**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**   
   
**2.1 Wi-Fi Peer-to-Peer**   
Mobile chat applications let users communicate with each other from 2 end devices. 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. This research will need a mobile chat application that is not provisioned with a web server. It should only rely on a peer-to-peer networking for the sending of messages.

One way of doing this is through the use of Wi-Fi Direct™ connection which enables 802.11/ Wi-Fi capable devices 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 to other devices.

The user-end devices in this research that will act as endpoints of the mesh will be the smartphones which the users will use to type and send their messages. The messages will then be passed through the microcontrollers to a destination device.

The source states however that Wi-Fi Direct makes devices act as access points so a simultaneous two way connection is not possible. This is an area to be researched further as the paper progresses.   
 

**2.2 Mesh and Star Network topologies**  
A wireless mesh network is a network setup wherein devices (or nodes), are interconnected with each other. 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 somessages 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.   
   
This research will be using a partial mesh topology in combination of a star 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 commonly use 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.  
 http://www.webopedia.com/quick\_ref/topologies.asp

**2.3 Reduced Instruction Set Computing (RISC)**

A processor design that emphasizes the use of software. It makes processors “…use simple instructions that can be executed within one clock cycle.” (Roberts, 2006)

**2.4 Advanced RISC Machines (ARM) Boards**

Boards that uses 32-bit microprocessors. They were first used for simple task but are currently growing to meet the demands of current technology. Currently, ARM boards are used in embedded systems and in internet of things.

**2.5 ARM Based Devices (Arduino, Raspberry Pi, BeagleBone)**

Arduino and Raspberry Pi are the popularARM based devices that are often compared to one another about its functions and capabilities. Many people say that Raspberry is ultimately better than Arduino since it can function as a normal computer. Yet there are also many that say such assumptions are “unreasonable" because of what Arduino can do.

Raspberry Pi could function similarly to a computer because it has some features that a computer has. It has its own memory, graphics driver, processor, etc. Also, it has its own Ethernet port so networking is more convenient. It is capable of doing works like doing spreadsheet, word, internet browsing, and gaming. It also has its own operating system (Linux). But to fully utilize sensors integrated with it, different software is sometimes required. Same as normal computers, it needs to be turned off properly. Arduino, on the other hand, does not act and function like a computer. It does not have all necessary parts of a normal computer and it does not operate a full operating system. It just runs the codes in it. For networking purposes, it does not have an Ethernet port so additional external hardware for physical connection applications is required. For integration of sensors, accomplishing it is easier in Arduino because the “interpretation” & “response” can be done effortlessly with the use of simple codes. It can work and execute its codes by connecting it to a USB cable to a computer (and any other ways). (Bourque, 2015)

BeagleBone on the other hand is similar to Raspberry Pi with some Arduino properties. Its software and hardware capabilities are comparable to Raspberry Pi since it also has its own operating system and functions like a computer. Its hardware specifications are generally lower than Raspberry Pi but it contains more GPIO pins (Raspberry Pi has 40, BeagleBone has 92). This makes BeagleBone is better for circuitry projects because the number of its GPIO pins while Raspberry Pi better for software based projects such as this study because of its superior hardware specs (Retrieved on August 19, 2016 from https://pimylifeup.com/beaglebone-vs-raspberry-pi/).

**2.6 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 very powerful tool to experiment on small programs and sensors but would not be best used for the purpose of this research. For one, Wifi modules are not integrated into the circuit that comes out of the box so one must purchase a modules that can be attached called shields. The add-ons also do not guarantee full support because the microcontroller itself has many limitations.

The device and its modules are currently being used for small projects concerning the internet of things (IoT) connecting multiple home devices and sensors .Arduino is very 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 this device as a routing tool. It has been discussed that Arduino has too little RAM and computing power to handle becoming an node for routing in a high activity mesh network. Most of the problems that arise in the form of code crashes are caused by this limited memory. (Retrieved in August 18, 2016 from http://forum.arduino.cc/index.php?topic=166151.0)

**2.7 ESP8266**    
ESP8266 is a microcontroller that can access 802.11 connections. It is Arduino compatible, meaning it can run programs that are created for Arduino and could also be used as an integrated module. It was initially introduced as a Serial-to-WiFi adaptor that supports AT command set. AT command set, also known as Hayes command set, is a language that is made up of multiple short strings that is used for simple operations such as hanging up and dialing in modems. In this research, this microcontroller will be used to give network access to Arduino so that they can communicate with each other. 

**2.8 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).

**2.9 Debian**

Created by Debian Project, people who aim to develop a free operating system. It currently uses the Linux or FreeBSD kernel. It is an open source Linux based operating system that contains over 43000 free packages that enable users different functionalities and a level of flexibility to their devices (Retrieved in August 18, 2016 from https://www.debian.org/). Because of its simplicity, functionalities, and flexibility, a lot of Raspberry Pi operating systems are based on Debian. An example for this would be the initial recommended OS in the Raspberry Pi website, the Raspbian.

**2.10 Routing protocols**

Routing protocols dictates where and how the router distributes packets. There are a lot of different routing protocols currently available and each of them has its own design and serves different 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 to consider that table based protocols grows bigger as nodes increase and packet header grows bigger as more nodes are included when choosing the proper routing protocol for your network.

**Classifications of routing protocols**

**Dynamic Routing Protocols –** According to Cisco Networking Academy (2014), this “…allow routers to dynamically share information about remote networks and automatically add this information to their own routing tables”. Because of this sharing of information, the network automatically adapt with its topology. This also allows routers to discover new networks and repair broken ones.

**a.** External Gateway Protocols (EGP) **–** Used for routing systems that are handled by different organizations.

**b.** Interior Gateway Protocols (IGP) **–** Used for routing routers that are handled by a single organization.

**b.i.** Distance Vector Routing Protocols –Routers that use this routing protocol are not aware of the entire map of its network. The router only knows the distance and the vector to a device connected to the network. Vector is the direction of the next hop and distance is how many hop counts or bandwidth or some other metrics, it will take to reach a destination (Cisco Networking Academy, 2014). Most routing protocols that are under this classification enable its routers to send periodic updates to all devices in the network to maintain a relevant distance and vector values (Thomas, 2008).

**b.i.i**Ad-hoc On-Demand Distance Vector Routing (AODV)**-** 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).

**b.i.ii.** Babel – a Distance Vector routing protocol that uses periodic updates. To reduce the overhead caused by these updates, it “… allows to omit subnet prefixes when multiple addresses are sent in a single packet…” (Vinas et al., 2012). Babel is used in both wired networks and wireless mesh networks. It has a “…loop avoiding distance-vector routing protocol that is designed to be robust and efficient…” This loop avoiding property makes it ideal for networks that have unstable connections. However, this also causes performance degradation for huge environments with stable network. Babel can be configured to route networks using the shortest path or other metrics. It also uses a hybrid routing protocol that can be used by IPV4 or IPV6 networks (Chroboczek, 2011).

**b.ii.**Link State Routing Protocols – Routers that use this routing protocol is aware and “has a complete view” of all the connections in its network (Cisco Networking Academy, 2014). It first completes a map of the entire network then calculates the best path to a destination using an algorithm. This process allows updates to converge faster at the expense of additional overhead due to the flooding it causes(Alberghetti, 2015).

**b.ii.i.** Optimized Link State Routing Protocol (OLSR) –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.

**2.11 Available networking projects**

**Broadband-HamnetTM -** “…a high speed, self discovering, self configuring, fault tolerant, wireless computer network…” (Kinter, 2010)They are formerly called HSMM-MeshTM (High-Speed Multimedia), and their 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 - A**n 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**- 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.

**2.12 Raspberry Pi Operating Systems**

**Raspbian –** closely based on Debian that is optimized for Raspberry Pi. It is completed in 2012 but is still in active development. Raspbian was created by developers who are fans of Raspberry Pi and the Debian Project. A lot of documentations of Debian are applicable in Raspbian since the only differences that the two have are caused only by the hardware differences of a computer and a Raspberry Pi (Retrieved in August 18, 2016 from https://www.raspbian.org).

**Arch Linux –** an independently developed Linux distribution that prides itself for its lightweight characteristic. Things are kept simple and bare with users doing most of the job themselves. It does not even have a default Graphical User Interface after installation (Retrieved in August 18, 2016 from http://elinux.org/RPi\_Distributions). This can give a certain level of difficulty to beginners. It first came out as a computer operating system but is later modified to run in ARM-based devices (Retrieved in August 18, 2016 from https://wiki.archlinux.org/index.php/).

**2.13 OSI Model**

Layered models are usually used in network communications to illustrate the processes involved when data is transmitted. The Open Systems Interconnect (OSI) model is the most known layered model and it acts as an abstract representation of how network enabled devices communicate with each other. It was developed by International Organization for Standardization (ISO) to serve as an outline of protocol stack used by non-proprietary protocols.

The OSI model can be visualized as a diagram with seven layers with each layer receiving message from the one directly above or below it depending on whether the device is receiving or transmitting data. These layers are usually divided into two groups, the upper layers and the lower layers. The upper layers consist of Layers 5, 6, and 7 and their focus is on user interaction and identifying the message so that it will be delivered to the right receiver. These layers are handled in the software side. The lower layers, consisting of Layers 1, 2, 3, and 4, focus on the transmission of the message and are handled by both hardware and software.

Layer 7 or the Application layer handles the services which let the end user interact with user applications. It lets the message sender interact with the software to get the desired message into the system. Alternatively, it also lets receiver acquire the message transmitted to him.

The next layer, Layer 6, is the Presentation layer. It is responsible for providing the application the format of a data. Different systems use different data format and protocols. Presentation layer acts as a translator to translate incoming messages to a format that the application uses.

Layer 5 or the Session layer manages sessions between devices. It maintains the connection of devices allowing them to start sessions and restart them when the connection is interrupted.

Transport layer, Layer 4, handles the procedure called segmentation. When data is sent to the Transport layer, it is divided into different segments so that they would be transmitted easier. It keeps track of each segment to ensure that all of them are transmitted. Because of the nature of data transmission, segments do not arrive at the right order so the Transport layer reassembles these segments by numbering them. This layer enables the device to have multiple concurrent connections since it divides the data into smaller pieces instead of a one whole stream.

Network layer, Layer 3, address packets with a source IP address and destination IP address that are both stored in an IP header. The process of adding these information is called encapsulation. After encapsulation, a segment is then referred to as a packet which is ready to be transmitted over a network. As multiple routes are usually available for a packet to reach its destination, Network layer handles the routing by calculating the best path.

Data Link layer, layer 2, enables the layers that deal with the software processes, layers 3 to 7, communicate with the layer that deals with the hardware, layer 1. Layer 2 encapsulates the packets into frames to format the message into pieces than can be interpreted by different media. Another role of the data link layer is to control how frames are sent to or received by the media. This is called Media Access Control(MAC). Since the data will pass through different media as it travels from its source to its destination, MAC allows enable devices to handle and read the data.

Physical layer, layer 1, is the only layer that is solely implemented by hardware devices. Its primary role is to encode frames from the data link layer into bits and its transmission as a signal over a physical media such as copper wires and fiber cables. These physical media allows the transfer of signals across different devices as an electrical pulse, light pulse, or radio wave (Dye et al., 2008).

**2.14 Serval Mesh**

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 WiFi 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.

**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 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).

**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.

**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

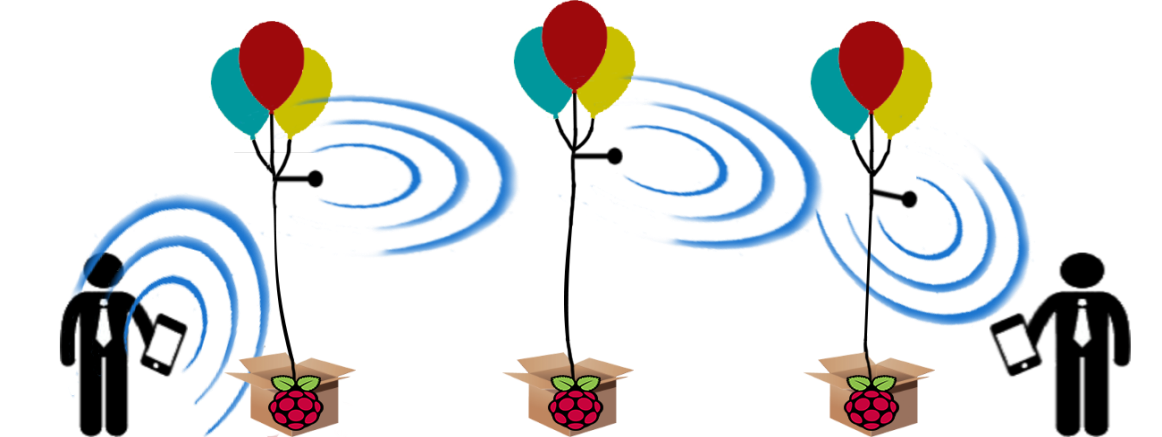
****

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

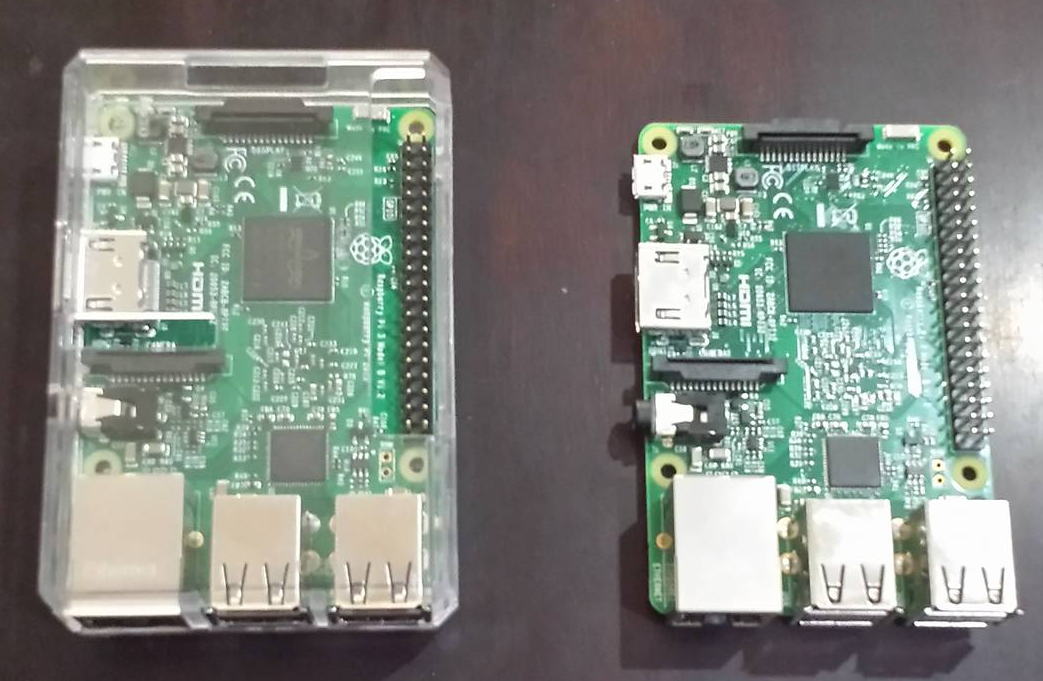


Figure 3. Components for the Node



Figure 4. Assembled Node with Battery Pack



Figure 5. Use Case Diagram

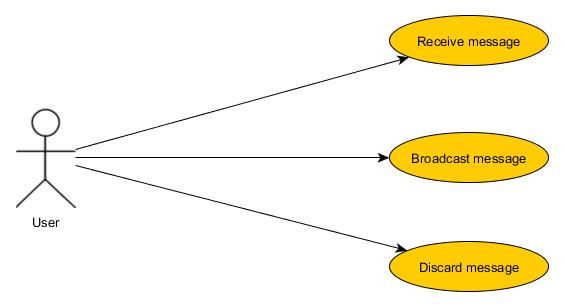
****

Figure 6. Communication Diagram

