Improving the Usability of Existing Wireless Sensor Network Platforms

Author: Dan Collins
Supervisor: Dr. Richard Nelson

March 23, 2015
Abstract

There needs to be a way for a consumer to create Wireless Sensor Networks without requiring knowledge of how they work. By creating a simple provisioning process, this project aims to improve the usability and usefulness of future Wireless Sensor Network (WSN) projects. To be considered successful, the provisioning process needs to be simple and reliable for the user. Requiring knowledge of the underlying networking techniques should not be required to create the network to allow the user to focus on their application.
Contents

Acronyms ii

1 Introduction 1

2 Background 1
  2.1 Software .................................................. 1
    2.1.1 ContikiOS ............................................. 2
    2.1.2 TinyOS ................................................. 3
    2.1.3 RIOT OS ................................................ 3
  2.2 Hardware ................................................. 4

3 Method 4

4 Evaluation 5

5 Conclusion 5
List of Acronyms

WSN  Wireless Sensor Network
IoT  Internet of Things
6LoWPAN  IPv6 over Low-Power Wireless Personal Area Network
CoAP  Constrained Application Protocol
DTLS  Datagram Transport Layer Security
1 Introduction

WSNs are a collection of geospatially separated nodes capable of collecting data from sensors and relaying that data between other nodes on the network. These networks have been well studied for the past 20 years and a recent surge (called the Internet of Things (IoT)) has greatly increased the number of available options. With many of these available options lies a common problem - creating a system for use in the real world is still difficult. There needs to be a way for a consumer to create networks without requiring knowledge of how they work. It is not yet possible to go into a farming supply store and buy a new sprinkler that will interface with existing soil moisture sensors in order to keep a paddock sufficiently watered. Perhaps the reason for this is that a wireless networking solution that is easy to deploy in this kind of environment does not yet exist. This project covers the design and implementation of a provisioning method that allows users to enter some information about new nodes into their network and, by activating a discovery mode on the new node, the network will be able to automatically identify and provision the new node. The key aspect that needs to be observed throughout the design is that it needs to be easy for the user to both create new networks as well as add new nodes to existing networks.

2 Background

There are a number of different software and hardware platforms available for the creation of Wireless Sensor Networks. These platforms are at varying states of completion, reliability and adoption making the choice somewhat difficult. By choosing to create an IEEE 802.15.4 network we narrow down both the hardware and the software selection. This choice was made due to the range of available software solutions from various vendors as well as the availability of the ICs from a variety of silicon vendors. The variety creates competition which drives the price point down. New Zealand is limited to just the 2.4 GHz band of the IEEE 802.15.4 standard so the hardware needs to support a 2.4 GHz radio.

2.1 Software

There are a number of different projects that are accessible for this project, some of which are better suited than others. The ZigBee Alliance provides standards on creating a network which might be deployed in a home automation system. The vendors of the IEEE 802.15.4 radios often also provide a
ZigBee implementation which can be used to get the networking side of the project solved and out of the way. The disadvantage here is that you need a gateway node that can translate the networking used in ZigBee into standard IP based networking if you want the nodes to be able to access the internet. Worse than that is the need to come up with your own application protocol for exchanging the data between the nodes and the internet. Another available option is to use IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN). 6LoWPAN uses standard internet technologies which allow the nodes to use application protocols (such as Constrained Application Protocol (CoAP)) to easily communicate with the internet. A by-product is that each node has a globally routable IP address which might be useful in some applications. Having globally addressable nodes in a WSN is unlikely to be desirable as this can result in increased network load on the node leading to poor battery life. The gateway can be designed to limit the access to the WSN which mitigates these concerns. It should be noted that ZigBee also offers standards for ZigBee IP which also makes use of standard internet technologies however this has not yet become widely adopted like the original ZigBee standard.

A new project called Thread aims to create a new standard and standards body to manage the IEEE 802.15.4 space. Much like the Wi-Fi Alliance and the ZigBee Alliance, the Thread Group is developing a new standard to define how nodes in the Thread network will operate. Thread is designed to be the obvious choice when developing a product for the IoT. The goal is to be simple for consumers, secure, power efficient and support existing IEEE 802.15.4 hardware. This technology is not yet available and access to the standard requires membership fees which prevents any further evaluation. If the Thread Group meet their goals and the implementations of the standard really are interoperable it is likely to become the solution for WSN much like Wi-Fi became the solution for wireless networking.

This project will focus on a solution that uses the 6LoWPAN as it simplifies the infrastructure surrounding the WSN. While there are likely to be many other options that were missed in the search for a WSN framework, it is likely that the techniques discussed at the conclusion of the project will be applicable to other frameworks. The pros and cons of ContikiOS, RIOT OS and TinyOS have been considered.

2.1.1 ContikiOS

ContikiOS has been around for around 9 years and has seen active contribution on github from 91 different contributors. Marketed as “The Open Source OS for the Internet of Things”, ContikiOS targets a range of different hard-
ware platforms with and without IEEE 802.15.4 radios. There is an optional MAC layer that will improve power consumption by allowing routers within the network to sleep. These so called ‘sleepy routers’ reduce their power consumption between messages to facilitate battery power. The drawback to ContikiOS is that getting an example working is not as easy as the documentation suggests. Reading the blog postings of Isabelle Scheffield (a Waikato University WAND summer scholarship student who was asked to work with ContikiOS) shows a lot of effort being put into getting a connection from the WSN to the outside world. However, because of this summer research project, even more information on getting ContikiOS working is available. Plenty of the groundwork has been done and it is possible that the next step is this project.

2.1.2 TinyOS

TinyOS is an OS targeted at low power wireless devices such as those found in a WSN. A major point of difference, and a reason TinyOS was decided against, is that TinyOS uses its own programming language called nesC. nesC is an event driven language loosely based on C that is designed to make development for TinyOS easier. TinyOS is very popular and is used in a number of different research papers which gives TinyOS a large number of different components that can be put together. These components, such as different routing protocols, can be quickly wired together to build the user application. The real disadvantage to using nesC is not only having to learn a new programming language but also that the codebase becomes less portable. If the developer decides to change OS, a large porting effort will have to be made as not only the low level parts will have to be ported but the entire application will need to be rewritten.

2.1.3 RIOT OS

RIOT OS is comparatively new at 6 years and is aimed to create a more efficient operating system, comparing itself to ContikiOS, TinyOS and Linux. ContikiOS and Linux both use more RAM and ROM than RIOT OS which may permit the use of a lower cost microcontroller. RIOT OS, like ContikiOS, is bundled with sample applications and a native (Linux or Mac OSX) simulator to facilitate rapid development. RIOT OS is missing some of the supporting applications that ContikiOS has such as a gateway router which may mean the project gets delayed by first having to create a router. A gateway router is not strictly necessary to create a provisionable network with security however such a router is needed in order to make the network
useful. Without a gateway the internet standards don’t add value like being able to make a simple CoAP request to insert data into a database running on a Linux server in the cloud.

2.2 Hardware

The Texas Instruments CC2538 is well supported by all three software platforms considered. Not only is it well supported but it is also the chosen platform for previous work on ContikiOS conducted by WAND. Texas Instruments provides development boards with working radios (previous experience has shown that getting radio hardware working is difficult) and also a packet sniffing dongle which can be used to examine the RF network.

3 Method

By creating a development environment for the CC2538 development kit it will be possible to further evaluate the ContikiOS and RIOT OS software platforms. A simple network can be created and it will be possible to compare the experiences with both operating systems to choose one for further development.

With a working network, the next step will be to get some form of Datagram Transport Layer Security (DTLS) working. DTLS will provide secure transport, but it will create a new issue of how to provision new nodes on the network. Provisioning is how to connect a new node that doesn’t know the security credentials in a way that an attacker can not use to also gain access. From the user’s point of view, this method needs to be easy to use. If the user needs to recompile binaries with the network keys then the method is not simple enough. One method could be to have the user add the device to their network database (managed by a server on their LAN) by entering a unique number on the back of the device. The user then presses a button on the new node to put it into ‘discovery’ mode which allows the existing deployment to securely provision the new device.

The final issue to consider is how the user will create a new network with no existing nodes. This could be as simple as connecting their gateway to their network and installing an application that manages new nodes. With this set up, adding one or many nodes would follow the same process of entering the unique number and pressing the discovery button on the node.
4 Evaluation

To gauge the success of the project some kind of criteria need to be met. The aim is to create a secure and reliable sensor network that a user can quickly and easily add new nodes into. The project should be considered successful if the instructions for provisioning can be provided in a simple diagram with only a small amount of text. If the user needs to know any specifics of how the network operates behind the scenes then the result is not simple enough.

5 Conclusion

Existing and available technologies to create Wireless Sensor Networks are not simple enough to allow developers to create easy to use products. These technologies do not provide an adequate solution to build a secure network while also facilitating simple creation and modification. By ensuring the security methods used in the network follow acceptable internet standards, such as DTLS, the hard part of verification is already done. The rigorous mathematical proofs behind the workings of the cryptography will already have been done. To create a simple user experience thought will be put into how to get the security credentials into the nodes of the network without requiring the user to understand how the nodes communicate. It is thought that by solving this issue with existing solutions new products will start to emerge allowing the internet of things to be an even more pervasive technology.