IST-Africa 2020 Conference Proceedings
Paul Cunningham and Miriam Cunningham (Eds)
IIMC International Information Management Corporation, 2020
ISBN: 978-1-905824-57-1

Network Monitoring System for Network Equipment Availability and Performance Reporting

Baphumelele MASIKISIKI, Siyabulela DYAKALASHE and Mfundo Shakes SCOTT
Department of Computer Science, Private Bag X 1314,
University of Fort Hare, Alice,5700, South Africa
Email: bmasikisiki@ufh.ac.za, sdyakalashe@ufh.ac.za, sscott@ufh.ac.za

Abstract: The culmination of the digital world has been the need for computer networks increasing daily, because they enable communication between two or more connected peripheral devices and sharing of resources. However, the increasing demand of computer networks may be chaotic and a bit challenging at times, and that can affect the rate of performance of the network. When such chaotic events occur on the network they might make it slow, cause a network and unreachable and also be exposed to lot of vulnerabilities. Over the years network specialists have been working on finding ways of reducing such events and one of such ways was the development of Network Monitoring Systems (NMS). A NMS can be defined as a computer network's systematic effort to detect slow, such as failing routers, switches and any other network devices. They were developed as means of improving the process of manually checking for faults on the network, it does this by frequently providing adequate information that the network administrators can use to analyze, monitor, troubleshoot and configure network devices. This paper presents a prototype implementation of Network Equipment Availability and Performance Reporting at the University of Fort Hare. Campus area network was used as the testing platform.

Keywords: Network Management Protocol Java Version 4 (SNMP4J), Simple Network Management Protocol (SNMP) Object Navigator, Network Availability, Network Performance and System Automation

1. Introduction

A network is a collection of computers and devices interconnected by communications channels that facilitate communication and allows sharing of resources and information among interconnected devices. It is critical for the network to provide the highest levels of availability and performance to the end-users in favour for more productivity. To ensure that computer networks perform well and the users are able to do their work without any delays, continuous monitoring is required. When dealing with large networks that are used in big companies and in institutions of high learning it becomes very complex to manually configure the network. Therefore, it is safe to say some of the network equipment can be mistakenly left out during the monitoring and that could result into faulty network devices which can cause the following on a network:

- Network devices to be unreachable
- Network to be slow and congested
- Packets to be dropped or lost
- Network to be exposed to lot of vulnerabilities.

The University of Fort Hare (UFH) consists of many labs that are being used by students to perform their academic work. However, some of these labs were designed in such a way that they can accommodate both undergraduates and postgraduates. Even though the labs were designed for accommodating students, but the design plan of these

labs can sometimes be a problem to the network due to the number of students who occupy the labs. Sometimes the network can be over-utilized and that could result into slow performance. Additionally, it is possible to find some of the students busy downloading movies, games software's that have huge file sizes and in such cases the network will perform slow. Since the labs are connected using Ethernet cables some students usually remove the cables and place them in their own computers and then leave labs without placing the cables back to their location. When such cases occur the network in the computers turn to be unavailable and perform poorly.

This research work therefore undertaken the development a system that will monitor and detect faults on a network. This system focused on network availability and network performance, the researchers believed that this development could bring solutions to the challenges faced by the network administrator s and lab technicians of UFH as case study. In terms of network performance, the research focused on Central Processing unit (CPU) utilization and memory as a baseline for monitoring. The system was successfully developed using technologies such as Simple Network Management Protocol (SNMP), HyperTerminal, Object Navigator and Java Standard Edition (JSE). Therefore, this paper discusses the abridged developed NMS for network equipment availability and performance reporting. Generally, such a system produces customized reports that present different network results on an hourly basis, weekly and monthly basis.

1.1 Network Monitoring Systems (NMS)

Network management refers to the broad subject of managing computer networks and telecommunication. It has been known for keeping track of network devices and making it possible for a network to be available to the users at all times. The history of NMS can be traced back to the study of Network Management. However, NMS are essential in running the complex computer networks and they ensure all faults on the network are known and assist the network operator in fixing these faults. Moreover, NMS can monitor traffic, latency, performance, availability and other network problems. They are meant to improve the process of manually monitoring and troubleshooting network devices and also notify network administrators (via email and SMS) in case of outages [2]. NMS are an important technique for improving the availability, reachability, and performance of a network.

1.2 What Can Be Monitored

It is crucial to capture the current status of network devices, such as routers, switches and critical network servers. The table below provides some of network aspects that need to be critical monitored on a network.

Type	**WHAT TO MONITOR**	**WHY TO MONITOR**
Hardware	Availability of network devices (such as switches, routers, Servers, etc.)	Inability monitor hardware health makes identifying issues (high CPU load, excessive heat, power fluctuations, etc.) challenging
Software	Percentage of your routers' maximum throughput utilized on average.	The whole network doesn't have to be down to have a negative impact; loss of email, HTTP, or FTP server availability.
Traffic	Availability of all critical services on your network.	To ensure availability and smooth operations on a computer network.

Table 1: Reasons for monitoring

Table 1, presents computer aspects that needs to be constantly monitored. In order to make sure that the network is available it is very judiciously to continuous monitor

connected network devices. Furthermore, to avoid any sorts of abnormalities that might affect the network performance it is also good to monitor network traffic.

2. Developmental Objectives to Address

The main aim of this research was to develop NMS for network equipment availability and performance reporting with UFH being used as a test platform. This project initially focused on a small environment where switch, router and computers were deployed for testing. After testing in a small network environment the testing scale then moved to a live Campus Local Area Networks (LAN) such as the Computer Science Department network. Additionally, the system was developed to monitor network devices and send automation reports on the operational status of specific a network device in terms of hourly based, weekly based and monthly status. The system also aimed at having an ability to detect worst performing network devices (i.e. a device that always has a network failure). Furthermore, the system was developed produce customized reports and that was achieved with help of JfreeCharts. Lastly an ability of collaborating network configuration management and SNMP to monitor network devices was also a success.

3. Methodological Approach

The success development of the system was mainly based on the interviews that were conducted from the network administrators. Short questionnaires were created to conduct interviews. Furthermore, qualitative method and quantitative method were also used as a way of information gathering. To gain more knowledge and to address research question qualitative method was used. However, in terms of data coding and scoring quantitative techniques were applied. Lastly for data analysis and results presentation JFreeCharts was used, JfreeCharts is open-source that is used to generate charts.

3.1 System Development Techniques

For system development Iterative model was used, which is the process that starts with a simple implementation of a subset of the software requirements and iteratively enhances the evolving versions until the full system is implemented [5]. This model gave the developer an ability to revisit the phases that were already developed to fix the loopholes.

Figure 1, represented a graphical phases of the iterative model. During the system development the system was developed based on the phases iterative model. For example, for system requirements which is phase one of the model interviews were conducted at the University ICT Department to get a gist of the challenges that they are facing based on the scope of the research work. Moreover, during phase 3 (coding and unit testing) this model gave the developer an ability of redoing of the functionality of charts.

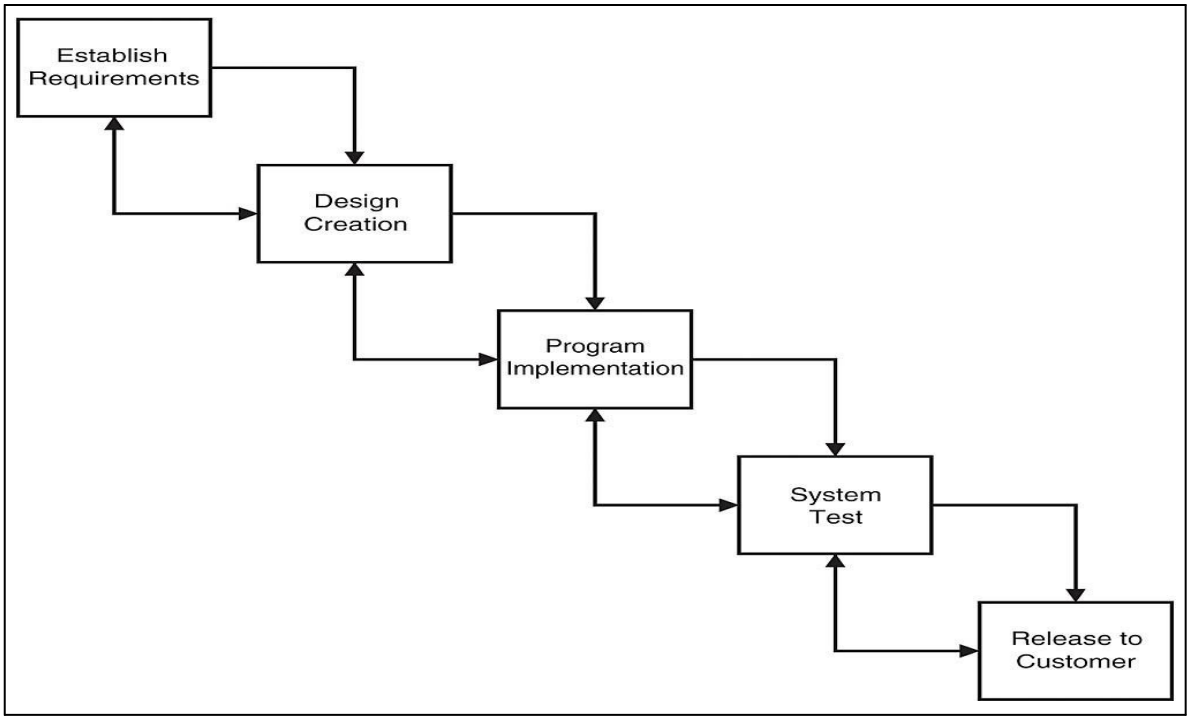

Figure 1: Iterative Model Structure [5, 11]

4. Developmental Technology

The system was successfully developed using technologies following technologies:

- SNMP
- Object Navigator
- Java Standard Edition (JSE).

4.1 Simple Network Management Protocol (SNMP)

SNMP is an application layer protocol that facilitates the exchange of management information among network devices, such as nodes, switch and routers [6]. It is a protocol that is widely used in network management for monitoring a network [6]. Additionally, SNMP consists of three components SNMP managed device, SNMP agent, and Management Information Base (MIBs) [8, 12]. SNMP manager is a computer that is configured to poll SNMP agent for information, it can be any machine that can send query requests to SNMP agents with the correct credentials [8, 12]. Whereas SNMP Agent is responsible for gathering information about the local system and storing them in a format that can be queried. MIB is a hierarchical, pre-defined structure that stores information that can be queried [6]. However, for this research, SNMP was used to get a status and information of a network device. SNMP commands were used to tell SNMP agent to go gather information about the configured device. In summary SNMP help the developer to communicate with network devices and store all information that was required about the device in MIB for later usage.

SNMP deals with two sorts of devices, the managed and unmanaged devices. Unmanaged devices do not have the capacity of being analysed by network management protocol, whereas managed devices has a collector agent. Figure 2, is a graphical representation of SNMP process functionality, where the developer was able to sends a quest into SNMP Manager and the Manager sends that request into SNMP Agent then SNMP Agent gathers information about the devices and then send back the response about the device back to the developer.

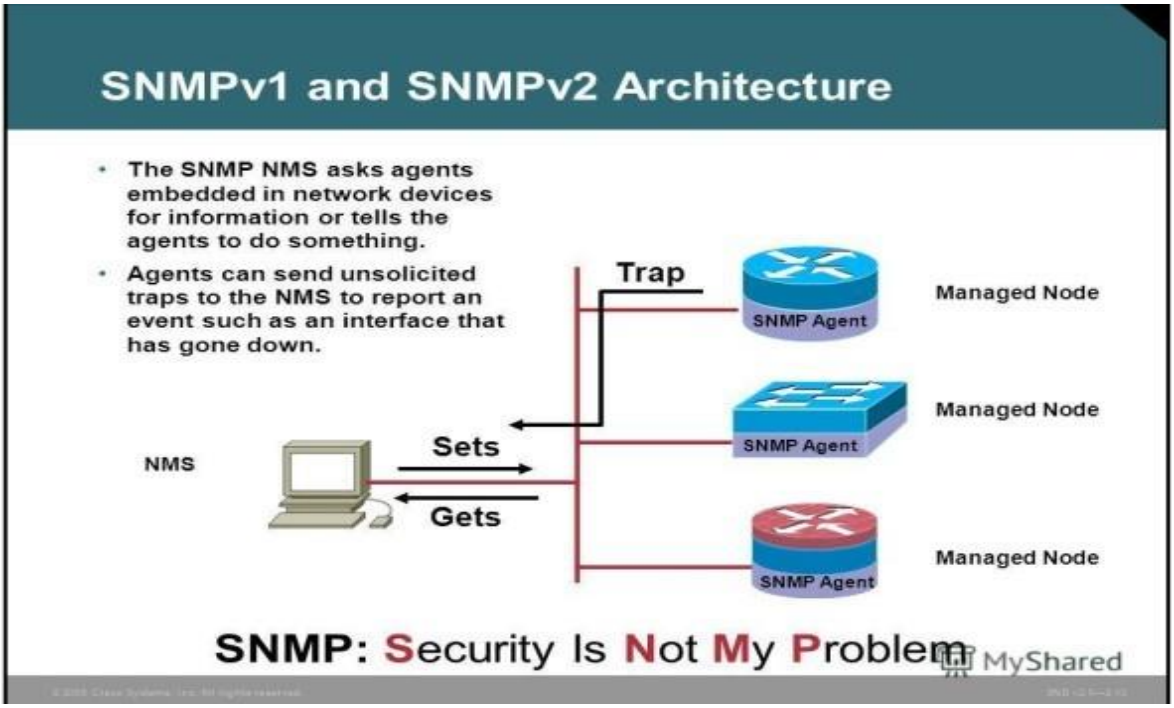

Figure 2: SNMP Architecture [8]

4.2 SNMP4J

SNMP4J is a Java API that provides some classes in order to implement a SNMP Agent and Manager [7]. It is an enterprise class free open source SNMP implementation for Java SE. SNMP4J provides two set of classes, BaseAgent which can be subclassed to build custom agent, and TestAgent which is an executable class that extends BaseAgent [7, 13]. It also provides classes and interfaces for creating, sending, and receiving SNMP messages [7, 13]. With these upstanding benefits SNMP4J was used by the developers to send request to SNMP agent about network devices and receiving SNMP messages. During the implementation of the system SNMP4J was used for connecting SNMP functionality with Java programing language. The quest when send using java portion of code together with SNMP4J java libraries.

4.3 SNMP Object Navigator

When SNMP agent was sending back responses about a certain device they were in a form of numerical numbers and these numerical number are called object identifiers. Object identifiers are basically a unique identification of a certain device and a way of showing that a device has different type of traps. Even so these number were readable, but still they were needed to be translated into a proper meaning. SNMP Object Navigator was an online software that was used to translate the meaning of object identifies. When SNMP agent returned traps to NetBeans integrated development environment (IDE) the developer had to first go online to translate each trap then store the trap into a database. This was done because it was going to challenging for the network administrators to read a database that is full of numbers and it is very easy to forget numerical number than word format.

Figure 3, is the online interface of SNMP Object Navigator that allow a user to input object identifier and then be translated into all the crucial information of a certain device including a type of a trap that occurred in a device and also a brief description about the trap.

Figure 3: SNMP Object Navigator [7]

4.4 JFreecharts

JFreeCharts is a free Java chart library that makes it easy for developers to display professional quality charts in their applications [15]. For results analyses and results presentation JFreecharts was used. This was selected mainly because the system was developed using java programing language. With the help of java portion of code JFreecharts was able to retrieve information from MySQL database in order to generate charts.

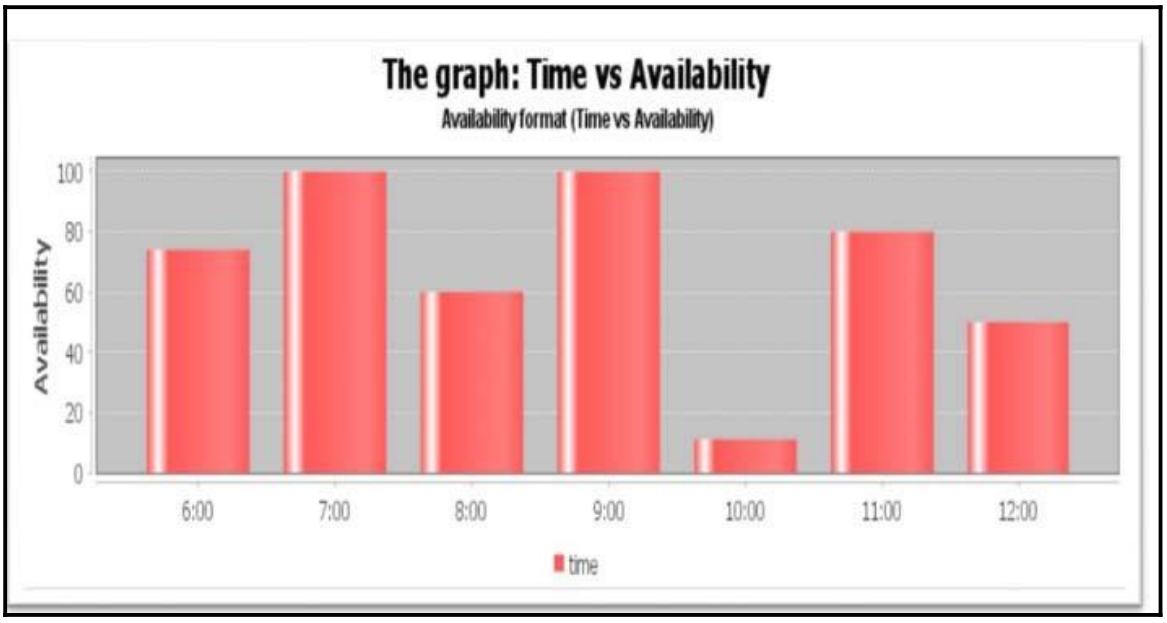

Figure 4: Availability vs Time graph

Figure 4, is the graph that was created using JFreecharts, this graph presents the status of network availability hourly. For example, at 7 the network was working very well but at 10 the level of network dropped mainly because around 10 the students are busy doing their academic work and the labs turn to be full around this time.

4.5 JavaMail API

One of the objectives was to send notifications every time a fault occurs on a network, which Java was achieved using Java mails. The JavaMail API is an optional package

standard extension for reading, composing, and sending electronic messages. [14]. since the system was developed using java standard edition JavaMail was a suitable framework because it supports java standard edition and java enterprise edition. For sending an email message it gave the developer an ability to create a session, creating and filling a message, and also an ability to configure Gmail Simple Mail Transfer Protocol (SMTP) server. Lastly, for reading mail it gave the developer an abilty to connect the usernames and password with a mailbox using java programming language.

5. Development Prototyping

NetBeans was selected as development environment mostly because the system was developed using Java standard edition. NetBeans is an open-source integrated development environment for developing with java, Hypertext Preprocessor (PHP) and other programing languages [10]. More over SNMP was used for gathering information about the network devices, but in order for the developer to be able to send quest to SNMP and receive feedback about the device to Java SNMP4J was used as a way of connecting SNMP with java. In a nutshell, the core of the system development was mostly based on the functionalities of the three components of SNMP. The aim was develop a system that will enable these three components to communicate with each other and return feedback to the client via java platform. SNMP4J provided some java classes that helped to implement the functionalities of SNMP Agent and SNMP manager. SNMP read access on managed object was triggered by using GET, GETNEXT and GETBULK. The developer used these three operation for polling SNMP request operations from the Agent.

```
FIRST PART
public static void main(String[] args) throws IOException {

    TrapReceiver snmp4jTrapReceiver = new TrapReceiver();

    //snmp4jTrapReceiver.listen(new UdpAddress("192.88.2.5/162"));
    snmp4jTrapReceiver.listen(new UdpAddress("172.5.52.4/162"));



}
SECOND PART
  public synchronized void listen(TransportIpAddress address)
          throws IOException {

    AbstractTransportMapping transport;

    if (address instanceof TcpAddress) {

        transport = new DefaultTcpTransportMapping((TcpAddress) address);

    } else {

        transport = new DefaultUdpTransportMapping((UdpAddress) address);

    }
```

Code Snippet1: Trap Receiver

Figure 4, is just a brief presentation of how the implementation of SNMP queries and responses was done. This piece of code is a code listens to SNMP agent and it is meant to detect traps that are occurring and then report them if they occurred. The Internet Protocol (IP) address was used for differentiating the devices and also port numbers were used.

6. Testing and Results

The law of normal distribution states that given random and independent samples of N observations the distribution of sample means is normal and unbiased regardless of the size of N [16]. With reference to the law of normality during the system testing the sampling number of N was equal to 4. The system was tested by 4 network administrators of the department of computer science. Based on the proposed expected results even if the N size of network administrators was above 4 the result would still be normal and unbiased. Additionally, network administrators were asked to focus more on the system accessibility, usefulness of the system, system responsive and system recoverability from the errors.

6.1 Availability Testing

In terms availability, users tasted availability they were able to get the expected results, the system was able to tell when they link is up or when they network status changes, when the cable is removed, the reasons for a network to change and also report the port numbers of interface. Basically network availability was tested using the following:

- Up /Down thresholds
- Removal of a network cable
- Rebooting Device

6.2 Up/Down threshold

For testing this threshold two set of commands we used, which are "no shutdown" and "shutdown". For up threshold (no shutdown) we obtained the following traps:

- (Interface0/0 up/ or interface0/0 changed from down to up).
- Whereas for down threshold (shutdown) we obtained the following traps: (interface0/0 administratively down/down)
- These two types of traps show which network interface is available or not available.

```
Router (config) #interface fastEthernet0/0
Router (config-if) #ip address 17.25.12.3 255.255.0.0
Router (config-if) #no shutdown
```

6.3 Removal of an Ethernet cable

An Ethernet cable was also used for testing availability, during system testing, physical removal of Ethernet cables from interface ports was applied, and just after removing the cable the system will generate a "lost carrier" trap.

6.4 Rebooting Device

Rebooting was also applied as a way to test if whether the system will respond to cold start. This was done by shutting down the network device, then trap that was returned format was (System Uptime) which simple measure the time a device has been working and available.

6.5 Performance Testing

System performance was tested based on two aspects, CPU utilization and memory utilization. CPU utilization was tested based on the two types of threshold:

- Rising Threshold
- Falling Threshold

6.6 CPU Rising

A rising CPU utilization threshold specifies the percentage of CPU resources that have exceeded the maximum configured threshold. The monitored network devices are configured with a value that is regarded as maximum threshold value for CPU raising, for example, the value for rising threshold would be 80 and 40 for falling threshold. If the returned percentage of the rising CPU utilization is beyond the specified threshold value, the network device was then regarded as being over-utilized.

6.7 CPU Falling

A falling CPU utilization threshold specifies the percentage of CPU resources that have. Went below the configured threshold. The monitored network devices are configured with a value that is regarded as minimum threshold value for CPU falling, for example, the value for falling threshold would be 40. When obtained a percentage of the falling CPU utilization that was below 40 %, then the network device was regarded as being under-utilized. Even though memory and CPU were tested the same way, but they had different usage measurements, the memory usage was measured using the following:

- Memory Threshold Notifications
- Memory Reservation

6.8 Memory Reservation

Memory Reservation is the memory that is reserved for critical operations and also ensures that management processes, such as event logging, continue to function even when router memory is exhausted. Whereas Memory threshold notifications is divided into two parts Low Free Memory Threshold and Reserving Memory for Critical Notifications [9]. Low free memory threshold is used configure a network device to issue notifications when available memory falls below a specified threshold, whereas Reserving Memory for Critical Notifications is used when a network device is overloaded by processes and when the amount of available memory might fall to levels insufficient for it to issue critical notifications [9]

In summary the obtained results were mainly based on these following questions:

- Q1: Is the system able to generate availability traps?
- Q2: Is the system able to generate memory repots?
- Q3: Is the system able to generate CPU reports?
- Q4: Is the system able to send notifications about the traps that occurred?
- Q5: Is the system able to generate availability charts?
- Q6: Is the system able to generate CPU charts?
- Q7: Is the system able to generate memory charts?

After the questionnaire analyses the obtained results were then presented in the following table:

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
Learnability	0	0	1	0	2
Efficiency	0	0	0	3	0
Satisfaction	0	0	2	1	0
Error Tolerance	0	0	3	0	0

Table 2: Usability Testing

Table 2, portray that in terms of learnability the users found the system to be

informative. However, in terms efficiency all users found the system to be efficient and also in terms of user satisfaction the system found to be neutral. However, in terms of error tolerance all users found the system to be also neutral.

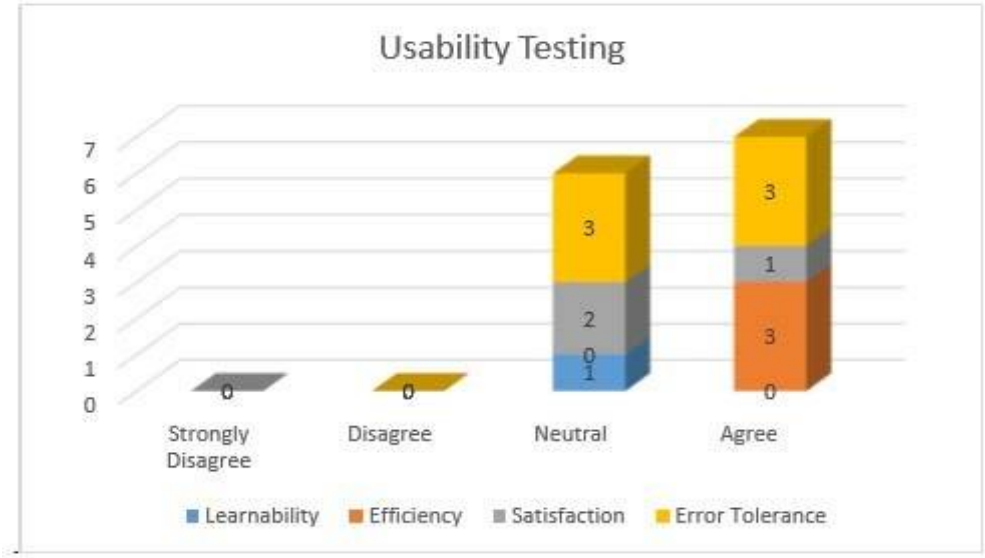

Figure 5: The usability Graph

Figure 6, is the graph that graphically presents the usability testing of the system. In short these results shows that the system was able to achieve the proposed objectives.

7. Industrial Significance and Eventual Benefits

Even though the main aim of the research was to develop a network monitoring system for UFH community, however this NMS can be replicated for other networks. Broadly, there are many tangible benefits that are associated with NMS business processes, it is safe to say many institutions and some companies operate on critical networks which requires to be monitored every time. Given enough equipment and enough time this NMS can be expanded to operate on larger networks that are more complex than UFH network. Moreover, the idea could be sold to huge companies that depend on a network in other to produce high level productivity. During system implementation and testing the Internetwork Operating System (IOS) of the routers did not support memory configurations and that led to some challenges that affected the success implementation of the system because some of the objectives were not met due to this type of challenge. In a nutshell, if in the future the university can have routers that have IOS that supports all types of configurations, then this system can be very useful not to the university only, but to all other surrounding institutions so that there is a common platform for monitoring the network for institutions of higher learning.

8. Technical/Business Case Quality

This type of NMS was meant to monitor network availability and network performance. In terms of network availability, the developed system reports when the system went down and when the system is up. Additionally, it also has a functionality to also report the reasons that made the network to be down such that the administrator can know exactly where to fix, it also has an ability to produce different timeframe of network availability graphically. However, in terms of network performance the developed system has a functionality of portraying when the performance of computers is underutilized and over utilized. Furthermore, with help of this system users can know which time is the best for

using a network with help of different graphs that show availability, For example in the morning around 6 am the network is always available so if one wants a fast network it is recommended to use it in the morning when some of the labs are still empty.

9. Conclusion

There are many NMS that have been developed such as Wireshark, but not most of them focus on network availability and network performance, for example wireshark is a network analyzer. Due to the shortage of NMS that mainly focus on network availability and network performance this research work proposed to develop a prototype of NMS that will focus on network equipment availability and network performance. This system is meant to help network administrators with troubleshooting network devices because it will tell them exactly where to go in order to fix the network. Additionally, the prototype will also have an ability to generate charts that presents the network performance in and untreatable format and also send notifications to the network administrators in cause of any outages.

Even though, the prototype was successfully developed, for example some of the features were not fully implemented due to few challenges that were faced. For examples, Router's IOS was one of the challenges that were faced, the routers that were used during testing did not support memory configurations, they only had an ability to support flash memory configurations which only focused on the hard drive memory performance rather than the entire computer performance. The limitation of memory configurations led to the failure of memory performance which was one of the objectives that were proposed. Furthermore, reporting notifications with alarms were not successfully implemented due to the network restrictions of the university network.

Lastly, given enough time and enough resources all the objectives could have been implemented successfully and even add new functionalities to the prototype. Lastly, with the support of developed prototype the researchers and developers can now have an ability to expand the system by developing a system that will not only report and point where is the fault on a network, but a system that will have an ability to automatically solve the problem itself and keep record of the worse performing network devices on a network so that they can be easily removed.

References

[1] http://cs.sru.edu/~mullins/cpsc100book/module08_networks/module08-01_networks.html. [Accessed 04 December 2016].

[2] Techopedia, "Network Monitoring," Techopedia, [Online]. Available: https://www.techopedia.com/definition/24149/network-monitoring. [Accessed 6 November 2016].

[3] V. N. Gourov, "Network Monitoring with Software," Delft University of Technology, 30, August, 2013.

[4]" Research Methodologies Mixed-Methods," The Qualitative Report, vol. Volume 17, pp. 254-280, 2012.

[5] I. A. F. YOU, I ANSWER FOR YOU, 2007. [Online]. Available: http://www.ianswer4u.com/2011/12/spiral-model-advantages-and.html#axzz3qaE5E7tn. [Accessed 12 AUGUST 2016].

[6] TechNet, [Online]. Available: https://technet.microsoft.com/enus/library/cc776379(v=ws.10).aspx. [Accessed 21 July 2016].

[7] "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Java_Database_Connectivity. [Accessed 4 June 2016].

[8] TechNet, [Online]. Available: https://technet.microsoft.com/enus/library/cc776379 (v=ws.10).aspx. [Accessed 21 July 2016].

[9] C. Systems, "Defining Memory Threshold Notifications," Defining Memory Threshold, vol. 2, p. 182, 2013

[10] D. Schreckmann, "An Introduction to Java Development with NetBeans IDE," An Introduction to Java Development with NetBeans IDE, 2003.

[11] Dr. Sharon Bender "Implementation of an Iterative and Incremental SDLC (Systems Development Life Cycle) Model Development Project for a Financial Services Organization" March 3, 2006

[12] B. Wijnen, IBM T. J. Watson Research R. Presuhn BMC Software, Inc. McCloghrie, View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP) April 1999
[13] W. L. Bai, Y. M. Zhang, B. Song, "SNMP4J-based design and implementation of MAS network management ", Applied Mechanics and Materials, Vols. 268-270, pp. 1714-1717, 2013
[14] Karjoth, G., D.B. Lange and M. Oshima (1997), "Mobile agents with Java: The Aglet API" IEEE Internet Computing 1, 4, 68–77.
[15] JFreeCharts. http://www.jfree.org/jfreechart [Accessed 12 September 2016].
[16] J. Toby Mordkoff, The Assumption(s) of Normality, Copyright © 2000, 2011, 2016