

Санкт-Петербургский государственный университет

Программная инженерия

Дулетов Дмитрий Евгеньевич

Разработка системы передачи сообщений в децентрализованной сети подвижных узлов

Отчёт о прохождении производственной практики

Научный руководитель:
д.ф.-м.н., профессор кафедры СП О.Н. Граничин

Санкт-Петербург
2024

Оглавление

1. Введение	3
2. Постановка задачи	5
3. Обзор	6
3.1. Mesh-сеть	6
3.2. Ad-hoc-сеть	7
3.3. Широковещательные сети	8
3.4. ESP-NOW	8
4. Система передачи сообщений	10
4.1. Алгоритм работы узла	10
5. Детали реализации	12
5.1. ESP32	12
5.2. Интеграция с MartOS	13
6. Заключение	16
Список литературы	17

1. Введение

В современном мире развитие технологий и искусственного интеллекта позволяет говорить о возможности создания децентрализованной сети роботов, способных выполнять различные задачи и функции, от автоматизации производства до оказания помощи в решении сложных задач [2]. Такое разнообразие возможностей делает данное направление развития технологий очень перспективным в ближайшем будущем.

Децентрализованная сеть роботов представляет собой группу роботов, взаимодействующих между собой и с окружающей средой без централизованного управления. Каждый робот в такой сети обладает определенной степенью автономности и может принимать решения на основе информации, полученной от других роботов и датчиков. Благодаря этому, такие сети обладают потенциалом для создания эффективных и гибких систем, способных адаптироваться к изменяющимся условиям. [1]

Концепция децентрализованных систем широко применяется в организации роя роботов. Такие сети состоят из множества подвижных узлов, которые взаимодействуют между собой без центрального сервера, что обеспечивает их высокую масштабируемость и устойчивость к сбоям. Подвижные узлы могут быть представлены как квадрокоптеры, так и машинами передвигающимися по земле.

В контексте децентрализованных сетей разработка системы передачи сообщений является актуальной и важной задачей. Узлам необходимо оперативно делиться информацией, чтобы регулировать работу сети. В связи с отсутствием центрального сервера, каждый узел обеспечивает передачу информации и от качества доставки сообщений зависит скорость работы всей сети. [3]

Однако, в связи с большой автономностью узлов, они редко обладают мощными передатчиками и большими вычислительными мощностями. В связи с этим встаёт вопрос о создании системы эффективной передачи сообщений большому количеству адресатов при ограничен-

ных ресурсах.

2. Постановка задачи

Целью работы является реализация алгоритма обмена сообщениями между роботами в реальном времени при помощи широкополосных сигналов.

Для достижения поставленной цели были поставлены следующие задачи.

1. Выполнить обзор решений в области передачи сообщений в крупных сетях.
2. Реализовать систему передачи сообщений.
3. Интегрировать систему передачи сообщений в действующие проекты НОЦ «Математическая робототехника и искусственный интеллект».
4. Протестировать систему и провести эксперименты.

3. Обзор

Передача сообщений в децентрализованной сети подвижных узлов является очень важной задачей. Поскольку в сети отсутствует центральный узел, не существует единственного сервера, через который можно было бы вести обмен сообщениями. Также система предполагает движение узлов в пространстве, в связи с чем доступность окружающих абонентов может меняться. Последним фактором, ограничивающим систему является вычислительная мощность агентов. Размер сети может быть достаточно большим, вплоть до сотен узлов, и скорость обработки информации не позволяет отправлять сообщение отдельно каждому узлу внутри нашей сети.

Исходя из этих ограничений рассмотрим несколько подходов к организации системы обмена сообщениями:

3.1. Mesh-сеть

Mesh-сеть или ячеистая сеть — это топология локальной сети, в которой узлы инфраструктуры (для децентрализованных сетей все узлы это клиенты) подключаются напрямую, динамически и неиерархически к как можно большему количеству других узлов и взаимодействуют друг с другом для эффективной маршрутизации данных к и от клиентов.

Отсутствие зависимости от одного узла позволяет каждому узлу участвовать в передаче информации. Ячеистые сети динамически самоорганизуются и самонастраиваются, что позволяет снизить затраты на установку. Возможность самостоятельной настройки обеспечивает динамическое распределение рабочих нагрузок, особенно в случае сбоя нескольких узлов.

В отличие от традиционных топологий локальной сети, в которых клиенты напрямую связаны лишь с небольшим подмножеством других клиентов, а связи между этими соседями по инфраструктуре являются иерархическими., mesh-сеть позволяет динамически подключать к клиенту больше других участников сети напрямую. При этом поставщики

ячеистых сетевых устройств еще не все пришли к согласию по общим стандартам, а совместимость между устройствами разных производителей еще не гарантирована [10].

Поскольку к одному клиенту может быть подключено множество адресатов в peer-to-peer соединении, ячеистые сети сильно нагружают устройство отправки сообщений, что вызывает задержки при ее использовании [6].

3.2. Ad-hoc-сеть

Ad-hoc-сеть или Беспроводная одноранговая сеть — это децентрализованный тип беспроводной сети. Данный вид сетей не зависит от уже существующей инфраструктуры, такой как маршрутизаторы или точки беспроводного доступа. Вместо этого каждый узел участвует в маршрутизации, пересылая данные другим узлам. Определение того, какие узлы пересылают данные, производится динамически на основе сетевого подключения и используемого алгоритма маршрутизации.[2]

В таких беспроводных сетях отсутствуют сложности настройки и администрирования инфраструктуры, что позволяет устройствам создавать сети и присоединяться к ним «на лету».[3]

Каждое устройство в ad-hoc-сети может свободно перемещаться независимо в любом направлении и поэтому будет часто менять свои связи с другими устройствами. Каждый из них должен пересылать трафик, не связанный с его собственным использованием, и, следовательно, быть маршрутизатором. Основной задачей при создании такой является оснащение каждого устройства для постоянного хранения информации, необходимой для правильной маршрутизации трафика. Это становится сложнее по мере увеличения масштаба в следствие следующих причин [4].

- Желание маршрутизировать пакеты к каждому другому узлу или через него.
- Процент служебного трафика, необходимого для поддержания статуса маршрутизации в реальном времени.

- Всем узлам приходится использовать ограниченную полосу связи, например радиочастоты WiFi.

При этом так же растут расходы на поддержание топологии сети, поскольку узел должен самостоятельно искать новых абонентов при их появлении и поддерживать актуальность списка адресатов, удаляя из него недоступные больше узлы.

3.3. Широковещательные сети

Протокол WiFi предполагает соединение, где узел подключается к другому узлу напрямую через PtP соединение, однако при указании адреса 255.255.255.255 может отправлять широковещательные пакеты. Такие сообщения нужны для определения MAC адреса новых узлов сети. В условия большого количества узлов вокруг, в том числе новых, при помощи широковещательных сигналов можно эффективно распространять информацию. При этом данные получают все, для кого сигнал оказался достаточно мощным. Использование широковещательного сигнала является аналогом использования радиосвязи.

3.4. ESP-NOW

ESP-NOW — это протокол беспроводной связи, основанный на канальном уровне, который сводит пять уровней модели OSI к одному. Протокол ESP-NOW заменяет собой сетевой уровень, транспортный уровень, сеансовый уровень, уровень представления и уровень приложений. Кроме того, для данного протокола нет необходимости в заголовках пакетов или распаковщиках на каждом уровне, что приводит к ускорению передачи, уменьшая задержку, вызванную потерей пакетов в перегруженных сетях.

В ESP-NOW данные приложения инкапсулируются в единый пакет данных, определяемый библиотекой, а затем передаются с одного устройства Wi-Fi на другое без соединения.

Пакет данных ESP-NOW строго определен и выглядит следующим образом.

- MAC Заголовок, 24 байта – MAC-адрес отправителя.
- Категория сообщения, 1 байт – Указатель, является сообщение адресным или широковещательным.
- Идентификатор, 3 байта – первые три байта MAC-адреса, определённые EspressIF.
- Случайное значение, 4 байта – случайно сгенерированные цифры, защищающие от ретрансляционных атак.
- Полезная информация, 7-257 байт – адрес получателя и сообщение.
- FCS, 4 байта – Frame check sequence, сообщение из четырех байт, выполняющее роль контрольной суммы.

4. Система передачи сообщений

Проектируемая система передачи сообщений должна обладать следующими характеристиками:

- Узлы должны передавать небольшие сообщения ближайшим узлам;
- Сообщения имеют высокую частоту и должны иметь низкую задержку, чтобы не терять актуальность;
- Во время работы взаимное расположение узлов может меняться, что влечёт за собой изменение ближайших узлов;
- Актуальность информации узла уменьшается при удалении от него;
- Число узлов в сети может быть настолько велико, что отправка сообщений с необходимой частотой каждому узлу отдельно станет невозможной.

В связи с поставленными ограничениями была предложена система обмена сообщениями на основе широковещательного канала. Все узлы работают в клиентском режиме, получая, обрабатывая и отправляя сообщения. Узел отправляет свои сообщения по широковещательному каналу, принимая сообщения от всех узлов и обрабатывая в соответствии с требованиями системы.

4.1. Алгоритм работы узла

На каждой итерации узел выполняет следующие операции:

- Считывает данные с внутренних детекторов и корректируя курс;
- Принимает входящие сообщения, добавляя их в список и сортируя по силе сигнала RSSI;

- Из полученных сообщений выбирает пять с наилучшим сигналом и из них случайно выбирает одно или обрабатывает все сообщения;
- Пересчитывает курс, опираясь на данные, полученные в сообщении;
- Отправляет свой курс по широковещательному каналу, указывая свою уверенность в нём.

Данный подход позволяет отправлять сообщения сразу всем адресатам, что решает сразу две проблемы. Во-первых, широковещательный сигнал не требует подключения устройств друг к другу и экономит время на рассылке системных пакетов не несущих полезной информации. Во-вторых, каждое сообщение отправляется только один раз, нет необходимости рассылать данные каждому узлу по отдельности.

В зависимости от количества узлов в системе и поставленных задач входящие сигналы могут обрабатываться по разному. В случае, когда узлов мало и информация с каждого из них важна, узел принимает и обрабатывает все приходящие сообщения. В случае, когда узлов много и обработка всех сообщений будет задерживать отправку следующего сообщения, можно пренебречь некоторой информацией. При этом обрабатываться будет выбранное случайным образом сообщение из списка, отсортированного по качеству сигнала. Чем сильнее полученный узлом сигнал, тем ближе находится узел отправитель и тем релевантнее полученная от него информация.

5. Детали реализации

Для реализации был выбран протокол ESP-Now, разработанный Espressif для популярного модуля ESP-32. Данный протокол позволяет передавать сообщения длиной не более 250 символов в режиме Peer-to-Peer соединения, используя Wi-Fi модуль. При таком способе передачи время задержки гораздо ниже, чем при использовании классического способа [5]. ESP-Now позволяет не только отправлять сообщения для конкретного получателя, но и использовать широковещательные запросы, а также получать сообщения, адресатом которого является модуль.

5.1. ESP32

Реализация для ESP32 написана на Arduino с использованием библиотек WiFi и esp_now. Программа состоит из двух больших блоков – инициализации и главного цикла.

При инициализации запускаются службы WIFI, ESPNOW и подключаются UART порты, с которых будет считываться полезная информация.

В главном цикле, который стартует после окончания инициализации и работает всё время работы платы ESP32, реализована отправка сообщения. Программа проверяет доступность буфера UART и читает оттуда данные. После этого данные необходимо закодировать, так как отправляемое сообщение по протоколу ESP-NOW должно оканчиваться нулем. Поскольку наши данные могут содержать нули, считывание сообщения может прерваться раньше.

Для избежания данной проблемы данные разбиваются на отдельные байты, шестнадцатичное представление каждого байта превращается в символ и образованная таким образом строка формирует из себя сообщение. Такой способ кодирования увеличивает длину сообщения в два раза, с 24 символов до 48, однако позволяет избавиться от проблемы встречающихся нулей.

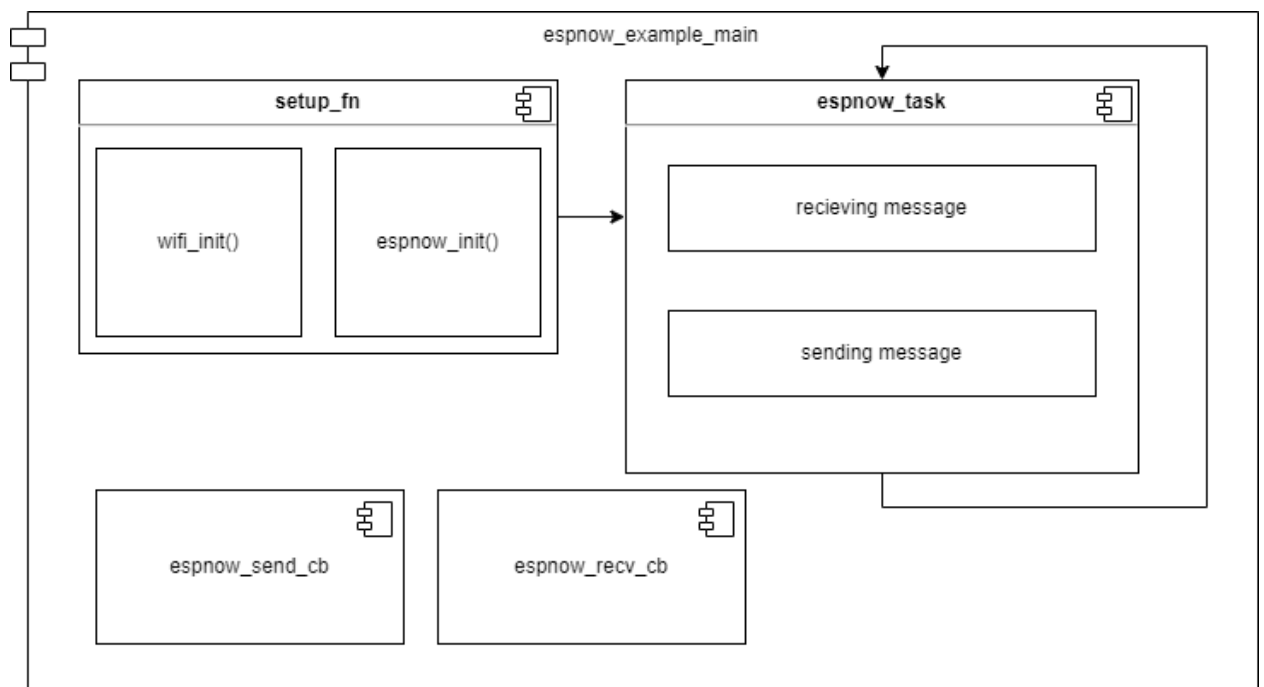
После того, как сообщение сгенерировано, оно отправляется на broadcastAddress – 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF при помощи

функции `esp_now_send`. Такое сообщение обработают все узлы, которые его получают.

Обработка сообщения происходит в функции `receiveCallback`, которая вызывается системой ESP каждый раз, когда приходит новое сообщение. В этой функции проверяется контрольная сумма сообщения, после этого буффер декодируется, чтобы восстановить изначальное сообщение. Далее сообщение передается на UART, где его сможет обработать навигатор.

5.2. Интеграция с MartOS

Программа состоит из трёх модулей - инициализации, задачи основного цикла и функций обратного вызова. Программа вызывается операционной системой MartOS начиная с функции инициализации, затем в бесконечном цикле запускается функция `espnow_task`. Функций обратного вызова две, первая отвечает за отправленные сообщения, вторая за полученные и вызываются они из модуля WiFi при наступлении соответствующего события.



5.2.1. Инициализация

Инициализация программы проходит в две стадии, запуск WiFi и запуск систем espnow.

Инициализация систем WiFi происходит внутри `wifi_init()` при помощи следующих функций.

- `esp_netif_init()` - создает абстракцию над стеком TCP/IP.
- `esp_event_loop_create_default()` - создает цикл для отправки обратных сообщений.
- `esp_wifi_init()` - инициализирует модуль WiFi в режиме станции, который позволяет ESP32 отправлять сообщения.
- `esp_wifi_start()` - запускает модуль WiFi.

Ошибки в любой из функций инициализации регистрируются при помощи функции `ESP_ERROR_CHECK`.

После того, как проверка запуска систем WiFi закончена, начинается инициализация систем espnow. Для этого используется функция `espnow_init()`.

Система espnow запускается при помощи функции `esp_now_init()`, а при помощи функций `esp_now_register_send_cb()` и `esp_now_register_recv_cb()` задаются функции обратной связи, в этой программе - `espnow_send_cb` и `espnow_recv_cb`.

Следующим шагом инициализации является добавление широковещательного адреса в список возможных адресатов. Для этого создается подключение с адресом `s_broadcast_mac`, состоящим из байтов `0xFF`.

Далее необходимо задать данные, используемые для отправки сообщений. Эти данные хранятся в глобальной переменной `send_param`. Здесь задается максимальная длина сообщения, буффер отправки, задержка между сообщениями и другие параметры отправки.

5.2.2. Функции обратного вызова

Для WiFi модуля ESP32 принято передавать данные сообщений через функции обратного вызова. Отдельно представлены функции отправки и приема сообщения, `espnw_send_cb()` и `espnw_recv_cb()` соответственно.

В функции `espnw_send_cb` программа обрабатывает отправку сообщения. В лог при помощи функции `ESP_LOGE` записывается факт отправки сообщения и программа выставляет истину в переменную, разрешая отправку следующего сообщения.

В функции `espnw_recv_cb` программа обрабатывает получение сообщения. Для этого сообщение записывается в очередь принятых сообщений, для того, чтобы в основном цикле программа смогла его обработать.

5.2.3. Функция главного цикла

Данная функция запускается в бесконечном цикле сразу после окончания функции инициализации и завершается цикл только при отключении устройства.

На каждой итерации цикла происходит обработка пришедших сообщений и отправка нового сообщения.

Обработка полученных сообщений происходит через очередь `message_queue`. В ней хранятся данные обо всех сообщениях, пришедших с последней итерации. Полученные сообщения обрабатываются по очереди. Обработка сообщения происходит в функции `espnw_data_parse()`, где проверяется длина сообщения и считывается информация.

Для отправки нового сообщения подготавливается буфер отправки при помощи функции `espnw_data_prepare()`, после чего новое сообщение отправляется по широкополосному каналу при помощи функции `esp_now_send()`.

6. Заключение

В ходе работы были достигнуты следующие результаты.

1. Выполнен обзор существующих систем передачи сообщений - Mesh-сети, Ad-hoc-сети, сети без маршрутизации.
2. Выполнена реализация системы передачи сообщений на базе библиотеки EspressIF для платформы ESP32.
3. Система передачи сообщений интегрирована в операционную систему MartOS.

Список литературы

- [1] Lundberg I.D., Decentralized communication. How to send messages to unresponsive clients in a chat network.// Bachelor's thesis. – 2015. - С. 1 – 25.
- [2] Довгаль В.А., Интеграция сетей и вычислений для построения умной системы управления роем дронов как сетевой системы управления.// Научный журнал "Вестник АГУ". - Вып. 1 (296). – 2022. - С. 62 – 76.
- [3] Paulo Chainho, Steffen Drusedow, Ricardo Lopes Pereira, Ricardo Chaves, Nuno Santos, Decentralized Communications: Trustworthy Interoperability in Peer-To-Peer Networks.// Altice Labs, Deutsche Telekom AG. – 2017.
- [4] Navid Alibabaei, Wireless Mesh Networks: a comparative study of Ad-Hoc routing protocols toward more efficient routing.// Master Thesis. – 2019.
- [5] Eridani, Dania Rochim, Adian Cesara, Faiz. Comparative Performance Study of ESP-NOW, Wi-Fi, Bluetooth Protocols based on Range, Transmission Speed, Latency, Energy Usage and Barrier Resistance. // International Seminar on Application for Technology of Information and Communication, 2021. - с. 322-328.
- [6] J. Jun, M.L. Sichitiu, The nominal capacity of wireless mesh networks.// IEEE Wireless Communications, vol 10. – 2003. - С. 8 - 14.
- [7] Dimitrios J Vergados, Natalia Amelina, Yuming Jiang et al., Toward optimal distributed node scheduling in a multihop wireless network through local voting.// IEEE Transactions on Wireless Communications. — 2017. — Vol. 17, no. 1. — P. 400–414.
- [8] https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/network/esp_now.html, дата обращения 03.03.2024

- [9] Omar Sami Oubbati, Mohammed Atiquzzaman, Pascal Lorenz et al., Routing in flying ad hoc networks: Survey, constraints, and future challenge perspectives .// IEEE Access. — 2019. — Vol. 7. — P. 81057–81105.
- [10] Cilfone Antonio, Davoli Luca, Belli Laura. Wireless Mesh Networking: An IoT-Oriented Perspective Survey on Relevant Technologies. // Future Internet. – 2019. 11 (4): p. 99