

УДК: 004.72+004.942

Исследование и оптимизация работы беспроводной сенсорной сети на основе протокола ZigBee

С. В. Трифонов^а, Я. А. Холодов

Московский физико-технический институт,
Россия, 141700, Московская обл., г. Долгопрудный, Институтский пер., 9

E-mail: ^аsvtrifonov@gmail.com

Получено 13 ноября 2012 г.
после доработки 19 декабря 2012 г.

В работе рассматриваются вопросы алгоритмов функционирования беспроводных сетей на основе модифицированного стека протоколов ZigBee/IEEE 802.15.4 и проблемы энергосбережения с одновременным уменьшением времени доставки сообщений. Даны теоретические выкладки и описаны алгоритмы распределения ролей и установки расписаний для маршрутизаторов. Приведены и проанализированы результаты проведенных натуральных экспериментов, а также численных экспериментов выполненных с помощью открытого программного комплекса ns-2.

Ключевые слова: сенсорные сети, беспроводные сети, ZigBee, IEEE 802.15.4, дискретно-событийное моделирование, оптимальное расписание

Study and optimization of wireless sensor network based on ZigBee protocol

S. V. Trifonov, Ya. A. Kholodov

Moscow Institute of Physics and Technology, 9, Institutskii per., Dolgoprudnyi, Moscow region, Russia, 141700

Abstract. — Algorithms of wireless sensor networks operation based on modified ZigBee/IEEE 802.15.4 protocol stack and problems of energy saving with simultaneous decrease of network latency are studied. Theoretical computations are given. Roles distribution and routers schedule assignment algorithms are described. Both results of experiments carried out with real devices and results of simulations with ns-2 (open-source network simulator) are given and analyzed.

Keywords: sensor network, wireless network, ZigBee, IEEE 802.15.4, discrete-event simulation, optimal schedule

Citation: *Computer Research and Modeling*, 2012, vol. 4, no. 4, pp. 855–869 (Russian).

Введение

Для изучения протоколов передачи данных широко используются метод дискретно-событийного имитационного моделирования, позволяющий представить процесс функционирования системы как последовательность событий. Каждое событие происходит в определенный момент времени и может изменять состояние системы [Banks и др., 2009]. Сегодня для решения задач дискретно-событийного имитационного моделирования сетей связи существует достаточно широкий спектр программных средств: от библиотек функций для стандартных компиляторов до специализированных языков программирования [Парамонов, 2000; Fall, Vagadhan, 2001].

Однако имитационное моделирование не может учесть всех аспектов реальной моделируемой системы. Всегда вводятся предположения, позволяющие упростить и, как следствие, ускорить вычислительный расчёт. Но интуитивно понять, какие предположения не повлекут за собой расхождения модели и реальной системы, сложно. Поэтому в данной работе для проверки модели используется стендовое моделирование. Это экспериментальное исследование небольшой реальной системы, позволяющее учесть все аспекты взаимодействия её частей и влияние внешних факторов. Так как стендовые исследования проводятся с небольшим числом устройств, то для обобщения результатов на системы большей размерности необходим переход к имитационной модели. В нашей работе используется имитационная модель, построенная на базе дискретно-событийного сетевого симулятора ns-2.

В работе рассматриваются беспроводные сети, для которых характерно сверхнизкое энергопотребление и большое количество миниатюрных устройств, обменивающихся относительно небольшим количеством информации по протоколу ZigBee. Принципиальной отличительной особенностью беспроводных сенсорных сетей является принцип ретрансляции данных по цепи. Это позволяет собирать информацию со значительных объектов, превосходящих по своим размерам радиус связи одного элемента. Элементы сенсорной сети являются автономными, содержат датчики для контроля внешней среды, микрокомпьютер и маломощный радиоприемопередатчик.

Возможности использования сенсорных сетей простираются практически во все сферы деятельности человечества. В качестве наиболее очевидных областей их применения эксперты называют промышленный мониторинг, автоматизацию строений («умный дом»), логистику, здравоохранение, системы безопасности и оборону.

Целью данной работы является построение и изучение алгоритмов, управляющих работой беспроводной сети для достижения минимального энергопотребления устройств. Задачи работы включают в себя:

- теоретическое, экспериментальное и численное исследования поведения сети работающей в соответствии со спецификациями IEEE 802.15.4 и ZigBee;
- основанная на данных исследованиях разработка оптимизационных алгоритмов и соответствующих программ, включая модифицированный стек протоколов;
- проведение ряда сравнительных стендовых экспериментов с использованием модифицированного и стандартного стеков протоколов для измерения и сравнения характеристик работы сети (времени жизни устройств и времени доставки сообщений);
- разработка модели модифицированного стека протоколов на базе дискретно-событийного симулятора ns-2;
- проведение ряда сравнительных численных экспериментов с использованием моделей модифицированного и стандартного стека протоколов с целью изучения эффективности масштабирования разработанных алгоритмов оптимизации при увеличении числа узлов сети.

Работа над задачами связанными со стендовым моделированием была выполнена совместно с институтом точной механики и вычислительной техники имени С. А. Лебедева РАН.

Модель работы беспроводной сети на основе протокола ZigBee

Основные характеристики ZigBee

Спецификация ZigBee предусматривает передачу информации в радиусе от 5 до 75 метров с максимальной скоростью 250 кбит/с. За стандартом ZigBee закреплены 27 каналов в трех частотных диапазонах — 2.4 ГГц, 915 МГц и 868 МГц. Максимальная скорость передачи данных для этих эфирных диапазонов составляет соответственно 250 кбит/с, 40 кбит/с и 20 кбит/с.

В сущности, ZigBee — это не один протокол: спецификация ZigBee [ZigBee Alliance Std., 2007] регламентирует стек протоколов, в котором протоколы верхних уровней используют сервисы, предоставляемые протоколами нижележащих уровней. В качестве двух нижних уровней (физического уровня РНУ и уровня доступа к среде MAC) используется стандарт IEEE 802.15.4 [IEEE Std. 802.15.4, 2003]. MAC-уровень в сети ZigBee реализует механизм CSMA-CA (прослушивания несущей и устранения коллизий), сетевой уровень (NWK) ответственен за маршрутизацию сообщений, а уровень APS (поддержки приложений) обеспечивает интерфейс с уровнем приложения.

Предлагаемые в данной работе оптимизационные алгоритмы основываются на особенностях и модификации протоколов сетевого (NWK) и канального (MAC) уровней.

Принцип работы протокола канального уровня

Спецификация IEEE 802.15.4 описывает два режима работы сети: маячковый и без маяков. В режиме без маяков маршрутизаторы постоянно прослушивают эфир, потому нельзя построить сеть со сверхнизким энергопотреблением.

В маячковом режиме все устройства с определённым периодом посылают в эфир специальные кадры, называемые маяками. Интервал между маяками (*BI*) включает в себя активный период, называемый суперфреймом, и неактивный период (см. рис. 1). В течение суперфрейма возможна передача кадров данных. Во время неактивного периода узлы засыпают.

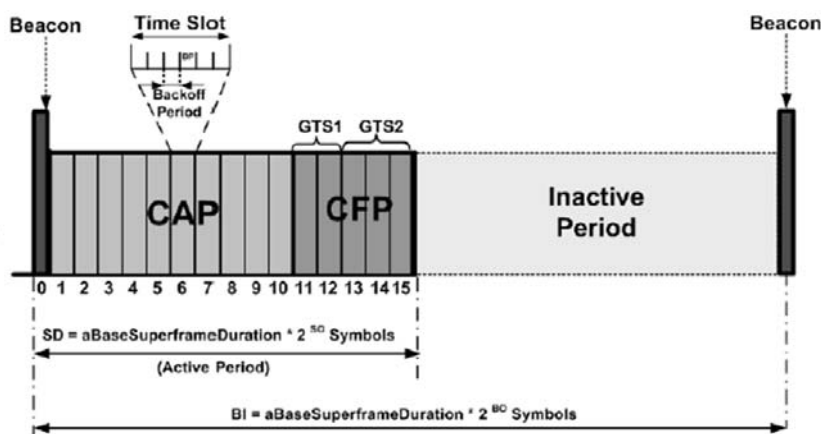


Рис. 1. Принцип работы сети в маячковом режиме

Интервал между маяками и длительность суперфрейма определяются параметрами Beacon Order (*BO*) и Superframe Order (*SO*) соответственно:

$$BI = aBaseSuperframeDuration \cdot 2^{BO}, 0 \leq BO \leq 14, \quad (2.1)$$

$$SD = aBaseSuperframeDuration \cdot 2^{SO}, 0 \leq SO \leq BO, \quad (2.2)$$

где *aBaseSuperframeDuration* обозначает минимальную длительность суперфрейма, соответствующую *SO* = 0. Эта величина фиксирована и равна 15.36 мс [IEEE Std. 802.15.4, 2003], предполагая 250 кбит/с в частотном диапазоне 2.4 ГГц.

В течение периода конкурентного доступа (САР) узлы соперничают за получение доступа к физической среде, используя механизм слотного CSMA-CA (Slotted Carrier Sense Multiple Access with Collision Avoidance). Этот механизм довольно сложен, его всестороннее исследование средствами ns-2 и OPNET представлены в [Ramachandran, Das, Roy, 2007] и [Koubâa, Alves, Tovar, 2006] соответственно. В протоколе IEEE 802.15.4 также предусмотрен период неконкурентного доступа к среде (CFP), который в данной работе не рассматривается.

Протокол сетевого уровня

Спецификация ZigBee предлагает использование одного из двух протоколов сетевого уровня. Первый подход — реактивный протокол динамической маршрутизации AODV (Ad hoc On-Demand Distance Vector). Второй — протокол HERA (HiErarchical Routing Algorithm), использующий дерево ассоциации узлов, которое строится и поддерживается на канальном уровне. В отличие от AODV, разрешаются связи только одного вида координатор-узел. Протокол HERA является проактивным, поэтому не требуется никаких дополнительных служебных сообщений для построения маршрутов.

В [Cuomo и др., 2007] были проведены сравнительные исследования протоколов AODV и HERA средствами сетевого симулятора ns-2, которые показали, что использование протокола HERA имеет ряд преимуществ по сравнению с AODV. Причем в типичных сенсорных сетях информация передается с неподвижных источников на неподвижный приемник. Ввиду этих аргументов в данной работе используется протокол HERA.

Оптимизационные алгоритмы управления работой сети

Задача минимизации энергопотребления

Сформулируем исходные предположения о структуре и назначении сети:

1. Подключения к сети новых устройств и изменения ее конфигурации происходят относительно редко.
2. На сеть не возлагается задача передачи сигнала между произвольными узлами — достаточно иметь возможность передавать сигнал от узла к координатору и обратно.
3. Объем информации, генерируемый сетью в единицу времени, не превышает ее пропускной способности.

Ввиду этих предположений возникает возможность разделения устройств по ролям, энергопотребление в которых сильно различается. Отказ от идеи самоорганизации и централизация управления сетью приводит к отсутствию необходимости в собственном суперфрейме на конечных устройствах. Это снижает энергопотребление конечных устройств почти вдвое.

Дальнейшая оптимизация связана с возможностью конечного устройства засыпать сразу же после родительского маяка, не слушая суперфрейм, при условии, что устройству не адресовано данных и у него нет данных для отправки. Эта оптимизация позволяет снизить энергопотребление ещё в 3–4 раза. Результаты теоретического расчёта приведены на рис. 2.

Из рисунка видно, что чем более быстрой является сеть, тем более эффективны оптимизации. Но при $BO - SO \leq 1$ из формул (2.1) и (2.2) следует, что $BI/SD \leq 2$, и становится невозможно построить сеть с маршрутизаторами из-за проблемы скрытой станции. Поэтому имеет смысл рассматривать только сети с $BO - SO = |SO = 0| = BO \geq 2$. В такой ситуации батареи маршрутизаторов будут разряжаться в 6–7 раз быстрее, чем батареи конечных устройств и время жизни всей сети¹ будет определяться временем жизни маршрутизаторов. Чтобы этого избежать, предлагается следующий подход.

¹ Под временем жизни сети здесь понимается время до полной разрядки хотя бы одного из устройств сети.

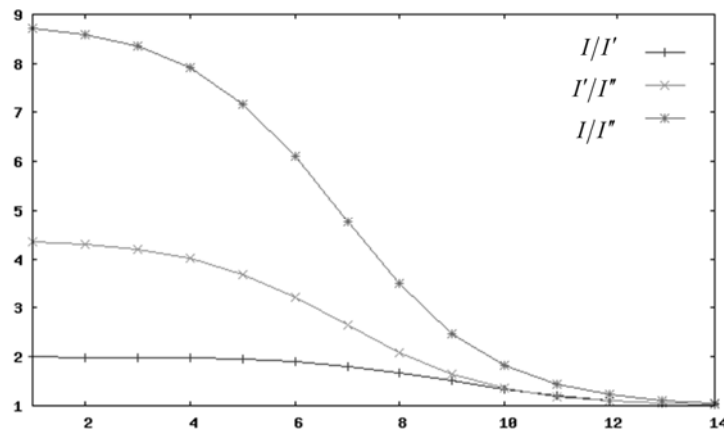


Рис. 2. Зависимость отношений средних сил токов от BO (I' — сила тока с учётом первой оптимизации, I'' — с учётом обеих, I — без учёта оптимизаций)

Рассмотрим сеть, состоящую из N узлов. Обозначим через $V = \{a_1, a_2, \dots, a_N\}$ множество узлов. Два узла, находящиеся в области прямой видимости друг друга, образуют ребро. Таким образом, мы получим граф смежности $G = \{V, E\}$. Для простоты считаем граф связанным и неориентированным. Узел a_0 — координатор сети. Построим все связанные подграфы T_k , $k \in \overline{1, K}$ графа G , являющийся деревом с корневым элементом a_0 и содержащий все вершины G . R_k — набор всех маршрутизаторов графа T_k .

Если динамически менять роли узлов и перестраивать топологию сети, то можно приблизить время жизни сети ко времени жизни конечных устройств. Это увеличение времени жизни возможно за счёт того, что большую часть времени каждый из узлов будет находиться в роли конечного устройства. При этом наборы одновременно работающих маршрутизаторов циклически сменяют друг друга. Решения о перестроении топологии принимаются на координаторе сети.

Решим задачу максимизации времени жизни сети. Рассмотрим произвольное подмножество $\{R_{k_m}\}_{m=1}^M$ множества $\{R_k\}_{k=1}^K$. Имеем M_i — число наборов из $\{R_{k_m}\}_{m=1}^M$ содержащих a_i . Тогда средняя сила тока в узле a_i выразится формулой²

$$I_i = I_R \left(\frac{M_i}{M} \right) + I_E \left(\frac{M - M_i}{M} \right) = I_E + (I_R - I_E) \frac{M_i}{M}, \quad (3.1)$$

где I_R — средняя за время BI сила тока в узле, находящимся в роли маршрутизатора; I_E — средняя за время BI сила тока в узле, находящимся в роли конечного устройства. Время жизни сети определяется устройством с самым коротким временем жизни

$$T = \min_i \frac{Q_i}{I_i} \rightarrow \max, \quad (3.2)$$

где Q_i — заряд батареи устройства a_i . Далее для простоты предполагаем, что в начальный момент времени все устройства имеют одинаковый заряд батареи $Q_i = Q$. Тогда условие (3.2) переходит в

$$\max_i I_i = I_E + (I_R - I_E) \frac{\max_i M_i}{M} \rightarrow \min. \quad (3.3)$$

² Предполагается, что затраты энергии на переключение ролей и перестроение топологии сети пренебрежимо малы в силу того, что интервал $\Delta t = \text{const}$ между переключениями ролей удовлетворяет условию $BI \ll \Delta t \ll T$.

Если наборы $\{R_{k_m}\}_{m=1}^M$ независимы, то $\forall i \in \overline{1, N} \rightarrow M_i \in \{0; 1\}$, поэтому $\max M_i = 1$, и остаётся лишь условие

$$M \rightarrow \max. \quad (3.4)$$

Это и есть искомое условие. Итак, для максимизации времени жизни сети необходимо найти максимальное число независимых наборов маршрутизаторов.

Алгоритм распределения ролей

Требуется решить задачу о поиске максимального количества M независимых наборов маршрутизаторов $\{R_m\}_{m=1}^M$ графа $G = \{V, E\}$ и соответствующих им деревьев $\{T_m\}_{m=1}^M$, покрывающих всю сеть.

Точное решение этой задачи для больших графов требует больших затрат времени, что неприемлемо. Поэтому используется простой алгоритм на графе (рис. 3), который позволяет найти не максимальное, а достаточно большое количество наборов маршрутизаторов. Этот алгоритм является «жадным». На каждом шаге он старается покрасить как можно больше вершин, что обеспечивает одновременно минимальность получаемых наборов маршрутизаторов и высокую скорость работы алгоритма.

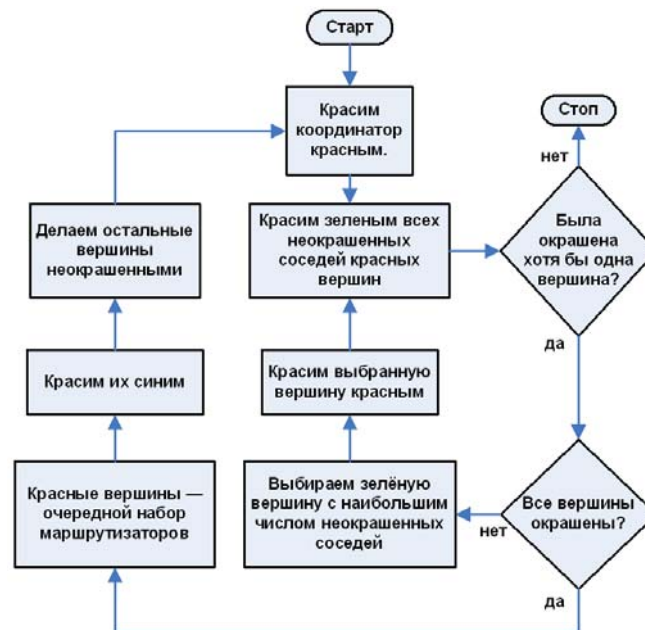


Рис. 3. Схема алгоритма распределения ролей

В результате работы алгоритма мы получаем число M и наборы $R_m, m \in \overline{1, M}$. Для того чтобы получить дерево T_m соответствующее набору R_m , пользуемся следующими правилами.

1. Вначале соединяем координатор со всеми его соседями.
2. К присоединённым соседям-маршрутизаторам подсоединяем их соседей³.
3. Повторяем пункт 2, пока все узлы не будут соединены.

Можно сделать оценку числа M сверху. Рассмотрим произвольную вершину графа $a_i \in V$. Пусть P_i — множество соседей вершины a_i и состоит из k_i элементов. Для того чтобы $\forall m \in \overline{1, M} \rightarrow a_i \in T_m$ требуется, чтобы в каждый момент времени хотя бы один из её соседей был

³ Здесь может возникнуть неоднозначность, если было присоединено более одного маршрутизатора за раз и множества соседей этих маршрутизаторов пересекаются. Однако она не мешает и, более того, позволяет делать оптимизации по таким параметрам, как, например, длины маршрутов до узла и загруженности маршрутизаторов.

маршрутизатором или координатором, то есть $\forall m \in \overline{1, M} \exists a \in P_i : a \in R_m \cup a_0$. Если $a_0 \in P_i$, то это условие всегда выполняется. Если $a_0 \notin P_i$, то с учётом $\forall m_1, m_2 \in \overline{1, M} \rightarrow R_{m_1} \cap R_{m_2} = \emptyset$, удовлетворить условию можно только если $M \leq k_i$. Поскольку выбор $a_i \in V$ был произволен, то $\forall i \in I \rightarrow M \leq k_i$, причём I определяется из условия $\forall i \in I \rightarrow a_0 \notin P_i$. В итоге получаем

$$M \leq \min_{i \in I} k_i, \quad (3.5)$$

где I — множество индексов вершин, не являющихся соседями координатора; k_i — число соседей узла a_i .

Модификация протоколов канального и сетевого уровней

Для реализации описанной выше идеи требуется централизованное управление распределением ролей. Эту задачу решает координатор. Для получения матрицы смежности графа G после построения в стандартном режиме ZigBee произвольной топологии сети, включающей все устройства, нужно чтобы каждое устройство в течение хотя бы одного интервала между маяками прослушивало эфир. Поскольку в стандартном режиме ZigBee все устройства посылают маяки, то будет установлена полная матрица смежности графа G . Данные с каждого узла отправляются на координатор. Координатор запускает алгоритм распределения ролей и выбирает топологию сети.

Дальнейший принцип поведения справедлив как для координатора, так и для любого маршрутизатора, который узнаёт структуру поддерева, за которую он отвечает. Зная набор адресов A узлов, которые должны быть подчинены маршрутизатору, и набор адресов B своих текущих подчинённых, маршрутизатор делит узлы на множества $A \cap B$, A/B , B/A и остальные узлы.

Связи с узлами из B/A разрываются при помощи сообщения о переходе сети в режим построения фиксированной топологии. Если узел получает такое сообщение, то он передаёт его своим подчинённым, отсоединяется и начинает сканировать эфир в поисках маяка, в котором указан его адрес. Для подсоединения к себе узлов из A/B , маршрутизатор добавляет в каждый отправляемый им кадр маяка адреса этих узлов. Узлам из $A \cap B$ маршрутизатор сообщает о переходе сети в новый режим другим способом — рассылкой сообщений специального типа, содержащих структуру поддерева, за которую будет отвечать принимающий узел. Такое сообщение посылается также и присоединившимся узлам из множества A/B .

При получении сообщения содержащего структура поддерева, узел начинает действовать по уже описанному принципу поведения.

Задача минимизации времени доставки сообщений координатору

Рассчитаем время доставки Δ_a сообщений координатору от произвольного узла сети $a \in V$. При постоянной топологии сети сообщения координатору от a всегда передаются по одной и той же цепочке узлов от детей к родителям. Обозначим её через $\{b_i\}_{i=0}^d$, причём $b_0 = a_0$ (координатор) и $b_d = a$. Время между возникновением события и передачей сообщения о нём от b_d к b_{d-1} ⁴ есть некоторая случайная величина τ_1 , распределённая в интервале $[0, BI - SD)$. Дальнейшая часть времени доставки определяется задержками на маршрутизаторах. Задержка на маршрутизаторе равна промежутку времени между собственным суперфреймом и родительским суперфреймом на этом маршрутизаторе. В нашем случае множество маршрутизаторов

⁴ Сам процесс передачи будем считать мгновенным в силу того, что его длительность много меньше интервала между маяками.

$\{b_l\}_{l=1}^{d-1}$ состоит из $d-2$ элементов. Задержка на каждом из них это число τ_{b_l} из интервала $[SD, BI - SD)$.

Последнее слагаемое определяется моментом времени внутри суперфрейма координатора, когда произошла реальная передача сообщения координатору. Будем считать его случайной величиной τ_2 , распределённой в интервале $[0, SD)$ ⁵. Итак,

$$\Delta_a = \tau_1 + \sum_{l=1}^{d-1} \tau_{b_l} + \tau_2. \quad (3.6)$$

Далее будем считать случайные величины равномерно распределёнными на своих интервалах. Тогда математическое ожидание первого и последнего слагаемых дадут $E(\tau_1 + \tau_2) = BI / 2$.

Можно изменять задержки на маршрутизаторах, устанавливая смещения собственного маяка маршрутизатора относительно маяка его родителя. Если собственный суперфрейм находится непосредственно перед суперфреймом родителя, то достигается минимальная задержка, но этот вариант не всегда возможен из-за наложения маяков и суперфреймов. Возможны различные варианты наложения маяков при изменении смещения на произвольном узле. Возможны также коллизии с суперфреймами соседних, родительского или дочерних, узлов. Рассчитаем среднее (по всем узлам) время доставки сообщений τ_T для произвольного дерева топологии T , зная времена задержек на всех маршрутизаторах $R \subset V$ этого дерева:

$$\tau_T = \frac{1}{N-1} \sum_{a \in V/a_0} \Delta_a = \frac{1}{N-1} \sum_{a \in V/a_0} \left(\sum_{l=1}^{d(T,a)-1} \tau_{b_l(T,a)} + \frac{BI}{2} \right).$$

Здесь обозначения $b_l(T, a)$ и $d(T, a)$ указывают на то, что цепочка, направленная от координатора к узлу a , зависит от дерева топологии T и самого узла a . Раскрыв скобки и сгруппировав члены, получим

$$\tau_T = \frac{BI}{2} + \frac{1}{N-1} \sum_{a \in V/a_0} w(T, a) \tau_a, \quad (3.7)$$

где $w(T, a)$ — размер поддерева, корнем которого является узел a , не считая самого узла. Это получается потому, что число цепочек $\{b_l\}_{l=0}^d$, содержащих произвольный узел a в качестве маршрутизатора, равно размеру указанного поддерева. Можно также сказать, что $w(T, a)$ — вес узла a в топологии T . Задача минимизации время доставки сообщений о событиях в сети координатору сводится к минимизации величины $\tau_T \rightarrow \min$ при фиксированном BI :

$$\sum_{a \in V/a_0} w(T, a) \tau_a \rightarrow \min. \quad (3.8)$$

Алгоритм распределения слотов суперфреймов

Разделим интервал между маяками BI на $S = 2^{BO-SO}$ слотов. Каждый из них будет равен длительности суперфрейма SD . Поставим в соответствие каждому маршрутизатору $a \in R$ число $s = \text{Sf}(a) \in \overline{1, S}$. Число s называется номером слота суперфрейма узла a и определяет промежуток времени Δt между началом собственного маяка устройства a и началом маяка координатора сети по формуле $\Delta t = SD \cdot s$. Имея функцию $\text{Sf} : R \rightarrow \overline{1, S}$, можно установить величину

⁵ Реально передача возможна только после маяка, т. е. интервал будет $[BD, SD)$, где BD — длительность маяка.

ны $\tau_a = (\text{Sf}(a) - \text{Sf}(p))$ для любого $a \in R$, где p — родитель узла a . Таким образом, можно свести задачу предыдущего раздела к отысканию функции Sf , минимизирующей значение величины τ_T , и удовлетворяющей условиям:

- $\text{Sf}(p) \neq \text{Sf}(c)$, если p является родителем c ;
- $\text{Sf}(c) \neq \text{Sf}(n)$, если c является соседом n ;
- $\text{Sf}(c) \neq \text{Sf}(l)$, если l является родителем соседа c ;
- $\text{Sf}(c) \neq \text{Sf}(g)$, если g является соседом дочернего узла c .

Для приближённого решения этой задачи предлагается простой алгоритм. Будем последовательно назначать каждому маршрутизатору значение функции Sf так, чтобы на каждом шаге выполнялись требуемые ограничения. Из допускаемых ограничениями значений функции будем выбирать значение, минимизирующее величину задержки τ_a на этом маршрутизаторе. Обход вершин начнём от координатора и продолжим в порядке уменьшения значений $w(T, a)$.

Анализ результатов натурных экспериментов

Экспериментальное измерение энергопотребления устройств

В основу экспериментальных измерений положено то, что, имея экспериментально установленные осциллограммы суперфреймов устройств, можно измерить энергопотребление устройства за один собственный суперфрейм и один родительский суперфрейм. Также возможно исследовать, как энергопотребление зависит от событий, происходящих во время суперфреймов. Для этого необходимо получить осциллограммы и вычислить площадь под ними для следующих типов суперфреймов:

1. Пустой родительский суперфрейм.
2. Пустой собственный суперфрейм маршрутизатора.
3. Родительский суперфрейм с передачей одного типичного сообщения.
4. Собственный суперфрейм маршрутизатора, получающего одно типичное сообщение.
5. Родительский суперфрейм с получением одного типичного сообщения.
6. Собственный суперфрейм маршрутизатора, отправляющего одно типичное сообщение.

Полученные данные позволяют рассчитать среднее энергопотребление устройств при любых интервалах между маяками и при трафике. Было сделано по три измерения для каждого из предполагаемых видов суперфреймов. Общая длина передаваемого сообщения во всех случаях — 50 байт.

На основе полученных данных можно сделать вывод, что влияние любого трафика на время жизни устройств очень мало. Оно оценивается сверху величиной в 2 % времен жизни устройства. Сравнение результата с теоретической оценкой показывает, что отношения (для исследованных $BO = 4$ и $BO = 5$) средних сил токов маршрутизатора и конечного устройства оказались на 13 % меньше, чем теоретические значения на рис. 2.

Измерение времени доставки сообщений

Целью эксперимента было измерение среднего и максимального времени доставки сообщений, частот потери и дублирования пакетов. Сравнения проводились по топологии «цепочка» при включённом и выключенном алгоритме установки расписаний.

Запускалась сеть из 5–10 устройств. Строилась топология «цепочка». Последнее устройство в цепи периодически генерировало сообщения. Оно связывалось проводом с компьютером, к которому был подключён координатор. При генерации сообщения узел также посылал сигнал по проводу. Последовательно проводилось два испытания с использованием одной и той же топологии: с оптимальным и случайным (спонтанным) расписанием.

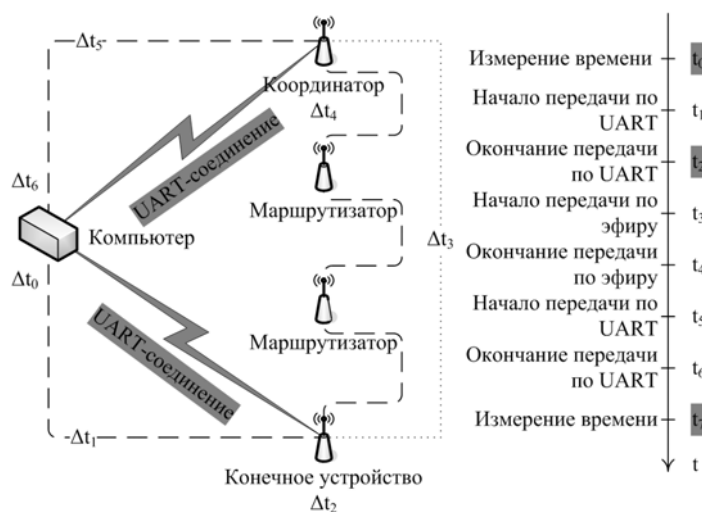


Рис. 4. Схема эксперимента и жизненный цикл сообщения.

Было сделано шесть испытаний, в процессе каждого проделана серия измерений времени. Исследовались сети с параметрами $BO = 4$ и $BO = 5$. Схема эксперимента показана на рис. 4. Серым цветом выделены моменты времени, в которые проводились замеры. Искомое время доставки сообщений — $t_7 - t_2$. По результатам эксперимента построены гистограммы распределения сообщений по временам их доставки.

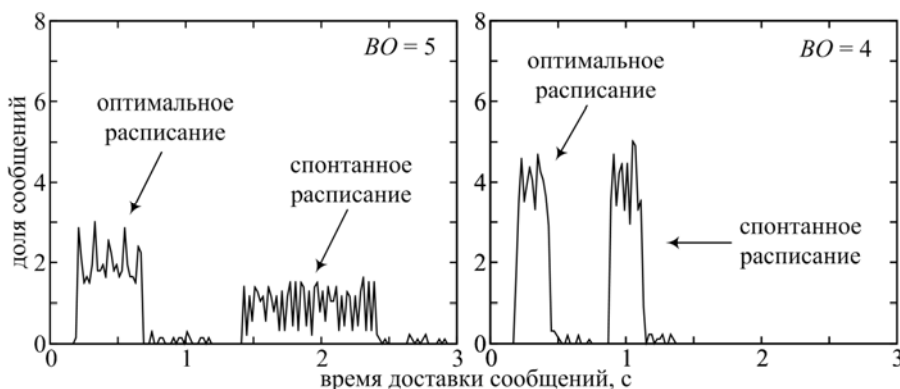


Рис. 5. Гистограммы распределения сообщений по временам их доставки.

Наличие двух границ на гистограммах указывает на равномерное распределение времени ожидания $t_3 - t_2$ на конечном устройстве и практически постоянное время доставки за одну волну $t_4 - t_3$. Найденные значения представлены в таблице ниже.

Таблица 1. Анализ гистограмм

BO	Расписание	Среднее время доставки, с	Минимальное время доставки за одну волну, с	Максимальное время доставки за одну волну, с
4	Случайное	1.009	0.890	1.110
	Оптимальное	0.321	0.190	0.430
5	Случайное	1.933	1.430	2.390
	Оптимальное	0.459	0.210	0.670

Проделанные испытания позволяют сделать следующие выводы. При включённом алгоритме установки расписаний происходит уменьшение среднего времени доставки в 3.1 и 4.2 раза в испытаниях с $BO = 4$ и $BO = 5$ соответственно. Минимальное время доставки за одну волну в оптимальном режиме не зависит от BO и составляет ~ 200 мс. Полученные экспериментально данные для оптимального расписания хорошо согласуются с теоретической оценкой.

Сетевой симулятор ns-2

Архитектура симулятора

Ns-2 является объектно-ориентированной системой, ядро которой реализовано на языке C++. Язык скриптов OTcl используется в качестве интерпретатора. Ns-2 поддерживает компилируемую иерархию классов C++ и подобную иерархию классов интерпретатора OTcl. Существует однозначное соответствие между классом одной иерархии и таким же классом другой [Кучерявый].

Использование двух языков программирования в ns-2 объясняется следующими причинами [Fall, Varadhan, 2001]. С одной стороны, для детального моделирования протоколов необходимо использовать системный язык программирования, обеспечивающий высокую скорость выполнения и способный манипулировать достаточно большими объемами данных. С другой стороны, для удобства пользователя и быстроты реализации различных сценариев моделирования привлекательнее использовать язык программирования более высокого уровня. C++ позволяет обеспечить высокую производительность и возможность работы с пакетами на низком уровне абстракции. А OTcl позволяет обеспечить: простоту построения сценария моделирования, возможность соединения воедино блоков, выполненных на системных языках программирования, и простую манипуляцию ими. Базовые идеи скриптовых языков описаны в [Ousterhout, 1998].

Моделирование ZigBee в ns-2

В рамках проекта ns-2 реализована возможность моделирования работы беспроводной сети на основе протокола ZigBee на трех уровнях: PHY, MAC, NWK. На физическом уровне (PHY) и уровне доступа к каналу (MAC) реализованы все существующие возможности протокола IEEE 802.14.5 [Zheng, Lee, 2004]. Однако отсутствует реализация протокола HERA сетевого уровня, выбранного для проведения экспериментов. Потому потребовалось разработать соответствующие программные модули для ns-2. Реализация предлагаемых в работе оптимизационных алгоритмов также требует модификации протокола сетевого уровня. Все необходимые изменения моделей, описанные в предыдущих разделах, были выполнены.

Анализ результатов численных экспериментов

Численное измерение времени доставки сообщений

Был проведен ряд имитационных экспериментов по измерению времени доставки сообщений, аналогичных натурным. Результаты представлены на рис. 6 и в табл. 2 (в скобках указана относительная ошибка по сравнению с натурным экспериментом).

Для данного эксперимента было отключено моделирование ошибок на канальном уровне, чтобы все пакеты доставлялись за одну волну, так как при анализе результатов натурального эксперимента учитывались только такие пакеты. Этим вызвано отсутствие небольших «хвостов» гистограмм, которые видны на рис. 5. По результатам проведенного сравнения можно сделать вывод, что используемая модель обладает приемлемой точностью и дает ошибку не более 5 % для определения средних величин доставки сообщений.

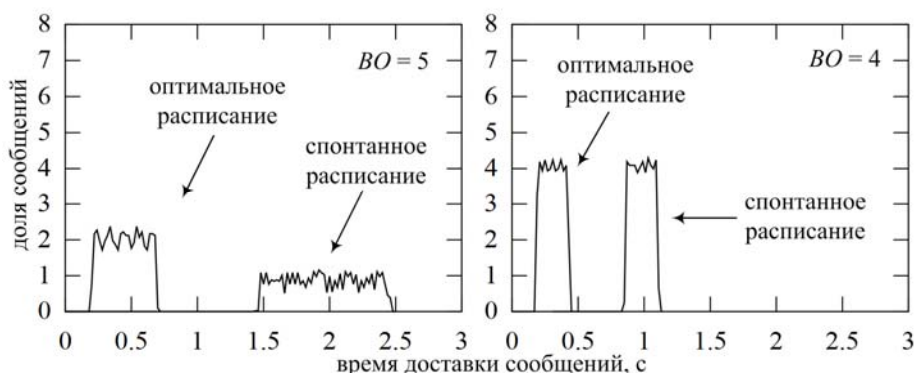


Рис. 6. Гистограммы распределения сообщений по временам их доставки

Таблица 2. Анализ и сравнение гистограмм

ВО	Расписание	Среднее время доставки, с	Минимальное время доставки, с	Максимальное время доставки, с
4	Случайное	0.981 (2.8 %)	0.858 (3.6 %)	1.104 (0,5 %)
	Оптимальное	0.307 (4.4 %)	0.182 (4.2 %)	0.430 (<0,2 %)
5	Случайное	1.951 (0.9 %)	1.456 (1.8 %)	2.444 (2,3 %)
	Оптимальное	0.447 (2.6 %)	0.198 (5,7 %)	0.691 (3,1 %)

Численная оценка времени доставки сообщений в больших сетях

Убедившись, что для простого эксперимента результаты натурального и численного испытаний находятся в хорошем согласии между собой и с теоретической оценкой, можно проводить испытания с большим числом узлов. Такого рода имитационные эксперименты немислимы без использования численной модели, поскольку для их подготовки и проведения требуется большое количество времени и усилий.

Алгоритм распределения ролей на выходе дает набор деревьев, обладающих некоторыми особенностями по сравнению с деревьями, которые получаются при построении сети в спонтанном режиме. Рассмотрим типичные деревья, получающиеся при построении сети на примере сетки 10 на 10 узлов с координатором в центре и связностью по горизонтали и вертикали (без диагоналей). На рисунке ниже представлены результаты вычислений для спонтанной и оптимальной топологий сети. Узлы, имеющие больше детей, изображены более темным цветом.

Видно, что дерево оптимальной топологии менее разветвленное — у него есть несколько основных стволов из маршрутизаторов, которые редко разветвляются. Также каждый маршрутизатор имеет больше подчиненных узлов и больше узлов являются конечными устройствами (не имеют детей). Это положительно влияет на суммарное энергопотребление. Однако неразветвленное дерево имеет большую глубину в силу того, что оно не сбалансировано: обычно большая часть поддеревьев некоторого узла имеет глубину 1, так как большая часть детей любого маршрутизатора являются конечными устройствами. Время доставки сообщений возрастает с увеличением глубины дерева.

Для уменьшения времени доставки сообщений используется алгоритм распределения слотов суперфреймов. Этот алгоритм работает для некоторого узла тем лучше, чем меньше у него детей, являющихся маршрутизаторами, так как каждый дочерний маршрутизатор получает выделенный слот, увеличивая среднее время (по всем дочерним маршрутизаторам) между маяком родителя и маяком дочерних маршрутизаторов. Малое количество дочерних маршрутизаторов автоматически обеспечивается несбалансированностью дерева.

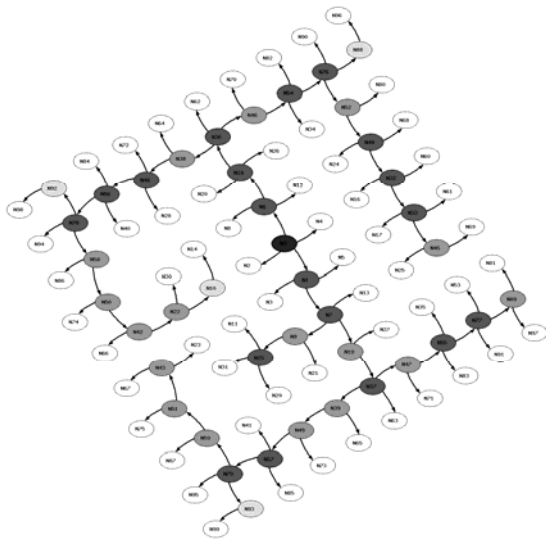


Рис. 7. Оптимальное дерево топологии

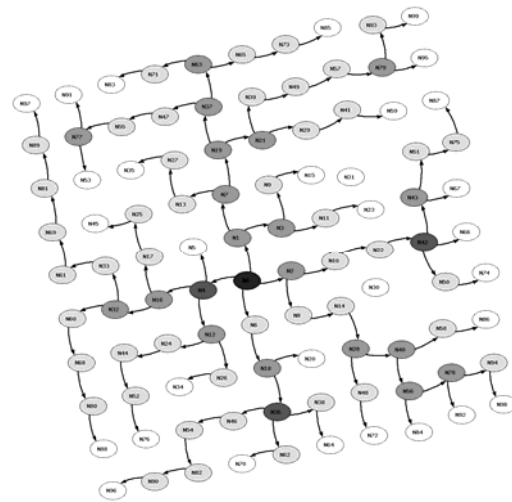


Рис. 8. Спонтанное дерево топологии

Для сравнения времени доставки сообщений были проведены имитационные эксперименты на квадратных сетках с горизонтальной и вертикальной связностью. Общие параметры экспериментов: $BO = 4$, $SO = 0$, все узлы поочередно генерируют пакеты, в среднем отправляется 1 пакет в секунду, размер данных пакета — 80 байт. Результаты экспериментов представлены на рисунке ниже: для 25 узлов слева, для 100 — посередине и для 225 — справа.

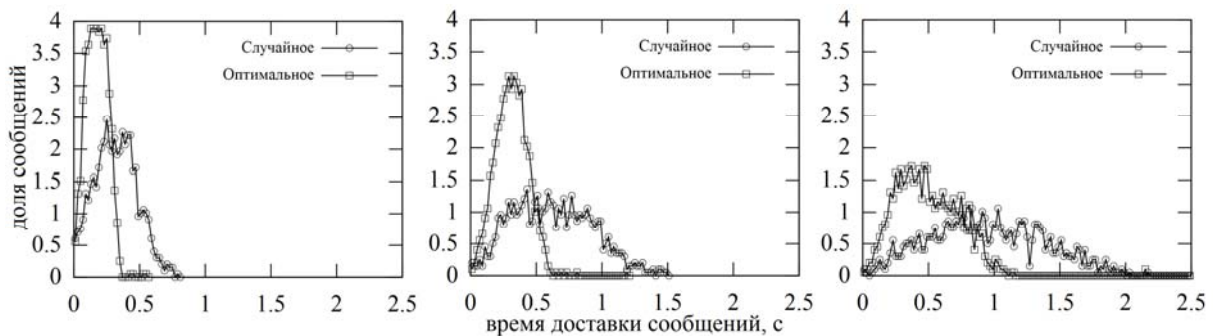


Рис. 9. Гистограммы распределения сообщений по временам их доставки

Из результатов следует, что применение обоих алгоритмов оптимизации дает ускорение времени доставки почти в два раза в широком диапазоне размеров сетей.

Исследование алгоритма распределения ролей на сетках

Для оценки эффективности алгоритма распределения ролей было проведено исследование его поведения на квадратных сетках из узлов. Варьировали два параметра: число узлов на стороне квадратной сетки и расстояние между узлами. Разные расстояния между узлами дают различный шаблон связности для узлов. Было исследовано четыре вида шаблона, которые представлены на рис. 10.

Для каждого шаблона была выполнена серия имитационных экспериментов для различного числа узлов. Общие параметры выбирались так же, как и для имитационных расчетов, описанных выше. В результате измерялось число независимых наборов маршрутизаторов, которое можно построить с помощью алгоритма распределения ролей на основе графа связности, получаемого при сканировании сети в спонтанном режиме. Это стандартная процедура инициализации сети для получения информации, необходимой при вычислении оптимальной топологии

и расписания. В сетях с высокой плотностью устройств увеличивается число коллизий, что может влиять на качество получаемого графа.

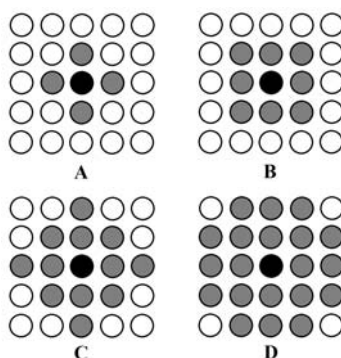


Рис. 10. Исследованные шаблоны связности узла

Нельзя так же забывать и про проблему скрытой станции, возникающую при использовании спонтанного расписания. Узлы, рядом с которыми оказались маршрутизаторы, одновременно передающие маяки, не могут присоединиться ни к одному из них. Это случилось с узлами N30 и N31 на рис. 8. Однако, даже при неполной информации о графе связности, в оптимальном режиме этим узлам удалось присоединиться к сети на рис. 7. Хотя один из них присоединился не оптимальным образом.

Результаты экспериментов приведены в табл. 3 ниже. «0» обозначает, что маршрутизаторы не нужны – топология «звезда» является оптимальной. В колонке max приведена оценка сверху максимально возможного числа независимых наборов маршрутизаторов для данного шаблона, рассчитанная по формуле (3.5).

Таблица 3. Число независимых наборов маршрутизаторов при различных параметрах сетки

Число узлов	9	16	25	36	49	64	81	100	121	max
Шаблон А	2	1	1	1	1	1	1	1	1	2
Шаблон В	0	2	2	2	2	2	2	2	2	3
Шаблон С	0	4	4	3	4	4	3	4	4	5
Шаблон D	0	7	7	5	6	6	6	5	6	7

Проанализировав полученные результаты, можно сделать вывод, что оценка формулы (3.5) может достигаться только для графов с максимальной глубиной дерева, равной двум. Однако и в больших сетях оценка оказывается завышенной всего на один набор маршрутизаторов. А упомянутые проблемы неполноты информации о графе связности при моделировании могут уменьшать число независимых наборов маршрутизаторов еще на единицу.

Заключение

Целью работы являлось построение и изучение алгоритмов, управляющих работой беспроводной сети ZigBee-устройств для минимизации энергопотребления и времени доставки сообщений. Предложены алгоритмы, использующие особенности протокола канального уровня и идеи о динамическом изменении сетевых ролей устройств совместно с установкой расписания.

Предложено две оптимизации, теоретически позволяющие увеличить время жизни конечных устройств в 6–7 раз в зависимости от параметров сети. Для проверки было выполнено экспериментальное измерение энергопотребления. Показано, что теоретическая оценка завышена на 13 %. Предложена схема динамического изменения ролей устройств, позволяющая приблизить время жизни сети ко времени жизни конечных устройств. Найдено теоретическое условие, обеспечивающее максимально возможное приближение. Описан алгоритм распреде-

ления ролей, который обеспечивает выполнение этого условия, и схема сетевого протокола, позволяющего его реализовать.

Проведены исследования алгоритма распределения ролей на квадратных сетках. Показано, что алгоритм может давать 2–7 независимых наборов маршрутизаторов, обеспечивающих связность топологии. Это позволяет, в конечном счете, увеличить время жизни сети в 1.7–3.8 раза.

Для ускорения доставки сообщений предложен алгоритм распределения слотов суперфреймов. Были проведены экспериментальные и численные измерения времени доставки сообщений. Результаты измерений хорошо согласуются между собой. Численные расчеты проводились при помощи сетевого симулятора ns-2, в котором нами была реализована модель нового стека протоколов. Показано, что алгоритм уменьшает среднее время доставки сообщений в 2–4 раза в зависимости от размера топологии и особенностей её построения.

Список литературы

- Кучерявый Е. А.* NS-2 как универсальное средство имитационного моделирования сетей связи. Tampere University of Technology, Telecommunications Laboratory, Tampere, Finland.
- Парамонов А. И.* Имитационное моделирование систем и сетей связи // ЛОНИИС, 2000.
- Banks J., Carson J. S., Nelson B. L., Nicol D. M.* Discrete-Event System Simulation, 5th Edition, Prentice Hall, 2009.
- Cuomo F., Luna S. D., Monaco U., Melodia T.* Routing In ZigBee: Benefits from Exploiting the IEEE 802.15.4 Association Tree. IEEE International Conference on Communications 2007, June 2007, pp. 3271–3276.
- Fall K., Varadhan K.* The ns Manual. 2008.
- IEEE TG 15.4. Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs), IEEE standard for Information Technology. IEEE-SA Standards Board, 2003.
- Koubâa A., Alves M., Tovar E.* A Comprehensive Simulation Study of Slotted CSMA/CA for IEEE 802.15.4 Wireless Sensor Networks. Proceedings of the 6th IEEE International Workshop on Factory Communication Systems (WFCS 2006), Torino (Italy), 2006.
- Ramachandran I., Das A. K., Roy S.* Analysis of the contention access period of IEEE 802.15.4 MAC // ACM Transact. on Sensor Networks. — 2007. — Vol. 3, no 1.
- Ousterhout J.* Scripting: High-level programming for the 21st century. IEEE Computer, March 1998, N31(3), pp. 23–30.
- Zheng J., Lee M.* A Comprehensive Performance Study of IEEE 802.15.4. IEEE Press, 2004.
- ZigBee Specification. ZigBee Alliance Std., 2007.