

Аннотация

Темой данной выпускной квалификационной работы является «Разработка ПО и алгоритмов для настройки и маршрутизации трафика Mesh-сети».

Объектом данной работы является Mesh-сеть, ее характерные признаки, особенности топологии и способы маршрутизации трафика сети.

Предметом работы являются алгоритмы, лежащие в основе методов маршрутизации Mesh-сети.

Для достижения поставленной цели были исследованы различные способы достижения оптимальной маршрутизации, рассмотрены и сгруппированы, исходя из приоритетных задач маршрутизации, алгоритмы, сформулирована проблема многокритериальной маршрутизации в беспроводной Mesh-сети.

Данная работа представляет из себя: введение, четыре основных раздела, заключение, а также список используемой литературы и список источников.

Введение затрагивает актуальность данной темы, а также производит постановку задачи для реализации оптимальной многокритериальной маршрутизации.

Первая глава содержит описание уже существующих методов построения оптимального маршрута трафика в сети.

Вторая глава содержит описание и сравнение алгоритмов, на которых строятся актуальные протоколы маршрутизации.

Третья глава содержит в себе анализ среды функционирования Mesh-сети, влияние внешних факторов на качество связи.

В четвертой главе описывается практическая реализация локальной Mesh-сети в конкретных условиях.

Заключение представляет из себя выводы, сделанные в результате проделанной работы.

В работе было использовано 16 формул, 10 рисунков, 3 листинга и 4 таблицы. Общий объем квалификационной работы составил 56 страницы.

ABSTRACT

This graduation work deals with developing settings and routing algorithms of a Mesh network.

The aim of the research is to model a local Mesh network and to set its traffic routing.

The object of the investigation is the Mesh network, its topology features and the ways to route the network traffic.

The subject of the graduation work are the algorithms underlying the Mesh network routing methods.

The graduation work describes in details different ways to achieve optimal routing and the corresponding algorithms as well as states the problem of multi-criteria routing in a wireless Mesh network.

We first discuss the relevance of this topic, some wireless technologies and their significance.

The first part of the graduation work explains the general concept of a Mesh network, what principle it is built on, its advantages and disadvantages. It also discusses the existing routing methods and reveals the process of forming information about the network and its routes. The first part of the research dwells on forming the information on a single network node that has locally connected devices as well.

Next, we elucidate what algorithms exist that the current routing protocols are based on and compare them. We explain the advantages and drawbacks of each class of protocols, as well as cover the necessary algorithm criteria.

Then, we analyze the Mesh network environment and the influence of the external factors on the quality of communication. We also consider the main problems that complicate the routing process in wireless mesh networks.

In conclusion, we would like to underline that the effectiveness of the network depends on many parameters. It is also pointed out that the problem of multi-criteria optimization problem is incorrect because the particular quality

criteria conflict with each other: the improvement of one particular quality criterion deteriorates one or more other particular quality criteria.

Содержание

Введение	7
1 Маршрутизация Mesh-сетей – принципы и протоколы.....	10
1.1 Общее определение Mesh-сетей.....	10
1.2 Существующие методы маршрутизации Mesh-сетей.....	11
1.2.1 «Проактивные протоколы» маршрутизации Mesh-сетей.....	14
1.2.2 Реактивные протоколы маршрутизации Mesh-сетей	16
1.2.3 Гибридные протоколы маршрутизации Mesh-сетей	18
2 Алгоритмы маршрутизации Mesh-сетей.....	22
2.1 Общие критерии выбора алгоритма маршрутизации для всех видов сетей.....	22
2.2 Особенности расчета алгоритма маршрутизации для Mesh-сетей	24
2.3 Алгоритмы нахождения кратчайшего пути	28
2.4 Альтернативные алгоритмы маршрутизации	38
3 Выбор оптимального алгоритма маршрутизации для Mesh-сетей.....	40
3.1 Факторы, влияющие на эффективность маршрутизации.....	40
3.2 Проблема многокритериальной маршрутизации	42
4 Моделирование локальной mesh-сети.....	45
4.1 Определение задачи исследования.....	45
4.2 Реализация задачи исследования.....	46
4.3 Результаты проведенной работы	49
Заключение	54
Список используемых источников	55

Введение

С каждым годом число потребителей беспроводного Интернета растет, происходит цифровизация различных аспектов жизни (концепция Internet of Things – Интернет вещей [14]), что, в свою очередь, увеличивает требования к доступности и качеству связи. По состоянию на январь 2020 года во всем мире насчитывалось 4,34 миллиарда активных пользователей Интернета, через год, в январе 2021, их уже стало 4,66 миллиардов человек, причем большая их часть – свыше 90 процентов, выходили в Интернет, используя мобильные устройства. Согласно статистическим исследованиям Digital 2021, современный человек тратит почти четыре часа каждый день на использование Интернета на своем мобильном телефоне [25].

Актуальная на сегодняшний день задача – создать концепцию и реализовать новое поколение беспроводных технологий, предоставляющее пользователям мобильный и повсеместный доступ к Интернету вне зависимости от географической удаленности от центральных магистралей. На сегодняшний момент используется целый ряд беспроводных технологий, наиболее известные пользователям по следующим названиям:

- Bluetooth, версия 5.2 [27] – беспроводные сети личного пользования (WPAN – Wireless Personal Area Networks), регулируются стандартом IEEE 802.15.1 [26].
- Wi-Fi [17, с. 12] – беспроводные сети с территориальным ограничением (WLAN – Wireless Local Area Networks), работа которых регламентируется стандартом IEEE 802.11 [2];
- WiMAX [15, с. 113] – беспроводные сети с огромным множеством узлов, способные покрывать территорию среднего или крупного населенных пунктов (WMAN – Wireless Metropolitan Area Networks), стандарт IEEE 802.16 [15, с. 114].
- GPRS [11, с. 8] – беспроводные глобальные сети, охватывающие

континенты (WWAN - Wireless Wide Area Network) [16].

Беспроводные сетевые технологии максимально подходят для удовлетворения постоянно растущих потребностей Интернет-потребителей, поскольку обладают следующими качествами:

- гибкостью топологии (возможность изменения сети при подключении или передвижении пользователей с сохранением связи),
- высокой скоростью передачи информации,
- быстротой проектирования и развертывания,
- надежностью защиты от несанкционированного доступа [17, с. 104] и др.

Среди беспроводных сетей наиболее перспективными выглядят так называемые Mesh-сети – компьютерные сети, построенные по принципу «ячеистой» топологии. Размеры таких сетей эволюционировали от домашних, состоящих из десятка узлов, до региональных, насчитывающих десятки тысяч пользователей. Mesh-сети соединяют в себе различные сетевые и радио технологии, и потому в лучшей степени отвечают современным требованиям технологий передачи сообщений (скоростная передача данных, неприкосновенность передаваемой информации). Перспектива реализации с помощью Mesh-технологии локальных (LAN) и городских (MAN) сетей, простота их присоединения к глобальным сетям (WAN), безусловно увеличивает число сторонников среди как корпораций, так и персональных пользователей.

Благодаря наличию доступных и недорогих радиointерфейсов можно снабдить каждый беспроводной узел несколькими радиомодулями, реализуя таким способом одновременную передачу по доступным ортогональным каналам и уменьшая помехи [16]. Возможность самомаршрутизации осуществляется посредством сетевых протоколов, которые оптимально распределяют и/или используют имеющиеся сетевые ресурсы: сетевой трафик (объем информации), пропускные способности каналов связи, последовательность соединений, выбор частоты каналов.

Однако фундаментальной задачей при проектировании беспроводных ячеистых сетей с множеством радиосвязей остается решение проблемы совместного назначения каналов и маршрутизации. Проблема заключается в том, как назначить доступные каналы радиоинтерфейсам и как рассчитать скорость потока, который необходимо направить по каждому каналу для достижения поставленной цели [18]. Для начала определим влияние типичных параметров беспроводной сети, таких как передача данных, скорость, мощность передачи и частота передачи на общую производительность беспроводной ячеистой сети. Таким образом, в этой работе мы сосредоточимся на обзоре современных, активно используемых алгоритмов оптимальной маршрутизации трафика Mesh-сетей.

Целью работы является моделирование локальной Mesh-сети и настройка маршрутизации трафика в этой сети. Для достижения этой цели были решены следующие задачи:

- проанализированы наиболее используемые методы маршрутизации Mesh-сетей,
- исследованы различные группы алгоритмов, на основе которых были разработаны известные методы маршрутизации,
- выбраны программные средства для моделирования и исследования беспроводных сетей и систем связи.

1 Маршрутизация Mesh-сетей – принципы и протоколы

1.1 Общее определение Mesh-сетей

Mesh-сеть или «ячеистая сеть» — это топология компьютерной сети, устроенная таким образом, что каждое устройство (узел) соединяется со всеми или почти всеми узлами сети одноранговыми связями, работая не только на прием или передачу данных, но при этом выполняя функцию маршрутизатора. Узлы в такой сети могут передвигаться в пространстве, уже имеющиеся могут отключаться или подключаться новые, иначе говоря, количество узлов и соединений между ними непостоянно, физическая среда, через которую проходит сигнал сети, так же неоднородна и нестабильна, соответственно маршруты между устройствами постоянно меняются.

Несмотря на свою специфику, «ячеистые» сети должны соответствовать стандартизированным требованиям, предъявляемым ко всем сетевым образованиям – предоставление высокоскоростного соединения [17, с. 217], достаточного для передачи данных, чувствительных к потере скорости (видео и аудио контент), надежность соединения, защита передаваемых данных от потерь и ошибок, простота и свобода доступа к сети. Существующие на сегодняшний день способы организации беспроводных сетей в большинстве своем рассчитаны на стационарные беспроводные устройства, к примеру, это batman-adv, разработанный на базе операционной системы UN*X, и стандарт 802.11s [3, с. 200]. Принципы, эффективные для построения неподвижных Mesh-сетей, как то: выбор маршрута на базе конечного числа метрик (количество шагов или каналов), недостаточны для мобильных беспроводных сетей, число и положение узлов которых меняется с течением времени.

Подытоживая, Mesh-сеть представляет собой саморганизующуюся одноранговую форму построения сети, которая обладает дополнительными возможностями: масштабируемостью, т.е. способностью разрастаться без потери качества связи за счет присоединения новых узлов; каждое новое устройство служит ретранслятором для других, вследствие чего увеличивается

зона покрытия сети; быстротой развертывания сети и высокой отказоустойчивостью, поскольку в «ячеистых» сетях большинство соединений избыточны. Mesh-сети больших размеров, свыше сотни узлов, строятся как совокупность кластеров, в каждом из которых может находиться до 16 устройств связи. Одно устройство становится является узловой точкой и подключается к глобальной сети любым доступным способом, в свою очередь, кластеры соединяются между собой в единую сеть через подобные точки доступа. При этом узловые точки продолжают оставаться рядовыми узлами. Обобщенная схема строения топологии mesh-сети изображена на рисунке 1.

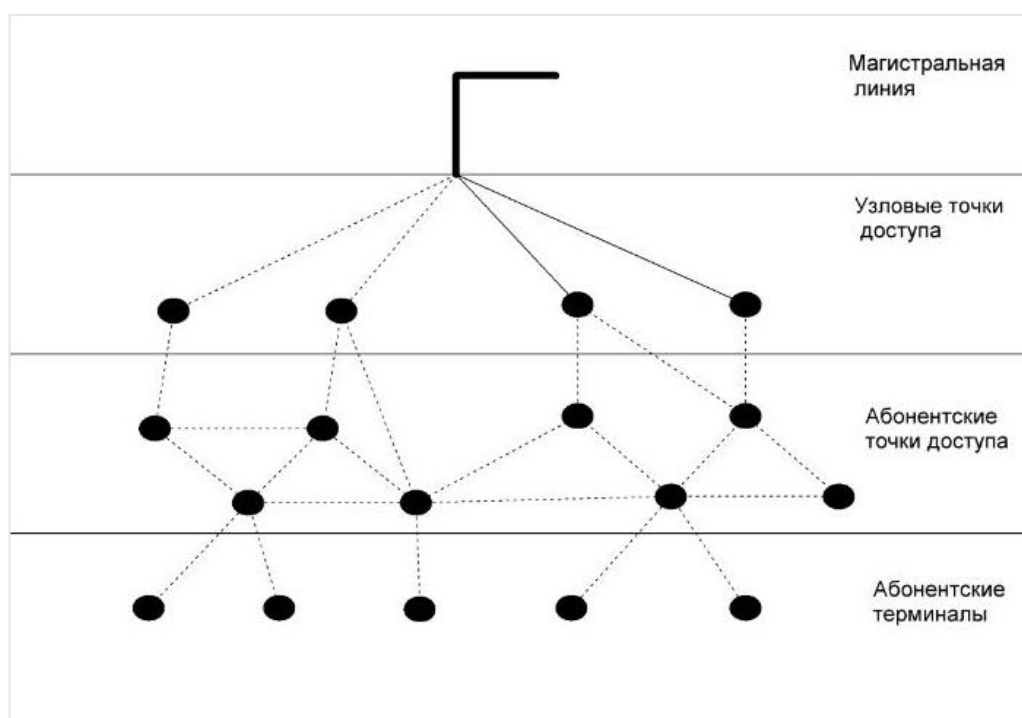


Рисунок 1 – Топология Mesh-сети

1.2 Существующие методы маршрутизации Mesh-сетей

Любое устройство при подключении к глобальной сети «передает в Интернет свой уникальный адрес физического присутствия, так называемый MAC-адрес, с некоторым максимально допустимым коэффициентом доступности K_{max} при условии отсутствия локально подключенных к нему

точек. Если же локальные подключения наличествуют, то данный узел помимо своего идентификационного номера передаёт в сеть MAC-адреса всех соседних с ним устройств связи с тем же коэффициентом K_{max} » [18].

Соседнее с исследуемым узлом устройство, получая информацию о MAC-адресах, «узнает» и дополнительно запоминает время получения сообщения, сведения о надежности и энергозатратности соединения, отсутствии посреднических узлов и о пропускной способности канала связи.

Таким образом, соседний узел имеет необходимые данные для расчета коэффициента ограничений соединения с исходным узлом как источником передачи, так и промежуточной точкой для дальнейшей маршрутизации. В работе [18] утверждается, что для этого «достаточно вывести функцию расчета коэффициента достижимости с учетом предыдущего значения. Тогда формула нахождения коэффициента, получаемого от соседней точки связи, будет выглядеть следующим образом:

$$K' = F(F_f, F_n(1), F_e, F_p, K), \quad (1)$$

где F_f – надежность узла, F_n – число промежуточных узлов, F_e – энергоресурс соединения, F_p – пропускная способность узла, F_t – возраст данных о маршруте.

Данные, полученные позднее, имеют естественное преимущество перед более ранними, из чего рассчитаем формулу нахождения коэффициента от времени

$$K'' = F'(F_t(n), K), \quad (2)$$

где n – число микросекунд, прошедших с момента последнего обновления по времени» [18].

На таком принципе основано функционирование «проактивной» части прокладывания маршрута для передачи сообщений, рассмотрим теперь цикл работы реактивной части этого процесса.

«Каждый узел строит отдельную таблицу маршрутизации, данными для которой являются доступные MAC-адреса и коэффициенты достижимости,

собственный MAC-адрес узла и MAC-адреса локально подключенных устройств имеют коэффициент K_{max} , а другие узлы имеют текущий расчетный коэффициент, находящийся в интервале $K_{min} < K < K_{max}$.

При получении сообщения узел производит перерасчет коэффициентов доступности для узлов, имеющих коэффициент меньше K_{max} локальной таблицы маршрутизации по формуле

$$K'' = F'(F_t(n), K), \quad (3)$$

где n – число микросекунд с момента предыдущего пересчета. Исключаются маршруты, достигшие значения коэффициента K_{min} . Коэффициенты для маршрутов из анонса пересчитываются по формуле

$$K' = F(F_f, F_n(1), F_e, F_p, K), \quad (4)$$

т.е. применяются штрафы к соединению.

После расчета выполняется объединение данных из имеющихся таблиц по следующему алгоритму:

- если актуальный MAC-адрес отсутствует в таблице маршрутизации, он копируется из приветственного сообщения вместе со свежим коэффициентом доступности, тогда сообщивший этот MAC-адрес узел становится промежуточным для дальнейшей маршрутизации;
- если для сообщаемого MAC-адреса имеется маршрут, то путь из сообщения добавляется в нужную таблицу;
- временные метки в таблице маршрутизации заменяются на актуальные.

В конечном итоге у каждого сетевого узла наличествует постоянно обновляемая таблица маршрутизации, в которой для каждого MAC-адреса в сети имеется не менее одной записи с MAC-адресом соседнего узла, с которого начинается маршрут трафика к исходному MAC-адресу» [18].

Постоянно изменяющаяся топология mesh-сети создает проблему с построением оптимального маршрута передачи сообщений между адресантом

и адресатом. Решением этой задачи занимаются протоколы маршрутизации, которые можно разделить на два класса:

- проактивные (табличные), при которых таблица маршрутизации строится каждым узлом сети, информация о данных таблицы постоянно обновляется и распространяется среди соседних точек. Примеры: OLSR, Destination Sequenced Distance Vector;
- реактивные (работающие по запросу), при которых таблицы маршрутизации составляются по мере необходимости, а для определения пути отправки сообщения отправляется широковещательный запрос на все точки сети. Примеры: Dynamic Source Routing, AODV [5].

Чаще всего используют гибридные протоколы, которые способны работать как в режиме проактивных, так и в режиме реактивных протоколов, в зависимости от поставленных задач маршрутизации. Одним из часто применяемых гибридных протоколов можно назвать протокол HWMP [5].

1.2.1 «Проактивные протоколы» маршрутизации Mesh-сетей

Несмотря на принадлежность одному классу «проактивные протоколы» имеют существенные различия.

OLSR (в переводе с англ. «Оптимизированная маршрутизация состояния канала») [30] использует постоянно обновляемый сбор и переработку информации о состоянии сети, с последующим её распространением среди всех узлов сети. Вследствие чего каждый узел строит модель актуального состояния сети в виде графа, которым изображается карта сети для конкретного узла. При помощи такой схемы каждый узел может построить свою таблицу маршрутизации, вычислив длину кратчайшего пути ко всем абонентским узлам, и выбрать оптимальный маршрут до нужной точки (рисунок 2). Преимущество этого протокола в том, что он эффективно работает при незапланированных и внештатных ситуациях, как, например:

- внезапный выход из строя и последующий ремонт устройств сети,

- блокировка произвольного числа элементов сети вследствие перегрузок,
- переподключения абонентов.

В массивных сетях с большим количеством подключений на единицу площади протокол OLSR показывает себя наилучшим образом, поскольку всегда использует ресурс пропускной способности для служебных сообщений.



Рисунок 2 – Работа алгоритма OLSR

Проактивный протокол DSDV использует другой принцип, при котором каждый узел имеет собственную таблицу маршрутизации, где представлены данные о всех потенциальных маршрутах, количестве шагов до конечного узла и номер варианта. С определенной периодичностью и в случае табличных изменений узлы обмениваются данными таблиц с ближайшими точками сети. В отличие от вышеописанного протокола OLSR, который всегда использует служебный канал, протокол DSDV предписывает обмен полными данными между узлами лишь при серьезных топологических переменах, в остальных случаях ограничивая пересылку только мелкими обновлениями.

При увеличении числа абонентов или при изменении карты сети зачастую переполняется параметр номера варианта маршрутизации, поскольку для каждой версии маршрута создается новый порядковый номер. Поэтому DSDV плохо подходит для динамичных плотных сетей, но лучше проявляет себя при построении мелких, с небольшим количеством узлов, одноранговых сетей с редко меняющейся топологией.

Протокол DSDV в настоящее время используется реже, чем OLSR, хотя его основной принцип применяется в различного вида гибридных протоколах. К достоинствам проактивных протоколов можно отнести:

- использование комплексной оценки по нескольким параметрам при выборе оптимального маршрута (числу узлов, пропускной способности каналов, задержке доставки пакетов);
- каждый узел хранит информацию о топологии сети.

К недостаткам проактивных протоколов можно отнести:

- быструю загруженность сети служебным трафиком,
- необходимость хранения и своевременного обновления актуальной конфигурации сети.

Ниже представлена сравнительная таблица характеристик упомянутых проактивных протоколов.

Таблица 1 – Сравнительный анализ основных характеристик проактивных протоколов

Имя	Маршрутная структура	Число таблиц	Частота обновления	Особенности
DSDV	Плоская	2	Периодически и по запросу	Защита от петель
OLSR	Плоская	3	Периодически	Защита от петель на основании предшествующей информации
WRP	Плоская	4	Периодически	Использует MPR

1.2.2 Реактивные протоколы маршрутизации Mesh-сетей

Класс протоколов реактивной маршрутизации функционирует по иному принципу. Протокол DSR [10], являясь одним из ранних представителей этого класса, собирает данные о маршруте, в отличие от проактивных протоколов, в самом пакете запроса. После чего пакет рассылается всем узлам сети, при этом в заголовке пакета запроса делается отметка о каждом пройденном узле. Таким образом создается карта пути между исходным и конечным узлами. При

возникновении маршрутной петли в момент повторного получения пакет уничтожается узлом. Логично предположить, что при длительных маршрутах в сети с большим числом абонентов размер пакета запроса может вырасти до неудобных размеров, что повлечет за собой перегрузку канала связи.

Протокол DSR регулярно улучшался и дорабатывался, чтобы его специфические особенности не приводили к нарушениям в работе сети, в какой-то момент он стал основой для создания и развития новых реактивных протоколов, как, например, DSRFLOW. В новом протоколе в заголовок пакета запроса вносится не весь адрес транзитного узла, а особая метка, за счет этого размер пакета остается в рамках приемлемого.

Протокол AODV (в переводе с англ. «Дистанционно-векторная маршрутизация») также был разработан на базе DSR. В отличие от DSR, таблицы маршрутизации создаются традиционным способом на каждом узле, это экономит время передачи сообщений между точками сети. Соседние друг с другом точки обмениваются данными из своих таблиц, получая и передавая карты кратчайшего пути в любую часть сети, обновляя собственные таблицы маршрутизации. Через определенное количество таких обменов происходит выбор самого короткого пути для передачи сообщения.

Как можно заметить, протокол AODV использует традиционный способ построения таблицы маршрута, но делает это только по необходимости, когда маршруты долго не используются и/или истекло время их «жизни». При сбоях в работе сети протокол AODV способен обнаружить источник неполадки и проложить другой маршрут в обход разрыва, не делая широковещательного сообщения об этом на всю сеть.

Таблица 2 – Сравнительный анализ основных характеристик реактивных протоколов

Имя	Маршрутная структура	Множественные маршруты	Метод расчета маршрут	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

Механизм работы реактивных протоколов обуславливает их сильные стороны:

- низкая загруженность сети в период простоя,
- увеличенная пропускная способность сети,
- отсутствие необходимости хранения в каждом узле полной информации о сети.

Слабые стороны реактивных протоколов, способные затормозить работу сети:

- требуется дополнительное время на поиск и построение маршрута,
- частые задержки при активной подвижности узлов и/или изменении их количества,
- использование одной метрики для построения оптимального пути.

1.2.3 Гибридные протоколы маршрутизации Mesh-сетей

Гибридные протоколы способны действовать, чередуя или совмещая алгоритмы и проактивных, и реактивных протоколов, в зависимости от приоритетных целей. Чаще всего это реализуется через разделение всей сети на множество подсетей, маршрутизация внутри каждой из которых строится по принципам проактивных протоколов, подсети же в свою очередь сообщаются друг с другом согласно реактивным протоколам [12]. За счет

такого членения происходит существенное сокращение размеров маршрутных таблиц каждого узла, а значит и уменьшаются объёмы служебных сообщений, рассылаемых в общей сети, так как основная часть информации распространяется внутри подсетей.

Гибридный протокол HWMP [5] (в переводе с англ. «Гибридный протокол беспроводной сети») был создан на базе популярного реактивного протокола AODV, естественно, с некоторой модернизацией. В протоколе HWMP прокладывание маршрутов и создание таблиц реализуются на уровне MAC-адресов узловых точек, где имеется информация и о всех соседних с ними узлах. В реактивном режиме каждый узел сети рассылает т.н. PREQ – это пакет запроса пути. В каждом таком пакете есть специальный раздел, в котором находится информация о метрике, на основе которой выбирается маршрут. Во время передвижения от точки к точке эта информация о метрике обновляется и дополняется, и, в конце концов, создается полная метрика маршрута адресат-адресант. За счет того, что подобные действия (рис.3) производит каждое сетевое устройство, заполняя данными собственную таблицу, создается полная картина состояния сети.

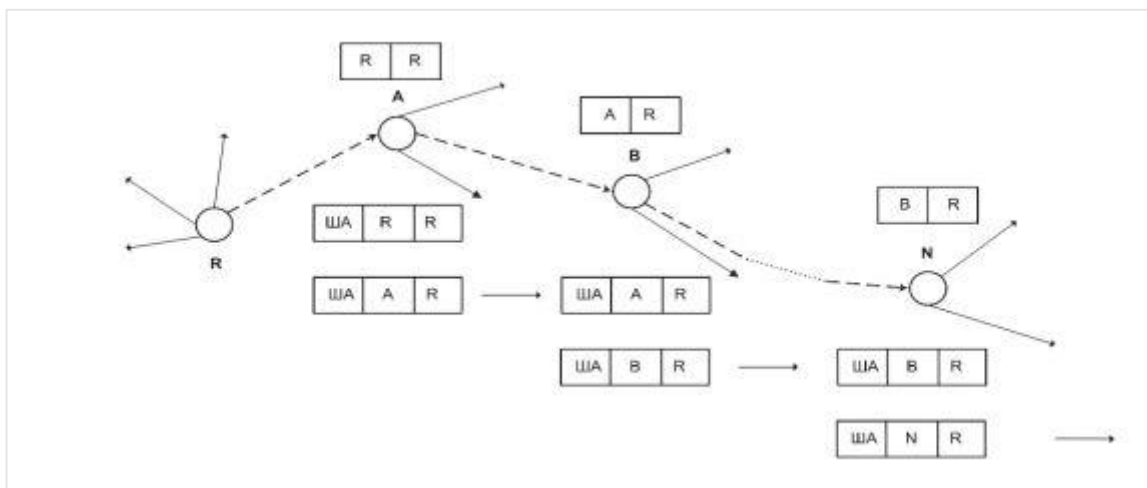


Рисунок 3 – Реактивный режим работы протокола HWMP: ША – широковещательный адрес рассылки, R – начальный адрес, A, B, N – адреса промежуточных узлов, ретранслирующих широковещательный пакет

Каждая точка сети, получившая этот пакет (на рис. 3 это узел A), без проверки содержимого, убеждается в первичном и единственном получении PREQ-пакета с адресом адресанта R. В случае с повторным получением пакет отбрасывается узлом. Если отправление с таким адресом пришло впервые, содержимое сообщения заносится в новую строку таблицы MAC-адресов (запись RR). В список адресов транзитных точек вписывается собственный идентификатор узла A, после чего обновленный пакет данных посредством широковещания рассылается всем абонентам. И в конечном итоге в каждом узле формируется таблица всех адресов, в которой число строк соответствует числу узлов всей сети [5].

В свою очередь начинает действовать «проактивный» режим, во время которого тоже происходит рассылка пакетов запроса сети, но от единственного узла R, назначаемого корневым. Главное отличие от аналогичных действий в реактивном режиме заключается в пересылке пакета подтверждения выполненной команды PREP каждым узлом напрямую корневому узлу R. На рисунке 4 изображена схема выполнения этого циклического действия.

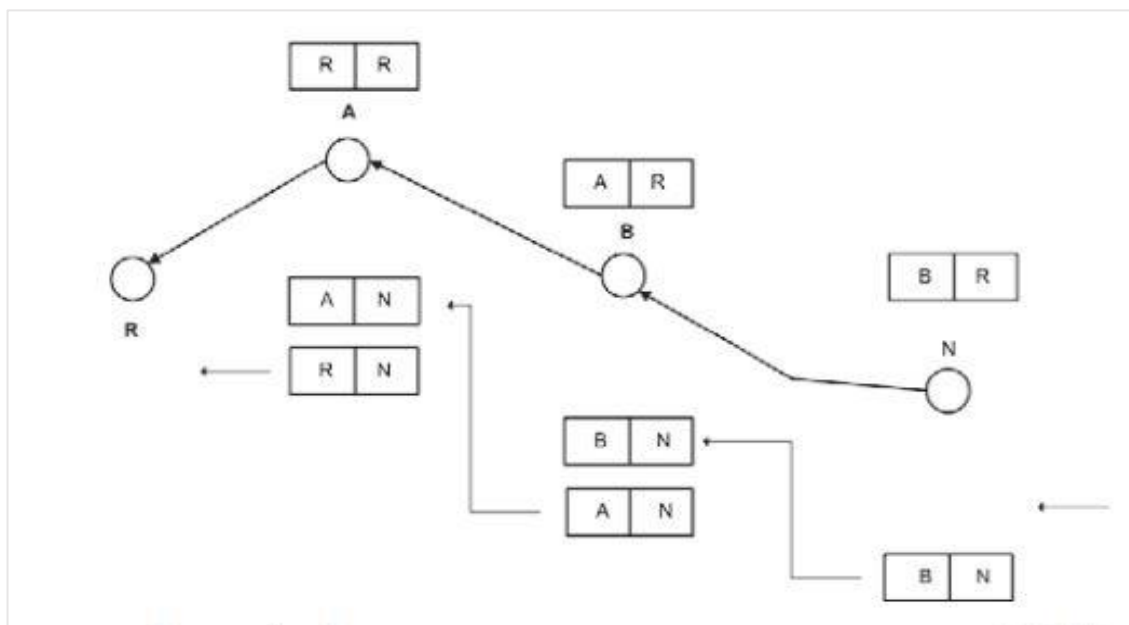


Рисунок 4 – Механизм отправления пакета-подтверждения от рядового узла сети
корневому

Для того, чтобы передать сообщение в сеть, каждый узел вначале находит в своей маршрутной таблице адрес получателя и отправляет пакет на ближайший доступный узел, транзитный узел, в свою очередь, уточняет в собственной таблице конечный адрес и отправляет дальше по сети своему соседу [5]. Так повторяется до тех пор, пока отправление не придет в нужное место.

Протокол маршрутизации HWMP является базовым протоколом маршрутизации для «ячеистых» беспроводных сетей и обязателен для всех устройств по условию стандарта IEEE 802.11s как протокол по умолчанию.

Стоит упомянуть протокол LANMAR [3, с. 206] – еще один гибридный протокол маршрутизации, особенность которого заключается в объединении узлов сети в группы в зависимости от их передвижения. В каждой из таких групп произвольно выбранный узел становится так называемой «точкой отсчета» или реперной точкой, относительно которой выстраивается маршрут передачи сообщений между узлами различных групп. Таблица маршрутизации LANMAR состоит только из узлов подсети и главных узлов. В результате этого, поскольку уменьшились размеры таблиц маршрутизации, появляется возможность уменьшить объем служебной информации об обновленных маршрутах, а также улучшить предпосылки к масштабируемости сети [3, с. 208].

2 Алгоритмы маршрутизации Mesh-сетей

2.1 Общие критерии выбора алгоритма маршрутизации для всех видов сетей

Несмотря на вышеперечисленные для каждого класса протоколов недостатки и достоинства, в каждом из них используется таблица маршрутизации, на основе которой выбирается оптимальный путь. Таблица маршрутизации заполняется данными согласно выбранным алгоритмам и рассматривается в виде графа. Оптимум маршрута зависит от приоритетных задач, поставленных при разработке алгоритма. Чаще встречаются следующие критерии выбора алгоритма:

- оптимальность,
- рациональность и конструктивность,
- жизнестойкость и стабильность,
- согласованность,
- адаптивность.

Оптимальность – основное свойство алгоритма маршрутизации подстраиваться под сетевые условия и выбирать самый короткий маршрут или наиболее предпочтительный. Зачастую выбор маршрута зависит от метрик числа переходов и/или от суммы других показателей, позволяющих, к примеру, избегать коллизий. Возможны ситуации, когда короткий путь будет отброшен алгоритмом маршрутизации из-за временных задержек, которые с числом шагов только бы увеличивались.

Рациональность и конструктивность подразумевают, что алгоритмы маршрутизации разрабатываются в простой и лаконичной форме без чрезмерного нагромождения деталей, при этом оставляя возможность внесения изменений или дополнений. Механизм маршрутизации должен быть эффективным в выполнении заявленных задач даже при работе с устройствами с ограниченной мощностью.

От алгоритма маршрутизации ожидается высокая работоспособность даже в условиях некорректного функционирования сети, как то выход из строя элементов сети, канальные перегрузки, ошибки в передаче данных и т.п.

Сходимость или согласованность – это одинаковое понимание всеми узлами сети текущей её топологии. По-другому определяется как непротиворечивость информации данных таблиц маршрутизации всех точек связи. Любое изменение в строении сети должно получать широковещательную огласку для последующего обновления данных маршрутных таблиц, в результате чего все сетевые узлы «приходят к соглашению» маршрутам передачи сообщений. В противном случае возникают конфликты маршрутных таблиц, что приводит к появлению «закольцованности маршрутизации», коллизиям, частичной или полной потере пакетов данных, т.е. так или иначе сеть выходит из строя.

Адаптивность или приспособляемость – способность алгоритма подстраиваться под различные ситуации, возникающие в сети. Изменение топологии или обрыв связи с узловой точкой должны быть разрешимой для алгоритма проблемой, в любом случае механизм должен не только продолжать функционировать, но и отобрать из имеющихся вариантов самый лучший [6].

Эти критерии подбора алгоритмов общие для всех видов сетей, однако в случае самоорганизующихся ячеистых сетей есть свои особенности. В частности, спонтанность и непредсказуемость изменения топологии сети в сочетании с меняющимися энергоресурсами каналов связи между узлами, а также стандартная схема выделения временного интервала, в рамках которого устройством связи будет совершен процесс передачи одного пакета данных. Потребность в высокоскоростной связи и короткие временные промежутки устойчивости сетки вещания требуют высокого уровня приспособляемости от системы маршрутизации. Отсутствие центрального сетевого администрирования подводит к идее применения самоуправляемых и/или распределенных (мультиагентных) алгоритмов.

2.2 Особенности расчета алгоритма маршрутизации для Mesh-сетей

Привлекательность Mesh-сетей помимо всего прочего заключается в их динамической топологии и доступной масштабируемости. Существующие в настоящее время алгоритмы маршрутизации до определенного момента, хотя и с оговорками, справлялись со своей основной функцией [19]. Однако с развитием технологий заметно и планомерно увеличивается объем новых подключений к сетям, а нагрузки на каналы связи давно вышли за пределы расчетных. Принятый за стандарт главный критерий поиска «лучшего маршрута», основанного на минимальном числе «узловых» переходов, становится менее актуальным в быстрорастущих многопользовательских сетях, протоколы маршрутизации, построенные на таком подходе, при росте количества узловых точек в сети резко теряют в эффективности и явно обнаруживают свои недостатки [6]: размеры служебных сообщений, передаваемых при «проактивных» методах, резко увеличиваются, что удлиняет время рассылки и способствует возникновению ошибок при передаче, «реактивные» протоколы со своей стороны увеличивают объем поисковой рабочей информации, рассылаемой по сети, что чрезмерно расходует каналный ресурс и усложняет вычислительные алгоритмы [6]. Самый короткий путь в таких условиях может оказаться не самым быстрым или надежным.

Неэффективность, точнее ограниченная эффективность при определенных условиях используемых в настоящее время протоколов маршрутизации ставит проблему поиска иного – комплексного подхода к разработке критериев качества работы информационных сетей, и на их основе новых видов маршрутных протоколов [23]. Анализ научных публикаций по этой теме показал, что тема многокритериальной маршрутизации становится в последнее время актуальной, в частности, рассматривается метод многопутевой маршрутизации, при котором «для передачи трафика одновременно используется несколько маршрутов к одному получателю»

В работе [1] автором сформулирована «задача многокритериальной маршрутизации в беспроводной Mesh-сети на графе между узлом-источником и узлом-назначением через ввод системы частных критериев, которая характеризует качество передачи информации от узла-источника к узлу-назначения на каждом промежуточном узле». Было высказано предположение, что «канальная скорость передачи на которой ведется передача данных с последующим узлом (например, для стандарта 802.11g возможны скорости 54, 48, 36, 24, 18, 12, 9, 6 Мбит/сек) оценивается частным критерием I_1 , загруженность узла I_2 , время задержки узлом I_3 , заряд батареи узла I_4 . Отметим, что критерии I_1 и I_4 максимизируются, а критерии I_2 и I_3 минимизируются. Приведем все к минимальному виду, для этого заменим критерии I_1 критерием $I_1 = I_{1m} - I_1$, где I_{1m} — максимальное возможное значение параметра. Аналогично поступим с критерием I_4 . Далее переходим к системе относительных частных критериев I_1 / I_{1m} , I_2 / I_{2m} , I_3 / I_{3m} , I_4 / I_{4m} . Очевидно, что диапазон изменения этих параметров лежит в пределах от нуля до единицы» [1].

«Известно, что задача многокритериальной оптимизации является некорректной, так как частные критерии качества конфликтуют между собой. Улучшение одного частного критерия качества ухудшает один или несколько других частных критериев качества. Регуляризация некорректной задачи многокритериальной оптимизации выполняется скалярной сверткой частных критериев качества по нелинейной схеме компромиссов

$$J = \sum_{i=1}^n \frac{a_i}{1 - \frac{I_i}{I_{im}}}, a_i \geq 0, \sum_{i=1}^n a_i = 1 \quad (5)$$

где I_i — i -й частный критерий качества; a_i — весовые коэффициенты; I_{im} — предельно-допустимое значение частного критерия качества I_i . При использовании протокола маршрутизации предлагается присваивать связям

между узлами метрику пропорциональный скалярной величине J , которая определяется по нелинейной схеме компромиссов (5).

Математическая модель mesh-сети в виде графа, веса (длина) ветвей которого рассчитываются по выражению (5) позволяет реализовать многокритериальную оптимизацию маршрутов передачи информации от узла-источника к узлу назначения путем минимизации критерия качества:

$$\min_i L = \sum_{j=1}^r \sum_{i=1}^n \frac{a_{ij}}{1 - I_{ijm}}, a_{ij} \geq 0, \sum_{i=1}^n a_{ij} = 1 \quad (6)$$

где I_{ij} – i -й частный критерий качества в j -й ветви графа; I_{ijm} – предельно-допустимое значение i -го частного критерия качества в j -й ветви графа; a_{ij} – весовые коэффициенты; n – количество частных критериев качества; r – количество ветвей графа вдоль маршрута от узла-источника к узлу-назначения.

Весовые коэффициенты a_{ij} позволяют задать приоритеты тех или иных каналов, если это необходимо. Такая ситуация может возникнуть, когда существуют несколько канальных интерфейсов и необходимо расставить приоритет между ними.

Задача минимизации критерия качества известна как задача о кратчайшем пути между узлом-источником и узлом-назначения. Следовательно, применение для расчета весов графа математической модели компьютерной сети свертки по нелинейной схеме компромиссов сводит задачу многокритериальной маршрутизации к известной задаче о кратчайшем пути, которая может быть решена алгоритмом Дейкстры либо параллельными средствами маршрутизации. Однако при использовании реактивных протоколов, когда каждый узел не имеет полной информации о полном графе сети, воспользоваться алгоритмом Дейкстры не представляется возможным, но в этом случае мы все же имеем большую вероятность, что построенный маршрут окажется кратчайшим путем.

Свойство адаптации нелинейной схемы компромиссов (6) к изменению ситуации в Mesh-сети позволяет повысить эффективность маршрутизации следующим образом. Допустим критерий I_4 (уровень заряда батареи беспроводного узла) начинает приближаться к своему минимально допустимому значению. В этом случае веса ветвей графа сети, входящих в этот узел, начнут возрастать и тем самым исключат его из кратчайших маршрутов, снизив нагрузку на этот узел, а, следовательно, и его энергопотребление, тем самым сохранив на большее время связность графа. В случае если узел получает неограниченное питание (к примеру, подключается к стационарной энергосети) критерий I_4 исключается из оптимизации. В этом случае составляющие весов ветвей графа по частному критерию качества дают постоянную величину.

Метод многокритериальной маршрутизации может быть усовершенствован [9, с. 28], если его сочетать с многопутевой маршрутизацией, которая позволяет найти несколько независимых параллельных маршрутов передачи информации с близкими оценками по выражению (6). Предлагается на втором этапе решения задачи повышения информационной безопасности перейти к решению задачи многопутевой маршрутизации по оценкам (2). В задаче многопутевой маршрутизации требуется найти несколько параллельных и независимых маршрутов передачи данных между узлом-источником и узлом-назначения информации, имеющие наименьшие характеристики длины маршрутов по оценкам (6). Особенность многопутевой маршрутизации заключается в том, что необходимо найти такие параметры $K > 1$ параллельных маршрутов, которые имеют наиболее близкие характеристики по длине к кратчайшему маршруту, но не имеют общих ветвей графа. Возможно, решить такую задачу многопутевой маршрутизации последовательным способом согласно следующей методике:

- однопутевая маршрутизация при помощи какого-либо протокола маршрутизации;

- исключение найденного маршрута между узлом-источником и узлом назначения информации из топологии графа, моделирующего компьютерную сеть;
- переход к первому пункту, если количество маршрутов меньше требуемого количества» [1].

2.3 Алгоритмы нахождения кратчайшего пути

Классическими решениями задачи на определение кратчайшего пути [20] во взвешенных графах являются алгоритм Дейкстры [8, с. 31], алгоритм Флойда-Уоршелла [8, с. 36] и алгоритм Форда-Беллмана [4].

Рассмотрим алгоритм Дейкстры [8, с. 31].

Поскольку топология Mesh-сети – ячеистая, математической моделью такой сети является неориентированный граф, вершинам которого соответствуют конечные узлы сети (компьютеры) и коммуникационное оборудование. Всем вершинам графа задается некоторое числовое значение – вес маршрута от начала до данной. Также существует понятие, как выделение вершины. Это означает, что путь от данной вершины до начальной является кратчайшим, однако если он таковым не является, то такой путь временный. При обходе графа, высчитывается маршрут для каждой из вершин, и, при условии, что данный путь кратчайший, вершина выделяется, и её вес становится равным весу пути [7]. При этом, такой же процесс происходит и с остальными вершинами, однако алгоритм их не выделяет. По окончании работы, алгоритм доходит до конечной вершины и присваивает значение веса конечной вершины кратчайшему пути.

Шаг 1. Всем вершинам, за исключением первой, присваивается вес равный бесконечности, а первой вершине – 0.

Шаг 2. Все вершины не выделены.

Шаг 3. Первая вершина объявляется текущей.

Шаг 4. Вес всех невыделенных вершин пересчитывается по формуле: вес невыделенной вершины есть минимальное число из старого веса данной

вершины, суммы веса текущей вершины и веса ребра, соединяющего текущую вершину с невыделенной.

Шаг 5. Среди невыделенных вершин ищется вершина с минимальным весом. Если таковая не найдена, то есть вес всех вершин равен бесконечности, то маршрут не существует. Следовательно, выход. Иначе, текущей становится найденная вершина, она же и выделяется.

Шаг 6. Если текущей вершиной оказывается конечная, то путь найден, и его вес есть вес конечной вершины.

Шаг 7. Переход на шаг 4.

Рассматривая реализацию алгоритма Дейкстры в программной части, то для всех построенных множеств S вершин, кратчайшие пути от начала заданы. По ходу программы, вершина, у которой будет наименьшее расстояние до начальной, будет прибавляться к множеству S . В процессе используется массив D , задача которого запоминать наиболее короткие пути для каждой из вершин. И только тогда, при условии, что множество S содержит абсолютно все вершины, массив D будет иметь длину, измеряемую от начальной вершины к каждой.

Помимо указанных массивов используется матрица длин C , где элемент $C[i, j]$ – длина ребра (i, j) , если ребра нет, то ее длина полагается равной бесконечности, то есть больше любой фактической длины ребер. Фактически матрица C представляет собой матрицу смежности, в которой все нулевые элементы заменены на бесконечность.

Для определения самого кратчайшего пути берется массив P вершин, где $P[v]$ будет содержать вершину, непосредственно предшествующую вершине v в кратчайшем пути.

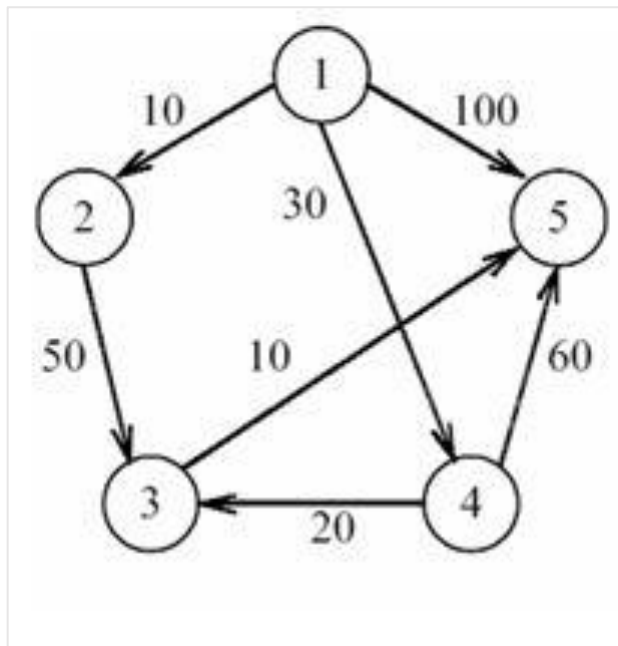


Рисунок 7 – Демонстрация алгоритма Дейкстры

Таблица 3 – Демонстрация алгоритма Дейкстры

Итерация	S	w	$D[2]$	$D[3]$	$D[4]$	$D[5]$
начало	{1}	–	10	∞	30	100
1	{1, 2}	2	10	60	30	100
2	{1, 2, 4}	4	10	50	30	90
3	{1, 2, 4, 3}	3	10	50	30	60
4	{1, 2, 4, 3, 5}	5	10	50	30	60

Массив P :

X	1	4	1	3
---	---	---	---	---

Кратчайший путь из 1 в 5: {1, 4, 3, 5}

Листинг 1. Описание функции алгоритма Дейкстры

```
void Dijkstra(int n, int **Graph, int Node){
    bool *S = new bool[n];
    int *D = new int[n];
    int *P = new int[n];
    int i, j;
    int Max_Sum = 0;
    for (i = 0 ; i < n ; i++)
```

```

    for (j = 0 ; j < n ; j++)
        Max_Sum += Graph[i][j];
for (i = 0 ; i < n ; i++)
    for (j = 0 ; j < n ; j++)
        if (Graph[i][j] == 0)
            Graph[i][j] = Max_Sum;
for (i = 0 ; i < n ; i++){
    S[i] = false;
    P[i] = Node;
    D[i] = Graph[Node][i];
}
S[Node] = true;
P[Node] = -1;
for ( i = 0 ; i < n - 1 ; i++ ){
    int w = 0;
    for ( j = 1 ; j < n ; j++ ){
        if (!S[w]){
            if (!S[j] && D[j] <= D[w])
                w = j;
        }
        else w++;
    }
    S[w] = true;
    for ( j = 1 ; j < n ; j++ )
        if (!S[j])
            if (D[w] + Graph[w][j] < D[j]){
                D[j] = D[w] + Graph[w][j];
                P[j] = w;
            }
    }
for ( i = 0 ; i < n ; i++ )
    printf("%5d",D[i]);
cout << endl;
for ( i = 0 ; i < n ; i++ )
    printf("%5d",P[i]+1);
cout << endl;
delete [] P;
delete [] D;
delete [] S;
}

```

Реализация алгоритма Дейкстры, состоит в том, что происходит построение множества вершин, у которых уже есть данные о кратчайших

путях. Всё, что идет дальше, это данные множества, к которым прибавляются вершины, по одной, без изменений длин оптимальных путей. Проблема алгоритма Дейкстры заключается в том, какие способы нахождения, хранения и обновления используются. В случае небольшой нагрузки на сеть этот алгоритм довольно эффективен, однако по мере увеличения трафика промежуточные узлы, которые находятся на путях маршрутов, выбираются чаще по сравнению с остальными и получают больше пакетов, чем могут обработать. В результате длина очереди в этих промежуточных узлах увеличивается, что приводит к увеличению среднего времени доставки пакетов.

Следующий алгоритм нахождения кратчайшего пути – алгоритм Флойда-Уоршелла [8, с. 36]. Этот алгоритм более общий по сравнению с алгоритмом Дейкстры, так как он находит кратчайшие пути между любыми двумя вершинами графа. В алгоритме Флойда [8] используется матрица A размером $n \times n$, в которой вычисляются длины кратчайших путей. Элемент $A[i, j]$ равен расстоянию от вершины i к вершине j , которое имеет конечное значение, если существует ребро (i, j) , и равен бесконечности в противном случае.

Основная идея алгоритма:

Пусть есть три вершины i, j, k и заданы расстояния между ними. Если выполняется неравенство $A[i, k] + A[k, j] < A[i, j]$, то целесообразно заменить путь $i \rightarrow j$ путем $i \rightarrow k \rightarrow j$. Такая замена выполняется систематически в процессе выполнения данного алгоритма.

Шаг 0. Определяем начальную матрицу расстояния A_0 и матрицу последовательности вершин S_0 . Каждый диагональный элемент обеих матриц равен 0, таким образом, показывая, что эти элементы в вычислениях не участвуют. Полагаем $k = 1$.

Основной шаг k . Задаем строку k и столбец k как ведущую строку и ведущий столбец. Рассматриваем возможность применения замены описанной выше, ко всем элементам $A[i, j]$ матрицы A_{k-1} . Если выполняется неравенство

$$A[i, k] + A[k, j] < A[i, j], (i \neq k, j \neq k, i \neq j) \quad (8)$$

тогда выполняем следующие действия:

- создаем матрицу A_k путем замены в матрице A_{k-1} элемента $A[i, j]$ на сумму $A[i, k] + A[k, j]$;
- создаем матрицу S_k путем замены в матрице S_{k-1} элемента $S[i, j]$ на k .
Полагаем $k = k + 1$ и повторяем шаг k .

Таким образом, алгоритм Флойда делает n итераций, после i -й итерации матрица A будет содержать длины кратчайших путей между любыми двумя парами вершин при условии, что эти пути проходят через вершины от первой до i -й [8, с. 36]. На каждой итерации перебираются все пары вершин и путь между ними сокращается при помощи i -й вершины.

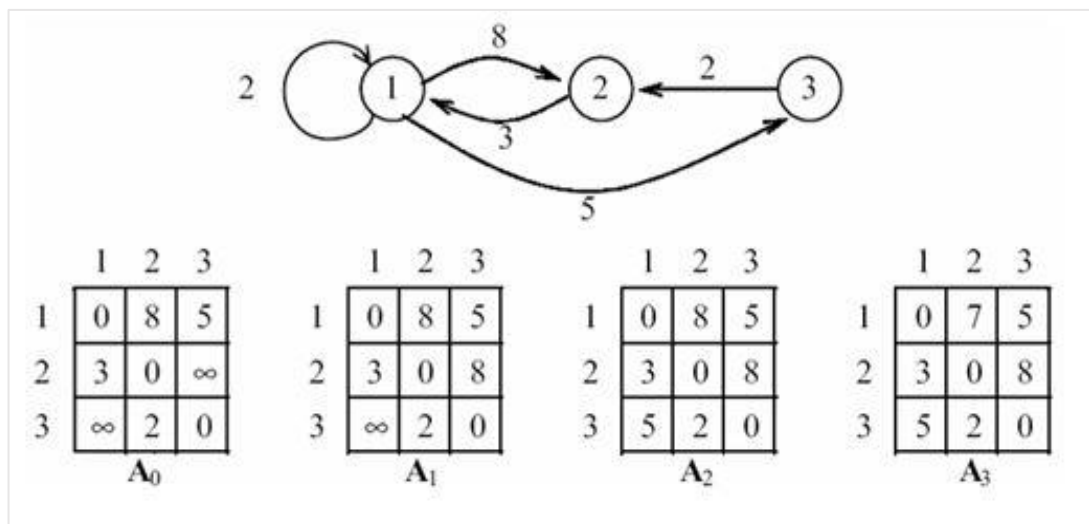


Рисунок 8 – Демонстрация алгоритма Флойда

Листинг 2. Описание функции алгоритма Флойда

```
void Floyd(int n, int **Graph, int **ShortestPath){
    int i, j, k;
    int Max_Sum = 0;
    for ( i = 0 ; i < n ; i++ )
        for ( j = 0 ; j < n ; j++ )
            Max_Sum += ShortestPath[i][j];
    for ( i = 0 ; i < n ; i++ )
        for ( j = 0 ; j < n ; j++ )
            if ( ShortestPath[i][j] == 0 && i != j )
```

```

    ShortestPath[i][j] = Max_Sum;
for ( k = 0 ; k < n; k++ )
  for ( i = 0 ; i < n; i++ )
    for ( j = 0 ; j < n; j++ )
      if ((ShortestPath[i][k] + ShortestPath[k][j]) <
          ShortestPath[i][j])
        ShortestPath[i][j] = ShortestPath[i][k] +
          ShortestPath[k][j];
}

```

Рассматривая метод Флойда, можно прийти к выводу, что его основа состоит в том, что в графах, у которых вес ребер положительный, наипростейший, наименьший по длине путь содержит другие кратчайшие пути. При этом, в случае, когда граф изображен в виде матрицы смежности, выполняться алгоритм будет в порядке $O(n^3)$ [8].

Полученные результаты привели к необходимости разработки адаптивных алгоритмов маршрутизации, в которых путем исследования сети формируются альтернативные маршруты, имеющие большую длину с точки зрения количества прыжков, но приводящие к меньшему времени доставки. В таких алгоритмах каждому узлу необходимо владеть информацией о сети, которую он может использовать для принятия решений о выборе маршрута, и для которой, кроме того, существует механизм обновления сведений о состоянии сети. Такому запросу соответствует алгоритм Беллмана-Форда.

Более эффективным использованием данного алгоритма Беллмана-Форда, если работать с графами, у которых вес ребер имеет отрицательное значения. Однако область применения при этом ограничивается работой не всеми такими графами, так как при каждом прогоне пути, состоящего из отрицательных значений, будет улучшаться необходимое значение. Тем не менее, даже если проводить улучшение большое количество раз, невозможно будет добиться оптимального значения. Следовательно, данный алгоритм плохо работает с графами, у которых циклы отрицательны. Однако есть возможность узнать, наличие подобных.

Основная идея алгоритма Беллмана-Форда [24]:

Пусть есть граф и начальная вершина `src`. Необходимо найти кратчайшее расстояние до всех вершин от `src`. Если попадается цикл отрицательного веса, то самые короткие расстояния не вычисляются, выводится сообщение о наличии такого цикла.

Шаг первый. Инициализируются расстояния от исходной вершины до всех остальных вершин, как бесконечные, а расстояние до самого `src` принимается равным 0. Создается массив `dist[]` размера $|V|$ со всеми значениями равными бесконечности, за исключением элемента `dist[src]`, где `src` — исходная вершина.

Шаг второй. Вычисляются самые короткие расстояния. Следующие шаги нужно выполнять $|V|-1$ раз, где $|V|$ — число вершин в данном графе.

Произведите следующее действие для каждого ребра `u-v`:

Если $\text{dist}[v] > \text{dist}[u] + \text{вес ребра } uv$, то обновите `dist[v]`
 $\text{dist}[v] = \text{dist}[u] + \text{вес ребра } uv$

Шаг третий. Сообщается, присутствует ли в графе цикл отрицательного веса. Для каждого ребра `u-v` необходимо выполнить следующее:

Если $\text{dist}[v] > \text{dist}[u] + \text{вес ребра } uv$, то в графе присутствует цикл отрицательного веса.

Идея третьего шага заключается в том, что шаг второй гарантирует кратчайшее расстояние, если граф не содержит цикла отрицательного веса. Если мы снова переберем все ребра и получим более короткий путь для любой из вершин, это будет сигналом присутствия цикла отрицательного веса.

Листинг 3. Алгоритм Беллмана-Форда

```
# shortest path algorithm.  
# Class to represent a graph  
class Graph:  
  
    def __init__(self, vertices):  
        self.V = vertices # No. of vertices  
        self.graph = [] # default dictionary to store graph
```

```

# function to add an edge to graph
def addEdge(self, u, v, w):
    self.graph.append([u, v, w])

# utility function used to print the solution
def printArr(self, dist):
    print("Vertex Distance from Source")
    for i in range(self.V):
        print("%d \t\t %d" % (i, dist[i]))

# The main function that finds shortest distances from src to
# all other vertices using Bellman-Ford algorithm. The function
# also detects negative weight cycle
def BellmanFord(self, src):

    # Step 1: Initialize distances from src to all other vertices
    # as INFINITE
    dist = [float("Inf")] * self.V
    dist[src] = 0

    # Step 2: Relax all edges |V| - 1 times. A simple shortest
    # path from src to any other vertex can have at-most |V| - 1
    # edges
    for i in range(self.V - 1):
        # Update dist value and parent index of the adjacent
        # the picked vertex. Consider only those vertices which
        # are still in
        # queue
        for u, v, w in self.graph:
            if dist[u] != float("Inf") and dist[u] + w < dist[v]:
                dist[v] = dist[u] + w

    # Step 3: check for negative-weight cycles. The above step
    # guarantees shortest distances if graph doesn't contain
    # negative weight cycle. If we get a shorter path, then there
    # is a cycle.

    for u, v, w in self.graph:
        if dist[u] != float("Inf") and dist[u] + w < dist[v]:
            print "Graph contains negative
weight cycle"

    return

```

```

        # print all distance
        self.printArr(dist)

g = Graph(5)
g.addEdge(0, 1, -1)
g.addEdge(0, 2, 4)
g.addEdge(1, 2, 3)
g.addEdge(1, 3, 2)
g.addEdge(1, 4, 2)
g.addEdge(3, 2, 5)
g.addEdge(3, 1, 1)
g.addEdge(4, 3, -3)

# Print the solution
g.BellmanFord(0) [24]

```

При выполнении алгоритма Беллмана-Форда каждый узел связи хранит две маршрутные таблицы. Одна из них содержит минимальные значения весовых коэффициентов, выбор которых позволяет достичь любого узла назначения (таблица стоимости). Другая – идентификационный номер соседнего узла, которому должен быть передан пакет с целью достижения пункта назначения с минимальной стоимостью (таблица маршрутизации). Обновление вышеуказанных таблиц осуществляется путем периодического обмена информацией таблицами соседних узлов. Для условий низкой нагрузки минимальная стоимость в алгоритме Беллмана-Форда измеряется с точки зрения количества прыжков, по мере увеличения нагрузки – с точки зрения времени доставки пакета. Несмотря на возможность выбора параметра, на основе которого рассчитываются кратчайшие пути, в этом алгоритме остаются два существенных изъяна. Если узел, имеющий минимальное значение стоимости, случайно выпал из сети на короткий промежуток времени и пропал в момент запроса на построение маршрута, после возвращения в сеть он какое-то время не будет использоваться, даже если станет лучшим выбором. Это свидетельствует о недостаточной поддержке стабильности маршрутизации.

Помимо этого, алгоритму требуется больше пропускной способности сети из-за частых обменов таблицами между соседями.

2.4 Альтернативные алгоритмы маршрутизации

Одним из интересных направлений в развитии алгоритмов маршрутизации является применение алгоритмов семейства Q-routing, использующих принципы обучения с подкреплением [29].

Маршрутизация на основе обучения с подкреплением основана на использовании одного из способов машинного обучения, в ходе которого испытываемая система (агент) обучается, взаимодействуя с некоторой средой.

Обучение с подкреплением является подразделом машинного обучения, исследующим как агент должен действовать в окружении, чтобы максимизировать некоторый долговременный выигрыш [21]. Алгоритмы с частичным обучением пытаются найти стратегию, приписывающую состояниям окружающей среды действия, которые должен предпринять агент в этих состояниях. Одним из алгоритмов обучения с поощрением является алгоритм Q-learning [28].

Q-learning – метод, применяемый при построении искусственного интеллекта при агентном подходе. Относится к экспериментам вида «обучение с подкреплением». На основе получаемого от среды вознаграждения агент формирует функцию полезности Q, что впоследствии дает ему возможность уже неслучайно выбирать стратегию поведения, а учитывать опыт предыдущего взаимодействия со средой. Одно из преимуществ Q-обучения то, что оно в состоянии сравнить ожидаемую полезность доступных действий, не формируя модели окружающей среды. Применяется для ситуаций, которые можно представить в виде марковского процесса принятия решений [28].

Алгоритм Q-обучения можно представить следующим образом:

1. Задаем параметр γ и матрицу вознаграждений R.
2. Устанавливаем значения элементов матрицы Q – матрицы полученного обучения – равными 0.

3. Для каждого эпизода (подразумевается один раз нахождение выхода из лабиринта) выполняем следующие действия:

- выбираем случайным образом исходное состояние;
- пока не найден выход для текущего состояния, выбираем один из возможным из возможных переходов в другое состояние;
- рассматривая этот переход как следующее состояние, в нем выбираем максимальное значение Q из всех возможных переходов;
- вычисляем вознаграждение для выбранного перехода по следующей формуле:

$$Q_{(\text{состояние,переход})} = R_{(\text{состояние,переход})} + \gamma \cdot \text{Max}[Q_{(\text{следующее состояние,все переходы})}] \quad (8)$$

- переходим в следующее состояние, устанавливаем его текущим, выполняем вышеперечисленные действия до тех пор, пока агент не достигнет выхода из лабиринта [20].

3 Выбор оптимального алгоритма маршрутизации для Mesh-сетей

3.1 Факторы, влияющие на эффективность маршрутизации

На эффективность маршрутизации оказывает влияние множество факторов, таких как: интенсивность формирования исходящего трафика в сети, количество узлов в сети, скорость переконфигурирования сети, зависящая от мобильности каждого узла в сети, связность сети или количество соединений с соседними узлами в зоне радиовидимости, вероятность доставки информации любому из действующих узлов сети, уровень заряда батареи узлов [10]. Кроме того, на выбор маршрута влияет наличие очереди сообщений в буфере каждого конкретного узла, расстояние между узлами, скорость передачи и приема между каждой отдельной парой связанных узлов, создание петель при передаче пакета, время на обработку пакета и передачу при необходимости другому узлу, время, требуемое на формирование информации о конфигурации сети [12]. В зависимости от типа беспроводной сети при проведении ранжирования факторов полученные результаты могут незначительно отличаться друг от друга. В связи с этим необходимо выполнить ранжирование факторов, обеспечивающих повышение эффективности маршрутизации для Mesh-сетей. Построение линейки факторов по степени значимости необходимо провести относительно влияния вышеопределенных факторов на качество передачи данных [13, с. 16]. Качество связи в интернет-сети традиционно складывается из совокупности следующих характеристик:

1. Максимальная функциональность.
2. Скорость выполнения произвольных операций.
3. Пропускная способность.
4. Отказоустойчивость и надежность соединений.
5. Низкий уровень ошибок.
6. Безопасность информации.
7. Доступность сети.
8. Масштабируемость.

9. Универсальность сети.

Первый параметр подразумевает предоставление сетью доступа ко всем ресурсам сети, организацию совместной работы узлов и реализацию всех протоколов и стандартов работы. Второй характеризуется количеством обработанных за единицу времени запросов пользователей сети. Третий параметр определяется объемом данных, передаваемых через сеть за единицу времени. Среднее время работы сетевых узлов до выработки ресурса прочности представляет четвертый параметр. Под низким уровнем ошибок подразумевается своевременность и достоверность получаемой пользователем информации. Способность защитить информацию от несанкционированного доступа определяет уровень безопасности сети. Седьмой параметр аналогичен инкапсуляции в объектно-ориентированном программировании, характеризующийся сокрытием от пользователя реализации объекта и доступностью интерфейса. Возможность расширения сети без заметного снижения ее производительности и подключение различных типов и видов устройств характерны для восьмого и девятого показателей качества.

С точки зрения возможностей алгоритма маршрутизации доступными являются первые пять показателей. Одним из наиболее важных факторов, влияющих на полноту выполняемых сетью функций, является функционирование сети в целом, то есть доступность услуги зависит от наличия связей между узлами сети, отсутствие или нарушение связности сети приводит к ухудшению показателей качества услуг или полному её отсутствию. Следующие три параметра: производительность, пропускная способность и надежность сети - также зависят от наличия связей между узлами, так как при отсутствии достаточной связности сети основная нагрузка будет приходиться на конкретные узлы, поэтому количество обработанных за единицу времени запросов в них будет падать с течением времени. В результате этого, те немногочисленные связи в сети, которые имеются, будут со временем разрушаться, что будет сказываться на функционировании сети. Достоверность информации обеспечивается выбором наиболее эффективных

маршрутов, которые определяет алгоритм маршрутизации, учитывая основные факторы, характеристики которых влияют на скорость и качество передаваемой по сети информации. В результате можно сделать вывод, что четыре из пяти показателей, так или иначе, зависят от связности сети, то есть от наличия связей между узлами сети.

В Mesh-сетях, как правило, связность обеспечивается цепочками между узлами, то есть в них радиус действия одного узла слишком мал, чтобы обеспечить передачу от пункта «А» до пункта «Б» [10]. Поэтому в таких сетях всегда строятся цепочки, по которым передается информация. У этих цепочек очень часто небольшая связность. То есть между узлом А и Б есть несколько критичных узлов, в обход которых информацию передать невозможно. Например, множество узлов находится в диапазоне одного единственного узла, который в то же время находится в зоне видимости другого множества узлов, в то время как оба множества не пересекаются друг с другом. В таком случае – данный узел является критичным узлом, обойти который не представляется возможным. Когда связность небольшая, выход из строя любого такого узла обрушает связность сети в полном объеме. Когда критичный узел один, выбора нет, информация передается через него до того момента, пока данный узел находится в доступе. Когда таких узлов несколько (с ограниченным ресурсом энергозаряда), отсутствие учета такого фактора как «уровень заряда батареи» приводит к возможности быстрого обрушения связности сети.

3.2 Проблема многокритериальной маршрутизации

Теоретически для выбора наилучшего способа маршрутизации необходимо учесть целый ряд характеристик, чтобы картина состояния сети была полной, однако на практике расчет универсальной формулы эффективной маршрутизации ячеистых самоорганизующихся сетей с учетом всех параметров стремится к бесконечному усложнению как самого алгоритма, так и способа его воплощения. Реалистичнее выглядят решения, рассчитанные на

определенную модель использования сети. Вот несколько примеров ситуативных сценариев, в которых нужно организовать беспроводную сеть:

- Локально организованные небольшие Mesh-сети, рассчитанные на использование в стенах одной квартиры или дома на одну семью для интеллектуальной связи, так называемой системы «Умный дом» [14], устройств бытовой техники, стационарных и портативных компьютеров, смартфонов, телевизоров и пр.
- Офисные Mesh-сети, соединяющие компьютерные устройства внутри одного офиса или офисного здания в локальную сеть, и обеспечивающие безопасный и бесперебойный доступ в Интернет.
- Mesh-сети университетских городков или общежитий предприятий, осуществляющие беспроводную мобильную связь со стабильным сигналом на ограниченной территории.
- Общественные, созданные коммерсантами или муниципальными органами самоуправления Mesh-сети, предоставляющие в социально значимых районах со значительным пешеходным трафиком бесплатные точки доступа к Интернет-связи.

В зависимости от условий, в которых будет реализовываться сеть, технические характеристики узловых элементов mesh-сетей будут оцениваться по-разному. Так, в домашних mesh-сетях преимущество будет на стороне простоты обслуживания и низкой стоимости устройств, а также умеренности энергопотребления и мобильности [14]. В офисных Mesh-сетях приоритет будет отдаваться поддержке качества обслуживания и скорости соединения. В коммерческих общественных mesh-сетях – свобода доступа к сетевым ресурсам, стабильность сигнала.

Как можно понять из вышесказанного, дифференциация области применения беспроводных ячеистых сетей делает практически нереализуемым детальный и регламентируемый сценарий организации mesh-сетей, общий для любых условий. Поэтому последняя версия дополнения IEEE 802.11s носит условно ограничительный характер: озвучены

основополагающие правила работы mesh-сетей [22], при этом на усмотрение производителей оставляется техническая реализация этих правил. Перечислен необходимый и достаточный состав механизмов, который должен быть включен в любое устройство, заявленное как элемент mesh-сети. Это гарантирует совместимость устройств любых производителей, хотя и не обещает высокую эффективность сетевых услуг.

4 Моделирование локальной mesh-сети

4.1 Определение задачи исследования

Развитие беспроводных самоорганизующихся сетей давно перестало быть инновацией, многие крупные производители сетевого оборудования выпустили на рынок доступные для бытового использования системы маршрутизаторов для построения небольшой локальной Mesh-сети. Естественно, такие же системы, только усложненные для увеличения мощности, предлагаются для бизнес-проектов. Стоит учесть, однако, что, если количество устройств сети (допустим, датчиков влажности или дымоуловителей) чрезмерно, стоимость реализации подобного проекта превышает разумные пределы финансовых затрат и сопоставима с потерями, которые предполагалось компенсировать созданием такого рода сетей.

Рассмотрим крупный объект промышленного назначения, на котором планируется установить сеть из датчиков движения. Датчики, расположенные в определенном порядке, должны фиксировать любые движения на территории объекта и передавать информацию центральному узлу, который имеет возможность выхода в глобальную сеть. Сенсоры движения имеют ограниченную зону охвата – не более 10м, и могут связываться только с ближайшими соседями. Задача заключается в объединении всех устройств в общую одноранговую Mesh-сеть, для возможности получения и передачи информации каждым датчиком. Для решения этой задачи необходимо подобрать или разработать такой алгоритм маршрутизации, чтобы сетью могли выполняться следующие условия: передавать и получать пакеты данных на все узлы сети, определять вышедшие из строя точки, находить им замену и изменять маршрут, работать в энергосберегающем режиме, чередуя циклы «сна» и «работы», сохранять эффективную работоспособность как можно долго. Требуется смоделировать полученную сеть с вышеперечисленными свойствами на одном из языков программирования, опробовать работу различных алгоритмов маршрутизации в этой сети и сравнить среднее время

передачи данных между узлами с целью выбора наиболее оптимального способа маршрутизации.

4.2 Реализация задачи исследования

Как уже упоминалось, сеть планируется организовать из датчиков движения и/ или присутствия, работающих на сочетании инфракрасного и радиоволнового принципов действия. В данной работе планируется построить модель сети и протестировать ее работу на языке C# под платформой .NET Framework 4.5. В ходе работы использовалась IDE Visual Studio 2013.

Механизм работы сети следующий: во время получения сигнала о движении любым датчиком, информация об этом передаётся устройством в сеть, конечным получателем значится корневой узел, имеющий выход в Интернет, в отсутствие срабатывания сенсоров корневой узел отправляет в сеть широковещательный запрос с определенной периодичностью с целью выявления вышедших из строя узлов и уточнения топологии сети. Как можно заметить, планируется передавать большое число сообщений, соответственно оптимальным будет использовать проактивный протокол маршрутизации AODV, поскольку узлы сети стационарны, и топология сети не будет динамически изменяться.

При подключении каждый датчик должен установить положение и удаленность корневого узла, для чего им делается широковещательный запрос, в котором есть информация об адресе отправителя и список адресов уже пройденных узлов.

Таблица 4 – Формат широковещательного запроса

Id агента инициатора 9 бит	Список посещенных агентов по 9 бит
----------------------------------	--

При получении этого запроса транзитный узел сети проверяет список адресов и вписывает свой, если запрос получен впервые, после чего запрос передается дальше. После получения сообщения корневым узлом –конечным адресатом – им же отправляется ответный запрос-подтверждение по маршруту, состоящему из списков адресов к исходному отправителю.

Получив запрос-подтверждение, отправитель включает список адресов в таблицу маршрутизации и при следующей передаче использует этот путь. Однако если корневым узлом получен запрос с более коротким списком адресов, то в сеть отправителю передается новый запрос-подтверждение, и адресант меняет данные в своей таблице на актуальные. То же происходит, если один из узлов выходит из строя и не принимает участия в приеме-передаче сообщений – хотя маршрут удлиняется, информация о нем вносится в обновляемые данные (рис. 9).

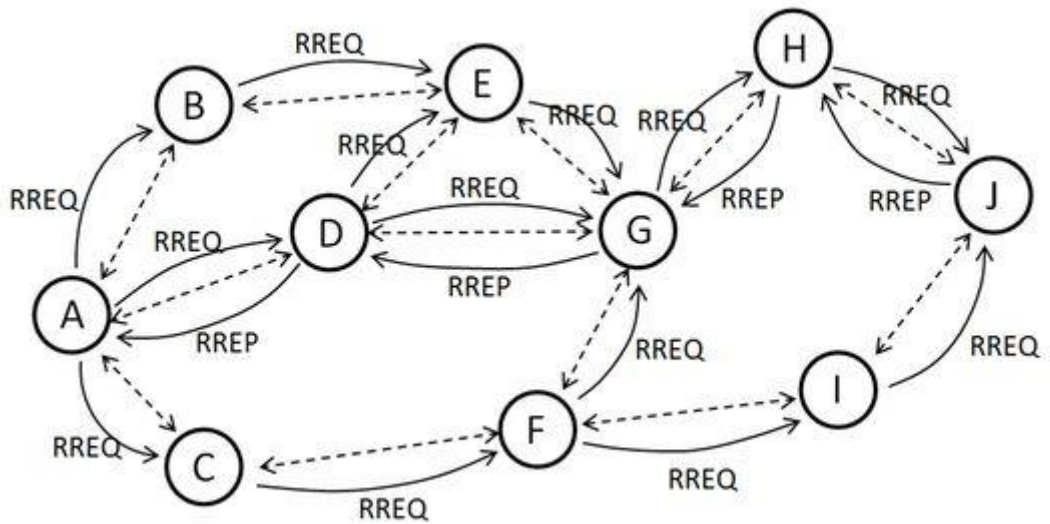


Рисунок 9 – Установление связи между агентами

Задачей текущего исследования стоит нахождение оптимального маршрута и ускорение времени пересылки сообщения. Датчики движения/присутствия пересылают небольшой объем рабочей информации, пакеты служебной информации значительно превышают их по размеру, следовательно, рабочими объемами можно пренебречь, а уменьшить пересылаемые служебные сообщения. Для кодирования адреса ID каждого датчика достаточно использовать девять бит, что дает 512 вариантов. Тогда служебное сообщение будет выглядеть следующим образом:

- ID агента отправителя (9 бит),
- ID агента получателя (9 бит),
- список ID адресов транзитных точек (9 бит для каждого).

Таблица 5 – Заголовок пакета с данными

Id агента отправителя 9 бит	Id агента получателя 9 бит	Список агентов по 9 бит
-----------------------------------	----------------------------------	----------------------------

Можно вывести формулу размера для каждого пакета служебных данных

$$18+9N,$$

где N – число датчиков, участвовавших в передаче.

При организации сети следует учесть, что нагрузка на каналы связи будет распределена неравномерно, очевидно, что через ближайšie к корневому узлу устройства будет проходить значительно больше сообщений, чем в среднем по сети. Это увеличивает вероятность выхода датчиков из строя, помимо неизбежного возникновения коллизий. Значит, необходимо внести дополнения в протокол маршрутизации AODV, чтобы дозировать нагрузку на проблемные узлы, допустим, чередовать маршруты через соседние с корневым узлом точки, объединять транзитным узлом в процессе передачи сообщения с одинаковым маршрутом, что кстати уменьшит объем служебной информации. Помимо этого, изначально можно расположить рядом с корневым узлом большее число датчиков.

4.3 Результаты проведенной работы

В процессе работы была создана модель сети с неоднородно расположенными сенсорами движения по некоторой области. Для исследования работы протокола маршрутизации в различных условиях произвольно менялись исходные данные: число узлов в сети, физическая топология сети, размеры объекта расположения сети.

Для решения поставленных задач работа проводилась в два этапа. На первом этапе была исследована работа протокола AODV до и после внесения корректировок, сокращающих объем служебного пакета данных. Следует отметить, что размер сообщения не влияет на количество ошибок при передаче, поскольку даже при одной ошибке во время беспроводной связи отбрасывается весь пакет.

Для проверки каждого комплекса метрик было смоделировано пятьдесят сетей сенсорных датчиков, а затем последовательно запускались протокол AODV и дополненный протокол. Для приближения загруженности сети к

реальным условиям, когда сенсорный датчик реагирует на движение/присутствие, было взято за основу возникновение такой ситуации с однородной частотой для каждого узла сети. В случае выхода из строя какой-либо точки сети маршрут перестраивался, если выходило из строя более 20 процентов, работа сети прекращалась. Замерялось число всех сообщений, включая служебные, переданных по сети, затем рассчитывалось среднее число, полученное корневым узлом от каждого устройства. Результаты замеров представлены в Таблице 6.

Таблица 6 – Результаты тестирования без объединения сообщений

Количество агентов	30	30	300	300	300	300
Радиус связи	15	50	15	50	50	75
Размер поля	50	200	200	200	500	500
Среднее количество сообщений (AODV)	2200	3800	1800	4300	3100	3900
Среднее количество сообщений (балансировка)	2700	4400	2600	6100	4200	4800
Медиана (AODV)	2500	3800	1700	4500	2500	4000
Медиана (балансировка)	3000	4400	2400	6300	4000	4900
Максимум (AODV)	4100	6500	4700	7000	5900	6300
Максимум (балансировка)	4800	8300	5200	8100	7400	7500
Минимум (AODV)	1300	1600	1100	1300	900	1100
Минимум (балансировка)	1500	2000	1300	1400	1300	1600
Доверительный интервал 95% (AODV)	(1900-2600)	(3100-4500)	(1400-2300)	(3900-4800)	(2700-3800)	(3500-4300)
Доверительный интервал 95% (балансировка)	(2100-3000)	(3500-4900)	(2100-3000)	(5500-7000)	(3700-4700)	(4400-5100)

Данные таблицы указывают на явное преимущество протокола, дополнительно сокращающего объем служебных сообщений.

Второй этап исследования был идентичен первому, за исключением того, что теперь измерялась и сравнивалась работа протокола AODV с работой протокола, допускающего объединение служебных сообщений. Необходимо было определить насколько часто возникают ситуации, в которых объединение возможно. При редком наступлении такого рода событий заметного

сокращения объем передаваемых данных нет, а значит внесение корректировки избыточно. Чтобы увеличить частоту наступления желаемых условий, было решено делать объединение после поступления не менее десяти пакетов с одинаковыми маршрутами. Результаты выполнения второго этапа представлены в Таблице 7.

Таблица 7 – Результаты тестирования с объединением сообщений

Количество агентов	30	30	300	300	300	300
Радиус связи	15	50	15	50	50	75
Размер поля	50	200	200	200	500	500
Среднее количество сообщений (AODV)	2600	4400	2300	4900	3300	4500
Среднее количество сообщений (балансировка)	3100	4600	2900	6500	4600	5100
Медиана (AODV)	2600	3900	2400	4800	3500	4600
Медиана (балансировка)	3200	4500	3000	6500	4800	5300
Максимум (AODV)	4500	6900	4600	7000	6400	6500
Максимум (балансировка)	5000	8600	5300	7900	7600	7700
Минимум (AODV)	1400	1700	1300	1500	1200	1200
Минимум (балансировка)	1600	2400	1800	1700	1300	1800
Доверительный интервал 95% (AODV)	(2200-3100)	(3900-5000)	(1600-2900)	(4400-5400)	(2900-3800)	(4000-5100)
Доверительный интервал 95% (балансировка)	(2500-4000)	(4100-5300)	(2400-3500)	(5900-7000)	(4000-5100)	(4600-5400)

Заметно, что число транзакций выросло, однако необходимо учесть, что частота сообщений задана искусственно, а в реальности важно своевременное получение сообщения от сенсорного датчика корневым узлом. Это частный

случай работы сети по изменённому протоколу, когда число рабочих сообщений от датчиков настолько велико, что оптимальным будет объединение сообщений во избежание коллизий.

Заключение

В ходе проведённой работы были исследованы различные алгоритмы маршрутизации, построенные с учетом различных параметров качества сети. Сформулирована задача многокритериальной маршрутизации в беспроводной ячеистой сети посредством введения различных метрик в механизм передачи сообщений, решение которой обуславливает качество и надежность сетевой связи.

Выяснено, что задача выбора оптимального способа маршрутизации трафика является многоуровневой, так как различные параметры качества вступают в противоречие между собой – улучшение одной характеристики качества может повлечь за собой ухудшение других характеристик связи.

Эффективность работы сети зависит от многих параметров. Повысить ее и довести до оптимальной возможно лишь анализируя максимальное количество параметров-критериев ее работы. Процесс анализа должен быть распределенный (все узлы в равной мере анализируют работу сети). После анализа должна проводиться адаптация по некому оптимизирующему правилу. Особенно это важно в сетях с быстро изменяющейся топологией.

Список используемых источников

1. Афанасьев А.Л., Гармонов А.В. Многокритериальная многопутевая маршрутизация в mesh-сетях. // Связь и телекоммуникации – инновационное развитие регионов: науч.-техн. конф. Воронеж. 2011. URL: <http://masters.donntu.org/2013/fkita/gusev/library/3.htm> (дата обращения 30.05.2021)
2. Бельфер Р.Ф. IEEE 802.11// Национальная библиотека им. Н. Э. Баумана. 28.04.2012. URL: https://ru.bmstu.wiki/IEEE_802.11 (дата обращения 21.05.2021)
3. Вишневский В.М., Портной С.Л., Шахнович И.В. Энциклопедия WIMAX. Путь к 4G. М.: Техносфера. 2009. 472с.
4. Гданский Н.И., Куликова Н.Л., Чумакова Е.В. Точное решение задачи поиска минимального ациклического пути во взвешенных графах, содержащих ребра отрицательного веса. // Программные продукты и системы. 2017. Том 31. №2. С. 118–124.
5. Голубничая Е.Ю., Лихтциндер Б.Я. Упрощённый алгоритм маршрутизации в Wi-Fi Mesh-сетях мониторинга. // Инфокоммуникационные технологии. 2014. Том 12. №2. С 53–57.
6. Даденков С.А. Разработка алгоритмов проектирования подсистем сбора и передачи информации автоматизированных систем управления технологическими процессами с требуемыми временными характеристиками. Пермский национальный исследовательский политехнический университет. Пермь. 2016. 200 с.
7. Дмитриев М. Г., Ломазов В. А. Оценка чувствительности линейной свертки частных критериев при экспертном определении весовых коэффициентов // Искусственный интеллект и принятие решений. 2014. № 1. С. 52–56.
8. Дольников В.Л., Якимова О.П. Основные алгоритмы на графах: учеб. пособие для студентов. Ярославль: ЯрГУ. 2011. 80 с.

9. Евсева О.Ю. Модель маршрутизации и распределение канальных ресурсов WiMax-сети // Телекоммуникационные сети и систем. 2011. С. 111 – 115.
10. Кирьянов А.В. Анализ механизмов построения логической топологии в сетях MANET // Информационные процессы. 2015. Том 15. №2. С. 183–197.
11. Колыбельников А. И. Обзор технологий беспроводных сетей // Труды Московского физико-технического института. 2012. Том 4. №2-14. С. 4–28.
12. Кулаков Ю.А., Воротников В.В. Энергоэффективная иерархическая маршрутизация в самоорганизующихся динамических сетях. // Управляющие системы и машины. 2014. №1. С. 70 – 76.
13. Ногин, В. Д. Принятие решений в многокритериальной среде. Количественный подход. / М.: Физматлит. 2002. 144 с.
14. Плотников А.В. Интернет вещей: инновационные технологии, меняющие человечество. // Московский экономический журнал. 2019. №7. С. 658–665.
15. Прокопенко А.Ю. Группа стандартов WiMAX // Научно-технический вестник информационных технологий, механики и оптики. 2006. №29. С. 113–117.
16. Разиньков С.Н. Основные направления развития и базовые технологии создания систем радиосвязи со сверхширокополосными сигналами// Воздушно-космические силы. Теория и практика. 2019. №11. С. 38–44.
17. Смирнова Е.В., Пролетарский А.В., Ромашкина Е.А., Балюк С.А., Суоров А.М. Технологии современных беспроводных сетей Wi-Fi. М.: ИМГТУ им. Н.Э. Баумана. 2017. 448 с.
18. Соколов В. А., Корсаков С. В., Смирнов А. В., Башкин В. А., Никитин Е. С. Инструментальная система для поддержки разработки и исследования программно-конфигурируемых сетей подвижных объектов. // Моделирование и анализ информационных систем. 2015. Том 22. №4. С. 546–562.

19. Хоров Е.М. Выбор периода резервирования канала в самоорганизующихся беспроводных сетях // Информационные процессы. Передача информации в компьютерных сетях. 2015. Том 15. № 1. С. 78–88.

20. Шилова Ю. А. Q-routing algorithm simulation in Anylogic // Инновационные процессы в исследовательской и образовательной деятельности // Материалы VI регион. науч.-практ. конф. молодых ученых. 2017. П.: ИПНИПУ. 2017. С. 57–62.

21. Шилова Ю.А. Верификация имитационной модели алгоритма маршрутизации Adaptive Rate Full Echo, разработанной в среде имитационного моделирования Anylogic // Моделирование, оптимизация и информационные технологии. 2019. С. 1-11.

22. Aiache H., Conan V., Lebrun L., Rousseau S. A load dependent metric for balancing Internet traffic in Wireless Mesh Networks // 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems. 2008. С. 629-634.

23. Antonio de la Oliva. Providing throughput guarantees in heterogeneous wireless mesh networks. // Wireless communications and mobile computing. 2013. С. 10–25.

24. Bellman–Ford Algorithm DP– 23//Электронный ресурс Geeksforgeeks.com. URL: <https://www.geeksforgeeks.org/bellman-ford-algorithm-dp-23/> (дата обращения 19.05.2021)

25. Global Digital 2021// Электронный журнал We Are Social. 20.01.2021. URL: <https://datareportal.com/reports/digital-2021-global-overview-report> (дата обращения: 25.05.2021)

26. IEEE Standard for Information technology – Local and metropolitan area networks – Specific requirements//Part 15.1a: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Wireless Personal Area Networks (WPAN). URL: https://standards.ieee.org/standard/802_15_1-2005.html (дата обращения 21.05.2021)

27. Martin Woolley Bluetooth Core. Specification Version 5.2. Feature Overview. 9.12.2020. С.37.

28. Pacut A., Gadomska M., Igielski A. Ant-routing vs q-routing in telecommunication networks // Proceedings of the 20-th ECMS Conference. 2006. C. 67–72.

29. Ramzi A. Haraty, Badieh Traboulsi MANET with the Q-Routing Protocol.// Proceedings of the 11th International Conference on Networks. 2012. C 187–192.

30. Uludag S, Imboden T., Akkaya K. A taxonomy and evaluation for developing 802.11-based wireless mesh network testbeds.//International Journal of Communication Systems. 2012. №25 (8). C. 965–990.