Метод коммуникации в децентрализованной сети автономной группы мобильных роботов[1]

В. Н. Калитеевский
Санкт-Петербургский государственный университет
vkalit@gmail.com

В статье рассматривается задача сохранения целостности данных в децентрализованной сети автономных мобильных роботов, а также программно-аппаратная возможность реализации такой сети. Для коллективного хранения информации в группе мобильных роботов предлагается использовать алгоритмы подсчета контрольных сумм и балансировки загрузки вычислительных узлов. Для программной реализации передачи данных применяются мультиагентные технологии. В качестве мобильного робота рассматривается сверхлегкий беспилотный летательный аппарат (БПЛА).

Ключевые слова: мультиагентные системы, групповое взаимодействие, балансировка загрузки вычислительных узлов, подсчет контрольных сумм, средства связи, архитектура мобильного робота.

1. Введение

Применение различного рода мобильных роботов для выполнения задач мониторинга и исследования местности все чаще входит в нашу повседневную жизнь [1, 2]. В последнее время все больший интерес представляют не только обособленные мобильные роботы, но и их группы [3, 4]. Использование группы мобильных роботов позволяет увеличить гарантии выполнения поставленной задачи, уменьшить время ее выполнения.

В большинстве случаев при реализации алгоритмов группового взаимодействия мобильных роботов используется централизованная схема коммуникации, в которой один из роботов является лидером, а все остальные члены группы либо повторяют его движения, либо ориентируются на него [5, 6, 7]. Под централизованной сетью

[1] ©В. Н. Калитеевский, 2015

далее будем понимать сеть, в которой членам группы для кобщениянь необходимо взаимодействовать с каким-либо общим устройством, как показано на Рис. 1 (роутером, общими вспомогательными программами и т. п.). В такой схеме при потере связи с лидером или общим устройством мобильный робот продолжает выполнять задания обособленно, а при выходе из строя лидера или общего устройства группа роботов перестает работать как группа [8]. Также при централизованной схеме общения между устройствами может оказаться проблемой увеличение количества узлов из-за пропорционального увеличения нагрузки на центральный узел.

Другой схемой реализации группового взаимодействия является децентрализованная. Она изображена на Рис. 2 [9]. Такая сеть предполагает полное отсутствие общего устройства, необходимого для связи в группе. Также в децентрализованной системе либо происходит постоянное двухстороннее взаимодействие между роботами, либо возможна смена временного лидера.

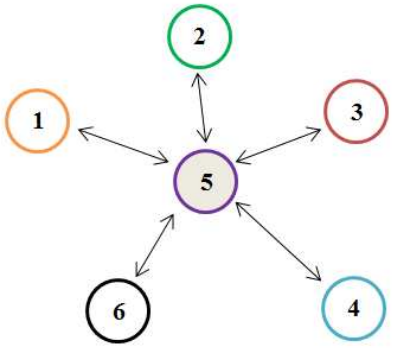

Рис. 1: Пример централизованной сети с постоянной топологией связи и с узлом-координатором (узел 5).

Сложность реализации децентрализованной сети заключается в отсутствии центрального элемента сети, через который можно передавать сообщения и хранить в нем адреса всех других роботов (узлов сети). В централизованной же сети такой проблемы нет, потому что каждый узел знает, как отправить сообщение, как ми-

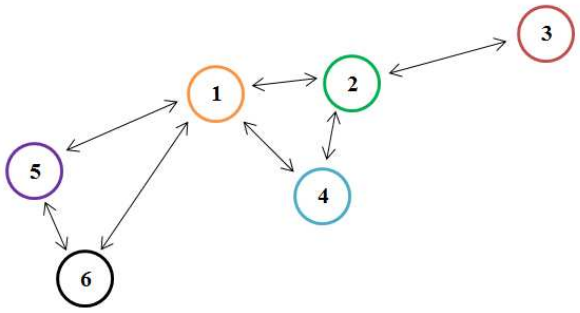

Рис. 2: Пример децентрализованной сети с переменной топологией связи.

нимум, одному узлу-координатору. Координатор, в свою очередь, может отправлять сообщения любому узлу сети.

Как правило, координатор используется для передачи сообщений между всеми узлами, а значит к нему могут предъявляться высокие требования по производительности и сетевому взаимодействию. Остальные узлы в сети могут быть менее производительными, и поэтому в случае выхода координатора из строя его роль ни один из них на себя взять не сможет.

В децентрализованной сети координатор отсутствует, поэтому каждый узел может посылать сообщения только узлам “соседям”, которые находятся в зоне видимости этого узла в текущий момент времени. Таким образом, существенно изменяется характер взаимодействия из-за того, что каждый узел видит только своих соседей.

Одним из практических примеров реализации децентрализованной сети может служить автономная система сверхлегких летательных аппаратов без выделенного координатора, в которой члены группы могут летать на большом расстоянии друг от друга, и каждый из них, теоретически, может выйти из строя или оказаться вне зоны досягаемости других БПЛА.

Для обеспечения эффективного выполнения общей задачи децентрализованной системой все большую популярность набирают мультиагентные технологии, где основными действующими элементами являются агенты, которые могут иметь как одинаковую, так

и разную значимость и функциональность в системе [10, 11, 12]. Применение мультиагентных технологий дает возможность реализовать автономное групповое взаимодействие в гомогенной или гетерогенной сети мобильных роботов с переменной сетевой топологией.

Для реализации связи между мобильными роботами в децентрализованной сети с переменной топологией также предстоит решить вопрос аппаратного оснащения узлов. В связи с этим встает задача по разработке технологии коммуникации, включающая в себя аппаратную, программную и алгоритмическую части.

В этой работе предложена схема аппаратной реализации коммуникации в децентрализованной сети мобильных роботов на примере группы БПЛА вертикального взлета. Также предложен алгоритм взаимодействия в группе, и описана его программная реализация. Для предложенной системы в работе рассматривается одна из наиболее актуальных задач для БПЛА — хранение данных и сохранение их целостности (возможность восстановления утерянных данных). Предложен алгоритм реализации RAID-подобной схемы для децентрализованного подсчета контрольных сумм данных на мобильных роботах.

2. Обзор предметной области

Аппаратное обеспечение связи мобильных роботов.

В качестве средств связи в автономной группе мобильных роботов можно рассмотреть следующие варианты:

1. Модули Bluetooth.
2. Модули Wi-Fi.
3. Модули мобильной связи (GPRS, EDGE, 3G, 4G).
4. Специализированные модули связи.

Bluetooth и Wi-Fi обеспечивают беспроводную связь между устройствами на удалении до 100 метров, однако, это может слишком

сильно сковывать группу мобильных роботов, заставляя держаться на небольшом расстоянии друг от друга. Также минусом таких видов связи является сложная система идентификации или необходимость в центральном узле [6]. Для использования мобильной сети необходимо находится в зоне покрытия мобильной связи, что приводит к привязке к дополнительным устройствам.

В качестве специализированных модулей связи для обеспечения децентрализованной коммуникации группы мобильных роботов можно рассмотреть XBee-модемы. XBee и XBee PRO модули были разработаны для работы со стандартом IEEE 802.15.4 и поддержки дешевых, отличающихся низким энергопотреблением, беспроводных сетей [13]. Также они могут обеспечивать относительно большое расстояние передачи данных (до 1600 метров) и поддерживают MESH-топологии (сетевые топологии, где каждый элемент может связывать между собой своих соседей), что принципиально в случае развертывания полностью децентрализованной сети. Среди особенностей модулей XBee следует выделить:

1. Возможность шифрования данных (AES-128).
2. Поддерживают, как стандартную (1 мВт), так и повышенную мощность (2 мВт).
3. Поддерживают индустриальный температурный диапазон: (-40C.. 85C).

Мультиагентные технологии.

Большинство современных систем управления группой мобильных роботов не имеют автономного генератора новых заданий, позволяющего группе оперативно принимать эффективные решения по изменению сценария выполнения поставленной задачи. Типичными примерами событий, вызывающих необходимость в постановке новых задач, являются: появление новой выгодной информации для более эффективного выполнения задачи, выход из строя части имеющихся ресурсов, а также изменение критериев принятия решений. Чем выше неопределенность, чем более распределенный характер имеют процессы принятия решения, и чем чаще случаются незапланированные события, тем ниже эффективность существующих систем, неспособных самостоятельно принимать решения и

автоматически перестраиваться под изменения в среде [14]. Кроме того, любая модификация схем принятия решений в традиционных системах представляет собой весьма сложный и трудоемкий процесс и требует высокой квалификации исполнителей, что делает разработку и эксплуатацию рассматриваемых систем крайне дорогостоящими. Для работы в таких условиях хорошо зарекомендовали себя МАС (мультиагентные системы). В отличие от классического способа, когда проводится поиск некоторого четко определенного (детерминированного) алгоритма, позволяющего найти наилучшее решение проблемы, в мультиагентном подходе система со временем выходит на оптимальный уровень в результате взаимодействия множества самостоятельных целенаправленных программных модулей — так называемых агентов. Характерными особенностями агентов являются:

1. Коллегиальность, т. е. способность к коллективному целенаправленному поведению в интересах решения общей задачи. Выбор действий каждым членом коллектива осуществляется только на основе информации о коллективной цели, стоящей перед группой, а также в зависимости от ситуации в среде.

2. Автономность, т. е. способность самостоятельно решать локальные задачи.

3. Активность, т. е. способность к активным действиям ради достижения общих и локальных целей.

4. Информационная и двигательная мобильность, т. е. способность активно перемещаться и целенаправленно искать и находить информацию, энергию и объекты, необходимые для кооперативного решения общей задачи.

5. Адаптивность, т. е. способность автоматически приспосабливаться к неопределенным условиям в динамической среде.

Таким образом, можно определить понятие кагентаѣ. Агент — разумная сущность, наблюдающая за окружающей средой и действующая в ней, при этом его поведение рационально в том смысле, что он способен к пониманию, и его действия всегда направлены на достижение какой-либо цели [11].

Каждому агенту соответствуют такие ментальные категории, как убеждения (beliefs), желания (desires) и намерения (intention) [11, 15]:

1. Убеждения — представления о текущем состоянии среды.
2. Желания — состояния среды, в которые агент стремится среду перевести.
3. Намерения — множество избранных, совместимых и достижимых желаний.

Разделение намерений и желаний необходимо, так как агент может иметь несовместимые желания, или желания могут быть недостижимы. Поскольку агент ограничен в ресурсах и не может достичь всех желаний одновременно, естественно выбирать наиболее значимые цели — намерения. Можно выделить три аспекта, характеризующие агентов: они содержат знания, ощущают окружающую среду и выполняют действия.

Работающая по таким принципам группа мобильных роботов имеет ряд преимуществ по сравнению с централизованной группой, благодаря следующим особенностям:

1. Может осуществляться передача информации в децентрализованной сети за счет возможности передачи информации от агента к агенту.
2. Изменение задачи для группы и для каждого члена группы может происходить в режиме реального времени даже при отсутствии связи с базовой станцией.
3. Возможность перераспределять задания между членами группы для достижения глобальной цели (например, при выходе из строя одного или нескольких членов группы).

Эти возможности кардинально отличают мультиагентные системы от существующих кжесткоѣ организованных систем.

Примером применения мультиагентных технологий является использование алгоритма локального голосования, который позволяет достигать консенсуса в стохастической динамической сети с неполной информацией и задержками в измерениях [11, 14, 21].

Платформы для реализации МАС.

За последние годы появилось множество инструментариев, поддерживающих различные агентные архитектуры и предоставляющие библиотеки для протоколов взаимодействия: Jason, SPADE, JADE, Cougaar, Jadex и т. д. Были приняты стандарты в области мультиагентных систем, такие как KQML (Knowledge Query and Manipulation Language) [16], который изначально создавался для экспертных систем (позже был принят и в области мультиагентных технологий) и FIPA (Foundation for Intelligent Physical Agents) [17].

Целью этих стандартов является спецификация поведения агентов в системе. Сюда входят способы сосуществования агентов в системе, коммуникация и взаимодействие между собой. Согласно FIPA-спецификациям выделяются три служебных роли агентов:

1. Система управления агентами (Agent Management System, AMS) — этот агент отвечает за доступ к справочнику всех агентных идентификаторов (AID). Каждому AID в справочники сопоставляется адрес агента в системе, через который будет возможна передача сообщения по транспортному протоколу. Все агенты платформы регистрируются в справочнике через AMS при появлении в системе, чтобы получить AID. Взаимодействие разных платформ между собой через AMS показано на Рис. 3.

2. Справочный агент (Directory Facilitator, DF) — предоставляет сервис “Желтых страниц” другим агентам в системе. Каждый из агентов должен зарегистрировать свои сервисы в этом справочнике. При обращении к DF для каждого конкретного сервиса клиенту возвращается список агентов, реализующих этот сервис.

3. Сервис транспорта сообщений (Message Transport Service, MTS) — отвечает за передачу сообщений между агентами в рамках одной платформы, а также за связь с агентами в других платформах.

У каждой платформы можно выделить общий набор свойств: технологии, лежащие в основе платформы, область применения,

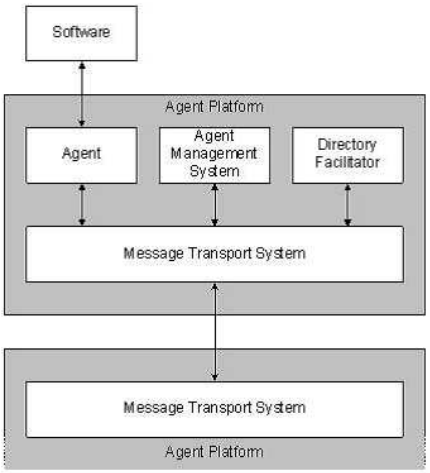

Рис. 3: Архитектура FIPA.

расширяемость платформы, возможность интеграции, качество сопроводительной документации, лицензия и другие.

Также можно выделить основные свойства платформы, связанные с разработкой МАС: поддержка методологии разработки (определяет виды работ, результаты работы на всех этапах разработки МАС: анализ, проектирование, реализация, тестирование и отладка, развертывание), наличие инструментария разработки и особенности реализации агентов и МАС (языки описания агентов, наличие механизмов интеллекта, коммуникации).

В распределенном JADE (Рис. 4) согласно [18] один из контейнеров (хостов), называемый главным, должен содержать агенты-координаторы. Для работы агентов в сети компьютеров, которые могут в любой момент связаться с главным контейнером, такой подход не мешает децентрализованному общению между агентами. Для децентрализованной сети автономных мобильных роботов, выделение главного контейнера означает централизацию системы. Выделение агентов-координаторов, присущее и другим платформам не позволяет сделать систему полностью децентрализованной.

В частности, такие фреймворки как JADE могут оказаться слиш-

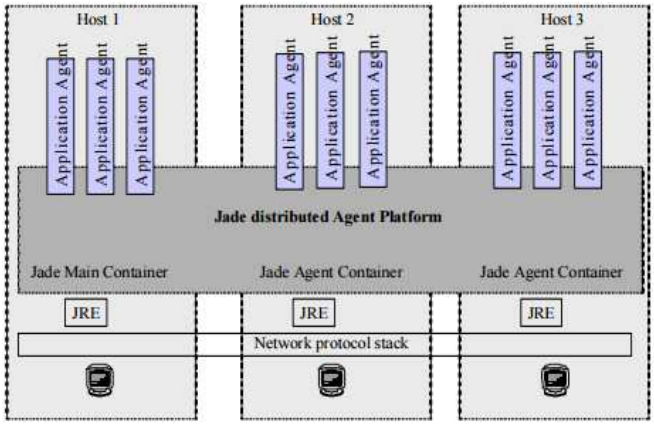

Рис. 4: Распределение основных элементов системы по роботам (хостам) на платформе JADE.

ком тяжелыми для аппаратного обеспечения мобильного робота, на котором, как правило, стоит достаточно слабый процессор и малое количество оперативной памяти.

3. Постановка задачи

Целью данной работы является проектирование и реализация метода коммуникации в децентрализованной сети группы мобильных роботов. Для достижения этой цели выделены следующие подзадачи:

1. Разработать технологию программирования бортового микрокомпьютера мобильных роботов для организации их взаимодействия в децентрализованной сети с использованием FIPA-спецификаций.

2. Реализовать на практике RAID-подобную схему для децентрализованного подсчета контрольных сумм данных в автономной сети мобильных роботов с применением разработанных технологий.

3. Разработать конфигурацию модулей XBee для передачи информации в децентрализованной автономной сети мобильных роботов.

4. Технология коммуникации в децентрализованной сети мобильных роботов

Прототип технологии коммуникации группы роботов включает в себя аппаратную, программную и алгоритмическую части. Аппаратная часть включает в себя бортовой микрокомпьютер, средства коммуникации, а также архитектуру устройства робота. Аппаратная часть самого мобильного робота также интересна, но в этой работе большее внимание будет уделяться именно системам коммуникации между роботами. Программная часть включает реализацию системы взаимодействия агентов и сетевого стека протокола общения. Все вышеперечисленное реализуется в рамках создания собственного расширяемого и стандартизированного фреймворка для взаимодействия группы мобильных роботов. Алгоритмическая часть включает в себя алгоритмы обработки передаваемой информации, группового взаимодействия роботов в децентрализованной сети, а также, для апробации и внедрения, алгоритм реализации RAID-подобной системы хранения данных с ненадежными связями.

Программная часть.

В связи с обозначенными требованиями качестве программной реализации взаимодействия мобильных роботов спроектирована собственная децентрализованная система взаимодействия агентов За основу такой системы взяты FIPA-стандарты по кобщениюь между агентами. В рамках такой схемы [17] агенты общаются между собой на языке Agent Communication Language (ACL). Язык определяется набором параметров, передаваемых вместе с сообщением. Обязательным параметром является performative: он указывает на сам тип языка. Также ожидаются параметры sender / receiver / reply-to для большинства сообщений, указывающие на того, кто является отправителем, кто получателем, а кому предназначается данное сообщение. Полный список параметров приведен в Таблице 1.

Таблица 1: Параметры сообщения для FIPA ACL.

Параметр	Категория
performative/Тип сообщения	Тип коммуникации
sender/Отправитель	Роль коммуникации
receiver/Получатель	Роль коммуникации
reply-to/Адресат	Роль коммуникации
content/Содержание	Содержание сообщения
language/Язык	Техническое описание
encoding/Кодировка	Техническое описание
ontology/Онтология	Техническое описание
protocol/Протокол	Контрольная характеристика
conversation-id/Идентфикатор	Контрольная характеристика
reply-with	Контрольная характеристика
in-reply-to	Контрольная характеристика
reply-by	Контрольная характеристика

В качестве языка для реализации фреймворка был выбран Python версии 2.7. Он является кроссплатформенным, а код на нем, как правило, получается лаконичен и понятен. Следует учитывать, что целевой архитектурой настоящего фреймворка является автономный мобильный робот. Однако с точки зрения энергопотребления выбор языка несущественен, поскольку затраты на работу двигателя на порядки выше. Версия 2.7 языка Python была выбрана потому что для этой версии реализована библиотека PySerial, сильно облегчающая работу с портами. Также эта версия является стабильной сборкой и хорошо себя зарекомендовала среди пользователей Python.

Основные элементы фреймворка представлены на Рис. 5. Здесь каждый робот (хост) отождествляется с одним агентом. Модуль Transmit отвечает за передачу сообщений на другие роботы и на различные аппаратные элементы этого же робота, а также отвечает за список тех роботов, которым можно послать сообщение в данный момент. Модуль Algorithm отвечает за работу каждого алгоритма, реализованного на роботе. Алгоритмы балансировки загрузки и подсчета контрольных сумм — примеры децентрализован-

ных алгоритмов, которые могут работать на данной системе.

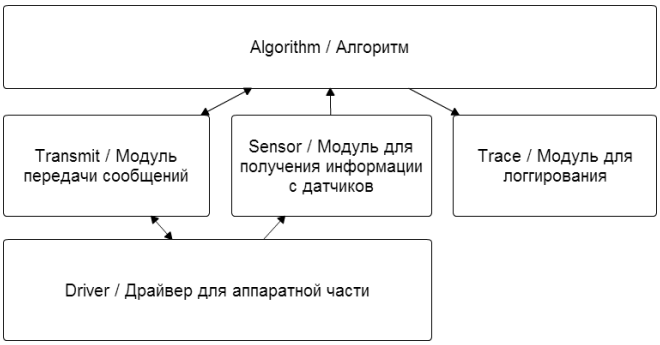

Рис. 5: Архитектура фреймворка.

Более подробную структуру системы показывает диаграмма классов, изображенная на Рис. 6.

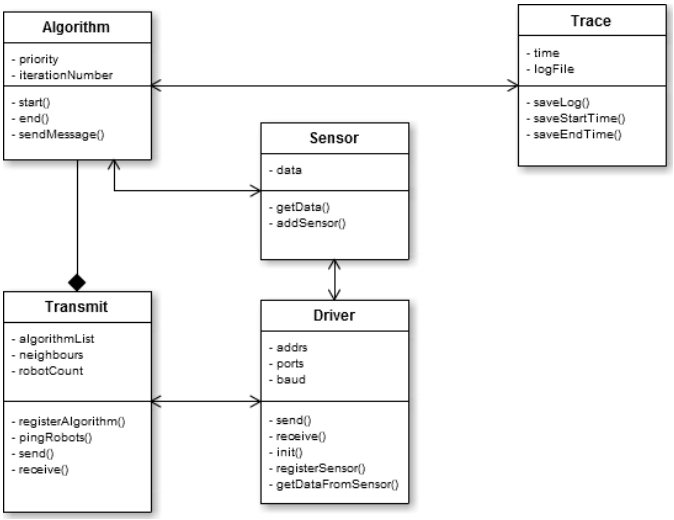

Рис. 6: Диаграмма классов UML.

Алгоритм передачи сообщений

Алгоритм передачи сообщений служит для того, чтобы одинаковые алгоритмы (Модуль Algorithm) на разных роботах могли передавать друг другу сообщения. Для этого модуль Transmit создает каждому алгоритму очередь сообщений, куда помещает новые пришедшие сообщения для этого алгоритма. Алгоритм читает сообщения из этой очереди и обрабатывает. Прочитав сообщение, алгоритм анализирует новые, хранящиеся в нем данные. Когда алгоритму необходимо самому отправить какую-либо информацию другому агенту, он добавляет нужные параметры согласно языку ACL в заголовок и передает это сообщение модулю Transmit с указанием номера агента, которому предназначается это сообщение. Модуль Transmit добавляет собственный заголовок к сообщению, в котором указывает имя алгоритма отправившего сообщение, а также номер итерации для данного алгоритма и передает модулю Driver. Модуль Driver, работающий с модулями передачи информации, добавляет свой заголовок к полученному от модуля Transmit сообщению, в котором абстрактный номер агента-получателя заменяется конкретным MAC-адресом. Сформировав пакет, он отправляет его с помощью физического модуля передачи данных. На другом агенте алгоритм, которому предназначалось это сообщение, получает его и обрабатывает в обратном порядке.

Свежую информацию о количестве агентов на связи алгоритмам предоставляет модуль Transmit, который по запросу может вызвать функцию $pingRobots()$, отправляющую каждому роботу в системе запрос на подтверждение связи. Получив запрос на наличие связи а другом агенте такой же модуль Transmit отвечает на него вне очереди. После этого, выждав определенный интервал времени, модуль Transmit считает количество полученных подтверждений и обновляет свою таблицу соседей, т. е. роботов, находящихся на связи. С роботами, сообщение от которых не пришло или пришло по истечении таймаута, связь считается потерянной до следующего вызова функции $pingRobots()$. Блок-схема процедуры обработки входящих сообщений модулем Transmit показана на Рис. 7.

Для отслеживания времени работы алгоритмов, а также хранения логов, используется модуль Trace, отсчитывающий время работы с начала запуска системы. Если алгоритму требуется получить доступ к каким-то данным с датчиков или к оборудованию, напри-

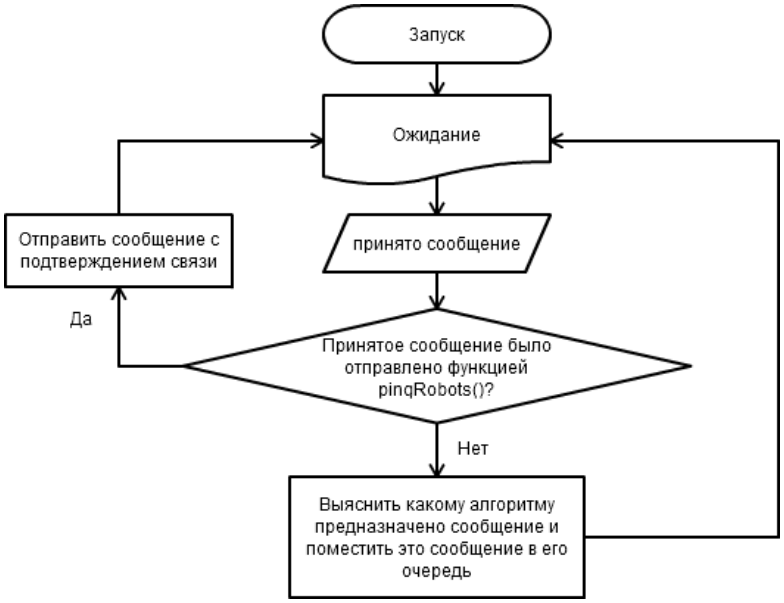

Рис. 7: Обработка полученного сообщения модулем Transmit.

мер, получить снимок с фотоаппарата, эти данные предоставляет ему модуль Sensor.

Для примера рассмотрим передачу сообщения *алгоритма* k на *роботе* n *алгоритму* k на *роботе* m, как показано на Рис. 8: Модуль Algorithm передает текст сообщения модулю Transmit с ACL-заголовком, где указывает код алгоритма-отправителя, код-алгоритма-получателя и все другие необходимые поля. Модуль Transmit, в свою очередь, добавляет MTP-заголовок, в котором указывает адрес робота-отправителя, адрес робота-получателя, номер итерации алгоритма и другие согласно формату. Затем передает сообщение модулю Driver, который отправляет пакет по сети. Driver на *роботе* m пакет принимает, передает Модулю Transmit, тот видит, что сообщение предназначено *алгоритму* k и передает его ему.

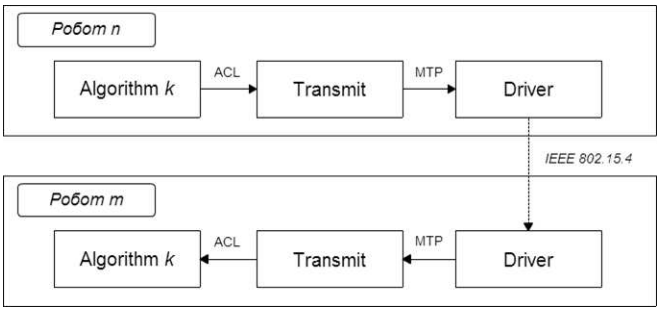

Рис. 8: Пример передачи сообщения между алгоритмами.

5. Апробация и внедрение

Аппаратная часть.

Для апробации системы, для начала надо подготовить аппаратную часть. В качестве аппаратной части мобильного робота предлагается использовать трехуровневую систему управления мобильными роботами, которая изображена на Рис. 9. Основной идеей такой системы является наличие дополнительного микрокомпьютера, помимо автопилота, с помощью которого появляется возможность применения мультиагентных технологий [10]. В качестве микрокомпьютера используется Raspberry Pi model b+, который построен на базе процессора BroadcomSoC ARM11. По умолчанию на устройстве установлена ОС Debian [19].

В качестве модулей связи используются модули связи XBee серии Pro, которые позволяют передавать данные на расстояние до 1.5 км, а также реализовать радиосвязь в группе с отсутствием единого центра, а именно, каждое устройство может выступать в качестве инициатора передачи данных и каждое может принимать данные [13]. XBee-модуль изображен на Рис. 10.

Модули работают на частоте 2.4 ГГц. Скорость передачи составляет 250 бит в секунду. Подключение модулей к микрокомпьютеру Raspberry осуществляется через FTDI плату с использованием UART интерфейса на базе RS-232. Модули имеют PAN ID (Personnel Area Network ID), который позволяет им общаться в одной сети без привязки к адресам друг друга. Для защиты сети мо-

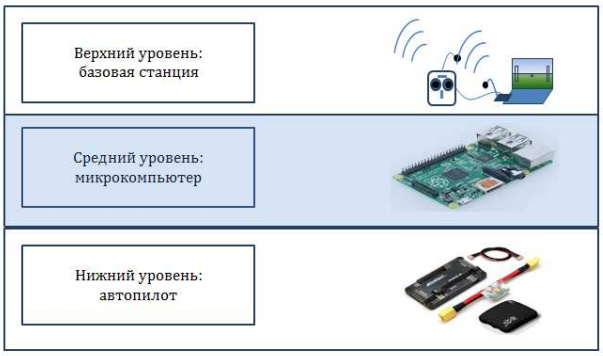

Рис. 9: Трехуровневая система.

Рис. 10: XBee-модуль.

дули XBee имеют возможность настройки кодировки передаваемой информации. Модули XBee можно использовать для организации беспроводного канала связи между роботами, микроконтроллерами, компьютерами и любыми устройствами с последовательным портом. Для передачи сообщения на другой модуль необходимо знать его MAC-адрес. Таким образом, зная адреса всех устройств в сети, можно построить многосвязную топологию ккаждый с каждымнь.

Системы хранения данных.

Одной из основных задач для автономных мобильных роботов является сбор данных и передача их пользователю [1]. При хранении данных в группе мобильных роботов важно обеспечить их целостность даже при отсутствии связи части мобильных роботов группы с пользователем. Самым простым способом сохранения целостности является хранение на каждом мобильном роботе копий всех данных других роботов. Однако в таком случае может потребоваться большое количество памяти на жестких дисках мобиль-

ных роботов. Чтобы хранить меньше избыточных данных, предлагается использовать RAID-подобные схемы, которые вычисляют контрольные суммы и затем по ним восстанавливают данные.

Системой хранения данных (СХД) называется совокупность оборудования и программного обеспечения, предназначенная для хранения больших объемов информации и бесперебойного доступа к ней. В распределенных СХД появляется проблема передачи данных между устройствами, если они расположены далеко друг от друга или соединены медленными и ненадежными каналами связи.

Рассмотрим системы хранения данных, построенные по технологии RAID. RAID (redundant array of independent disks) — это массив из нескольких устройств хранения данных, связанных между собой скоростными каналами передачи информации и воспринимаемых внешней системой как единое целое. Такой массив должен сохранять целостность данных при выходе из строя одного или нескольких устройств, а также обеспечивать высокую скорость чтения или записи данных. Каждое устройство хранения данных можно воспринимать как полностью независимый объект, находящийся сколь угодно далеко от других устройств. В этом контексте RAID-массив можно представить как мультиагентную систему, где в качестве одного агента выступает одно устройство хранения данных. Причем некоторые агенты либо вообще не могут общаться друг с другом, либо общаются по медленным и не надежным каналам передачи данных, а также только в определенные моменты времени.

Распределенной RAID-подобной СХД с ненадежными связями между устройствами хранения данных служит группа БПЛА, где каждый БПЛА выступает в качестве одного устройства хранения данных. Передача данных между двумя БПЛА возможна только тогда, когда они подлетают близко друг к другу. При выходе из строя нескольких БПЛА в дальнейшем возможно восстановление данных, которые были на них.

Алгоритмическая часть.

При использовании группы автономных мобильных роботов актуальной задачей является сохранение целостности данных при возможности отсутствия связи с несколькими членами группы или выхода их из строя. Предлагается решать эту проблему, используя

RAID-подобные схемы [20], что сделает работу с данными более безопасной и сделает возможным их восстановление в случае утраты. Также для автономной передачи данных в децентрализованной сети используется алгоритм балансировки загрузки вычислительных устройств [21].

Целостность данных в децентрализованной сети обеспечивается алгоритмом подсчета контрольных сумм [12]. Предположим, каждый автономный мобильный робот хранит свои данные в виде набора файлов. Если взять по одному файлу со всех роботов кроме i-ого, то для такого набора согласно [12] можно посчитать контрольную сумму и расположить ее на i-ом роботе. По аналогии можно считать и хранить несколько контрольных сумм для набора файлов со всех роботов кроме тех, на которых будут храниться контрольные суммы для этого набора.

Контрольные суммы для групп файлов вычисляются периодически в заданные интервалы времени, и после окончания интервалов можно восстанавливать часть утерянных файлов в этих группах. Перед началом каждого интервала предполагается, что все файлы из группы доступны для чтения.

При моделировании в [12] использовался выделенный агент, который имел доступ ко всем остальным агентам и определял момент окончания подсчета контрольных сумм. В системе из группы автономных мобильных роботов нет выделенного члена группы, который бы имел постоянный доступ ко всем остальным мобильным роботам. Поэтому для определения момента окончания подсчета контрольных сумм между всеми роботами постоянно курсирует маркер, который служит для определения того, что вычисляемая контрольная сумма имеет одинаковое значение на всех роботах. Если какой-то робот не получал маркер в течение долгого времени, то он сам генерирует маркер и пускает его по кругу.

На Рис. 11 показана схема размещения файлов на пяти БПЛА. При таком подходе каждый робот хранит как данные, так и контрольные суммы. Будем называть страйпом набор файлов, в котором ровно по одному файлу от каждого робота. При таком подходе перед вычислением контрольных сумм все файлы округляются нулями до размера самого большого файла в страйпе. В дальнейшем при восстановлении утерянных файлов будут восстановлены файлы с нулями на конце. Чтобы обрезать эти нули нужно знать исходные

размеры всех файлов. Поэтому на каждом БПЛА нужно хранить файл, в котором указаны точные размеры всех файлов в системе.

Каждый страйп хранит несколько видов контрольных сумм. В дальнейшем можно будет восстанавливать файлы с любых m роботов, где m –Ц количество контрольных сумм разного типа (то есть можно восстановить файлы с любых m мобильных устройств).

Робот 1	Робот 2	Робот 3	Робот 4	Робот 5
Мета 1	Мета 2	Мета 3	Мета 4	Мета 5
К.С. 1.1	К.С. 2.1	Фото 3.1	Фото 4.1	Фото 5.1
Фото 1.1	К.С. 2.2	К.С. 3.1	Фото 4.2	Фото 5.2
Фото 1.2	Фото 2.1	К.С. 3.2	К.С. 4.1	Фото 5.3
Фото 1.3	Фото 2.2	Фото 3.2	К.С. 4.2	К.С. 5.1
К.С. 1.2	Фото 2.3	Фото 3.3	Фото 4.3	К.С. 5.2
К.С. 1.3	К.С. 2.3	Фото 3.4	Фото 4.4	Фото 5.4
Фото 1.4	К.С. 2.4	К.С. 3.3	Фото 4.5	Фото 5.5

Рис. 11: Файлы с данными и с контрольными суммами на каждом агенте.

Рассмотрим детально децентрализованный подсчет контрольных сумм на пяти роботах для одного страйпа (несколько страйпов считаются последовательно):

1. Вычисляем точный размер самого большого файла с данными в страйпе - X.
2. Каждый робот загружает свой файл в оперативную память и если его размер меньше X, то дополняет его нулями до размера X. Если на мобильном устройстве для данного страйпа файл с данными отсутствует, то в качестве такого файла

в оперативной памяти берем кусок заполненный нулями по размеру X.

3. На каждом роботе разбиваем загруженный в память файл на набор чисел и в зависимости от типа контрольной суммы вычисляем соответствующую функцию от каждого числа.

4. По алгоритму локального голосования вычисляем среднее значение для каждого набора чисел и умножаем его на количество роботов. Таким образом, получим значение контрольной суммы нужного типа.

5. Сохраняем полученную контрольную сумму на нужном мобильном устройстве (расположение контрольных сумм показано на Рис. 11).

6. Пишем точные размеры всех файлов в страйпе в файл метаданных на каждом мобильном устройстве.

Восстановление файлов после их потери будет происходить централизованным способом после возвращения беспилотников на базу с помощью решения систем линейных уравнений, в которых в качестве неизвестных выступают утерянные данные. После решения системы линейных уравнений для каждого набора чисел в страйпе (для каждой дорожки в страйпе) останется только обрезать полученные файлы до точного размера с помощью значений из файла с метаданными.

Апробация системы.

Для проверки корректности работы алгоритма и правильности выдаваемых им результатов были сделаны различные тесты, указанные в Таблице 2. Во всех рассматриваемых случаях на каждом роботе находится один файл размером 50 KB.

В рамках отдельного эксперимента в воздух были запущены три БПЛА и еще два были реализованы на земле в виде микрокомпьютеров с модулями XBee. Каждый БПЛА сделал одну фотографию местности, а после этого был запущен децентрализованный алгоритм расчета контрольных сумм, которые сохранялись в виде файлов на соответствующих БПЛА. Также, на каждом БПЛА создавался файл с метаданными, который содержал размеры всех файлов. В какой-то момент беспилотники, поднятые в воздух, улетели

Таблица 2: Время сходимости в минутах в зависимости от топологии и количества роботов.

Кол-во роботов, N	Кольцо, мин.	Линия, мин.
4	3	8
5	6	20
6	11	40
7	17	60
8	25	82

на расстояние недосягаемости сигнала до БПЛА, находящихся на поверхности земли. После возвращения их на базу, два беспилотника были выключены вручную, а оставшиеся подключены к базовой станции. Несмотря на временный разрыв связи между воздушными БПЛА и теми БПЛА, которые находились на земле, после копирования файлов на базовую станцию утраченные фотографии (сделанные выключенными БПЛА) были полностью восстановлены по просчитанным контрольным суммам.

На Рис. 12 изображена зависимость необходимого размера хранимой части избыточных данных в % от количества недоступной для восстановления информации (количества недоступных БПЛА группы, которая состоит из 10 БПЛА). На графике видно, что при возможной утрате нескольких БПЛА группы для восстановления всего набора данных по данным оставшихся БПЛА, контрольные суммы занимают меньше памяти, чем весь набор со всех БПЛА. Так например, если мы предполагаем, что есть возможность возврата 50 процентов группы, то дополнительная информация для восстановления в виде контрольных сумм будет занимать память в два раза меньше нежели копировать всю информацию со всех БПЛА. Также показано, что хранение 100% всей контрольной суммы равносильно по объему хранению всех данных на каждом БПЛА со всех БПЛА. Поэтому хранение контрольной сумма выгодно для сохранения места в памяти при 100% вероятности прилета не менее 3 членов группы.

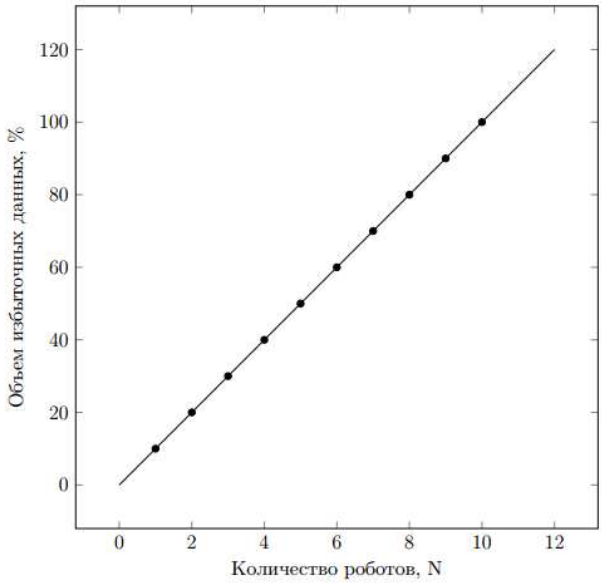

Рис. 12: График зависимости необходимого размера хранимой части избыточных данных в % от количества недоступной для восстановления информации (количества недоступных БПЛА группы, которая состоит из 10 БПЛА).

6. Заключение

В работе представлена технология организации связи в децентрализованной сети группы мобильных роботов. Предложена реализация децентрализованного алгоритма по подсчету контрольных сумм данных для сохранения целостности данных при возможном отсутствии связи с частью мобильных роботов или выходе их из строя. Спроектированная и реализованная система отличается повышенной устойчивостью, благодаря отсутствию зависимости от центрального элемента, и позволяет реализовывать децентрализованные алгоритмы.

В ходе исследования были изучены возможности модулей беспроводной связи XBee, а также была выделена конфигурация, позволяющая настроить систему передачи сообщений в децентрали-

зованной сети мобильных роботов. С использованием полученных знаний было организовано “общение” мобильных роботов при помощи разработанной технологии программирования их бортовых микрокомпьютеров.

Предложенные методы, программная и аппаратная реализация апробированы на группе сверхлегких БПЛА вертикального взлета. Апробация показала надежность и перспективность развития полученной системы, а также эффективность выполнения децентрализованных алгоритмов.

Показано, что контрольная сумма занимает меньше места в памяти, нежели все данные со всех мобильных роботов, и за счет нее можно восстановить данные со всех роботов при отсутствии связи с несколькими членами группы или выходе их из строя. Также показано, что подсчет контрольных сумм может считаться при изменяющейся топологии связей между узлами. Система работает корректно даже, когда в отдельные моменты времени количество имеющихся связей между БПЛА достаточно мало.

Список литературы

[1] *Сечин А.Ю., Дракин М.А., Киселева А.С.* Беспилотный летательный аппарат: применение в целях аэрофотосъемки для картографирования. — М.: “Ракурс”. 2011.

[2] *Howard A., Smith B., Egerstedt M.* Realization of the sensor web concept for Earth science using mobile robotic platforms // Aerospace Conference. 2007. IEEE. PP. 1–6.

[3] *Амелин К.С.* Легкий беспилотный летательный аппарат для автономной группы // Стохастическая оптимизация в информатике. №6. С. 117–126.

[4] *Chao H.* Cooperative Remote Sensing And Actuation Using Networked Unmanned Vehicles. — Diss. Ц Utah State University. 2010.

[5] *Рыжова Т.С.* Система управления коллективом мобильных роботов // Мехатроника, автоматизация, управление. №4. С. 45–50. 2014.

[6] *Vasarhelyi G., Viragh Cs., Somorjai G., Tarcai N., Nepusz T., Vicsek T.* Outdoor flocking and formation flight with autonomous aerial robots // IROS с. 3866–3873. 2014.

[7] *Юревич Е.И.* Управление роботами и робототехническими системами. — СПб.: Изд. СПбГПУ. 2001.

[8] *Vasarhelyi G., Viragh Cs., Abel D., Tarcai N., Nagy M., Vicsek T. Varkonyi P* Patterns, transitions and the role of leaders in the collective dynamics of a simple robotic flock // Journal of Statistical Mechanics: Theory and Experiment. P04010. 2011.

[9] *Каляев И.А., Шеремет И.А.* Военная робототехника: выбор пути // Мехатроника, автоматизация, управление. 2008. №2. С. 32–34.

[10] *Скобелев П.О., Царев А.В.* Сетецентрический подход к созданию больших мультиагентных систем для адаптивного управления ресурсами в реальном времен // Материалы межд. научно - практической мультиконференции “Управление большими системами” Москва, 14–16 ноября с. 263–267. 2011.

[11] *Амелин К.С., Граничин О.Н.* Мультиагентное сетевое управление группой легких БПЛА. — Нейрокомпьютеры: разработка, применение. №6. С. 64–72. 2011.

[12] *Тюшев К.И.* Мультиагентные технологии для построения RAID-подобных распределенных систем хранения данных. — “СПИСОК-2014”. СПб. С. 477–481. 2014.

[13] Digi International [Official website]. `http://www.digi.com/`

[14] *Amelin K., Amelina N., Granichin O., Putov V.* Task distribution algorithm for the cooperating group of light autonomous unmanned aerial vehicles // In: Proc. of the 2nd IFAC Workshop on Research, Education and Development of Unmanned Aerial Systems, RED-UAS, Compiegne, France, November 20-22, 2013. P. 152-155.

[15] *Амелин К.С., Антал Е.И., Васильев В.И., Амелина Н.О.* Адаптивное управление автономной группой беспилотных летательных аппаратов. — Стохастическая оптимизация в информатике. №5. С. 157–166. 2009.

[16] KQML specifications [Electronic resource]. http://www.csee.umbc.edu/csee/research/kqml

[17] FIPA specifications [Electronic resource]. http://www.fipa.org/specs

[18] JADE documentation [Electronic resource]. http://jade.tilab.com/doc/programmersguide.pdf

[19] Raspberry Pi [Official website]. http://www.raspberrypi.org/

[20] RAID specifications [Electronic resource]. http://www.snia.org/tech_activities/standards/curr_standards/ddf

[21] *Amelina N., Fradkov A., Vergados D., Jiang Y.* Approximate Consensus in Stochastic Networks with Application to Load Balancing. — IEEE Transactions on Information Theory. April 2015. Vol. 61. Issue 4. PP. 1739–1752. 2015.