**Lightweight Easily-deployable Nodes for Temporary Wireless Mesh Networks**

Ellana, Allen

Rivera, Benjamin

Tison, Alexander Alfonsin

Tonoike, Tomio

**I. Introduction**

**1.1 Project Context**

Mobile phones are regarded as ubiquitous in the field of communication (Schmidt et al., 2006). These mobile phones rely on telecommunication networks to enable communication (Woodford, 2016). This research will focus on creating an ad-hoc mesh architecture that lets mobile phones connect to each other using Wi-Fi(802.11 standard) without relying on the telecommunications network.

Moore's Law states that, as technology advances, hardware cost and size decreases. The rapid growth of technology is creating more and more electrical components that have more functions with less sizes and cost(Moore, 1965). This makes it possible today to easily create and deploy a network composed of multiple small nodes on a budget.

These nodes will be interconnected to create a partial mesh network. This network relays the information to each other through protocols that effectively establish reliable communication between devices. Having more nodes creates a network that accommodates a bigger land area which increases the range and effectiveness of the network. It can also be easily scalable so the number of individual nodes can be increased and decreased as the situation demands. The reduced prices on the hardware components also made network nodes easily replaceable if upon deployment a single device stops functioning.

When a device sends a message using the chat application, the message will be sent to an access point, which is a node of the mesh network of multiple access points. The access points that will receive the message will rebroadcast it through the mesh network until the message is received by the particularaccess point where the endsmartphone is connected to.

**1.2 Purpose and Description**

According to Business Monitor International (2012), the number of its subscribers will reach 117 million by the end of 2016. Which meansPhilippines is a heavy user of mobile phones.Mobile phones primarily use radio communications to send and receive data. With this information, it can be said that Filipinos rely heavily on radio communications as a mean to connect with each other. When the commonly used cellular networks are down or is out of range, the mobile phones that majority of people rely on becomes useless as a tool for communication.   
   
Time and time again the services network providers give are cut by natural disasters physically destroying connections and towers or by the occasional power failures. The effect of these cuts is more severe in remote provinces where the equipment are scarce and obsolete. Due to the design of the current network infrastructure, it takes a significant time to repair and restore the connection.

A good example of a disruptive event would be Typhoon Yolanda, internationally known as Haiyan. The storm is considered to be one of the strongest storms recorded on the planet (Mullen, 2013). It struck the Philippines on November 8, 2013 affecting 1,473,251 families with a casualty count of 6,300. The typhoon knocked out power lines and damaged the 63 cell sites of all the carriers in the area. (Camus, 2013) Right after the storm passed, the whole area was leveled and rescuers had to search for survivors with almost no radio communications as the network infrastructure also went down with all the other structures.  With communications down, coordination of actions among response teams in multiple areas was almost impossible. It added an additional layer of difficulty for the teams in handling situations that demand cooperative action. Even radio, TV and news stations found it difficult to communicate with their own teams on the ground. With the severity of the situation, the UN took notice and decided to help because the local agencies in the area are also having difficulty restoring the communications network. (Ambil, 2013)    
   
Despite Filipinos being heavy users of mobile devices, there are still places in the Philippines that are not reached by telecommunication signals.Examples are mining sites and rural towns who do not even have access to constant electrical supply. Mining sites are usually located far from any city, town, or any place that is being used by people because of its physical hazards and its nature of producing chemical waste(Section 19 of Republic Act No. 7942). Since the target of cell sites are mostly highly populated areas, mountainous sites where population density is not great usually are out of their range. This makes it difficult for the workers to communicate with anyone outside the area like family or friends.Besides the issue on priority, the geographical structures of the Philippines also adds to the difficulty of building new cell towers to give telecommunication access to rural places. These difficulties limit the capabilities of the devices most Filipinos own to only the location where network infrastructure are present.

This research on an easy-setup implementation of mesh networking will have many applications.In events of a network failure or a power outage after a natural calamity for example, real-time connections are preserved by creating a reliable network infrastructure that can be easily established and removed as needed. This will improve the efficiency on all levels because it is a fact that communication is key to good coordination between parties. It could be used by emergency response teams' during crises as it will improve their coordination by creating an open line of communication between them when cellular networks are down and when there are blackouts. It can also be used by workers in remote mining operations where signals are obstructed and completely blocked big geographic formations. It can be used by just anybody that wants their own personal network that is ‘off the grid'. If improved to withstand bad weather, it can even be deployed in open waters where fishermen can communicate to each other.

The end product of this research will be very adaptable to the current situation becauseit uses a resource that majority of the Filipino citizens have, the smartphone. Because of this, the network can be easily implemented and used. It also won't require and additional instruction or equipment aside from the lightweight device and its deployment method(i.e. balloon/pole/tree/post).(See Figure 1 in Appendices)

**1.3 Objectives**

This study aims to find a cost efficient and easily deployable method to improve the communication among users of Wi-Fi capable mobile devices who are situated in a place where there is no currently available network infrastructure to connect to.

**Specific Objectives**

• To connect multiple microcomputers as intermediary nodes of a mesh network;    
• To enable data transfer between microcomputers;    
• To connect mobile devices to the nodes of the network;    
• To relay a message to and from endpoints of the network(mobile devices);    
• To enable the network nodes to automatically detect and connect to nearby nodes;

• To test and modify different available software in order to find the one most suitable to be used locally;  
• To create a solution that will be cost effective.

**1.4 Scope and Limitations**

The scope of this research covers the use and modifications of existing technologies to develop network nodes that can be used to establish a private network. This private network will consist of smartphones with 802.11 capabilities as end devices and microcomputers as nodes of a wireless mesh network.

The wireless mesh network will be intended for areas with no available network service from telecommunications companies as a viable temporary communication infrastructure,  
   
The research includes the nodes being raised in the air with the use of balloons to cover an area with network connectivity which is anchored to the ground using heavy boxes.   
   
The study covers the development of the network nodes along with its input and output interfaces. Integrating the networking protocols with hardware consisting of microcomputers and 802.11 devices to establish a reliable connection between 2 or more smartphones. The connection will be able to allow users to send messages to each other using an existingpeer to peer chat application called Walkietooth.

After the development of the nodes, further studies will be done to fully know the ideal considerations (i.e. effective range, data transfer speed, and traffic capacity.) for proper deployment and application.

The paper will also discuss different scenarios that are likely to happen in the Philippines wherein these nodes will be useful. The ideal way of positioning the nodes based on these scenarios will be included.

### **II. Related Literature**    Inspirations

The research is inspired by the disaster that wreaked havoc in Leyte, Philippines on November of 2013. This disaster, Typhoon Yolanda, almost levelled some of the affected areas damaging all infrastructures standing, including telecommunication sites. It took two weeks to repair these sites and at that time telecommunication services were down. The researchers decided to address this communication problem for the research. Hence, the project’s goal is to provide a temporary means of communication to areas affected by these kinds of calamities and help in post disaster recovery operations. A possible solution that has been focused on by the researchers is the use of Wireless Mesh Networks (WMN).   
  
In the beginning, the researchers thought of ways to enable communication among people inside an area without relying on telecommunication services. The idea was to utilize the use of smartphones since a large number of the population use it for communication. The initial solution was to create a peer-to-peer mobile chat application. Peer-to-Peer applications let users communicate with each other from 2 end devices (Retrieved in August 15, 2016 from <http://medianetlab.ee.ucla.edu/papers/chapter_P2P_hpark.pdf>). Chat applications usually need a web server in order to work. A web server may be offline, such as XAMPP, or online, such as an online server or a registered domain . The research needs a mobile chat application that is not provisioned with a web server and should rely only on a peer-to-peer networking for the sending of messages. One way of doing this is through the use of Wi-Fi Alliance’s Wi-Fi Direct™ connection (Retrieved in August 15, 2016 from<https://developer.android.com/guide/topics/connectivity/wifip2p.html>).  
  
Wi-Fi Direct™ enables 802.11/ Wi-Fi capable devices to communicate with each other without an internet connection or an access point. It makes devices emit a signal to let other Wi-Fi Direct capable devices know that a connection is available. It can be used to send files, sync data, and other things that originally needed internet as long as there is a connection between the devices (Retrieved in August 15, 2016 from <http://www.wi-fi.org/discover-wi-fi/wi-fi-direct>).   
  
The researchers discovered however, that this solution was already made by an organization called Open Garden. They developed FireChat, a mobile application that utilizes Bluetooth and peer-to-peer Wi-Fi of smartphones to create a mesh network that enables users send and receive messages and photos that are sent by other users within the network. It has an initial range of 200 feet (60.96m) that can be extended by tens of thousands users (Retrieved in August 15, 2016 from <https://play.google.com/store/apps/details?id=com.opengarden.firechat&hl=en>).  
  
FireChat has two features that are significant to the study, peer-to-peer public and private messaging. Both features utilize the multi-hop capability of the app such that when a message is sent, it is broadcasted by the user’s device to other devices which in turn repeats the process. If the message is public, the message can be opened by all devices broadcasting it. If the message is private however, it is encrypted by the app so other devices will not be able to open it and will only continue rebroadcasting until the message reaches the receiver’s device where it will be decrypted (Retrieved in August 15, 2016 from <http://www.opengarden.com/how-to.html>).  
  
Although the researchers find FireChat as a good communication tool that does not rely on telecommunication services, it has a major problem if it will be used in disaster recovery. As stated above, it relies on extending its range by hopping to multiple smartphones with each of them having a range of 60.96 meters. This means that it is best for dense communities with a lot of users but bad for communities with sparse users. If users are more than approximately 61m apart, they will not be able to join the network and this issue can only be solved by adding other smartphones running FireChat between them. FireChat is free but proprietary software meaning its codes are not available to the public. The researchers could not study its structure and how the application actually works so it difficult to try and find ways to create a range extender.   
  
This pushed the researchers to look into other possible solutions that has the same general idea but does not share the same limitation. This kind of solution is found in Mobile Ad-hoc Networks (MANET).  
  
MANET is type of network that does not rely on any telecommunication service. Devices connected to the network act as routers or nodes that relay data that they receive to other nodes. This enables them to send data through multi hopping. Each node is free to move within the network and if ever they are disconnected because of range or other reasons, the network will adjust accordingly. This means that MANET’s structure is ever changing and continues to adjust to whatever connections are available (Retrieved in August 15, 2016 from <http://airccse.org/journal/graphhoc/papers/0310jgraph7.pdf>). To solve the issue of range when using MANET, a device that has networking capabilities should act as a node and placed between users far from each. This in effect, extends the range of the MANET. The researchers tried to look into two possible devices, Arduino and Raspberry Pi.  
  
Initially the researchers tried to use microcontrollers as nodes for the network since it is generally cheaper than routers and Raspberry Pi. Since the goal of the project aims to deploy temporary networks to post disaster scenarios, each node should be cheap and easy to replace. The microcontroller that was addressed by the researchers was the Arduino. According to the Arduino website, “Arduino is an open-source prototyping platform based on easy to use hardware and software”. Currently it is mostly used to read sensor values, online messages, or data from an external memory card then create an output based on the program of the user. Arduino is aimed towards students because of its simplicity in design and application. Its flexibility makes prototyping easy even for users with limited knowledge in electronics and also allows experts to build complex projects (Retrieved in August 19, 2016 from <https://www.arduino.cc/>). Arduino is a powerful tool to experiment on small programs and sensors but is not fit for the purpose of the research. Wi-Fi modules are not integrated into the circuit that comes out of the box so modules that can be attached called shields must be purchased first. One such module that enables network is called ESP8266. ESP8266 is currently used for small projects concerning the Internet of Things (IoT) connecting multiple home devices and sensors. Arduino is apt for simple applications like networking or sensor networks but other more complicated projects are not recommended. There have been problems reported by many that tried to use the device as a routing tool. It had been discussed that Arduino has poor RAM capability and also poor computing power to act as a node for routing in a high activity mesh network. Most of the problems that arise in the form of code crashes are caused by its limited memory. (Retrieved in August 18, 2016 from <http://forum.arduino.cc/index.php?topic=166151.0>).  
  
Another device that can act as nodes is the Raspberry Pi. It is a credit card sized microcomputer that runs mostly on Linux-based operating systems. Raspberry Pi is created by Raspberry Pi Foundation with the objective of giving children a background in computer programming. However, because of its available functionalities, it is currently used in complex computer projects wherein it sometimes replaces routers or even the computer itself (Retrieved on August 18, 2016/ <http://elinux.org/RPi_Hub>).

### Wireless Mesh Networks

As Raspberry Pis are confirmed to be able to act as nodes, the next step is defining its network topology. The network topology chosen for the project is the mesh topology. A wireless mesh network is a network setup wherein devices (or nodes), are interconnected with each other. WMN is considered to be a type of MANET. This is different to the widely used star network where all devices are just connected to a central device (i.e. Wi-Fi Access Points). When a connection is initiated in the mesh, each node acts as a transmitter that carries the connection to the next connected node. Messages are not relayed to a central device like in a star topology so messages have to be properly routed through multiple intermediary nodes until it reaches its destination. This setup makes the network highly volatile because if the mesh consists of only a few nodes, the loss of a connection to one can separate whole networks. The solution to this problem of networks being fragile is to add numerous additional nodes that create redundant routes. These routes allow connections to be 'self-healing' or to just reroute messages if in case some intermediate nodes fail. There are two kinds of mesh topology: full mesh topology and partial mesh topology. In the full mesh topology, all nodes are interconnected to each other. While on the partial mesh topology, nodes only communicate to neighbor devices and only relay communications to the intended nodes(source and why manet is not applicable).  
  
To be able to make smartphones connect to the network, Wireless Access Point (WAP) is addressed for the project. An access point lets devices to be connected to it to let that device be connected with the network.  
  
WMN uses different routing protocols to handle its connection. Routing protocols dictates where and how a router/ node distributes packets. There are a numerous routing protocols currently available and each of it has its own design and purposes (Cisco Networking Academy, 2014). The key considerations that raised the need for efficient routing when implementing mesh networks are the overhead of ID per hop jumped, maintenance of nodes, send/receive overhead, power consumption, and interference. It is also important when choosing the proper routing protocol for the network to consider that table based protocols grows bigger as nodes increase and packet header grows bigger as more nodes are included. The routing protocols considered for the project are the Ad-Hoc On-Demand Distance Vector Routing Protocol (AODV) and the Optimized Link State Routing Protocol (OLSR).  
  
Ad-hoc On-Demand Distance Vector Routing (AODV) is a Distance Vector routing protocol specifically designed for mobile ad-hoc networks. Nodes only search for a route when it needs to transmit/retransmit a message, hence on-demand. It does not need periodic advertisements and only uses connection when needed; this means that there is less traffic in the network allowing it to have a bandwidth that is significantly higher than other routing protocols (Perkins & Royer 2003).  
  
Optimized Link State Routing Protocol (OLSR) is a revision of Link State routing protocol. OLSR is designed for mobile ad hoc networks wherein all devices connected to the network act as a node. One of its main differences from link state routing is that every node in OLSR sets a multipoint relays (MPR) (Clausen & Jacquet, 2003). Nodes will only receive transmissions from these MPR and allows them to control its traffic (Retrieved in August 17, 2016 from <https://www.youtube.com/watch?v=3V19nPxpMp8>). MPR are selected in such a way that the node that selected them will receive all transmissions in the network but with less duplicates. These duplicates contribute to the flooding that is experienced by nodes in link state routing.

### Related Projects

For other projects related or similar to the project: Broadband-HamnetTM is a “…a high speed, self-discovering, self-configuring, fault tolerant, wireless computer network…” (Kinter, 2010) Formerly called HSMM-MeshTM (High-Speed Multimedia), and the main motivation is giving communication during emergency situations given that their network can act as an ad hoc. They provide network that uses OLSR and is currently supported by different Linksys routers and Ubiquiti radios. They do, however, encourage users to test their work on other devices (Retrieved in August 17, 2017 from<http://www.broadband-hamnet.org/images/hsmm_docs/WRT54Shop.pdf>).  
  
Commotion is an open source networking tool that provides mesh networks. Their objective is to create a tool that can be set up and used by anyone. Commotion can share internet access, applications, and files when one of the nodes has it. A lot of its properties are hardware and situation dependent but it is possible to connect thousands of nodes together. Commotion is supported in different routers, Linux and Mac computers, and rooted Android phones (Retrieved in August 18, 2016 from <https://commotionwireless.net/>).   
  
Project Byzantium is an operating system for implementing a wireless ad-hoc mesh network which connects devices using 802.11a/b/g/n without relying on the internet. It is a distribution of Linux which can be installed to a device or run from a removable media. Any Wi-Fi enabled computer can be made into a Byzantium node just by running Byzantium Linux. It uses OLSR as its routing protocol. Each of these nodes connects to each other directly, forming an ad-hoc mesh network. If one of the nodes in a network has an active internet connection, all other nodes can use this connection too. Byzantium Linux is already available in Github however, the last commit was done back in 2014.  
  
Serval Mesh is an application by a group of network enthusiasts by the name of Serval Project. It lets mobile phones make the use of Wi-Fi and Bluetooth to enable communications even if it is not connected to the GSM network. (Retrieved on August 16, 2016/ <https://play.google.com/store/apps/details?id=org.servalproject&hl=en>). The Serval Mesh application will benefit the project since the nodes will serve as an access point and everyone will be under the same Wi-Fi network. The experimental application will serve as a good proof of concept of how real communication can be established and maintained in the researched network deployment.  
  
The Raspberry Pi 3 Model B was chosen to be used as nodes for the project since the specifications of the Pi provides more flexibility than the other microcontrollers/microcomputers. Raspbian was the chosen operating system for the Pi’s for its simple/user-friendly interface and its functionalities. The research will be using a partial mesh topology in combination of a star for the network topology. Each microcomputer will have separate star networks where end devices can connect to it. The central bus of these star networks will act as nodes of a mesh network that will be used to establish the connection between two end devices in separate nodes. The access points people use commonly for their communications that only depend on only one antenna for all devices connected. This creates an easier to manage network where everything can be controlled and monitored through the central device. (Retrieved on August 18, 2016/ <http://www.webopedia.com/quick_ref/topologies.asp>).

# Technical Background

Currently, there are a lot of available hardware and software that can be used to achieve the goals of this study. However, to be able to find the ideal protocols and software tools, it is necessary for the hardware used to be flexible when it comes to software compatibility. There are portable routers available online that are capable of ad hoc mesh networking which can be used as nodes for the study. These routers however, have limited compatibility when it comes to software so it might be necessary to purchase different kinds of them in order to test software. To save costs, the study will focus on using ARM based devices since they can achieve the same functionalities of a node and more.

**Raspberry Pi 3 Model B**

The latest model of Raspberry is the Raspberry Pi 3 Model B which was released on February 2016. It has 1.2GHz 64-bit quad-core ARMv8 CPU, a built-in 802.11n Wireless LAN and Bluetooth 4.1 (Retrieved on August 16, 2016/ <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>). It uses Broadcom System on Chip (SoC) BCM2837. SoC is an integrated circuit that combines all computer components such as CPU, graphics processor, memory, and other peripherals into a single chip(Anthony, 2012). BCM2837 has the same basic circuit design as the SoC of both Raspberry Pi 1and 2 so it is backward compatible. The hardware update made Raspberry Pi 3 perform 50-60% better than Pi 2. Previous models of Raspberry Pis do not have a built-in Wireless LAN and Bluetooth making Pi 3 cheaper than Pi 2 because of the additional costs of these modules.

Its built-in wireless radio, BCM43438, allows it to connect to connect to WiFi and also act as an access point for other devices.

The power supply of Raspberry Pi 3 uses a +5.1V micro USB. It is recommended to have a 2.5A output power supply when using its full capabilities but 1.2A is enough for basic applications. As peripherals and functionalities are added, Pi would require more power and supplying it with insufficient power will cause some random malfunctions in the Pi or even may cause it to reboot completely (Retrieved on August 19, 2016 from <https://www.raspberrypi.org/help/faqs/>).

**Tenda W311u+**

W311u+ is a 150Mbps wireless adapter with a removable antenna. It is IEEE802.11n/g/b compatible and supports 64/128-bit WEP and WPA/WPA2 security. It also supports software enabled access point (SoftAP) that allows it to act as a wireless access point in a network (Retrieved on August 19, 2016 from/ <http://www.tendacn.com/in/product/W311U+.html>).

**OLSR**

OLSR is mainly used in mobile ad hoc networks. Nearby devices that run OLSR are automatically connected and configured to join a network. It is best suitable for huge, dense networks because of its application of MPR (Clausen, 2003).

Compared to AODV, OLSR has higher overhead since it maintain a relevant routing table for the whole network. In AODV, when nodes continually send messages, the nodes keep on looking for the best path and it floods the network. In OLSR, because of its property of mapping the whole network, as the network gets bigger, the overhead gets significantly bigger. This however is compensated by the MPR application of OLSR (Vinas, 2012).

To select its MPR, node A will broadcast a packet to know its 1st and 2nd hop neighbors. All 1st hop neighbors that has node A’s 2nd hop neighbor that is exclusively connected to it, is automatically selected as MPR. The rest of the 1st hop nodes will then try to cover the rest of the 2nd hop nodes, the 1st hop nodes that has the most efficient connections to the 2nd hop nodes will be added to the MPR list.

**HSMM-Pi**

HSMM-Pi is an open source tool that enables Raspberry Pi and BeagleBone to connect to other devices such as routers and radios with the mesh network provided by HSMM. It is tested in Raspbian OS in Raspberry Pi, Debian in BeagleBone, and Ubuntu 12.04 in BeagleBoneBlack. For the hardware, it needs a WiFi adapter and an SD card to run. An HSMM-Pi node can be set to Mesh Gateway Mode that routes internet traffic to and from an Ethernet port or as an Internal Mode that routes traffic across the mesh network (Smith, 2016). It is designed to connect wired and wireless networks with or without internet access.

It includes a PHP web application that shows the nearby nodes and its link quality.This web application also enables users to monitor and change the settings using a GUI. DHCP service could also be turned on and the start and end host address can be set. When a user changes a setting, the Raspberry Pi has to reboot and HSMM will run a script on boot to reconfigure the Pi’s settings.

HSMM offers a good starting point for OLSR implementations. It only needs to be installed and all of its functionalities are automatically set and all are available via a GUI. It does, however have some issues when working with other software tools as it sometimes overwrites their settings making them unusable. HSMM also modifies some manual configurations. Files such as the rc.local that runs on boot has a note that it is automatically managed by HSMM and that manual configurations will be overwritten. This pose some flexibility issues when working with HSMM.

**hostapd**

hostapd is the daemon used by IEEE to setup access points and authentication servers. It is currently supported in Linux and FreeBSD (Malinen, 2012). It is a software tool that enables network devices to function as an AP. hostapd can create multiple AP using a single card and also convert multiple network cards in a device to function as AP with just a single hostapd instance. However, it does not give DHCP services and does not handle routing.

**Walkietooth**

Walkietooth is a mobile application created by Massimo Milazzo and is available for download in Google Play. It allows two way communication between two Android devices connected in the same network and does not need internet connection. It has two main functions, voice call and share camera view. The share camera view includes a chat option and voice functions. It has three available options for connectivity, WiFi, Bluetooth, and WiFi Direct. In WiFi and Bluetooth connections, a user can be a client or a server. A device running on server mode will just display its IP address while waiting for a client to manually connect to it using the app’s interface. The WiFi Direct option will show all available devices nearby and can be used without knowing the IP address of each other (Retrieved in August 19, 2016 from <https://play.google.com/store/apps/details?id=it.masmil.walkietooth>).

**III. Technical Background**

Currently, there are a lot of available hardware and software that can be used to achieve the goals of this study. However, to be able to find the ideal protocols and software tools, it is necessary for the hardware used to be flexible when it comes to software compatibility. There are portable routers available online that are capable of ad hoc mesh networking which can be used as nodes for the study. These routers however, have limited compatibility when it comes to software so it might be necessary to purchase different kinds of them in order to test software. To save costs, the study will focus on using ARM based devices since they can achieve the same functionalities of a node and more.

**Raspberry Pi 3 Model B**

The latest model of Raspberry is the Raspberry Pi 3 Model B which was released on February 2016. It has 1.2GHz 64-bit quad-core ARMv8 CPU, a built-in 802.11n Wireless LAN and Bluetooth 4.1 (Retrieved on August 16, 2016/ https://www.raspberrypi.org/products/raspberry-pi-3-model-b/). It uses Broadcom System on Chip (SoC) BCM2837. SoC is an integrated circuit that combines all computer components such as CPU, graphics processor, memory, and other peripherals into a single chip(Anthony, 2012). BCM2837 has the same basic circuit design as the SoC of both Raspberry Pi 1and 2 so it is backward compatible. The hardware update made Raspberry Pi 3 perform 50-60% better than Pi 2. Previous models of Raspberry Pis do not have a built-in Wireless LAN and Bluetooth making Pi 3 cheaper than Pi 2 because of the additional costs of these modules.

Its built-in wireless radio, BCM43438, allows it to connect to connect to WiFi and also act as an access point for other devices.

The power supply of Raspberry Pi 3 uses a +5.1V micro USB. It is recommended to have a 2.5A output power supply when using its full capabilities but 1.2A is enough for basic applications. As peripherals and functionalities are added, Pi would require more power and supplying it with insufficient power will cause some random malfunctions in the Pi or even may cause it to reboot completely (Retrieved on August 19, 2016 from https://www.raspberrypi.org/help/faqs/).

**Tenda W311u+**

W311u+ is a 150Mbps wireless adapter with a removable antenna. It is IEEE802.11n/g/b compatible and supports 64/128-bit WEP and WPA/WPA2 security. It also supports software enabled access point (SoftAP) that allows it to act as a wireless access point in a network (Retrieved on August 19, 2016 from/ http://www.tendacn.com/in/product/W311U+.html).

**OLSR**  
OLSR is mainly used in mobile ad hoc networks. Nearby devices that run OLSR are automatically connected and configured to join a network. It is best suitable for huge, dense networks because of its application of MPR (Clausen, 2003).

Compared to AODV, OLSR has higher overhead since it maintain a relevant routing table for the whole network. In AODV, when nodes continually send messages, the nodes keep on looking for the best path and it floods the network. In OLSR, because of its property of mapping the whole network, as the network gets bigger, the overhead gets significantly bigger. This however is compensated by the MPR application of OLSR (Vinas, 2012).

To select its MPR, node A will broadcast a packet to know its 1st and 2nd hop neighbors. All 1st hop neighbors that has node A’s 2nd hop neighbor that is exclusively connected to it, is automatically selected as MPR. The rest of the 1st hop nodes will then try to cover the rest of the 2nd hop nodes, the 1st hop nodes that has the most efficient connections to the 2nd hop nodes will be added to the MPR list.

**HSMM-Pi**

HSMM-Pi is an open source tool that enables Raspberry Pi and BeagleBone to connect to other devices such as routers and radios with the mesh network provided by HSMM. It is tested in Raspbian OS in Raspberry Pi, Debian in BeagleBone, and Ubuntu 12.04 in BeagleBoneBlack. For the hardware, it needs a Wi-Fi adapter and an SD card to run. An HSMM-Pi node can be set to Mesh Gateway Mode that routes internet traffic to and from an Ethernet port or as an Internal Mode that routes traffic across the mesh network (Smith, 2016). It is designed to connect wired and wireless networks with or without internet access.

It includes a PHP web application that shows the nearby nodes and its link quality. This web application also enables users to monitor and change the settings using a GUI. DHCP service could also be turned on and the start and end host address can be set. When a user changes a setting, the Raspberry Pi has to reboot and HSMM will run a script on boot to reconfigure the Pi’s settings.

HSMM offers a good starting point for OLSR implementations. It only needs to be installed and all of its functionalities are automatically set and all are available via a GUI. It does, however have some issues when working with other software tools as it sometimes overwrites their settings making them unusable. HSMM also modifies some manual configurations. Files such as the rc.local that runs on boot has a note that it is automatically managed by HSMM and that manual configurations will be overwritten. This pose some flexibility issues when working with HSMM.

**hostapd**

hostapd is the daemon used by IEEE to setup access points and authentication servers. It is currently supported in Linux and FreeBSD (Malinen, 2012). It is a software tool that enables network devices to function as an AP. hostapd can create multiple AP using a single card and also convert multiple network cards in a device to function as AP with just a single hostapd instance. However, it does not give DHCP services and does not handle routing.

**Walkietooth**

Walkietooth is a mobile application created by Massimo Milazzo and is available for download in Google Play. It allows two way communication between two Android devices connected in the same network and does not need internet connection. It has two main functions, voice call and share camera view. The share camera view includes a chat option and voice functions. It has three available options for connectivity, WiFi, Bluetooth, and WiFi Direct. In WiFi and Bluetooth connections, a user can be a client or a server. A device running on server mode will just display its IP address while waiting for a client to manually connect to it using the app’s interface. The WiFi Direct option will show all available devices nearby and can be used without knowing the IP address of each other (Retrieved in August 19, 2016 from https://play.google.com/store/apps/details?id=it.masmil.walkietooth).

**IV. Design and Methodology**

**Sourcing**

Sourcing of the selected components came to be a challenge because not all ideal materials were readily available locally in electronic or computer stores. There are online advertisements in Philippine stores but offered are mostly the older and more expensive versions of the Pi that do not have a built in Wi-Fi adapter attached.

The 2 Raspberry Pi 3 model B's (See Figure 2 in Appendices) were purchased through Amazon for $35 each on 7/24/2016 because there are no available Pi 3's at Manila. It has been also identified that the other components needed to sourced elsewhere as even the 5V 2.5A power supply required by the Pi is not readily available here. The 2 power supplies cost $9.99 each for a total of $19.98. The 16Gb micro SD memory card were easily bought in a local store for 400php each and the last component, the Tenda W311U+ Wi-Fi dongle were bought for 800php each. The total cost in php was 6719php(with the current exchange rate of 48php per dollar inclusive of Amazon's 3% peso to dollar conversion fee)

Discussed are the only necessary hardware to create the network nodes.(See Figure 3 in Appendices) A portable power source is needed to be able to deploy the configured mesh nodes but for configuration purposes wall outlets are used.

**Initial Setup**

First, the purchased micro SD card is formatted to FAT32 so that the flash memory can be readable and usable by the OS. The image of Raspbian Jessie is downloaded at https://www.raspberrypi.org/. Before operation of the Raspberry Pi, the image of the current Raspbian installer is loaded onto the micro SD memory card by just copy and pasting the extracted files into the directory of the card. The card is then inserted into the Raspberry Pi 3's memory card slot. The Wi-Fi dongle is inserted to the USB slot for device detection. (See Figure 4 in Appendices)

Internet connectivity to the Pi was added by connecting an Ethernet cable to the on board Ethernet adapter. This is done to be able to download the initial updates of the operating system and the packages needed to set up the node. The input and output devices: mouse, keyboard, and monitor are then connected before plugging the Pi to an outlet. When plugged in, the Pi's automatically turn on and display a selection of operating systems to be installed.

**Operating System Installation and Initial Setup**

Raspbian is selected from the choices and then continued to the installation. The installation sequence is straightforward as the wizard clearly explained each step during the whole process. Upon completion of the installation, a terminal is opened and the OS is updated.

**HSMM-Pi Installation and Setup**

Git is installed and the researchers clone the repository of HSMM-Pi to be able to run the install.sh script. Once the installation is finished, HSMM-Pi is configured through a web dashboard hosted locally on the Pi. The necessary settings are configured to enable ad-hoc networking on the Tenda W311U+ Wi-Fi interface. After any changes in the configuration, the raspberry pi is rebooted. More information can be found in the website: https://github.com/urlgrey/hsmm-pi

**Mesh Connectivity Testing (Node-to-Node)**

After HSMM-Pi is cloned, installed and configured, the researcher checks if olsrd is installed and running. The process is killed if it is running to be able to run the daemon in debug mode. Olsrd is run in debug mode where the discovery of nearby nodes is monitored to know if HSMM-Pi has been configured properly. Once the mesh network is verified to be running properly, ping tests are conducted to verify inter node communication.

**DHCP Setup**

On the HSMM console the second network interface card is configured to dispatch ip addresses by simply configuring the IP address of the card and the range of IP addresses to be given by the access points. The researchers ensured that the IP addresses assigned manually and dynamically will not cause conflict in the IP addressing of the future mesh network.

**Wireless Access Point Setup**

The interfaces are configured in the /etc/network/interfaces scripts to be able to create an wireless access point using the built in Wi-Fi adapter. The smartphones are to be connected to these interfaces when the nodes are deployed. All the communication between networks are then routed using the mesh network linked to the Wi-Fi dongle. The configured interfaces are then placed on the configuration of hostapd.

**Network Address Translator Configuration**

Data crossing from the access point interface to the mesh network is configured buy assigning the access point Wi-Fi adapter as the gateway. The firewall rules necessary to establish NAT in all nodes are inserted and configured via the iptables command and tests are conducted. This setup would enable all the devices connected to the access point to see the mesh network nodes. However, even if the access points clients are able to see the mesh network, it is still not able to locate clients in other access points.

**Routing Tables Configuration**

The proper routes are added onto the routing table to enable discovery of the clients to other access points. Because the dongles are connected together in a mesh via olsrd, only the routes from the node to their respective paired gateways are to be configured.

**Automatic Start on Boot**

After all the setup and configuration, non persistent commands must be placed in the rc.local file for the Pi to load it all again on boot. This enables the Pi to run "headless" or without and peripheral devices attached to it. The Pi's that are already configured are now only managed through ssh.

**End Device-to-End Device Tests**

Connection persistence, data integrity, and communication efficiency are tested by connecting smartphones to the access points of different Pi's and using them to ping the device on the opposite network. Applications like 'Walkietooth' are also used to test the usability of the deployed network.

Range is tested on makeshift environments to portray events where other network infrastructure is unavailable. It is also tested in numerous test cases; places that are both open and have multiple obstructions. Places that are both urban and rural to take into account the interference of other radios.

**V. Results and Discussion**

Initial tests were conducted along San Martin Street in Magallanes Village, Makati City. The street is around 5 meters wide, however, being located in an urban village, a lot of cars were parked on both sides of the street.

In the first test, the range of the AP was tested. Two smartphones, Samsung Galaxy S1 and Galaxy S4 were connected to a node. All devices were initially in the same place and its coordinate was plotted using Google Maps app. The initial point is referred to as point A. Both smart phones run Walkietooth and initialized its camera share feature. The Samsung Galaxy S4 then slowly made distance while maintaining line of sight vision with the node and S1. When the connection of the camera share stopped, the coordinates of S4 was plotted using Google Maps and is referred to as point B. While in the point B, S1 and S4 initialized a connection with each other using the Serval Mesh app. S4 again started moving slowly while exchanging “a” messages with S1. When the “a” messages stopped coming, the coordinates of S4 was recorded again and referred to as point C. The distance from point A to point B is 151ft while the distance from point A to point C is 466ft.

The second test was the same as the first but used two different nodes. This test is conducted to know the maximum range of the mesh connection of the nodes. S1 was connected to Node1 and S4 was connected to Node2. All devices were initially at point A. S4 started moving slowly with Node2, while exchanging “a” messages with S1 that remained in point A with Node1. Once the messages stopped coming, the coordinates was recorded as point D. The distance from point A to point D is 476ft.

**VII. Appendices**

**I. References**

Bourque B. (2015). Arduino vs. Raspberry Pi: Mortal Enemies, or Best Friends? Retrieved on March 10, 2016/ http://www.digitaltrends.com/computing/arduino-vs-raspberry-pi/

Schmidt A., Häkkilä J., Rukzio E., Atterer R. (2006). Mobile Phones as Tool to Increase Communication and Location Awareness of Users. Retrieved on August 10, 2016/ https://www.medien.ifi.lmu.de/pubdb/publications/pub/schmidt2006ieemobility/schmidt2006ieemobility.pdf

Krag T., Büettrich S. (2004). Wireless Mesh Networking. Retrieved on March 10, 2016/ http://archive.oreilly.com/pub/a/wireless/2004/01/22/wirelessmesh.html

Ciarlone J. (2015). Wireless Mesh Networking: Recommended Maximum Distances. Retrieved on March 20, 2016/ http://info.hummingbirdnetworks.com/blog/wireless-mesh-networking-recommended-maximum-distances

Ishfaq M. (2014). Comparative Analysis of Ad Hoc Routing Protocols in Wireless Mesh Network Environment. Retrieved on March 20, 2016/ http://www.idosi.org/wasj/wasj29(3)14/14.pdf

GSMA Intelligence (2014).Country Overview: Philippines Growth through innovation. Retrieved on March 20, 2016/ https://www.gsmaintelligence.com/research/?file=141201-philippines.pdf&download

Project Byzantium (2011). Retrieved on July 23, 2016/ http://project-byzantium.org/

Woodford C. (2016). How Cellphones Work. Retrieved on July 10, 2016/ http://www.explainthatstuff.com/cellphones.html

Mullen J. (2013). Super Typhoon Haiyan, One of Strongest Storms Ever, Hits Central Philippines. Retrieved on March 9, 2016/ <http://edition.cnn.com/2013/11/07/world/asia/philippines-typhoon-haiyan/>

Roberts E. (December 2006). RISC Architecture. Retrieved August 19, 2016/ <https://cs.stanford.edu/people/eroberts/course>s/soco/projects/risc/whatis/index.html

Kozierok C., 2005, The TCP/IP Guide, <http://www.tcpipguide.com/free/t_OverviewOfKeyRoutingProtocolConceptsArchitecturesP.htm>. Retrieved August 16, 2016.

Perkins & Royer, 2003, <https://www.cs.cornell.edu/people/egs/615/aodv.pdf>, Retrieved August 16, 2016.

Cisco Networking Academy, 2014 <http://www.ciscopress.com/articles/article.asp?p=2180210&seqNum=4>, Retrieved August 16, 2016.

Thomas J., 2008 <http://www.omnisecu.com/cisco-certified-network-associate-ccna/introduction-to-link-state-routing-protocols.php>, Retrieved August 17, 2016.

Alberghetti, D, 2015 <http://www.danscourses.com/CCNA-2/link-state-routing-protocols.html>, Retrieved August 17, 2016.

Clausen T, Jacquet P., 2003 <https://www.ietf.org/rfc/rfc3626.txt>, Retrieved August 17, 2016

Chroboczek J, April 2011, <https://tools.ietf.org/html/rfc6126>, Retrieved August 17,2016.

Vinas R. B., Terren D. R., Neumann A., Lopez E., September 12, 2012, Evaluation of Dynamic Routing Protocols on Realistic Wireless Topologies, <http://www.bmx6.net/attachments/129/master_thesis__evaluation.pdf>, Retrieved August 17, 2016.

Kinter J., January 2010, <http://www.broadband-hamnet.org/>. Retrieved August 17, 2016.

Smith C., May 2016, HSMM-Pi, <https://github.com/urlgrey/hsmm-pi>, Retrieved August 18, 2016.

Open Technology Institute, <https://commotionwireless.net/contact/>, Retrieved August 18, 2016

Dye M, McDonald R, Rufi A, 2008, Network Fundamentals, CCNA Exploration Companion Guide.

Debian, https://www.debian.org/ , Retrieved August 18, 2016

Raspbian, <https://www.raspbian.org>. Retrieved August 18, 2016

Arch Linux, https://wiki.archlinux.org/index.php/ , Retrieved August 18, 2016

RPi Distributions, http://elinux.org/RPi\_Distributions , Retrieved August 18, 2016

Arduino (n.d.). Retrieved August 18, 2016/ <https://www.arduino.cc/>

Raspberry (April 2012). Retrieved August 18, 2016/ <http://elinux.org/RPi_Philosophy>

Raspberry Pi 3 Model B (n.d.). Retrieved August 18, 2016/https://www.raspberrypi.org/products/raspberry-pi-3-model-b/

Raspberry Pi (n.d.). Retrieved August 19, 2016/https://www.raspberrypi.org/help/faqs/

(September 2015). BeagleBoneVs Raspberry Pi 2: Choosing The Right Board, Retrieved August 19, 2016/ https://pimylifeup.com/beaglebone-vs-raspberry-pi/

Anthony S. (April 2012). SoCvs CPU – The Battle For The Future Of Computing/http://www.extremetech.com/computing/126235-soc-vs-cpu-the-battle-for-the-future-of-computing

Tenda (n.d.). Retrieved August 19, 2016/http://www.tendacn.com/in/product/W311U+.html

Malinen J. January 2012, hostapd. Retrieved August 19, 2016 from <https://w1.fi/hostapd/>

**Additional Sources:**

<http://www.digitaltrends.com/computing/arduino-vs-raspberry-pi/><https://learn.sparkfun.com/tutorials/what-is-an-arduino><https://www.raspberrypi.org/help/faqs/#introWhatIs><http://www.wi-fi.org/discover-wi-fi/wi-fi-direct><http://stackoverflow.com/questions/29018122/android-pure-p2p-chat-application><http://info.hummingbirdnetworks.com/blog/wireless-mesh-networking-recommended-maximum-distances><http://archive.oreilly.com/pub/a/wireless/2004/01/22/wirelessmesh.html><http://www.idosi.org/wasj/wasj29(3)14/14.pdf><http://sts.bwk.tue.nl/7y500/readers/InstellingenRegelaars.pdf>[https://www.techopedia.com/definition/575/at-command-set](https://www.techopedia.com/definition/575/at-command-set )<http://makezine.com/2015/04/01/esp8266-5-microcontroller-wi-fi-now-arduino-compatible/><https://gsmaintelligence.com/research/?file=141201-philippines.pdf&download><http://www.mgb.gov.ph/images/stories/RA_7942.pdf><http://pluto.ksi.edu/~cyh/cis370/ebook/ch05b.htm>

<http://project-byzantium.org/>

<https://www.medien.ifi.lmu.de/pubdb/publications/pub/schmidt2006ieemobility/schmidt2006ieemobility.pdf>

<http://www.explainthatstuff.com/cellphones.html>

<http://edition.cnn.com/2013/11/07/world/asia/philippines-typhoon-haiyan/>

**II Figures**

Figure 1. Layout of Node Deployment

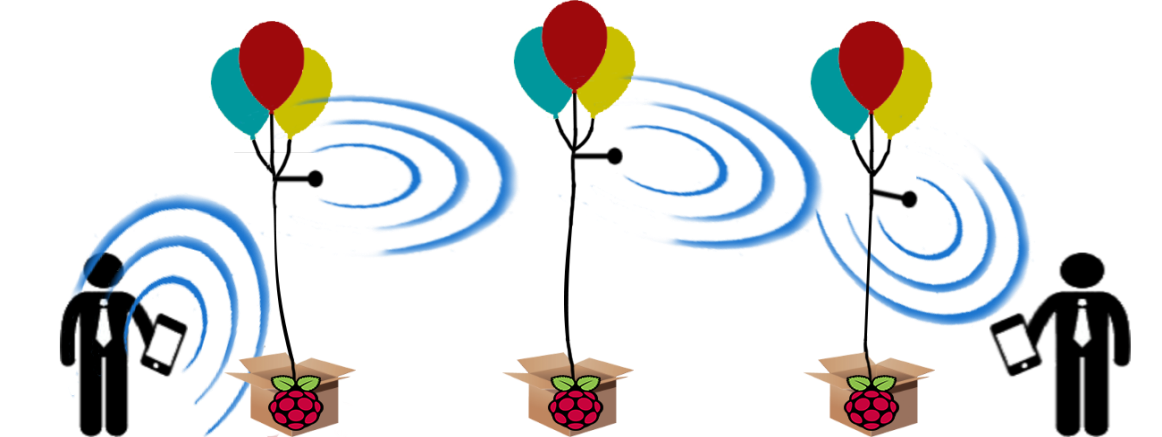
****

Figure 2. Raspberry Pi 3 Model B (With and without case)

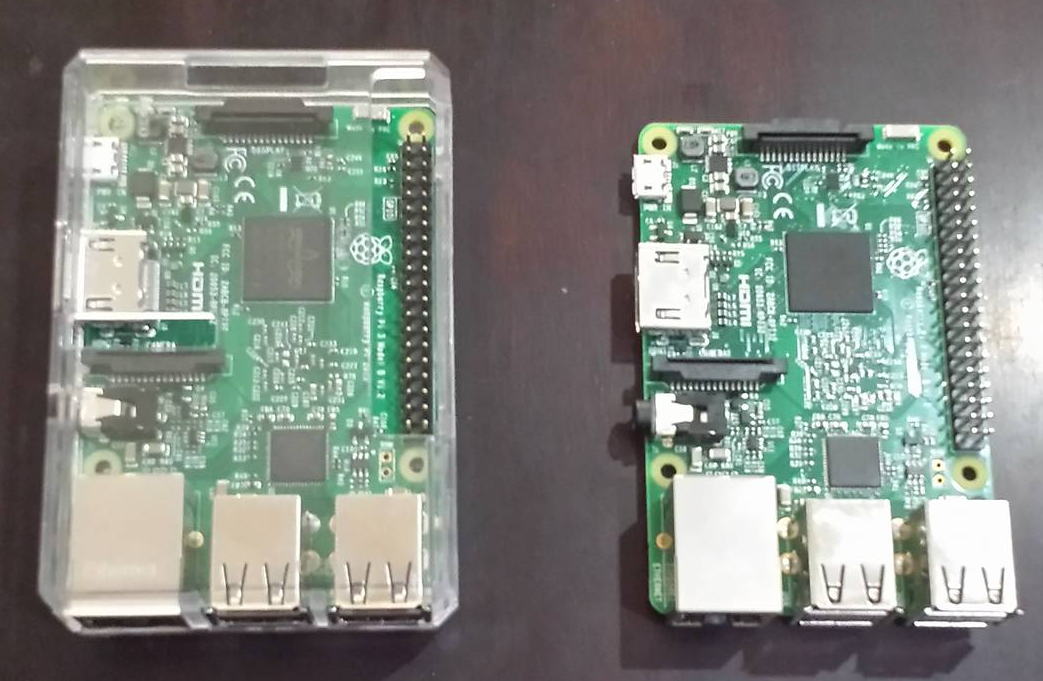


Figure 3. Components for the Node (Tenda W311u+ USB WiFi, Raspberry Pi 3, Mini SD cards)



Figure 4. Assembled Node with Battery Pack



Figure 5. Use Case Diagram

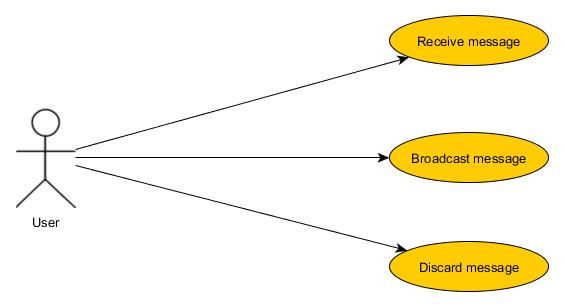
****

Figure 6. Communication Diagram



Figure 7. Range of Camera Share (Device-Node-Device)

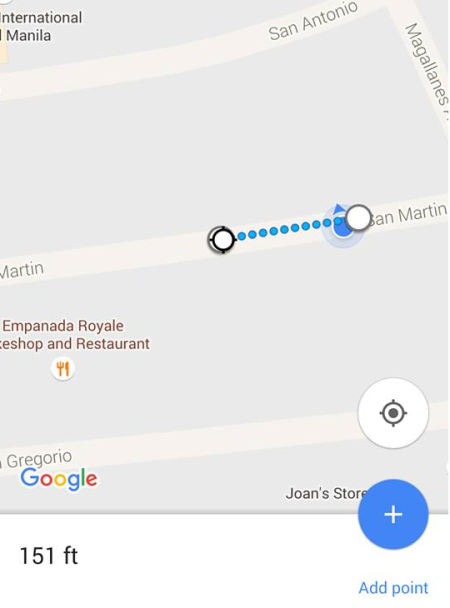


Figure 8. Range of Voice Call (Device-Node-Device)

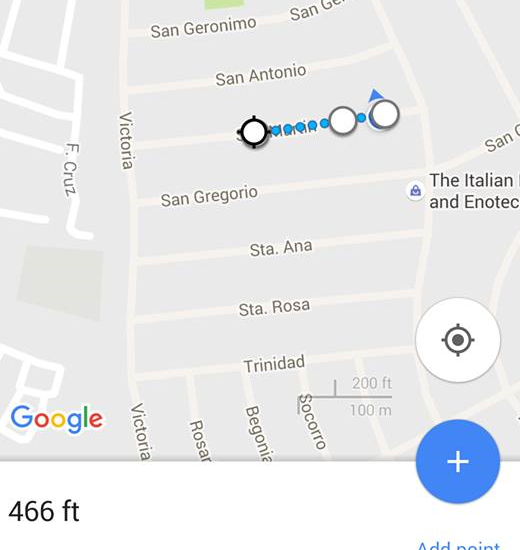


Figure 9. Range of Voice Call (Device-Node-Node-Device)

