***Developing Mobile Raspberry Pi Base Station Cloud***

For Wireless Sensor Networks

Andrew K. Wilson

Department of Information Technology

Radford University

Radford, Virginia, U.S.

awilson8@radford.edu

*Abstract*—There are several issues in the institution of a mobile wireless sensor network including network lifetime and data collection. We propose the institution of a mobile base station cloud to simultaneously solve both matters. We will build a cloud base station network from multiple nodes that have the ability to move to the optimal position for both network lifetime optimization and data collection.

Keywords-wireless sensor network; mobile; cloud; base station; data collection; optimization;

# Introduction

The base station is essentially the brain of a wireless sensor network. It handles data aggregation and processing, intercommunication of sensor motes and the end user, as well as optimization strategies to extend the lifetime of the network.

The necessary numbers of hops required for message passing are decreased when instituting multiple base station node. A problem arises in this structure when focusing on network lifetime as a base station nodes closest mote takes on not only its own messages but also those of the motes in the near vicinity. The institution of a base station cloud that periodically moves to new locations based on an optimization problem solves this because the “routing” mote(s) continually change. Additionally, the institution of a WiTricity charging protocol will also greatly increase the network lifetime.

Another issue is the loss of sensor motes that can move or can be moved. A multiple node mobile base station can recover lost motes based on last known location and signals sent from the mote. Since each base station node is mobile it can move to the last known location of said mote and physically return it to the network and reestablish its connectivity.

Previously, a base station in a wireless sensor network has been a single, static machine with limited capabilities. However, the institution of a mobile cloud base station gives way to much greater computational power than that of a single node base station through way of distributed computing, extends the lifetime of the network by moving each node to the optimal location, and the ability to physically restore the sensor motes of the network. Thus, in this paper we propose a mobile base station cloud comprised of multiple Raspberry Pi powered Lego Mindstorms NXT robots.

# Benefits Of Base Station Cloud

There are many benefits to the institution of a base station cloud which include the following:

1. For Large Data Collection
2. For Decision Making Capabilities
3. To Solve Optimization Problems

These benefits cannot be easily implemented without the formation of a cloud. Figure 1 shows the proposed cloud network and the communication of its nodes. The benefits of this are better defined below:

1. *For Large Data Collection*

Data collection is the cornerstone of a WSN. The institution of a cloud base station makes data collection much easier as the large computations required can be distributed across the cloud to the many nodes for quicker, more efficient work.

For this purpose we have proposed the incorporation of Hadoop/MapReduce.

1. *For Decision Making Capabilities*

The base station cloud allows for the ability of decision-making distributed across the multiple nodes. For instance, in a wireless sensor network, when large amounts of data are gathered from motes by the base station cloud it takes a large amount of computational power to sort and become useable. Distributing this work across the cloud, again, fulfills this need. For this purpose we have proposed the incorporation of Hadoop/MapReduce.

1. *To Solve Optimization Problems*

Decision-making can be done by solving an optimization problem. Again, solving optimization problems needs a large amount of computational power. The base station cloud again solves this issue. For instance, in a wireless sensor network using a distributed power protocol, when a mote shouts for help (in need of power) and optimization problem must be solved to determine what is the most efficient way to get the power to that mote (the fastest path with the least amount of energy leakage). This optimization problem is solved much more efficiently through distributed computing across the cloud.

1. *Related Work*

Figure 2 depicts two different sensor network deployments as proposed by [3]. Sensor node A in Figure 2(b) is only one hop away from its nearest base station while in Figure 2(a) it takes multiple hops. Sensor B is the same distance in both Figure 2(a) and 2(b). Therefore, by employing multiple base stations “we have effectively either reduced or retained the hop count of each sensor node in the network” [3].

This is most important from the standpoint of energy consumption since the energy expenditure of a message from its sensor node to its nearest base station is directly proportional to the number of hops it must travel. Thus, the institution of multiple base stations reduces the energy consumption per message.

# Benefits Of A Mobile Base Station

There are many benefits of a mobile base station in a mobile or non-mobile wireless sensor network. The following 3 are most pertinent to this research:

1. *Communication*
2. *To Build A Mobile Cloud*
3. *For Decision Making Capabilities*

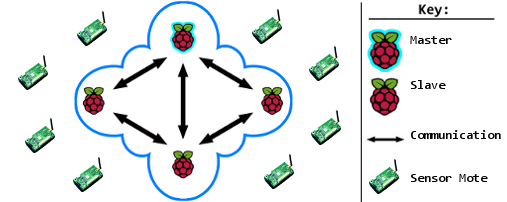
The addition of a mobile base station gives way to the ability of the nodes of the base station to physically move to different areas throughout the network. By doing so, the three previously mentioned benefits become possible. These benefits are better defined below:

1. *Communication*

Communication between motes and the base station(s) is essential in a WSN. The proposed mobile base stations increases the communication because they would be able to physically move to the most beneficial position.

For instance, in the institution of a WiTricity charging protocol in the WSN, when a mote “shouts” for help in need of charging the base station could move towards that specific mote to send electricity much more efficiently.

Another example is when a mote gets disconnected from the network. When the network recognizes that a mote is missing, a node of the base station can move to the motes last known location (as determined by the network) and search, based on some sort of signal emitted by the mote, and



1. Proposed Cloud Network

physically bring the mote back in range of the network and reestablish the connection.

1. *To Build A Mobile Cloud*

With the introduction of multiple mobile nodes comes the ability of a mobile cloud. In the case of a Mobile Wireless Sensor Network (MWSN) where the motes of the network can move it is necessary for the base station, in this case the network of mobile nodes that creates a cloud, to also be mobile.

As the motes of the MWSN move the mobile cloud base station would move with it. Also, the cloud base station could move to the more beneficial position (in terms of efficiency).

For instance, it might be more beneficial for more nodes of the cloud base station to be in one area of the network for more efficient data collection.

1. *For Decision Making Capabilities*

The institution of a mobile cloud base station can increase the network lifetime by making decisions based on the location of the base station nodes and their ability to move. By determining the location of each node we can determine the most efficient path of movement to assist throughout the network.

We have proposed the institution of nodes that will work in conjunction with a WiTricity charging protocol to solve an optimization strategy to extend the lifetime of the network.

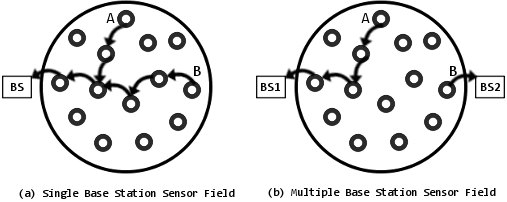
1. *Related Work*

It has been show by [4] that sensor nodes that are only one hop away from a base station consume more energy as they essentially become the main routing node – forwarding not only their own messages but also those of the others belonging to that base station. This depletion of energy renders the node and others like it in the network inoperational that in turn does the same to the network.

[3] suggests two strategies for periodically changing the locations of the base station nodes to increase the lifetime of the sensor network.

# How To

Each node of the mobile base station cloud is most significantly comprised of one Raspberry Pi and one Lego Mindstorms NXT robot connected through the NXT’s provided USB A-to-B cord. The following explains the set up for both parts:



1. Circular Field of Interest
2. *Raspberry Pi*

We first start out with a fresh installation of Raspbian on our Raspberry Pi. This can be obtained from <http://www.raspberrypi.org/downloads> but most packages come with this pre-installed on an SD card. We then follow the typical setup instructions, making sure to expand the filesystem to ensure that all of the SD card storage is used.

Next, update the operating system with the following commands:

sudo apt-get update

sudo apt-get upgrade

Next, to ensure that we can run Java (which will be covered later), we must install the Java Development Kit (JDK). Some Raspbian images come with this pre-installed but in the case that it does not enter the following in a command prompt:

sudo apt-get install oracle-java7-jdk

1. *Lego Mindstorms NXT*

To better control the NXT we will flash the brick with LeJOS, a firmware replacement allowing the NXT to be programmed with Java. LeJOS can be obtained from <http://www.lejos.org/nxj-downloads.php>. The Java Development Kit (JDK) must be installed on your PC. To flash the brick it must be connected to your PC or Raspberry Pi with the supplied USB cable (libusb must be installed – it is preinstalled on many Linux distributions) and in firmware update mode by pressing and holding the reset button located in the top left corner on the back of the brick (the brick will make a ticking noise when in firmware update mode).

Once done, run either the nxjflash or nxjflashg command, the latter of which opens the GUI version. The connected NXT bricks in firmware update mode will be listed. Select the one you wish to flash and from there the LeJOS firmware should be immediately uploaded.

1. *Testing The Flash*

To test that the flash worked correctly we will write, upload and run a simple “Hello World” program.

1. Create a file named HelloWorld.java with the following contents

|  |
| --- |
| import lejos.nxt.Button;  public class HelloWorld {  public static void main (String[] args) {  System.out.println(“Hello World!”);  Button.waitForAnyPress();  }  } |

1. Compile and Link

Open a command prompt and execute the following command:

nxjc HelloWorld.java

This will compile the .java file and create a file name HelloWorld.class. Since the LeJOS firmware does not directly execute .class files we must link the HelloWorld program with the LeJOS linker using the following command:

nxjlink -o HelloWorld.nxj HelloWorld

“This will load the class HelloWorld and all dependencies. The classes are then merged together into a single file.”

1. *Upload to Brick*

For the file to be executed it must be uploaded to the brick. To upload the .nxj file to the file, execute the following command:

nxjupload -r HelloWorld.nxj

The –r parameter means that the program will be executed immediately after uploading. “By omitting the parameter, nxjupload will just upload the file and it will not [be] executed immediately” [1, 2].

1. *Cloud Network*

The mobile base station cloud is essential to this work. To build the cloud we will set up one Raspberry Pi as a host access point according to [5]. This will act as the master in the master-slave relationship of the cloud network. The following are necessary for implementation:

* Internet connection through Ethernet
* Wi-Fi USB hub
* Raspberry Pi

The host access point is set up as follows:

1. Follow the directions in IV.1 to properly set up the Raspberry Pi (the JDK is unnecessary)
2. Install isc-dhcp-server and hostapd with the command sudo apt-get install hostapd isc-dhcp-server
3. Set up DHCP server by editing the file /etc/dhcp/dhcpd.conf to resemble

#option domain-name “example.org”;

#option domain-name-servers ns1.example.org, ns2.example.org;

#If this DHCP server is the official DHCP server for the local

#network, the authoritative directive should be uncommented.  
authoritative;

1. Add the following to the bottom of the same file

subnet 192.168.42.0 netmask 255.255.255.0 {

range 192.168.42.10 192.168.42.50;

option broadcast-address 192.168.42.255;

option routers 192.168.42.1;

default-lease-time 600;

max-lease-time 7200;

option domain-name "local";

option domain-name-servers 8.8.8.8, 8.8.4.4;

}

1. Change the following line in /etc/default/isc-dhcp-server to resemble

INTERFACES=“wlan0”

1. Set up wlan0 for static IP by first running sudo ifdown wlan0
2. Edit /etc/network/interfaces to resemble the following and put a # in front of all other lines

auto lo

iface lo inet loopback

iface eth0 inet dhcp

allow-hotplug wlan0

iface wlan0 inet static

address 192.168.42.1

netmask 255.255.255.0

1. Assign static IP to Wi-Fi adapter with the command sudo ifconfig wlan0 192.168.42.1
2. Configure the Access Point by creating a new file /etc/hostapd/hostapd.conf and adding the following contents – changing ssid= to the desired network name and wpa\_passphrase= to the desired network passphrase

interface=wlan0

driver=rtl871xdrv

ssid=YourSSID

hw\_mode=g

channel=6

macaddr\_acl=0

auth\_algs=1

ignore\_broadcast\_ssid=0

wpa=2

wpa\_passphrase=YourPassPhrase

wpa\_key\_mgmt=WPA-PSK

wpa\_pairwise=TKIP

rsn\_pairwise=CCMP

1. Edit the file /etc/default/hostapd to tell the Raspberry Pi where to find the configuration file by changing the following line to resemble

DAEMON\_CONF=“/etc/hostapd/hostapd.conf”

1. Configure Network Address Translation by editing the file /etc/sysctl.conf and adding the following on a new line at the bottom

net.ipv4.ip\_forward=1

1. Run the following to activate it immediately

sudo sh –c “echo 1 > /proc/sys/net/ipv4/ip\_forward”

1. Run the following to create the network translation between the Ethernet port and Wi-Fi port

sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

sudo iptables -A FORWARD -i eth0 -o wlan0 -m state --state RELATED,ESTABLISHED -j ACCEPT

sudo iptables -A FORWARD -i wlan0 -o eth0 -j ACCEPT

1. To make this happen automatically on boot run

sudo sh –c “iptables-save > /etc/iptables.ipv4.nat”

1. Edit the file /etc/network/interfaces and add the following to the end

up iptables-restore < /etc/iptables.ipv4.nat

1. Update hostapd by getting the new version with the following

wget <http://www.adafruit.com/downloads/adafruit_hostapd.zip>

1. Extract the file

unzip adafruit\_hostapd.zip

1. Move old version

sudo mv /usr/sbin/hostapd /usr/sbin/hostapd.ORIG

1. Replace with new version

sudo mv hostapd /usr/sbin/hostapd

1. Change permissions so that it is valid to run

sudo chmod 755 /usr/sbin/hostapd

1. Manually test by running the following

sudo /usr/sbin/hostapd /etc/hostapd/hostapd.conf

1. Check on another machine to see the created network
2. Set up as daemon by running the following

sudo service hostapd start

sudo service isc-dhcp-server start

1. If both start successfully, to make this happen automatically on boot run

sudo update-rc.d hostapd enable

sudo update-rc.d isc-dhcp-server enable

1. Run the following to reboot and verify that everything works automatically

sudo reboot

1. Enable auto login by editing the file /etc/inittab and changing the following line to resemble

#1:2345:respawn:/sbin/getty 115200 ttyl

1. Add the following line directly below the previous

1:2345:respawn:/bin/login –f pi tty1 </dev/tty1 >/dev/tty1 2>&1

The master of the network is now created and gives the ability to secure shell (SSH) in to and out from the slave nodes. At the time of this writing the cloud network has not been completely integrated.

# Future Work

The cloud is an essential piece to our solution. The cloud is in progress as shown above. This will allow the intercommunication of the sensor motes and the base stations.

In addition to the devices used to create the mobile base station it would be beneficial to develop software that would allow the user to physically drive each individual node of the of the base station as well as view data as it collected.

As mentioned previously, the institution of a WiTricity charging protocol alongside the mobile base station cloud would greatly extend the lifetime of the sensor network.

Finally, we will institute Hadoop and MapReduce to solve a linear problem related to network lifetime in the institution of a WiTricity charging protocol and a mobile base station cloud.

##### Acknowledgment

//If Necessary

##### References

1. LeJOS, “Getting started on linux,” <http://www.lejos.org/nxt/nxj/tutorial/Preliminaries/GettingStartedLinux.htm>, November, 2013.
2. LeJOS, “Writing and running your first lejos nxj program,” <http://www.lejos.org/nxt/nxj/tutorial/Preliminaries/FirstProgram.htm>, November, 2013.
3. S.R. Gandham, M. Dawande, R. Prakash and S.Venkatesan. Energy Efficient Schemes for Wireless Sensor Networks With Multiple Mobile Base Stations. *Globecom 2003*, pages 377-381.
4. A. Manjeshwar and D.P. Agrawal. TEEN: a routing protocol for enhanced efﬁciency in wireless sensor networks. *Intl. Proc. of 15th Parallel and Distributed Processing Symp.*, pages 2009 – 2015, 2001.
5. Adafruit, “Setting up a raspberry pi as a WiFi access point,” <http://learn.adafruit.com/setting-up-a-raspberry-pi-as-a-wifi-access-point/install-software>, March, 2014.