

УДК 004.92

МОДЕЛИРОВАНИЕ И ВИЗУАЛИЗАЦИЯ БЕСПРОВОДНЫХ СЕНСОРНЫХ СЕТЕЙ*Левжинский А.С., Телятников А.О.**Донецкий национальный технический университет,
кафедра автоматизированных систем управления
alex.lev89@gmail.com*

Рассмотрена актуальность, структура и топология беспроводных сенсорных сетей. Определены средства для моделирование работы беспроводных сенсорных сетей. Определен метод визуализации графа на физических аналогиях. Описана разработанная программа для визуализации работы беспроводных сенсорных сетей.

Постановка задачи

Беспроводные сенсорные сети (WSN) – перспективная технология, имеет ряд преимуществ перед обычными вычислительными сетями. В частности это полное отсутствие каких бы то ни было кабелей, возможность компактного размещения и интеграции мотов в объекты окружающей среды и надежность всей системы в целом. WSN могут быть использованы для решения широкого спектра задач от контроля окружающей среды до использования в системах обороны и обеспечения безопасности.

В целях экономии средств на доработку и исправления уже внедренных беспроводных сенсорных сетей их прорабатываю с использованием эмулятора, для определения проблем, узких мест. Для значительного облегчения анализа будущей системы информацию эмуляции работы сети необходимо визуализировать.

Задачей дипломной работы является создание программы, которая будет моделировать работу беспроводной сенсорной с помощью эмулятора TOSSIM, отобразит граф сети и ее работу с соблюдением норм и критериев.

Структура беспроводных сенсорных сетей

Беспроводные сенсорные сети (wireless sensor networks) состоят из миниатюрных вычислительно-коммуникационных устройств – мотов (*от англ. motes – пылинки*), или сенсоров. Мот представляет собой плату размером обычно не более одного кубического дюйма. На плате размещаются процессор, память — флэш и оперативная, цифроаналоговые и аналого-цифровые преобразователи, радиочастотный приемопередатчик, источник питания и датчики, которые подключаются через цифровые и аналоговые коннекторы (чаще других используются датчики температуры, давления, влажности, освещенности, вибрации). Набор применяемых датчиков зависит от функций, выполняемых беспроводными сенсорными сетями. Питание мота осуществляется от небольшой батареи. Моты используются только для сбора, первичной обработки и передачи сенсорных данных.

Основная функциональная обработка данных, собираемых мотами, осуществляется на узле, или шлюзе, который представляет собой мощный компьютер.

Для получения данные узел оснащается антенной. Но доступными для узла оказываются только моты, находящиеся достаточно близко от него; другими словами, узел не получает информацию непосредственно от каждого мота. Проблема получения сенсорной информации, собираемой мотами, решается следующим образом. Моты могут обмениваться между собой информацией с помощью приемопередатчиков, работающих в радиодиапазоне. Они обмениваются сенсорной информацией, считываемой с датчиков и информацией о состоянии устройств и результатах процесса передачи данных. Информация передается от одних мотов другим по цепочке, и в итоге ближайшие к шлюзу

моты сбрасывают ему всю аккумулированную информацию. Если часть мотов выходит из строя, работа сенсорной сети после реконфигурации должна продолжаться. Но в этом случае, естественно, уменьшается число источников информации.

Для выполнения функций на каждый мот устанавливается специализированная операционная система. В настоящее время в используется ОС TinyOS, разработанная в Университете Беркли.

Протокол работы сетей

В беспроводных сенсорных сетях используется протокол ZigBee, который базируется на разработанном ранее стандарте IEEE 802.15.4, который описывает физический уровень и уровень доступа к среде для беспроводных сетей передачи данных на небольшие расстояния (до 75 м) с низким энергопотреблением, но с высокой степенью надежности

Некоторые характеристики радиопередачи данных для стандарта IEEE 802.15.4 приведены в таблице 1.

Таблица 1. Характеристики радиопередачи данных для стандарта IEEE 802.15.4

Полоса частот, МГц	Нужна ли лицензия	Регион	Скорость передачи данных, Кбит/с	Число каналов
868,3	Нет	Европа	20	1
902-928	Нет	Америка	40	1-10
2405-2480	Нет	Весь мир	250	11-26

Топология

В начале работы происходит идентификация всех мотов, а затем уже формируется схема маршрутизации. Все моты в стандарте ZigBee по уровню сложности разбиваются на три класса. Первый и высший из них – координатор – управляет работой сети, хранит данные о ее топологии и служит шлюзом для передачи данных, собираемых всей беспроводной сенсорной сетью, для дальнейшей обработки. В сенсорных сетях, как правило, используется один координатор. К следующему классу относятся моты маршрутизаторы, они принимают и передают данные, а также могут определять направление передачи. И наконец, самый простой мот может лишь передавать данные ближайшему маршрутизатору. Для оптимизации работы сети, на современном этапе развития протокола, отдельно моты-маршрутизаторы и простейшие моты в большинстве случаев объединены в один, который выполняет одновременно обе функции.

Моделирование

Для моделирования работы беспроводной сенсорной сети используется эмулятор TOSSIM. Он был специально разработан для подобных сетей и дает результаты приближенные к реальным. Также эмулятор позволяет выполнять тот же код, что и сенсорные узлы, которые используются на практике в сенсорных сетях.

TOSSIM использует очень простую, но невероятно мощную модель беспроводной сети. Сеть представляется в виде графа, в котором каждая вершина – беспроводной узел и каждой дуге между узлами поставлено в соответствие некоторое значение – вероятность ошибки. Каждый узел имеет локальную переменную, куда заносится то, что принимается им по радиоканалу.

Визуализации графа с помощью физических аналогий

Структура сети можно представить в виде ненаправленного графа, к тому же расположение мотов таково, что сеть имеет минимально количество пересечений. Его можно визуализировать с помощью алгоритма рисования графов на основе физических аналогий.

Главным преимуществом такого метода рисования графа то, что он дает в итоге хорошее

отображение графа. В большинстве случаев он строит симметричные изображения графов.

Метод заключается в том, что граф рассматривается как система тел с силами взаимодействия между ними. Вершины графа считают телами, а ребра пружинами. В таком случае алгоритм находит конфигурацию тел с локальной минимальной энергией – так называемую конфигурацию равновесия сил, в которой каждое тело занимает такую позицию, что сумма всех сил, приложенных к телу, равна нулю. Идея схематически изображена на рисунке 1.

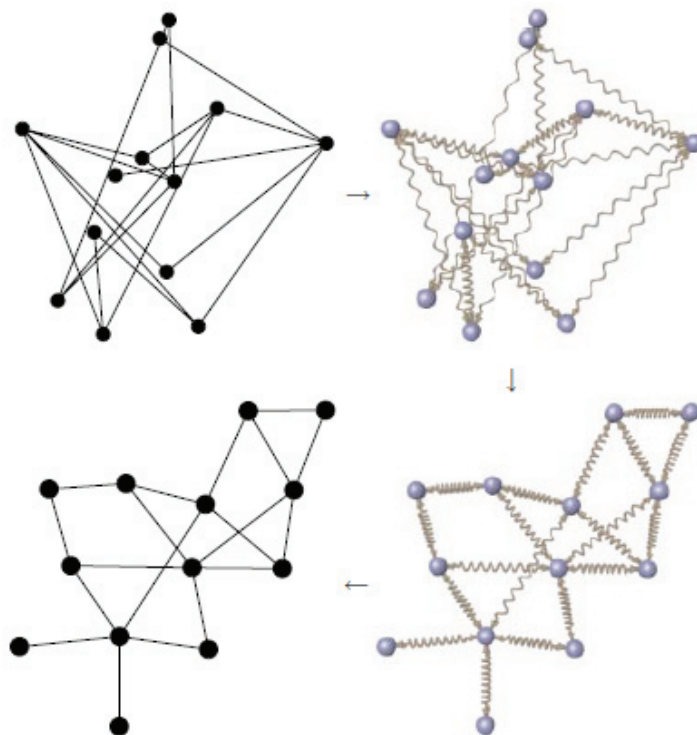


Рисунок 1. Идея метода физических аналогий

Для реализации метода граф представляют в виде комбинации пружин и электронных сил, когда каждое ребро рассматривается как пружина, а вершины считаются одинаково заряженными частицами, между которыми действуют силы отталкивания. Формула силы приложенной к вершине p

$$F(p) = \sum_{u=(p,q) \in E} f_u + \sum_{(p,q) \in V^2} g_{(p,q)},$$

где f_u – сила растяжения, действующая на вершину p из-за пружины (p,q) , а $g_{(p,q)}$ – это сила отталкивания, существующая между частицами p и q . По закону Гука f_u пропорциональна разности расстояний от p до q и длиной пружины с минимальной Энергией; а сила $g_{(p,q)}$ следует обратному квадратному закону.

Пусть $d(p,q)$ обозначает расстояние на плоскости между p и q . Тогда для x -й координаты $F(p)$ можно использовать формулу

$$\sum_{u=(p,q) \in E} k_u^{(1)} (d(p,q) - l_u) \frac{x_p - x_q}{d(p,q)} + \sum_{(p,q) \in V^2} \frac{k_{(p,q)}^{(2)}}{(d(p,q))^2} \frac{x_p - x_q}{d(p,q)},$$

где – параметры, которые не зависят от позиции вершин на плоскости и интерпретируются следующим образом:

- это естественна (с нулевой энергией) длина пружины между p и q . Если пружина имеет длину l_u , то не возникает сил растяжения между p и q .
- коэффициент жесткости (упругости) пружины между p и q . Чем больше он, тем сильнее пружина стремится установить расстояние между p и q , равным l_u .
- коэффициент силы отталкивания между p и q .

Алгоритм работы метода. На первом этапе вершины размещаются на плоскости случайным образом. Второй этап – это последовательность итераций для стабилизации, на каждой из которых вычисляются для всех вершин p силы $F(p)$ и для тех их них, для которых $F(p) \neq 0$, происходит перемещение вершины в направлении этой силы на расстояние пропорциональное модулю силы. Пропорция сдвига всех вершин и является еще одним параметром алгоритма.

Входные данные для визуализации работы сенсорной сети

Топология для работы эмулятора и программы загружается из текстового файла, каждая строчка которого обозначает связь между двумя узлами. Формат записи следующий:

```
1 2 -54.0
```

где 1 и 2 это номера узлов сети, а -54.0 – уровень затухания сигнала.

В ходе работы сети генерируются служебные сообщения о событии происходящее в работе сети. Например это информация о получении, отправке пакета данных или данные о неполадках на узле, также указывается время происхождления события.

Для генерирования служебных сообщений в коде эмулируемой программы, которая написана на языке `nesC` и установлена на узлах, вставляются специальные команды, генерирующие дебаг-сообщения. Они имеют следующий синтаксис:

```
dbg("Boot", "Boot at time: %s\n", sim_time_string());
```

где «Boot» ключевое имя канала, с помощью которого можно отфильтровать сообщения. «Boot at time: %s\n», `sim_time_string()` - само сообщение.

Так как общая структура тестируемых программ одинакова, то есть возможность автоматического анализа кода и вставка команд для вывода дебаг сообщений. Следовательно возможно автоматизировать обработку исходного кода. Это значительно упрощает подготовку кода к тестированию и делает разрабатываемую программу более универсальной и легкой во внедрении и освоении.

Служебные сообщения, генерируемые во время эмуляции работы сети являются входными данными для дальнейшей визуализации.

Данные, полученные на этапе эмуляции, приводятся в формат JSON. Далее, полученные данные преобразуются в формат JSON. Для этих целей используется скрипт написанные на языке Ruby.

Пример входного файла в формате JSON с данными для отображения событий:

```
{ "events": [ { "action": "senPackage", "source": 1, "target": 0, "type": 0, "time": 12:12:32.100 },  
  { "action": "senPackage", "source": 1, "target": 0, "type": 0, "time": 12:12:32.100 },  
  { "action": "senPackage", "source": 1, "target": 0, "type": 0, "time": 12:12:32.100 } ] }
```

Визуализация

Разработанная программа на JavaScript реализующая отрисовку графа и визуализацию работы сети. Пример работы программы показан на рисунке 2.

Заключение

Перспективность развития беспроводных сенсорных сетей очевидна. Уже сейчас во многих отраслях начинают использовать WSN. Это и мониторинг экологии, авто трафика, мониторинг погоды. С совершенствованием технологий и усложнение различных производств потребность в беспроводных сенсорных сетях будет только расти. Поэтому разработка программы для визуализации, которая бы помогла в тестировании и разработке новых беспроводных сенсорных сетей является важной и актуальной.

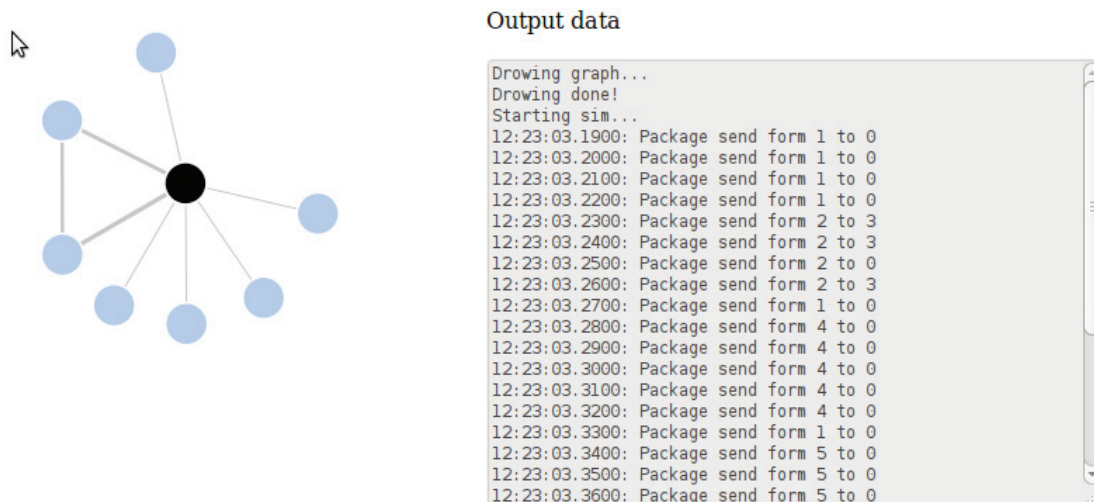


Рисунок 2. Результат работы программы

Литература

- [1] Michael Kaufmann, Dorothea Wagner. Drawing Graphs. Methods and Models. Springer-Verlag Berlin Heidelberg, 2001 – 325 с.
- [2] Касьянов В.Н., Евстигнеев В.А. Графы в программировании: обработка визуализация, применение. С-П. 2003 – 1104 с.
- [3] Philip Levis. TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications. 2009 – 23 с.
- [4] Maneesh Varshney. Detailed Models for Sensor Network Simulations and their Impact on Network Performance. 2006
- [5] Ioannis G. Tollis, Giuseppe Di Battista. Graph Drawing: Algorithms for the Visualization of Graphs. Printise-Hall 2003
- [6] Jesus Garcia. Ext JS in Action. Manning Publication 2010 – 495 с.