

УДК 004.057.4

## ПРОТОКОЛЫ МАРШРУТИЗАЦИИ В БЕСПРОВОДНЫХ СЕНСОРНЫХ СЕТЯХ

**Игнатов Руслан Андреевич**

магистрант

**Лосев Владимир Владимирович**

магистрант

Российский новый университет, Москва

**Аннотация.** В настоящее время всё большее распространение получают беспроводные сенсорные сети (БСС). Для обеспечения бесперебойной работы любой сети, а в частности и БСС, необходимо выбрать правильный протокол маршрутизации. Кроме того, выбор правильного протокола позволит добиться не только работоспособности сети, но и повысит её отказоустойчивость, снизит затраты на развёртывание и обслуживание, обеспечит более стабильную связь. В данной статье будет представлен обзор ряда популярных протоколов, применяемых в БСС с выделением их слабых и сильных сторон.

**Ключевые слова:** самоорганизующиеся компьютерные сети, беспроводные сенсорные сети, сети с перемещением узлов, сети с пропаданием/появлением, протоколы маршрутизации, оптимизация энергопотребления сетей.

## **Введение**

Беспроводная сенсорная сеть (БСС) представляет собой самоорганизующуюся распределённую информационную (РИС) систему, состоящую из удалённых друг от друга датчиков, собирающих информацию о своём окружении, и обеспечивающих совместную передачу данных во внешнюю информационную сеть посредством радиоканала [1]. Кроме того, сенсорная сеть может быть двунаправленной и обеспечивать управление сенсорами.

Сенсорные сети обладают хорошей масштабируемостью: подобная сеть может включать в себя тысячи узлов и покрывать, за счёт этого, большую площадь. Данные узлы обычно состоят из нескольких частей: радиопередатчик со встроенной антенной или подключенный к внешней антенне, микроконтроллер, электрическая сеть, обеспечивающая подключение к датчикам и источнику питания, обычно представляющим из себя батарею или альтернативный источник питания.

Устройства, представляющие собой узлы сенсорной сети, могут быть различных размеров: от размеров коробки до размеров песчинки. Стоимость отдельного устройства также может сильно различаться – от нескольких долларов и до сотен – в зависимости от сложности и технических характеристик.

Беспроводные сенсорные сети получили большое распространение в различных областях – от бытовых нужд и до машиностроения: они используются для сбора телеметрии с оборудования на производстве, организации охраны в музеях, обеспечивают работу «умного дома» и многое другое.

### **Виды динамических сетей**

Все динамические сенсорные сети можно разделить на три типа:

1. сеть с пропаданием и/или появлением узлов;
2. сеть с перемещением узлов;
3. смешанная сеть.

Сети с пропаданием/появлением узлов широко распространены – обычно даже в тех ситуациях, когда при штатной работе никакие узлы не покидают сеть, есть ситуации, в которых появляются новые узлы сети. Причём появление лишь одного нового узла может привести к перестройке почти всей сети, для обеспечения максимального быстродействия при передаче данных.

Сети с перемещением узлов получили значительно меньшее распространение в первую очередь по той причине, что ситуация с перемещением устройств в пространстве встречается не так часто. К примеру, узлами подобной сети могут быть беспилотные летательные аппараты (БПЛА или дроны) осуществляющие слежение за какой-либо местностью и обеспечивающие совместную передачу данных вовне. Как и в случае с пропаданием/появлением узлов, в ситуации с их перемещением даже незначительное изменение положения одного узла может потребовать перестройки топологии сети.

В случае применения подобной сети дронов в военных целях, в ситуации, если противник начнёт выводить отдельные БПЛА из строя, сеть становится смешанной – в ней одновременно пропадают одни узлы и происходит перемещение других.

## **ZigBee**

Протокол ZigBee описывается в спецификации IEEE 802.15.4 и использует для работы частоту в 2.4 ГГц. Использование данного протокола позволяет добиться низкого энергопотребления и высокой надёжности, при этом его использование дёшево. ZigBee предусматривает крип-

тографическую защиту передаваемых данных и гибкую настройку политики безопасности.

Сеть ZigBee состоит из трёх типов устройств: координатор, маршрутизатор и оконечное устройство. Координатор отвечает за запуск и управление работающей сетью, обеспечивает работу политики безопасности, осуществляет конфигурирование новых узлов сети. Маршрутизатор отвечает за передачу пакетов, решает задачи по динамической маршрутизации. При запуске сети маршрутизаторы подсоединяются к координатору или к другим маршрутизаторам, а уже к ним подключаются оконечные устройства в количестве до 32 штук к каждому. Оконечное устройство не может осуществлять маршрутизацию, но может принимать и отправлять пакеты, но не занимается их трансляцией и маршрутизацией. Оконечные устройства могут подключаться к координатору или маршрутизатору и не могут иметь дочерних устройств, но могут переходить в режим «сна» для экономии энергии.

Таблица 1

### Краткая характеристика протокола ZigBee

Автоматическая перестройка топологии	Да
Ориентация на сеть с появляющимися / исчезающими узлами	Да
Ориентация на сеть с перемещающимися узлами	Нет
Видимость других узлов	Нет
Наличие узлов-сборщиков данных	Да
Возможность эмуляции сетей unicast / multicast / broadcast	Нет

Существуют аппаратные решения, позволяющие значительно упростить применение протокола ZigBee. В качестве примера можно привести модули серии ETRX35X производства SiliconLabs. Данные модули имеют размер всего 25×19 мм, вес 3 г и обеспечивают передачу данных на скорости до 250 кб/с [2]. Они позволяют взаимодействовать как с

цифровыми, так и с аналоговыми устройствами, используя AT-команды. В том случае, если функционал стандартной прошивки модуля недостаточен, возможно приобретение и установка прошивки с расширенным функционалом EmberZNet PRO.

## LEACH

Протокол LEACH (Low-Energy Adaptive Clustering Hierarchy) является MAC-протоколом, основанным на множественном доступе с разделением времени (TDMA). Данный протокол предназначен для увеличения времени жизни датчиков за счёт снижения их энергопотребления.

В протоколе LEACH существуют две стадии: фаза настройки и фаза работы. Процесс работы по протоколу состоит из последовательности шагов, в которой каждый шаг включает в себя обе стадии. Использование подобного подхода позволяет значительно снизить потребление энергии.

В фазе настройки происходит кластеризация и выбор необходимого количества главных узлов с учётом текущего заряда батареи – главными назначаются устройства, имеющие максимальный заряд. Причём устройство не может стать главным 2 раза подряд – перед этим все остальные узлы сети также должны побывать главными. Благодаря этому обеспечивается равномерный расход энергии всех устройств [3]. После того как главные узлы будут определены, устройства, которые не были назначены главными, назначаются дочерними и отправляют запрос на подключение к ближайшему главному узлу. Экономия энергии на дочерних узлах достигается полным отключением передатчика и его включением только непосредственно на время передачи данных. Время передачи для конкретного дочернего узла определяется по расписанию, создаваемому главным узлом на основании количества подключенных к нему дочерних устройств [4].

В фазе работы уже происходит передача данных дочерними узлами главным. Члены каждого кластера могут обмениваться только со своими головными узлами, которые агрегируют полученные данные. Затем собранные данные вместе передаются на базовую станцию через головные узлы по статичным путям, и через определённое время сеть переходит обратно к фазе настройки.

Таблица 2

### Краткая характеристика протокола LEACH

Автоматическая перестройка топологии	Да
Ориентация на сеть с появляющимися / исчезающими узлами	Да
Ориентация на сеть с перемещающимися узлами	Нет
Видимость других узлов	Нет
Наличие узлов-сборщиков данных	Да
Возможность эмуляции сетей unicast / multicast / broadcast	Нет

### PEGASIS

В протоколе PEGASIS (Power-Efficient Gathering in Sensor Information Systems) узлы принимают от соседних узлов и передают им данные. В начале работы все узлы самоорганизуются в цепь с помощью жадного алгоритма, или базовая станция может выполнить расчёт топологии самостоятельно, а затем передать её узлам. В том случае, если узел сети пропадает, сеть самоорганизуется ещё раз для его обхода.

В процессе работы за передачу собранных данных на базовую станцию отвечают все узлы по очереди, что позволяет обеспечить равномерное распределение затрат энергии по всем устройствам. Для сбора пакета данных каждый узел получает данные от одного соседнего узла, объединяет их с собственными и передаёт полученный пакет дальше по цепочке. Объединение данных происходит на каждом узле, ис-

ключая последние в цепочке – каждый узел будет объединять свои данные, полученные с соседнего устройства, с собственными, создавая единый пакет и передавая его дальше [5].

Таблица 3

### Краткая характеристика протокола PEGASIS

Автоматическая перестройка топологии	Да
Ориентация на сеть с появляющимися / пропадающими узлами	Да
Ориентация на сеть с перемещающимися узлами	Нет
Видимость других узлов	Нет
Наличие узлов-сборщиков данных	Нет
Возможность эмуляции сетей unicast / multicast / broadcast	Нет

### DSDV

Протокол DSDV (Destination-Sequenced Distance Vector) – таблично-ориентированный протокол, основанный на улучшенной версии классического алгоритма маршрутизации Беллмана-Форда. Каждый узел имеет таблицу маршрутизации, содержащую все устройства, имеющиеся в сети, следующий переход и его метрику, а также порядковый номер, сгенерированный на целевом узле. Главным отличием от классического протокола является наличие векторов расстояния, применяемых для решения задачи маршрутизации. Ограничением данного подхода является невозможность существования более одного маршрута между двумя узлами [6].

Когда узел получает запрос на обновление, соседние узлы обновляют свою таблицу путём увеличения метрики на единицу. В случае изменения топологии сети, таблицы маршрутизации обновляются и ретранслируются по всей сети. Для уменьшения количества избыточных обновлений, в DSDV каждый узел отслеживает (для каждой связи) сред-

нее время между временем, когда узел получает первое обновление какого-либо маршрута следования, и тем временем, когда он получает лучшее (имеющее минимальную метрику) обновление данного маршрута [7]. При попытке обновления оно откладывается на время, равное средневзвешенному времени стабилизации, в надежде на то, что потребуется отправка только одного обновления (с лучшей метрикой), вместо нескольких.

Каждое устройство имеет две таблицы маршрутизации: одну для отправляемых пакетов и вторую для рассылки обновлённых пакетов маршрутизации. Информация о маршрутизации периодически рассылается узлами и содержит порядковый номер, адрес назначения, число переходов для достижения и маршрут следования узла-получателя.

Таблица 4

#### Краткая характеристика протокола DSDV

Автоматическая перестройка топологии	Да
Ориентация на сеть с появляющимися / исчезающими узлами	Нет
Ориентация на сеть с перемещающимися узлами	Да
Видимость других узлов	Все
Наличие узлов-сборщиков данных	Нет
Возможность эмуляции сетей unicast / multicast / broadcast	Unicast, multicast и broadcast

#### SEAD

В основу протокола SEAD (Secure Efficient Adhoc Distance vector routing) легла улучшенная версия протокола DSDV – DSDVSQ (Sequence Number). Главное отличие заключается в том, что протокол SEAD не использует средневзвешенное время стабилизации для обновления, для защиты от узлов, которые могут умышленно не использовать задержку



[8]. Кроме того, в отличие от DSDV, в том случае, когда узел обнаруживает отсутствие связи до следующего узла в маршруте, он не инкрементирует порядковый номер целевого узла в таблице, если это приведёт к тому, что его метрика будет стремиться к бесконечности.

В дополнение к вышесказанному протокол SEAD обладает следующими отличиями от протокола DSDV: защита нижней границы каждой метрики с помощью аутентификации и аутентификация отправителя получателем; использование однонаправленных хэш-цепочек. Однонаправленная хэш-цепочка обеспечивает проверку нижней границы метрики во время обновления, на практике это означает, что другой узел при обновлении может только увеличить метрику, но не уменьшить её.

Таблица 5

### Краткая характеристика протокола SEAD

Автоматическая перестройка топологии	Да
Ориентация на сеть с появляющимися / исчезающими узлами	Нет
Ориентация на сеть с перемещающимися узлами	Да
Видимость других узлов	Все
Наличие узлов-сборщиков данных	Нет
Возможность эмуляции сетей unicast / multicast / broadcast	Unicast, multicast и broadcast

### GeRaF

В протоколе GeRaF (Geographic Random Forwarding), с каждой передачей пакета, узел стремится минимизировать расстояние, которое необходимо преодолеть пакету для достижения адресата [9]. Другими словами, при последующей передаче пакета, Евклидово расстояние будет сведено к минимуму. Процедура поиска и нахождения оптимального пути передачи требует передачи определённого числа запросов, кото-

рые, в свою очередь, требуют передачи определённого числа сообщений.

Каждый узел работает в соответствии со своим расписанием активности, определяющим время его работы и нахождения в режиме «сна». Расписание устроено таким образом, что время работы узлов асинхронно. Для выполнения маршрутизации протокол делит область покрытия на регионы, каждый из которых ограничен одной или двумя дугами с центром в главном узле. GeRaF исходит из того, что каждому узлу известна, как собственная позиция, так и позиция главного узла. Всю остальную необходимую географическую информацию они получают, используя сигнальные сообщения. Связь инициируется передатчиком с помощью запроса на передачу (RTS), содержащего в себе индекс первого региона. RTS-сообщение «глушит» все соседние узлы, которые могут негативно повлиять на продвижение сигнала. Затем передатчик отправляет запрос к активным узлам в том регионе, который является ближайшим к центральному узлу. Затем между этими узлами начинается «борьба» за выбор пути передачи. «Борьба» может развиваться по следующим сценариям:

- Если в первом регионе активен только один узел, тогда он отвечает на RTS разрешением отправки (CTS). Этот узел выбирается для передачи и отправляет пакет данных. После корректного приёма пакета данных, ответ узла содержит номер подтверждения (ACK), после этого узел имеет представление о канале дальнейшей передачи пакета;
- Если в первом регионе активно более одного узла, тогда отправляется пакет COLLISION, который запускает процесс бинарного расщепления: каждый узел, с вероятностью 50 % решает отправлять CTS или нет. Этот процесс может повторяться до тех пор, пока не будет получен единственный канал передачи пакета, либо не будут перебраны все возможные варианты передачи;

- Если в первом регионе активных узлов не обнаружено, тогда передатчик отправляет RTS, содержащий индексы последующих регионов, до тех пор, пока не будут обнаружены активные узлы (начинается «борьба» за выбор пути передачи). В том случае, если после проверки всех областей, не было обнаружено активных узлов, передача прекращается.

Таблица 6

### Краткая характеристика протокола GeRaF

Автоматическая перестройка топологии	Да
Ориентация на сеть с появляющимися / исчезающими узлами	Да
Ориентация на сеть с перемещающимися узлами	Нет
Видимость других узлов	Нет
Наличие узлов-сборщиков данных	Да
Возможность эмуляции сетей unicast / multicast / broadcast	Нет

### COUGAR

Протокол COUGAR (Constrained Anisotropic Diffusion Routing) рассматривает сеть, как огромную систему распределённых баз данных [10]. Основная идея протокола заключается в использовании декларативных запросов для абстрактной обработки запросов функций сетевого уровня, таких, как: выбор подходящего датчика, составление плана запроса и т. д. COUGAR использует агрегацию сетевых данных для достижения большей экономии энергии. Абстракция поддерживается уровнем вложенности запросов, который находится между сетевым и прикладным уровнем. Протокол представляет собой архитектуру базы данных датчиков, в которой происходит выбор ведущего узла (лидера), который занимается агрегацией и передачей данных на базовую станцию. Базовая станция, в свою очередь, отвечает за формирование плана выпол-

нения запроса (план), содержащего необходимую информацию о потоке данных и внутрисетевых вычислениях для входящего запроса, а также за последующую передачу плана выполнения запроса доступным узлам. Кроме того, в плане содержится описание того, как выбрать лидера для запроса. Подобная архитектура обеспечивает внутрисетевые вычисления, которые приводят к снижению энергозатрат при генерации большого объёма данных. COUGAR – это независимый от сетевого уровня метод запроса данных. Но у данного протокола есть и недостатки:

- Добавление нового слоя запроса на каждом из узлов может привести к увеличению потребления энергии и памяти;
- Для получения оптимальных вычислений требуется синхронизация между узлами (не все входящие данные принимаются одновременно) для последующей передачи данных лидеру;
- Ведущие узлы должны динамически поддерживаться, чтобы не допустить появления сбоев в сети.

Таблица 7

### Краткая характеристика протокола COUGAR

Автоматическая перестройка топологии	Да
Ориентация на сеть с появляющимися / пропадающими узлами	Да
Ориентация на сеть с перемещающимися узлами	Нет
Видимость других узлов	Активные узлы
Наличие узлов-сборщиков данных	Да
Возможность эмуляции сетей unicast / multicast / broadcast	Unicast и multicast

### ACQUIRE

Протокол ACQUIRE (ACtiveQUery forwarding InsensOR nEtworks) рассматривает сеть, как распределённую БД, где сложные запросы могут

быть разделены на несколько более простых.[11]. Операции в ACQUIRE можно описать следующим образом:

- Базовая станция отправляет запрос, который перенаправляется каждым узлом, получившим этот запрос;
- Каждый узел пытается частично ответить на полученный запрос, используя информацию из кэша, а затем пересылает запрос следующему элементу сети.
- Если кэш не обновляется, то узлы собирают информацию у соседних узлов для устранения этой проблемы;
- Как только будет получен ответ на первоначальный запрос, он отправляется к базовой станции по изначальному пути, либо по самому короткому маршруту.

Из этого следует, что ACQUIRE может обрабатывать сложные запросы, опрашивая  $d$ -множество активных узлов и собирая от них ответы.

Направленное распространение данных не может использоваться для сложных запросов в силу того, что это требует больших энергозатрат, т.к. такая передача данных использует отправку множества простых подзапросов, для получения ответа на первичный сложный запрос. С другой стороны, ACQUIRE обеспечивает эффективное выполнение агрегированного запроса за счёт регулирования параметра  $d$ : когда  $d$  равно размерности сети, то ACQUIRE начинает массовую рассылку запросов, но если  $d$  мало, то запрос должен обойти много большее количество узлов. Для того, чтобы выбрать узел, которому будет отправлен запрос, протокол основывается на поиске и выборе самого оптимального узла, либо адресат выбирается случайным образом. Выбор следующего узла основан на одном из двух методов распространения информации: CARD (Constrained Anisotropic Diffusion Routing), либо IDSQ (Information-Driven Sensor Querying). Также запрос может быть перенаправлен на узел, который имеет необходимые данные для дальнейшего вычисления запроса.

**Краткая характеристика протокола ACQUIRE**

Автоматическая перестройка топологии	Да
Ориентация на сеть с появляющимися / исчезающими узлами	Да
Ориентация на сеть с перемещающимися узлами	Нет
Видимость других узлов	Все
Наличие узлов-сборщиков данных	Да
Возможность эмуляции сетей unicast / multicast / broadcast	Unicast, multicast и broadcast

**Rumor**

В данном протоколе каждый запрос, сгенерированный источником, распространяется по всем узлам, связанным с данным запросом [12]. Для этого узел, который связан с запросом, вводит долгоживущий пакет, называемый агентом. Агенты распространяются по сети, поэтому удалённые узлы имеют представление о связях узлов с теми или иными событиями. Для оптимизации поведения агента приняты следующие меры: когда агент достигает узла, который связан с другим новым событием, агент перенаправляется дальше и агрегирует новое событие. Кроме того, агенты сохраняют список последних посещённых узлов, поэтому за цикливание отправки агента частично исключено.

При приёме агентов узлы могут получать обновлённую информацию о событиях в сети, которая содержится в кэше событий узла. Наличие данного кэша обеспечивает более удобную отправку запросов.

Некоторые узлы могут не иметь информации о других событиях – в этом случае запрос передаётся последовательно одному из соседних узлов, выбранных случайным образом. При доставке запроса узлу, в кэше которого хранится информация о требуемом событии, запрос пе-

ренаправляется по определённом пути. Благодаря данному алгоритму отправка большого количества запросов внутри сети, имеет минимальные энергозатраты.

Таблица 9

### Краткая характеристика протокола Rumor

Автоматическая перестройка топологии	Да
Ориентация на сеть с появляющимися / исчезающими узлами	Да
Ориентация на сеть с перемещающимися узлами	Нет
Видимость других узлов	Соседние
Наличие узлов-сборщиков данных	Нет
Возможность эмуляции сетей unicast / multicast / broadcast	Multicast и broadcast

### GAF

Протокол GAF (Geographic Adaptive Fidelity) направлен на оптимизацию производительности беспроводных сенсорных сетей путём идентификации эквивалентных узлов, по отношению к отправляемым пакетам [13]. Два узла считаются эквивалентными, когда они поддерживают один и тот же набор соседних узлов, следовательно, могут принадлежать к одним и тем же маршрутам связи. Источник и получатель не задействованы при определении эквивалентности узлов. Для того, чтобы определить являются ли два узла эквивалентными, необходимо знать их позиции, на основании которых создаётся виртуальная сетка. Её образуют ячейки, размер которых обеспечивает прямое соединение узлов данной ячейки с узлами соседних ячеек. Из этого следует, что ячейки эквивалентны.

Узлы определяют эквивалентные элементы сети за счёт периодического обмена специальными сообщениями внутри своей ячейки. При помощи информации, содержащейся в данном сообщении, узлы опреде-

ляют, кто из них будет поддерживать связь (связующие узлы), а остальные узлы отключаются. Благодаря этой процедуре сохраняется правильность маршрутизации, то есть непрерывная связь между ключевыми узлами. Однако выбранный узел периодически сменяется для равномерного распределения энергопотребления, то есть остальные узлы время от времени пробуждаются для повторного выбора связующего узла.

Таблица 10

### Краткая характеристика протокола GAF

Автоматическая перестройка топологии	Да
Ориентация на сеть с появляющимися / исчезающими узлами	Да
Ориентация на сеть с перемещающимися узлами	Да
Видимость других узлов	Соседние
Наличие узлов-сборщиков данных	Нет
Возможность эмуляции сетей unicast / multicast / broadcast	Unicast, multicast и broadcast

### SPIN

В основе работы протокола SPIN (Sensor Protocol for Information via Negotiation) лежит использование метаданных для описания передаваемой информации – непосредственно перед передачей последней, узлы обмениваются между собой метаданными. Использование этой технологии позволяет решить проблему избыточной рассылки данных и обеспечить снижение потребления электроэнергии [14]. Кроме того, в данном протоколе процедура смены топологии локализована за счёт того, что каждому узлу необходимо знать только ближайших соседей (соседей, до которых требуется только один переход).



Всего в SPIN существует три типа сообщений, применяемых при передаче: ADV – позволяет сенсору разослать мета-данные, REQ – служит для запроса конкретных данных; DATA – обеспечивает передачу данных. Процесс передачи информации узлом А узлу В проходит следующим образом:

1. Узел А отправляет узлу В мета-данные сообщением ADV;
2. Узел В запрашивает у узла А необходимые ему данные;
3. Узел А передаёт узлу В запрошенные данные;
4. Узел В повторяет пункты 1-3 для соседних с ним узлов.

Но у данного протокола есть и недостаток – он не гарантирует доставку данных. Передача данных невозможна, если узел, заинтересованный в них, отделяют от источника этих данных другие узлы, которые сами не заинтересованы в получении этих данных.

Таблица 11

### Краткая характеристика протокола SPIN

Автоматическая перестройка топологии	Да
Ориентация на сеть с появляющимися / исчезающими узлами	Да
Ориентация на сеть с перемещающимися узлами	Да
Видимость других узлов	Нет
Наличие узлов-сборщиков данных	Нет
Возможность эмуляции сетей unicast / multicast / broadcast	Нет

### Выводы

В результате анализа различных алгоритмов были сделаны следующие выводы: несмотря на кажущееся сходство алгоритмов, их можно разделить по областям применения. Были выделены следующие области применения:

- протоколы, ориентированные на организацию сетей с перемещающимися узлами (например, сеть БПЛА, осуществляющих сканирование местности);
- протоколы, ориентированные на организацию сетей с пропадающими/появляющимися узлами (например, «умные» минные поля; сеть датчиков, осуществляющих слежение за местностью);
- универсальные протоколы.

Рассмотренные сети были разделены по данным областям применения следующим образом:

- Ориентированные на перемещение:
  - DSDV
  - SEAD
- Ориентированные на пропадание/появление:
  - ZigBee
  - LEACH
  - PEGASIS
  - GeRaF
  - COUGAR
  - ACQUIRE
  - Rumor
- Универсальные:
  - SPIN
  - GAF

Используемые методы организации сетей различны: одни используют общие базы данных узлов сети, иные используют непрерывную отправку подзапросов и обмен сигнальными сообщениями для построения маршрута; в некоторых присутствуют узлы-сборщики информации. Возможности и различия алгоритмов позволят выбрать необходимый алгоритм, который будет максимально приближённо отвечать по-

ставленным требованиям. В частности, немаловажной характеристикой протокола является возможность создания uni-, multi- и broadcast сетей поверх него. Требования и обоснование выбора конкретного протокола маршрутизации предполагается определить в дальнейшем исследовании.

### **Список использованных источников**

1. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. СПб: Питер, 2010.
2. SiliconLabs. ETRX35x ProductManual. 2015 [Электронный ресурс]. Режим доступа: <https://www.silabs.com/documents/public/data-sheets/TG-PM-0516-ETRX35x.pdf>
3. Reshma I. Leach Protocol in Wireless Sensor Network: A Survey. Tandel, Shri S'adVidya Mandal Institute of Technology, 2016. P. 1894-1896.
4. Heinzelman W.B., Chandrakasan A.P., Balakrishnan H. An Application-Specific Protocol Architecture for Wireless Microsensor Networks. IEEE. Trans. Wirel. Commun, 2002. P. 660-670.
5. Stephanie Lindsey, Cauligi S. Raghavendra. PEGASIS: Power-Efficient Gathering in Sensor Information Systems. L.A.: The Aerospace Corporation, 2002. P. 3–5.
6. Guoyou He. Destination-Sequenced Distance Vector (DSDV) Protocol. Helsinki University of Technology, 2002.
7. Tanenbaum A.S. Computer Networks. Prentice Hall, Inc. 1996.
8. Yih-Chun Hu, Johnson D.B., Perrig A. SEAD: secure efficient distance vector routing for mobile wireless ad hoc networks. Elsevier. 2003. P. 180-185.
9. Zorzi M., Rao R.R. Geographic random forwarding (GeRaF) for ad hoc and sensor networks: energy and latency performance. IEEE Transactions on Mobile Computing. 2003. P. 349-365.

10. Yao Y., Gehrke J. The Cougar Approach to In-Network Query Processing in Sensor Networks. SIGMOD Rec. 2002.
11. Sadagopan N., Krishnamachari B., Helmy A. The ACQUIRE mechanism for efficient querying in sensor networks. Anchorage, AK, IEEE, 2003. P. 149-155.
12. Braginsky D., Estrin D. Rumor Routing Algorithm for Sensor Networks. Proceedings of the First ACM International Workshop on Wireless Sensor Networks and Applications (WSNA). Atlanta, GA, 2002. P. 22-31.
13. Xu Y., Heidemann J., Estrin D. Geography-informed energy conservation for Ad Hoc routing. N.Y., CRC Press, 2001. P. 70-84.
14. Vidhi S. Patel, Chandresh R. Parekh. Survey on Sensor Protocol for Information via Negotiation (SPIN) protocol. Gandhinagar, Government Engineering College, 2014. P. 1453-1455.