И.М. Никольский [1]

ОБ ОДНОМ ИНСТРУМЕНТЕ ОТЛАДКИ ПРОТОКОЛОВ МАРШРУТИЗАЦИИ В БЕСПРОВОДНЫХ СЕНСОРНЫХ СЕТЯХ

1. Введение

Беспроводная сенсорная сеть (БСС) представляет собой совокупность небольших устройств, называемых «умными сенсорами» (smart sensors или motes). Основными компонентами типичного умного сенсора являются датчик, микроконтроллер, модуль беспроводной связи и элемент питания (батарейка). Узлы БСС способны замерять одну или несколько физических величин (температуру, концентрацию газа, освещённость и т. д.) и обмениваться информацией по беспроводному каналу. В конечном счёте, как правило, данные из БСС аккумулируются в облачном хранилище. Соответствующую инфраструктуру для хранения данных предоставляют большинство провайдеров облачных услуг. Сенсорные сети могут быть как инструментом мониторинга, так и основой более сложных конструкций, используемых в туманных вычислениях (fog computing).

В простейших БСС имеется некоторое множество рабочих узлов (для краткости будем называть их *сенсорами*), занимающихся непосредственно сбором и передачей информации, а также узел — накопитель информации, так же называемый *стоком* (sink). В обязанности стока входит сбор информации с рядовых узлов для последующей передачи на базовую станцию [1].

Существуют и более сложные БСС с несколькими стоками. Такие БСС могут иметь кластерную структуру: каждый сток является центром кластера из нескольких рабочих узлов. В этом случае информация передаётся на базовую станцию по цепочке от стока к стоку. Отметим, что роль стока может переходить от одного узла к другому. Так, например, в протоколе PEGASIS [2] каждый узел становится стоком один раз каждые N раундов работы протокола (где N- число узлов сети), что позволяет более эффективно расходовать заряд элементов питания.

Разработка протоколов маршрутизации является одной из важных тем в области исследований БСС. На данный момент ни один из протоколов не является доминирующим. Это связано с многообразием

[1] МГУ имени М.В.Ломоносова, факультет ВМК, кафедра СКИ, nikolsky@cs.msu.ru.

сенсорных сетей, разрабатываемых под различные задачи и различные требования.

Новые подходы к маршрутизации данных в БСС появляются регулярно. К сожалению, многие идеи остаются в виде описаний алгоритмов или компьютерных симуляций, не доходя до испытаний на физических сенсорных сетях.

Переход к натурному тестированию новых протоколов на узлах реальных БСС (даже построенных на основе такой дружественной платформы как Arduino) сопряжено с целым рядом трудностей, которые для многих исследователей становятся непреодолимыми. Наибольшей среди этих проблем является отсутствие простого механизма работы с управляющим кодом БСС, который позволял бы тестировать его в симуляционной среде на ПК без необходимости создания отдельной симуляционной программы.

Отметим, что симуляция на ПК является важным этапом при разработке протокола. Как было выяснено автором при разработке протокола [3], отладка непосредственно на БСС занимает много времени, т. к. после каждой корректировки программы требуется загрузка кода на узлы сети. Более разумным подходом представляется использование на предварительном этапе отладки протокола на ПК.

Между тем существующие открытые симуляторы сетей - Omnet++[4], NS3[5] - достаточно сложны в освоении и, насколько известно автору, не предоставляют возможности автоматической трансляции симуляционной программы в управляющий код узлов БСС. Возможность автоматической генерации кода для Arduino предоставляет Matlab Simulink[6], однако этот продукт является платным.

В данной работе представлен программный инструмент (симулятор), который позволяет отлаживать логику протокола на ПК, а затем автоматически получать управляющий код узлов БСС в виде программ для платформы Arduino. Предлагаемый симулятор выполнен по принципу, развиваемому в симуляторе Omnet++. Он представляет собой фреймворк, с помощью которого пользователь может «проигрывать» симуляцию работы своего протокола в пошаговом режиме. Для этого требуется лишь предоставить основные компоненты протокола (логика работы узлов, описание структуры служебных сообщений и т.д.) в виде совокупности структур и функций на языке Си в соответствии с достаточно простым набором правил.

Отладочный вывод и логирование, а также сбор статистических данных позволяют отыскать ошибки в логике протокола. Предоставляется возможность параллельного выполнения набора симуляций на суперкомпьютере (используется технология MPI).

Отлаженный протокол может быть перенесён на узлы БСС без переписывания кода с нуля. В настоящее время в качестве целевой архитектуры узлов БСС используется платформа Arduino[7] с радиомодулем NRF24[8]. Функционал симулятора может быть легко расширен для поддержки других платформ и радиомодулей.

Основная часть данной работы имеет следующую структуру. В п.2 описывается БСС, построенная автором на основе Arduino и NRF24. В п.3 изложены основные принципы построения симулятора, которые позволяют отлаживать работу протокола маршрутизации данных БСС на ПК. Архитектура симулятора описана в п.4. Основные выводы представлены в заключении (п.5).

2. Целевая БСС

На сегодняшний день существует большой выбор специализированных аппаратных решений для сенсорных сетей. Наиболее известными являются Crossbow Mica2, Intel Mote, а также Telos разработки университета Беркли. К сожалению, эти сенсоры достаточно дороги и не всегда доступны в продаже. В связи с этим при построении тестовой БСС, предназначенной для исследования новых протоколов сбора информации, целесообразно выбирать более бюджетное оборудование. Особенно это важно, если предполагается создание сети с большим количеством узлов.

Построение доступного по цене сенсорного узла возможно при использовании в качестве основы платы Arduino. Эти платы представляют собой устройства, построенные на основе микроконтроллера ATmega, с достаточно большим количеством цифровых и аналоговых портов ввода/вывода. Предназначены они для быстрого прототипирования электронных устройств и ориентированы на непрофессиональных пользователей.

Кроме умеренной цены (особенно в этом отношении выделяется Arduino Nano) эти платы обладают и другими достоинствами. Устройства программируются на языке высокого уровня (диалекте языка Си), таким образом работа с Arduino возможна без наличия глубоких знаний в области программирования микроконтроллеров. Доступна удобная свободно распространяемая IDE с большим количеством библиотек. Отлаженная программа может быть загружена на устройство с ПК через USB-порт. Плата популярна среди радиолюбителей всего мира, поэтому в свободном доступе имеется большой массив информации по всем сторонам работы с ней.

К сожалению, Arduino не обладает возможностью беспроводной передачи данных, поэтому возникает необходимость в подключении внешнего радиомодуля. Достаточно часто в узлах БСС используются модули ZigBee [9], однако они весьма дороги. Кроме того, при работе с

ZigBee пользователь оказывается привязанным к определённому набору протоколов сетевого и прикладного уровней, что может быть нежелательным при разработке новых протоколов.

Бюджетной альтернативой ZigBee являются хорошо зарекомендовавшие себя трансиверы NRF24, разработанные норвежской компанией Nordic Semiconductor. Эти модули работают на нелицензируемой частоте 2,4 ГГц , передача информации производится в полудуплексном режиме. Используется GFSK модуляция. Пользователю предоставляется возможность работать на одном из 126 частотных каналов, также можно регулировать скорость передачи информации (250кбит, 1Мбит и 2Мбит), мощность передатчика и усиление приёмника.

Модули NRF24 обладают целым рядом достоинств. Они популярны среди радиолюбителей, их легко найти в продаже. С точки зрения энергоэффективности NRF24 выигрывает у модулей ZigBee ([10], [11]). Максимальный потребляемый ток 13.5мА, в режиме «power down» - 26мкА, в режиме «standby» - 900 нА. В то же время самый известный из ZigBee модулей - Xbee - в режиме активной передачи потребляет 45 мА, а в режиме активного приёма 50 мА. Стоимость NRF24 намного меньше, чем у Xbee.

Недостатком NRF24 является длина поля данных — всего 32 байта, что значительно меньше, чем у ZigBee (82 или 100 байт). Это может быть негативным фактором в случае. если узлы несут большое количество датчиков. Следует также иметь в виду, что на частоте 2,4 ГГц работают также WiFi и Bluetooth. Практика, однако показывает, что радиомодули NRF24 являются достаточно удачным выбором для обеспечения связи в БСС (см напр. [12]).

3. Реализация выполнения управляющего кода БСС на ПК

При реализации программы, позволяющей выполнять код узлов БСС на ПК, необходимо учитывать несколько факторов. Во-первых, это структура программы для целевой архитектуры БСС (Arduino). Традиционная программа(скетч) для Arduino устроена следующим образом. Файл скетча (как правило, он имеет расширение ino) включает в себя две обязательные функции - setup() и loop(). Функция setup() выполняется при запуске программы, функция loop() вызывается в бесконечном цикле. Отметим, что подобную структуру имеют программы не только для Arduino, но и для многих других плат (Freeduino, LilyPad, NodeMCU и т.д.).

Во-вторых, при симуляции работы БСС возникает необходимость вызывать управляющие коды всех узлов из одного цикла симуляции. Это можно реализовать, оформив итерации циклов работы узлов в виде функций sensor_step() и sink_step() для сенсоров и стоков соответственно.

В этом случае в цикле симуляции можно вызывать эти функции, а также функции-шаги симулятора.

Третьим важным моментом является необходимость реализации некоторых функций в двух вариантах - для симуляции на ПК и для выполнения на узле на БСС. К таким функциям относятся:

- Посылка и приём сообщений. На ПК эти действия эмулируются путём копирования сообщений в некоторый буфер (или из буфера в случае приёма сообщения); при работе на узле приём и отправка выполняется с помощью вызова соответствующих команд радиомодуля.
- Выдача различных информационных и отладочных сообщений. При эмуляции на ПК сообщения просто выводятся на экран. При работе на узле отладочные сообщения должны передаваться через последовательный порт на ПК либо выводиться на подключённый LCD экран; ещё одним вариантом может быть отключение сообщений.
- Получение идентификатора узла. Мы предполагаем, что каждый узел имеет идентификатор, анонимные сети в нашей работе не рассматриваются. При эмуляции на ПК идентификаторы узлам присваиваются симулятором. При работе на узле считаем, что идентификатор заранее загружен в EEPROM память Arduino.

4. Архитектура инструмента отладки протокола

4.1 Формат протокола. Для обеспечения возможности тестирования логики протокола в симуляционной среде на ПК необходимо выделить основные компоненты протокола, а именно:

1) структура данных узла (набор данных, хранимых на узле и необходимых для работы протокола);
2) структура служебного сообщения;
3) множество состояний стока;
4) множество состояний сенсора;
5) логику работы стока;
6) логику работы сенсора.

Платы Ардуино программируются на диалекте языка Си, в связи с чем удобно описывать перечисленные компоненты следующим образом: 1) и 2) в виде структур; 3) и 4) в виде перечислений (enum); 5) и 6) в виде функций sink_step() и sens_step() соответственно. Предполагается, что sink_step() и sens_step() вызываются на каждом шаге работы соответствующего узла.

Перечисленный набор сущностей размещается в нескольких модулях, хранимых в специальной папке протокола. Необходим также заголовочный файл, который обеспечит подключение протокола к инструменту отладки и управляющим программам узлов БСС.

4.2 Принцип работы отладчика. Предлагаемый отладчик представляет собой программу, написанную на языке Си, имитирующую работу беспроводной сенсорной сети с ненадёжным каналом связи. Для отладки своего протокола пользователю необходимо оформить его в соответствии с принципами, описанными в предыдущем подпункте.

Протокол выполняется в пошаговом режиме. На каждой итерации основного цикла программы поочередно вызываются функции работы каждого из узлов (sink_step() либо sens_step() в зависимости от типа узла). При этом узел может послать либо принять сообщение, а также сменить своё состояние.

Передача и приём сообщений эмулируются копированием сообщения в/из буфера симулятора. При этом имитируются потери сообщений, распределённые по геометрическому закону.

Кроме того, на каждой итерации основного цикла производится запись сообщений о ходе выполнения протокола в лог, а также обновляется статистика (например, количество посланных сообщений).

Отметим, что представленный в данной статье инструмент не предназначен для моделирования процессов передачи радиосигнала. Он предназначен для предварительной отладки логики протокола и оценки некоторых статистических показателей работы протокола.

При большом количестве узлов симуляция может занимать достаточно длительное время. В связи с этим пользователю предоставляется возможность параллельного запуска нескольких симуляций на узлах суперкомпьютера с распределённой памятью. Таким образом процесс сбора необходимой статистики по работе протокола может быть ускорен. Распределение задач по узлам суперкомпьютера, а также сбор на корневом узле общей статистики по параллельным сериям экспериментов осуществляется с помощью технологии MPI.

4.3 Генерация управляющей программы для Arduino. После отладки протокола на ПК пользователь может сгенерировать соответствующие программы управления для стоков и сенсоров БСС. Для этого потребуется минимум усилий, поскольку наш инструмент отладки включает шаблон сктечей (программ для Arduino) sensor.ino и sink.ino, имеющие следующий вид:

sink.ino

```
#include “proto_name.h”
void setup()
{
    //инициализация модуля NRF24
sink_setup();
}
void loop()
{
    sink_step();
}
```

sensor.ino

```
#include “proto_name.h”
void setup()
{
    //инициализация модуля NRF24
sens_setup();
}
void loop()
{
    sens_step();
}
```

Здесь sink_setup(), sink_step(), sensor_setup(), sensor_step() - функции, определённые в соответствующих модулях, составляющих описание протокола. Благодаря директивам условной компиляции действие функций отправки и приема сообщений переключается с использования механизмов симулятора на вызов соответствующего функционала радиомодуля NRF24.

Как видно из приведённых листингов для определения протокола необходимо лишь подключить необходимый заголовочный файл.

5. Заключение

В работе представлен программный инструмент, позволяющий в пошаговом режиме разыгрывать возможные сценарии работы БСС под управлением заданных программ узлов. Инструмент позволяет ускорить отладку протоколов сбора данных, может быть использован для оценки эффективности работы протокола. Предоставляется возможность параллельного выполнения нескольких серий экспериментов на узлах суперкомпьютера.

Использование инструмента не требует написания отдельной симуляционной программы. Протокол, оформленный в специальном формате, может быть легко преобразован в управляющий код целевой БСС с узлами, построенными на основе платы Arduino и радиомодуля NRF24. Архитектура описанной программы позволяет легко расширить спектр целевого оборудования.

Литература

1. *W. Dargie, C. Poellabaue* Fundamentals of Wireless Sensor Networks: Theory and Practice Wireless Communications and Mobile Computing John Wiley & Sons, 2010, 336 p.
2. *S. Lindsey, C. S. Raghavendra* PEGASIS: Power Efficient Gathering in Sensor Information Systems // Proceedings of the IEEE Aerospace Conference, Vol. 3, Big Sky, 2002, pp. 1125-1130.

3. *I.M. Nikol'skii, K.K. Furmanov* Efficiency of Retransmission in Sensor Networks // Computational Mathematics and Modeling, Consultants Bureau, № 31, pp 471-476
4. *A. Varga, R. Hornig* 2008. An overview of the OMNeT++ simulation environment. // Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops (Simutools '08). ICST, Brussels, BEL, Article 60, pp 1–10.
5. *G. F. Riley, T. R. Henderson* The ns-3 Network Simulator.. In K. Wehrle, M. Günes & J. Gross (ed.) // *Modeling and Tools for Network Simulation* . Springer, 2010, pp. 15-34
6. Using Simulink // The MathWorks, Inc 1990 – 2004 http://users.isr.ist.utl.pt/~alex/micd0506/simulink.pdf
7. Y. A. Badamas The working principle of an Arduino *// 11th International Conference on Electronics, Computer and Computation (ICECCO)*, 2014, pp. 1-4
8. nRF24L01+ Single Chip 2.4 GHz Transceiver Preliminary Product Specification v1.0 https://www.sparkfun.com/datasheets/Components/SMD/nRF24L01Pluss_Preliminary_Product_Specification_v1_0.pdf
9. *R. Faludi.* Building Wireless Sensor Networks with ZigBee, XBee, Arduino, and Processing. O'Reilly Media, 2010, 322 p.
10. *H. Saha, S. Mandal, S. Mitra, S. Banerjee, U. Saha.* Comparative Performance Analysis between nRF24L01+ and XBEE ZB Module Based Wireless Ad-hoc Networks// International Journal of Computer Network and Information Security, Vol.9, No.7, 2017, pp.36-44.
11. *N. Sizen.* A Comparative study of Wireless Star Networks Implemented with Current Wireless Protocols // Masters Theses, 2019, 920 p.
12. *A. Rahman, N. Athilah, A. Jambek.* Wireless sensor node design //3rd International Conference on Electronic Design (ICED), August 11-12, 2016, Phuket, Thailand, pp. 332-336.