UNIVERSITY OF CAPE TOWN
DEPARTMENT OF COMPUTER SCIENCE

CS/IT Honours Final Paper 2019

Title: Quality of Service Monitoring Tool

Author: David Kheri

Project Abbreviation: QOSMON

Supervisor(s): Josiah Chavula

Category	*Min*	*Max*	Chosen
Requirement Analysis and Design	*0*	*20*	20
Theoretical Analysis	*0*	*25*	0
Experiment Design and Execution	*0*	*20*	0
System Development and Implementation	*0*	*20*	20
Results, Findings and Conclusion	*10*	*20*	10
Aim Formulation and Background Work	*10*	*15*	10
Quality of Paper Writing and Presentation	*10*		10
Quality of Deliverables	*10*		10
Overall General Project Evaluation (*this section allowed only with motivation letter from supervisor*)	*0*	*10*	
Total marks		**80**	**80**

Measurement Orchestration In Community Networks

David Kheri

Department of Computer Science
University of Cape Town
South Africa

April 2019

Abstract

In the quest to provide a sustainable solution to address connectivity gaps in rural and remote areas of the world, community networks have been proposed by many as the key to bridge these connectivity gaps. In this paper, we discuss the research done on community networks, different approaches used in the obtaining of network measurements as well as how we can leverage this in the creation of a measurement platform with the aim of ultimately enhancing performance and security in community networks. In this paper we mainly focus on design and implementation of the orchestration component of the platform and how it facilitated measurement collection on the network.

CCS Concepts •**Networks** → *Network performance analysis*

Keywords Community Networks,Measurement Orchastartion, Network Measurements

1. Introduction

Community networks can be referred to as large-scale, self-organized and decentralized networks built with the aim of meeting the communities' communication needs[8].Community networks are usually found in very underserved and remote areas, where network access from commercial network operators is limited to non-existent[8].Considering the fact that technology is advancing at a very rapid rate, there still remains big connectivity gap in urban, rural, remote areas of many developing countries which leaves them at a great disadvantage[7].

In communities where network access is present, network operators usually lack the tools required to accurately give information of what is happening in the network thus making it very difficult to isolate points of failure in the network and troubleshoot[6]. This has prompted research in different methodologies for network measurements. For example MONROE project focused on performing measurements in Mobile Brodband networks[4], LiveLab focused on measuring wireless networks using smartphones as an apparatus for network measurements[16].

Despite the different challenges experienced when performing network measurements, the data collected plays a crucial part in the improvement of performance and security in the network[**?**].

1.1 Project Aims

This project's primary aim is to create a measuring platform that will enable researchers to perform different kinds of measurements on community networks such as Inethi and Zenzeleni, which can be used to further enhance the design and performance of such networks and in turn provide a better experience to its users.

On top of creating a measurement platform, we aim to provide users of the community network a clear summary of their data usage, which will help them be more conscious of where most of their data usage lies and thus help them manage it.

1.2 Structure of Report

This report firstly introduces community networks and the different monitoring systems deployed on such networks.The report is then split into two main sections, Requirement analysis and Design and how the orchestrator component of system was developed and implemented.

2. Background

Community networks are decentralized communication networks that are large scale and self organized, built by the people for the people [15].

Community networks have taken different shapes and forms from when they originally emerged in the late 90s[12]. The rise of commercial high speed broadband networks has rendered some of the community networks obsolete in the areas they operated while others have prospered especially in the rural areas where connectivity to high speed internet is limited to non existent[12] [12].

2.1 Community Networks

Technically, community networks are built in a decentralized manner [8] implying that the nodes in the network are not dependent on a single server point, which in turn makes the network highly scalable and prevents bandwidth issues as no operational bottlenecks can be caused due to no single server point[12]. At the heart of community networks are network nodes(also known as routers), these devices either relay network data or provide network access[12]. The links between the network nodes can be wired(DSL,fiber), wireless(WI-FI,cellular), both wired and wireless where they complement each other so as to ultimately provide network access to the end user[12]

Currently there are many community networks but considering how informal and unpopular some of them are its impossible to list them all but the following are some of the community networks:

- South Africa:Inethi community network that is deployed in Ocean View and Masiphumelele regions in Cape Town[10] and Zenzeleni community network found in the Eastern Cape region[18]
- Spain:Guifi.net is a very large network comprising of greater than 20,000 Nodes and more than 24,000 links[5]
- Greece:Athens Wireless Metropolitan Network comprises of more than 2500 nodes[12]

The Networks above are fully operational but still experience some challenges in their operation, preventing further expansion of community networks. For example, community networks usually use wireless connectivity as it is very cost efficient when trying to establish large scale networks. However these provide a great challenge since absence of cables will require extensive planning especially in dense suburban areas as channel allocation among competing users in the shared network becomes difficult to achieve correctly[8]. Also Standardization of community networks is a major challenge as it greatly hinders further growth and sustainability of the present community networks[12].

3. Network Monitoring System

In a world of increasing connectivity, it is simply not sufficient to create a network[17]. Having a holistic view of the network is essential in the pursuit of optimal performance and reliability within the network[4]. Monitoring of the networks entails observation of the network with the aim of detecting and recovering from failures in the network, which is very crucial for service providers and network operators whenever the need to determine specifically which link or router within the providers network is the cause of poor application performance or total loss of service[9]. Surveillance of the network is not only critical in the provision of a smooth user experience to the end users, but also influence, decisions on the design of future network infrastructures through the collection and analysis of the data collected[4].

However, experience from the real world can attest that no single method to monitoring that can highlight every problem in the network, as well as no single factor that can explain all failures and shortcomings in the network. This makes it very difficult to decide what metrics to be used in the monitoring of the network[4].

Measurements have to be performed so as to efficiently and accurately monitor a network, as seen by the shift towards deployment of a number of measurement platforms in the quest of providing network operational support[6]. A number of applications have been implemented to provide a clear picture of the network performance eg., Speedtest by Google, Speedtest by OOKLA or running a special measurement application[11]. Drawbacks from applications such as Speedtest is that the measurements taken lack both scalability and repeatability as well as no means of providing a guarantee on the accuracy and availability of the collected metadata eg.,type of device, which operating system its running, location information[4].

Given the drawbacks of some of this applications, further research has been done on following different methodologies on network measurements.

3.1 MONROE

MONROE is a tool used to measure and assess mobile brodband networks[14]. Availability of 4G and 4G+ technologies coupled with the increase in number of mobile phones, has led to the explosion of mobile brodband networks[4]. This advancement has prompted the importance of monitoring these networks, with the aim of properly assessing their performance and reliability[4].

MONROE aims to conduct repeatable, independent, multi-homed, large-scale measurements and experiments in operational mobile broadband networks[14]. MONROE platform comprises of nodes scattered over four European countries(Norway, Sweden, Spain, Italy) and a backend system that collects the measurement results[13].

Through the collection of data from operational mobile broadband networks, it enhances fundamental characteristics of mobile broadband networks and the relationship between popular applications and performance of the network[4]. This has proved to be quite beneficial in the provision of feedback on the design of the upcoming 5G technologies as well as improving user experience for mobile users currently still using 3G/4G technologies[4].

3.2 LiveLab

LiveLab is a Methodology employed in the measuring of wireless networks using smartphones, this was mainly motivated by the fact that there is a significant increase in the number of smartphones in the world and continuing to rise rapidly[16]. It involves leveraging mobile users as a network sampling tool through the collection of data from smartphone users thus providing a more accurate picture of network upon analysis of the data[16].
This methodology did experience some practical challenges:

- Mobility of the user caused a significant variation in network quality by the users, which in turn led to a diversity of user experiences in the network[16]
- Privacy as well was also a serious issue as users were afraid of some of the privacy concerns that would occur as a result of logging smartphone activity which was resolved through the anonymizing of the data collected[16].

Despite its practical challenges, LiveLab's main strength was its ability to provide in-field programmability of their network sampling devices as that provided researchers with the ability to update it and schedule a new measurement like you would do with a lab computer[16]. This feature is quite useful as it makes measuring of the network very dynamic and in turn adaptable to different scenarios.

4. Requirement Analysis and Design

Requirement gathering, analysis and design is a very important phase in the establishment of clear goals and objectives that is to be expected of the system to be developed. In this stage we gathered requirements from the prospective users, analysed them and designed a model of the system.

4.1 Objective of the System

The main objective of the system is to provide a holistic view of what is happening in the network from the network administrators perspective as well as providing a framework that will allow researchers to orchestrate measurements on the network.The system as well should enable users to keep track of their data usage on the network.

4.2 Requirement Gathering

4.2.1 Usability Study

In the pursuit of creating software that will be useful to the Inethi community, we organized a meeting with members of the community representing network managers as well as a regular users who are among the users we plan developing the software for.

We developed a mock-up of the user interface of the prospective system and explained to each participant what features we planned on implementing. Through this we collected their feedback on several aspects of the system based on the mock-up indicated below.

- What features did they find most useful to them
- How hard it was to find things on the Application Interface
- Did the System provide them with enough information as a user
- How effective will this platform be useful to Inethi community as a whole

Through the feedback gathered from the usability study, it provided us with guidance on the design and decision making processes towards a correct solution.

4.2.2 Communication with Supervisor

Throughout the project duration we had constant communication with the project supervisor(who is also one of the users of the Inethi community network) through various channels such as email and regular meetings. Through this approach, system development progress would be discussed and any concerns during development would be addressed and in turn this provided us with confidence that the system to be developed would meet the communities' requirements.

4.3 Functional Requirements

For this project to be a success, the following list of functional requirements have to be performed by the system.

- Allow researchers perform measurements on the network based on different user-defined fields such latency, signal strength or throughput.
- Scheduling network measurements to run at a specific time on a user defined number of nodes in the network.
- Detect anomalies in the network due to network misconfiguration, network attacks or a network failure.
- Provide nicely visualized network data usage to end users of the system.

- Provide nicely visualized data of network measurement results to both network managers as well as researchers

4.4 Non-Functional Requirements

On top of the set of functionality the system has to perform, the following set of requirements provide a specification on how well the system function

- Should ensure security of users data as well preventing any ties of the data to particular users if a hack does happen
- Should be reliable, that is provide a guarantee of little to no system downtime
- Should ensure accuracy of the network data usage of each end user as well as the experimental data recorded by a researcher or network manager.
- should provide a user friendly user interface to the system

4.5 Use Case

In this section we will be outlining descriptions on how users will be performing tasks on the system from their point of view which is pivotal in the establishment of the complexity of the system. The following are the use cases of the system.

4.5.1 Check Network Data Usage

Actor : Regular User

Preconditions : User is logged in to the system

Postconditions : User receives total bytes uploaded and downloaded distributed across different mobile applications

4.5.2 Generate Statistics on Data Usage in the Network

Actor:Network Administrators

Preconditions : User is logged in the system

Postconditions : User obtains a record of data usage in the network across different time periods(weekly,monthly)

4.5.3 Schedule Network measurements

Actor : Researchers, Network Administrators

Preconditions : user is logged in the system, indicate what kind of measurement they want to run indicate how many nodes they want to run measurements on, indicated start and end time, indicated interval between which the job will run again(0 if non-recurring measurement).

Postconditions : user receives a success response indicating that the measurement is scheduled

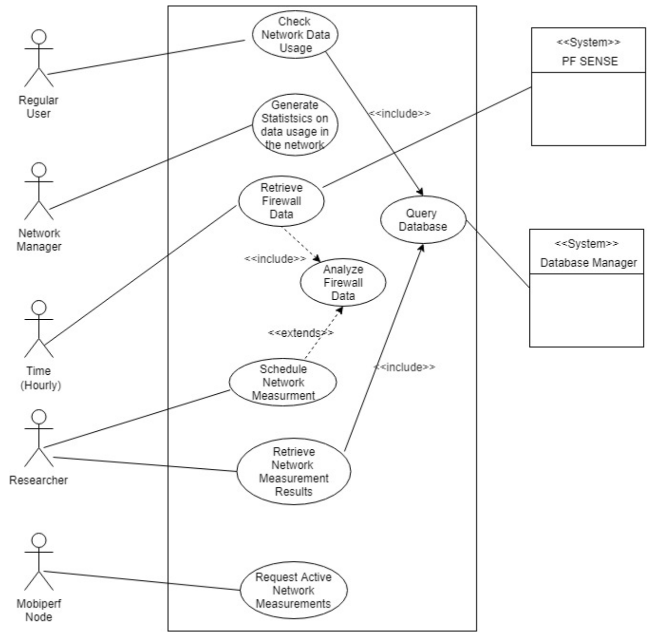

Figure 1. Use Case Diagram for Network Monitoring System

4.5.4 Retrieve Network measurements

Actor : Researchers, Network Administrators

Preconditions : user is logged in, user has scheduled measurement(s) previously.

Postconditions : User receives results of the measurements they scheduled

4.5.5 Request active network measurement

Actor : Inethiperf Node

Preconditions : None

Postconditions : A list of active scheduled measurements are sent(if any)

4.5.6 Retrieve Firewall Data

Actor : Time(Hourly)

Preconditions : Firewall data is collected by PF-SENSE

Postconditions : Hourly firewall data is retrieved and ready for analysis for any anomalies in the network

4.6 Sequence Diagrams

In order to have an accurate system design, the interaction logic between the different entities in the system has to be examined and in the order that the interactions are happening.

The two main flows that will be evaluated are

- Scheduling Network Measurements as depicted by the Figure 2

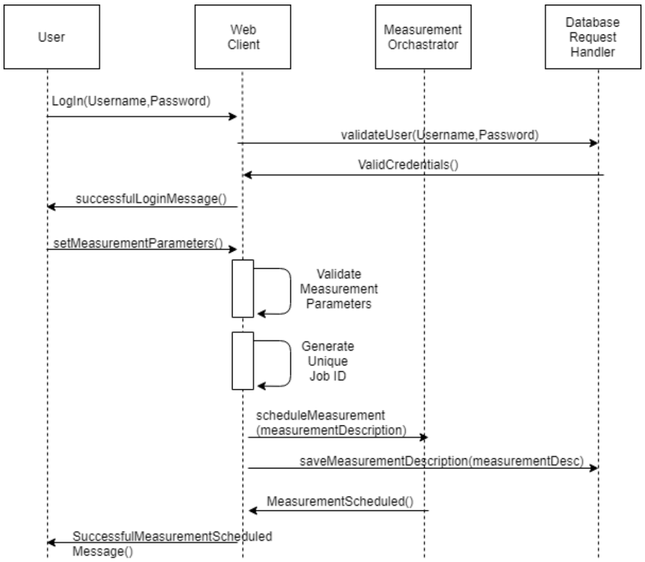

Figure 2. Sequence Diagram of scheduling of network measurements

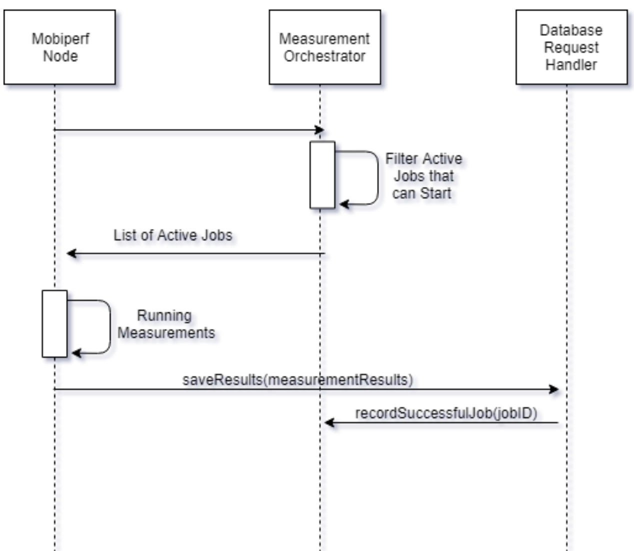

Figure 3. Sequence Diagram of Inethiperf Nodes requesting active jobs

- Checking for Active Measurements by a Inethiperf Node as depicted by Figure 3

5. System Development and Implementation

5.1 System Overview

Through analysis of the requirements and functionality that is to be delivered, a suitable system was developed as shown on Figure 4 consisting of 3 main components

- Web User interface : This is the main way users will interact with the system so as to use its functionality
- Orchestrator : This component is responsible for coordinating how user-defined measurements are run in the network.It receives input via web interface and sends active results to the mobile nodes in the network
- Database Request Handler: This component is responsible for receiving requests to read and write results or user data to the Databases

5.2 Network Orchestration

In order to fully understand how orchestration in the network is performed, this section will discuss mainly how a user-defined measurement is scheduled, Inethiperf(the software used in collection nodes in the network), how the collection nodes get active jobs from the server and finally how data collected from the firewall triggers measurements in the network.

5.2.1 Inethiperf

Inethiperf is an adaption to M-Lab's open source application Mobiperf for measuring network performance on mobile platforms. Inethiperf runs on mobile phones in the collection of network metrics such as DNS Lookup and TCP throughput(both uplink and downlink).

The following are the set of parameters Inethiperf requires in performing of measurements

- Type : which indicates the kind of measurement to be done
- Key : this is a unique Job ID which is used to tie a job result and the measurement scheduled
- Start time : this indicates the start time at which the measurement can run
- End time : this indicates the time that after which the measurement should not be run
- Count : this value dictates how many times the same job should be run on a phone
- Priority : this indicates the priority level of a particular job among the rest. The higher the value the higher the priority
- Parameters : this indicates specific fields pertaining to a particular measurement example if its a TCP throughput test it would indicate if its upload or download as well as for measurements such as Ping or DNS lookup a target endpoint would be declared here.

5.2.2 Type of Measurement Jobs

During scheduling of measurements there are two types of jobs that are dealt with

- Recurring Jobs : These are the kind of jobs that usually span over longer durations and can be repeated

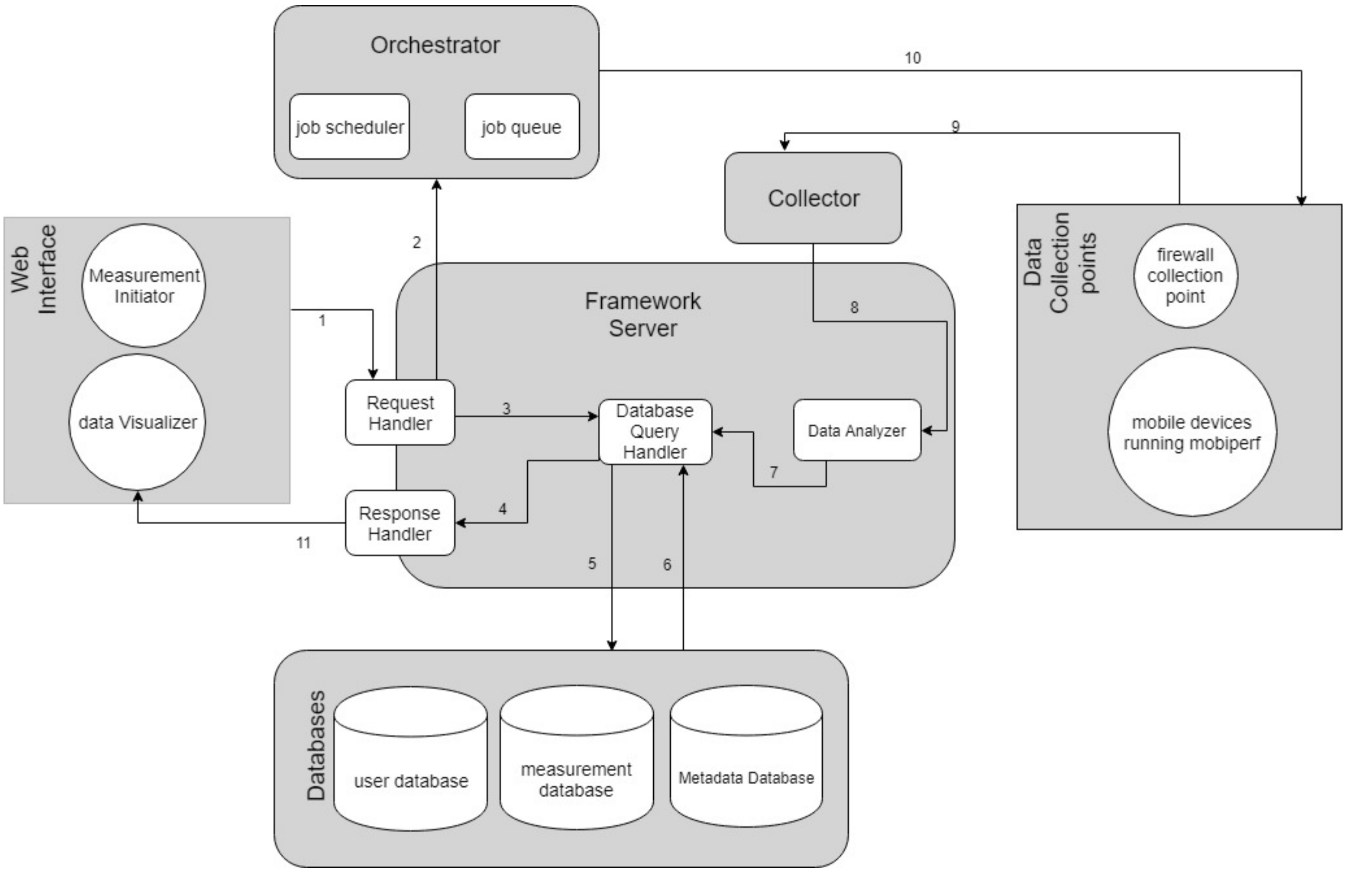

Figure 4. Showing the different components of the system and how they interact.

at regular interval.For example a job, can be scheduled on Monday 12PM and ends next week Monday same time but can happen in say intervals of 24 hours.This implies that once Monday's measurement requirements are satisfied, it will be inactive until 12PM the next day.

- Non-Recurring Jobs : These are the Jobs that are no longer active after the measurement requirements are satisfied.

To be able to distinguish these two kinds of jobs, a job interval field is added to the job description received from the web client, which provides more information in the differentiating of the two. A job interval of zero indicates that the user does not want the job to be repeated at regular intervals and any other value greater than zero in the job interval field qualifies the job as a recurring job.

5.2.3 How are Jobs Scheduled

Using all the provided job metadata such as node count, job interval, a job instance is created and added to a Job Array List on the orchestrator and mainly keeping track of fields such as a job's successful node count and its next reset time if the job is recurring. The next reset time is calculated using the start time and job interval which acts a period of the measurement.

During checkin(Inethiperf nodes querying the server for jobs) only jobs that satisfy all the following conditions, are handed out to the nodes

- The Job will be handed out if the current time is after the jobs start time implying it can start.
- The job is not removable, implying that job has not elapsed(current time is not after its end time). Also if the job is not recurring, its node count is not reached.
- The job is not resettable implying that if the job is recurring and its node count is not reached or current time is not after its next reset time.

Whenever a job successfully runs on the nodes, the result is sent back to the server for recording and success response is acknowledged by the orchestrator and the jobs successful node count is incremented.

We decided not to store jobs that had a longer lifespan but shorter intervals on the nodes. This was done in order to keep the mobile application as lightweight as possible by reducing its memory usage on the users mobile phones to prevent any inconvenience it might create.

5.2.4 Job Tracking

Jobs that are no longer active, need to be either removed or reset based on the jobs state. Periodically, a scan of all the jobs is performed and any inactive jobs that need to be removed or reset are done so accordingly.

5.2.5 Concurrency Issues

During updating of a job's node count there is a potential to lose updates, since multiple phones might try to update the same job at the same time and if not properly coordinated some increments might be lost. To counter this problem an atomic Integer was used so as to enforce the incrementing procedure is done atomically and thus ensuring no update is lost and successfully captured.

Also Read Write Locks were used to prevent possible race conditions which could happen as a result of a simultaneous read(during check in from the mobile nodes) and write(either adding of a new job or deletion of a removable job), or multiple writes happening simultaneously. This approach was used over synchronizing entire objects as that would greatly hinder performance as it would make the code run more serially than parallel. It is also anticipated that more reads(phones checking in for active jobs) than writes will happen and Read Write Locks are optimized for such scenarios.

5.3 Development framework and Technology

From the beginning of the project, currently accepted development methodologies were used. In this section the methodologies employed are discussed.

5.3.1 Programming Language and Framework

Development of the Orchestration module was done using Java. This was used due to its rich support of libraries providing the necessary functionality required. The orchestrator which is part of the back-end server can together be aggregated into a jar file and run on any compatible machine making it highly portable. Maven was used as the build system ,Git was used for version control and the code is hosted on Github

5.3.2 Libraries used During Implementation

Libraries mainly used during development process were

- Java's Utility package to provide Date utility functions such as formatting and parsing dates, ReentrantReadWriteLocks to ensure concurrency is well coordinated and the Array list in the storing of submitted jobs.
- External JSON library was used extensively to convert to and fro JSON objects.

5.4 Software Development Methodology

For this project, we employed an agile approach towards the development process. This approach emphasized on development process to be done in iterations and was selected as it optimizes the development process towards projects whose scope and requirements are constantly changing as it was to be expected.

5.5 Testing

In terms of Testing, several phones were loaded with Inethiperf software and multiple jobs(both recurring and non recurring) were submitted to the server and the mobile nodes checked in periodically and collected measurements as expected. Inactive jobs were reset and removed appropriately and the results collected from jobs sent to the mobile nodes were ready for viewing from the web interface.

6. Results and Findings

This section highlights results from the experiments conducted when the software was deployed on a testing environment within the Eduroam network.

6.1 Check In

During check in mobile nodes used approximately 1.6MB of data in three days, this includes the JSON data sent out to the server to indicate a check in as well the JSON array of active jobs received from the server.The total number of jobs received from the server was 210 during this time period.

6.2 Running of Measurements

From the 210 jobs received 20 failed due to network disconnections and 190 Jobs run successfully. These jobs consisted of Ping, DNS Lookups to multiple targets as well as measuring TCP uplink and downlink. The data consumed from this experiment cumulatively summed to approximately 5MB in three days.

7. Conclusions

In this paper, we discuss community networks and how they can be monitored and measured. We also discuss how requirements were gathered and how the system specifically the orchestrator was designed and implemented.Also from the experiment conducted it was observed that Inethiperf is very lightweight interms of data consumption

8. Future works

8.1 Deployment of Non-User mobile nodes

At the moment all measurements requested to be performed are run on the phones of users of the Inethi network. This approach has its drawbacks which mainly

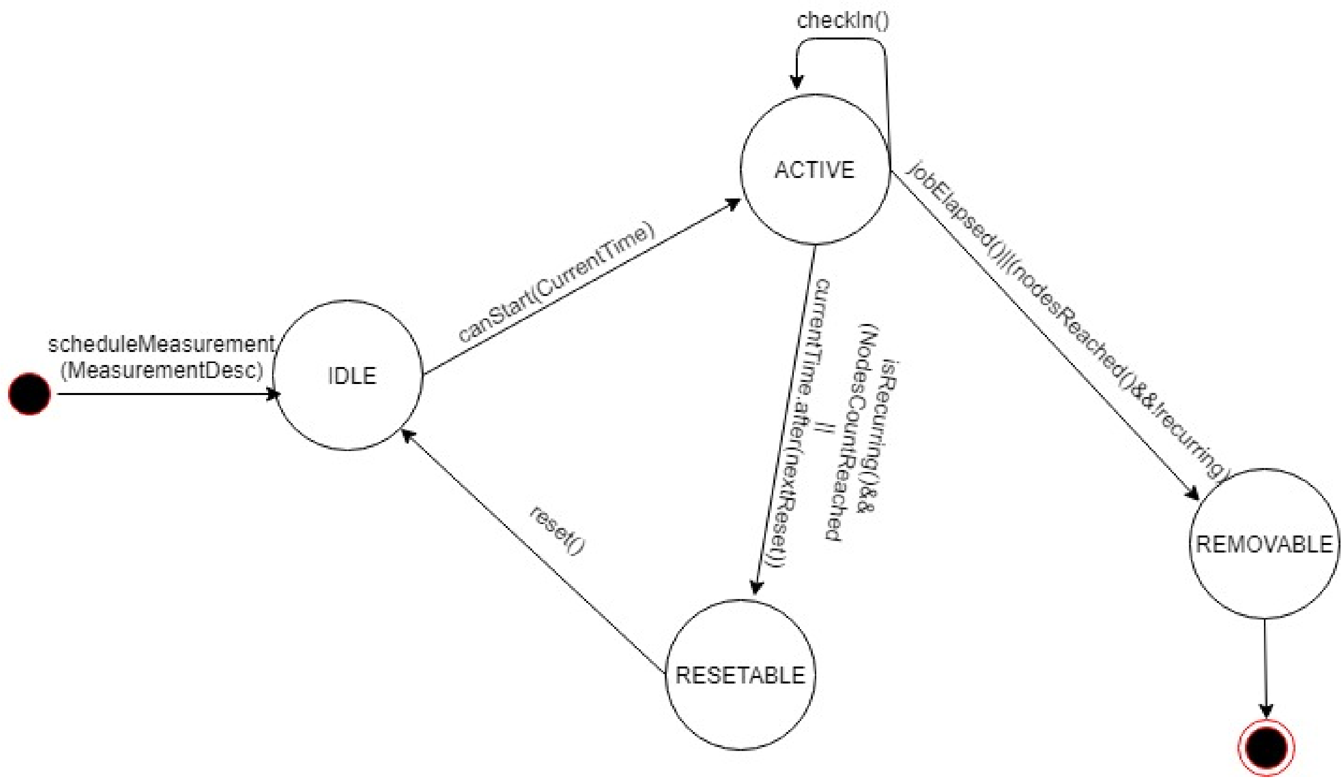

Figure 5. Showing Different states of a Job instance.

limits the frequency of the measurements run due to preventing battery drain from the users device as well as failure to pinpoint the exact location the measurements are performed due to privacy concerns. Through installation of non-user mobile nodes we can guarantee measurements will always run as requested as well as accurately map a measurement result with the location it was performed.

8.2 Use of Machine Learning on Analysis of Firewall Data

Firewall data that is currently collected could be intelligently analysed so as to enhance security in the network through anomaly detection in the collected Pcap files. This can be accurately done for common security attacks such as Denial of service and User to root as large labelled datasets such as the NSL-KDD could be used to accurately train the model[1].

References

[1] Ahmed, M., and Mahmood, A. N. A novel approach for outlier detection and clustering improvement. In *2013 IEEE 8th Conference on Industrial Electronics and Applications (ICIEA)* (June 2013), pp. 577–582.

[2] 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.

[3] Ahmed, M., Mahmood, A. N., and Islam, M. R. A survey of anomaly detection techniques in financial domain. *Future Generation Computer Systems 55* (2016), 278 – 288.

[4] Alay, ., Lutu, A., García, R., Peón-Quirós, 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.

[5] Baig, R., Roca, R., Navarro, L., and Freitag, F. Guifi.net: A network infrastructure commons. In *Proceedings of the Seventh International Conference on Information and Communication Technologies and Development* (New York, NY, USA, 2015), ICTD '15, ACM, pp. 27:1–27:4.

[6] Bajpai, V., and Schönwälder, J. A survey on internet performance measurement platforms and related standardization efforts. *IEEE Communications Surveys Tutorials 17*, 3 (thirdquarter 2015), 1313–1341.

[7] 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.

[8] 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.

[9] Calyam, P., and Swany, M. Research challenges in future multi-domain network performance measurement and monitoring. *SIGCOMM Comput. Commun. Rev. 45*, 3 (July 2015), 29–34.

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

[11] Kreibich, C., Weaver, N., Nechaev, B., and Paxson, V. Netalyzr: Illuminating the edge network. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement* (New York, NY, USA, 2010), IMC '10, ACM, pp. 246–259.

[12] Micholia, P., Karaliopoulos, M., Koutsopoulos, I., Navarro, L., Baig Vias, R., Boucas, D., Michalis, M., and Antoniadis, P. Community networks and sustainability: A survey of perceptions, practices, and proposed solutions. *IEEE Communications Surveys Tutorials 20*, 4 (Fourthquarter 2018), 3581–3606.

[13] Schwind, A., Seufert, M., Alay, ., Casas, P., Tran-Gia, P., and Wamser, F. Monroe: Measuring mobile broadband networks in europe.

[14] Schwind, A., Seufert, M., Alay, ., Casas, P., Tran-Gia, P., and Wamser, F. Concept and implementation of video qoe measurements in a mobile broadband testbed. In *2017 Network Traffic Measurement and Analysis Conference (TMA)* (June 2017), pp. 1–6.

[15] Selimi, M., and Freitag, F. Towards application deployment in network clouds. In *Proceedings of the 14th International Conference on Computational Science and Its Applications — ICCSA 2014 - Volume 8584* (Berlin, Heidelberg, 2014), Springer-Verlag, pp. 614–627.

[16] 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.

[17] traviskeshav. Survey of network performance monitoring tools.

[18] Zenzeleni. Zenzeleni community network. `http://zenzeleni.net/`. Accessed April 29, 2019.