

12. *Guerci J.R.* Space-time adaptive processing for radar. Boston and London: Artech House, 2015, 298 p.
13. *Hansen R.C.* A One-Parameter Circular Aperture Distributions with Narrow Beam-width and Low Sidelobes, *IEEE trans.*, 1967, Vol. AP-24, pp. 477-480.
14. *Hansen R.C.* Microwave scanning antennas. Vol. 1. Apertures. New York and London, Academic press, 1964. – 536 p.
15. *Hansen R. C.* Microwave scanning antennas. Vol. 2. Array theory and practice. New York and London, Academic press, 1966, 496 p.
16. *Hansen R.C.* Phased array antennas. 2nd ed. Wiley, 2009, 558 p.
17. *Marple Jr.S.L.* Digital spectral analysis with application. 2nd ed. Dover Publications Inc., 2019, 432 p.
18. *Melvin W.L.* Space-time adaptive processing and adaptive arrays: special collection of papers, *IEEE Trans. Aerospace and Electronic Systems*, 2000, Vol. 36, No. 2, pp. 508-509.
19. *Monzingo R.A.* Introduction to Adaptive Arrays. New York, John Wiley and sons, 1980, 446 p.
20. *Taylor T.T.* One-Parameter Family of Line Sources Producing Modified $\sin \pi u / \pi i$ Pattern, rep. TM 324. Hughes Aircraft Co., Culver City, CA, 1953.

Статью рекомендовал к опубликованию д.т.н. А.Л. Приоров.

Лаврентьев Александр Михайлович – Ярославское высшее военное училище противовоздушной обороны; e-mail: alexalawrentew24@icloud.com; г. Ярославль, Россия; тел.: 89108293521; д.т.н.; профессор; начальник кафедры радиотехнических систем.

Калашников Роман Васильевич – e-mail: karomm12@mail.ru; тел.: 89201023588; к.т.н.; старший преподаватель кафедры радиотехнических систем.

Бабушкин Евгений Александрович – e-mail: bebs37@mail.ru; тел.: 89056306556; адъюнкт.

Lavrentyev Aleksandr Mihailovich – Yaroslavl higher military college of air defense; e-mail: alexalawrentew24@icloud.com; Yaroslavl, Russia; phone: +79108293521; dr. of eng. sc.; professor; head of the department of radio engineering systems.

Kalashnikov Roman Vasil'evich – e-mail: karomm12@mail.ru; phone: +79201023588; cand. of eng. sc.; senior lecturer of radio engineering systems.

Babushkin Eugeniy Alexandrovich – e-mail: bebs37@mail.ru; phone: 89056306556; adjunct.

УДК 519.714

DOI 10.18522/2311-3103-2024-1-202-213

К.Г. Кебкал, А.А. Кабанов, В.В. Альчаков, В.А. Крамарь, М.Э. Димин

ПРОГРАММНЫЕ ПОДХОДЫ ДЛЯ РЕШЕНИЯ ЗАДАЧ ГИДРОАКУСТИЧЕСКОЙ СВЯЗИ В СИСТЕМАХ МОРСКОГО ИНТЕРНЕТА ВЕЩЕЙ *

При одновременной работе нескольких гидроакустических модемов в районе взаимного покрытия могут возникать коллизии пакетов данных, поступающих на прием от нескольких источников, что приводит к потерям части или всей информации. С ростом числа одновременно работающих гидроакустических модемов алгоритмы физического уровня не обеспечивают стабильную передачу данных и вероятность возникновения коллизий повышается, что делает работу модемов неэффективной или даже невозможной. Для обеспечения эффективной работы в условиях гидроакустической среды распространения сигнала и для уменьшения или исключения коллизий при обмене и доставке данных между двумя модемами, не обладающими возможностью синхронной работы, а также для уменьшения времени доступа к среде распространения сигнала требуются методы уровня управления доступом к среде с применением протоколов канального уровня. Обычно, такая задача решается при помощи кодового разделения гидроакустических

* Работа частично выполнена при поддержке Минобрнауки России, проект FEFM-2024-0015.

каналов. Модемы общаются как бы на разных частотах, что не создаёт коллизий, это позволяет общаться абонентам подводной сети в формате «точка-точка», либо в режиме «multicast», то есть всем отдельно, однако, в случае, если надо сделать передачу по сети, такой вариант уже не подойдёт, так как сетевая передача, предполагает работу на основе «broadcast» сообщений. При практическом использовании указанные протоколы удобно поместить в состав программной среды разработки (фреймворк) конкретных пользовательских приложений для решения задач сетевой г/а связи. Такой фреймворк принято называть программным каркасом, он позволяет выполнять пользовательскую модификацию имеющихся в составе каркаса сетевых алгоритмов, а также включение силами пользователя новых алгоритмов сетевой гидроакустической связи. Для построения прогнозирующей модели в работе использовались протоколы DACAP, T-Lohi, Flooding и ICRP. Реализация алгоритмов выполнена на языке Erlang. В работе приведены алгоритмы реализации указанных протоколов. Приводится сравнительный анализ сетевой работы с использованием протоколов и без них. Оценена эффективность и скорость работы. Даны рекомендации по дальнейшей разработке программного каркаса.

Морской интернет вещей; гидроакустическая связь; сетевые протоколы; программный каркас.

K.G. Kebkal, A.A. Kabanov, V.V. Alchakov, V.A. Kramar, M.E. Dimin

THE SOFTWARE APPROACHES FOR SOLVING HYDROACOUSTIC COMMUNICATION PROBLEMS IN MARINE INTERNET OF THINGS SYSTEMS

When several hydroacoustic modems operate simultaneously in an area of mutual coverage, collisions of data packets received from several sources may occur, which leads to the loss of some or all information. With the increase in the number of simultaneously operating hydroacoustic modems, physical layer algorithms do not provide stable data transmission and the likelihood of collisions increases, which makes the operation of modems ineffective or even impossible. To ensure effective operation in a hydroacoustic signal propagation environment and to reduce or eliminate collisions during the exchange and delivery of data between two modems that do not have the ability to operate synchronously, as well as to reduce the access time to the signal propagation environment, methods of the medium access control layer are required using link layer protocols. Typically, this problem is solved using code separation of hydroacoustic channels. Modems communicate as if at different frequencies, which does not create collisions, this allows subscribers of the underwater network to communicate in a point-to-point format, or in multicast mode, that is, everyone separately, however, in case it is necessary to make a transmission over the network, this option is no longer suitable, since network transmission involves working on the basis of "broadcast" messages. In practical use, it is convenient to place these protocols into a software development environment (framework) for specific user applications for solving network communication problems. Such a framework is usually called a software framework; it allows for user modification of the network algorithms available in the framework, as well as the inclusion of new network hydroacoustic communication algorithms by the user. To build a predictive model, the DACAP, T-Lohi, Flooding and ICRP protocols were used in the work. The algorithms were implemented in Erlang. The paper presents algorithms for implementing these protocols. A comparative analysis of network operation with and without protocols is provided. Efficiency and speed of work were assessed. Recommendations for further development of the software framework are given.

Marine Internet of Things; hydroacoustic communications; network protocols; software framework.

Введение. В области Морского интернета вещей (МИoT) накопился задел технических решений, который способен обеспечить быстрое достижение практических результатов на пути создания подводных гидроакустических (г/а) сетей [1]. Совместно работающие подводные агенты, такие как автономные и телеуправляемые подводные аппараты (АНПА/ТНПА), подводные датчики, буи и т.д., требуют постоянного обмена данными [2–5]. Это же относится и к устройствам, действующим в группе, учитывая изменяющуюся геометрию формируемой ими сети из-за изменений положения внутри неё [6]. МИoT предлагает различные варианты беспроводной связи и формирования сетей на основе г/а, оптической, радиочастотной

связи и связи на основе магнитного синтеза. При этом, г/а связь является наиболее подходящей для подводной среды из-за ее возможностей передачи данных на большие расстояния, от нескольких сотен метров до десятков километров [1, 7–9], имея при этом ряд ограничений [10, 11].

Решение задачи передачи данных под водой с помощью г/а канала предполагает включение г/а модемов в состав сенсорных модулей подводных устройств [12, 13]. Для воздушного пространства существует множество методов для маршрутизации и разрешения коллизий данных, но в гидроакустике они могут применяться не всегда и, как правило, с большими ограничениями.

Для передачи данных в г/а сети, характеризующейся постоянно изменяющимися условиями излучения и приема сигналов, в последнее время получили развитие специализированные сетевые методы, способные учитывать случайным образом меняющееся число узлов сети, находящихся в области взаимной или односторонней "слышимости". Важным направлением в создании сетевых методов связи является развитие алгоритмов формирования ad-hoc сетей [14–16].

Для работы множества устройств в одной зоне покрытия и разрешения коллизий необходимо использование специализированных сетевых протоколов [17–22]. При практическом использовании указанные протоколы удобно поместить в состав программной среды разработки конкретных пользовательских приложений, которую принято называть программным каркасом. Он позволяет выполнять пользовательскую модификацию имеющихся в составе каркаса сетевых алгоритмов, а также включение силами пользователя новых алгоритмов сетевой г/а связи.

Для использования в подводных системах связи на сегодня известны четыре развитых программных каркаса, способных интегрировать разнородные устройства связи и превратить их в сетевые устройства, а также обеспечить возможность гибкой интеграции модемов с различными сенсорными устройствами, исполнительными механизмами, манипуляторами, датчиками или системами датчиков. Эти программные каркасы существуют в открытом доступе: 1) SUNSET (университет "Сапиенца", Италия), 2) DESERT (университет Падуя, Италия), 3) UNetStack (национальный университет Сингапура), и 4) EviNS (компания Evologics, Германия).

Наименее требовательным к вычислительным ресурсам является программный каркас EviNS (Evologics intelligent Networking Software Framework). Он представляет собой компактное ПО с открытым исходным кодом, разработанным на языке программирования Erlang. К преимуществам относится то, что ряд базовых программ, реализующих протоколы управления доступом к среде и протоколы маршрутизации, уже включены в состав программного каркаса.

В качестве образцов модемов, над которыми надстраивается программный каркас EviNS, в данной работе были выбраны два г/а модема: uWave («Лаборатория подводной связи и навигации», Россия) и г/а модем серии S2C (Evologics, Германия) (рис. 1).



Рис. 1. Образцы: а – г/а модем Evologics S2C; б – г/а модемы UC&NL uWave

Для использования обширного задела (встроенных возможностей) программного каркаса EviNS были разработаны пользовательские приложения в виде протоколов уровня управления доступом к среде и протоколов сетевого уровня, кото-

рые оснащались возможностями работы с разными ролями, которые, в свою очередь обеспечивали техническую возможность подключения к программному каркасу EviNS модемов разных производителей (а именно, путем разработки набора ролей, способных выполнять синтаксический разбор наборов команд этих модемов). В частности, для включения г/а модема uWave в состав программного каркаса был разработан синтаксический анализатор команд модема uWave.

В статье для разработки пользовательского программного обеспечения, предназначенного для решения задач сетевой г/а связи в системах MIoT, в качестве программного инструмента предложено использование программного каркаса EviNS, который может быть развернут на стороннем вычислителе или прямо на вычислителе г/а модема и исполняться как часть стандартного ПО, превращая модем из простого устройства симплексной цифровой связи в полноценный узел сетевой связи.

Универсальность и концепция построения программного каркаса EviNS. Программный каркас EviNS обладает универсальностью за счет использования языка программирования Erlang, который обеспечивает возможность кроссплатформенной компиляции разработанного программного каркаса. Это позволяет использовать EviNS на различных вычислительных платформах, включая как настольные ПК, так и встраиваемые системы с невысокой производительностью. Кроме того, универсальность взаимодействия пользовательского приложения с программным каркасом EviNS обеспечивается открытым списком команд в сетевом уровне, позволяя разработчикам разрабатывать собственные команды и макросы для изменения и расширения функциональности программного каркаса. В целом, программный каркас EviNS можно рассматривать, как универсальное средство для разработки пользователем своих собственных приложений, который могут исполняться на вычислительной платформе, выбранной им из широкого круга моделей.

Распространение EviNS осуществляется по лицензиям GPL/MIT, что позволяет свободное использование и распространение каркаса. Исходный код программного каркаса EviNS открыт и расположен на веб-сервисе GitHub (github.com/okebkal/evins). В его состав может также включаться множество пользовательских библиотек с закрытым исходным кодом, запрещая таким образом свободное распространение пользовательской (специализированной) версии программного каркаса.

На рис. 2 представлена базовая концепция программного каркаса EviNS – модель структурирования процессов, основанная на взаимодействии рабочих окружений и супервизоров (<https://github.com/EvoLogics/evins/wiki/arch>). В рабочее окружение входят рабочие процессы, выполняющие полезные вычисления. Это могут быть агенты, интерфейсные обработчики и конфигураторы.

Супервизоры представляют собой процессы, которые следят за функционированием рабочего окружения или за работой других супервизоров. Супервизор может перезапускать процессы рабочего окружения при обнаружении в них сбоев. Как представлено на рис. 2 конечный автомат (КА) - супервизор верхнего уровня (КА-СВУ) отслеживает работу КА конфигулятора программного каркаса (КА-КПК) и КА супервизора агентов (КА-СА). В случае сбоя КА-КПК или одного из КА-СА супервизор верхнего уровня КА-СВУ перезапускает соответствующий процесс, не затрагивая другие запущенные процессы. Каждый КА-СА настраивается и запускается конфигуратором КА-КПК. В соответствии со своей конфигурацией каждый КА-СА запускает и контролирует определенные ему модули, реализующие функции КА рабочего окружения (КА-РО), роли рабочего окружения (РРО) и других КА, в том числе являющихся частью Erlang/OPT. Общий подход в реализации программного каркаса EviNS состоит в разделении кода процесса на общую (функциональные модули) и специальную части (модули обратного вызова).

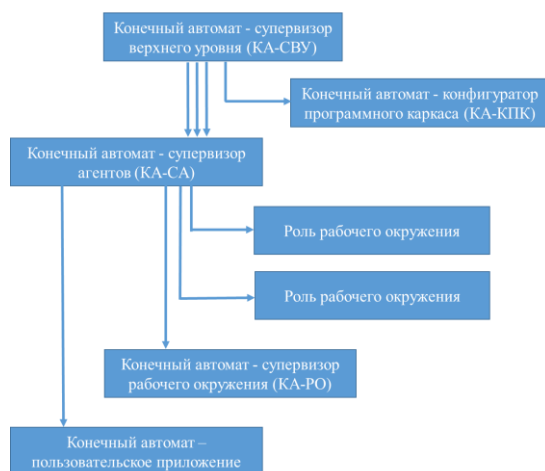


Рис. 2. Модель структурирования процессов – взаимодействие рабочих процессов и супервизоров

Функциональные возможности РРО обеспечивают связь через интерфейсы, поддерживаемые программным каркасом. Интерфейсные обработчики являются реализациями ролей рабочего окружения. Для реализации новых функциональных возможностей РРО необходимо назначение обратных вызовов, отвечающих за синтаксический разбор и за восстановление данных, инициализацию обратного вызова, управление конфигуратором обратного вызова, и остановку обратного вызова при завершении процесса. КА-КПК может контролировать несколько интерфейсных обработчиков в соответствии с конфигурациями агентов. КА-РО поддерживает инициализацию агента и его взаимосвязь с интерфейсными обработчиками. КА обеспечивает запуск в работу таких агентов, как пользовательский конечный автомат или автомат с магазинной памятью.

Агенты, каждый из которых явно определен в виде конечного автомата или автомата с магазинной памятью, управляются событиями, генерируемыми извне интерфейсными обработчиками или внутри обработчиками событий или таймерами. Они изолированы друг от друга для защиты от сбоев, гарантируя, что сбой одного агента не повлияет на работу другого. Работа каждого агента контролируется супервизором, который в случае ошибки перезапускает его. Агенты, как автономные программные компоненты, обладают четко определенными функциями и интерфейсами для обмена сообщениями. Интерфейсные обработчики связывают агентов с внешними процессами, а конфигурация агентов определяется конфигуратором или агентом конфигурирования.

В программном каркасе EviNS используются четыре типа интерфейсов (рис. 3): 1. Сокет TCP (может использоваться как для взаимодействия агента с агентом, так и для взаимодействия агента с внешним приложением); 2. Порт Erlang (обеспечивает основной механизм связи с внешним миром, можно использовать только для взаимодействия агентов с внешними приложениями). 3. Очередь сообщений Erlang (используется для прямого обмена сообщениями между агентами без синтаксического разбора данных интерфейсными обработчиками); 4. HTTP-сервер Cowboy. Это небольшой, быстрый и модульный сервер HTTP, написанный на Erlang. Преобразует действия пользователя в веб-браузере в сообщения, доставляемые через интерфейсные обработчики агентам, и формирует необходимые уведомления на запросы пользователя.

На этом стоит остановиться подробнее. На данный момент EviNS включает в свой состав роль АТ-команд, которая выполняет функцию синтаксического анализатора только для команд г/а модема компании Evologics. В то же время существует большое число производителей г/а модемов, которые не имеют прямой возможности использовать протоколы EviNS для собственных задач сетевой г/а связи.

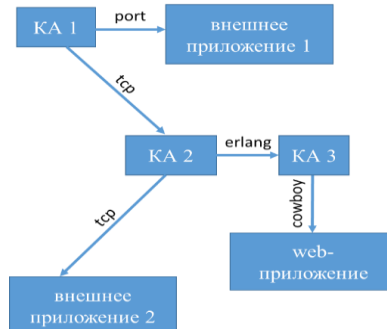


Рис. 3. Типы интерфейсов, используемых в программном каркасе EviNS

Для решения задачи интеграции произвольного г/а модема с EviNS есть два пути. Первый состоит в изменении роли АТ-команд, чтоб протоколы канального и сетевого уровней смогли работать с командами выбранного модема. Второй путь состоит во встраивании в EviNS дополнительных ролей, которые бы наряду с ролью АТ-команд, обеспечивали возможность работы протоколов канального и сетевого уровней с командами сторонних модемов. Для подключения к EviNS любого другого модема в каркас должна быть включена дополнительная (вновь разработанная) модификация протокола уровня управления доступом к среде, и, кроме нее, также дополнительная (вновь разработанная) модификация сетевого протокола. Т.е. сколько модемов необходимо подключить к EviNS, столько и пар модифицированных или вновь разработанных пар протоколов необходимо иметь в составе программного каркаса. Для каждого из модемов понадобится также разработка роли, обеспечивающей синтаксический разбор команд соответствующего модема.

Благодаря открытости кода EviNS, все это реализуемо, однако, такая реализация требует существенного изменения программного каркаса. Во многих смыслах что-либо менять в хорошо работающем программном каркасе нежелательно. В связи с этим, на наш взгляд существует еще один путь, который состоит в том, чтобы не изменять программный каркас, а добавлять в него только такие приложения, которые изначально были бы способны работать не с одним, а разными модемами (с разными ролями). Т.е. в дополнение к существующему протоколу или вместо него можно разработать протокол, который бы работал не только с ролью АТ-команд, но также и с ролями других команд. Это смогло бы существенно сократить затраты на расширение возможностей программного каркаса, в частности, дополнением его возможностями создания гидроакустических сетей на основе г/а модемов разных производителей. В свою очередь, это способствовало бы расширению круга прикладных задач Морского интернета вещей.

В рамках текущей работы для расширения возможностей программного каркаса выполнена разработка пользовательских протоколов уровня управления доступом к среде и протоколов сетевого уровня, которые способны работать с множеством ролей, обеспечивающих выполнение функций синтаксического анализатора команд г/а модемов разных производителей – г/а модема uWave и г/а модема Evologics. Такие роли позволяют выполнять синтаксический разбор команд соответствующего модема и конвертировать их в формат внутренних сообщений,

свойственных приложениям EviNS. Также и в обратную сторону – выполнять синтаксический разбор команд от приложений EviNS и конвертировать их в формат внутренних сообщений соответствующего модема.

Ниже представлено описание протоколов сетевого и канального уровней, реализованных и внедренных авторами для использования в составе программного каркаса EviNS.

Протоколы канального и сетевого уровня. Для обеспечения эффективной работы в условиях г/а среды распространения сигнала и для уменьшения или исключения коллизий при обмене и доставке данных на модемы, не обладающие возможностью синхронной работы, а также для уменьшения времени доступа к среде распространения сигнала был разработан алгоритм передачи данных с применением протокола канального уровня DACAP. Это асинхронный протокол управления доступом к среде, он основан на подтверждении приёма-передачи отдельных соединений между модемами.

Для осуществления обмена данными модем-отправитель, планирующий передать информацию другому модему, генерирует запрос на разрешение передачи (RTS). При получении запроса на отправку данных (RTS) г/а модем немедленно отправляет в ответ запрос на готовность к отправке (CTS), затем ожидает пакет данных. Когда после отправки пакета, разрешающего передачу, модем-получатель обнаруживает активность в среде распространения, которая может привести к повреждению сигнала, он генерирует короткий предупредительный сигнал, что приводит к остановке передачи сообщения. После получения разрешающей команды модем-отправитель выдерживает краткую паузу. Если в течение этой паузы происходит активность других модемов или модем-отправитель получает предупреждающий сигнал от модема-получателя, передача данных отменяется.

Для подводной г/а сети сенсоров малого радиуса действия разработан энергоэффективный протокол T-Lohi, который обеспечивает стабильный доступ к среде передачи при условии небольшой мощности бортового передатчика и небольшого энергоресурса. Протокол обладает механизмом резервирования каналов, гарантирующим отсутствие коллизий между сообщениями при передаче. Помимо программной реализации, протокол подразумевает наличие специального оборудования, имеющего функцию пробуждения по тональному сигналу. Таким образом достигается решение проблемы конфликтов при резервировании каналов без необходимости больших энергозатрат.

Существуют три различные реализации данного протокола: ST-Lohi (синхронизированный T-Lohi), обеспечивающий наиболее энергоэффективную передачу, aUT-Lohi (Aggressive Unsynchronized T-Lohi), позволяющий организовать максимальную из трёх реализаций пропускную способность, и cUT-Lohi (консервативный несинхронизированный T-Lohi), который обеспечивает наиболее надежную передачу данных практически без потерь.

Протокол T-Lohi основан на использовании коротких сообщений (тонов) для резервирования канала связи и разрешения конкурентных запросов на доступ к каналу от других узлов сети. Принцип работы протокола состоит в том, что при наличии данных для передачи у того или иного узла, он уведомляет об этом остальные устройства и входит в соревновательный круг (CR). Каждое устройство, получившее тон, переходит в состояние “Backoff” на период w – нормально распределённое случайное значение, лежащее в диапазоне от 0 до CTC (число претендентов). Резервирует канал в итоге то устройство, которое раньше всех вышло из состояния “Backoff” или то, которое в него не входило. Для остальных начинается следующий соревновательный раунд.

Для обнаружения коллизий и оценки числа соперников используется механизм разрешения коллизий, учитывающий пространственно-временную неопределённость и большую латентность доставки пакетов данных. По числу пришедших тонов

происходит подсчёт претендентов на канал (СТС). Когда устройство приняло тон, оно уходит в состояние “Backoff”. Если по истечению CR устройство не вошло в “Backoff”, оно резервирует канал для передачи пакета данных и выходит из CR.

Flooding – протокол сетевого уровня, является частью протоколов маршрутизации, при которой пакеты отправляются всем узлам сети. Это поведение известно как «затопление». Если «затопление» не контролируется, каждый узел пересылает пакеты всем соседям, что приводит к неуправляемым ширококестельным сообщениям и отказу сети («широковещательный шторм»). Это значительно замедляет пропускную способность и требует больших сетевых ресурсов. Для предотвращения «широковещательного шторма» реализуется условная логика с помощью протокола DPFlooding (Dynamic Probabilistic Flooding) – протокола маршрутизации, основанного на динамическом вероятностном «затоплении». Протокол принимает решение о перенаправлении поступившего пакета данных, исходя из текущего числа доступных соединений.

ICRP (Information-Carrying Based Routing Protocol) – протокол маршрутизации, основанный на передаче совмещенных пакетов со служебной информацией и полезными данными, У этого вида протоколов служебные данные, предназначенные для установления маршрута доставки, передаются как часть пакета с данными полезной нагрузки. Благодаря этому, механизм маршрутизации и передачи данных является эффективным с точки зрения потребления энергии и характеризуется относительно малым временем маршрутизации. Он не зависит от числа участников сети и их взаимного расположения и, соответственно, легко масштабируется для произвольного числа участников, их пространственного разнесения и/или мобильности.

Каждый узел на постоянной основе ширококестельно посылает свою таблицу маршрутизации всем своим соседям. После получения узлом этой информации от соседа, происходит сравнение полученной таблицы маршрутизации со своей. Маршруты в полученной таблице сравниваются маршрутизатором с маршрутами в его таблице маршрутизации. Если новый маршрут имеет лучшую метрику, он может заменить существующий. Для оптимизации работы протокола используется обобщенная метрика, которая служит для более точной характеристики маршрута. Для обеспечения устойчивости маршрутизации вместо использования одного маршрута с наименьшей метрикой, трафик может распределяться по нескольким маршрутам, чьи метрики попадают в указанный диапазон.

Маршрут, имеющий наименьшую обобщенную метрику, будет наиболее предпочтительным. Поскольку пакет управления установлением маршрута передается пакетом данных, механизм передачи информации не только энергоэффективен, но и обеспечивает низкую задержку доставки из конца в конец.

Экспериментальная часть. При последовательной передаче пакетов от одного устройства к другому в условиях, указанных выше, метрики приведены на рис. 4,а. Успешность доставки данных – 96%, среднее время доставки – 1.5 секунды (с учётом расстояния), усреднённая скорость передачи (с учетом потерянных пакетов) – 104 байт/сек. Отметим, что протокол физического уровня, может позволить получить пакет, отправить квитанцию отправителю, но пакет может содержать ошибки.

При попытке принять информацию от двух устройств одновременно, возникают коллизии, приводящие к повреждениям пакетов (рис. 4,б). Успешность доставки данных составляет 56-63%, среднее время передачи – 2.8 секунды (с учётом расстояния), усреднённая скорость передачи (с учетом потерянных пакетов) – 98-130 байт/сек. Это свидетельствует о том, что коллизии, которые возникли с увеличением количества модемов приводят к увеличению нагрузки на принимающий модем и, как следствие, к повреждению пакетов.

Метрики, полученные при передаче данных с использованием протокола канального уровня DACAP, приведены на рис. 4,в. Успешность доставки данных составляет 100%, среднее время доставки – 2.7 секунды (с учётом расстояния), усреднённая скорость передачи – 93 байт/сек.

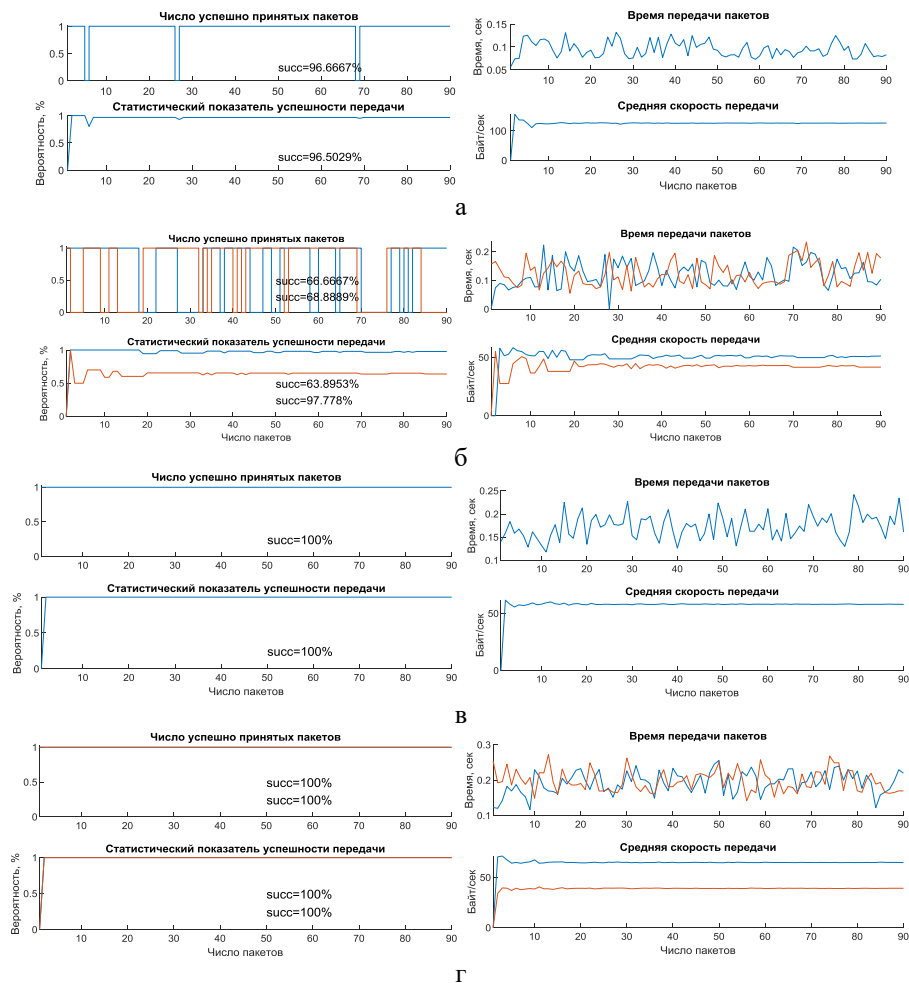


Рис. 4. Графики метрик для случаев: а – передачи в формате «точка-точка»; б – передачи одновременно с двух устройств; в – передачи с использованием протокола DACAP; г – передачи одновременно с двух устройств при использовании протокола DACAP

Использование протокола DACAP повысило надёжность до 100%. Такой результат был достигнут за счёт введения «рукопожатий» и контроля ошибочных пакетов. С использованием протокола, при передаче данных возросла нагрузка на пропускную способность, поэтому средняя скорость с учётом контроля пакетов (ретрансляция) и рукопожатий стабилизировалась на показателе ниже (93 байт/сек).

Метрики, полученные при одновременной передаче данных с двух модемов на один приведены на рис. 4,г. Успешность доставки данных составляет 100%, среднее время доставки – 2.8-3.8 секунды, усреднённая скорость передачи – 49-67 байт/сек.

Несмотря на возросшую нагрузку, благодаря использованию протокола DACAP, надёжность передачи была на уровне 100%, однако скорость передачи, в этом случае, упала до 49-67 байт в секунду.

Проиллюстрируем результаты тестирования протокола сетевого уровня. Пусть очередность передачи пакетов в сети, заданной модемами, расположенными как указано на рис. 5,а. Для передачи данных использовался стек, состоящий из сетевого протокола Flooding и канального протокола DACAP.

Среднее время доставки данных (через всю сеть) составляет 6.7 секунд. Среднее время доставки между двумя узлами – 3.4 секунды. Усреднённая скорость передачи (с учетом потерянных пакетов) составляет 7 байт/сек.

Сетевой протокол Flooding позволяет опосредованно передавать пакеты внутри сети, время и количество принятых пакетов конечным устройством зависит от максимального количества пролетов сети.

Метрики, полученные при передаче данных приведены на рис. 5,б.

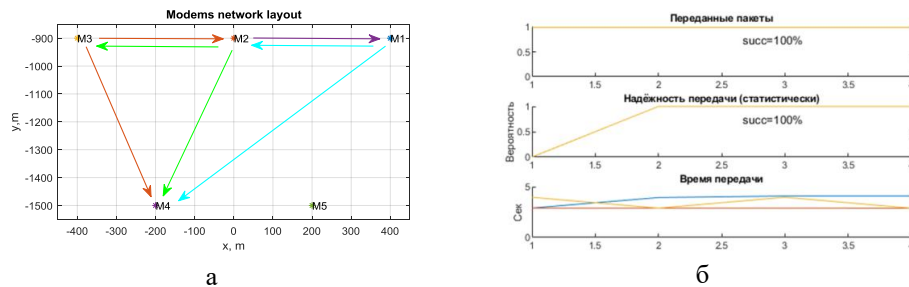


Рис.5. а – очередность передачи пакетов внутри сети; б – метрики, полученные при передаче данных между устройствами внутри сети

Заключение. В статье приведены результаты разработки пользовательской надстройки для программного каркаса EviNS, которая решает задачу реализации сетевой г/а связи. Надстройка включает в себя стек протоколов канального уровня DACAP и T-Lohi, служащих для обработки коллизий и основывающихся на использовании физических задержек при передаче. Стек протоколов сетевого уровня DPFlooding и ICRP служит для объединения группы г/а модемов в сеть и для достижения устойчивой передачи пакетов полезных данных внутри этой сети. Исследование параметров передачи данных по г/а каналу показало, что внедрение протоколов сетевой связи позволило повысить вероятность успешного обмена пакетами данных между группой устройств ценой потерь скорости, возникающих в виду манипулирования значениями задержек при передаче. С использованием протоколов, вероятность успешной отправки пакета с данными устремляется к 100%, однако скорость передачи может быть снижена. Таким образом, подтверждается надёжность и скоростная разница примерно в два раза. В сумме, программный каркас EviNS с реализованными в нём протоколами подводной сетевой г/а связи может быть эффективным образом внедрён в концепцию МПот или группового управления подводными роботами, где важна надёжность канала связи, поскольку повреждённые данные, представляющие собой навигационную информацию или управляющие сигналы для других устройств, могут привести к неверно обработанным командам подводного аппарата и аварийным ситуациям.

Работа выполнена в рамках выполнения проекта FEFM-2024-0015 и внутреннего гранта Севастопольского государственного университета 42-01-09/241/2022-2.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Kabanov A., Kramar V. Marine Internet of Things Platforms for Interoperability of Marine Robotic Agents: An Overview of Concepts and Architectures // Journal of Marine Science and Engineering. – 2022. – No. 9 (10). – 1279.
2. Abreu P. Widely Scalable Mobile Underwater Sonar Technology: An Overview of the H2020 WiMUST Project // Marine Technology Society Journal. – 2016. – No. 4 (50), pp. 42-53.
3. Ali M.F. Recent Advances and Future Directions on Underwater Wireless Communications, Archives of Computational Methods in Engineering, 2020, No. 5 (27), pp. 1379-1412.
4. Dikarev A. Position Estimation of Autonomous Underwater Sensors Using the Virtual Long Baseline Method, International Journal of Wireless & Mobile Networks, 2019, No. 2 (11), pp. 13-25.

5. Chitre M. Bhatnagar R., Soh W.-S. UnetStack: An agent-based software stack and simulator for underwater networks St. John's, NL: IEEE, 2014.
6. Shah S.M. Advancements in Neighboring-Based Energy-Efficient Routing Protocol (NBEER) for Underwater Wireless Sensor Networks, *Sensors*, 2023, No. 13 (23), 6025.
7. Heidemann J., Stojanovic M., Zorzi M. Underwater sensor networks: applications, advances and challenges, *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 2012, No. 1958 (370), pp. 158-175.
8. Schirripa Spagnolo G., Cozzella L., Leccese F. Underwater Optical Wireless Communications: Overview, *Sensors*, 2020, No. 8 (20), 2261.
9. Islam K.Y. Green Underwater Wireless Communications Using Hybrid Optical-Acoustic Technologies, *IEEE Access*, 2021, 9, pp. 85109-85123.
10. Gupta S., Singh N. P. Underwater wireless sensor networks: a review of routing protocols, taxonomy, and future directions, *The Journal of Supercomputing*, 2023, 3681.
11. Hanashi A. M., Algoul S. The Effectiveness of Dynamic Probabilistic Flooding in On-Demand Routing Protocols for MANETs was Assessed Through a Performance Analysis, *European Journal of Theoretical and Applied Sciences*, 2023, No. 5 (1), pp. 935-941.
12. ISO/IEC 30140-1:2018, Information technology – Underwater acoustic sensor network (UWASN). – Part 1: Overview and requirements.
13. Leonov A., Naniy O., Treshchikov V. Sovershenstvovanie formatov modulyatsii v opticheskikh sistemakh svyazi dwdm [Improving modulation formats in dwdm optical communication systems], *LAST MILE Russia*, 2019, No. 8 (85), pp. 30-36.
14. Petrioli C. The SUNSET framework for simulation, emulation and at-sea testing of underwater wireless sensor networks, *Ad Hoc Networks*, 2015, 34, pp. 224-238.
15. Sasson Y., Cavin D., Schiper A. Probabilistic broadcast for flooding in wireless mobile ad hoc networks New Orleans, LA, USA: IEEE, 2003.
16. Shang W. Vectorsync: distributed dataset synchronization over named data networking, *Proceedings of the 4th ACM Conference on Information-Centric Networking*, 2017, pp. 192-193.
17. Peleato B., Stojanovic M. Distance aware collision avoidance protocol for ad-hoc underwater acoustic sensor networks, *IEEE Communications Letters*, 2007, No. 12 (11), pp. 1025-1027.
18. Petrioli C., Petroccia R., Stojanovic M. A comparative performance evaluation of MAC protocols for underwater sensor networks, Quebec City, QC: IEEE, 2008.
19. Syed A.A., Ye W., Heidemann J. T-Lohi: A New Class of MAC Protocols for Underwater Acoustic Sensor Networks Phoenix, AZ, USA: IEEE, 2008.
20. Syed A.A., Ye W., Heidemann J. Comparison and Evaluation of the T-Lohi MAC for Underwater Acoustic Sensor Networks, *IEEE Journal on Selected Areas in Communications*, 2008, No. 9 (26), pp. 1731-1743.
21. Abu Zant M., Yasin A. Avoiding and Isolating Flooding Attack by Enhancing AODV MANET Protocol (AIF_AODV), *Security and Communication Networks*, 2019, pp. 1-12.
22. Liang W. Information-Carrying Based Routing Protocol for Underwater Acoustic Sensor Network, Harbin, China: IEEE, 2007.

Статью рекомендовал к опубликованию к.т.н. А.Н. Греков.

Кебкал Константин Георгиевич – Севастопольский государственный университет; e-mail: konstantin.kebkal@latena.ru; г. Севастополь, Россия; д.т.н.; ведущий научный сотрудник НИЛ «Робототехника и интеллектуальные системы управления».

Кабанов Алексей Александрович – e-mail: kabanovaleksey@gmail.com; тел.: +79787622582; к.т.н.; доцент; директор Института радиоэлектроники и интеллектуальных технических систем.

Альчаков Василий Викторович – e-mail: alchakov@mail.ru; тел.: +79787489941; к.т.н.; доцент кафедры «Информатика и управление в технических системах».

Крамарь Вадим Александрович – e-mail: kramarv@mail.ru; тел.: +79787927340; д.т.н.; профессор кафедры «Информатика и управление в технических системах».

Димин Максим Эдуардович – e-mail: dimin.maksim@yandex.ru; тел.: +79788112909; лаборант-исследователь НИЛ «Робототехника и интеллектуальные системы управления».

Kebkal Konstantin Georgievich – Sevastopol State University; e-mail: konstantin.kebkal@latena.ru; Sevastopol, Russia; dr. of eng. sc.; leading researcher at the Research Laboratory of Robotics and Intelligent Control Systems.

Kabanov Aleksey Aleksandrovich – e-mail: kabanovaleksey@gmail.com; phone: +79787622582; cand. of eng. sc.; associate professor of the department «Informatics and Control in Technical Systems».

Alchakov Vasilii Viktorovich – e-mail: alchakov@mail.ru; phone: +79787489941; cand. of eng. sc.; associate professor of the department «Informatics and Control in Technical Systems».

Kramar Vadim Aleksandrovich – e-mail: kramarv@mail.ru; phone: +79787927340; dr. of eng. sc.; associate professor of the department «Informatics and Control in Technical Systems».

Dimin Maksim Eduardovich – e-mail: dimin.maksim@yandex.ru; phone: +79788112909; laboratory research assistant, Research Laboratory "Robotics and Intelligent Control Systems".

УДК 004.02:007.52

DOI 10.18522/2311-3103-2024-1-213-225

Ю.С. Баричев, О.П. Гойдин, В.П. Носков, С.А. Собольников

**ФОРМИРОВАНИЕ ИНФОРМАЦИОННО-НАВИГАЦИОННОГО ПОЛЯ
РОБОТОВ ВОЗДУШНОГО И НАЗЕМНОГО БАЗИРОВАНИЯ
В УРБАНИЗИРОВАННОЙ СРЕДЕ**

Обоснована возрастающая в последнее время востребованность использования гетерогенных групп роботов (роботов воздушного и наземного базирования) повышенной автономности при проведении спецопераций в индустриально-городских средах, включая здания. Сформулирована актуальная задача формирования по данным бортовых систем технического зрения беспилотного летательного аппарата информационно-навигационного поля, обеспечивающего автономный целенаправленный безопасный полет и движение роботов воздушного и наземного базирования в экранированных зонах урбанизированной среды. Формирование обобщенной геометрической модели внешней среды можно обеспечить путем задания множества целевых положений в плане рабочей зоны, которые должен посетить БПЛА в заданной последовательности и вернуться в точку старта. В процессе посещения достижимых целевых точек формируется обобщенная геометрическая модель внешней среды и определяются текущие координаты БПЛА. Описаны методы и алгоритмы построения различных моделей внешней среды и решения навигационной задачи, которые обеспечивают планирование и отработку целенаправленных безопасных траекторий движения в реальном времени по данным бортовых средств, что и является основой автономного управления в том числе и управления гетерогенной группой роботов. В основу систем автономного управления движением роботов воздушного и наземного базирования положены методы и алгоритмы выделения семантических объектов (плоскостей опорной поверхности и вертикальных стен), которыми изобилуют урбанизированные среды, и экстремальной навигации по двумерным горизонтальным сечениям 3D-изображений (облакам точек), полученным с помощью лидара или камеры глубины. Приведены результаты работы созданных программно-аппаратных средств систем автономного управления роботами воздушного и наземного базирования по формированию информационно-навигационных полей и решению навигационных задач по данным бортовых систем технического зрения в реальной индустриально-городской среде, подтвердившие эффективность и практическую ценность предлагаемых методов и алгоритмов. Использование единого информационно-навигационного поля, с одной стороны, существенно повышает автономность группы роботов за счет возможности самостоятельного планирования действий при выполнении сложных спецопераций, а с другой стороны, повышает ситуационную осведомленность операторов роботов, предоставляя в удобной форме информацию о месте проведения работ.

Беспилотный летательный аппарат, наземный мобильный робот, групповое управление, локализация, картографирование.