

SAINT-PETERSBURG STATE UNIVERSITY

Mathematics & Mechanics Faculty

Department of Software Engineering

Kavokin Aleksandr

Message transmission energy minimisation in mesh networks

Master's Thesis

Admitted for defence.

Head of the chair:

Professor Andrey Terekhov

Scientific supervisor:

Professor Andrey Terekhov

Reviewer:

Popov Kirill

Saint-Petersburg

2016

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Математико-механический факультет

Кафедра системного программирования

Кавокин Александр Сергеевич

Передача сообщений в mesh-сети с минимальными затратами энергии

Магистерская выпускная работа

Допущен к защите

Зав.кафедрой:

д.ф.-м.н., профессор А.Н.Терехов

Научный руководитель:

д.ф.-м.н., профессор А.Н.Терехов

Рецензент:

Попов К.В.

Санкт-Петербург

2016

Оглавление

Введение.....	4
Обзор предметной области.....	5
Mesh-сеть.....	5
Протоколы маршрутизации mesh-сетей.....	7
Протокол AODV.....	7
Гибридный протокол маршрутизации mesh-сети (Hybrid Wireless Mesh Protocol).....	9
Динамическая маршрутизация от источника (Dynamic Source Routing).....	9
Постановка задачи.....	10
Реализация.....	11
Инструменты.....	11
Алгоритм.....	11
Сокращение количества служебной информации.....	12
Перебалансировка таблиц маршрутизации.....	13
Объединение сообщений.....	14
Результаты.....	14
Сравнение с AODV.....	14
Выступления.....	17
Публикации.....	17
Заключение.....	17
Литература.....	18

Введение

В настоящее время существует большое количество задач, связанных с организацией различного рода сетей для сбора и передачи информации. Ни для кого не секрет, что наиболее известной является сеть Интернет, созданная в конце 60-ых годов XX века, которая в своем продолжении объединила крупные сети. Но доступ в Интернет является централизованным, поскольку весь трафик проходит через точки доступа у провайдеров. В случае выхода из строя или блокировки какой-то из таких точек, возможность выхода в сеть будет прервана для всех зависимых узлов. Для некоторых отраслей, например в военной, это неприемлемо. В таких случаях требуется иметь возможность создать избыточную сеть, в которой каждый канал связи многократно дублируется.

Если узлы в сети автономны и не имеют доступа к постоянному источнику электроэнергии, то требуется максимально продлить жизнеспособность системы, чтобы каждый узел имел гарантированную возможность доставить сообщение до любого другого. Для этого необходимо минимизировать затраты энергии при взаимодействии и передаче данных между узлами в сети, а также динамически изменять нагрузку путем балансировки всей системы, чтобы наиболее востребованные узлы существовали максимально долго.

Например, рассмотрим сеть из разбросанных на ограниченной площади большого количества устройств, способных собирать информацию об окружающей среде. Данные устройства могут обмениваться данными путем передачи сообщений в радиусе действия своих передатчиков. Эту сеть необходимо настроить, чтобы каждое устройство могло передать по цепочке другим узлам сети собранную информацию. При администрировании этой сети требуется учитывать, что все узлы автономны и не имеют доступа к постоянному источнику электроэнергии, у каждого существует ограниченный заряд батареи. Для решения подобного рода задач используются набирающие популярность и широкое распространение mesh-сети.

Обзор предметной области

Mesh-сеть

Mesh-сеть или ячеистая сеть - это сетевая топология, состоящая из беспроводных устройств, каждое из которых соединено со многими другими. Многие соединения в сети избыточны, благодаря чему достигается высокая отказоустойчивость всей системы. Устройства обмениваются информацией по радиоканалу, способны объединяться в кластеры, переопределять свою роль в сети, что позволяет говорить об “интеллектуальности” сети. Пример mesh-сети приведен на Рисунке 1. Также данные можно перенаправить в случае перегрузки или при выходе из строя некоторых точек доступа. Территория покрытия ячеистой сети разделяется на кластерные зоны. В каждом кластере размещается несколько, чаще всего от 8 до 16 точек доступа. Среди них выбирается одна узловая, которая связывается с остальными узловыми точками других кластеров. За счет этого mesh-сети легко поддаются масштабированию. Все это вкупе с отсутствием требований к дорогостоящей инфраструктуре делает mesh-сети очень удобной и экономной системой в эксплуатации.

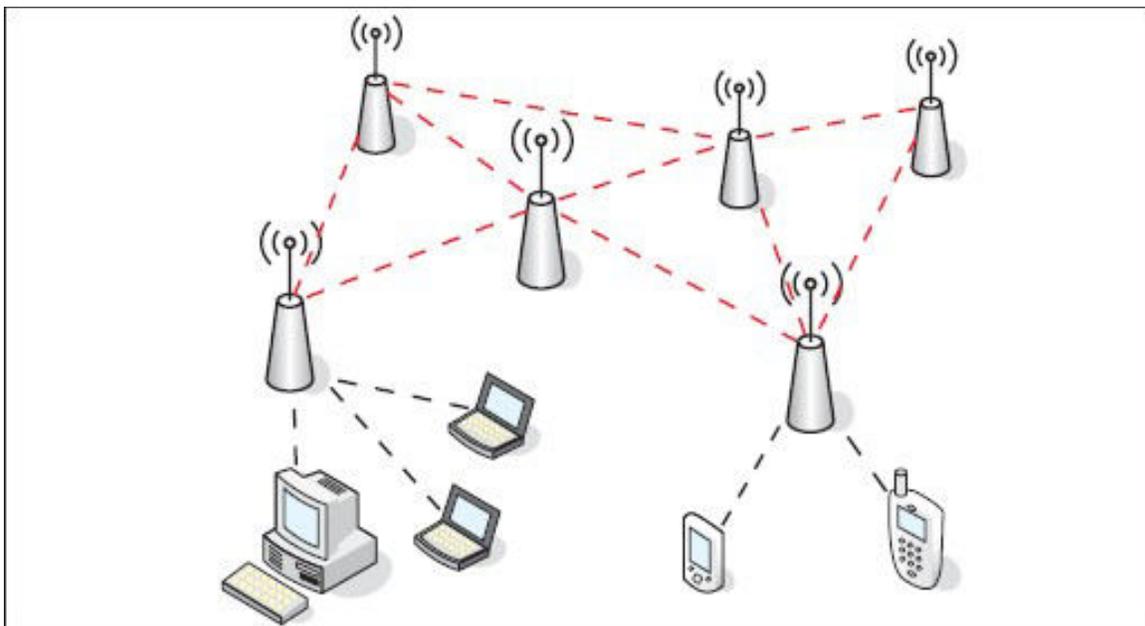


Рисунок 1. Пример mesh-сети.

Таким образом, особенности mesh-сети заключаются в следующем:

- «Интеллектуальность» сети.

Каждое устройство при подключении в сеть автоматически получает всю необходимую информацию о других узлах сети и своей роли в ней. Из этого следует, что сеть легко масштабируема.

- Самовосстановление сети.

В случае выхода из строя какого-то из узлов, сеть способна самостоятельно перенаправить данные по другому пути.

- Быстрое и недорогое развертывание.

Все узлы в сети являются беспроводными, благодаря чему не требуется дорогостоящая инфраструктура для прокладки кабелей. Самовосстановление системы обеспечивает экономию к эксплуатации.

Ячеистые сети применяются для широкого спектра задач.

- В военной отрасли для налаживания связи между компьютерами при полевых операциях.
- При налаживании связи в крупных корпоративных средах провайдером для преодоления проблемы последней мили.
- В спутниковой связи существует «созвездие» спутников Иридиум, которое функционирует как mesh-сеть. Звонок между двумя спутниковыми телефонами передается внутри «созвездия» без необходимости взаимодействия со станциями связи на Земле.
- При организации сети в бизнесе. Например, для банкоматов или камер видеонаблюдения. Если на улице существует mesh-сеть, то при добавлении нового устройства его требуется всего лишь подключить в сеть, где он сам сможет не только получать доступ ко всем другим, но и улучшать связь в

самой сети, выступая в роли ретранслятора.

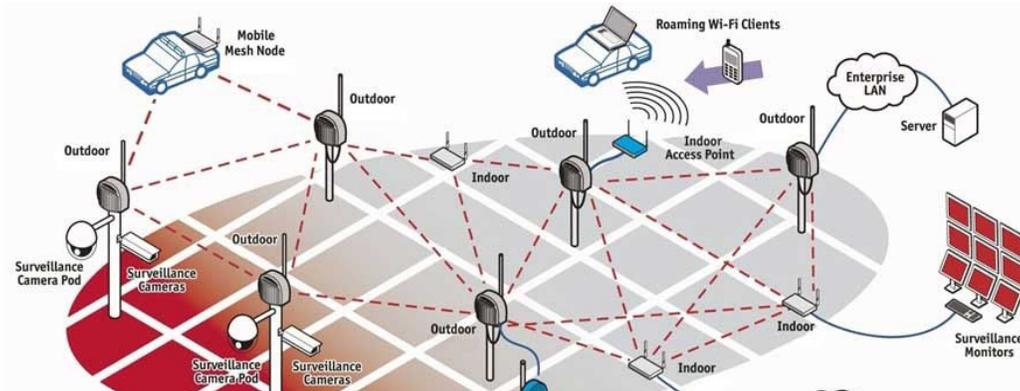


Рисунок 2. Пример использования mesh-сети.

Протоколы маршрутизации mesh-сетей

Существует несколько протоколов по организации ячеистых сетей, из которых наиболее часто используемыми являются следующие:

Протокол AODV.

Протокол AODV является протоколом, используемым для создания топологии сети реактивным способом, что означает его свойство устанавливать маршрут до нужного хоста по требованию.

Если узлу сети А требуется передать данные узлу сети В, а соответствующий маршрут неизвестен, то узел А посылает широковещательный запрос PREQ для поиска маршрута до узла В. Этот запрос передается соседям узла А, те пересылают его своим соседям и т.д. Когда к какому-то узлу приходит запрос PREQ, он записывает туда данные о себе и пересылает дальше, соответственно, каждый узел, получивший запрос знает маршрут, по которому запрос прошел от узла-инициатора до него. Через какое-то время этот запрос дойдет либо до узла В, либо до узла, который знает маршрут от себя до узла В. В обоих случаях узлу А отправляется ответ PREP, который идет уже не широковещательно, а целенаправленно по найденному маршруту. В результате выстраивается двунаправленный маршрут. Как только узел А получает ответ PREP, он сразу же начинает передачу данных по этому маршруту, не дожидаясь ответа по другим каналам. Если через некоторое время окажется, что существует более короткий

маршрут, узел автоматически продолжит отправлять пакеты по более короткому пути.

Поля сообщения RREQ выглядят следующим образом

- Тип (8 бит)— «1»
- Флаг «J»— устанавливается, если RREQ передаётся широковещательно
- Флаг «R»— устанавливается при восстановлении маршрута
- Флаг «G»— устанавливается в том случае, если необходимо установить не только маршрут до узла-получателя, но и обратный маршрут от узла-получателя до узла-инициатора.
- Флаг «D»— устанавливается, если необходимо доставить сообщение RREQ непосредственно до узла-получателя.
- Флаг «U»— устанавливается в том случае, если порядковый номер узла-получателя неизвестен
- 11 бит зарезервировано
- Hop Count (Количество переходов) — 8 бит
- RREQ ID (идентификационный номер сообщения RREQ)
- IP-адрес узла-получателя (Destination IP Address)
- Порядковый номер узла-получателя (Destination Sequence Number)
- IP-адрес узла-инициатора (Originator IP Address)
- Порядковый номер узла-инициатора (Originator Sequence Number)

тип 8 бит	флаг J	флаг R	флаг G	флаг D	флаг U	резерв 11 бит	Hop Count 8 бит	PREQ ID	IP узла получателя	номер узла получателя	IP узла инициатора	номер узла инициатора
--------------	-----------	-----------	-----------	-----------	-----------	------------------	--------------------	------------	-----------------------	--------------------------	-----------------------	--------------------------

Рисунок 3. Формат PREQ.

После того, как маршрут создан, отправляются пакеты без маршрутной информации. Агенты А и В хранят информацию о связывающем их маршруте. Если эта связь долгое время не используется, то все данные о ней стираются.

Гибридный протокол маршрутизации mesh-сети (Hybrid Wireless Mesh Protocol)

Данный протокол основан на протоколе AODV. Согласно этому протоколу узлы сети могут искать пути двумя способами, в том числе и сочетая их:

- Реактивный способ. Подразумевает поиск пути непосредственно перед отправкой информации аналогично протоколу AODV.
- Проактивный способ. Означает, что поиск пути происходит регулярно, вне зависимости от требования передать данные в текущий момент времени. Процедуру инициирует корневой узел, в результате на сети строится граф путей с вершиной в корневом узле. Корневой узел выбирается в самом начале, то есть сеть является централизованной.

Динамическая маршрутизация от источника (Dynamic Source Routing)

DSR – протокол маршрутизации с топологией mesh. Схож с AODV тем, что формирует маршрут по требованию через широковещательный запрос. Однако существенное различие между этими двумя протоколами в том, что DSR постоянно обновляет маршрутизацию. То есть информация о маршруте хранится в каждом сообщении.

Все описанные выше протоколы способны настроить ячеистую сеть, но не учитывают того, что узлы могут быть автономны и с ограниченным зарядом

батареи.

Постановка задачи

Рассмотрим множество неподвижных приемопередатчиков для сбора информации на крупной территории. Например, передатчики разбрасываются в лесу и измеряют температуру окружающей среды с целью раннего оповещения о лесных пожарах. Радиус связи каждого передатчика не охватывает всю территорию. Каждый агент имеет возможность связываться всего с несколькими соседними агентами. Требуется предложить алгоритм, позволяющий объединить все эти передатчики (далее агенты) в ячеистую сеть, в которой агенты смогут связываться друг с другом и передавать информацию. Полученная сеть должна удовлетворять следующим свойствам:

- Каждый агент имеет ограниченный заряд батареи.
- Система состоит из нескольких сотен агентов, равномерно распределенных по большой площади. Каждый агент имеет ограниченный радиус связи, в который попадает несколько других агентов. При консультации с экспертами было установлено, что для охвата заданной территории требуется порядка 300 агентов.
- Рядом с сетью существует агрегатор с бесперебойным источником питания, который может запрашивать информацию, хранящуюся на конкретных узлах mesh-сети. Дальность действия приёмопередатчика агрегатора такова, что может взаимодействовать лишь с некоторыми наиболее близкими агентами сети.
- Сеть должна быть устойчивой к потере агентов и уметь восстанавливать маршруты в таком случае.
- Система должна гарантированно функционировать в среднем дольше, чем при использовании существующих алгоритмов. Система будет считаться вышедшей из строя, если какой-то не вышедший из строя агент потеряет связь с агрегатором и не сможет ее восстановить.

Также требуется смоделировать полученную сеть на одном из языков программирования и сравнить сроки максимальные сроки гарантированного функционирования системы с сетью, постоянной по протоколу AODV. Система будет считаться вышедшей из строя, если какой-то из работающих узлов безвозвратно потеряет возможность установить связь с агрегатором.

Реализация

Инструменты

В реальных условиях система будет реализована непосредственно на устройствах. В рамках нашей задачи требуется только спроектировать систему и протестировать ее на одном из языков программирования. Для тестирования алгоритма был выбран язык C# под платформой .NET Framework 4.5. В ходе работы использовалась IDE Visual Studio 2013.

Алгоритм

Было принято решение создать новый протокол на базе уже существующих и описанных выше. В качестве базового протокола был выбран AODV.

Поскольку в нашей системе предполагается большое количество сообщений, заново устанавливая таблицы маршрутизации перед каждой отправкой данных будет накладно. Поэтому был выбран проактивный метод определения топологии ячеистой сети, поскольку он более эффективный с точки зрения энергозатрат в рамках текущей задачи.

Сообщения передаются по радиоканалу. Принцип передачи радиосигнала следующий:

Информация кодируется модуляцией несущей частоты. Передатчик излучает электромагнитные волны в некотором промежутке частот. Изменениями частот в данном промежутке кодируется сигнал. Передатчик тратит энергию на генерацию электромагнитных волн и обработку сообщений, в связи с чем можем считать, что зависимость между количеством передаваемой информации и энергозатратами линейная. Помимо генерации волн передатчик тратит энергию на включение

излучения, но в первом приближении этими затратами можно пренебречь.

Сокращение количества служебной информации

При включении каждый агент рассылает широковещательный запрос на поиск агрегатора. Запрос состоит из сообщения, в котором содержатся:

- Id агента-инициатора;
- Список уже посещенных агентов.

Когда агент получает такой запрос, он проверяет, содержится ли его id в списке посещенных агентов. Если нет, то добавляет его туда и рассылает обновленный запрос. В противном случае никаких действий не происходит. Когда такой запрос доходит до агрегатора, он отправляет ответный запрос по пути, хранящемуся в списке посещенных агентов. Когда ответный запрос достигает агента-инициатора, тот начинает передачу данных по полученному пути. Если же в какой-то момент до агрегатора доходит запрос с более коротким путем, то отправляется еще один ответный запрос по обновленному пути, а агент-инициатор, получив его, незамедлительно начинает использовать более короткий путь. Таким образом, видно, что основа алгоритма схожа с алгоритмом AODV.

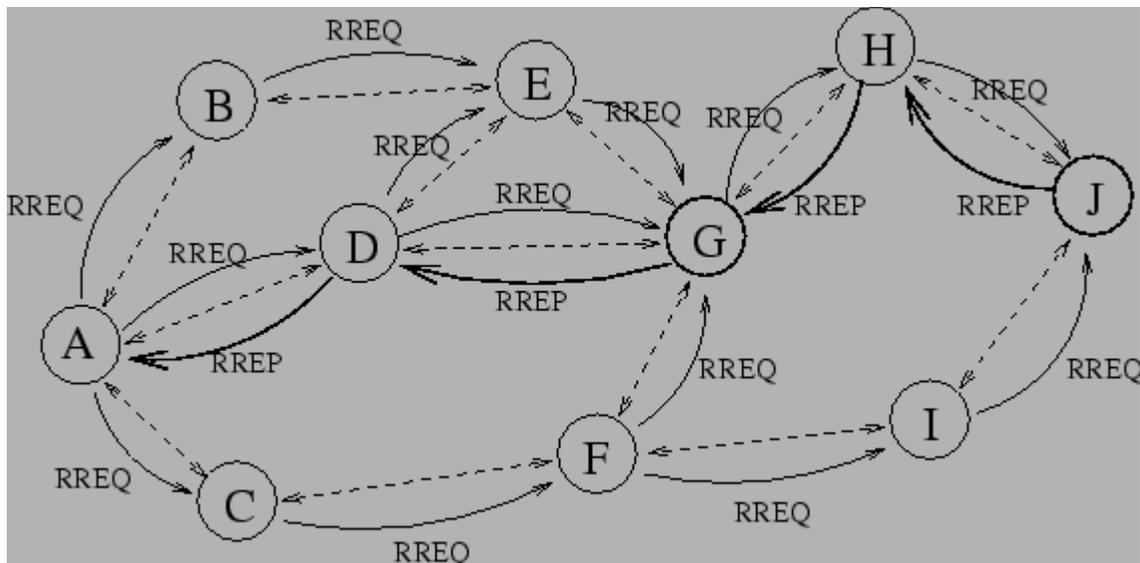


Рисунок 4. Установление связи между агентами.

Как было сказано в постановке задачи, в предполагаемом использовании запланировано примерно 300 агентов. Чтобы минимизировать энергозатраты и максимально продлить жизнь системе, мы можем видоизменить алгоритм конкретно под данную задачу.

Объем передаваемой информации планируется не очень большим, поскольку агенты собирают информацию об окружающей среде, например температуру и влажность. Таким образом получается, что объем полезной информации сравним с объемом служебной информации. Первым шагом очевидно предположить, что стоит уменьшить объем служебной информации сообщения. Для того, чтобы хранить Id каждого агента достаточно всего 9 бит (т.к таким количеством бит мы можем закодировать до 512 чисел).

Само сообщение имеет следующий вид:

- Id агента отправителя (9 бит).
- Id агента получателя (9 бит).
- Список агентов, через которые сообщение должно пройти (по 9 бит на каждого).

Итого получается, что каждое сообщение имеет вес $(18+9N)$, где N – количество агентов, через которое проходит сообщение.

Перевыравнивание таблиц маршрутизации

Также требуется, чтобы вся система функционировала максимально долго. Это предполагает, что каждый агент должен существовать максимально долго.

Возможна ситуация, когда у разных агентов в начальный момент времени разные уровни заряда батарей. Также в процессе работы неизбежна несбалансированная нагрузка на агентов. Очевидно, что чем ближе агент находится к штабу, тем больше сообщений будет проходить через него. В случае использования стандартного алгоритма AODV самые загруженные агенты выйдут из строя, что может нарушить целостность системы. Чтобы этого избежать предлагается

использовать пороговое значение для уровня батареи агентов, при котором сеть будет перестраивать существующие таблицы маршрутизации, чтобы по минимуму нагружать устройства с севшими аккумуляторами. Агрегатор хранит всю информацию о маршрутах в сети, в то время как агенты знают только об оптимальном пути. Каждый список маршрутов до конкретного узла отсортирован по количеству промежуточных звеньев. Если уровень заряда батареи какого-то из агентов A достиг критического минимума, агрегатор для каждого агента, использующего данный узел A , выбирает следующий по длине маршрут, не содержащий A и шлет указание соответствующим агентам использовать эти маршруты.

Объединение сообщений

Помимо этого, логично, что агенты, расположенные ближе всего к агрегатору, будут загружены больше, поскольку через них идут каналы связи от более удаленных агентов. Агент может собирать сообщения, у которых одинаковый дальнейший путь до агрегатора, объединять их в одно большое сообщение и отправлять дальше. Это также позволит уменьшить суммарный объем служебной информации. Время, которое агент будет ждать для такого сбора сообщений находится на основании периодичности поступления данных ранее.

Результаты

В ходе исследования был предложен алгоритм, позволяющий настроить mesh-сеть, состоящую из неподвижных автономных узлов.

Сравнение с AODV

В качестве тестирования были сгенерированы сети с равномерно распределенными агентами по квадратным областям. В ходе экспериментов изменялись следующие параметры:

- количество агентов;
- радиус связи агентов;
- размеры исследуемой области.

В рамках нашей работы было проведено 2 тестирования.

В рамках первого тестирования проводилось сравнение работы сети, построенной по протоколу AODV с работой сети, построенной по описанному выше алгоритму, но без объединения сообщений. Сделано это было поскольку неизвестно, как часто в реальной системе будет возникать ситуация, благоприятная для объединения сообщений. Во-первых, сообщения могут приходить очень редко. Во-вторых, в случае сильных радиопомех лучше передавать данные маленькими частями, чтобы при потере какого-то пакета было проще его восстановить.

Для каждого набора параметров создавалось 5 сетей из агентов. На каждом полученном множестве были проведены испытания протокола AODV и предложенного протокола с балансировкой нагрузки. Поскольку неизвестно, как часто у агентов в реальной системе будут возникать события, при которых требуется отправить данные по сети агрегатору, было принято решение вызывать такие события с одинаковой частотой у каждого агента. После этого агент отправлял одинаковый объем информации, символизирующий информацию об окружающей среде, агрегатору по наиболее оптимальному в рамках текущего алгоритма пути. Как только какой-то из узлов безвозвратно терял возможность доставить данные до агрегатора, то есть целостность системы была нарушена и не подлежала восстановлению, передача сообщений прекращалась. В качестве параметра сравнения было выбрано среднее количество сообщений, доставленное от каждого агента до агрегатора. Учитывались все сообщения, в том числе и служебные для организации таблиц маршрутизации. По полученным результатам бралось среднее арифметическое. Итоговые результаты тестирования записаны в таблице 1.

Количество агентов	Радиус связи	Размер поля	Среднее количество переданных сообщений (AODV)	Среднее количество переданных сообщений (балансировка)
30	15	50	2300	4600
30	50	200	2200	3900
300	15	200	1500	2100
300	50	200	2700	1300
300	100	200	2600	5400

Таблица 1. Результаты тестирования без объединения сообщений.

Как видно из таблицы, предложенный протокол способен поддерживать целостность сети в некоторых случаях даже в 2 раза дольше чем стандартный протокол AODV.

Второе тестирование проводилось с использованием объединения сообщений. Поскольку неизвестно, как часто сообщения будут приходить к агентам в реальной системе, каждый агент с определенной периодичностью накапливал до десяти сообщений, которые требовалось отправить по одному маршруту. Затем формировал из них единое сообщение и отправлял дальше. Во всем остальном алгоритм был аналогичен предыдущему тестированию. Результаты тестирования приведены в Таблице 2.

Количество агентов	Радиус связи	Размер поля	Среднее количество переданных сообщений (AODV)	Среднее количество переданных сообщений (балансировка)
30	15	50	2300	5000
30	50	200	2200	4100
300	15	200	1500	2100
300	50	200	2700	3500
300	100	200	2600	4800

Таблица 2. Результаты тестирования с объединением сообщений.

Как видно из таблицы, в рамках второго тестирования количество передаваемых сообщений вырастает еще больше. Но следует учитывать, что в данном тестировании считалось, что сообщения генерируются с большой частотой. Если агенты будут реже обмениваться сообщениями, то смысл в объединении сообщений пропадает, поскольку при накоплении сообщений возникнет большая задержка между временем отправки сообщения агентом и временем получения этого сообщения агрегатором.

Выступления

Результаты работы были представлены на конференциях SEIM-2016 и СПИСОК-2016.

Публикации

Результаты работы будут опубликованы в сборнике конференции СПИСОК-2016.

Заключение

В результате данной работы предложен алгоритм, позволяющий увеличить время автономной работы mesh-сети в рамках поставленной задачи. В дальнейшем предполагается добавить учет объема сообщений при балансировке и возможность разбиения сообщения на несколько пакетов по аналогии с TCP/IP. Также планируется усовершенствовать принцип балансировки.

Еще одним интересным направлением для дальнейшей работы является изучение

ситуации, когда вокруг поля с агентами расположено несколько агрегаторов. В таком случае сеть становится децентрализованной, в отличии от предложенного алгоритма.

Литература

1. Казаков М.Ф. ПОСТРОЕНИЕ САМООРГАНИЗУЮЩЕЙСЯ СЕТИ МОБИЛЬНЫХ УСТРОЙСТВ. <https://cyberleninka.ru/article/n/protokoly-marshrutizatsii-v-besprovodnyh-setyah>
2. Карманов М.Л. ПРОТОКОЛ МАРШРУТИЗАЦИИ ДЛЯ АД-НОС СЕТЕЙ. <http://dspace.susu.ac.ru/bitstream/handle/0001.74/768/10.pdf?sequence=1>
3. Осипов И.Е. Mesh-сети: технологии, приложения, оборудование. http://www.tsonline.ru/articles2/fix-op/mesh_seti_tehn_prilozh_oborud
4. RFC 3561. Ad hoc On-Demand Distance Vector (AODV) Routing. <https://tools.ietf.org/html/rfc3561>
5. AODV <https://ru.wikipedia.org/wiki/AODV>