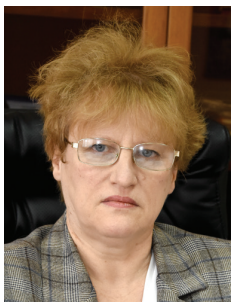
Оригинальная статья
https://doi.org/10.21683/1729-2646-2020-20-2-35-42

Разработка алгоритмов для надежного обмена данными между автономными роботами на основе принципов самоорганизующейся сети

Александр В. Ермаков[1]*, Лариса И. Сучкова[1]
[1]Алтайский государственный технический университет им. И.И. Ползунова, Российская Федерация, Алтайский край, Барнаул
**tour0@ya.ru*

Александр В. Ермаков



Лариса И. Сучкова

Резюме. *Рассмотрены факторы, влияющие на надежность передачи данных в сетях с узлами с периодической доступностью. Приведены принципы передачи данных между роботами, показана необходимость глобальной связности коммуникаций внутри автономной системы, так как отсутствие информации о намерениях других автономных роботов понижает эффективность робототехнической системы в целом и отрицательно влияет на отказоустойчивость в условиях распределения работ в коллективе независимых исполнителей поставленного задания. Показано, что существующие решения задачи обмена данными на основе IP-сетей общего назначения обладают рядом недостатков, поэтому в качестве основы организации сетевого взаимодействия автономных роботов использовались наработки в области топологических моделей систем связи, что позволяет строить самоорганизующиеся сети. Перечислены требования к проектируемой сети для надежной передачи сообщений между автономными роботами, выбран вариант организации надежной доставки сообщений с помощью оверлейных сетей, позволяющих расширить функционал сетей со стабильной конфигурацией. Приведен обзор существующих управляемых и неуправляемых оверлейных сетей, произведена оценка их применимости для коммуникации внутри коллектива автономных роботов. Описаны требования к механизму обмена данными в связи с особенностями и спецификой работы коллектива автономных роботов. Для описания алгоритмов и архитектуры оверлейной самоорганизующейся сети использовались общепринятые методы построения децентрализованных сетей с нулевой конфигурацией. В результате работы были предложены общие принципы функционирования спроектированной сети, описана структура сообщений для алгоритма доставки, произведено выделение служебных маршрутизируемых потоков данных, описаны алгоритмы пересылки сообщений между узлами сети, разработаны алгоритмы сбора и синхронизации глобального статуса сети. Для повышения надежности и отказоустойчивости работы сети предложено хранение глобального статуса сети на каждом из узлов. Описаны принципы функционирования распределенного хранилища данных. Для информирования об изменениях в статусе сети предложено использование отдельного канала управления для внутрисетевых служебных сообщений, не пересекающегося с передаваемыми данными. Разработан алгоритм лавинной маршрутизации для уменьшения задержек и ускорения процесса синхронизации глобального статуса сети и поддержки его консистентности. Предложено использовать hello-протокол для установки и поддержания соседских отношений между узлами сети. Приведены примеры добавления и удаления узлов сети, рассмотрены возможные проблемы масштабируемости разрабатываемой сети и способы их решения. Подтверждены критерии и показатели достижения эффекта самоорганизации отдельных узлов в сеть. Произведено сравнение спроектированной сети с существующими аналогами. Для разработанных алгоритмов приведены примеры расчетных оценок временных задержек доставки сообщений. Указаны теоретические ограничения оверлейной сети при наличии преднамеренных и непреднамеренных дефектов, а также приведен пример восстановления работоспособности сети после сбоя.*

Ключевые слова: *надежность; доставка сообщений; гарантированная доставка данных; оверлейная сеть; автономный робот; групповое взаимодействие; мультиагентная робототехническая система.*

Для цитирования: *Ермаков А.В., Сучкова Л.И. Разработка алгоритмов для надежного обмена данными между автономными роботами на основе принципов самоорганизующейся сети // Надежность. 2020. № 2. С. 35-42. https://doi.org/10.21683/1729-2646-2020-20-2-35-42*

Поступила *21.11.2019 г.* / **После доработки** *14.04.2020 г.* / **К печати** *17.06.2020 г.*

Введение

Успешность выполнения задач коллективом роботов зависит от надежности коммуникаций между членами коллектива, а именно от гарантии доставки сообщения до исполнителя и получения ответа.

Проблему повышения надежности передачи данных в сети с мерцающими узлами рассматривал Лавров Д.Н. в работах [1–2]. Под мерцающим узлом понималось промежуточное устройство, способное передавать сообщения и характеризующееся нестабильностью работы, либо непостоянством присутствия в сети, в том числе, в результате перемещения узла в пространстве.

Концепция узлов с периодической доступностью применима для крупных подвижных объектов – кораблей, самолетов, поездов, робототехнических систем [3]. Существуют определенные наработки в области топологических моделей систем связи, которые также могут быть применимы к самоорганизующимся компьютерным сетям. Главная особенность таких алгоритмов – невозможность гарантировать передачу информации по заданному маршруту в связи с динамической природой сети и меняющейся топологией.

Вопросы взаимодействия между членами коллектива мобильных роботов начали возникать в конце 80-х годов XX века, ранее исследования были сосредоточены на единичных робототехнических системах или на распределенных системах, не связанных с робототехникой [4].

Исследования Юна Ота [5] подтверждают существование класса задач, которые оптимально решаются с применением групповой робототехники, одним из постулатов которой является способность параллельного и независимого выполнения подзадач, сокращающего общее время выполнения поставленной задачи. Любая система, в которой используется множество взаимозаменяемых агентов, позволяет увеличить отказоустойчивость простой заменой вышедшего из строя робота рабочим, однако, создание многофункциональных агентов связано с большими затратами по сравнению с созданием агентов, выполняющих специализированные задачи. Распределенный подход позволяет проектировать специализированных роботов, которые выполняют нагрузку, вызывающую сложности у других агентов [6].

Из работы Майкла Кригера, Жан-Бернара Биллера и Лорана Келлера [7] известно, что при распределении работ по коллективу роботов может наблюдаться снижение эффективности системы в целом. Например, даже если суммарная стоимость мультиагентной системы окажется ниже, чем монолитное решение, управлять такой системой может оказаться затруднительно из-за децентрализованности или отсутствия глобального хранилища данных. Отсутствие информации о намерениях других агентов может привести к ситуации, когда отдельные роботы будут мешать друг другу выполнять поставленные задачи. Чтобы избежать этого, требуется глобальная связность, обеспечивающая надежный обмен данными между автономными роботами для глобального и локального планирования и выполнения локальных задач каждым агентом в последующем.

Обмен данными в постоянно меняющихся внешних условиях является фактором, непосредственно влияющим на стабильность и эффективность работы коллектива роботов. В связи с этим разработка и исследование надежных алгоритмов коммуникации актуальны и являются средством повышения надежности функционирования роботизированной системы в целом. В работе [8] проведены экспериментальные исследования отказоустойчивости, показывающие важность обеспечения надежности связи применительно к коллективу роботов.

Особый интерес представляет исследование алгоритмов коммуникации между автономными роботами, так как от надежности и стабильности их работы зависит скорость принятия решений и возможность согласованной работы коллектива в целом.

В настоящее время для создания связи на коротких расстояниях используются mesh-сети, представляющие собой распределенные самоорганизующиеся сети с ячеистой топологией, разворачиваемые на основе беспроводных сетей Wi-Fi [9].

Вышележащие протоколы, такие, как TCP, гарантируют надежную доставку сообщений по такой сети. Однако в связи с ростом объема коммуникаций в сети Интернет и необходимостью бесперебойной работы сети стало затруднительным внесение новых базовых протоколов и изменений в их структуру для предоставления новых услуг и развертывания новых сервисов [10]. Оверлейные сети позволяют расширить функционал сети, не затрагивая нижележащие базовые протоколы [11], и могут предоставлять следующие услуги: создание отказоустойчивых сетей [12], точки рандеву [13], поиск [14–15], причем эти услуги сложно обеспечивать на уровне IP протокола.

Решение проблемы надежной доставки сообщений до узла в сети существующие популярные оверлейные сети осуществляют различными способами. Одни сети специализируются на анонимности (tor [16], I2P [17]), гарантируя безопасную доставку, другие – на быстром развертывании беспроводной сети Wi-Fi (MANET [18], netsukuku [19]).

Оверлейные сети абстрагируются от нижележащих протоколов, так, например, сеть может использовать различную среду передачи данных в разных сегментах гетерогенной сети. Единственным требованием к сетям, поверх которых работает оверлейная сеть, является наличие маршрута между подсетями. На рисунке 1 приведен пример оверлейной сети, построенной поверх IP-сети.

Авторы работы [10] классифицируют оверлейные сети на две большие категории:

- управляемые сети, где каждому узлу сети известны все узлы сети и их возможности;

- неуправляемые сети, где ни одному из узлов не известна полная топология сети.

Неуправляемые сети, как правило, строятся на основе

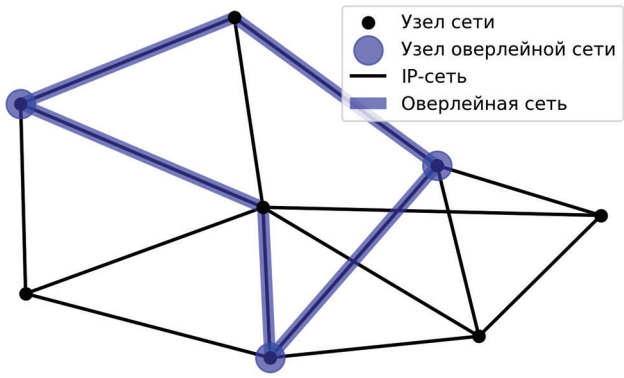


Рис. 1. Пример оверлейной сети, построенной поверх IP-сети

локальных сетей. В противоположность им, управляемые оверлейные сети централизованы или имеют один из механизмов распределенного хранения глобального статуса сети, например, распределенные хэш-таблицы (Distributed Hash Tables, DHT) [20–22].

Сеть tor использует TCP-потоки для связи между узлами сети и луковую маршрутизацию для пересылки сообщений внутри сети. Она не является полностью децентрализованной, так как существуют сервера каталогов, хранящие информацию о состоянии сети [16]. Сеть tor требует обязательного наличия Интернета.

У прочих пиринговых сетей или сетей на основе DHT отсутствует функционал пересылки сообщений другим узлам, и для работы они также требуют наличия сети Интернет.

Таким образом, в настоящее время отсутствуют сетевые решения для обеспечения надежной коммуникации в коллективе автономных роботов.

При разработке алгоритмов для надежной передачи сообщений между автономными роботами к самоорганизующейся сети предъявляются следующие требования:

- отсутствие ручной настройки узлов;
- клиент сети должен быть прост в реализации и установке (в том числе не требовать патчей ядра или определенную версию операционной системы);
- сеть должна работать на уровне пользователя без всяких специфических привилегий;
- сеть должна работать поверх стандартных протоколов TCP и/или UDP.

Рассмотренные выше существующие оверлейные сети не удовлетворяют перечисленным требованиям, поэтому была поставлена задача разработать алгоритмы для надежного обмена данными с использованием оверлейной сети.

Структура самоорганизуюшейся сети

Для обмена данными автономными роботами предлагается использовать оверлейную сеть, в которой обмен информацией осуществляется на прикладном уровне по модели OSI поверх стандартных протоколов TCP и UDP.

Клиентское вычислительное устройство, в том числе, бортовой компьютер автономного робота, для подключения к оверлейной сети стартует программное обеспечение, которое устанавливает соединения с другими узлами сети и осуществляет пересылку данных между промежуточными узлами. Каждый узел сети в каждый момент времени поддерживает несколько соединений, обеспечивая резервирование каналов передачи данных.

Структура сообщения для алгоритма доставки

Сообщение является минимальной единицей данных, которые передаются внутри оверлейной сети. UDP-пакеты используются для анонсирования изменений сети и низкоприоритетных уведомлений, принимаются и разбираются целиком, что сокращает время обработки. TCP требует более сложного алгоритма обработки. Однако, благодаря фиксированному размеру заголовка сообщения и наличию поля длины данных, становится возможным разбор входящего TCP-потока на отдельные сообщения.

Сообщение описывается формальной структурой: {IDsrc; IDdest; CMD; LEN; P}, где:

IDsrc – идентификатор узла-отправителя (8 байтов);

IDdest – идентификатор узла назначения (8 байтов);

CMD – тип сообщения (1 байт);

LEN – длина поле данных (беззнаковое целое, 2 байта);

P – поле данных длиной LEN байт, закодированное протоколом protobuf2 [23].

Таким образом, сообщение состоит из заголовка длиной 19 байт и поля данных переменной длины.

Заголовок ячейки сети {IDsrc; IDdest; CMD; LEN} содержит идентификатор отправителя и идентификатор получателя. Идентификаторы узлов являются 64-битными числами, которые состоят из пары IP-адресов: {IPiface; IPext}, где:

IPiface – IP сетевого интерфейса;

IPext – внешний IP-адрес.

Мы считаем такой способ задания идентификаторов достаточным для наших целей. Он позволяет узлам самостоятельно генерировать себе уникальные идентификаторы, сводя вероятность коллизий к нулю.

В зависимости от типа сообщения CMD происходит выбор обработчика сообщения. Ячейки классифицируются на две группы: управляющие и передающие. Управляющие ячейки обрабатываются узлами-получателями. Например, это могут быть команды проверки доступности узла, запросы и ответы на изменения статуса сети (см. рисунок 2). Передающие ячейки содержат в себе данные, которые нужно обработать, если идентификатор получателя совпадает с текущим узлом, либо переслать дальше по сети.

Предложенная авторами технология обработки сообщений была реализована программно [24].

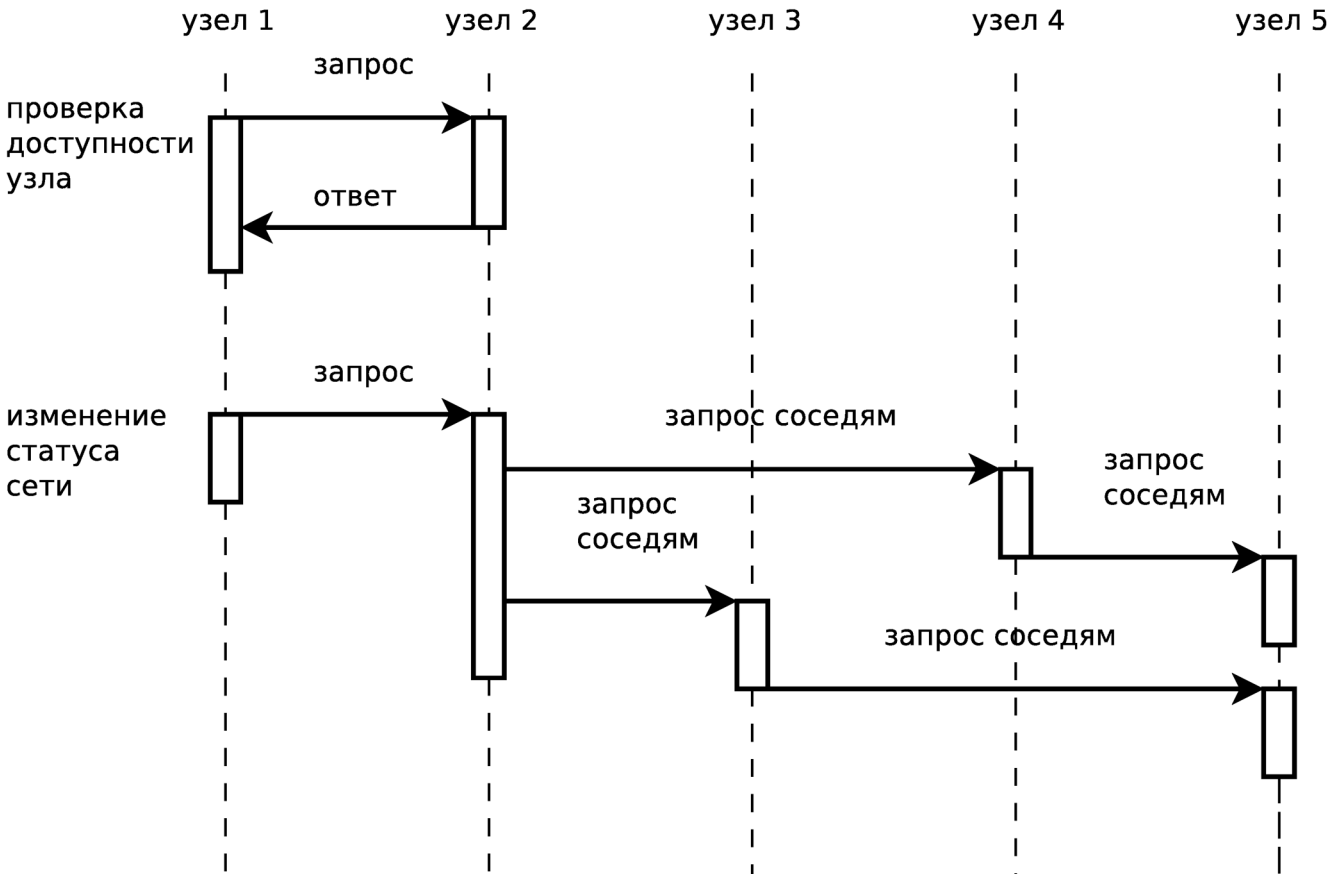


Рис. 2. Диаграмма последовательности обработки принятого сообщения узлами-соседями на примерах проверки доступности узла-соседа и информировании изменений в статусе сети

Алгоритмы сбора и синхронизация статуса сети

Сеть обмена данными между автономными роботами является управляемой, то есть для нее существует глобально обновляемый статус, содержащий сведения обо всех узлах сети. Информация о статусе сети хранится в полном объеме на каждом узле, таким образом, дублирование информации увеличивает надежность сети и ее отказоустойчивость в целом.

При добавлении нового узла для обнаружения других узлов сети в локальной сети используется широковещательная рассылка, при успешном обнаружении устанавливаются соединения с соседними узлами и синхронизируется статус сети. На этом этапе осуществляется обмен сообщениями одновременно по всем каналам («лавинная» рассылка) [1].

Для информирования при изменении статуса сети, осуществляемом, например, при добавлении нового узла, используется лавинная маршрутизация, когда узел сети пересылает полученные пакеты по всем своим непосредственным соседям, за исключением того узла, с которого он был получен. Такой подход увеличивает надежность передачи служебной информации и увеличивает вероятность получения сообщения всеми узлами сети. Проблема дублирования сообщений решается путем кеширования принятых сообщений и запрета повторной отправки сообщения.

На рисунке 3 приведена схема работы системы. Узел принимает сообщение из сети, в зависимости от значения типа сообщения CMD запускается алгоритм лавинной маршрутизации. Принятый пакет проверяется в буфере кеша пакетов, располагающегося в оперативной памяти. Если пакет был найден в кеше, то есть этот пакет был принят ранее, то алгоритм завершает работу, отбрасывая пакет и не обрабатывая его. В противном случае пакет добавляется в кеш, вытесняя из кеша самые старые записи. Далее поле данных P из пакета дешифруется и применяется к собранному глобальному статусу сети. После фиксации изменений пересчитывается хеш статуса сети. На этом локальные изменения завершаются, далее происходит информирование узлов-соседей путем массовой рассылки принятого сообщения. Формируется актуальный список узлов-соседей, а тем узлам, от которых недавно приходил HELLO-пакет, осуществляется немедленная отправка пакета по протоколу UDP, для остальных узлов сети сформированный пакет отправляется в очередь для дальнейшей асинхронной отправки.

Помимо лавинной маршрутизации, поддержание консистентности данных о статусе сети на всех узлах осуществляется путем периодической пересылки соседям хеша от списка известных идентификаторов узлов сети. При несовпадении хеша запускается процесс синхронизации между соседями. Наличие возможности получения данных о статусе сети от соседа позволяет

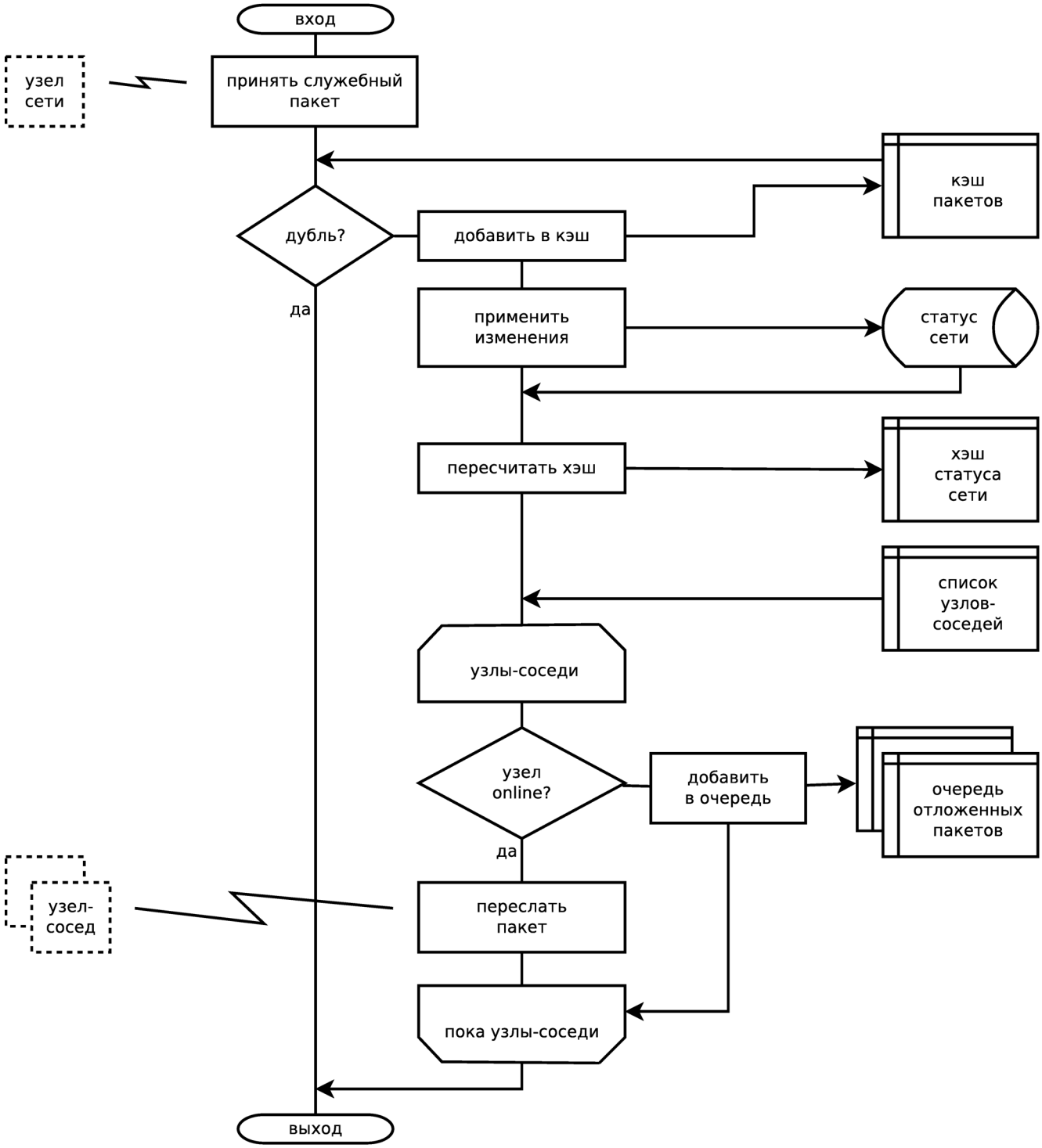


Рис. 3. Схема работы системы для алгоритма лавинной маршрутизации

ускорить добавление нового узла сети и не нагружать сеть пересылкой множества служебных сообщений [25].

Каждый узел рассылает соседям HELLO-пакеты, информируя о своей доступности. Перед завершением работы клиента сети, он рассылает в сеть уведомления об отключении. При разрыве соединения между узлами сети или при истечении таймаута ожидания HELLO-пакета от соседа узел делает несколько попыток соединиться с потерянным узлом. Если узел оказывается недоступен, то генерируется сообщение об удалении идентификатора из статуса сети. Таким образом, детектируются неисправности в сети и становится возможным оперативно реагировать на них [26].

Надежность является сложным физическим свойством, поэтому не существует одного обобщенного критерия и показателя, который бы достаточно полно характеризовал надежность техники. Только семейство критериев позволяет оценить надежность сложной

технической системы. Выбор критериев зависит от типа технического объекта, его назначения и требуемой полноты оценки надежности [27].

Одним из критериев надежности проектируемой оверлейной сети являются временные задержки. Предполагается, что оверлейная сеть должна обеспечивать надежность доставки сообщений при временной недоступности связи между узлами-соседями, в том числе из-за преднамеренных или непреднамеренных дефектов. Кроме того, необходимо учесть специфику применения предлагаемой сети для автономных робототехнических комплексов, когда приоритетом является мгновенная доставка сообщения, причем временный сбой при доставке предпочтительнее, чем получение сообщения с длительной задержкой (в некоторых задачах от 500 мс). Другой особенностью является независимость сообщений друг от друга. В предложенной сети не важен порядок доставки сообщений, что позволяет нам оптимизировать алгоритмы и протокол доставки по этому критерию.

Экспериментальное тестирование разработанных алгоритмов обмена данными

Для проведения эксперимента была создана тестовая сеть, состоящая из маршрутизатора Cisco Catalyst 2960 и шести компьютеров, работающих под управлением ОС Ubuntu 18.04. Для эмуляции нескольких сетей на коммутаторе было сконфигурировано пять VLAN, в VLAN 0 разместились два компьютера, в остальных – по одному. Правилами маршрутизации был запрещен прямой обмен IP-пакетами между всеми подсетями, за исключением VLAN 0. В результате эксперимента было получено подтверждение самоорганизации сети, а также исследована работоспособность разработанных алгоритмов обмена данными.

Было произведено экспериментальное тестирование существующих сетевых протоколов TCP и UDP на действующей тестовой сети. Для этого осуществлялась пересылка данных между двумя маршрутизируемыми узлами сети. Потери пакетов эмулировались на сетевом интерфейсе узла правилом в iptables и модулем statistic, позволяющим осуществлять отбор части пакетов по условию. Для TCP открывалось одно соединение, внутри которого пересылались ячейки оверлейной сети. В UDP отсутствует механизм подтверждения доставки, поэтому принятие каждого пакета подтверждалось принимающей стороной. Если подтверждение не пришло по истечении таймаута, то пакет посылался повторно.

Рисунок 4 демонстрирует полученные в результате экспериментального исследования временные задержки на доставку каждого пакета с использованием стандартных протоколов TCP и UDP при запланированных потерях 0%, 5% и 10% пакетов.

При отсутствии потерь UDP показал минимальные задержки во время передачи данных, однако, даже при

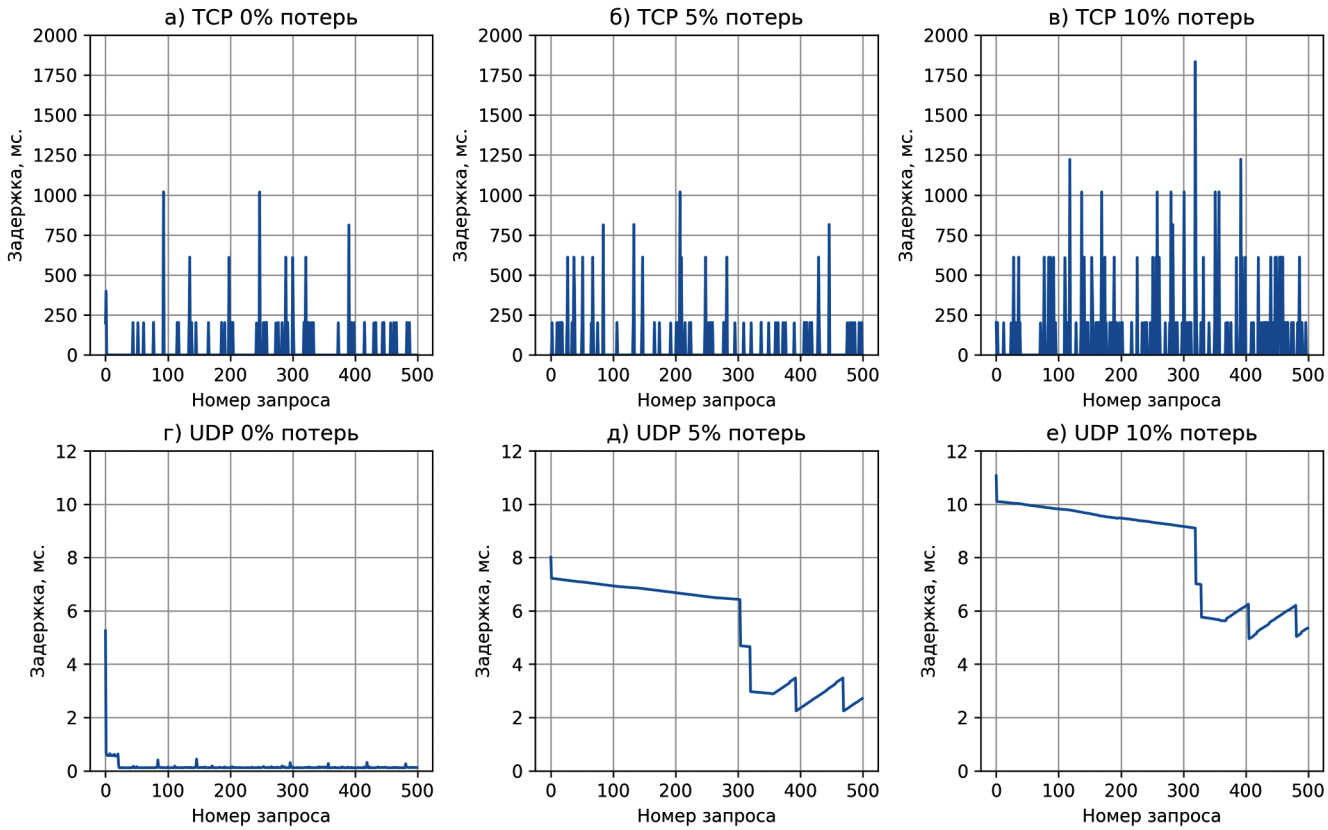


Рис. 4. Сравнение временных задержек доставки пакета стандартными протоколами при наличии потерь пакетов в сети: а) TCP, 0% потерь; б) TCP, 5% потерь; в) TCP, 10% потерь; г) UDP, 0% потерь; д) UDP, 5% потерь; е) UDP, 10% потерь

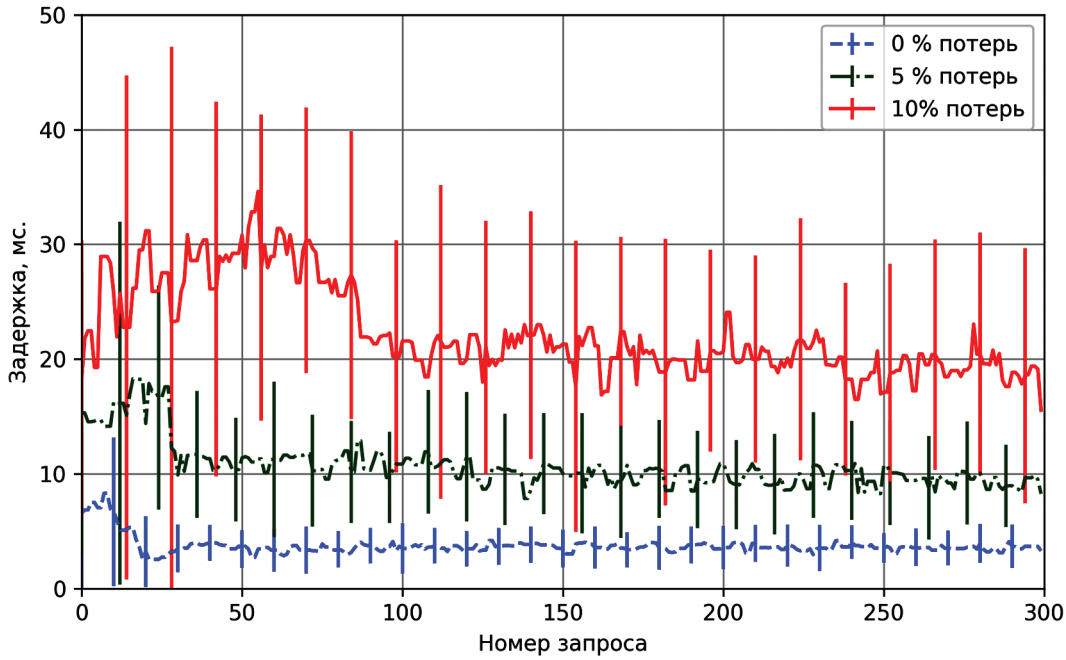


Рис. 5. Сравнение временных задержек доставки пакета внутри оверлейной сети при наличии дефектов сети

минимальных потерях увеличивается задержка и количество повторно отправляемых данных. Спустя 300-400 отправленных пакетов задержка стабилизируется на одном уровне (рисунок 4, д и 4, е).

При использовании TCP отмечается долгое установление соединения (вплоть до секунды в некоторых случаях) и установка повторных соединений при разрыве. Столь долгие единичные задержки недопустимы для применения в сети для автономных робототехнических комплексов.

Зная результаты исследования временных задержек при использовании существующих протоколов обмена данными, стало возможным проводить оценку надежности разработанных алгоритмов. Тестирование оверлейной сети происходило в тех же условиях.

Предложенный нами алгоритм обмена данными характеризуется меньшими задержками после выхода системы на этап нормальной эксплуатации и показывает более высокую надежность за счет обеспечения мгновенной доставки и сведению к минимуму отказов, которые возникли бы при условии несвоевременной доставки сообщений. Применение 0-RTT handshake (установление соединения с нулевой задержкой) обеспечило требуемую производительность оверлейной сети.

Стабильность решения была проверена эксплуатацией в течение месяца ежедневного запуска сети, при этом деградации производительности или увеличения задержек доставки сообщений не замечено. Итоговые результаты эксперимента приведены на рисунке 5.

Заключение

Авторами данной статьи были разработаны алгоритмы функционирования оверлейной сети с учетом специфики ее использования автономными роботами. Предполагаемый подход позволит обеспечить надежный обмен данными внутри автономной системы, тем самым достигая эффекта коллективного выполнения задач с распределением ролей и подцелей, что было бы неосуществимо при отсутствии межагентного взаимодействия и обмена текущей информацией.

Данные алгоритмы явились основой для построения тестовой программной системы, предназначенной для исследования процесса обмена данными в коллективе автономных роботов.

Библиографический список

1. Лавров Д.Н. Принципы построения протокола гарантированной доставки сообщений // Математические структуры и моделирование. 2018. № 4(48). С. 139–146. DOI: 10.25513/2222-8772.2018.4.139-146

2 Гусс С.В., Лавров Д.Н. Подходы к реализации сетевого протокола обеспечения гарантированной доставки при мультимаршрутной передаче данных // Математические структуры и моделирование. 2018. № 2(46). С. 95–101. DOI: 10.25513/2222-8772.2018.2.95-101

3. Сорокин А.А., Дмитриев В.Н. Описание систем связи с динамической топологией сети при помощи модели «мерцающего» графа // Вестник Астраханского государственного технического университета. Серия: Управление, вычислительная техника и информатика. 2009. № 2. С. 134–139.

4. Parker L.E. Distributed Intelligence: Overview of the Field and its Application in Multi-Robot Systems // AAAI Fall Symposium: Technical Report, FS-07-06. 2008. P. 5–14. DOI: 10.14198/JoPha.2008.2.1.02

5 Ota J. Multi-agent robot systems as distributed autonomous systems //Advanced engineering informatics. 2006. Vol. 20. No. 1. P. 59-70. DOI: 10.1016/j.aei.2005.06.002.

6. Arai T. et al. Advances in multi-robot systems // IEEE Transactions on robotics and automation. 2002. Vol. 18. No. 5. P. 655–661.

7. Krieger M. J. B., Billeter J. B., Keller L. Ant-like task allocation and recruitment in cooperative robots // Nature. 2000. Vol. 406. No. 6799. P. 992–995. DOI: 10.1038/35023164.

8. Winfield A. F. T., Nembrini J. Safety in numbers: Fault tolerance in robot swarms // International Journal on Modelling Identification and Control. 2006. Vol. 1. P. 30–37. DOI: 10.1504/IJMIC.2006.008645

9. Bicket J. et al. Architecture and evaluation of an unplanned 802.11 b mesh network // Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM. ACM, 2005. P. 31–42. DOI: 10.1145/1080829.1080833.

10. Srinivasan S. Design and use of managed overlay networks: дис. Georgia Institute of Technology, 2007.

11. Clark D. et al. Overlay Networks and the Future of the Internet // Communications and Strategies. 2006. Vol. 63. P. 109.

12. Benson K.E. et al. Resilient overlays for IoT-based community infrastructure communications // 2016 IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI). 2016. P. 152–163. DOI: 10.1109/IoTDI.2015.40

13. Stoica I. et al. Internet indirection infrastructure // ACM SIGCOMM Computer Communication Review. ACM, 2002. Vol. 32. No. 4. P. 73–86.

14. Ripeanu M. Peer-to-peer architecture case study: Gnutella network // Proceedings first international conference on peer-to-peer computing. IEEE, 2001. P. 99–100. DOI: 10.1109/P2P.2001.990433

15. Leibowitz N., Ripeanu M., Wierzbicki A. Deconstructing the kazaa network // Proceedings the Third IEEE Workshop on Internet Applications. WIAPP, 2003. IEEE, 2003. P. 112–120. DOI: 10.1109/WIAPP.2003.1210295

16. Dingledine R., Mathewson N., Syverson P. Tor: The second-generation onion router. Naval Research Lab Washington DC, 2004.

17. Herrmann M., Grothoff C. Privacy-implications of performance-based peer selection by onion-routers: a real-world case study using I2P // International Symposium on Privacy Enhancing Technologies Symposium. Springer, Berlin, Heidelberg, 2011. C. 155–174. DOI: 10.1007/978-3-642-22263-4_9

18. Tandon N., Patel N. K. An Efficient Implementation of Multichannel Transceiver for Manet Multinet Environment // 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT). IEEE, 2019. P. 1–6. DOI: 10.1109/ICCCNT45670.2019.8944505

19. Кудряшова Э.Е., Вовченко А.В., Олейников Р.А. Исследование сети NETSUKUKU на основе фрактальных множеств // Вестник Международной академии системных исследований. Информатика, экология, экономика. 2008. Т. 11. № 1. С. 55–57.

20. Kumar A. et al. Ulysses: a robust, low-diameter, low-latency peer-to-peer network // European transactions on telecommunications. 2004. Vol. 15. No. 6. P. 571–587. DOI: 10.1002/ett.1013

21. Rowstron A., Druschel P. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems // IFIP/ACM International Conference on Distributed Systems Platforms and Open Distributed Processing. Springer, Berlin, Heidelberg, 2001.

22. Stoica I. et al. Chord: A scalable peer-to-peer lookup service for internet applications // ACM SIGCOMM Computer Communication Review. 2001. Vol. 31. No. 4. P. 149–160.

23. Feng J., Li J. Google protocol buffers research and application in online game // IEEE conference anthology. IEEE, 2013. P. 1–4. DOI: 10.1109/ANTHOLOGY.2013.6784954

24. Ермаков А.В., Сучкова Л.И. Реализация протокола передачи данных между интеллектуальными автономными роботами. Свидетельство о государственной регистрации программы для ЭВМ № 2019666759 от 13 декабря 2019 г.

25. Ермаков А. В., Сучкова Л. И. Проектирование сетевой коммуникационной среды для реализации управления в коллективе автономных роботов // Южно-Сибирский научный вестник. 2019. Т. 2. № 4 С. 28–31. DOI: 10.25699/SSSB.2019.28.48969

26. Ermakov A., Suchkova L. Development of Data Exchange Technology for Autonomous Robots Using a Self-Organizing Overlay Network // 2019 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon). IEEE, 2019. P. 1–5. DOI: 10.1109/FarEastCon.2019.8934727

27. Половко А. М., Гуров С. В. Основы теории надежности: 2-е изд., перераб. и доп. СПб.: БХВ-Петербург, 2006. 702 с.: ил. ISBN 5-94157-541-6

Сведения об авторах

Александр В. Ермаков – аспирант кафедры ИВТиИБ АлтГТУ, 656038, Российская Федерация, Алтайский край, г. Барнаул, пр. Ленина, 46, e-mail: tour0@ya.ru

Лариса И. Сучкова – доктор технических наук, проректор по учебной работе АлтГТУ, 656038, Российская Федерация, Алтайский край, г. Барнаул, пр. Ленина, 46, e-mail: li.suchkova@yandex.ru

Вклад в работу

Ермаков А.В. – обзор литературы, разработка алгоритмов.

Сучкова Л.И. – формализация требований и постановка задачи.