**Asia Pacific College**

#3 Humabon Place, Magallanes

Makati City



Lightweight Easily-deployable Inexpensive Nodes for Temporary Wireless Mesh Networks w/ Distributed Databases

In partial fulfillment of the requirements

In CSSPROJ

A.Y. 2016 - 2017

**Team Members:**

|  |  |
| --- | --- |
| **Name** | **Role** |
| Rivera, Benjamin | Project Manager / Network Engineer / Quality Control |
| Tonoike, Tomio | Software Developer / GIS Designer / Programmer |
| Tison, Alfonsin | Documentation / Quality Assurance / Tester |
| Ellana, Allen Rodri | Quality Assurance / Network Engineer |
| Cariño, Patrick George | Database Designer / Software Developer / Tester |

**Project Advisor:**

Mr. Paulo Oblepias

**Professor:**

Mr. Manuel Sebastian

# Abstract

A proper communications system is vital when natural calamities strike. Communication may greatly improve the coordination of efforts among the people in-charge of disaster recovery and relief. In some situations however, disasters damage the infrastructures of communication systems people rely on, hence a need for a rapid deployment type of a communications network arises. These networks should maintain real-time data transfer and scalability as each event inevitably creates numerous time-sensitive activities and each event creates different situations with different environments and different size of land area. This research provides a framework for a scalable mesh network. The paper focuses on describing the outcomes of both the Post-Disaster Emergency Communication System built and the E-Nodes project. It shows the development of network and system architecture and the implementation of a master-less distributed database to maintain reliability, accessibility, and scalability. The results show how the network is a viable tool when faced with events where natural calamities destroy the persistent network infrastructure and how distributed databases are able to fulfill most requirements of an emergency communication system.

# 

Contents

[Abstract 2](#_Abstract)

[I. Introduction 6](#_I._Introduction)

[Background of the Problem 6](#_Toc469614341)

[Statement of the Problem 7](#_Toc469614342)

[Objectives 7](#_Toc469614343)

[General Objectives 7](#_Toc469614344)

[Specific Objectives 7](#_Toc469614345)

[Significance 8](#_Toc469614346)

[II. Scope and Limitations 9](#_Toc469614347)

[III. Review of Related Literature 10](#_Toc469614348)

[Inspirations 10](#_Toc469614349)

[Wireless Mesh Networks 12](#_Toc469614350)

[Distributed Databases 12](#_Toc469614351)

[Related Projects 13](#_Toc469614352)

[IV. Theoretical Background 15](#_Toc469614353)

[Wireless Mesh Network 15](#_Toc469614354)

[Distributed Databases 15](#_Toc469614355)

[Routing Protocols 16](#_Toc469614356)

[V. Design and Methodology 18](#_Toc469614357)

[Components 18](#_Toc469614358)

[Hardware 18](#_Toc469614359)

[Software 19](#_Toc469614360)

[Steps 20](#_Toc469614361)

[Sourcing of Hardware 20](#_Toc469614362)

[Initial Setup / Assembly / Initial boot 20](#_Toc469614363)

[Operating System Installation and Initial Setup 21](#_Toc469614364)

[Olsrd Installation and Setup 21](#_Toc469614365)

[Mesh Connectivity Testing (Node-to-Node) 21](#_Toc469614366)

[DHCP Setup 21](#_Toc469614367)

[Wireless Access Point Setup 21](#_Toc469614368)

[Network Address Translator Configuration 22](#_Toc469614369)

[Routing Tables Configuration 22](#_Toc469614370)

[Automatic Start on Boot 22](#_Toc469614371)

[End Device-to-End Device Tests 22](#_Toc469614372)

[Installation and configuration of Datastax Enterprise/Apache Cassandra 22](#_Toc469614373)

[Development of the Chat Application 22](#_Toc469614374)

[Development of the API 23](#_Toc469614375)

[Installation of QGIS 23](#_Toc469614376)

[Sourcing and Designing the Offline Map 23](#_Toc469614377)

[Customization of the Map Interface and Development of Plotting Functions 23](#_Toc469614378)

[Unit Testing 23](#_Toc469614379)

[Integration of Applications to Apache Cassandra 23](#_Toc469614380)

[System Testing / Usability Testing 23](#_Toc469614381)

[**VI. Results and Discussion** 25](#_Toc469614382)

[E-Nodes 25](#_Toc469614383)

[PDECS 26](#_Toc469614384)

[**VIII. Recommendations** 28](#_Toc469614385)

[Appendices 29](#_Toc469614386)

[Data Flow Diagram (E-Nodes) 29](#_Toc469614387)

[Diagram 0 29](#_Toc469614388)

[Diagram 1 29](#_Toc469614389)

[Diagram 2 29](#_Toc469614390)

[Diagram 3 30](#_Toc469614391)

[Process Specifications 30](#_Toc469614392)

[Diagram 1 30](#_Toc469614393)

[Diagram 2 31](#_Toc469614394)

[Diagram 3 31](#_Toc469614395)

[Network Architecture 31](#_Toc469614396)

[Use Case Diagram 32](#_Toc469614397)

[Context Level Diagram (PDECS) 33](#_Toc469614398)

[Data Flow Diagram 33](#_Toc469614399)

[Figures 34](#_Toc469614400)

[Tables 36](#_Toc469614401)

[References 40](#_Toc469614402)

[Project Status Report Template 43](#_Toc469614403)

[Project Status Report Template 46](#_Toc469614404)

[Project Status Report Template 49](#_Toc469614405)

[Project Status Report Template 52](#_Toc469614406)

# I. Introduction

## Background of the Problem

This research was inspired by the disaster that wreaked havoc in Leyte, Philippines on November of 2013. This natural disaster, Typhoon Yolanda (internationally called Haiyan), almost leveled the affected areas damaging all infrastructures standing, including telecommunication sites. Due to the design of the current network infrastructure, it takes laborious efforts to repair and restore the network. In Tacloban's case, it took two whole days for major telecommunication companies, GLOBE, SMART and SUN to restore the minimum of their services in major areas such as Tacloban and it took weeks before their services were restored in other remote areas affected. Since communication devices such as smartphones usually rely on these services, people lost the capability of long-distance communication during that time.

When telecommunication services are down, coordination of actions among response teams in multiple areas is more difficult than how it should have been. The lack of communication adds a layer of difficulty in handling situations that demand cooperative action. During these kinds of scenarios, people usually rely on walkie-talkies and satellite phones but those only provide voice communication and are quite expensive to source. Information that are best presented visually such as locations are being distributed using voice calls. Dissemination of other important information can only be done through the radio and it should be done repeatedly to ensure that the messages are received by the intended person. All these measures add unneeded workload and risk of miscommunication to people who would rather spend their energy on focusing on the more important aspects like rescue and recovery.

The absence of telecommunication services is also prevalent in places far from major towns and cities. Until now, there are still places in the Philippines that are yet to be covered by GSM signal from telecommunication companies. Examples of such areas are mining sites and rural towns. Since the target of cell sites are mostly highly populated areas, these far off places are usually out of their range. 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 wastes (Section 19 of Republic Act No. 7942). This limits the methods of communication for the workers to contact each other. Besides the issue on priority, the geographical structure of the Philippines also adds to the difficulty of building new cell towers to give telecommunication access to everyone.

These situations have since then inspired a creation of “E-Nodes”, a portable, easily-set up mesh network that enables communication to smartphone users via the IEEE 802.11 standard. It acts as a temporary solution to help in the coordination of efforts in places without the availability of the infrastructure people usually rely on. The product of this project is still very rigid and new. In its current stage, it is still unusable by the public.

The problem if the E-Nodes are to be set-up is that average citizens do not know how to utilize the network. It is a must that the network have an interface that even non tech savvy individuals can easily use if it aims to connect people. This user interface would have to be able to relay information properly and with the right timing. Especially in the world today where delivering information is very fast paced, the communications system on top of the E-Nodes Project that this research aims to develop should be able to comply with today's standards of communication.

## 

## Statement of the Problem

What is a possible way of creating a communications system that can easily be deployed during post-disaster scenarios and are cost effective

How can the requirements of a post-disaster communication system built on E-Nodes be fulfilled while maintaining the scalability, accessibility and reliability of the network.

## Objectives

### General 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.
* Also, to determine if the implementation of distributed databases in Raspberry Pis is an effective method to create an easily implemented and scalable emergency wireless mesh communications network.

### Specific Objectives

• To connect multiple microcomputers as intermediary nodes of a mesh network;

• To connect mobile devices to the nodes of the network;

• To enable data transfer between nodes;

• To relay a message to and from endpoints of the network (smartphones);

• To enable the network nodes to automatically detect and connect to nearby nodes;

• To design a software solution that will effectively comply with the needs of disaster recovery personnel;

• To design a web application that will be hosted on the microcontrollers that has basic chat functions and mapping services;

• To create a solution that will be cost effective.

## 

## Significance

During and after natural disasters in the Philippines, the Office of Civil Defense (OCD) becomes in charge of providing network communication to the response teams sent by National Disaster Risk Reduction and Management Council (NDRRMC). Currently, when telecommunication services are down, the mostly used communication tool between rescuers and other response teams are handheld radios. Some satellite phones are also used but due to its cost, they are mostly used only in major work stations. The two devices provide only one relevant function which is voice communication. Their network structures are rigid and allows little to no adjustments.

The use of the solution proposed by this paper will allow more than the usual voice as it uses the 802.11 standard. It will enable live-audio, live-video and other forms of visualizations. During post-disaster scenarios, most data that will be exchanged will be time sensitive. Having distributed databases as the backend of each hosted web application will offer the scalability, availability and reliability that wire mesh networks were generally built for. Aside from the emergency response software that the research uses to simulate real post-disaster data transfer, other different software can be easily created and configured into the mesh nodes to fit the needs of different types of users. There are numerous existing applications that are available which can be used with our system. Some possible features that enable real time communication are online chat messaging, voice call, camera sharing, and file sharing.

The research is to implement distributed databases to the wireless mesh network that make use of devices that majority of Filipinos already have, smartphones. Because of its popularity, it can be said that people now have a good grasp of how to use smartphones, apps, and websites in general, implying that the training required to use the system is minimal. Training will only consist of the correct way of handling the nodes and how to plan its deployment.

The successful implementation of a communications system will improve all aspects of most operations in the ground level during post-disaster scenarios. It will save costs incurred from improper provision of needs like water, food, and medicine and of services like medical checkups and treatment to victims. It will also hasten planning and implementation of rescue missions and remove the chances of error due to lack of communication. It will save time and money by improving planning times and methods. It may definitely increase the number of victims saved as human lives should always be prioritized.

# II. 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 Raspberry Pi 3 model B that act as nodes of a wireless mesh network and smartphones with 802.11 capabilities as end devices. The prototype being developed along with the research is limited to three Raspberry Pis due to budget constraints. Smartphones that will be used throughout the project will all be Android smartphones.

The study covers the development of the network nodes along with its input and output interfaces. A web application that users can use for communication during post-disaster situations will be developed using Laravel. This web application will be hosted in each node and will use its network for connectivity.

To ensure data localization, availability, and consistency, a distributed database management system, specifically Apache Cassandra, will be used together with the web application.

To limit the scope of the research, security will be limited to WPA2 security keys to be entered when connecting to the network. The system will then only rely on the security features that is implemented by Laravel.

Actual implementation of the system will not be a part of the research. Testing will be done using simulations in two different environments: line of sight and obstructed.

Power source and consumption of the nodes will be considered but not thoroughly discussed as it is part of a different field

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 in real post-disaster scenarios events.

# III. Review of Related Literature

### 

### Inspirations

To solve the problem of communication during times of disasters, the researchers had to think of ways to enable communication between people within an area that is independent of the current telecommunication services. The initial idea was to make use of smartphones since a large number of the population already rely on it for communication. The 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 apache or nodejs, 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 on August 15, 2016 from <https://developer.android.com/guide/topics/connectivity/wifip2p.html>).

Wi-Fi Direct™ enables IEEE 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. The company 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 on August 15, 2016 from <https://play.google.com/store/apps/details?id=com.opengarden.firechat&hl=en>).

Although the researchers consider 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 maximum range of approximately 60.96 meters. This limitation is due to the fact that it relies on Bluetooth to maintain a device’s connection to other devices in the network. This means that it is best used for dense communities with a lot of users but bad for communities with sparse users. If users are more than approximately 60m 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 service to enable connectivity, hence ad-hoc. 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 be placed between users far from each other. This in effect, extends the range of the MANET. The researchers tried to look for such devices and found a number of possible candidates. Among them, two are popular in the open source community which provide developers more available documentations and guides. These two are Arduino and Raspberry Pi. Initially the researchers looked into Arduino as nodes for the network since it has a number of clones that are cheaper than routers and Raspberry Pi. According to the Arduino website, “Arduino is an open-source prototyping platform based on easy to use hardware and software” (Retrieved in August 19, 2016 from <https://www.arduino.cc/>). Arduino is apt for simple applications like sensor networks and Internet of Things (IoT) but is not recommended for other more complicated projects, especially not for routing. There are some projects that tried to use it as a routing tool but had unsatisfying results. 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>).

On the other hand, Raspberry Pi is a credit card sized microcomputer that runs mostly on Linux-based operating systems. It has its own processor, memory, graphics card, and other parts typically found in a computer. Due to its hardware capacity, it is currently used in projects that require computers with low end specifications such as simple routing, ad-hoc website hosting, and IoT nodes (Retrieved on August 18, 2016 from <http://elinux.org/RPi_Hub>). It is also confirmed that Raspberry Pis are able to act as nodes in a mesh network (Besneatte, Commotion Pi, June 12 2014; Kidder, HSMM-Pi, July 21 2013).

As mentioned earlier, all devices in a MANET acts as a node whether it be a Pi or smartphones. Raspberry Pis are fully capable of this feature however, after conducting further research it was found out that Android smartphones are not. In the earlier versions of Android, it has a function called ad-hoc mode that a user can toggle when the device is rooted. It is similar to the ad-hoc mode of other devices such as laptops and routers wherein it allows the device to connect to a network that is not connected to the internet. With this option, it is possible for smartphones to act as nodes as shown by other projects (Serval Mesh). The said feature however, was removed by Google from the Android OS from API14 and above (year 2012) (source). A workaround for this problem is that only the Pis will act as nodes and connect to the mesh network while the smartphones will be the end-user device that is connected to an access point of the Pi. With this, the smartphones will not be relaying and rerouting messages to other smartphones. This kind of setup means that the nodes are not mobile therefore the network is not considered MANET but a Wireless Mesh Network (WMN).

### 

### Wireless Mesh Networks

WMN is a network setup wherein nodes are interconnected like in MANET but does not consider the mobility of nodes. 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. The type of WMN to be used in the project is partial mesh wherein not all nodes are interconnected to each other. Some nodes will only reach other nodes by passing through the network. As opposed to full mesh wherein all devices have a direct connection to each other.

WMN uses different routing protocols to handle its connection. Routing protocols dictate where and how the router, the mesh nodes in our case, will distribute packets throughout the network. There are a numerous routing protocols currently available and each of it has its own design and purposes (Cisco Networking Academy, 2014). The routing protocols mostly used in mesh networks are the Ad-Hoc On-Demand Distance Vector Routing Protocol (AODV) which only builds a route when a message needs to be sent and the Optimized Link State Routing Protocol (OLSR) which constantly maps the entire network.

As the connection between the nodes are accomplished using WMN, it is now necessary to enable smartphones to connect to the network. A Wireless Access Point (WAP) is needed to achieve this. An access point lets devices access the network by connecting to it. With this, the Raspberry Pi setup requires two NIC cards, one to connect it to the mesh and the other to act as a WAP. hostapd is installed in the Pi to enable a Wi-Fi dongle connected to the it act as a WAP (Retrieved August 18, 2016 from <http://w1.fi/hostapd/>). There is an exhaustive table (Retrieved on August 18 2016 from <http://elinux.org/RPi_USB_Wi-Fi_Adapters>) that provides a detailed list of different Wi-Fi dongles and its compatibility with Pi. Some dongles have issues with its driver compatibility so the table is used to decide which dongle to use. Considering compatibility, price, and availability in Manila where the research is being conducted, Tenda W311U/+ is selected to be used.

### 

### Distributed Databases

With all the information gathered up to this point, the researchers can choose to proceed using two different methods. First is developing a mobile application that will rely on the network provided by the Pi. This will act as the interface used by the users to communicate with each other. However, this kind of approach has a major issue: applications need to be installed in a device. This means that only the devices that were prepared before the deployment can be used limiting the number of possible users. Moreover, if those devices got lost or rendered unusable, it will reduce the effectiveness of the communication system. Because of this, it is deemed to be unfit for the research. The second possible method, which is opted by the researchers, is to host a web application in the Raspberry Pis that users can access using their smartphones. The users then only needs to know the password of the access points and they can use any smartphone that satisfies the minimum requirements of the web application. This kind of setup resembles a star topology.

The research will then be using a partial mesh in combination with a star network topology. Each node in the mesh network will have separate star networks. 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. 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>). Since the topology of the end users’ connection to the nodes is considered to be star, it is required that each node contain all data stored in all other nodes to enable communication. This can be achieved using distributed databases. A distributed database is a type of database wherein users interact with only one database but can access or modify data stored in other database as long as belong to the same cluster. All databases in a cluster store its data on multiple nodes by data replication and the communication from a node to the others is enabled by the computer network (Retrieved on October 30, 2016/ <https://docs.oracle.com/cd/A57673_01/DOC/server/doc/SCN73/ch21.htm>). Data replication refers to the synchronization of data in a distributed database (<https://www.tutorialspoint.com/mongodb/mongodb_replication.htm>). This type of database is managed by a software called a distributed database management system (DDBMS) (Retrieved on October 31, 2016/ <https://cs.uwaterloo.ca/~tozsu/courses/cs856/F02/lecture-1-ho.pdf>).

The research requires that nodes should not rely on a central device so it also needs a DDBMS that has master-master data replication. All nodes should be considered of equal level and no single node should control the access of the others. DDBMS is usually used for data availability and backup such that if the main (master) goes down, the other (slave) will go live. Hence, some DDBMS has a master-slave data replication wherein all data comes in to the master and passed down to the slave. One DDBMS that fits the criteria of the research is Apache Cassandra. Cassandra, from The Apache Software Foundation, is an open source DDBMS that was made to manage “very large” data among a distributed database with “…a highly available service with no single point of failure…”. This DDBMS uses NoSQL and is recommended for databases managing large amounts of data. Data from this database is synchronized using the Gossip protocol which was “inspired” from Dynamo. It is also told to be using BigTable’s “way of modeling”. Cassandra has its own query language called “Cassandra Query Language” or CQL. (Lakshman & Malik, 2008) or (Retrieved on October 26, 2016/ <http://immagic.com/eLibrary/ARCHIVES/GENERAL/WIKIPEDI/W120911A.pdf>).

### 

### Related Projects

The idea of wireless mesh networking is not new, but it is not old either. Up til now there are numerous organizations which continue to develop on the technology for all sorts of uses. Mostly for non-commercial purposes, the projects briefly discussed below helped shape the solution that the researchers chose to test distributed databases on.

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>).

Commotiond is an open source firmware / 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 by 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 is an application that lets mobile phones make the use of Wi-Fi or 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 can benefit the project since the nodes will serve as an access point and everyone will be under the same Wi-Fi network. It will serve as a good proof of concept and testing tool of how reliable communication can be established and maintained in the researched network deployment.

# 

# IV. Theoretical Background

### Wireless Mesh Network

Wireless mesh network (WMN) architecture is a good method of enabling fast and scalable data communication between nodes. The network is built by multiple self-organizing wireless nodes that routes data through the use of different protocols to enable multi-hop communication. This would be the most hassle free way of deployment of a temporary network because it will not need wires, and it can be used and reused in different locations. It will surely fit the communication needs of emergency response/recovery groups.

The most common setup in households for their wireless internet follows a star-topology in which the user's end devices are connected to a central access point. This is what people are used to and they already know how to connect to wireless access points easily given that they know the proper passphrase. This topology and method of connection will be used in each individual node such that the end user’s smartphones will be connected in the same method. It will remove the complexity of having to teach people how to connect to mesh networks, and at the same time solving the problem of android devices not having support for ad-hoc mode, the mode that is used to discover and connect peer-to-peer or device-to-device directly.

Compared to the BSS(Basic Service Set), where client devices connect to an access point, The IBSS (Independent Basic Service Set) or the network that follows wireless mesh networking architecture on the other hand, does not follow this centralization of data traffic and allows a device to wirelessly connect to other devices directly. Wireless mesh networks have a "self-healing" capability such that the connections are dynamically configured each time a node is added or removed. Packets of data can travel through multiple routes guided by the algorithms of different routing protocols. Depending on the proximity from other nodes, or if there are obstructions that may hinder the sending and receiving of radio waves, bandwidth, latency ,and jitter differs. But since this kind of network does not rely on any central network, the only way of shutting down a mesh network is by disabling all nodes in the network. It is the perfect setup for emergency situations as it removes the risk of having a single point of failure, at the same time, it creates a more resilient method to store data if each node in the system were to have a complete and up to date copy of all data at all times. This is where distributed databases come in.

### 

### Distributed Databases

Distributed databases are data stores where the data is not only stored in one location but in multiple. It may have different setups like separation of whole data into different chunks being stored in geographically remote data centers or having a full clone of a dataset in multiple data stores, what defines distributed databases is its data replication. It is what determines if the data is consistent and available within the network.

There are three main ways distributed databases do replication, there’s Master/Slave, Masterless and Primary Owner. Master/Slave replication is when writes are only made on designated master nodes. Each change to master nodes are then replicated to slave nodes connected to the master nodes. Reads can be done on both slave and master data stores. The only problem with this setup is, it is made for high read and little write. It forces the users to write only on data stores with the role of master. The masterless way on the other hand is that clients write functions are executed on multiple nodes, which then replicate the same data to all of its neighbors. This solves the problem of the first one by having all nodes writable but consistent reads / writes on multiple nodes will produce high network traffic and can bog down the system. The last one, Primary Owner, is the best of both worlds. Both reads and writes are executed on nodes called primary owners and then replicated and it reads and writes on every node. Here, every node is a primary node

The use of a masterless or a primary owner pattern of replication will be great for the post-disaster wireless mesh network because it means each node can have near identical datasets at all times. The researchers need not configure specific nodes to become master and force the web applicaiton to write only on those particular data stores. It maintains the easily scalable network because in most distributed databases that follow the master/slave pattern, the number of masters must be explicitly stated which makes adding additional nodes once the network infrastructure has been set up difficult.

Distributed database is perfect for a communications system that handle time-sensitive data. It can maintain availability of data by having complete and up to date information on every node because you can adjust the frequency of the data replication. It means that anyone connected to any node will almost always see the same data that everyone within the cluster sees. It also makes it easy for the users to retrieve information because the web service hosted on a node just gets it from its own local data store. It can easily be scaled up or down because each additional node will automatically update itself until it becomes a complete database once it is discovered as member of the database cluster. This removes the uncertainty of whether the data an end user sees on their web application is complete because even if a node is newly set up, it can automatically pull historical data from other nodes and display it on it’s service.

### 

### Routing Protocols

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 mostly used in mesh networks are AODV and OLSR.

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). This kind of routing protocol is suitable for networks with low traffic but unsuitable for ones with high traffic since the number of routes it needs to form is directly related to the number of messages transmitted.

OLSR is a revision of Link State routing protocol wherein all routers/nodes in the network maintains a map of the entire network. When a message needs to be sent, the routers already know which route to use. 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. This kind of routing protocol is unsuitable for network with low traffic because of its high overhead when maintaining routes. On the other hand, it is suitable for network with high traffic because its overhead is not dependent on the number of messages transmitted.

# 

# V. Design and Methodology

## Components

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.

### 

### Hardware

**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/>).

**Battery Pack / "Powerbank"**

20,000mAh is advisable for longer life of the node. This provides roughly 6 and a half hours of life for a node with high activity.

**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>).

**Class 10 MicroSD card**

This will be where all the software will be installed on. It is more advisable that a faster write speed version of the SD card be used because it will greatly affect the performance of the nodes overall

### 

### Software

**Optimized Link State Routing Daemon(OLSRD)**

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.

**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>).

**Ubuntu Xenial Xerus 16.04**

Ubuntu is a Linux based operating system that is opensource. This will be the operating system used for the project because it is the one that supports QGIS and its dependencies.

## 

## Steps

### Sourcing of Hardware

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 / Assembly / Initial boot

First, the purchased micro SD card is formatted to FAT32 so that the flash memory can be readable and usable by the OS. Before operation of the Raspberry Pi, the image of the current OS 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 also 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.

### Olsrd Installation and Setup

Olsrd is installed by using the apt-get install command from the Ubuntu repositories. Once the installation is finished, 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.

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

After olsrd is installed and configured, the researcher will run the daemon in debug mode. Olsrd is run in debug mode where the discovery of nearby nodes is monitored to know if it has been configured properly. This will be done installed and run on all nodes and they should be able to detect each other. Once the mesh network is verified to be running properly, ping tests are conducted to verify inter node communication.

### DHCP Setup

Download dnsmasq from apt package manager and nano into the configuration.

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

Also configure here the hostnames the access points clients will resolve because each access point will also be acting as their own dns servers.

### Wireless Access Point Setup

Download hostapd and change a line in /etc/default/hostapd to point to the configuration file at /etc/hostapd/hostapd.conf

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. If configured correctly, the smartphones should be able to detect the access point after running hostapd by typing "service hostapd start"

### Network Address Translator Configuration

Data crossing from the subnetwork through 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.

### Installation and configuration of Datastax Enterprise/Apache Cassandra

Download and install Datastax Enterprise from their website. It will come as a tar file but just extract it via 'tar -xvf <filename>'. Change the configuration files to assign the seed nodes for the database cluster and to change the database cluster name.

### Development of the Chat Application

A login/register template from Laravel is configured and used to be able to collect and store the MAC address and the Name of a user in a local databse. A simple chat application is then created via PHP where all message data together with the user's name is forwarded to the local Apache Cassandra database through an API.

### Development of the API

Because the available drivers of cassandra for php are mostly incompatible with 32-bit chips, the developers had to create an API to handle the creation and retrieval of data from Apache Cassandra. The API will be coded using the Python programming language as there is a python driver for Apache Cassandra that is up to date and available on GitHub.

### Installation of QGIS

QGIS is downloaded and installed on any computer for editting and viewing the maps before hosting.

### Sourcing and Designing the Offline Map

Free offline map layers can be downloaded separately from OpenStreetMaps.com and PhilGIS. These layers are then put together using QGIS and colored to highlight the important features that the creator wants to show.

### Customization of the Map Interface and Development of Plotting Functions

Custom buttons are designed then created to be plotted on the map. Buttons are then created on the toolbar of the QGIS client. Their corresponding functions are also created to be able to accept touch inputs and transfer coordinate data (longtitude,latitude) and marker data to the API which will then add it to the Apache Cassandra Database.

### Unit Testing

All the functions are tested independently to check for bugs. When all units pass the minimum required standards, the units are then combined together to be able to create the whole application

### Integration of Applications to Apache Cassandra

Each part of the PDECS is connected to Cassandra with utmost care as the RAM of the Raspberry Pi is relatively small.

### System Testing / Usability Testing

When all the parts of PDECS are successfully put together, the whole system will be tested with the mesh network nodes to see if it can be used in actual post-disaster scenarios and how it can be further improved

# 

**VI. Results and Discussion**

E-Nodes

The nodes were able to successfully establish a private WMN. Users are also able to connect to the access point and are given IP addresses by the DHCP.

To measure the effectiveness of the network established by the E-Nodes, it was tested using the following metrics: ping, TCP and UDP throughput, Packet Delay Variation (also known as jitter) and error rate.

Ping is the time it needs for a packet to be sent by a source, received by the destination, and then resent back to the source. It is the most common tool people use in order to test the availability or roughly measure the capability of a network or server. This is due to the fact that it is easily available and its results are easy to comprehend since it is just a measurement of time in milliseconds(ms). A high ping is unfavorable because it means that packets will take a lot of time before being received on the other side of the connection, so it is usually a sign of a bad connectivity or hardware capacity.

Due to how ping works, it can be easily deduced that its value is expected to increase along with distance. In the results of this research, the ping can be seen having irregular behavior but has an increasing trend. The ping values in different distances does not vary that much even after the 410m point. Due to irregularities in the environment of the testing location, some of data gathered were skewed. The spike in the 270m mark could be attributed to a depression in the testing area.

Although checking the ping provides a good benchmark in analysing the performance of a network, there are also other metrics to look out for. One of such metrics is throughput. Throughput is the amount of data that the network can successfully transfer from one point to the other. As data is sent through bits, it is usually represented as Bits/second.

Both TCP and UDP throughput values were obtained since the network is aimed to relay both types of packets. Because the medium used in the research is wireless, the throughput values showed a significant decline in values as distance increases. This may be due to the fact that the number of elements that may block the Wi-Fi signal increases along with distance. There is a significant drop in the throughput values after the 300m mark. The value obtained at 420m is only 1.405MBit/s which may be a bit slow specially when the traffic within the system is heavy.

PDV is related to the ping of a network. It is the variation in the ping values for a specified amount of time. A high PDV shows that a network may be unstable because the ping varies too much. Due to the same reason throughput values in the research decreased with distance, PDV values increased with it. This means that the network becomes more spiky with distance.

Because UDP packets are not ensured to reach its destination, the error rate values were also obtained. Error rate is the amount of packets that is lost therefore not received by the intended client or server. A high error rate means that the network is unstable. Due to the same reasons mentioned above, the error rate values in the research increases with distance. There are some variations in the pattern but the general trend can be seen to be increasing. Up until the 180m mark, the error rate is 0% meaning no packets were lost in the connection. Past that point however, the values spiked significantly. At 420m, the error rate value is already quite significant so it marked the limit of the testing.

The same metrics were used to measure the performance of the access points of the system. In the testing of the mesh connectivity, two Wi-Fi dongles with antenna were used. In the access point test on the other hand, the built-in Wi-Fi module of the Pi and a smartphone was used. Since different hardware were used in both tests, it is to be expected that the test would also yield different results. However, the trend persists in both tests. Ping, PDV, and error rate increases with distance while throughput decreases. After the 160m mark, the smartphone had some issues detecting the access point so the test was concluded.

PDECS

The integration of PDECS with its distributed database to E-Nodes are correctly passing and updating information to each other. When users are connected to any node in a cluster, their databases update the information when any user writes on a node. The websites are displayed properly and connect properly to the database. Each function on the website worked as well as it did on the Unit Tests and everything performed semi-smoothly. This is to be expected as all communication is done wirelessly and the type of smartphone is also a consideration.

**VII. Conclusion**

Three Raspberry Pis were able to connect to each other forming a small WMN. With the results obtained in the tests, it can be concluded that the maximum effective mesh range of the E-Nodes is approximately 400m. Since the materials used to enable this connection is just a regular Wi-Fi dongle with antenna and is relatively cheap, the results obtained showed good signs in the applicability of mesh networking in post-disaster scenarios. Just this bare network is actually already usable. Some mobile applications that rely on local networking for connectivity (such as Walkietooth) were able to successfully utilize the network and enabled communication between two users.

For the PDECS's implementation to E-Nodes, it can be said to be successful up to a certain point. E-Nodes can successfully host the PDECS while maintaining its WMN, however its resource consumption sometimes make the Pis or itself crash. Apache Cassandra consume approximately 60% of the memory of the Pi every time it is run. Given that the memory of Pi is only 1GB this is resource consumption is rather high and significantly slows down the Pi.

**VIII. Recommendations**

This research covers a wide range of disciplines that future researchers can expand. The research team wants to promote the focus on improving and adding to the functionalities of PDECS. It still has a long way ahead before it can be usable in scenarios where lives are at stake but the team envisions it to be a toolkit of disaster response teams in the future.

Aside from the front end of PDECS, there are still alot of considerations regarding the long term deployment of Cassandra. For sure, like alot of databases, it is prone to bloating since it clones all data across all nodes. If there is a way to preserve the data and provide continuity to the operation of the nodes then that would be a great add-on.

It is also a huge concern how big memory Apache Cassandra consumes even just with the startup. It maybe not the right time to deploy this database but surely in future versions of Raspberry Pi's, if there will be any, ram capacity would increase as IC's grow cheaper and smaller as time passes. The team is just banking on the principle that "Even if the software is too far ahead it's generation, the hardware always follows."

A good integration to this whole system would be an outdoor casing that considers signal propagation, weather, and insect protection.

Power consumption is another place to look at if ever anyone wants to improve this system further. Efficient power usage by optimizing hardware, software, or even procedures on how it should be deployed is a whole different study.

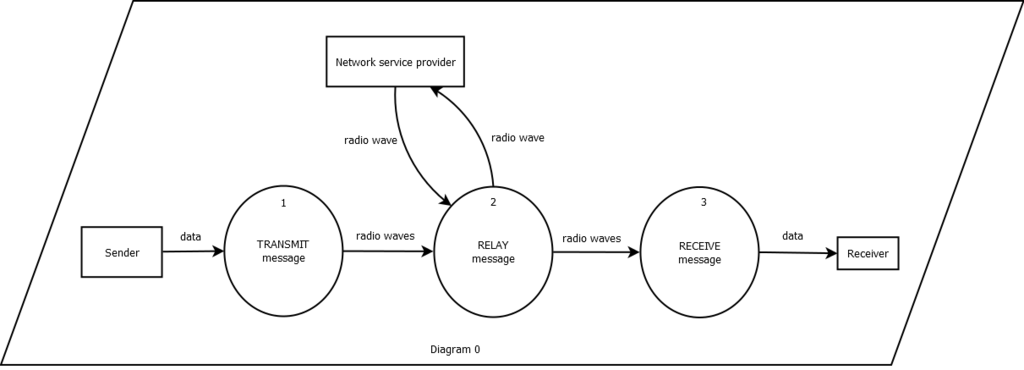
# Appendices

## 

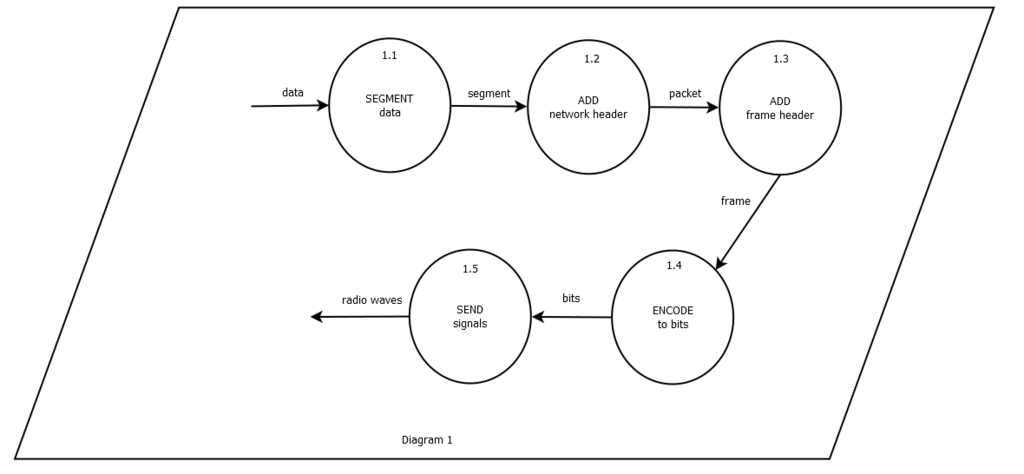
## Data Flow Diagram (E-Nodes)

### 

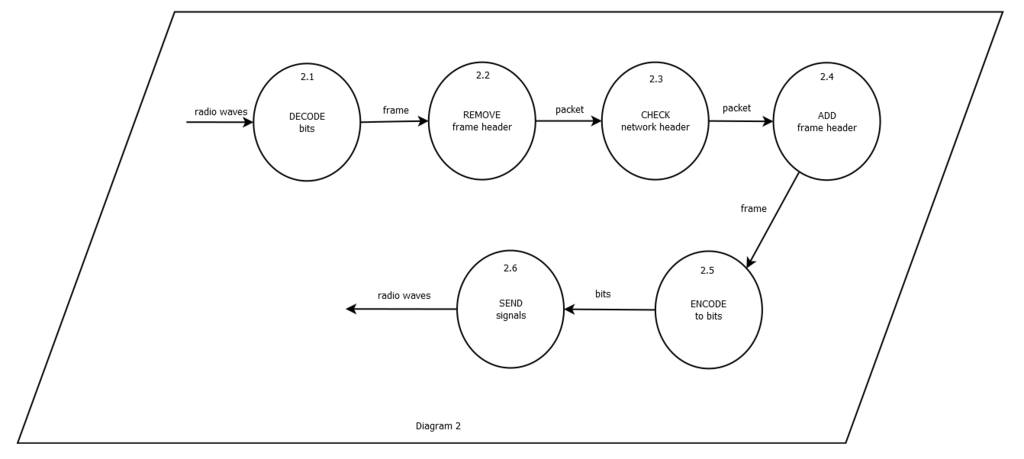
### Diagram 0



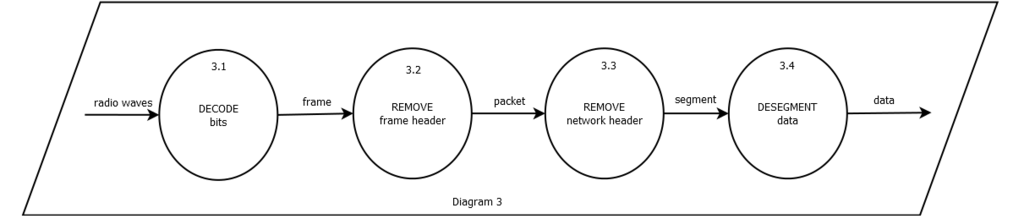
### Diagram 1



### Diagram 2



### Diagram 3



## 

## Process Specifications

### 

### Diagram 1

* **Process 1.1**

When data is inputted, it is divided into different segments so that they would be transmitted easier. Transport headers are added to each segment.

* **Process 1.2**

Network headers are added to each segment. These headers contain information such as source and destination IP address. The end products of this process are called packets.

* **Process 1.3**

Frame headers are added to each packet. These headers contain the MAC addresses of the source device and the destination device. The destination device is the device at the next hop in the path.

* **Process 1.4**

Frames are encoded into bits.

* **Process 1.5**

Bits are converted to radio waves to be able to travel through air.

### Diagram 2

* **Process 2.1**

Radio waves are converted to bits and then decoded into frames.

* **Process 2.2**

Frame headers are removed to access the packet.

* **Process 2.3**

Network headers are checked to know if the current device has the destination IP header.

* **Process 2.4**

Frame headers are changed to match the current MAC addresses of the source and destination devices.

* **Process 2.5**

Frames are encoded into bits.

* **Process 2.6**

Bits are converted to radio waves to be able to travel through air.

### 

### Diagram 3

* **Process 3.1**

Radio waves are converted to bits and then decoded into frames.

* **Process 3.2**

Frame headers are removed turning them into packets.

* **Process 3.3**

Network headers are removed. The end products are called segments.

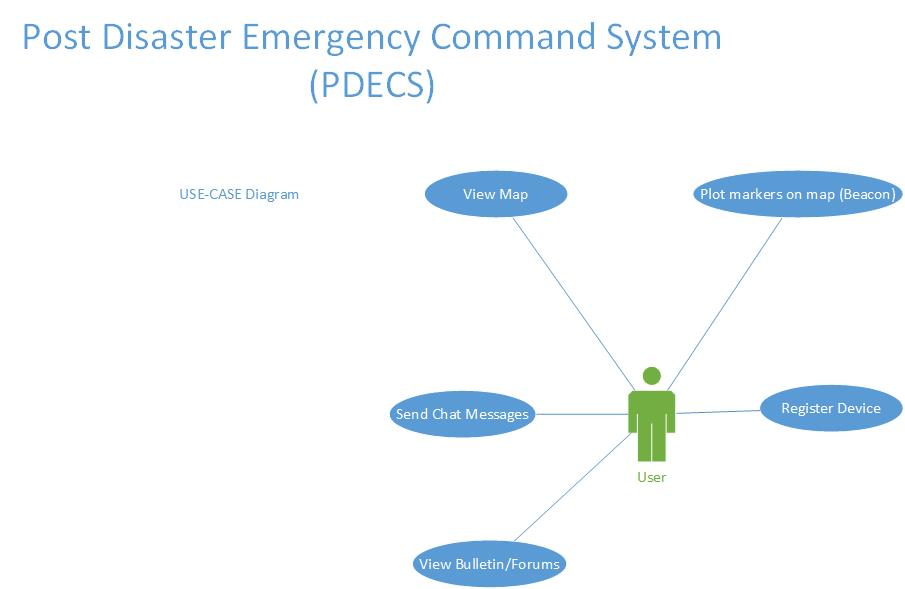
* **Process 3.4**

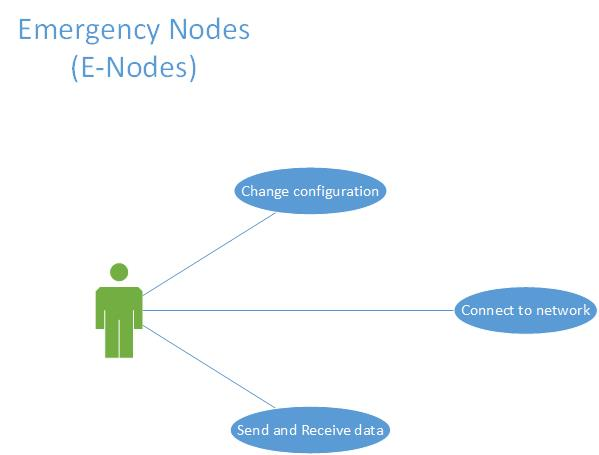
The segments are desegmented. They are rearranged to form the data that was sent by the sender.

## network-arki.jpg

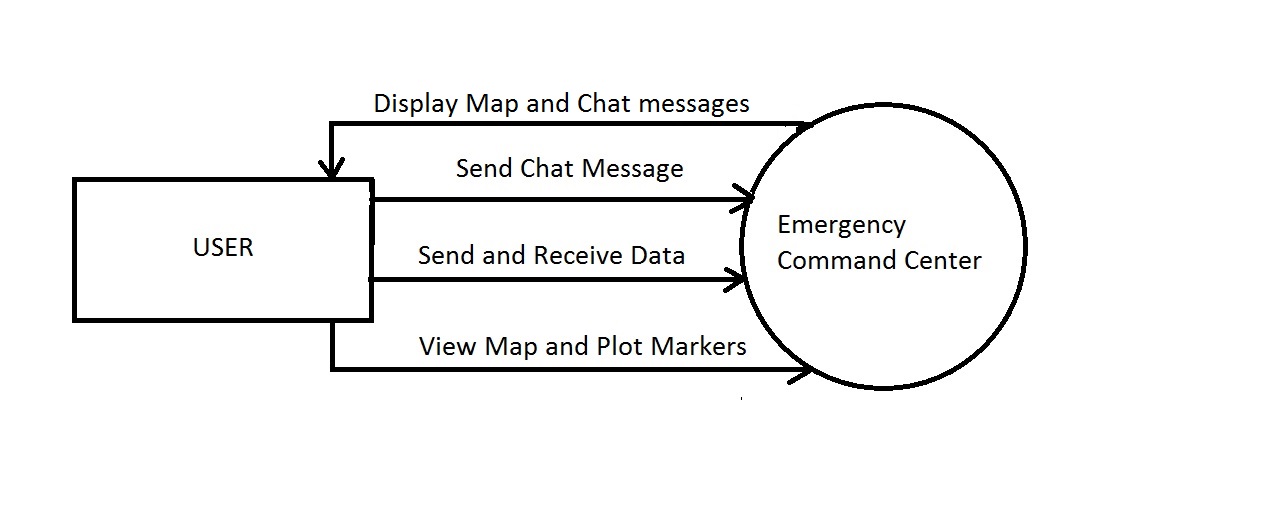
## Network Architecture

## Use Case Diagram

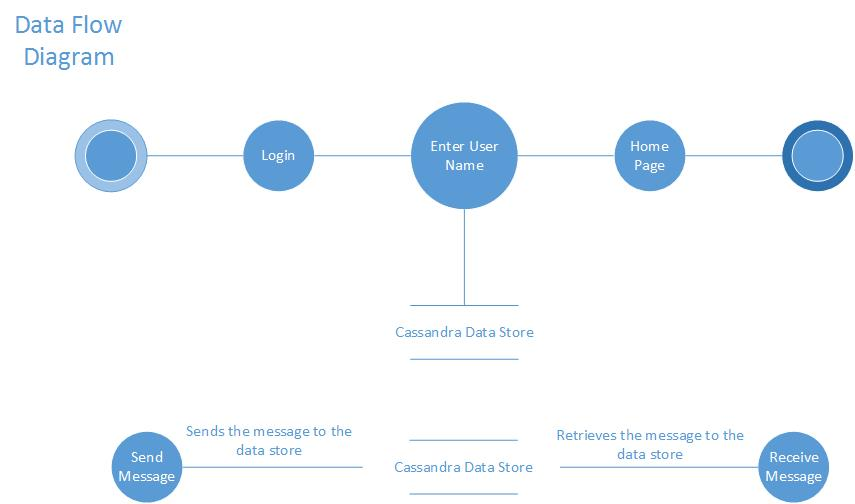




## Context Level Diagram (PDECS)



## Data Flow Diagram



## 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



## 

## Tables

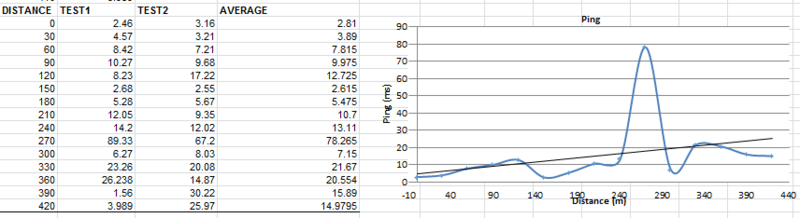
Table 1. Mesh Network Ping

Table 2. Mesh Network TCP Throughput

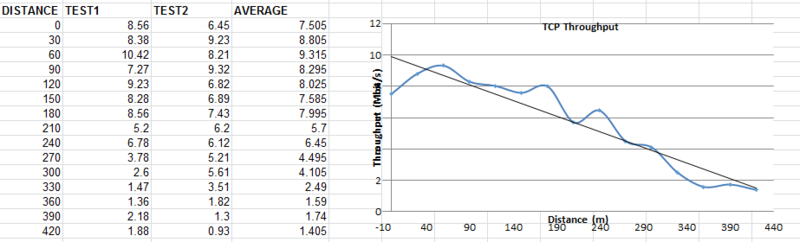
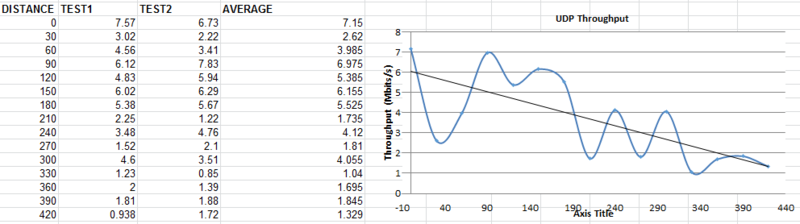
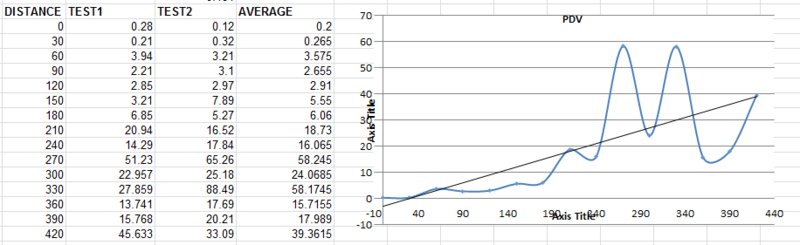
  
Table 3. Mesh Network UDP Throughput

Table 4. Mesh Network PDV



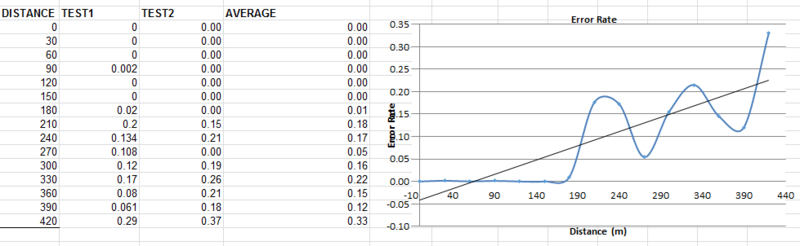
  
Table 5. Mesh Network Error Rate

Table 6. Access Point Ping

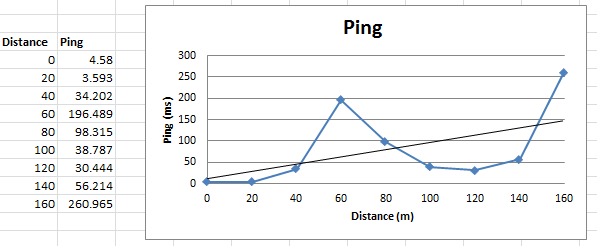


Table 7. Access Point TCP Throughput

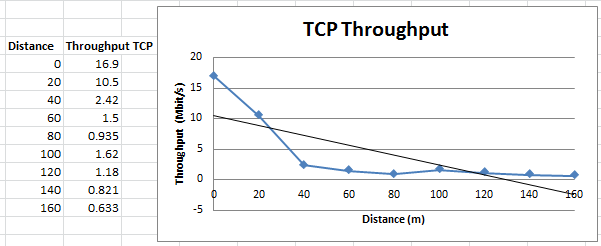
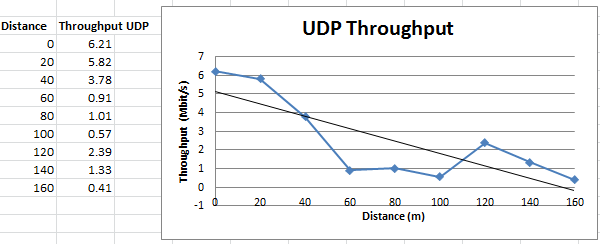
  
Table 8. Access Point UDP Throughput

Table 9. Access Point PDV

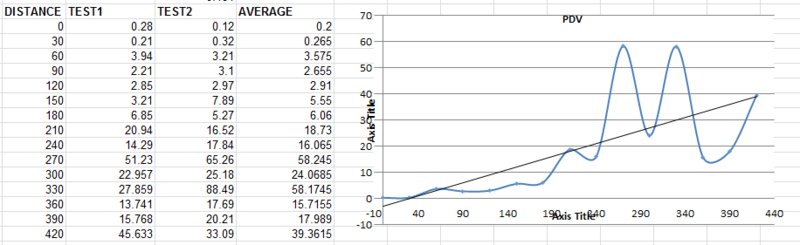
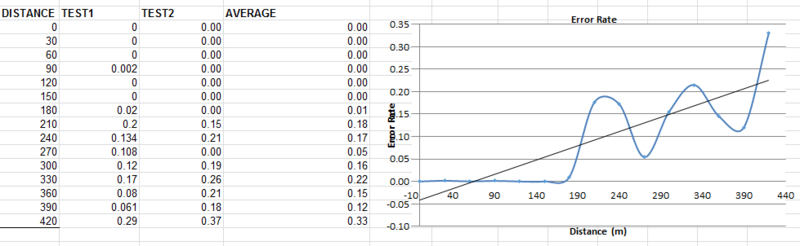


Table 10. Access Point Error Rate



## 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/courses/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](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>

<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/>



**Project Status Report**



**Project Name:** Lightweight Mesh Network Nodes

**Department:** Computer Science

**Focus Area:** Wireless Networking

**Product/Process:** Mesh Network Nodes



**Prepared By:**

|  |  |
| --- | --- |
| **Document Owner(s)** | **Project/Organization Role** |
| Benjamin Rivera | Project Manager |
| TomioTonoike | Senior Analyst |
| Allen Ellana | Software Specialist |
| AlfonsinTison | Documentation Head |

## Project Status Report Template

|  |  |  |
| --- | --- | --- |
| Project Name | | |
| Prepared By:  Tomio Tonoike  Allen Ellana | Date:  10/03/16 | Reporting Period:  09/26/16 to10/03/16 |
| Project Overall Status:  Currently in the testing phase of prototype of the project. Documentation about the prototype | | |
| Project Summary:  We are currently in the process of applying existing libraries and software technologies with their respective networking protocols Raspberry Pi units to create the mesh networks. We will compare and contrast these existing methods to identify and develop the ideal setup for a raspberry pi and smartphone based mesh network. | | |
| **Milestone Deliverables performance reporting over last period**   |  |  |  |  | | --- | --- | --- | --- | | **Milestone Deliverables** | **Due Date** | **% Completed** | **Deliverable Status** | | Raspberry Pi Networking software installation | | | | | * Test OLSRD | 09/30/16 | 100% | [**On Schedule**] | | * Implementation of OLSRD | 9/30/16 | 100% | [**On Schedule**] | | Connectivity of devices | | | | | * Installation of OLSRD on 3rd Pi | 9/3016 | 100% |  | | * Testing the range of the mesh network | 09/30/16 | 100% |  | | * Connectivity Between the three Pi’s | 08/25/16 | 100% | [**On Schedule**] | | | |
| **Milestone Deliverables scheduled for completion over next period**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Milestone Deliverables** | **Due Date** | **% Completed** | | **Deliverable Status** | | Indentification of Test Cases for networking the Pi | | | | | | * Testing Byzantium Pi | 09/26/16 | 50% | |  | | * Testing BABELD | 09/26/16 | 50% | |  | | * Testing CommotionD | 09/26/16 | 50% | | [**On Schedule**] | | Experimentation | | | | | | * Testing the througput | 09/28/16 | | 50% |  | | * Speed testing the mesh | 09/28/16 | | 0% |  | | * Establishing metrics | 09/28/16 | | 0% |  | | | |
| **Project impact of milestone success or failure for project remainder**   |  |  | | --- | --- | | We may have to remove the part about using multiple scenarios in deploying the different methods of deployment due to time constraints. | We will focus more on the metrics we will be identifying and compare and contrast the different existing methods by testing in only one or two simulated scenarios | | | |
| **ProjectBudget/Financial Status**   |  |  |  |  | | --- | --- | --- | --- | | **Budget Item** | **Planned Budget** | **Actual Cost** | **Variance/Explanation** | | 2 Units Raspberry Pi | 5000php | $80 | Raspberry Pi 3 are cheaper than their predecessors | | 2 Memory Cards (16GB) / 2 Charger (5W 2.5A) | 2000php | 2000php | No Variance | | | |
| **Project Risk Management Status**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Risk and Description** | **Risk Chance** | **Risk Impact** | **Risk Priority** | **Change from Last Review** | | Our hardware may not be fully compatible with all the methods of deploying mesh networks. | Medium | High | High | Same | | Weather | High | Medium | Low | Same | | | |
|  | | |
| **Project Recommendations**   |  | | --- | | * Locate a suitable location for testing * Find locate more consultants * Research on further applications once the project has finished | | | |
| **Objectives for Next Project Status Review**   |  | | --- | | By the next status review, we should be already in our testing and data collection phase.  The data collected will then be compared and contrasted among each other in order to formulate | | | |
| **Related Project Information**   |  | | --- | | None as of the moment | | | |





**Project Status Report**



**Project Name:** Lightweight Mesh Network Nodes

**Department:** Computer Science

**Focus Area:** Wireless Networking

**Product/Process:** Mesh Network Nodes



**Prepared By:**

|  |  |
| --- | --- |
| **Document Owner(s)** | **Project/Organization Role** |
| Benjamin Rivera | Project Manager |
| TomioTonoike | Senior Analyst |
| Allen Ellana | Software Specialist |
| AlfonsinTison | Documentation Head |

## Project Status Report Template

|  |  |  |
| --- | --- | --- |
| Project Name | | |
| Prepared By:  Alfonsin Tison  Allen Ellana | Date:  10/10/16 | Reporting Period:  10/03/16 to10/08/16 |
| Project Overall Status:  Currently in the testing phase of prototype of the project. Documentation about the prototype | | |
| Project Summary:  We are currently in the process of applying existing libraries and software technologies with their respective networking protocols Raspberry Pi units to create the mesh networks. We will compare and contrast these existing methods to identify and develop the ideal setup for a raspberry pi and smartphone based mesh network. | | |
| **Milestone Deliverables performance reporting over last period**   |  |  |  |  | | --- | --- | --- | --- | | **Milestone Deliverables** | **Due Date** | **% Completed** | **Deliverable Status** | | Raspberry Pi Networking software installation | | | | | * Iperf installation | 10/08/16 | 100% | [**On Schedule**] | | * Range testing | 10/08/16 | 100% | [**On Schedule**] | | Connectivity of devices | | | | | * Network throughput testing | 10/08/16 | 100% |  | | * Testing the range of the mesh network | 10/08/16 | 100% |  | | * Connectivity Between the three Pi’s | 10/08/16 | 100% | [**On Schedule**] | | | |
| **Milestone Deliverables scheduled for completion over next period**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Milestone Deliverables** | **Due Date** | **% Completed** | | **Deliverable Status** | | Indentification of Test Cases for networking the Pi | | | | | | * Testing lower overhead | 10/17/18 | 10% | | [**On Schedule**] | | * Testing mesh in open field | 10/17/18 | 0% | | [**On Schedule**] | | Experimentation | | | | | | * Speed testing the mesh | 09/28/16 | | 0% |  | | * Establishing metrics | 09/28/16 | | 0% |  | | | |
| **Project impact of milestone success or failure for project remainder**   |  |  | | --- | --- | | We may have to remove the part about using multiple scenarios in deploying the different methods of deployment due to time constraints. | We will focus more on the metrics we will be identifying and compare and contrast the different existing methods by testing in only one or two simulated scenarios | | | |
| **ProjectBudget/Financial Status**   |  |  |  |  | | --- | --- | --- | --- | | **Budget Item** | **Planned Budget** | **Actual Cost** | **Variance/Explanation** | | 2 Units Raspberry Pi | 5000php | $80 | Raspberry Pi 3 are cheaper than their predecessors | | 2 Memory Cards (16GB) / 2 Charger (5W 2.5A) | 2000php | 2000php | No Variance | | | |
| **Project Risk Management Status**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Risk and Description** | **Risk Chance** | **Risk Impact** | **Risk Priority** | **Change from Last Review** | | Our hardware may not be fully compatible with all the methods of deploying mesh networks. | Medium | High | High | Same | | Weather | High | Medium | Low | Same | | | |
|  | | |
| **Project Recommendations**   |  | | --- | | * Locate a suitable location for testing * Find locate more consultants * Research on further applications once the project has finished | | | |
| **Objectives for Next Project Status Review**   |  | | --- | | By the next status review, we should be already in our testing and data collection phase.  The data collected will then be compared and contrasted among each other in order to formulate | | | |
| **Related Project Information**   |  | | --- | | None as of the moment | | | |





**Project Status Report**



**Project Name:** Lightweight Mesh Network Nodes

**Department:** Computer Science

**Focus Area:** Wireless Networking

**Product/Process:** Mesh Network Nodes



**Prepared By:**

|  |  |
| --- | --- |
| **Document Owner(s)** | **Project/Organization Role** |
| Benjamin Rivera | Project Manager |
| TomioTonoike | Senior Analyst |
| Allen Ellana | Software Specialist |
| AlfonsinTison | Documentation Head |

## Project Status Report Template

|  |  |  |
| --- | --- | --- |
| Project Name | | |
| Prepared By:  Benjamin Rivera  Allen Ellana | Date:  10/11/16 | Reporting Period:  10/08/16 to10/17/16 |
| Project Overall Status:  Currently in the testing phase of prototype of the project. Documentation about the prototype | | |
| Project Summary:  We are currently in the process of applying existing libraries and software technologies with their respective networking protocols Raspberry Pi units to create the mesh networks. We will compare and contrast these existing methods to identify and develop the ideal setup for a raspberry pi and smartphone based mesh network. | | |
| **Milestone Deliverables performance reporting over last period**   |  |  |  |  | | --- | --- | --- | --- | | **Milestone Deliverables** | **Due Date** | **% Completed** | **Deliverable Status** | | Raspberry Pi Networking software installation | | | | | * Initial Draft of Project Vision and Scope Document | 10/11/16 | 100% | [**On Schedule**] | | * Initial Draft Work Breakdown Structure | 10/16/16 | 80% | [**Ahead of Schedule**] | | Connectivity of devices | | | | | * Network throughput testing | 10/08/16 | 100% |  | | * Testing the range of the mesh network | 10/08/16 | 100% |  | | * Connectivity Between the three Pi’s | 10/08/16 | 100% | [**On Schedule**] | | | |
| **Milestone Deliverables scheduled for completion over next period**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Milestone Deliverables** | **Due Date** | **% Completed** | | **Deliverable Status** | | Indentification of Test Cases for networking the Pi | | | | | | * Formulation of Cost Estimation | 10/20/18 | 40% | | [**Ahead of Schedule**] | | * Formulation of WBS diagram | 10/21/18 | 20% | | [**Ahead of Schedule**] | | * Formulation of Project Estimation | 10/21/18 | 10% | | [**On Schedule**] | | Experimentation | | | | | | * Speed testing the mesh | 09/28/16 | | 0% |  | | * Establishing metrics | 09/28/16 | | 0% |  | | | |
| **Project impact of milestone success or failure for project remainder**   |  |  | | --- | --- | | We may have to remove the part about using multiple scenarios in deploying the different methods of deployment due to time constraints. | We will focus more on the metrics we will be identifying and compare and contrast the different existing methods by testing in only one or two simulated scenarios | | | |
| **ProjectBudget/Financial Status**   |  |  |  |  | | --- | --- | --- | --- | | **Budget Item** | **Planned Budget** | **Actual Cost** | **Variance/Explanation** | | 2 Units Raspberry Pi | 5000php | $80 | Raspberry Pi 3 are cheaper than their predecessors | | 2 Memory Cards (16GB) / 2 Charger (5W 2.5A) | 2000php | 2000php | No Variance | | | |
| **Project Risk Management Status**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Risk and Description** | **Risk Chance** | **Risk Impact** | **Risk Priority** | **Change from Last Review** | | Our hardware may not be fully compatible with all the methods of deploying mesh networks. | Medium | High | High | Same | | Weather | High | Medium | Low | Same | | | |
|  | | |
| **Project Recommendations**   |  | | --- | | * Locate a suitable location for testing * Find locate more consultants * Research on further applications once the project has finished | | | |
| **Objectives for Next Project Status Review**   |  | | --- | | By the next status review, we should be already in our testing and data collection phase.  The data collected will then be compared and contrasted among each other in order to formulate | | | |
| **Related Project Information**   |  | | --- | | None as of the moment | | | |





**Project Status Report**



**Project Name:** Lightweight Mesh Network Nodes

**Department:** Computer Science

**Focus Area:** Wireless Networking

**Product/Process:** Mesh Network Nodes



**Prepared By:**

|  |  |
| --- | --- |
| **Document Owner(s)** | **Project/Organization Role** |
| Benjamin Rivera | Project Manager |
| TomioTonoike | Senior Analyst |
| Allen Ellana | Software Specialist |
| AlfonsinTison | Documentation Head |

## Project Status Report Template

|  |  |  |
| --- | --- | --- |
| Project Name | | |
| Prepared By:  TomioTonoike  Allen Ellana | Date:  10/24/16 | Reporting Period:  10/17/16 to10/24/16 |
| Project Overall Status:  Currently in the testing phase of prototype of the project. Documentation about the prototype | | |
| Project Summary:  We are currently in the process of applying existing libraries and software technologies with their respective networking protocols Raspberry Pi units to create the mesh networks. We will compare and contrast these existing methods to identify and develop the ideal setup for a raspberry pi and smartphone based mesh network. | | |
| **Milestone Deliverables performance reporting over last period**   |  |  |  |  | | --- | --- | --- | --- | | **Milestone Deliverables** | **Due Date** | **% Completed** | **Deliverable Status** | | Raspberry Pi Networking software installation | | | | | * Research on NoSQL Database | 10/18/16 | 100% | [**On Schedule**] | | * Installation of Cassandra DB in Pi | 10/22/16 | 100% | [**On Schedule**] | | Connectivity of devices | | | | | * Testing of Cassandra DB | 10/22/16 | 100% |  | | | |
| **Milestone Deliverables scheduled for completion over next period**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Milestone Deliverables** | **Due Date** | **% Completed** | | **Deliverable Status** | | Indentification of Test Cases for networking the Pi | | | | | | * Formulation of Cost Estimation | 10/20/18 | 40% | | [**Ahead of Schedule**] | | * Formulation of WBS diagram | 10/21/18 | 80% | | [**Ahead of Schedule**] | | * Formulation of Project Estimation | 10/21/18 | 10% | | [**On Schedule**] | | Experimentation | | | | | | * Speed testing the mesh | 09/28/16 | | 0% |  | | * Establishing metrics | 09/28/16 | | 0% |  | | | |
| **Project impact of milestone success or failure for project remainder**   |  |  | | --- | --- | | We may have to remove the part about using multiple scenarios in deploying the different methods of deployment due to time constraints. | We will focus more on the metrics we will be identifying and compare and contrast the different existing methods by testing in only one or two simulated scenarios | | | |
| **ProjectBudget/Financial Status**   |  |  |  |  | | --- | --- | --- | --- | | **Budget Item** | **Planned Budget** | **Actual Cost** | **Variance/Explanation** | | 2 Units Raspberry Pi | 5000php | $80 | Raspberry Pi 3 are cheaper than their predecessors | | 2 Memory Cards (16GB) / 2 Charger (5W 2.5A) | 2000php | 2000php | No Variance | | | |
| **Project Risk Management Status**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Risk and Description** | **Risk Chance** | **Risk Impact** | **Risk Priority** | **Change from Last Review** | | Our hardware may not be fully compatible with all the methods of deploying mesh networks. | Medium | High | High | Same | | Weather | High | Medium | Low | Same | | | |
|  | | |
| **Project Recommendations**   |  | | --- | | * Locate a suitable location for testing * Find locate more consultants * Research on further applications once the project has finished | | | |
| **Objectives for Next Project Status Review**   |  | | --- | | By the next status review, we should be already in our testing and data collection phase.  The data collected will then be compared and contrasted among each other in order to formulate | | | |
| **Related Project Information**   |  | | --- | | None as of the moment | | | |





**Project Status Report**

|  |  |  |
| --- | --- | --- |
| Project Name | | |
| Prepared By:  AlfonsinTison  Allen ­­Ellana | Date:  11/20/16 | Reporting Period:  10/25/16 to11/19/16 |
| Project Overall Status:  Currently in the testing phase of prototype of the project. Documentation about the prototype | | |
| Project Summary:  We are currently in the process of applying existing libraries and software technologies with their respective networking protocols Raspberry Pi units to create the mesh networks. We will compare and contrast these existing methods to identify and develop the ideal setup for a raspberry pi and smartphone based mesh network. | | |
| **Milestone Deliverables performance reporting over last period**   |  |  |  |  | | --- | --- | --- | --- | | **Milestone Deliverables** | **Due Date** | **% Completed** | **Deliverable Status** | | Raspberry Pi Networking software installation | | | | | * Reformat RPi from Raspbian to Ubuntu | 11/14/16 | 100% | [**On Schedule**] | | * Network Reconfiguration | 11/15/16 | 100% | [**On Schedule**] | | * Creating keyspace for web app in RPi | 11/15/16 | 100% | [**On Schedule**] | | * Create collections in Apache Cassandra | 11/15/16 | 100% | [**On Schedule**] | | * Research PhilGIS and QGIS | 11/16/16 | 100% | [**On Schedule]** | | * Install QGIS | 11/18/16 | 100% | [**On Schedule**] | | * Download map vector and rasters from PhilGIS | 11/18/16 | 100% | [**On Schedule**] | | Connectivity of devices | | | | | * Creation of keyspaces | 11/18/16 | 100% |  | | | |
| **Milestone Deliverables scheduled for completion over next period**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Milestone Deliverables** | **Due Date** | **% Completed** | | **Deliverable Status** | | Indentification of Test Cases for networking the Pi | | | | | | * Installation of NLoad | 11/19/16 | 100% | | [**On Schedule**] | | * Installation of IfTop | 11/19/16 | 100% | | [**On Schedule**] | | Experimentation | | | | | | * Testing using NLoad | 11/19/16 | | 100% | [**On Schedule** | | * Tesing usingIfTop | 11/19/16 | | 100% | [**On Schedule**] | | | |
| **Project impact of milestone success or failure for project remainder**   |  |  | | --- | --- | | We may have to remove the part about using multiple scenarios in deploying the different methods of deployment due to time constraints. | We will focus more on the metrics we will be identifying and compare and contrast the different existing methods by testing in only one or two simulated scenarios | | | |
| **ProjectBudget/Financial Status**   |  |  |  |  | | --- | --- | --- | --- | | **Budget Item** | **Planned Budget** | **Actual Cost** | **Variance/Explanation** | | 2 Units Raspberry Pi | 5000php | $80 | Raspberry Pi 3 are cheaper than their predecessors | | 2 Memory Cards (16GB) / 2 Charger (5W 2.5A) | 2000php | 2000php | No Variance | | | |
| **Project Risk Management Status**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Risk and Description** | **Risk Chance** | **Risk Impact** | **Risk Priority** | **Change from Last Review** | | Our hardware may not be fully compatible with all the methods of deploying mesh networks. | Medium | High | High | Same | | Weather | High | Medium | Low | Same | | | |
|  | | |
| **Project Recommendations**   |  | | --- | | * Locate a suitable location for testing * Find locate more consultants * Research on further applications once the project has finished | | | |
| **Objectives for Next Project Status Review**   |  | | --- | | By the next status review, we should be already in our testing and data collection phase.  The data collected will then be compared and contrasted among each other in order to formulate | | | |
| **Related Project Information**   |  | | --- | | None as of the moment | | | |



**Project Name:** Lightweight Mesh Network Nodes

**Department:** Computer Science

**Focus Area:** Wireless Networking

**Product/Process:** Mesh Network Nodes



**Prepared By:**

|  |  |
| --- | --- |
| **Document Owner(s)** | **Project/Organization Role** |
| Benjamin Rivera | Project Manager |
| TomioTonoike | Senior Analyst |
| Allen Ellana | Software Specialist |
| AlfonsinTison | Documentation Head |