Community Networks QoS Monitoring System

Meluleki Dube
Department of Computer Science
University of Cape Town
South Africa

David Kheri
Department of Computer Science
University of Cape Town
South Africa

Clayton Sibanda
Department of Computer Science
University of Cape Town
South Africa

Abstract

CCS Concepts • **Networks** → Network measurement

Keywords Community Networks, Data-driven Networking, Quality of Services

1. Project Description

This project will build a distributed platform for collecting and analysing Internet quality of service (QoS) metrics within the context of a wireless community network. The platform should be able to help detect sub-optimal or anomalous traffic flows, such as due to misconfiguration or network attacks.

This project is three fold: first to help community members understand the performance and utilisation of the network; secondly to help network administrators monitor and detect sub-optimal performance and lastly to help researchers manage and collect data.

To achieve this, the project will build a set of distributed Internet measurement units to gather Internet performance datasets in community networks. In addition, the monitoring system will incorporate feature selection mechanisms for detecting and correlating factors impacting network QoS.

The final output will be an intelligent QoS system that incorporates network measurements infrastructure and data processing pipeline. As a case study, this project will carried out within the context of iNethi Community Network, a localized content sharing and services platform being developed in Ocean View, a township in Cape Town.

1.1 Project Significance

Network measurement tools are now important to all users from administrators to general users. Without measurement platforms that provide a clear visualisation of network activity, administrators must scroll through long unorganized tables of data collected from the network to make sense of the activity in the network. This is an inefficient and time consuming task which deters the detection anomalous behaviour.

With a large number of people getting connected, users often want to know their network usage and their footprint on the internet. For example a user maybe interested in knowing the data they consumed over a specific amount of time, the services they use the most or how often they use the internet. Users also want this data delivered in a user friendly way hence the need for an elegant tool to visualise the data.

This project therefore aims at building a distributed platform for collecting and analysing internet quality of service(QoS) metrics within the context of a wireless community network. The platform should be able to help detect sub-optimal or anomalous traffic flows, such as due to misconfiguration or network attacks.

1.2 Issues and Difficulties

There has been industrial and academic endeavours to characterise the internet through network measurements. However none of the prominent tools have been used in a significant way in developing regions especially in wireless community networks.

One of the challenges is that community networks are vastly different from known large common Internet networks. They are implemented using cheap hardware and networking methods. Most of them also have very few physical connections within the network. This gives them a unique architecture which poses a big challenge when it comes to implementing a traditional network measurement tool [9].

Community networks are also plagued with cyber security risks as user privacy may not be guaranteed. Network measurement tools have been suspected to lead to cyber security back doors in networks.

2. Problem Statement

2.1 Aims of the Project

Community networks are decentralized communication networks which are large scale and self organized, built by the people for the people[9]. These networks provide a sustainable solution to address the connectivity gaps that exist in underserved urban, remote, and rural areas around the world[8]. Prominent Examples include Zenzeleni, Rhizomatica and Inethi which is our case study for the project.

Inethi which is a community network seeks to work with communities to co-design a content sharing and services platform for community wireless networks[14]. Its goal is to build more resilient communities by using information technologies to help its users tap into local creativity, innovation, and other resources, with an eye towards improvement of socio-economic status[14].

The aim of this project is to provide researchers and operators of the network with a tool they can use to perform measurements on the community network, also with the ability to schedule collection of network data at specified times. The project also aims to provide end users with the ability to view their data usage accurately in a very user friendly manner. All the functionality discussed previously will be collectively packaged together with a easy to use visual interface. The ultimate goal of the application is to use the data collected enhance quality of service, prevent attack on the network as well as contribute further to the research in community networks.

2.2 Requirements

Our research takes on the form of a software engineering project with members of the Inethi community network as the primary user of the application. The application needs to meet their requirements for it to be of use. Functional requirements of the application are as follows

- Allow researchers perform measurements on the network based on different user-defined fields such latency, signal strength or throughput.
- Scheduling network measurements either on demand or at specified time.
- Detect anomalies in the network due to network misconfiguration, network attacks, network failure.
- Allow users to maintain user profiles and track their network usage
- Provide data visualization to both end users in the viewing of their user profiles as well to network operators in the monitoring of the network.

Non-functional requirements for the application include:

- Should be secure in terms of the user data recorded.
- Should be reliable.
- Should be available for users all the time.
- Should give correct results and graphs.
- Should be easily usable for all different users.

Thorough testing and prototyping is paramount, to ensure we meet the requirements, and to provide Inethi with a tool that they are satisfied with.

2.3 Research Questions

In providing the above requirements, we will be answering the following questions:

- *How do different measurement collection techniques compare to each other in terms of power efficiency and network efficiency?* Where network efficiency focuses on the effects of collecting measurements on the network. Power efficiency looks into the effects on mobile phone power utilization of the measurement collection technique used.
- *To what extent does the number of nodes probed as well as the frequency at which they are probed affect network efficiency*? This focuses mainly on the amount of overhead introduced in the network when collecting measurements from the network nodes
- *Investigate the effectiveness of co-design process for network based web application interface through usability evaluation*?

3. Procedure And Methods

3.1 System Architecture

Figure 1 shows the system over, the components of the system and how everything will fit together. The lines from components are communication lines and suggest the communication direction between the comments. Following is a mapping for each of the line number and the communication information that is sent via that line:

1. Request sent from measurement initiator to orchestrator. This is the request for the orchestrator to run measurements on the network. This tool will be useful for researchers who want to carry out measurements on the network for further analysis.
2. Request sent from data visualizer to the request handler for the data that the visualizer is going to be rendering.
3. Request from orchestrator to Request handler to so that the request handler can summon the right components for starting network measurement collection process.

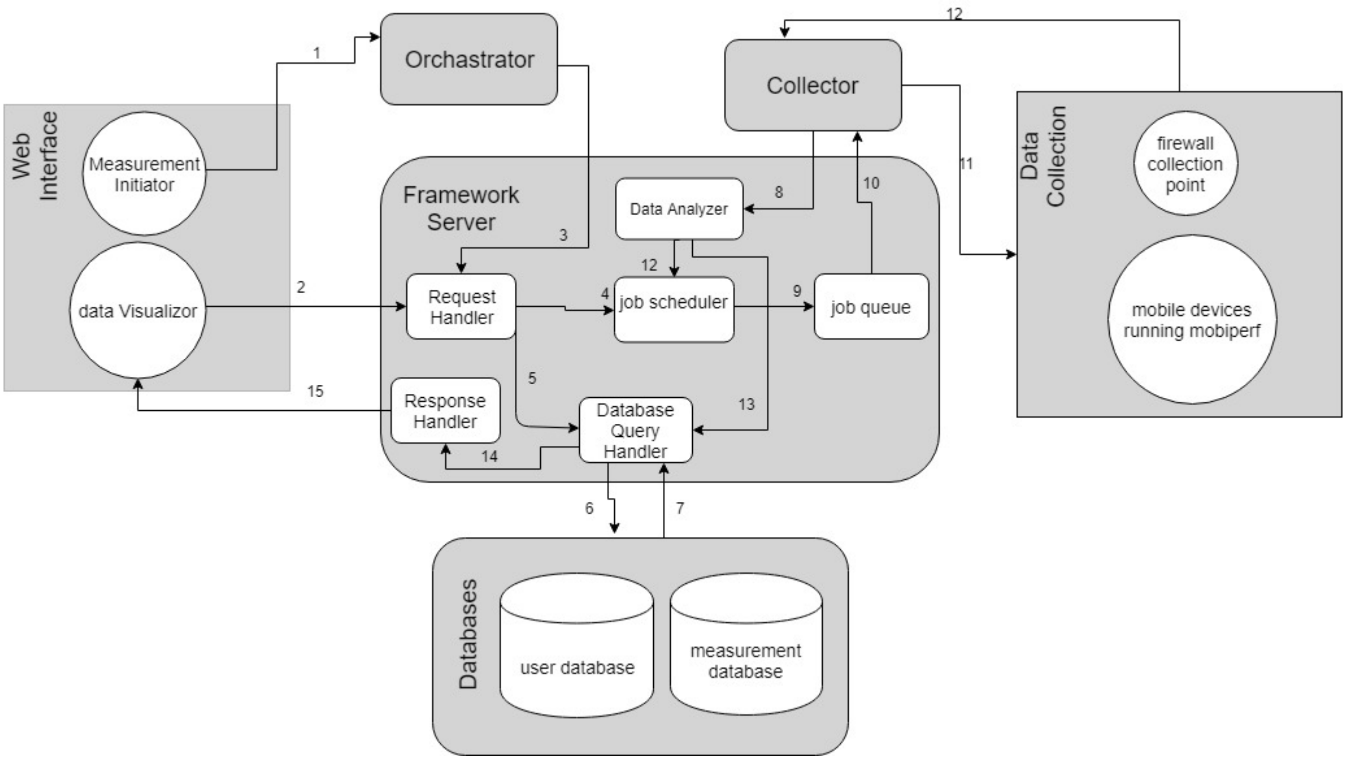

Figure 1. Showing the system overview, different components making the system and the communication directions between these components in the systems.

4. In the event that the request handler received a request to run measurement on the network this request has to be sent to a Job scheduler which will add the need to perform measurements on the job queue assuming there are other jobs that will be waiting to be done.
5. If on the other hand the request is to get data to render on the visualizer, this shows that the request handler needs to send this request to the Database querier to get the data from the database.
6. This communication link is for queries that are going to be sent t the database system. The information on the queries will specify the database to use and what type of data we are looking for.
7. This is the response from the database to be sent to the visualizer.
8. This line shows the data that comes from a data collector that need to be added to the database through the database query handler.
9. From the job scheduler, the job has to be sent to the Job queue so that when ready the job will be sent to the collector via 10 so the collector can initiate the probes to collect measurements.
10. Shows the job being transfered to the collector from the job queue.
11. The collector sends the measurement probes alerting them to start collecting measurements.
12. When the probes are done collecting measurements based on the technique used they will need to somehow send the data to the database. This line will be used to send the data to the collector which will send the data to the database via 8.
13. Sometimes when the data from the collector has arrived, further measurements may need to be taken for them to make sense, For this, the network analyzer will have to schedule a new job for collecting measurement.
14. After the network analyzer get the data it will send the data to the database via this line.
15. The Database Query handler responds back to the web interface via this line to the response handler.
16. This line is shows communication from the response handler to the data Visualizer.

The components of the system as seen from the figure 1 are:

Framework Server This is the base of the application. Other components like the orchestrator and the Collector rely on the server. The server provides communication to the database system, it also provides a means for scheduling measurements requested via the orchestrator as well as keeping track and ensuring that the jobs are sent to the collector for starting. It also consists of other tools like the request handler which is the communication point from the web interface, the job scheduler that schedules measurement collection jobs for researchers so as not to flood the network with measurement data. it also has a job queue where the scheduled jobs are kept and selected for running at appropriate times. Lastly it provided a component for handling the logic for communicating with the Databases. The reason for having only one central point for database communication is so that if we decided to change the database used, the only part of the system that needs to be modified is just this component. We also have a response handler which handles and sends results from the database to users that they are associated with in the web interface. Lastly, we will have an analyser of he network data which receives the data from the collector and performs analysis and then saves the data in the database. This part is where the machine learning algorithms can be implemented for making decisions and also adding more jobs on the scheduler.

Web Interface This consists of the Measurement initiator and the Visualizer tool. From this the users are able to view the network activities in form of graphs. The users are able to also view their profiles which will also be indicated by the visualizer. The measurement initiator is a tool for researchers who want to run network measurements on the network.

Orchestrator and Collector These are the two services that run on top of the framework server. They are responsible for accepting requests from researchers as well as initiating measurements form phones and collecting data from probes.

Data Collection points This consists mobile phones running a Mobiperf extended application. These are where we are measuring the network measurements that will be displayed by the visualizer.

Database Made up of two separate databases for storing measurement data and the other for storing user data.

3.1.1 Measurement Orchestration

In terms of orchestration, once measurements are scheduled a job will be stored on the web server with information on the measurement experiment to be run based on user-defined fields such as a number of nodes as well as parameters to measure.When the scheduled time is reached, the Server will send push notifications to online nodes with information on what parameters to be measured. Once the data is collected it will be sent to the measurement database.

3.1.2 Network Measurement Collectors

We need to develop a phone application that will be the means of collecting measurements for the internet metrics. We have decided to build an android application for the phones and tablets as Android is open and requires less resources to get started. According to [3], Android currently has 35% of the operating systems market. This shows how popular amongst users android is and thus will be a good platform to consider to build the application for first.

The application that we will be building will be extending the MobiPerf application which is built in Java [1]. Since Android Development is also done in Java it also adds to why we will be focusing on developing for android as it is easier to extend an already existing platform compared to creating one form the beginning. The mobile phone are the mobile clients shown on Figure 1. These clients will be talking to the HTTP server to send data to the database or to be triggered to collect network measurements.

3.1.3 Visualizer

To design the visualizer, a co-design process will adopted. During the co-design process, about three workshops will be conducted. A number of users from the Ocean View community will be invited to take part in these workshops. Different Human Computer Interaction methods will be employed to achieve the desired user experience. These will involve both end users and the team taking part in designing and come up with ideas. Co-design is sometimes called participatory design which is defined by [4] as a set of theories,practices and studies related to end-users as full participants in activities leading to software products. Thus the reason why we adopted this method as we want to involve users in the design process.

The Django python web framework will be used to build a visualizer web app from the ground up. The framework will be used to implement an HTTP server that will query data from either database1 or database2 as shown on the system design. Data will then be exposed in the form of API endpoints to a front end web framework that will display data in the form of graphs.

The advantage of choosing the Django web framework is that it is a simple framework that is easy to use and comes with most back-end functionalities already implemented [15]. Django also has a lot of support in the Python community, which makes it a reliable option for this project.

For the front end, the web app will leverage the react.js a JavaScript front end web framework to build user-friendly interfaces. React is a component-based library which allows rapid prototyping of web applications[12]. It also managed by Facebook hence making it a reliable and trustworthy tool to use.

4. Ethical And Legal Issues

For this project, We have identified that the legal and ethical issues to consider are associated with data collection and data storage. Most specifically, the issues arising with collecting data from user phones as well as handling the data collected from the users.

Data Collection

With Data collection, we have to worry of a number about things and these are:

- What type of data needs to be collected. This may include going to specific users and getting to what data they consider private and would not want to leave their phones. Sensitive data that will need user's permissions will include their particular location that we will need for other measurements we take to make sense.
- Since the system will need to keep track of the different applications that users will use. In attempts to measure which application used most of the data we will have to also consult first with the users to get their permission to do this. Furthermore, we will look into ways of anonymizing data collected. The aim is that when stored on the database, data will not be traceable back to the users. This solution will be adopted from the Live-Lab research project who accomplished anonymizing of data by hashing of user names of the users sending the data [17]. We will adopt the same approach and hash the messages that we are sending not the user names. The other data that users need to build their profile for network utilization statistics can be stored locally on the phone so that it is only accessible on their devices and no sensitive data leaves the users' phones.

Data Storage

We need to store data in a way that will make it almost impossible for anyone viewing the it to be able to track it to a particular user. That is to say, for each data item, no one can be able to detect the identity of a user by looking at the data. At the same time we need to know the different devices we are getting the data from so we can be able to keep track of the number of people using the system. This can be achieved by hashing the user name of the different users just before the data is saved on the database. Hashing will ensure that we have unique keys for the data we will be sending online and at the same time we can keep a count of the different keys to know how many people are using the application.

We will ensure that the user's sensitive data will not get to leave the users phone if necessary.

5. Related Works

When creating a Quality of Services monitoring system, one aspect that is needed is the measurements and collection of data from the mobile phones. Therefore, we looked at works that has been done in this regard. We looked into systems that collected data from phones and those that collected data from more complex nodes [6, 17].

Shepard et al [17] summarized the challenges involved when collecting measurements from mobiles phones. These challenges include legal issues and power consumed by these types of measurement systems. The Live-Lab systems [17] looked into eliminating these challenges by providing a systems which was capable of running internet measurements on different iPhone 3 mobile devices. We adopt the hashing of data measurement data sent from phones for ensuring anonymity of data sent from forms from the works done by [17].

From the work by [6, 11] Cassandra database was the database that was used for storing the collected for its fast loading and querying perks. We however are going to use a time series database as most of the queries that will be done from the database are going to be time based. For this reason we will be using Influx Database as the database for our system.

Other work included MONROE, which is a platform for running large-scale measurements and experiments in operational Mobile Brodband networks[6].One of MONROE functionalities is its ability to run real-time user defined measurements together with data analysis on a few selected nodes available across Europe[6]. These measurements are initiated through a user-friendly web graphical interface that remotely controlled hundreds of nodes scattered over four European countries, with the purpose of assessing network performance[6]. In the implementation of our measurement orchestration functionality we will replicate aspects of MONROE design to enable remote triggering of nodes in the community network to collect user-defined measurements at scheduled time periods.

In terms of anomaly detection, classification based systems are found to be the most accurate in the detecting intrusions in the network according to [5][7]. This is the case, due to the presence of several datasets available for evaluating network anomaly detection methods and systems for example NSL-KDD dataset [7]. They are found to be accurate on common attacks such as Denial of Service (DoS), User to Root (U2R), Remote to Local (R2L), and Probe or Scan[10]. In the case of new network attacks classification systems tend to be very ineffective as they can not accurately detect attacks they have not dealt with before[5]. Unsupervised methods on the other hand tend to perform well as they group data based on how similar they are and since normal network traffic tends to occur more than anomalous data, it can be detected relatively accurately[13].

6. Anticipated outcome

6.1 System

For the network managers and general users to see past and near-realtime performance of the network, the platform will employ a rich web based dashboard that will display performance measurement graphs for different servers on the network.

The interface will make use of different graphs to represent relevant patterns in the data. For frequency distributions, the tool will make use of histograms. Line graphs will also be used to show how the network has been performing for a certain amount of time. The visualiser will use blue and white for its user interface colours as these are simple and common colours.

The visualizer will be interactive such that users will be able to customise certain parameters like time so as to view the specific data they are interested in. Administrators will also be able to generate visual reports of the network.

The dashboard will also include an option for users to see a visualisation showing their past network usage which might include type of content being browsed, amount of data used and how often they use a certain service e.g Facebook.

6.2 Expected Project Impact

Large networks carry a lot of packets that may not be related and this makes it difficult for network managers to spot anomalous behaviours in the network[16].

Our project will include a visualiser that will clearly display data collected from the network traffic. The visualiser will employ graphs that will clearly show the administrator unusual behaviours that may be due to a potential cyber attack or a failure in the network.

For end users, the visualiser will make them aware of their present and past activity in the network in the form of graphs.

6.3 Key Success Factors

Success factors for the data visualizer:

- end-users are able to easily and effectively see their data usage in the network.
- Network operators/managers can easily visualize activity in the networks. For example number of users in the network or any point of failure.

Success factors for data collection and storage

- Accurately collect data from end user's smartphones based on different criteria such as latency, throughput and signal strength.
- Data collected is properly stored in a database and can be easily queried for display.

Success factors for orchestration tool

- Ability to schedule network measurements on demand or at a user defined time.
- Allow selection of different network metrics to run measurements on.

7. Project Plan

7.1 Risks and Risk Management Strategies

The risks for the project can be found on Appendix A. The biggest risk for the project will most likey be getting the community involved by using our application as the data that we will be showing depends on them.

7.2 Timeline

The project will run from April 2019 till 31 November 2019. The full timeline for the project is shown in Appendix B as a form of the Gantt Chart and a Milestone table.

7.3 Required Resources

The resources required are as follows:

- IDE for developing this application. For this we are going to use Android Studio [2].
- We will need a database system to store the data that we record. For this we have selected a time series database.
- We will need a server that will host the database.
- A server for the application is also needed.
- We will need users for testing the application.

7.4 Deliverables

The main deliverables for the application will be a QoS Monitoring system. This will comprise of a mobile application that is capable of collection network measurements from the end users and the firewalls. We will also have an Orchestration tool that will be useful to network administrators and researchers for scheduling and triggering network data collections from devices. Finally, we will have a visualizer tool for users to be able to have a visual idea of the network activity. The other deliverables are as follows:

- The Literature Review
- The Project Proposal
- Presentation of the Proposal
- Prototypes of the Systems
- Project Website
- A final Report

7.5 Milestones

The milestone for the project can be viewed in Appendix B together with the Gantt Chart. Milestones will include the project deliverables.

7.6 Work Allocation

The work is divided between the three people involved, Clayton Sibanda, David Kheri and Meluleki Dube. The division is as follows:

- Clayton Sibanda is working on the visualizer tool for the application. This will involve coming up with the website where users will be able to visit to see the network activity and their network activity profiles.
- David Kheri is working on the Orchestration tool that will help researchers and network administrators to trigger and schedule data collection from the end users. David will explore using this tool to confirm potential network threats.
- Meluleki Dube is working on creating the application for collecting network data and saving the measured data on a database that will be queried by different users for different reasons. This database will be queried by the visualizer that Clayton is working on and by David on when collecting the network results collected from the devices.

References

[1] M-lab. https://www.measurementlab.net/tests/mobiperf/.

[2] Meet android studio | android developers. https://developer.android.com/studio/intro.

[3] Operating system market share worldwide. http://gs.statcounter.com/os-market-share.

[4] *Participatory design: the third space in HCI.*

[5] Ahmed, M., Mahmood, A. N., and Hu, J. A survey of network anomaly detection techniques. *Journal of Network and Computer Applications 60* (2016), 19 – 31.

[6] Alay, Ã., Lutu, A., GarcÃDa, R., PeÃŞn-QuirÃşs, M., Mancuso, V., Hirsch, T., Dely, T., Werme, J., Evensen, K., Hansen, A., Alfredsson, S., Karlsson, J., Brunstrom, A., Khatouni, A. S., Mellia, M., Marsan, M. A., Monno, R., and Lonsethagen, H. Measuring and assessing mobile broadband networks with monroe. In *2016 IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)* (June 2016), pp. 1–3.

[7] Bhuyan, M. H., Bhattacharyya, D. K., and Kalita, J. K. Network anomaly detection: Methods, systems and tools. *IEEE Communications Surveys Tutorials 16*, 1 (First 2014), 303–336.

[8] Braem, B., Bergs, J., Blondia, C., Navarro, L., and Wittevrongel, S. Analysis of end-user qoe in community networks. In *Proceedings of the 2015 Annual Symposium on Computing for Development* (New York, NY, USA, 2015), DEV '15, ACM, pp. 159–166.

[9] Braem, B., Blondia, C., Barz, C., Rogge, H., Freitag, F., Navarro, L., Bonicioli, J., Papathanasiou, S., Escrich, P., Baig Viñas, R., Kaplan, A. L., Neumann, A., Vilata i Balaguer, I., Tatum, B., and Matson, M. A case for research with and on community networks. *SIGCOMM Comput. Commun. Rev. 43*, 3 (July 2013), 68–73.

[10] Buczak, A. L., and Guven, E. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys Tutorials 18*, 2 (Secondquarter 2016), 1153–1176.

[11] Casas, P., Vanerio, J., and Fukuda, K. Gml learning, a generic machine learning model for network measurements analysis. In *2017 13th International Conference on Network and Service Management (CNSM)* (Nov 2017), pp. 1–9.

[12] Gackenheimer, C. *What Is React?* Apress, Berkeley, CA, 2015, pp. 1–20.

[13] Hussain, B., Du, Q., and Ren, P. Semi-supervised learning based big data-driven anomaly detection in mobile wireless networks. *China Communications 15*, 4 (April 2018), 41–57.

[14] Inethi. Inethi community network. `http://inethi.org.za/`. Accessed April 29, 2019.

[15] McAllister, S., Kirda, E., and Kruegel, C. Leveraging user interactions for in-depth testing of web applications. In *Recent Advances in Intrusion Detection* (Berlin, Heidelberg, 2008), R. Lippmann, E. Kirda, and A. Trachtenberg, Eds., Springer Berlin Heidelberg, pp. 191–210.

[16] Ruan, Z., Miao, Y., Pan, L., Xiang, Y., and Zhang, J. Big network traffic data visualization. *Multimedia Tools and Applications 77*, 9 (May 2018), 11459–11487.

[17] Shepard, C., Rahmati, A., Tossell, C., Zhong, L., and Kortum, P. Livelab: Measuring wireless networks and smartphone users in the field. *SIGMETRICS Perform. Eval. Rev. 38*, 3 (Jan. 2011), 15–20.

Appendix

Task/Milestone	Start Date	End Date
Literature Review	15-03-19	02-05-19
Literature Review Draft	01-04-19	25-04-19
Literature Review Final	26-04-19	02-05-19
Project Proposal	28-04-19	29-05-19
Project Proposal	28-04-19	23-05-19
Proposal Presentation	27-05-19	29-05-19
Development	30-05-19	11-09-19
Paper Prototype and Brainstorming	30-05-19	07-06-19
Design and Prototype Iteration 1	07-06-19	14-06-19
Design and Prototype Iteration 2	14-06-19	21-06-19
Preparation for Deployment	08-08-19	10-08-19
Live Deployment and Testing Period	11-08-19	11-09-19
Demonstrations and Code Submission	19-07-19	31-10-19
Software Feasibility Demo	19-07-19	19-07-19
Code submission/project demo	02-10-19	31-10-19
Report	09-09-19	20-10-19
First Complete Draft of Final Report	09-09-19	09-09-19
Final Project Paper Handin	19-10-19	20-10-19
Project Media	07-08-19	11-10-19
Project Website Development	06-08-19	07-08-19
Final Website	23-10-19	11-10-19

Figure 2. Showing time frames at which tasks will be done.

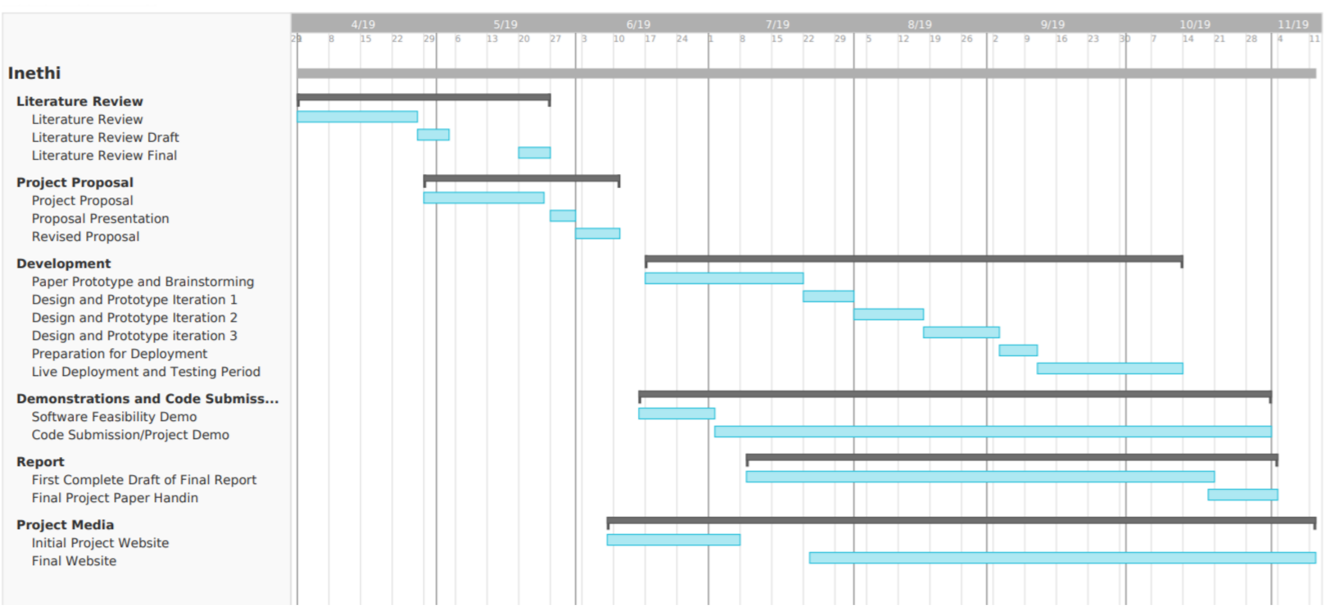

Figure 3. Gantt chart showing time frames at which tasks will be done.

Risks and Risks management Table				
Risk Number	Risk	Impact (/10)	Probability (/10)	Management
1	Network data breaches	9	5	• Make the data collected to be anonymous. • Employ cyber software only
2	Development takes longer than expect to finish	5	5	• Start development early • Divide development tasks accordingly
3	Project team loses a member	8	4	• Ensure the project is equally split among members
4	Scope of the project changes during development.	5	8	• Adopt agile development methodology
5	Stakeholders such as supervisor becomes unavailable	8	4	• Discuss the issue with the supervisor to resolve it.
6	Permissions to use INETHI community network not granted	7	4	• Talk to supervisor to use a different case study
7	Conflicts between members	6	5	• Introduce a neutral party to resolve the conflict.
8	Scope creep	5	5	• Direct all efforts on necessary features of the project

Figure 4. Showing risks and how they are managed.