COMMUNITY AWARENESS SYSTEM FOR ANDROID DEVICES

PROJECT PLAN (VERSION 1.0)

Group : DEC1618

Clients : Dr Daji Qiao, Dr. George Amariucai

Advisors: Dr. Daji Qiao, Dr. George Amariucai

Team Members/Role:

Jason Wong: Team Leader
Erik Fetter: Communication Leader
Matt Gerst: Team Webmaster
Shikhar Vats: Key Concept Holder
Brad Anson: Key Concept Holder
Adit Kushare: Key Concept Holder

Table of Contents

1 Introduction

1.1 Project statement

1.2 Purpose

1.3 Goals

2 Deliverables

3 Design

3.1 Previous work/literature

3.2 Proposed System Block diagram

3.3 Assessment of Proposed methods

Option 1: Serval Mesh

Option 2: Serval Mesh integrated with wireless routers

Option 3: Custom Software

3.4 Validation

4 Project Requirements/Specifications

4.1 Functional

4.2 Non-functional

5 Challenges

6 Timeline

6.1 First Semester

6.2 Second Semester

7 Conclusions

8 Useful Resources

9 References

10 Appendices

Figure 1: GANTT Chart: Project Timeline

Figure 2: Serval Mesh with Routers Setup

1 Introduction

1.1 Project statement

The National Security Agency has expressed an interest in developing a community awareness system for Android devices. In result, Dr. Daji Qiao and Dr. George Amariucai have been assigned to research and create a working prototype of such a system through our senior design project. The purpose of this project is to provide a squad of US soldiers the ability to create a secure network using their Android devices. Each soldier would carry an Android device and have the ability to view the approximate location of their squad members and query for information from these devices. The information to be queried includes the device's battery level, GPS coordinates, accelerometer reading, and 5 second audio/video clips from the target device. The query must return the information within 10 seconds.

1.2 Purpose

This networking component of this technology could potentially be used for military purposes. For example, a platoon of soldiers could be deployed in the field with their devices all networked together. The end goal is to have the network provide a medium of transmission for vital information such as troop movement and location. Additionally, this project allows for more interconnectivity among people in remote areas. Often times remote underdeveloped areas don't have the infrastructure to support wireless communication. The high cost of deploying routers and towers and the low mobility often makes this technology not applicable/feasible. However with mesh networking, we are now able to bring a flexible network that is not only mobile but also a cost effective solution to wireless networking between devices.

The data collection portion of this project will be fed into a Biometric Engine being developed by a separate development team. The data to be collected so far includes accelerometer information, GPS location, 5 second video and audio clips, etc. The Biometric Engine will use the data collected in the app to verify the identity of the person using the phone and their surroundings. For example, audio could be processed to indicate possible problems such as gunfire, unknown voices, or used to help locate lost soldiers.

1.3 Goals

Project Goals:

1. Successful proof of concept: The main goal is to showcase a working proof of concept to the NSA and have our project move closer to being deployed to real US soldiers.

2. Soldier safety: The main purpose of our project is to give troops a way to share vital information with their fellow troops such as relative location, audio and video snippets, and other information that can be used to calculate the safety of a soldiers well-being.
3. Cheaper communications: A possible implication for our project is to allow communications in remote areas or third world countries where there is a lack of infrastructure that supports communications. Cell towers, telephone lines, and other infrastructure have high costs and having the ability to create wireless networks of communications can be a cheaper alternative.
4. Publish our application to the Google PlayStore: Another possible outcome of our project is to publish our application to the Google Playstore for the general public to use.

Learning Goals:

1. Learn Android: Few of our team members are new or inexperienced in Android development and this project would give us a good way to learn Android.
2. Learn Network Principles: Many of us are not familiar with basic networking principles that will most likely be needed to complete our project such as IPv4, TCP, UDP, routing protocols, etc. Learning these networking principles are aligned with our interests and will help us in developing a quality product.
3. Learn Team Dynamics: All of us are aspiring to be successful engineers and we recognize that learning to work in a team is an important skill.
4. Learn Agile Development: Our goal is to approach this project with an Agile development lifecycle. Many of us have learned the basic principles of Agile during our coursework but have never applied it to a long term project such as this.

2 Deliverables

We will deliver an Android application which can create and maintain an ad-hoc mesh network with other Android devices as well as have the ability to collect and distribute various sensor data around the created network. For our project, we will also create a visualization portion that will show the network topology, users in the network, and sensor data collected from those users.

The ultimate goal is to integrate our network application to allow other Android applications to utilize the mesh networking capability. In result, along with our application we will provide an API in which other applications can connect to and utilize our application’s networking capability.

3 Design

3.1 Previous work/literature

The baseline of our solution comes from a previously developed application known as the Serval Mesh. The Serval Mesh is a software designed by a team known as the Serval Project which is based in South Australia. This team developed Serval Mesh as a “revolutionary, free, open-source software for mobile telephones” (1). It was originally founded by Dr Paul Gardner-Stephen back in 2010 and aims to bring “infrastructure-free mobile communication” between Android devices (1). In short, this means finding ways to make Android devices communicate via Wifi or bluetooth.

Some helpful resources we found are as follows:

- http://developer.servalproject.org/dokuwiki/doku.php
    - The Serval Project has a website which outlines the details of their project in a very specific and helpful way
- https://github.com/servalproject
    - The Serval Project has released their software and code as open source and thus we were able to get our baseline code from their GitHub repository.

The reason we decided to baseline off this existing project is that we would not have been able to develop all the functionality that we would need within the time frame of this two semester senior design project. The Serval Project is 6 years in the making and is still in the prototype phase of their project. To generate a quality product in less than 4 months seemed unreasonable. At the same time, the Serval Project doesn’t have all the functionality that we need and would still afford us the opportunity to extend their project and put our engineering skills to good use.

3.2 Proposed System Block diagram

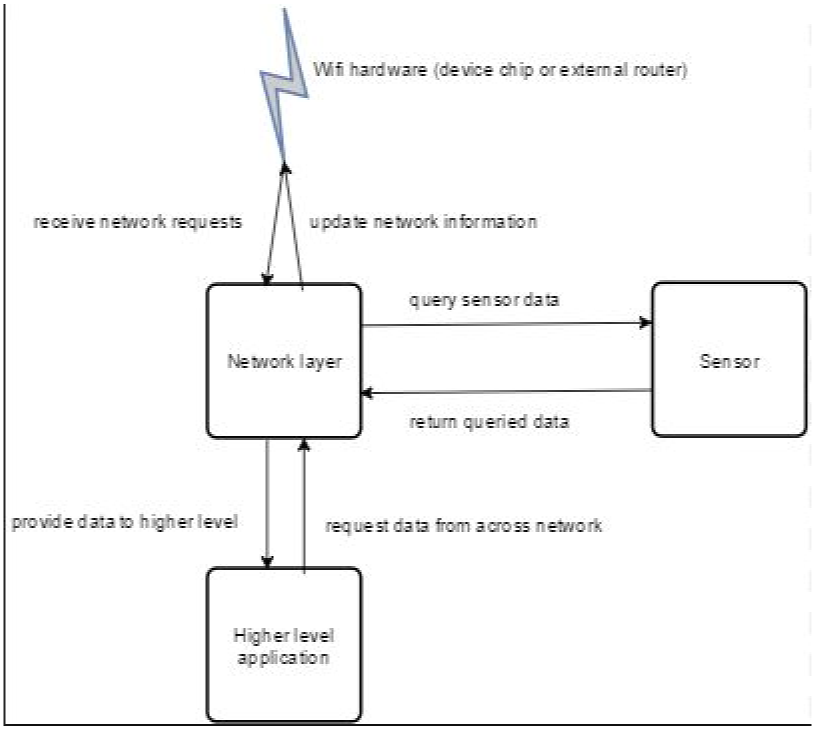
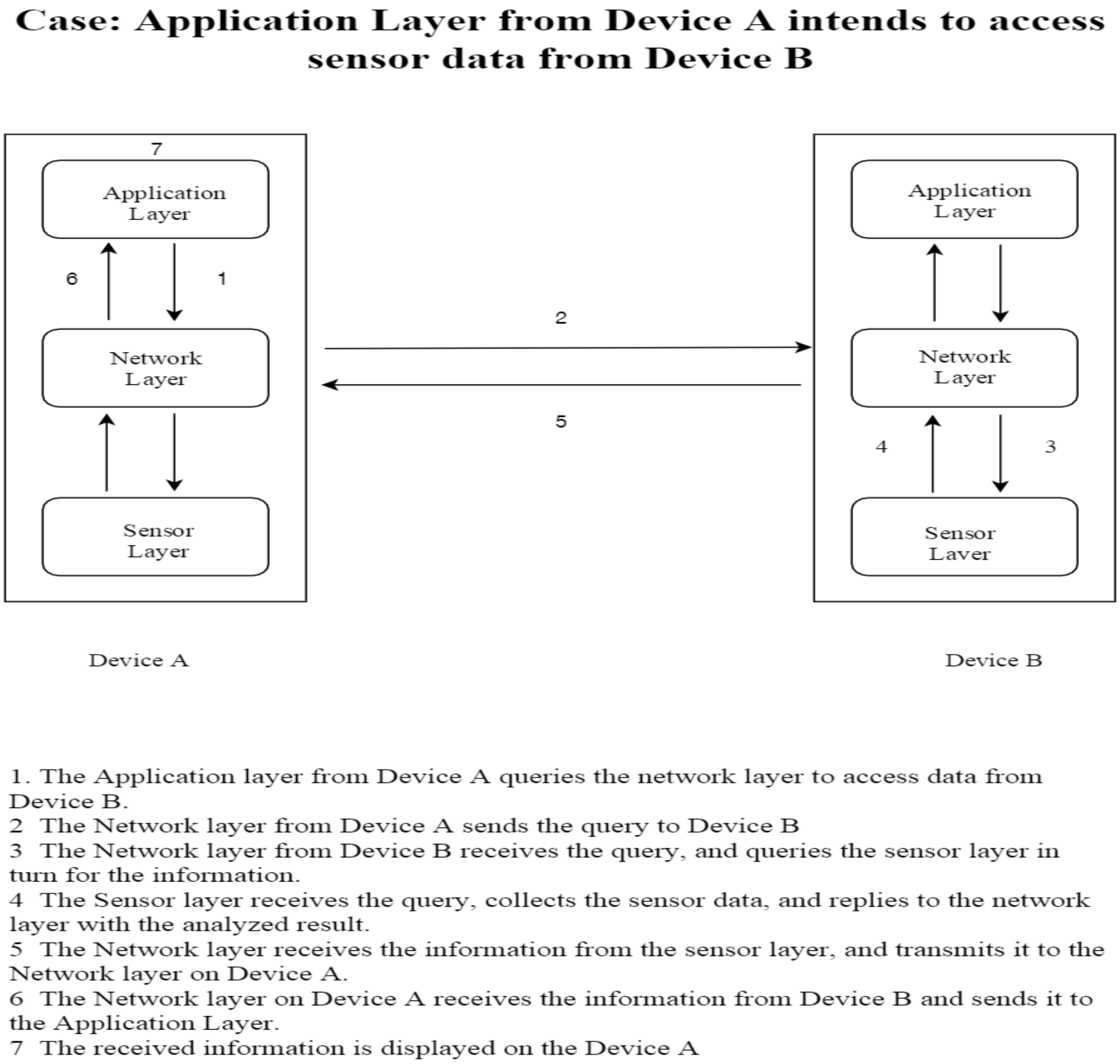

Wifi hardware (device chip or external router)
receive network requests
update network information
query sensor data
Network layer
Sensor
return queried data
provide data to higher level
request data from across network
Higher level application

Figure 1 Sample System Block Diagram

This project can be divided into three distinct interconnected system layers:

- Sensor Layer: The sensor layer essentially has functions to collect the raw data from the various sensors on the android device. This information from various sensors including, but not limited to, accelerometer, gyroscope, barometric sensors, is analyzed to generate a confidence number which would be used for communication with the other android devices. The results from the sensor layer are transmitted to the network layer which then uses the information to communicate with the other devices on the network.
- Network Layer: This layer encapsulates the core functionality of the project. The android devices in our system would be interconnected using wireless ad-hoc network mesh topology. The network layer is responsible for establishing and maintaining a connection between the android devices in the system. The higher level application layer queries information before transmitting the information, the network layer will query the sensor layer, which replies with the analyzed sensor data (see Sensor Layer). The network layer on the receiving device receives the information, and then transmits it to the higher application layer.

- Higher Level Application: This layer acts as the UI layer for the application. The flow of queries and information starts after the higher level application layer queries the network layer for information from any other device in the mesh network system.



Case: Application Layer from Device A intends to access sensor data from Device B

1. The Application layer from Device A queries the network layer to access data from Device B.
2 The Network layer from Device A sends the query to Device B
3 The Network layer from Device B receives the query, and queries the sensor layer in turn for the information.
4 The Sensor layer receives the query, collects the sensor data, and replies to the network layer with the analyzed result.
5 The Network layer receives the information from the sensor layer, and transmits it to the Network layer on Device A.
6 The Network layer on Device A receives the information from Device B and sends it to the Application Layer.
7 The received information is displayed on the Device A

Figure 2 Flow of information between two devices on the network

3.3 Assessment of Proposed methods

3.3.1 Option 1: Serval Mesh

One option is Serval Mesh which is the open source solution that currently exists which allows the wireless communication between certain Android phones.

- Main benefits of Serval Mesh
    - Creates an ad-hoc wifi mesh network with Android devices

    - Supports file sharing
    - Supports phone calls
    - Supports SMS
- Main downfalls
    - Android device must be rooted to use Serval Mesh in Ad-Hoc mode (rooting voids device warranty)
    - Android device uses a lot more energy in Ad-hoc mode
    - Serval Mesh has known problems with certain Android devices

3.3.2 Option 2: Serval Mesh integrated with wireless routers

The Serval Mesh app can operate in different modes:

1. Wifi mode where you connect to an Access Point
2. Ad-Hoc Mesh mode where the devices communicate directly

The basic premise of this solution is to use Wifi mode and have several routers communicate with each other to create the mesh network rather than having the phones talking directly. See diagram of setup in Appendix Figure 2.

Ideally, each Android device would be paired with its own portable router and connect to it like a regular Wifi Access Point. By flashing the router with custom software, the routers would create the mesh network amongst themselves.

- Main benefits of integrating routers with the Serval Mesh
    - Android device does not require to be rooted (rooting voids device warranty)
    - Android device does not require a new operating system (in this case we installed the CyanogenMod OS)
    - Works on all Android devices (several blacklisted Android devices do not support doing Ad-Hoc Mesh mode)
    - Android device uses a lot less energy in Wifi mode compared to Ad-Hoc Mesh mode
    - It gives us an extra option: the phones can do mesh networking directly, or we can have the routers doing the mesh networking
- Main downfalls
    - Each soldier would carry the portable router which also needs a power source such as a portable battery pack (the size is 2.6in x 2.91 in x 0.86in, and the weight is 0.13 pounds)
    - All phones on the mesh must use the same mode (either all Wifi mode, or all Ad-Hoc mode)

3.3.3 Option 3: Custom Software

Another choice we have is to develop a completely new solution from scratch.

- Main benefits
    - Proprietary Software
- Main downfalls

  - With our timeline this would be extremely tough to implement and still meet our deadlines
  - Team lacks experience in wifi communications

3.4 Validation

To confirm that our solution works, we will create an application that uses the system we are developing. This application will query data from the rest of the network, allowing us to show that the mesh network is working.

4 Project Requirements/Specifications

4.1 Functional

- A user should be able to find out who else is on the mesh network
- Users should be able to request information about other users on the network
- Network module should provide as many metrics, such as RSSI, if possible
- Users should be able to see a network topology, possibly organized to represent relative positions, which shows how many hops to another user
- The Application should expose both a network API usable by other applications and a sensor/inference API for the inference engine

4.2 Non-functional

- Periodic network updates should be quick without a major battery drain
- Network should be resistant to device loss by using the mesh topology
- Users should know in a reasonable amount of time if a device has left the mesh
- Remote data collection should be received in a reasonable amount of time (5-10 seconds)
- Collection of data should occur in the background, without UI interaction
- Android codebase should largely be modular and good/best practice (aside from the necessary rooting of the devices)
- The application does not need to scale very large, but it should be able to handle 10-20 devices on the network.

5 Challenges

One of the biggest challenges we face is integrating the Serval Project into a working Android app that works only at the application layer. Requiring the use of the network layer requires the user to root their device and possibly installing a new OS to ensure the correct kernel support is available. Currently we have found a workaround to this by using wireless routers that will take care of all the mesh networking and routing.

All of us are relatively inexperienced in networking and Android so the initial start of this project will be primarily research and learning the necessary skills for this project. There are some security concerns as well. Transmitting data securely to other devices can prove to be a challenge. Also since our application is mainly a service that will collect and feed information to and from other applications, finding a feasible solution for such inter application communication that is secure can be a challenge.

6 Timeline

The first working prototype of such a system is to be completed by Friday, April 29, 2016. This prototype will be used to demonstrate the basic capabilities to the NSA. The final product of such a system is to be completed by September 9, 2016 and will be in the form of two sets of APIs: one to periodically update the network overview information, and one to query for information on demand.

Because of the due date of the final product, our development will be mainly completed during the first semester. A detailed outline of our tasks and planned schedules are outlined in our GANTT chart found in the Appendix as Figure 1.

6.1 First Semester

For the first semester we will create an Android application which can establish communication between two android devices using an ad-hoc mesh network. This android app will also have the ability to collect and distribute various sensor data around the created network. For our project, we will also create a visualization portion that will show the network topology, users in the network, and sensor data collected from those users. We will also establish some ground rules/API for other applications to be able to use our app as a means of communication.

6.2 Second Semester

Second semester will be about revising and optimizing our application. Some UI improvements and beautification will likely wait until second semester, but the core functionality should already be in the application before the end of the first semester.

7 Conclusion

Our plan to develop a fully functioning data-sharing mesh-networking application seems both reasonable and comprehensive. We know we need to keep our sights on both the short and long-term, or we could face a number of pitfalls. We recognize that our goals and expectations must remain somewhat fluid, but also that there must be a certain level of rigor to them. If we remain consistent, though, throughout the project, we will likely all learn significant things throughout the process.

8 Useful Resources

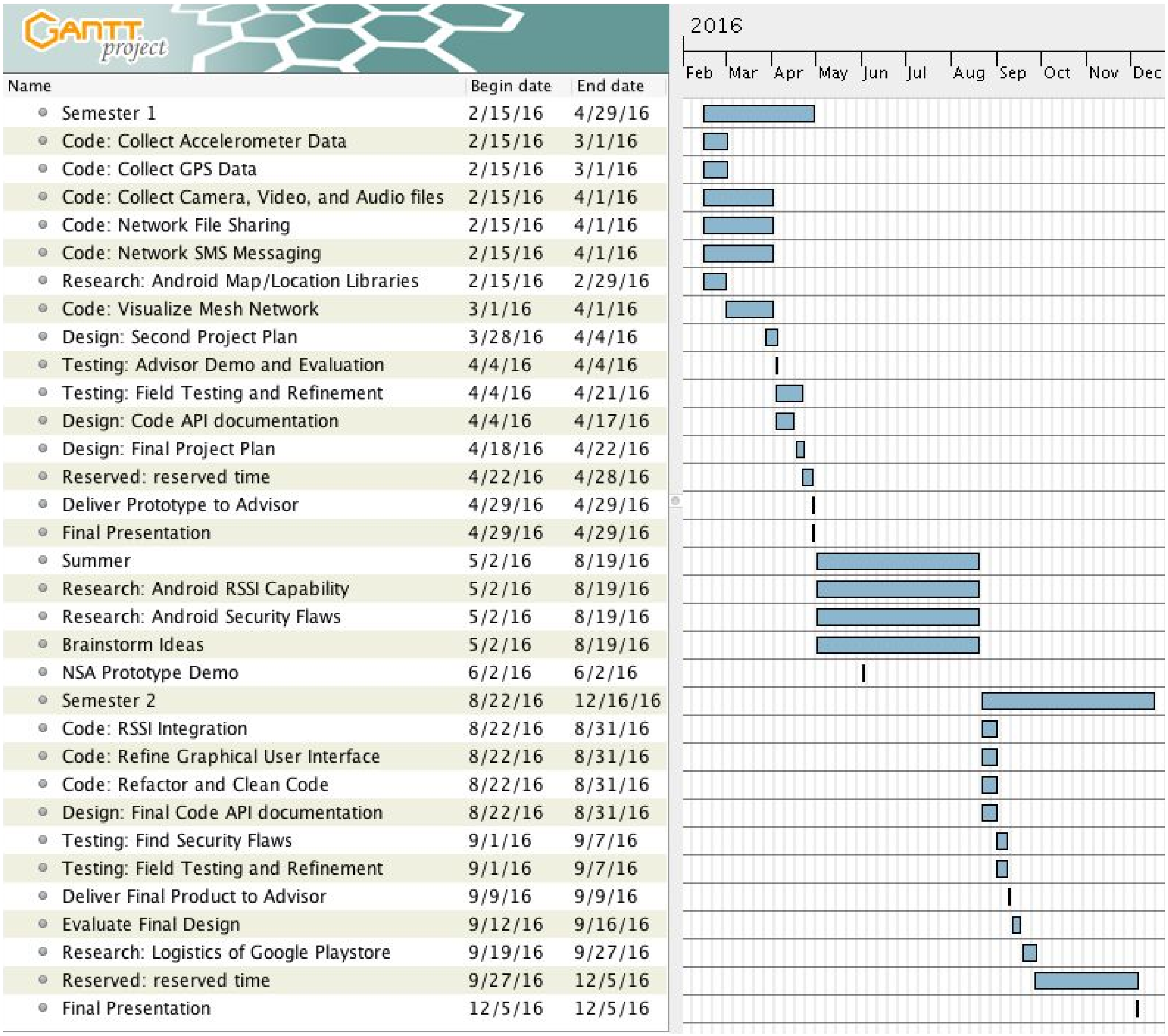
- CyanogenMod/Servel Mesh
  - Nexus 7 Rooting Instructions: http://www.teleread.com/uncategorized/how-to-root-and-mod-a-nexus-7-2012/
  - CyanogenMod Instructions: https://wiki.cyanogenmod.org/w/Install_CM_for_grouper
  - CyanogenMod Version: cm-12-20150625-SNAPSHOT-YNG4NAO09M-grouper.zip
- Mesh Extender
  - http://developer.servalproject.org/dokuwiki/doku.php?id=content:meshextender:prototyping_on_mr3020
  - http://www.amazon.com/TP-LINK-TL-MR3020-Wireless-Portable-Router/dp/B00634PLTW
  - http://www.ebay.com/itm/2pcs-RFD-900-MHz-Ultra-Long-Range-Radio-Telemetry-Modem-with-FTDI-Antenna-IN-US-/301579827973?hash=item46378efb05:g:9UQAAOSwqu9VGkS4
  - http://www.amazon.com/SanDisk-Cruzer-Low-Profile-Drive--SDCZ33-016G-B35/dp/B005FYNSZA/ref=sr_1_2?s=pc&ie=UTF8&qid=1454221838&sr=1-2&keywords=SanDisk+Cruzer+Fit%E2%84%A2+tiny+USB+memory+stick

9 References

1. http://developer.servalproject.org/

10 Appendices

10.1 GANTT Chart: Project Timeline



Name	Begin date	End date
Semester 1	2/15/16	4/29/16
Code: Collect Accelerometer Data	2/15/16	3/1/16
Code: Collect GPS Data	2/15/16	3/1/16
Code: Collect Camera, Video, and Audio files	2/15/16	4/1/16
Code: Network File Sharing	2/15/16	4/1/16
Code: Network SMS Messaging	2/15/16	4/1/16
Research: Android Map/Location Libraries	2/15/16	2/29/16
Code: Visualize Mesh Network	3/1/16	4/1/16
Design: Second Project Plan	3/28/16	4/4/16
Testing: Advisor Demo and Evaluation	4/4/16	4/4/16
Testing: Field Testing and Refinement	4/4/16	4/21/16
Design: Code API documentation	4/4/16	4/17/16
Design: Final Project Plan	4/18/16	4/22/16
Reserved: reserved time	4/22/16	4/28/16
Deliver Prototype to Advisor	4/29/16	4/29/16
Final Presentation	4/29/16	4/29/16
Summer	5/2/16	8/19/16
Research: Android RSSI Capability	5/2/16	8/19/16
Research: Android Security Flaws	5/2/16	8/19/16
Brainstorm Ideas	5/2/16	8/19/16
NSA Prototype Demo	6/2/16	6/2/16
Semester 2	8/22/16	12/16/16
Code: RSSI Integration	8/22/16	8/31/16
Code: Refine Graphical User Interface	8/22/16	8/31/16
Code: Refactor and Clean Code	8/22/16	8/31/16
Design: Final Code API documentation	8/22/16	8/31/16
Testing: Find Security Flaws	9/1/16	9/7/16
Testing: Field Testing and Refinement	9/1/16	9/7/16
Deliver Final Product to Advisor	9/9/16	9/9/16
Evaluate Final Design	9/12/16	9/16/16
Research: Logistics of Google Playstore	9/19/16	9/27/16
Reserved: reserved time	9/27/16	12/5/16
Final Presentation	12/5/16	12/5/16

Figure 3 Gantt Chart

10.2 Serval Mesh with Routers Setup

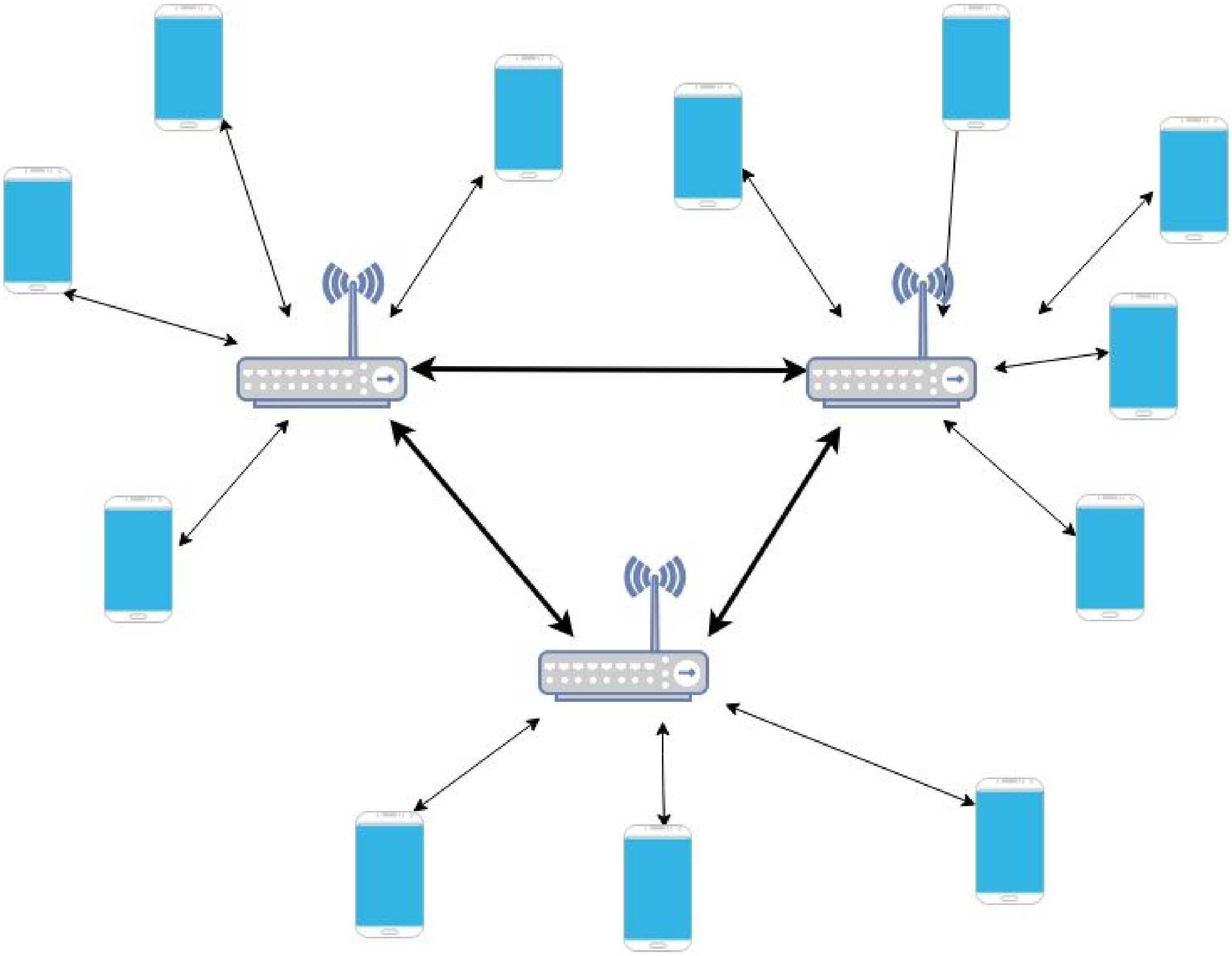


Figure 4 System setup with routers