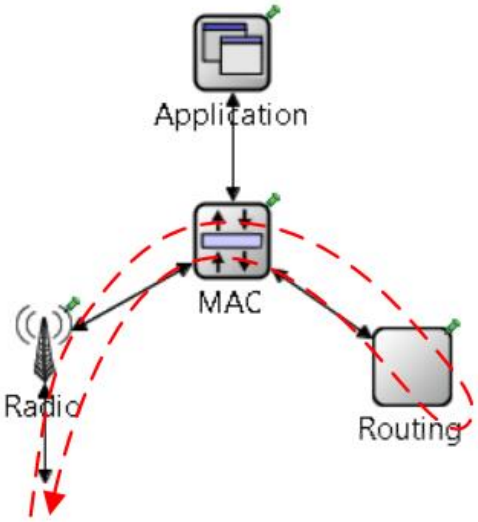

Рисунок 17 – Направления движения пакета по узлу, от радио уровня

Как видно из графиков (рисунок 18) время нахождения пакетов на узле составляет 68,5 и 75 мс. Первое время соответствует, времени движения пакета от уровня приложения, второе время соответствует движению пакета с радио уровня.

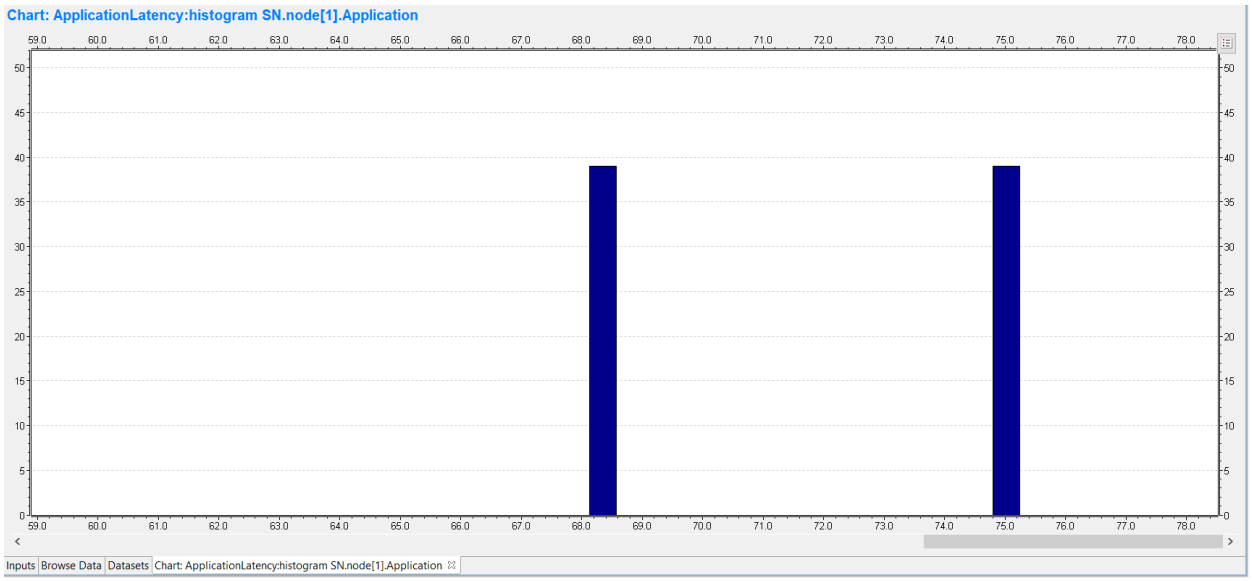

Рисунок 18 – График измерения характеристики Latency

Delay, основной параметр для маршрутизации, показывающий необходимое время для доставки пакета на узел назначения, при пересылке его через промежуточные узлы.

На рисунке 19, показанно движения пакета от узла отправителя, до узла назначения. В данном случае узел назначения является PAN координатор.

Delay 1, соответствует необходимому времени для того, чтобы пакет покинул Node 1 и прибыл в PAN координатор (Node).

Delay 2, соответствует необходимому времени для того, чтобы пакет покинул Node 2, был маршрутизирован на Node 1, и прибыл в PAN координатор.

Delay 3, пакет покинул Node 3, маршрутизирован на Node 2, далее маршрутизирован на Node 1, и прибыл в PAN координатор.

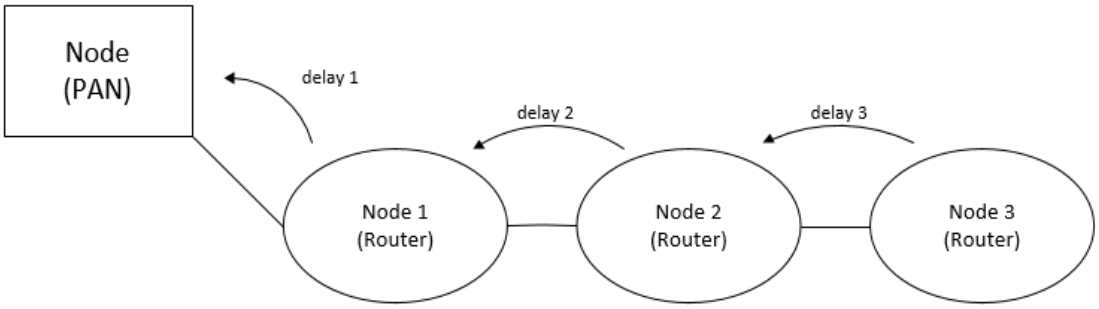

Рисунок 19 – Движения пакета от узла отправителя, до узла назначения

На рисунке 20, показано как увеличивается время доставки пакета, от узла отправителя до узла назначения, с возрастанием необходимости маршрутизации пакета, между узлами.

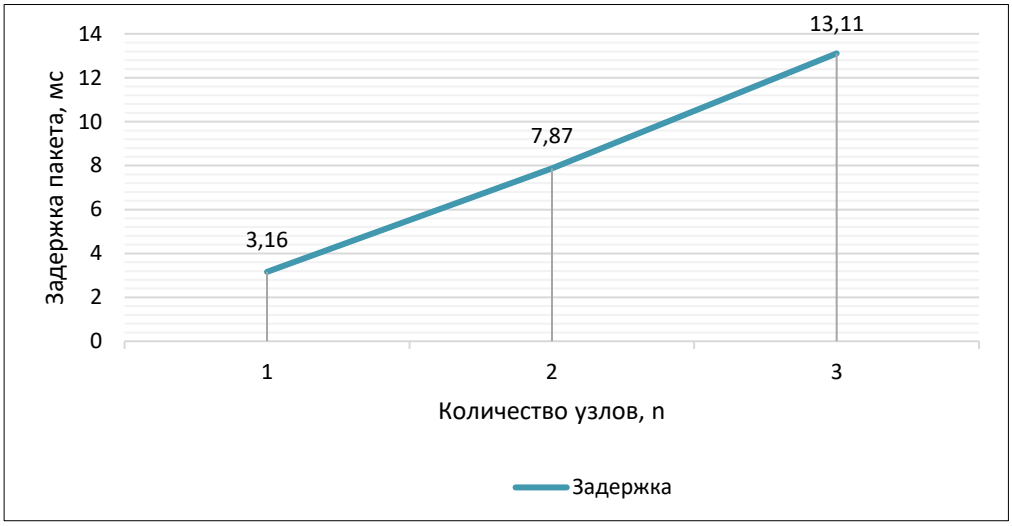

Рисунок 20 – Задержка пакетов при доставки

ЗАКЛЮЧЕНИЕ

В результате выполнения выпускной квалификационной работы была изучена предметная область и существующие аналоги. Был произведен анализ документации беспроводный сенсорных сетей, построенных на базе стандарта IEEE 802.15.4/ZigBee. Целью анализа было выяснить основные механизмы передачи данных, построенные топологии сети, с использованием механизмов маршрутизации.

При анализе стандарта была реализована диаграмма автомата. Диаграмма соответствует механизму передачи данных, с уровня приложения на сетевой уровень, анализу полученного пакета, сопоставлению полученных данных с данными находящимися на узле, изменение заголовка пакета, инициализации поиска маршрута и отправка пакета на уровень MAC.

В среде моделирования OMNeT++ была реализована модель, в соответствии с разработанной диаграммой автомата. На разработанной модели был проведен ряд экспериментов, для оценки времени задержки пакетов при передаче.

Проведенные эксперименты с использованием разработанных моделей, подтвердили гарантированную доставку пакетов, но показали при этом увеличение времени доставки.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Nitaigour, P. M. Sensor networks and configuration fundamentals, standards, platforms, and applications / P. M. Nitaigour. – Springer, 2007. – 510 p.
2. ZigBee PRO with Green Power [Электронный ресурс] : Zigbee alliance. – Режим доступа: http://www.zigbee.org/zigbee-for-developers/network-specifications/zigbeepro.
3. Сети ZigBee. Зачем и почему? [Электронный ресурс] : Ресурс для IT-специалистов. – Режим доступа: http://habrahabr.ru/post/155037.
4. Денисенко, В. Испытания электронной аппаратуры: быстро и эффективно / В. Денисенко, П. Ерещенко, Р. Кильметов, Е. Метелкин, А. Халявко, К. Шальнев // "Компоненты и технологии". – 2004. – № 4. – С. 216-220 ; № 5. – С. 158-166.
5. Пушкарев, О. Использование конечных спящих узлов в сети ZigBee / О. Пушкарев // Электронные компоненты. – 2011. – № 5. – С. 91–95.
6. GET IEEE 802 [Электронный ресурс] : IEEE-SA. – Режим доступа: http://standards.ieee.org/getieee802/download/802.15.4-2011.pdf.
7. Методы доступа в канал [Электронный ресурс]. – Режим доступа: http://pdst.narod.ru/_20_el_uch/ost_wpd_01/part09.html.
8. OMNeT++ [Электронный ресурс] : OMNeT++ Discrete Event Simulator. – Режим доступа: https://omnetpp.org.
9. Castalia [Электронный ресурс] : Castalia Wireless Sensor Network simulator. – Режим доступа: https://castalia.forge.nicta.com.au/index.php/en/.
10. Гудов, А. М. Имитационное моделирование процессов передачи трафика в вычислительных сетях / А. М. Гудов, М. В. Семехина // Управление большими системами. Выпуск 31. – 2010. – № 31. – С. 130-161.
11. Сайт исследовательской группы, занимающейся изучением беспроводных сенсорных сетей [Электронный ресурс] – Режим доступа: http://www.sensor-networks.org.

12. Балонин, Н. А. Беспроводные персональные сети на основе ZigBee. Учебное пособие / Н. А. Балонин, М. Б. Сергеев – Санкт-Петербург : ГУАП, 2015. – 58 с.

13. Технологии беспроводной передачи данных ZigBee, BlueTooth, Wi-Fi [Электронный ресурс] – Режим доступа: http://wireless-e.ru.

14. Ramachandran, I. Analysis of the contention access period of IEEE 802.15.4 MAC / I. Ramachandran, A.K. Das, S Roy // ACM Transactions on Sensor Networks,Vol.3. – 2007. № 3. – С. 24–29.

15. Рябинин, И. А. Надежность и безопасность структурно-сложных систем / И. А. Рябинин. – Санкт-Петербург : Политехника, 2000. – 248с.