Разработка модуля беспроводной передачи телеметрических данных в диапазоне частот 2,4 ГГц

(часть 3)

Александр Алый (Москва)

Универсальный модуль предполагает многообразие применений, но создание эффективного коммуникационного протокола – задача очень трудная. На помощь приходят спецификации, воплощающие коллективный труд сообщества разработчиков. ZigBee – лучший пример такой спецификации, и в этой статье автор анализирует возможности спецификации ZigBee для реализации программного обеспечения радиомодуля. В статье также приведён пример разработки недорогого многофункционального радиомодуля, встраиваемого в распределённые системы управления бытовыми приборами.

ZigBee как спецификация конкурирует на рынке с большим количеством так называемых «проприетарных», иначе говоря, частных или фирменных спецификаций, описывающих организацию взаимодействия радиомодулей в локальных сетях. Многие производители радиомодулей снабжают их собственным программным обеспечением, реализующим несложные протоколы взаимодействия. Характерные примеры: фирма Micrel бесплатно предоставляет в исходных текстах сетевой протокол для своих радиомодулей; фирма Microchip предлагает в исходных текстах протокол MiWi для той же аппаратной платформы, что и ZigBee; фирма Nokia намерена скоро опубликовать спецификацию Wibree для новой аппаратной платформы; существует амбициозный проект Zensys Z-Wave, прямо противопоставляющий себя ZigBee, и рабочая группа, работающая над переносом стека TCP/IP на беспроводную платформу, аналогичную ZigBee.

Существует мнение, что использовать ZigBee следует только для того, чтобы обеспечить совместимость с ZigBee-устройствами сторонних фирм. Однако критики ZigBee из лагеря конкурентов утверждают, что именно совместимость является одной из слабых сторон этой спецификации. В ZigBee так много различных параметров, возможностей реализации и уровней криптографической защиты, что, по их словам, беспроблемное выполнение единой задачи модулями разных производителей в одной сети маловероятно. И действительно, на январь 2007 г. из четырёх сертифицированных альянсом ZigBee продуктов не существует ни одного законченного устройства для потребительского рынка; к сожалению, не сертифицированные потребительские продукты не снабжены даже элементарными сведениями о поддерживаемых профилях.

С другой стороны, разрабатывать и производить устройства с поддержкой ZigBee могут только члены альянса ZigBee. Стать неполноправным членом альянса (Adopter member) можно уплатив минимальный членский взнос в размере 3,5 тыс. долл. за год пребывания в альянсе, при этом сертификация каждого продукта потребует ещё 1 тыс. долл. К счастью, разрешено использовать спецификацию в некоммерческих целях, поэтому ниже мы проанализируем технические детали и способы адаптации ZigBee для нашего проекта.

Преимущества ZigBee

В спецификации ZigBee воплощён опыт передовых коллективов разработчиков, работающих в сфере локальных, низкоскоростных и экономичных беспроводных сетей. Она интегрировала лучшие решения, реализованные при построении и эксплуатации локальных распределённых систем управления. Учтено практически все:

- пространственная масштабируемость – количество узлов сети можно увеличивать до тысячи и более;
- функциональная масштабируемость – одна сеть может использоваться во многих системах управления одновременно, и их количество и разнообразие можно наращивать без изменения программного обеспечения и перенастройки маршрутизаторов и координатора сети;
- лёгкость установки и наладки – конечные устройства сети сами объявляют о предоставляемых ими сервисах и возможностях и через координатора находят устройства, с которыми они должны взаимодействовать для выполнения целевых задач управления;
- лёгкость наблюдения за сетью и оптимизация её структуры с помощью специальных методов администрирования;
- решение проблем живучести сети – при потере связи с узлами сеть перестраивается, изменяя структуру и маршрутизацию. Можно предусмотреть и дублирование координатора при потере связи с основным координатором;
- решение проблем качества связи – при недостаточном качестве связи можно устанавливать дополнительные маршрутизаторы;
- высокая защищённость информации – криптографическая защита на трёх уровнях стека. Аутентификация узлов сети;
- открытость для реализации интеграторами собственных протоколов и технологий на базе сервисов, предоставляемых ZigBee.

Спецификация ZigBee уделяет большое внимание энергосберегающим режимам работы сети. Например, конечные устройства сети большую часть времени остаются в «спящем» состоянии. Чтобы работа сети не прерывалась, каждое устройство ассоци-

ировано с выделенным маршрутизатором, который не переходит в спящий режим и берёт на себя обязанность откликаться на запросы к устройству, пока оно спит. Маршрутизатор накапливает все пакеты, предназначенные для устройства, и передаёт их при первой возможности.

Спецификация ZigBee

Спецификация постоянно совершенствуется. Недавно на сайте www.zigbee.org появилась публикация спецификации ZigBee 2006 с дополненными схемами адресации и маршрутизации; далее планируется ввести в ZigBee технологию динамической смены частоты (сейчас смена частотного канала – процедура исключительная, инициируемая самим пользователем) и т.д. Поэтому мы рассмотрим особенности версии спецификации от 2004 г., для которой существуют открытые исходные коды.

Спецификация ZigBee весьма формализована, и это облегчает программную реализацию и тестирование, но объём документа становится чрезвычайно большим. Число страниц спецификации ZigBee вместе с сопутствующим стандартом IEEE 802.15.4 составляет более 1000.

Необычна терминология спецификации: она выдержана в абстрактном стиле, независимом от какого-либо языка программирования. Поскольку в статье подразумевается разработка на языке Си, автор оставляет за собой право переводить термины спецификации на язык Си, а нумерацию битов приводить так, как принято при программировании (справа налево).

Следует обратить внимание на термины «кластер» (cluster), «артибут» (attribute) и «примитив» (primitive). Термин «кластер» можно встретить в двух значениях: первое – это название характерного скопления узлов сети, второе значение – группа атрибутов. Атрибут, в программной реализации, – некая переменная, значение, параметр и т.п., отражающий внутреннее состояние программы или сигналов на внешней периферии. Причём кластеры могут быть входные, т.е. содержащие переменные только для записи, либо выходные, т.е. содержащие переменные для чтения.

Термин «примитив» перешёл в спецификацию ZigBee из стандарта IEEE 802.15.4 и означает некую структуру данных, передаваемую вместе с идентификатором команды от одного уровня стека ZigBee другому. В реализации на языке Си это может быть глобально объявленная структура, заполняемая вызывающей функцией одного уровня и передаваемая по указателю вызываемой функции другого уровня.

Архитектура ZigBee

Итак, спецификация определяет архитектуру ZigBee стека. Стек – это конструкция из слоёв (см. рис. 1), приведённая в соответствие с требованиями семиуровневой модели OSI.

Каждый слой содержит набор специфических функций (сервисов), вызываемых из верхних слоёв. На языке Си, чаще всего, – это отдельные программные модули. Вместо набора публичных функций, в модуле обычно реализованы только одна-две публичные функции, способные принимать в качестве аргументов разнообразные структуры данных вместе с идентификаторами команды, по которым функции определяют, что делать с данными. Такие же структуры функции выдают в качестве результата. Спецификация ZigBee написана в расчёте на такую реализацию стека. Публичные функции в каждом слое названы точками доступа к сервису – Service Access Point (SAP) и существуют парами: одна функция получает команды передачи данных, другая – команды управления. Передаваемые в виде структур данных аргументы называются примитивами и подробно описываются спецификацией, включая типы данных и назначение членов структур. SAP можно реализовать и в виде одной публичной функции.

Уровни MAC и PHY не описаны в спецификации ZigBee, но приведены в стандарте IEEE 802.15.4 (2003 г.). В спецификации имеются комментарии к реализации MAC-уровня, которые следует учитывать. Следует отметить, что стандарт IEEE 802.15.4 был создан для поддержки сетей малого радиуса – Personal Area Network (PAN) со структурой «точка-точка» или «звезда», т.е. изначально не предназначался для сетей типа ZigBee.

В спецификации ZigBee предусмотрено шифрование на трёх уровнях: MAC, NWK и APS. Соответственно, содержимое пакетов этих уровней может быть зашифровано независимо друг от друга. Кроме шифрования, спецификация определяет процедуры аутентификации, которые не позволяют неопознанным узлам подключаться к сети. Также определены процедуры обмена ключами шифрования и порядок управления ключами в сети.

Адресация в ZigBee

Чтобы пояснить состав и назначение сервисов стека, на рис. 2 представлена структура адресации в ZigBee.

Поскольку предусматривается совместное сосуществование нескольких сетей ZigBee на одном частотном канале, для их разделения введён 16-битный идентификатор сети (Personal area network ID, PAN ID). Все ZigBee-модули снабжены уникальным 64-битным идентификатором. Но передача такого длинного идентификатора требует больших накладных расходов, и предоставляемый им диапазон адресов явно избыточен. Поэтому в ZigBee был введён короткий 16-битный сетевой адрес устройства, назначаемый координатором сети при её организации.

Спецификация допускает решение одной сетью множества прикладных задач, не связанных друг с другом. Для различения пакетов по приложениям, для которых они предназначены, используются 8-битные номера конечных точек. Приложения надо понимать в широком смысле – это может быть управление освещением, или канал передачи данных во внешний шлюз сети другого типа, или контур управления отоплением и т.д.

Чтобы обеспечить полную совместимость устройств разных производителей и их способность взаимодействовать в рамках единого распределённого приложения, было введено понятие профилей, которые различаются в пакетах с помощью 16-битного идентификатора. Профиль описывает ряд технических параметров, соглашений о структурах данных и форматах сообщений, которых должны придерживаться изготовители, чтобы их изделия были совместимы. К сожалению, альянс ZigBee пока имеет только один стандартный профиль – управления бытовым освещением. Но частные, не стандартизированные профили также должны иметь уникальные идентификаторы, поэтому их выдачей занимается сам альянс ZigBee.

Кластеры представляют собой контейнеры для атрибутов и были введены для администрирования групп родственных атрибутов. Чтобы каж-

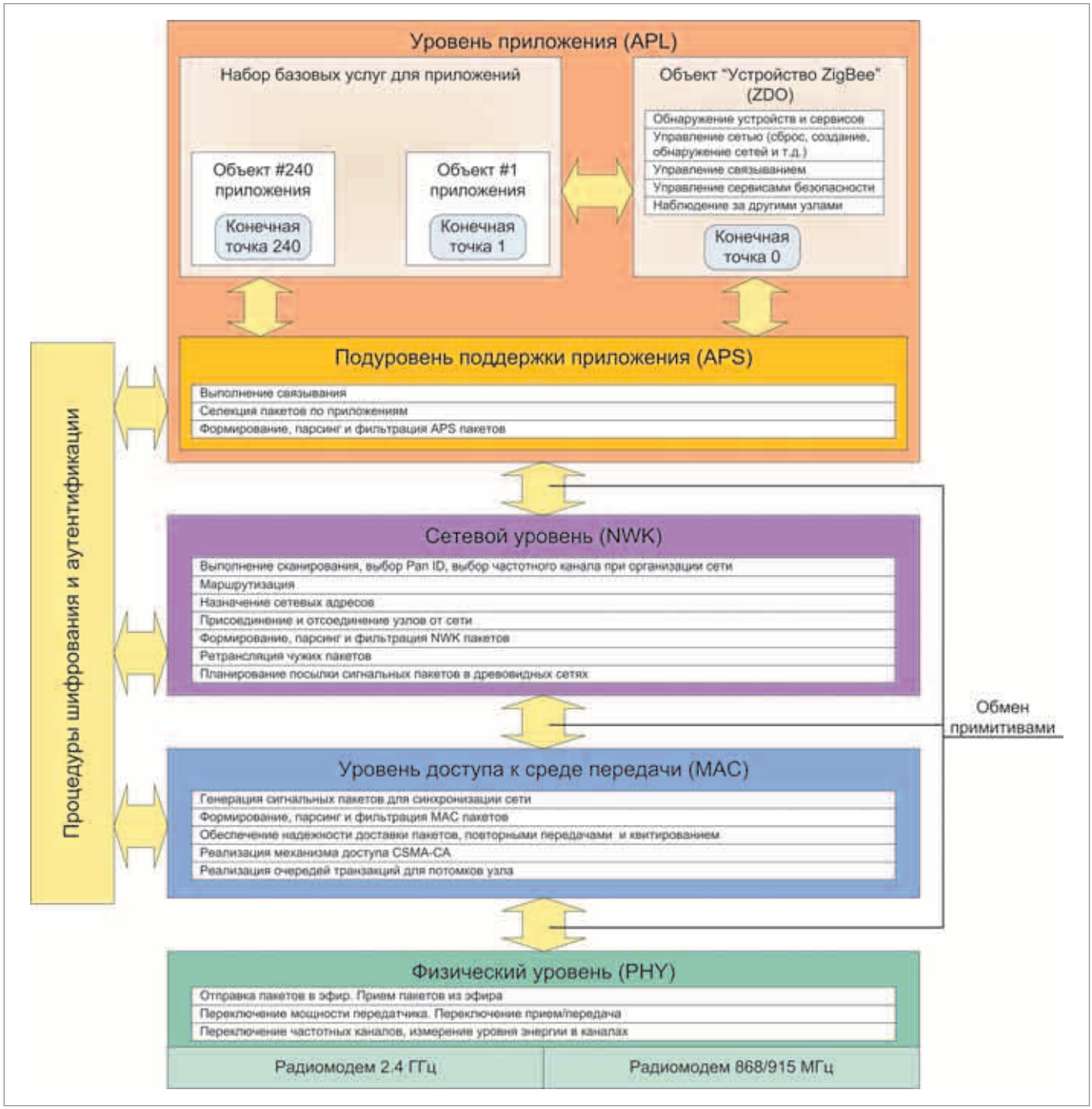

Рис. 1. Стек ZigBee

дый раз не перечислять атрибуты, можно сослаться на них, используя номер кластера. Кластеры применяются в технологии связывания и косвенной адресации, которая будет описана ниже. Идентификатор кластера имеет длину 8 бит. Смысл атрибута поясняется на рис. 2.

Топология сети

С точки зрения администратора, топология сети ZigBee всегда представляет собой иерархическое дерево, как показано на рис. 3 (дерево, как принято, изображено перевернутым). В основании находится координатор. Маршрутизаторы (роутеры) добавляют ветви в структуру сети и увеличивают её глубину. При их отсутствии топология сети вырождается в звезду.

К маршрутизаторам и координатору могут подключаться конечные устройства или устройства с ограниченными функциями (Reduced function device, RFD). Маршрутизаторы и координатор являются полнофункциональными устройствами (Full function device, FFD). Конечное устройство может быть необъявленным маршрутизатором, чтобы иметь возможность присоединиться к сети. Такой вариант упоминается ниже.

Процесс формирования сети

Даже если все устройства ZigBee включены и могут вести общение друг с другом, сеть не возникнет, пока не появится устройство, взявшее на себя функции координатора. Координатор – единственное устройство, которое может инициировать начало формирования сети. Координатор начинает с определения уровня помех на всех доступных частотных каналах, выбирает канал с наименьшим уровнем помех и определяет наличие в нём других работающих сетей ZigBee, запрашивая их идентификаторы. Затем координа-

тор случайным образом выбирает идентификатор для своей сети из диапазона 0x0000 – 0x3FFE, чтобы он не совпадал с идентификаторами других сетей в том же частотном диапазоне. Сетевой 16-битный адрес координатора всегда равен 0x0000. После этого координатор разрешает присоединяться к своей сети другим устройствам.

Другие устройства, до этого момента сканировавшие эфир на предмет доступных сетей, получают разрешение от координатора на присоединение к его сети по принципу ветвления. Присоединив некоторое количество конечных устройств и маршрутизаторов, координатор отказывается присоединять непосредственно к себе остальных, вынуждая их искать уже присоединившиеся маршрутизаторы (конечные устройства не могут никого присоединять). Таким образом продолжается ветвление соединений. Из кандидатов в родительские узлы предпочтение отдаётся устройствам с наименьшим числом шагов ретрансляции до координатора.

Реальные профили приложений жёстко ограничивают максимальное количество уровней в создаваемой древовидной структуре. Может возникнуть ситуация, когда очередному маршрутизатору соседи отказывают в присоединении. В таких случаях маршрутизаторы могут понизить свой статус до конечных устройств, поскольку на их присоединение ограничения менее жёсткие. На практике может сложиться ситуация, когда останутся устройства, которым соседи отказали в присоединении, но эта проблема должна решаться вне рамок спецификации, поскольку динамическая балансировка древовидной структуры в ZigBee не поддерживается.

В спецификации также предусмотрен способ предварительно запрограммированного подключения устройств к маршрутизаторам и координатору.

Во время присоединения к сети устройство получает уникальный в пределах данной сети 16-разрядный адрес. В спецификации описаны два способа назначения адресов: распределённый механизм назначения адресов (см. рис. 3) и назначение адресов уровнем приложения. По умолчанию применяется первый способ.

Профиль приложения, которому подчиняется сеть, среди прочего, определяет три константы:

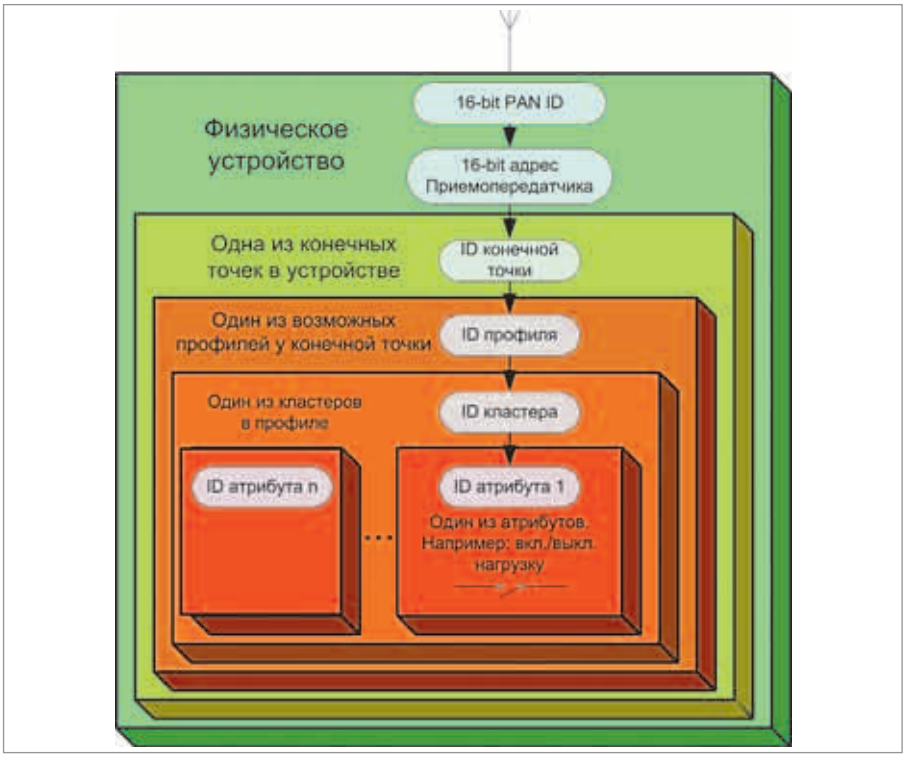

Рис. 2. Структура адресации в ZigBee

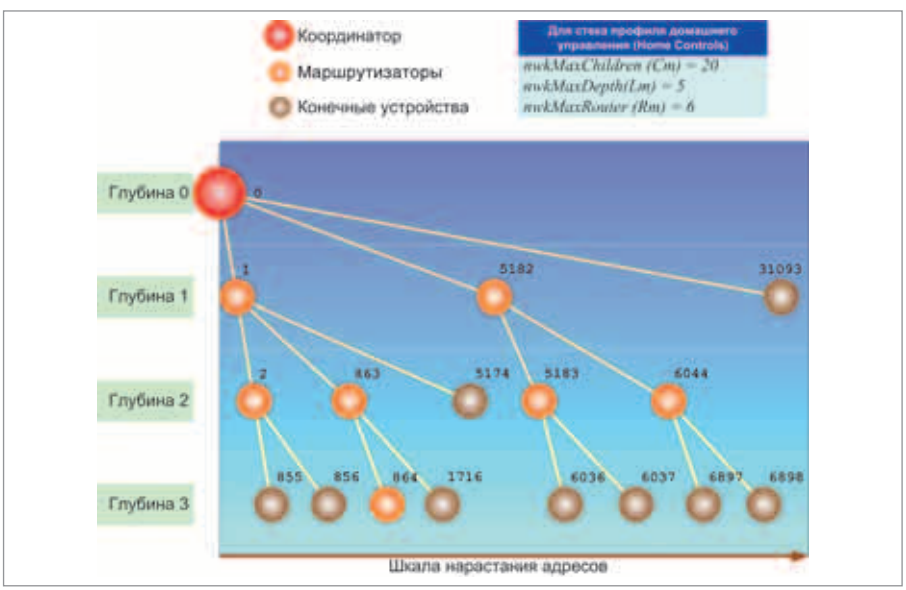

Рис. 3. Структура и пример адресации в сети ZigBee

- *nwkMaxChildren* (C_m) – максимальное число наследников у каждого узла;
- *nwkMaxDepth* (L_m) – максимальная глубина сети;
- *nwkMaxRouter* (R_m) – максимальное число маршрутизаторов среди наследников каждого узла.

На основе этих констант по приведённой ниже формуле (при $R_m > 1$) рассчитывается параметр C_{skip} для каждого уровня глубины d:

$$C_{skip}(d) = \frac{1 + C_m - R_m - C_m R_m^{L_m - d - 1}}{1 - R_m}.$$

Параметр C_{skip} равен интервалу адресов, которым на данной глубине может распоряжаться маршрутизатор для присвоения адресов своим наследникам.

Вычислив параметр C_{skip}, маршрутизатор или координатор, получивший собственный адрес, начинают распределять адреса.

Для наследников типа «маршрутизатор» адреса вычисляются по формуле:

$$A_n = A_{parent} + C_{skip}(d)(n - 1) + 1,$$

где A_{parent} – собственный адрес, A_n – адрес n-го маршрутизатора-наследника, n – порядковый номер наследника от 1 до R_m, d – текущая глубина.

Для координатора глубина всегда равна 0.

Для наследников типа «конечное устройство» адреса вычисляются по формуле:

$$A_n = A_{\text{parent}} + C_{\text{skip}}(d)R_m + n,$$

где n – порядковый номер наследника от 1 до $(C_m - R_m)$.

Такая схема назначения адресов приводит к очень простому критерию выбора направления маршрутизации по древовидной структуре. Когда маршрутизатор решает, куда послать ретранслируемое сообщение, он проверяет адрес назначения сообщения DestAddr на соответствие диапазону:

$$\text{LocalAddr} < \text{DestAddr} < \text{LocalAddr} + C_{\text{skip}}(d - 1),$$

где LocalAddr – собственный адрес маршрутизатора, d – его глубина в древовидной структуре. Если адрес находится в указанном диапазоне, то сообщение надо передавать вниз по древовидной структуре, если нет – то вверх.

Маршрутизация в сети

Первый и очевидный способ маршрутизации в ZigBee-сетях – иерархическая маршрутизация по ветвям древовидной структуры. Как было отмечено выше, существует простое правило, с помощью которого маршрутизатор может определить направление передачи сообщения. Если сообщение надо передавать вверх, маршрутизатор находит в предварительно созданной таблице соседей (см. табл. 1) адрес родительского узла и передаёт ему сообщение. Если сообщение надо передать вниз (для прямого потомка), его адрес будет равен адресу назначения. Если сообщение надо передать вниз по цепочке, адрес следующего маршрутизатора вычисляется по формуле:

$$\text{NextAddr} = \text{LocalAddr} + 1 + \text{trunc}[(\text{DestAddr} - (\text{LocalAddr} + 1))/C_{\text{skip}}(d)]C_{\text{skip}}(d),$$

где NextAddr – адрес следующего узла, куда надо переслать сообщение, DestAddr – адрес назначения, LocalAddr – собственный адрес маршрутизатора, trunc – функция взятия целого от деления.

По умолчанию иерархической маршрутизацией пользуются все маршрутизаторы и координатор, если у них закончились ресурсы для поддержки других типов маршрутизации.

Другим видом маршрутизации является сеточная (mesh) маршрутизация. Она поддерживается только полнофункциональными устройствами. Конечные устройства передают пакеты данных только своим родительским узлам, поскольку они не имеют таблиц маршрутизации. Полнофункциональные устройства, получив пакет данных, не предназначенный для узла-потомка или узла-родителя, и не имея соответствующей записи в таблице маршрутизации (см. табл. 2 и 3), инициируют процедуру обнаружения маршрута.

Обнаружение маршрута начинается с широковещательной рассылки команд всем маршрутизаторам (в пределах радиовидимости). Маршрутизаторы, принявшие команду, создают у себя временные записи о принятом запросе (см. табл. 4) и со случайно выбранной задержкой ретранслируют команду. Чтобы широковещательная ретрансляция не превратилась в «радиошторм», пакеты снабжены счётчиком ретрансляций, который уменьшается на единицу при передаче пакета через маршрутизатор. Как показано на рис. 4, возможно существование нескольких маршрутов прохождения пакетов до узла назначения, но каждый маршрутизатор отбрасывает пакеты с командами обнаружения маршрута, которые имеют большую стоимость пути, чем зафиксированная у предыдущих паке-

Таблица 1. Таблица соседей

Имя поля таблицы	Длина поля (платформенно-зависимая)	Описание
Расширенный адрес (Extended address)	8 байт	Уникальный 64-битный идентификатор устройства или IEEE-адрес. Это поле заполняется, если сосед является наследником или родителем данного узла
Сетевой адрес (Network address)	2 байта	16-битный сетевой адрес соседа
Тип устройства, к которому относится сосед (Device type)	Тип	0 – координатор 1 – маршрутизатор 2 – конечное устройство
RxOnWhenIdle	Boolean	TRUE – если у устройства всегда включен приёмник
Отношение с соседом (Relationship)	Integer	0 – сосед является родителем 1 – сосед является потомком 2 – сосед является потомком общего родителя 3 – сосед не является ни одним из вышеперечисленных
Ошибки передачи (Transmit Failures)	Integer	Принимает значение от 0 до 255 и отражает количество неудачных передач соседу
Оценка качества связи (LQI)	Integer	Значение выдаётся сервисом уровня PHY
Время прихода последнего сигнального пакета (Incoming beacon timestamp)	Integer	Время измеряется в интервалах, равных времени передачи одного символа в терминологии уровня PHY. Поле не обязательное
Смещение по времени передачи сигнального пакета (Beacon transmission time offset)	Integer	Смещение по времени выдачи сигнального пакета соседом по отношению ко времени выдачи сигнального пакета его родителем. Поле не обязательное

Таблица 2. Дополнительные поля таблицы соседей, используемые в течение обнаружения и присоединения к сети

Имя поля таблицы	Длина поля (платформенно-зависимая)	Описание
Расширенный идентификатор сети (Extended PAN ID)	Integer	Уникальный 64-битный идентификатор сети соседа. Обычно по умолчанию равен 64-битному идентификатору координатора сети
Логический канал (Logical channel)	Integer	Логический канал, на котором работает сеть
Глубина (Depth)	Integer	Глубина, на которой в древовидной структуре сети находится сосед
Параметр, отражающий частоту передачи сигнальных пакетов (Beacon order)	Integer	Может находиться в диапазоне от 0 до 15. Если равен 15, то сигнальные пакеты не передаются
Сосед допускает присоединение (Permit joining)	Boolean	TRUE – если сосед допускает присоединение
Потенциальный родитель (Potential parent)	Integer	0 – сосед не может быть родителем 1 – сосед может быть родителем

тов. Если пакет имеет ту же стоимость пути, данные в таблице обнаружения маршрута обновляются. Стоимость пути содержится в самом пакете и обновляется всякий раз, когда он ретранслируется маршрутизатором.

Спецификация предлагает несколько вариантов расчёта стоимости пути. Самый простой – подсчёт ретрансляций по маршруту, и этот способ был принят в примере на рис. 4. Более сложный способ – вычисление стоимости пути по сумме параметров качества связи между узлами по маршруту LQI (см. таблицу 1). И, наконец, самый правильный и трудно реализуемый способ – суммирование функций вероятности прохождения пакетов между узлами, которая, в свою очередь, вычисляется путём накопления статистических данных.

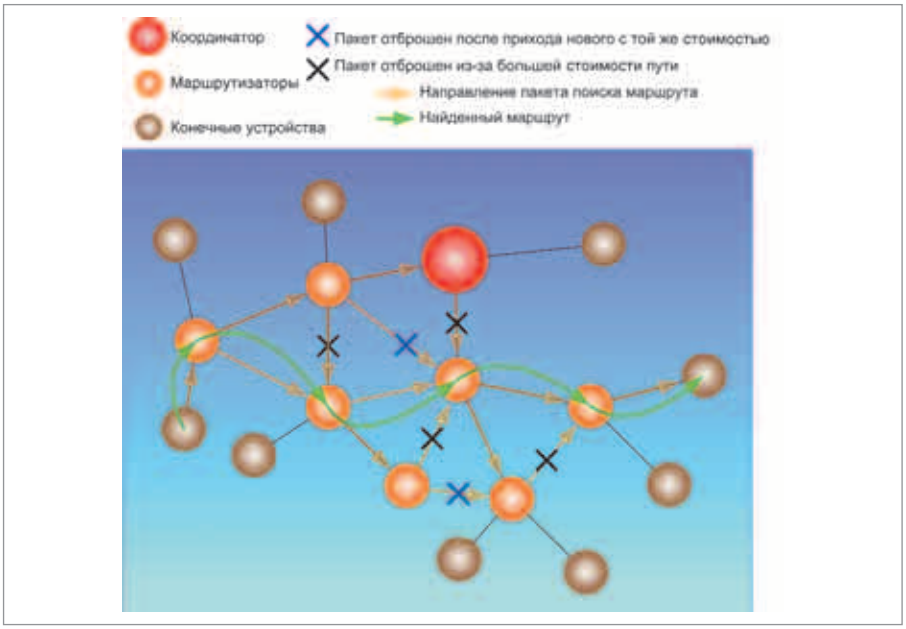

Рис. 4. Пример сеточной маршрутизации в сети ZigBee

Маршрутизатор, являющийся пунктом назначения для пакетов команды обнаружения пути или родительским узлом пункта назначения типа RFD, при получении пакета отвечает другим пакетом, который содержит команду подтверждения. Этот пакет отправляется адресно и проходит уже проложенным путём в обратном направлении. К этому моменту по пути его следования во всех промежуточных маршрутизаторах будут созданы правильные записи маршрутизации, и проходящий пакет подтверждения будет устанавливать их статус как активный. Дойдя до инициатора обнаружения маршрута, пакет подтверждения завершает процесс формирования маршрута. После этого уничтожаются все временные записи в таблицах обнаружения маршрута во всех промежуточных узлах, а записи таблиц маршрутизации в узлах сохраняются в долговременной памяти.

Описанный выше алгоритм сеточной маршрутизации создаёт однонаправленный путь. Если в стеке ZigBee константа nwkSymLink установлена как TRUE, этот же путь будет использоваться и для передачи в обратном направлении, иначе для обнаружения обратного пути потребуется запустить алгоритм маршрутизации заново. Очевидно, что обратный путь может не совпадать с прямым даже при расчёте цены по методу простого счётчика переходов, поскольку ветвления по маршруту выбираются на основе генератора случайных задержек.

В спецификации 2006 г. введены ещё два способа маршрутизации. Это групповая маршрутизация и маршрутизация типа «многие к одному». Необходимость в них была вызвана обнаруженной при некоторых условиях нестабильностью больших сетей с сеточной маршрутизацией.

Продолжение следует

Таблица 3. Содержание таблицы маршрутизации

Имя поля таблицы	Длина поля	Описание
Адрес назначения (Destination address)	2 байта	Содержит 16-битный сетевой адрес или идентификатор группы. Если устройство назначения является маршрутизатором или координатором, поле содержит действительный адрес устройства. Если устройство назначения является конечным устройством, поле содержит адрес предка этого устройства
Статус (Status)	3 бита	0 – активный 1 – идёт поиск маршрута 2 – ошибка поиска маршрута 3 – не активный 4 – идёт процесс подтверждения 5 – 7 – зарезервировано
Многие к одному (Many-to-one)	1 бит	Флаг, указывающий на то, что устройство назначения – концентратор, выдающий сообщения «многие к одному»
Требуется запись маршрута (Route record required)	1 бит	Флаг, указывающий на то, что перед посылкой следующего пакета данных в устройство назначения надо передать команду записи маршрута
Флаг идентификатора группы (GroupID flag)	1 бит	Флаг, указывающий на то, что поле адреса назначения содержит идентификатор группы
Следующий адрес перехода (Next-hop address)	2 байта	16-битный сетевой адрес следующего устройства по пути к адресу назначения

Таблица 4. Содержание таблицы обнаружения маршрута

Имя поля таблицы	Длина поля	Описание
Идентификатор запроса обнаружения маршрута (Route reques ID)	1 байт	Каждый запрос от какого-либо узла на обнаружение маршрута имеет идентификатор. Каждый следующий запрос имеет другой идентификатор
Сетевой адрес инициатора запроса (Source address)	2 байта	16-битный сетевой адрес узла инициатора запроса на обнаружение маршрута
Адрес узла отправителя (Sender address)	2 байта	16-битный сетевой адрес узла, от которого пришёл пакет с запросом на обнаружение маршрута с наименьшей стоимостью пути. При этом пакет имеет те же идентификатор запроса и адрес инициатора
Цена пройденного пути (Forward Cost)	1 байт	Суммарная стоимость пути от инициатора запроса до текущего узла
Цена оставшегося пути (Risidual Cost)	1 байт	Суммарная стоимость пути от текущего узла до узла, к которому прокладывается маршрут
Время до истечения срока существования данной записи (Expiration time)	2 байта	Таймер, отсчитывающий время в миллисекундах до того, как запись будет удалена, если она к тому времени ещё не будет удалена после обнаружения маршрута. По умолчанию выделено 10 с