

МОДЕЛИРОВАНИЕ СИНХРОНИЗИРОВАННЫХ СЕНСОРНЫХ СЕТЕЙ

Д.В. Рагозин

ЗАО «Интел А/О», 603950, Россия, Нижний Новгород, ул. Тургенева, 30.
Тел.: +7(813)416 2444; факс +7(813)416 2445.
E-mail: Dmitry.Ragozin@intel.com

Описана реализация модели синхронизированной сенсорной сети, предназначенная для разработки, оценки и тестирования разработанного автором метода синхронизированной работы узлов сенсорной сети. Рассмотрены ограничения и возможности разных подходов к моделированию сенсорной сети, описаны особенности реализации модели. Оценена промышленная реализация сенсорной сети, созданная на основе предложенной модели.

The paper describes an implementation of a synchronous sensor network model. The model is intended for development, evaluation and testing of the method of synchronous channel access by sensor network nodes, which was developed by the author. Limitation and benefits of different sensor network modeling methods and model implementation details are described. An industrial implementation of a sensor network, which is based on the described model, is evaluated.

Введение

Сенсорные сети являются одним из современных направлений развития самоорганизующихся отказоустойчивых систем наблюдения и управления процессами, в том числе распределёнными. Сегодня имеется немалое количество отраслей и сегментов рынка (производство, различные виды транспорта, обеспечение жизнедеятельности, охрана), готовых для внедрения сенсорных сетей, и это количество непрерывно увеличивается. Тенденция обусловлена усложнением технологических процессов, развитием производства, расширяющимися потребностями частных лиц в сегментах безопасности, контроля ресурсов и использования товаро-материальных ценностей. С развитием полупроводниковых технологий появляются новые практические задачи и теоретические проблемы, связанные с применениями сенсорных сетей в промышленности и жилищно-коммунальном комплексе. С другой стороны, рынок сенсорных сетей достаточно сильно сегментирован, и для автоматизации различных процессов должны применяться различные технологические решения.

Имеющиеся реализации сенсорных сетей [1] имеют ряд недостатков, в частности, сети в основном используют асинхронный доступ в радиоканал, что приводит к значительному расходу энергии и малому времени автономной работы узлов сети. Наши исследования посвящены локально синхронизированным сенсорным сетям, в которых доступ узлов сети к радиоканалу синхронизирован во времени. Синхронный доступ к радиоканалу позволяет сократить потребление энергии, продлевая срок автономной работы устройств.

Целью данной работы является описание и обоснование применяемого подхода высокоуровневого имитационного моделирования для синхронизированных сенсорных сетей. Для создания эффективной модели сети огрубляется модель физического канала передачи данных, основная нагрузка переносится на модель канального уровня, а приложения верхнего уровня формализуются. Такая организация модели позволяет: упростить и/или регулировать сложность физического уровня, например, простейшая модель, вероятностная модель или модель трассировки лучей; сосредоточиться на отладке параметризованной модели канального уровня; надстраивать сетевой уровень и уровень приложений в зависимости от степени детализации модели сети.

Предметная область

Сенсорная сеть состоит из программного и аппаратного компонента. Аппаратный компонент представляет собой множество оснащённых приёмопередатчиками микропроцессорных устройств, с помощью которых возможна передача информации между данными устройствами в сети. Программный компонент агрегирует канальный, сетевой уровень и уровень приложений, позволяющие поддерживать передачу данных от всех узлов сети к некоторым узлам-«стокам» (базовым станциям), которые соединены с промышленными SCADA-системами. Сенсорная сеть – самоорганизующая структура: в случае выхода узлов из строя или добавления новых узлов в сеть, поток данных от других узлов к базовой станции перераспределяется с учётом изменения конфигурации сети. На рис. 1 показана сенсорная сеть и возможные изменения её конфигурации.

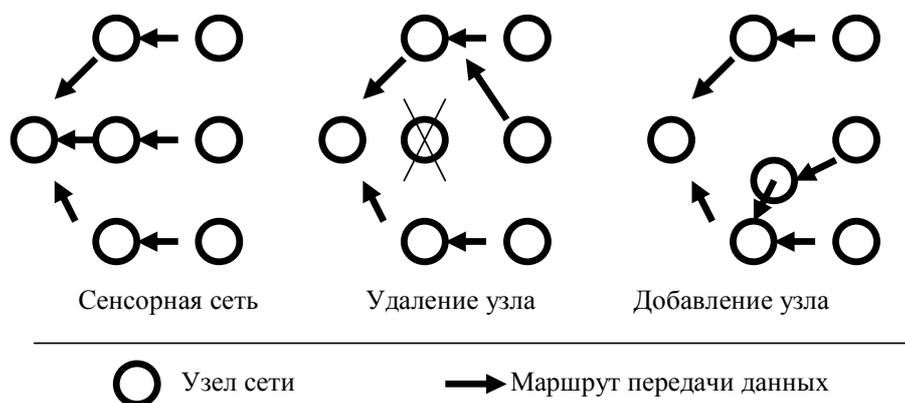


Рис. 1. Возможные варианты изменения конфигурации сенсорной сети

Отметим, что, несмотря на длительную историю сенсорных сетей [1] (как минимум 8 лет), нельзя отметить, что концепция построения сенсорной сети в настоящее время оформилась в программно-аппаратные (платформенные) решения, широко применяемые на промышленном рынке. Рынок сенсорных сетей является сильно сегментированным (транспорт, складской учёт, добывающая промышленность, военная техника и т.д.) с абсолютно разными требованиями к реализации узлов и программного обеспечения. С учётом конкретных требований, предъявляемых в рамках каждой индустриальной задачи, гипотетические реализации сенсорных сетей будет объединять самая общая концепция, так как могут изменяться базовые требования к реализации узлов и функционированию сети. Например, базовые требования к источнику питания могут существенно различаться: полностью батарейное питание; аккумуляторное питание с возможностью подзарядки от солнечных батарей или термальных элементов; питание от сети постоянного/переменного тока. В зависимости от типа источника питания оптимизируются алгоритмы работы сети, корректируются реализации канального и сетевого уровней.

Внедрение любой сенсорной сети состоит как минимум из трёх этапов: проектирование и моделирование структуры сенсорной сети, предназначенной для мониторинга и управления некоторым распределённым объектом; построение управляющих программ для узлов сенсорной сети; тестирование сенсорной сети в полевых условиях. Существующее программное обеспечение сенсорных сетей, как правило, основано на операционной системе TinyOS [2], разработанной в лабораториях Intel Research. TinyOS – не единственная система, на базе которой можно разрабатывать сенсорные сети. Специализированные конференции (например, ACM SenSys) включают десятки постеров и демонстраций как систем разработки, так и работающих прототипов сенсорных сетей. Однако, TinyOS – единственная система, позволяющая значительно снизить стоимость разработки сети, работающая с наиболее популярными процессорами (Texas Instruments MSP430, Atmel AVR, Intel XScale), так же и с популярными радиопередатчиками. При приобретении готовых узлов сенсорную сеть можно развернуть в течение нескольких дней. Экономический аспект – важнейшая разработка сенсорной сети «с нуля» крайне дорога.

TinyOS предоставляет программную модель, базовую операционную систему для узлов сети и быстрый цикл разработки как сенсорной сети, так и управляющей программы к ней: программирование управляющей программы на языке nesC [3] – проблемно-ориентированной надстройки над языком Си; отладка сенсорной сети с помощью системы моделирования TOSSIM [4]; программирование (т.е. создание и «заливка» образа управляющей программы в ПЗУ узла сети) узлов сенсорной сети и тестирование сети в полевых условиях. Так как внешняя среда, отображаемая в системе моделирования и реальная среда, в рамках которой функционирует сенсорная сеть, отличаются, возможны разнообразные проблемы функционирования сети на этапе окончательного тестирования. Точность представления внешней среды, как минимум, зависит от первоначальной модели, в рамках которой и разрабатывалась управляющая программа сети (возможно, разработчики имеют слабое представление о реальных условиях работы сети). Отметим, что фактически TinyOS является единственным широко используемым программным средством, позволяющим в настоящее время проводить разработку программного обеспечения для сенсорных сетей.

В случае моделирования синхронизированных сенсорных сетей, соответствующие инструменты практически отсутствуют. Возможны два варианта: расширение существующей среды TOSSIM [4] или любой другой с целью оптимизации и добавления ряда функций канального уровня; разработка собственной среды моделирования. Оба варианта имеют свои преимущества и недостатки, но по стоимости и по простоте сопровождения выигрывает второй вариант (на момент начала разработки модели модуль TOSSIM отсутствовал в имеющейся базе кодов TinyOS 2.0). Основной вопрос в данном случае сводится к выбору уровня точности моделирования компонентов синхронизированной сенсорной сети.

Уровни протоколов сенсорной сети

Одной из особенностей моделирования сенсорных сетей является введение модели физического уровня в модель сети. На рис. 2 показана модель уровней представления сети, поддерживаемых сенсорной сетью. Выделяем уровни модели, применяемые при описании стека протоколов TCP/IP и отличающиеся от формальной модели OSI в сторону упрощения взаимодействия компонентов стека протоколов (рис. 2) и уменьшения количества уровней модели. Рассмотрим стек протоколов TinyOS в рамках данной модели. На физическом уровне имеем упрощённую модель канала – как правило, рассчитывается только вероятность безошибочной доставки пакета между двумя узлами. Существуют реализации моделей распространения радиоволн для физических каналов беспроводных сетей (например, российские разработки [5, 6]), но они не входят в базовый пакет TinyOS/TOSSIM. На канальном уровне используется упрощённая модель доступа в канал: как правило, CSMA/CA (доступ в канал с предварительной проверкой активности канала). Канальный уровень модели, как правило, тесно связан с реализацией физического уровня. С точки зрения более высокого уровня (сетевой), канальный уровень предоставляет услуги посылки-приёма сетевых пакетов и, возможно, дополнительные услуги, связанные с контролем доставки пакета соседнему узлу. Физический уровень позволяет только передать бит данных в эфир и проверить, ведётся ли в канале передача другим узлом. Наиболее простые реализации физического уровня проверяют наличие коллизий в канале и определяют вероятность правильной передачи пакета от узла к узлу. Более сложные реализации моделируют передачу отдельных битов в канале, учитывая мощность сигнала от узлов, находящихся в зоне радиовидимости, однако такие не представлены в пакете TinyOS/TOSSIM.



Рис. 2. Стек сетевых протоколов сенсорной сети

Сетевой уровень отвечает за обработку и маршрутизацию пакетов в сенсорной сети, он использует услуги канального уровня и предоставляет абстрактный (скрывающий детали реализации) интерфейс пересылки пакетов для различных приложений пользовательского уровня. Сетевой уровень может предоставлять поддержку уровня качества услуг (например, приоритетность передачи пакетов или оптимизация передачи пакетов) сети. Сложность сетевого уровня часто зависит от уровня качества услуг, предоставляемых сетью и от задач, исполняемых маршрутизатором. В случае классической маршрутизации (передача пакетов только от любого узла к базовой станции или от базовой станции ко всем узлам), требования к маршрутизатору минимальны, а качество услуг часто определяется наличием свободной памяти для временного хранения обрабатываемых маршрутизатором пакетов. Уровень приложений, как правило, не отличается от программной реализации тех же приложений на реальных узлах сети – это связано с тем, что наибольший интерес вызывают (вызывали) модели распределённых приложений, исполняющихся на узлах сети.

Из такой структуры модели (в частности, применяемой в TinyOS) можно сделать некоторые выводы о применимости подобных моделей. Во-первых, все уровни модели разграничены чёткими интерфейсами и каждый уровень, как правило, ничего не знает о структуре соседних уровней, что позволяет изменять уровни модели настолько гибко, насколько это позволяет структура системы моделирования. Во-вторых, особое внимание уделяется возможности прямого переноса приложений из модели в реальную систему – в частности, это и была основная идея разработки языка NesC, с помощью которого написана TinyOS. С другой стороны, при таком подходе («сверху-вниз») возникает перекосяк в разработке модели сети (фактически модель сети скорее представлена приложениями верхнего уровня, а не поддерживающей инфраструктурой), связанный с уверенностью в том, что переносимость приложений между моделью и реальной системой является основной целью разработчика сети. К сожалению, в реальных сетях проблемы возникают на сетевом и канальном уровне, что крайне негативно влияет на устойчивость работы «отлаженного» уровня приложений.

TinyOS является не единственной разработкой: достаточное количество интегрированных решений, включающих системы моделирования, стеки протоколов, разрабатывается университетами и коммерческими предприятиями. Как пример приведем академическую разработку GloMoSim (<http://pcl.cs.ucla.edu/projects/gloimosim>), являющейся сетевым симулятором, моделирующим стек протоколов TCP/IP, несколько разных канальных и физических уровней, даже моделирование прохождения радиоволн. Ещё более впечатляющий набор моделей предоставляет коммерческий продукт QualNet Developer (<http://www.qualnet.com>), реализующий

длинный перечень протоколов уровня приложения (включая VoIP), транспортные протоколы TCP и ECN, два десятка протоколов маршрутизации, несколько протоколов доступа в канал и, как минимум, 9 методов расчёта прохождения радиоволн. Менее насыщен функциональной известный симулятор сетей с открытым кодом NS2 (http://nslam.isi.edu/nslam/index.php/Main_Page), что компенсируется возможностью реализовать практически любую модель, дописав симулятор. Из иных симуляторов можно отметить OMNet++ (<http://www.omnetpp.org>) и расширение моделирования сенсорных сетей SensorSimulator к нему (http://csc.lsu.edu/sensor_web), Ptolemy II (<http://www.ptolemy.berkeley.edu/ptolemyII/>). Отдельно стоит отметить COOJA – визуальный симулятор сенсорной сети на основе операционной системы реального времени Contiki – как альтернативное TinyOS решение. Имеются и исследовательские разработки, касающиеся функционирования агентных систем в сенсорных сетях, например, SensorWare (sensorware.sourceforge.net) и Agilla (mobilab.wustl.edu/projects/agilla), однако пока преждевременно говорить о реальном их использовании.

Неопределённость рынка, отсутствие выбора платформ для разработки, технические проблемы, связанные с неуверенностью в дальнейшей поддержке TinyOS разработчиками (да и всех других не коммерческих систем), вынуждают при построении специализированных (адаптированных под некоторую предметную область) сенсорных сетей строить оригинальные модели, в рамках которых основное внимание уделяется каналному и сетевому уровням. Далее будет описана модель, разработанная в рамках проекта WSNP Нижегородских лабораторий Intel Corporation.

Моделирование стека протоколов в синхронизированных сенсорных сетях

Асинхронные сенсорные сети имеют крайне важный недостаток для большого количества применения сетей: очень большие затраты энергии в режиме постоянного прослушивания сети. В случае введения схемы TDMA [7] (временное разделение доступа в канал) происходит значительный выигрыш (до 10 раз) в энергопотреблении, так как устройства «знают», в какой момент им необходимо прослушивать сеть. Введение синхронизации (схемы доступа TDMA в каналный уровень стека протоколов) в сенсорной сети приводит к смещению акцентов при разработке системы моделирования. Если в асинхронной сети реализация канального и физического уровней является простейшей, то в случае поддержки доступа в канал с разделением времени модель канального уровня становится нетривиальной и значительно превышает сложность уровня приложений. Решения, основанные на схеме доступа CSMA, тоже развиваются в сторону уменьшения энергопотребления при доступе в канал: в одной из схем доступа при уменьшении времени прослушивания канала увеличивается длина передачи преамбулы, которая должна быть не менее гарантированного интервала, в течение которого принимающий узел будет прослушивать сеть. Подобные схемы приемлемы в случае обслуживания конечных узлов, имеющих не более одного видимого узла-маршрутизатора, так как канал в ином случае будет использоваться неэффективно. Отметим, что проблема энергопотребления для конечных узлов, не маршрутизирующих большой поток данных, сильно преувеличена. При использовании простейших TDMA-схем энергопотребление конечных узлов будет значительно снижено (что подтверждается собственными экспериментальными данными), это позволяет в разы уменьшить ёмкость (и стоимость) батарей по сравнению с узлами-маршрутизаторами. В случае узлов-маршрутизаторов, снижение энергопотребления связано с введением более эффективной дисциплины доступа в канал, так как фактором, повышающим энергопотребление, является не потеря энергии при прослушивании канала, а коллизии пакетов и интерференция сигналов от разных узлов. Разрешение коллизий при большой нагрузке канала в случае CSMA неэффективно: в проводных сетях Ethernet, использующих тот же принцип доступа, при превышении числом активных узлов в сегменте сети некоторой величины (30 узлов), сеть перестаёт функционировать. То же наблюдается и в беспроводных сетях – при достаточно большой плотности узлов при превышении потоком данных некоторого порога сеть перестаёт функционировать.

Гарантированное функционирование сети и надёжная передача данных особенно важны в случае «густой» сети – большого количества узлов – вблизи базовых станций («стоков» пакетов сенсорной сети). При асинхронном доступе в канал количество коллизий пакетов стремительно увеличивается и расходуется энергия на повторную передачу пакетов. Синхронизированный протокол предоставляет гарантированные участки времени для передачи пакетов и позволяет значительно повысить отношение потраченной энергии к количеству переданной информации для ближайших к базовой станции узлов.

Перечислим основные отличия синхронизированного доступа в канал от простой асинхронной схемы. В асинхронной схеме для передающего узла не требуется «регистрация» в сети и получение разрешения на занятие канала от других узлов. Здесь и проявляется главный недостаток асинхронных сетей – при высокой относительной занятости канала узлы не могут гарантированно передать пакет по назначению в течении определённого времени, увеличивается количество коллизий пакетов и, учитывая небольшой объём оперативной памяти узла, теряются хранимые маршрутизатором пересылаемые пакеты. В итоге количество переданных пакетов становится значительно меньше количества принятых пакетов, пакеты теряются.

В случае реализации синхронизированной схемы каждый узел имеет гарантированный слот времени (или несколько слотов), когда он может, не конфликтуя с другими узлами сети, передавать пакеты данных

адресатам. Дополнительные накладные расходы на поддержку доступа с разделённым временем включают: формирование кадра (фрейма) сети, в рамках, которого осуществляется передача пакетов и регистрация узлов; процедуру регистрации узла в сети и выделения временного слота вошедшему в сеть узлу; процедуру уведомления узлов в сети о появлении нового узла; процедуру выхода узла из сети; поддержку синхронизации узлов в сети. Так как тактовые генераторы (системные часы) всех узлов имеют погрешности, узлам необходимо синхронизироваться в рамках каждого кадра и прогнозировать уход частоты часов (например, используя таблицы коррекции частоты генератора в зависимости от температуры окружающей среды). Даже при сравнительно хороших параметрах точности хода часов – $50 \cdot 10^{-6}$, при частоте синхронизации 32768 Гц рассинхронизация достигает 2 тактов системных часов в сек. Такая рассинхронизация требует обязательного введения процедур корректировки хода часов на узлах.

При разработке моделирующего инструмента решено более сосредоточиться на разработке модели канального уровня, чем на модели физического уровня. Достаточным условием для отработки алгоритмов канального уровня является регулируемая частота ошибок в канале, что достигается при простейшей модели канала. Тем более, на начальном этапе разработки модели неясна необходимая степень детализации, возможно и простейшая модель будет достаточна для проверки функционирования прототипа. В нашем случае модель физического уровня значительно упрощена, но в процессе моделирования возможны следующие реализации (перечисление идёт от простых к сложным): явное указание условий радиовидимости при отсутствии ошибок в канале; указание условий радиовидимости с ошибками в канале; вычисление условий радиовидимости непосредственно при передаче пакета; сложение радиосигналов от нескольких источников на принимающем устройстве; трассировка лучей. Выбор был сделан в пользу простейших моделей, так как в синхронизированном канале коллизии возможны лишь в строго оговоренных (в протоколе) случаях. Упрощение модели может приводить к огрублению физической модели, например, погрешность хода часов может приводить к нарушению синхронизации при чтении битового потока и начало пакета может быть принято не полностью. В упрощённой модели пакет будет однозначно отбрасываться, однако, в реальном мире в зависимости от условий синхронизации приёмника пакет с некоторой вероятностью может быть принят, например, в случае незначительной расстройки частот приёмника и передатчика. Но точное предсказание работы приёмника невозможно в принципе, а приближительная модель представляет собой худший (по сравнению с реальностью) случай, что позволяет нам использовать простую модель без оговорок.

Уровни маршрутизатора и приложений могут быть сильно упрощены для ряда применений, и это в ряде случаев не сказывается на точности представления модели. Изначально TinyOS разрабатывалась с учётом высокой вероятности изменения конфигурации сенсорной сети (введения новых узлов, удаления узлов, перемещения узлов). В реальных сетях, обслуживающих технологические процессы, изменения конфигурации сети происходят крайне редко: из-за отказа в источнике питания, из-за помех от приборов, использующих тот же диапазон и т.д. Поэтому, в самом простом варианте маршрутизатор может работать с неизменяемой таблицей маршрутов. В реальных сетях такие методики задания маршрутов используются в рамках «географической» маршрутизации, когда задаются предпочтительные маршруты передачи пакетов [8]. Далее, TinyOS была «испытательным полигоном» для большого числа разнообразных концепций распределённых приложений для сенсорных сетей, часто имеющих избыточную функциональность и сложность, причём необязательно, что большинство задач могут быть адаптированы для использования существующих сервисов TinyOS. Большинство существующих на сегодняшний момент задач наблюдения и управления не требуют столь сложной логики (например, достаточно лишь отсылать с определённым интервалом значения измеренных параметров), но могут иметь специфические требования, прямо не поддерживаемые существующими сервисами TinyOS (например, запись измеренных параметров в память). В основном, имеющиеся промышленные задачи позволяют формализовать и построить сравнительно простую модель генерации данных, которые необходимо передать через сенсорную сеть на базовую станцию.

Таким образом, в разрабатываемой модели для моделирования протокола синхронизированного доступа в сеть достаточно детально проработать канальный уровень модели и построить упрощённые уровни – физический, сетевой и уровень приложений. В будущем, по мере необходимости более детальных описаний данных трёх уровней, модель совершенствуется без изменения интерфейсов между уровнями стека протоколов.

Реализация системы имитационного моделирования

Существует немало количество систем имитационного моделирования как общего назначения, так и специально предназначенных для моделирования сетевых протоколов [9]. Ряд таких систем, предназначенных для моделирования сенсорных сетей, был упомянут ранее. Как правило, используется два диаметрально противоположных подхода: пользовательская реализация элементов системы моделирования на языке программирования общего назначения (TOSSIM) и использование полноценной системы моделирования общего назначения (например, AnyLogic). Отметим, что TOSSIM фактически подменяет физический уровень и некоторые сетевые интерфейсы, создавая «виртуальные» узлы сенсорной сети. Собственные наблюдения

свидетельствуют, что подменить физический уровень можно в практически любом стеке протоколов, в частности в собственных реализациях протокола синхронизированного доступа. При неизменности двух верхних уровней (сетевого и приложений) возможность подмены уровней позволяет практически без изменений переносить верхние уровни из модели в сеть. Но при описании такой модели разработчик должен заботиться о множестве деталей реализации, сопровождающих разработку приложения – фактически, модель создаётся в терминах операционной системы (TinyOS) и зависит от аппаратной конфигурации узлов, хотя для частной реализации сети «под задачу» данная проблема не столь существенна. В случае разработки «долгосрочной» модели, такой подход может быть неприемлем – в зависимости от среды, в которую погружена модель. В частности, при переходе от TinyOS 1.x к TinyOS 2.x многие интерфейсы TinyOS изменены, что привело к непереносимости ряда приложений. Более того, концепция TinyOS достаточно сложна для использования на последних фазах реализации системы, например, фазах программирования и отладки, и, как следствие, объём труда, затраченный на разработку в рамках TinyOS, слишком велик по сравнению с конкурирующими микро-операционными системами реального времени. Изучение nesC требует от программиста дополнительной квалификации. Ещё одним отрицательным качеством являются слабые возможности использования системы в режиме реального времени в рамках используемой программистской модели. В итоге, несмотря на формально развитую концепцию TinyOS, имеющийся хороший набор средств разработки и моделирования, находится достаточное количество неформальных причин, связанных с квалификацией персонала, управлением проектами и реальными требованиями, приводящих к тому, что использование TinyOS является достаточно рискованным.

Подход в рамках системы имитационного моделирования общего назначения приводит к практически полной непереносимости кода модели на аппаратную платформу (из-за оторванности высокоуровневых графических средств моделирования от реальных платформ). Например, внутренний язык программирования AnyLogic [10] – Java – за редкими исключениями не используется при программировании встраиваемых систем, что автоматически приводит к невозможности создания переносимого кода. Нишевые исключения для Java, например, мобильные телефоны, не являются определяющими в выборе языка. С другой стороны, это не мешает строить и проверять модели приложений. Получить в результате моделирования полностью переносимый на аппаратную платформу (узел сети) код крайне сложно, за редкими исключениями, так как это требует очень точного соответствия модельной среды реально существующей аппаратной реализации. Сложность здесь в основном обусловлена большой трудоёмкостью разработки и возрастанием стоимости модели, что не всегда приемлемо.

Для моделирования синхронной сенсорной сети создана модель, имеющая название SWSNS и позволяющая моделировать работу узлов синхронизированной сети с помощью разработанного фирменного протокола, являющегося предметом отдельной статьи. При разработке модели SWSNS (Synchronous Wireless Network System Simulator) выбран первый способ – основные компоненты системы имитационного моделирования (планировщик, очередь событий, поддержка соответствующих концептуальных сущностей) реализованы в виде отдельного модуля на языке Си. Система моделирования оперирует с событиями, функциями, вызываемыми при наступлении определённого времени на некотором узле сети. Основными концептуальными сущностями, с которыми работает система моделирования, являются узлы, пакеты и события. Основной код, реализующий алгоритм функционирования уровней стека протоколов, написан на языке Си, с тем, чтобы основная логика функционирования протокола могла быть перенесена на аппаратную платформу. Таким образом, схема реализации системы моделирования имеет черты системы TOSSIM.

Особенности реализации модели синхронизированного доступа в канал

Основной задачей, которую необходимо решить в рамках моделирования синхронизированного протокола канального уровня, является установление таких временных параметров протокола, чтобы при заданном минимальной точности хода системных часов узлы могли синхронизироваться в рамках одного кадра.

Отметим, что попытки реализации схемы доступа TDMA в рамках TinyOS предпринимались [11], но опубликованные результаты исследований не позволяют составить полное впечатление о масштабируемости имеющихся реализаций, несмотря на значительное повышение эффективности доступа в канал в сетях малого размера.

Идеология построения схемы синхронизации узлов в сети взята из [12], со значительными изменениями в технической реализации. Несмотря на то, что представленная в [12] концепция построения идеализирована, схема построения взаимодействия на основе выделения в начале кадра короткого промежутка времени для определения активности каждого узла и резервирования остатка кадра под передачу пакетов, является достаточно эффективной в случае даже значительных отклонений частот системных часов на узлах сети от номинала. Рассмотрим более детально формирование схему кадра в сети (схематично временная диаграмма кадра показана на рис. 3).

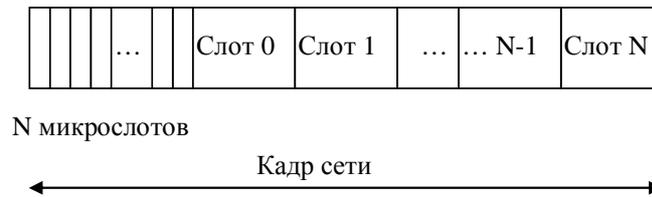


Рис. 3. Временная диаграмма кадра в сети

Кадр делится на две неравные части: короткая последовательность «микрослотов» и длинное по времени резервирование «слотов» для передачи пакетов. Микрослот является коротким промежутком времени, в рамках которого передаётся информация об активности узла (передаче узлом пакетов) в текущем кадре. Информация об активности передаётся в виде очень короткого пакета (1–4 байта). Сам временной промежуток гораздо шире, имеются достаточно большие зазоры между микрослотами, они используются для предотвращения коллизий между пакетами информации об активности из-за погрешности хода часов. Кроме того, информация о времени прихода пакета используется для корректировки хода собственных часов. Принятой в микрослоте информации достаточно для того, чтобы определить, какие узлы будут обмениваться информацией в эфире в течении следующего кадра. После последовательности микрослотов идёт «тело кадра», в котором строго зарезервированы слоты для передачи информации от каждого узла. В зависимости от реализации протокола, за узлом может быть закреплен более чем один слот. Между слотами и кадрами могут быть выделены временные промежутки – промежутки радиомолчания, используемые для обеспечения работы иных устройств, синхронизированных с сетью, в той же полосе частот, что востребовано в случае, если диапазон необходимо занять для передачи информации, не связанной с функционированием синхронизированного протокола канального уровня. Размер слота и кадра варьируется в зависимости от потребностей конкретной задачи, в нашей реализации он равен примерно 0.95 сек при максимальном количестве слотов в кадре равном 32 (при максимальной длине пакета данных 512 байт и скорости передачи данных 152 кБит/с).

Обязательные приём и передача информации происходят только в рамках последовательности микрослотов, что составляет от 2% – 5% времени кадра, таким образом затраты энергии на обязательное прослушивание сети при переходе на синхронизированный протокол уменьшаются на порядок. Остальные включения передатчиков и приёмников происходят строго согласно расписанию, закодированному в последовательности микрослотов. Нужно отметить, что потери на обязательное прослушивание могут быть больше в том случае, если оптимизируется формат кадра при особых конфигурациях сети. Оптимизация может быть продиктована большей необходимой пропускной способностью сети вблизи базовой станции и потери энергии будут второстепенными по сравнению с надёжностью передачи пакетов.

Широковещательные передачи производятся либо в слотах, занятых узлом, либо за счёт занятия слота 0. Вхождение нового узла в сеть производится с помощью занятия микрослота 0 и инициации протокола добавления узла в сеть, занимающего по времени 2 кадра, подобно схеме из [12]. В ряде реализаций из-за конструктивных особенностей приёмопередатчиков (большие времена включения и выключения устройств) два или более последовательных кадров могут формировать «эпоху», при этом в каждом кадре внутри эпохи могут быть некоторые ограничения на передачу пакетов от одного узла к другому узлу, но в рамках одной эпохи ограничений на передачу данных между узлами нет. Процедура вхождения узла в сеть заключается в поиске свободного слота, математическая постановка задачи состоит в раскраске связного графа (сети) в заданное количество цвета так, чтобы для каждой окрашенной вершины на расстоянии 2 от неё не было вершины, окрашенной в тот же цвет. В случае умеренной плотности узлов и соответствующей регулировке мощности передатчиков заданная раскраска становится возможной при числе цвета от 16 и более.

Основной особенностью моделирования синхронизированного доступа в канал является необходимость использования метода Монте-Карло для верификации модели. Так как сенсорная сеть географически распределена на достаточно большую площадь, при эксплуатации разные узлы находятся в разных температурных условиях, что вызывает случайные флуктуации частот и может привести к срыву синхронизации. Разумеется, моделирование с нулевыми погрешностями генераторов невозможно считать моделированием работы сети, так как оно является лишь отладкой протокола. Моделирование внешней среды (со всеми её случайностями) приводит к необходимости проведения достаточно длинных модельных испытаний (период модельного времени эквивалентный годам реального времени) и одновременной верификации функционирования сети. Под верификацией в данном случае понимается проверка наличия источников неверного функционирования протокола и выявление дефектов модели. Единственным ограничением в этом случае является сложность модели радиоканала, так как усложнение модели ведёт к экспоненциальному росту вычислительных затрат, особенно при большом количестве узлов.

Результаты моделирования

Построенная модель синхронизированной сети позволяет как отлаживать синхронизированный каналный уровень узлов сети без привязки к оборудованию, так и проводить длительные сеансы моделирования с помощью метода Монте-Карло. Модель реализует описанные ранее четыре уровня стека протоколов сенсорной сети: физический уровень модели включает отслеживание коллизий пакетов, сетевой уровень включает базовый маршрутизатор со статической таблицей маршрутизации. Работа базового маршрутизатора основана на методе посылки сообщения по маршруту с максимальным градиентом метрики расстояния до базовой станции [8]. Уровень приложений формализован до генераторов пакетов данных, интервал генерации пакетов может подчиняться равномерному или нормальному распределению (в большинстве практических задач информационные пакеты высылаются через фиксированный промежуток времени или при возникновении резкого отклонения измеряемой величины от заданной нормы – аварийной ситуации, которые случаются крайне редко).

Имеющаяся реализация модели сенсорной сети позволяет нам: построить модель, адекватно отражающую процесс функционирования синхронизированной сенсорной сети и проверить реализацию каналного уровня стека протоколов сети; подобрать временные параметры, при которых при имеющейся минимальной точности системных часов каждого узла каналный уровень сети будет корректно функционировать; оценить объём данных, который можно передать через сеть и определить пути оптимизации сетевой структуры для данной задачи. Среди практических результатов стоит отметить то, что модель позволила нам исправить несколько дефектов в спецификации сети, которые крайне трудно выделить из формального описания протокола. Оценка объёма передаваемых данных в единицу времени также является интересной для заказчиков сети, так как позволяет на раннем этапе скорректировать архитектуру сети для выполнения некоторой задачи и оценить стоимость и интервалы технического обслуживания сегментов сети.

Среди направлений развития модели можно отметить учёт затраченной энергии и моделирование поведения источников питания при различных климатических условиях, что позволит прогнозировать поведение сенсорной сети в условиях её постепенной деградации из-за разряда источника питания. Это важно для применения решения для трубопроводного транспорта и энергетических сетей в условиях малонаселённых районов и тяжёлых климатических условиях. Отдельную большую задачу представляет собой моделирование протокола с изменяемой длиной кадра, что полезно для увеличения пропускной способности. Немало задач связано также с оптимизацией передачи данных в сенсорной сети, где основная роль отводится алгоритмам маршрутизации – расширение пропускной способности сети связано с использованием особенностей структуры сети. Простые метрики, выбирающие всегда один кратчайший путь для прохождения пакетов, ограничивают пропускную способность сети вблизи базовых станций в разы, ограничивая соответственно сферы применения сетей.

Практические реализации

Разработанная модель послужила прототипом для создания сенсорной сети с синхронизированным доступом в канал. Алгоритмы разработанной модели перенесены на аппаратные платформы, проведено сравнение функционирования модели и реальной сенсорной сети.

В качестве платформы для отладки решения в полевых условиях разработаны узлы на базе платформы SmartARM фирмы Atmel (www.atmel.com) и трансиверов фирмы Semtech XE1205 (www.semtech.com), отличающиеся удобством разработки и низким энергопотреблением. Также перенесены алгоритмы на аналогичные промышленные узлы на базе микропроцессоров MSP430 фирмы Texas Instruments (www.ti.com) производства компании «Бинар» (www.binar.ru), сертифицированные для использования на газораспределительных станциях и принятые в эксплуатации ОАО «Газпром». В частности, узлы на базе различных платформ функционируют в рамках одной сети. Передача данных производится со скоростью 76 кБит/с в полосе частот 433 МГц (при работе в данной полосе частот не требуется обязательная государственная регистрация маломощных передатчиков).

Испытания показали, что поведение реальных узлов сети не отличается от поведения модельных узлов. Несмотря на упрощённые спецификации ряда процедур регистрации в сети, функционирование сети даже в случае значительной перестройки её структуры является стабильным. Проведены дополнительные исследования, посвящённые стабильной работе алгоритмов синхронизации узлов в сети, подтвердившие оценки ширины защитных интервалов, принятые в модели. Проведённым дополнительным исследованиям будет посвящена отдельная работа. При заявленной заказчиком интенсивности использования канала передачи данных ресурса батареи ёмкостью 20 А*ч достаточно для функционирования узла-маршрутизатора в течении года. Листовые узлы упрощённой конфигурации, не принимающие участие в маршрутизации пакетов, могут функционировать в «спящем» режиме до 5 лет от батарей ёмкостью 7-8 А*ч.

Основным результатом проведённых работ явилось подтверждение корректности модели сети. Опытные эксперименты показали, что синхронизированные сенсорные сети превосходят по экономическим параметрам

многие реализации сенсорных сетей и уже сейчас могут давать значительную экономию при проведении промышленной автоматизации объектов.

Выводы

В ходе исследования разработана модель канального уровня сенсорной сети, использующего схему доступа в канал с временным разделением. Модель перенесена на аппаратную платформу, решение протестировано в реальных условиях, подтвердило соответствие и корректность разработанной модели. Данное решение является оригинальным и позволяет перейти на новый уровень качества обслуживания в сенсорных сетях: улучшить надежность сети и уменьшить энергопотребление сети. Улучшение данных двух характеристик до приемлемого уровня (год функционирования каждого узла сети без дополнительного технического обслуживания) позволяет эффективно использовать сенсорные сети в промышленности, снижая стоимость владения АСУТП.

1. *Kiess W., Mauve M.* A survey on real-world implementations of mobile ad-hoc networks // J. of Ad Hoc Networks. – April 2007. – Vol. 5, Issue 3, P. 324–329.
2. *Levis P., Madden S., Polastre J. and dr.* “TinyOS: An operating system for wireless sensor networks” // W. Weber, J.M. Rabaey, E. Aarts (Eds.) // In Ambient Intelligence. – New York, NY: Springer-Verlag, 2005. – 374 p.
3. *Gay D., Levis P., Behren R., Welsh M., Brewer E., Culler D.* The nesC Language: A Holistic Approach to Networked Embedded Systems. // In ACM SIGPLAN Not., 2003. – Vol. 38(5): P. 1-11. – <http://nescc.sourceforge.net>
4. *Levis P., Lee N., Welsh M., Culler D.* TOSSIM: accurate and scalable simulation of entire TinyOS application // In Proc. Of 1st Int. Conf. on Embedded Networked Sensor Systems, Los Angeles, CA, USA, 2003. – P. 126–137.
5. *Умнов А.Л., Головачев Д.А., Ковалев П.А., Шишалов И.С.* Система сбора информации с узлов сенсорной сети // Радиотехника. Нелинейный мир. – 2005. – Т.2 № 4. – С. 249–253.
6. *Умнов А.Л., Головачев Д.А., Фильмонов В.А., Шишалов И.С.* Использование нелинейных рассеивателей в задачах связи и локации в беспроводных сенсорных сетях// Радиотехника. Нелинейный мир, 2004. – Т. 2, № 5/6. – С. 200 – 203.
7. *Rajendran V., Obraczka K., Garcia-Luna-Aceves J.J.* Energy-Efficient, Collision-Free Medium Access Control for Wireless Sensor Networks // In Proc. Of 1st Int. Conf. on Embedded Networked Sensor Systems, Los Angeles, CA, USA, 2003. P. 181–192.
8. *Abolhasan M., Wysocki T., Dutkiewicz E.* A review of routing protocols for mobile ad hoc networks // J. of Ad Hoc Networks, 2004. – Vol. 2(1), P. 1–22.
9. *Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, E.* Wireless sensor networks: A survey. Computer Networks, 2002. – Vol. 38(4). – P. 393–422.
10. *Карпова Ю.* Имитационное моделирование систем. Введение в моделирование с AnyLogic 5. – С-Пб.: BHV, 2006. – 400 с.
11. *Heidemann W.Ye,J., Estrin D.* An Energy-Efficient MAC protocol for Wireless Sensor Networks // In Proc. Oo IEEE Infocom Int. Conf., June USA, New York, NY, 2002. – P. 1567–1576.
12. *Kanzaki A., Uemukai T., Hara T., Nishio S.* Dynamic TDMA Slot Assignment in Ad Hoc Networks // In Proc. 17th Int. Conf. on Advanced Information Networking and Applications (AINA’03), 2003. – P. 330–336.