М. Д. Недев

Протоколы и алгоритмы в LoWPAN-сетях

Научный руководитель: к.т.н. Ю. В. Шевчук

Аннотация. Работа рассматривает алгоритмы и протоколы, применяемые в сенсорной сети, разработанной в ИПС РАН. Особое внимание уделено вопросам отладки элементов сенсорной сети на компьютерной модели, организации туннеля между LoWPAN-сетью и сетью Интернет. Также в статье описывается техника доступа к разделяемой радио-среде, основанная на расписаниях сеансов связи.

1. Введение

В последние годы заметно возросло внимание к такой области исследований, как LoWPAN-сети. Термин "LoWPAN" расшифровывается как Low power Wireless Personal Area Networks и используется для обозначения беспроводных сетей, имеющих ограниченные ресурсы. Такие сети также называют «сенсорными». Достижения в электронике сделали возможным использование сенсорных сетей в задачах мониторинга и управления для различных территориально-распределенных объектов, от химических производств до «умного дома».

Беспроводная сенсорная сеть состоит из десятков (иногда и сотен, и тысяч) небольших устройств, оснащенных датчиками и радиотрансивером. К ней выдвигаются такие требования, как: отказоустойчивость, масштабируемость, низкая стоимость. Узлы сенсорной сети, как правило, имеют ограниченные ресурсы: батарейное питание, малое количество памяти и низкие вычислительные способности.

Кроме того, современные сетевые протоколы не всегда способны эффективно работать в следующих условиях:

- в зоне доступа одновременно находится много узлов;
- высокие потери пакетов в сети;
- различные характеристики узлов (некоторые имеют батарейное питание, другие питаются от сети).

Эти обстоятельства указали на необходимость создания новых протоколов и алгоритмов.

Консорциумом IEEE был разработан стандарт [1], описывающий некоторые механизмы работы LoWPAN-сетей. В рамках проекта «Ботик-сенсор», ведущегося в ИЦМС ИПС РАН, принято решение следовать стандарту там, где это возможно. В случаях, когда стандарт не удовлетворяет требованиям проекта, предлагаются альтернативные методы.

2. Из чего состоит сенсорная сеть

На рис. 1 показан пример сенсорной сети. Ее структура предполагает использование управляющей станции (УС) (PC с операционной системой Linux), шлюза и рабочих узлов. УС соединена со шлюзом по проводному Ethernet, вокруг шлюза действует радио-сеть 802.15.4. Обязанности управляющей станции заключаются в сборе данных с узлов и координации их действий. От шлюза требуется работать одновременно в двух средах (Ethernet и радио) и осуществлять перевод пакетов из одной среды в другую. Остальные узлы должны выполнять указания, которые им передала УС.

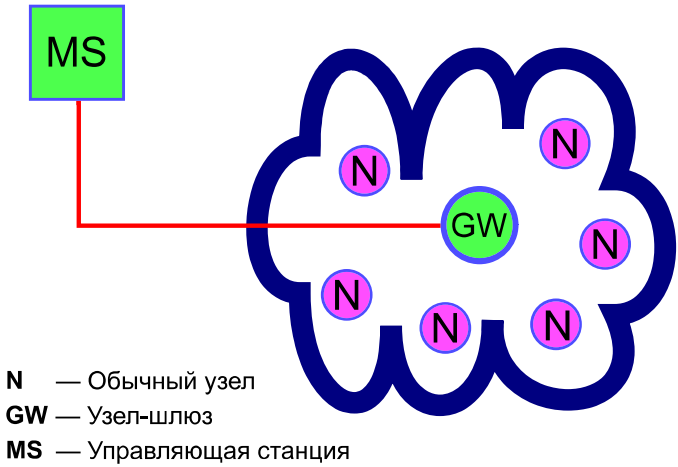

Рис. 1. Пример сенсорной сети

Радио-сеть на рисунке построена по топологии «звезда»: узлы не осуществляют маршрутизацию пакетов.

2.1. Contiki и Cooja

В качестве операционной системы была выбрана ОС Contiki [2], спроектированная специально для устройств, имеющих очень ограниченные ресурсы. Contiki написана на языке C, легко может быть портирована на различные платформы и включает в себя TCP/IP стек uIP, полностью совместимый с протоколом IPv6.

Выбор именно этой операционной системы был обусловлен еще и тем, что с ней поставляется удобная среда моделирования “Cooja” со многими возможностями. При работе с сенсорными сетями ее использование очень полезно: разработка, тестирование и отладка программ на реальных устройствах — это сложный и утомительный процесс. Моделирование позволяет аккуратно, по шагам, проводить те эксперименты, чистоту которых невозможно было бы проконтролировать при работе с реальными устройствами. Например — реализация и тестирование алгоритма маршрутизации.

3. Сетевой стек

На рис. 2 показаны сетевой стек узла сенсорной сети (слева) и абстрактная сетевая модель OSI (справа). Рассмотрим каждый из уровней стека (снизу вверх):

(1) Физический уровень. Он осуществляет передачу сигналов в радиоэфир, а также их прием и преобразование в биты данных. В нашем случае, кроме реального уровня (трансивер AT86RF231), существует также и моделируемый (среда Cooja), используемый для отладки.

(2) Канальный уровень. Поскольку было принято решение об использовании моделирования при разработке программ, то для сохранения независимости верхних уровней были созданы два драйвера: для реальных устройств и для среды моделирования. Драйверы предоставляют одинаковый интерфейс и реализуют базовые функции, например: посылка и перепосылка пакета, включение и выключение приемника, установка параметров трансивера. Реализации функций в каждом из драйверов различны, но результат их выполнения должен быть одинаков. Для систематизации обращений

к lowpan-драйверу вводится уровень расписаний, он накапливает исходящие пакеты, классифицирует входящие. АУ-протокол (инициализации) также относится к этому уровню.

(3) Сетевой уровень. По ряду причин выбран протокол IPv6. Из-за ограничений на MTU мы вынуждены ввести уровень адаптации 6lowpan.

(4) Транспортный уровень. Используется протокол UDP, предназначенный для доставки данных без ошибок, но не гарантирующий доставку пакета. Это удовлетворяет нашим требованиям.

(5) Уровень приложения. Этот уровень ответственен за прием узлом команд управления и подготовку результатов их выполнения.

Рис. 2. Сетевой стек узла

Об АУ-протоколе, уровне адаптации 6lowpan и расписаниях будет подробнее рассказано ниже. Описание протокола Etherbox находится за рамками данной работы.

3.1. АУ-протокол

Стандарт 802.15.4 определяет процедуры сканирования эфира для подключения новых устройств к сети. Однако, с ними есть некоторые проблемы: во-первых, не ясно, как выбирать сенсорную сеть среди нескольких найденных, во-вторых, сканирование в некоторых случаях требует серьезных энергозатрат.

В ИПС РАН разработана нестандартная процедура подключения узлов к сети:

(1) Новое устройство периодически передает маленькие АУ-пакеты по всем каналам. В АУ-пакете присутствует идентификатор устройства и показания его внутренних часов.
(2) Также новое устройство периодически включает режим приема на разных каналах. Расписание приема может быть вычислено управляющей станцией на основании паспортных данных устройства.
(3) УС может дать старым устройствам команду приема АУ-пакетов. Тогда старые устройства включают приемники на заданное количество секунд, выбирают из полученных пакетов не только свои, но и АУ-пакеты, и запоминают последние вместе с показаниями собственных часов на момент приема. УС собирает результаты и понимает, какие новые устройства есть в сети и к какому из старых они ближе.
(4) УС посылает старому устройству, которое ближе всех к новому, команду передать указанный пакет в указанное время на указанном канале. Пакет зашифрован внутренним кодом нового устройства и содержит полный набор конфигурационных данных: каналы, AES-ключи, расписание работы, соседей, маршруты.
(5) Успешно приняв конфигурационные данные, новое устройство становится участником сети.

3.2. 6lowpan

В нашей сети применяется протокол межсетевого обмена IPv6, что позволяет:

- строить сеть с использованием нескольких сетевых технологий, не обременяя прикладного программиста знанием ее структуры: для него каждый сенсорный узел выглядит непосредственно доступным по IP-адресу;

- снять ограничения на число узлов в сети;
- использовать любую существующую сетевую инфраструктуру в качестве технологической сети.

Однако мы встречаем проблему при его использовании. Как известно, минимальное значение MTU для IPv6 составляет 1280 байт, а определяемый стандартом IEEE 802.15.4 максимальный размер пакета в LoWPAN-сетях равен всего лишь 127 байтам. Как и всегда в таких случаях, вводится дополнительный уровень адаптации. Существует проект IETF по разработке стандарта на передачу IPv6-пакетов через сеть 802.15.4. Сформирована рабочая группа [3] под названием “6lowpan” (сокращение от “IPv6 over LoWPAN”).

Выпускаются промежуточные версии стандарта [4]. Разработка ведется как альтернатива проприетарному протоколу ZigBee [5].

Функции, которые предлагает 6lowpan:

(1) Сжатие заголовка IP-пакета: при отправке необходимо выбросить все, что можно потом восстановить при получении.
(2) Фрагментация: не всегда можно сжать IP-пакет до размеров пакетов LoWPAN-сетей.

Например, могут быть удалены поля “Version” и “Traffic Class”. В определенных случаях даже IP-адреса могут быть исключены из заголовка пакета. Для этого они должны быть назначены такими, что выводятся из MAC-адресов [6]. В лучшем случае удается сжать 40 байт заголовка до 4-х.

Рабочей группой 6lowpan предлагается сжимать также и заголовки пакетов транспортного уровня (UDP и TCP), которые передаются в полезной нагрузке (payload) IP-пакетов, но выигрыш это дает совсем небольшой — до нескольких байт.

Необходимая для нашего проекта функциональность 6lowpan была реализована в виде процессов для ОС Contiki.

3.3. Расписания

Стандартом IEEE 802.15.4 предусмотрено несколько вариантов организации доступа к среде передачи. В проекте «Ботик-сенсор» используется простейший безмаячковый режим (beaconless mode) и специфические расписания сеансов обмена. Расписания отвечают как за прием пакетов, так и за их отправку.

Постоянные расписания устроены просто — они начинают свою активность сразу же после их добавления, включают приемник или

же начинают отправку пакетов, после истечения времени жизни они удаляются. Для организации приватного режима общения между узлами в сети используются особые криптографические расписания.

3.3.1. *Устройство криптографических расписаний*

При создании крипто-расписания получают на вход следующие аргументы:

- список доступных каналов, их количество N_c;
- длительность сеанса связи T_s;
- длительность периода T_p, на который строится расписание;
- секретное 128-битное слово W_s.

Задача программы — запланировать N_c сеансов связи в течение периода T_p — по одному сеансу на каждом канале. Всего в течение периода возможно $\frac{T_p}{T_s}$ потенциальных сеансов связи. К примеру, АУ-протокол, использующий криптографическое расписание, имеет следующие параметры: $N_c = 16, T_s = 488\mu s, T_p = N_c \times 1000000\mu s$.

Выбор сеансов из числа потенциальных должен происходить случайным, но детерминированным образом. Причем, детерминированность очень важна: если два устройства хотят успешно связываться друг с другом, они должны выбирать одинаковые сеансы из большого числа возможных.

Для случайного выбора сеансов предлагается использовать AES-шифратор, поддержанный аппаратно трансивером AT86RF231. Ему на вход подается секретное слово W_s и время начала периода (ближайшее меньшее время, кратное T_p). Шифратор конкатенирует байты аргументов, в качестве ключа использует это же секретное слово W_s. Байты, подаваемые шифратором на выход, используются для определения времени начала сеанса. Если в W_s поменяется хоть один бит, расписание получится совершенно другим.

В результате экспериментов было установлено, что набор байтов, выдаваемый шифратором, может считаться достаточно случайным для целей построения расписания, то есть вероятности выбора каждого сеанса примерно одинаковы. Свойство детерминированности при использовании шифратора, разумеется, выполняется.

Секретное слово W_s является частью конфигурационных данных устройства. УС отвечает за правильную конфигурацию узлов — таким образом, чтобы узлы, которым необходимо общаться между собой, выполняли расписания с одинаковым W_s.

Использование крипто-расписаний также повышает устойчивость сети к помехам по сравнению с маячковым режимом (beacon-enabled mode).

4. Связь сенсорной сети с Интернет

Исходя из своего устройства, сенсорная сеть обязательно должна быть обеспечена связью с управляющей станцией. Также, впоследствии, может оказаться удобным включение всей сенсорной сети в сеть Интернет. В качестве сетевого протокола нами был выбран IPv6, в сети же Интернет сейчас используется протокол IPv4, поэтому требуются средства, позволяющие передавать IPv6-пакеты через IPv4-сеть.

4.1. Шлюз

В структуре нашей сенсорной сети предусмотрен специальный узел-шлюз. Это узел, питающийся от сети и имеющий два интерфейса, обычно один из них — радио 802.15.4, а другой — проводной Ethernet. Радио-интерфейс используется шлюзом для связи с другими узлами сенсорной сети, Ethernet — для связи с управляющей станцией. Задача шлюза — перевод пакетов из одной среды в другую. Например, управляющая станция может послать конфигурационный пакет одному из своих подопечных устройств, либо самому узлу потребуется сообщить управляющей станции показания своих датчиков.

4.2. Туннель

Сеть между шлюзом и управляющей станцией работает по протоколу IPv4. Шлюз, однако, получает из радио-интерфейса 6lowpan-фрагменты, из которых собирается IPv6-пакет.

Один из способов обеспечения связи между устройствами и станцией — туннелирование IPv6-in-IPv4 [7]. Как следует из названия, IPv6-пакеты передаются внутри пакетов IPv4. Туннель прокладывается между шлюзом и управляющей станцией. На стороне управляющей станции (ОС Linux) необходимо лишь настроить соответствующий интерфейс, все необходимые программы уже разработаны. На стороне же шлюза (ОС Contiki), необходимо было создать процесс, обрабатывающий необходимым образом входящие и исходящие пакеты.

Логика работы процесса, обслуживающего туннель, такова:

(1) Пакет получен из туннеля (Ethernet):
   (a) IPv6-пакет извлекается из IPv4-пакета;
   (b) если пакет адресован самому шлюзу, то он сразу передается в IP-стек для дальнейшей обработки;
   (c) иначе, пакет передается в 6lowpan и пересылается подопечным устройствам (либо выбрасывается);

(2) Пакет получен из радио-среды:
   (a) если пакет адресован самому шлюзу, он передается в IP-стек для обработки;
   (b) в противном случае пакет помещается внутрь IPv4-пакета и отправляется по туннелю.

Узел на другой стороне туннеля (это не обязательно управляющая станция) сам принимает решения: он может отправить IPv4-пакет дальше по сети, сам обработать извлеченный IPv6-пакет или маршрутизировать IPv6-пакет, если есть такая возможность и необходимость.

5. Заключение

В настоящее время завершается разработка аппаратной части и lowpan-драйвера для нее. Пока же описанные программы исполняются под управлением среды моделирования Cooja. На рис. 3 изображена текущая схема работы. В правой части рисунка находится виртуальная сенсорная сеть, связанная с приложением с помощью туннеля.

Рис. 3.

Таким образом, получены следующие результаты:

- освоены операционная система Contiki и поставляющаяся с ней среда моделирования Cooja;
- сформирован сетевой стек узла сенсорной сети и написаны программы, реализующие часть его уровней: lowpan-драйвер для Cooja, расписания, АУ-протокол, 6lowpan;
- изучены возможности установления связи между УС и шлюзом сенсорной сети, проложен туннель между ними, написаны необходимые приложения для ОС Contiki;
- с помощью средства Cooja перечисленные программы были отлажены на модели сенсорной сети.

Также в данный момент ведутся работы по поддержке Mesh-сетей [8], которые требуют использования алгоритмов маршрутизации.

Список литературы

[1] IEEE 802.15.4, http://standards.ieee.org/getieee802/802.15.html. ↑1
[2] The Contiki OS, http://www.sics.se/contiki/. ↑2.1
[3] 6lowpan Working Group, http://tools.ietf.org/wg/6lowpan/. ↑3.2
[4] 6lowpan RFC 4944, http://tools.ietf.org/html/rfc4944. ↑3.2
[5] ZigBee Alliance, http://www.zigbee.org/. ↑3.2
[6] IPv6 Addressing Architecture, http://tools.ietf.org/html/rfc4291. ↑3.2
[7] IPv6-in-IPv4, http://en.wikipedia.org/wiki/6in4. ↑4.2
[8] Mesh-сети, http://en.wikipedia.org/wiki/Mesh_networking. ↑5

M. D. Nedev. *Protocols and algorithms in LoWPAN-networks* // Proceedings of Junior research and development conference of Ailamazyan Pereslavl university. — Pereslavl, 2009. — p. 147–156. (*in Russian*).

Abstract. Paper considers algorithms and protocols which are used in sensor network developed in PSI RAS. Author describes the way to use the simulator for debugging programs designed for sensor networks. He also presents tunnel for joining LoWPAN-network and Internet. Finally, the paper describes a media access control technique based on communication session schedules.