Novelty Journals

ISSN 2394-7314

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 3, Issue 2, pp: (55-68), Month: May - August 2016, Available at: www.noveltyjournals.com

Network Monitoring System

[1]Martin Phiri, [2]Peter Mfupa

Abstract: **This paper reports on the benefits and also establishes how necessary network monitoring is on a network in academic and corporate organizations and on how to properly maintain and administer it. A network that is not monitored is a black hold and results in faults going unnoticed for extended periods of time.** Assertions **are confirmed in this paper that there are no technically enforced restrictions on what network traffic can go into or out of most college and corporate networks. Set rules which users usually agree to abide by are broken therefore rendering the network prone to viruses and advanced hackers.**

Views and attitudes of network users and administrators were investigated and how these can be factored into developing network monitoring systems that address today's network issues. To an extent network abuse is inevitable. In most organisations and especially academic institutions, it is very difficult to impose technical restrictions on network traffic. In corporate networks 'acceptable' traffic can often be clearly defined. This is not the case in colleges or learning institutions. Almost any port might be required for some reasonable purpose (now or in the future), so simply banning traffic by port would be difficult. Restriction by type would be hard too – for example P2P software is used by research groups for ease of collaboration, as well as by users sharing copyright material.

Systems and network administrators are burdened with fighting fires when something goes wrong on the network. They have no information of what may have caused a network failure. This is more so on computer networks that do not have any high grade network monitoring software installed. This scenario is common place in academic institutions because of the aforementioned reasons. This was the inertia for this research and project development.

Improved techniques for building a better network monitoring system were devised and investigated. A new network monitoring tool was built from the ground up using new technologies. Some of the new techniques devised were integrated into the new network monitoring tool to improve usability and usefulness of a network monitoring tool from the perspective of network users and systems administrators.

Keywords: **Network Monitoring System, P2P software.**

1. INTRODUCTION

Every student at ICU is required to select a research topic in the final year of study. This document outlines the research topic that this student identified and embarked on as the project for the final year. Living and working in the technological era has brought about inter connections of computers of all kinds. Computers are used in both work and home settings. This project investigated how developments in network traffic management has evolved over the years and also looked at the attitudes of users of computer networked environments. This research focused on the tools from the point of view of network administrators. Most small to medium sized businesses are unable to invest into robust network management software due to the huge capital investment that is required in such projects.

Systems and network administrators however require tools that are affordable to ease the management of computer networks. This research culminated into the development of a Network Monitoring Tool for network administrators. The tool has a light foot print and does not require the use of a database. It instead monitors network traffic as well as available network adapters on a networked computer. These features are aimed at assisting systems and network administrators to identify problems as they occur in real time. The two organizations that were investigated in this research are the University of Zambia (UNZA) and Tanzania Zambia Oil Pipeline (TAZAMA).

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 3, Issue 2, pp: (55-68), Month: May - August 2016, Available at: www.noveltyjournals.com

2. PROBLEM STATEMENT

This research is aimed at confirming the assertions that there are no technically enforced restrictions on what network traffic can go into or out of most college networks. There are rules and regulations which users agree to abide by, but if they choose to ignore these they are able to do so. This also makes the network more vulnerable to viruses and advanced hackers of the 21stcentaury. Inappropriate traffic can slow the network down, or even bring it to a complete shutdown, causing frustration to legitimate users of the network. Illegal traffic, such as pirated movies and music, can get both the college and the individual users into serious litigation.

To an extent network abuse is inevitable. In most organisations and especially academic institutions, it is very difficult to impose technical restrictions on network traffic. In corporate networks 'acceptable' traffic can often be clearly defined. This is not the case in colleges or learning institutions. Almost any port might be required for some reasonable purpose (now or in the future), so simply banning traffic by port would be difficult. Restriction by type would be hard too – for example P2P software is used by research groups for ease of collaboration, as well as by users sharing copyright material.

Systems and network administrators are burdened with fighting fires when something goes wrong on the network. They have no information of what may have caused a network failure. This is more so on computer networks that do not have any high grade network monitoring software installed. This scenario is common place in academic institutions because of the aforementioned reasons. This was the inertia for this research and project development.

3. PURPOSE OF THE PROJECT

The purpose of the proposed project is to find the best network monitoring solution from a system administration point of view. One solution is to create some way of monitoring what is going on, so that if a problem arises, such as possible virus-generated traffic, movement of copyright material or high network load, it can be easily spotted, traced to its source and dealt with.

From an academic point of view, the research and information generated by this project will be very valuable. Usage of academic networks has changed dramatically in the last few years. A great many more students have their own computers, and the percentage rises every year. New viruses and file-sharing programs are constantly appearing, and vulnerabilities in software are being discovered all the time. My research, which will include surveys of users and system administrators, will look at the changing attitudes of both towards security, legitimate network use and how they feel they use their network connection. The NetMon tool will monitor network traffic and bandwidth usage on a given computer and this data will then be co-related with the attitudes of users on the network. Both will provide invaluable information to system administrators of organisations and academic networks to help them devise a secure network for today's organisational and academic needs.

From a personal point of view, while I have obviously picked up a certain amount of knowledge about network security from my work, it is an area I would like to learn a lot more about. I have had minimal experience of network traffic monitoring, and so I would benefit greatly from the experience that I will get in setting up and using the systems whilst doing this project.

4. PROJECT OUTLINE

This project will focus on the University of Zambia network as well as the TAZAMA pipelines network. The selection of these types of networks will enable me to focus on two disparate network types as well as analyse the attitudes of two very different types of users of these networks. Both networks provide services for several hundred users and include university managed machines, organisational managed machines and those owned and managed by individuals.

The university network is predominantly for academic use whereas the organisational network is predominantly for work related use. Obviously having so many different machines and owners on the network is a cause for concern, and some users abuse their access. This project will investigate and implement a system to allow the Computer Officers to protect and manage the network more efficiently, including monitoring network use, tracing and dealing with network faults, problems and abuse. It will also provide extremely useful research into the constantly changing network usage patterns and the means that system administrators are employing to keep their networks open to legitimate use while tracking non-legitimate use.

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 3, Issue 2, pp: (55-68), Month: May - August 2016, Available at: www.noveltyjournals.com

5. JUSTIFICATION OF THE PROJECT

This project will provide information about the fast-changing attitudes of users of academic and organizational networks, the types of traffic seen on networks and the techniques system administrators must employ to keep control of today's networks. This has changed significantly, even over the last few years. Illegal file sharing is very common with many users seeing it as perfectly acceptable. Other users view such usage as a legitimate right because they claim the pay for such usage.

More generally, traffic management of an organization or academic network is more complex, with new and more sophisticated viruses and hacking attempts to keep watch for. Computer ownership increases noticeably every year, allowing more and more people, often inexperienced, to gain Internet access through the networks that they use.

All of this means that system administrators must be able to be one step ahead of their users. This project will collect information from users and administrators of similar systems to provide a wealth of data that would not be otherwise available. It will also provide a practical monitoring tool developed from the perspective of network and systems administrators.

6. BACKGROUND

6.1 Networks:

Computer networks can be viewed as a series of devices that are interconnected and able to communicate with each other. Most networked environments connect computers via switches and routers. Switches provide the forwarding capability that allows logically neighbouring devices to communicate. Routers add the routing capability that provides the network with structure and allows communication between sub-networks. This project is interested mainly in the Internet Protocol (IP) networks (Information Sciences Institute, 1981). Computer networks vary significantly in size and importance. The larger and more important networks can cost organisations that are running them large sums of money for every minute that they are unavailable or malfunctioning.

6.2 Network Monitoring Systems:

When networks grow and become considerably large such as for the UNZA, it becomes infeasible for one person to maintain a mental model of the entire network. When this happens, the network is an unknown entity where faults could occur at any time and not be detected by network operators. A Network Monitoring System is a software package used to solve this debacle and diagnose faults on the network. It achieves this by storing an internal model of what the network is supposed to be and uses this model to evaluate the current state to the network. This enables the network monitoring system to provide insight into the otherwise unknown entity. The system also provides performance data on how well the network is utilized and answers questions regarding economics, i.e is the network cost effective and meeting demand?

A network monitoring system should be able to monitor all these occurrences on the network without putting undue load on the devices being monitored. Not many network monitoring system are able to achieve this feature.

Different techniques can be used by network monitoring system to monitor a network. The rest of this section focuses on the features required in an ideal network monitoring system and those that should be provided to produce a general purpose network monitoring system.

6.2.1 Configuration System:

Well-designed configuration systems are important for network monitoring systems. Networks change frequently which could lead to a network monitoring system network model being outdated, resulting in "black holes" that are not monitored. If the network monitoring system is easy to initiate and its configuration is easy to maintain, the model of the network should be more accurate.

6.2.2 Service Polling:

Service polling is a type of network test where the network monitoring system regularly checks whether a device or a service is available and working within normal parameters. This allows for answering availability questions such as whether a server hosting our website on our network is reachable or not. More specific service polling tests may collect additional information about network state. One example of such additional information is verification of a web server's software; that it is running correctly, is responsive, and is serving the correct content without any errors.

6.2.3 Graphing:

Time-series graphs depicting performance data drawn by a network monitoring system can be useful for identifying trends and anomalies. Where a large quantity of data values is given, such graphs are frequently used to identify changes in network state. Graphs need to be tailored to suite the network being monitored and some data is more useful to graph than others. A CPU time-series graph is useful as it indicates whether or not devices are being overworked and how close they are to operating at full capacity. Graphing bandwidth usage on network interfaces shows how close the network is to running at full capacity as well. This information is useful to systems administrators and can be used to plan network upgrades and identify likely bottlenecks.

6.2.4 Notifications and Events:

Good or bad network changes should somehow reach someone especially those in charge of network monitoring. Network monitoring systems achieve this notification feature using event systems. Event systems can be very simple systems that check if some value is within a given threshold or has a certain value. If for example a host becomes unreachable, an event may be triggered. Complex event systems can also watch for undesirable trends or use anomaly detection to identify events that could impact the network. However, the notification system alone does not provide the full picture and some expertise may be required to determine what has caused an event to trigger.

6.2.5 Dashboard:

A dashboard is a user interface that provides a visual display of information using a single screen such that all this information can be monitored at a glance (Stephen Few, 2006). This is useful for network monitoring systems because it allows the whole network to be monitored from a large monitor in view of the system administrator and does not require actively searching through many pages of the network monitoring system to manually identify problems. Many dashboards include graphs that should be regularly checked for that network, for example performance graphs. It will also include summary statistics such as the current number of faults or a generalized network health measure. It should streamline the process of finding faults by ensuring that it is easy to discover the root cause of an event and by pointing the user in the direction of where they should start any further investigations.

6.3 Popular Network Monitoring Systems:

Organizations that have the financial muscle are able to implement robust network monitoring systems that offer all of the above features and more. Some of the more popular systems are briefly examined in this section.

6.3.1 Cacti:

Cacti (cacti.net, 2015) is a network monitoring system designed for drawing time-series graphs of performance data on a monitored network. It typically draws a different graph for each monitored data source. A graph can be drawn from multiple data sources, but requires a suitable template created using the cacti format. The configuration system is entirely web based and there is no provided method for performing bulk configuration. The additional effort required to update the network model in Cacti often discourages the user from monitoring everything. Cacti also does not include an event detection system or a notification system and therefore is usually used to supplement another network monitoring system by providing historical graphs. These provide more visibility and insight as to why an event may have triggered in another monitoring system.

Novelty Journals

ISSN 2394-7314

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 3, Issue 2, pp: (55-68), Month: May - August 2016, Available at: www.noveltyjournals.com

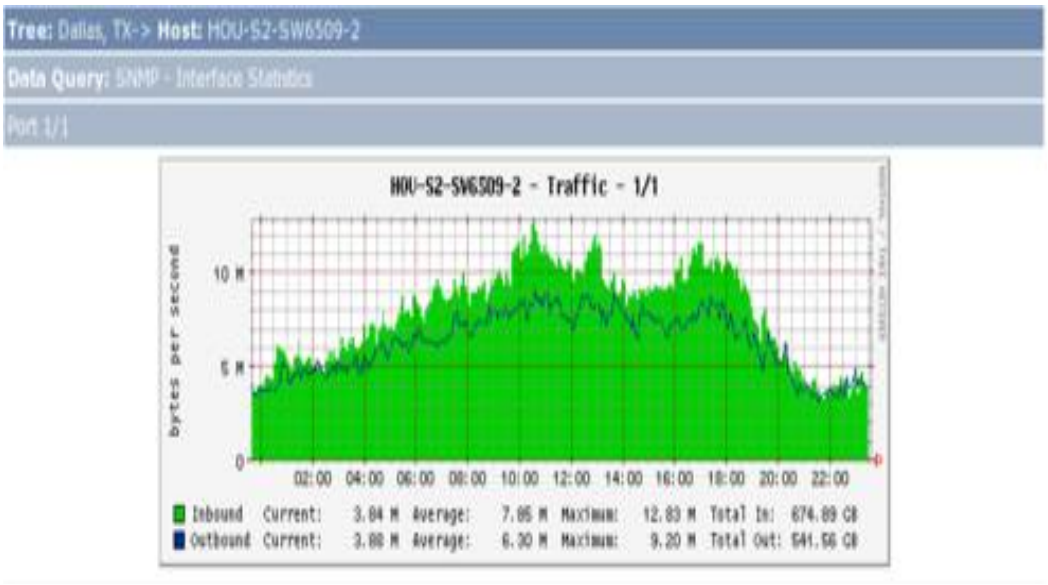


Figure 6.1: Traffic data graph from Cacti on network switch

Data to be graphed in Cacti is collected using SNMP (simple network management protocol) at a specified rate. This default to five minutes but with some effort can be reduced to a faster rate.

6.3.2 Icinga:

Icinga (Icinga, 2015) is a network monitoring system designed specifically for service polling, notifications and report generation. It is a fork of Nagios (Nagios, 2009-2015) and uses a large portion of Nagios code still in its core. Icinga service polling is modular: each service check is handled by a separate check script or process which is forked by the Icinga monitoring system. The check process will exit with an exit code that matches the severity of the problem (ok, warning or critical). Performance data from the test can be reported to Standard Output (stdout). Since Nagios has been an industry standard for the past ten years and is easy to script and modify, there are many different community maintained extensions that can extend Icinga and Nagios to support extra features. Data collection is handled by the check script and any protocol could be used to collect measurement data. SNMP is used exclusively for this purpose.

6.3.3 Wireshark:

Wireshark (Wireshark, 2015) is a network packet analyzer. A network packet analyzer will try to capture network packets and tries to display that packet data as detailed as possible. It can be thought of as a measuring device used to examine what's going on inside a network cable.

Wireshark can capture traffic from many different network media types – and despite its name – including wireless LAN as well. Which media types are supported, depends on many things like the operating system in use. However Wireshark is not an intrusion detection system. It will not warn you when someone does strange things on your network that he/she is not allowed to do. You have to manually decipher what is going on from the packets that Wireshark captures. Wireshark will not manipulate things on the network, it will only "measure" things from it. It does not send packets on the network or do other active things (except for name resolutions, but even that can be disabled.)

7. ISSUES WITH EXISTING SOLUTIONS

In the previous section we examined some of the existing network monitoring systems that are used to monitor networks. This examination is not comprehensive but is an indicative guide on how the systems operate and some of their strengths and weaknesses. These systems use a number of tool and libraries to achieve the functionality that they deliver. In this section we will identify common problems in these systems that make them unsuited to being used as general purpose network monitoring systems. These issues are the reason that it is common to deploy multiple network monitoring systems, each with their own strengths and features, to provide full monitoring coverage for a network.

7.1 Data Collection:

SNMP (Simple network management protocol) is the industry standard for data collection in a network monitoring system. SNMP is commonly configured in a centralized architecture, where a single SNMP collector collects data from every device on the network. A paper studying the behavior of SNMP collectors (Colin Pattinson, 2001) raised issues with the performance of the centralized design of SNMP when used for large-scale network monitoring. The paper suggests that a better design is a decentralized approach. Such an approach would use remote collector agents to monitor a subset of the network. Each remote collector would monitor a subset of the network. These collector agents periodically export data to a central system for storage. The benefit of this approach is that it works well when collecting data over the Internet. Moving the collector agent logically closer to where the devices are is also beneficial because often devices on a network are held in different geographical areas for redundancy. This is because a single expensive connection to a central storage server over the Internet is better than one expensive connection per monitored device.

SNMP is usually exclusively used in a polling configuration and this too is an issue. SNMP can be used to push data with the SNMP trap, which is built into the protocol for pushing data immediately to a collector. However, due to the difficulty of configuring traps correctly, the feature mostly goes unused. This leaves use relying on the poll interval of a SNMP collector being fast enough to detect important changes. However, monitoring a large network with a single SNMP collector will prevent a fast poll rate because of load and bandwidth constraints on the collector machine.

Network monitoring systems usually bundle a SNMP poller that they use to collect data. Once this is done it is either unchangeable or difficult to change. In a network where multiple network monitoring systems are deployed to provide the full suite of monitoring tools, this means we have to run multiple SNMP pollers as well. Some SNMP implementations do not perform caching of values so polling values on a device multiple times can be expensive to that device. This contradicts our requirement that monitoring should not greatly impact the services of the network.

7.2 Dashboard:

The systems outlined in the previous sections all have useful dashboards. This is because the dashboard is the interface that most users of the system deal with day to day. A network monitoring system will not be used to monitor a network by a network or systems administrator if it has a bad interface.

The main issue with dashboards is that networks must be monitored by multiple network monitoring systems to have full testing coverage over the entire network. This means that multiple dashboards need to be used to track the status of the network. A user will be glancing at multiple dashboards to scan for faults which can be difficult and distracting. A solution that can unify this all to exist on one dashboard would better satisfy the requirements we are seeking.

7.3 Configuration:

The common configuration systems that we have seen from the examined network monitoring systems are plain-text configuration files. These configuration files define the network model and how network monitoring systems should be monitoring the network.

Using plain-text files for configuration is usually a huge undertaking requiring a large amount of configuration. As an example, a small Icinga instance that monitors 34 devices investigated during this project used up to 3724 lines of configuration to define the network model. This magnitude of configuration is prone to human errors which if no attention is paid to detail can go unnoticed for extended periods of time. This has a negative impact on how well the network is monitored. Cacti has a limitation in its configuration because the configuration is locked in a custom relational database management system schema that is only accessible through a web interface. This interface does not provide bulk configuration options, which adds to the effort required to configure a whole network. These problems inhibit us from meeting the requirement of encouraging users to have high testing coverage by accurately setting up and maintaining the network model in the network monitoring system.

Running multiple network monitoring systems exacerbates the problem because the user must now maintain a number of different configuration systems. To make matters even worse, the configuration systems are usually incompatible. As a result, one change to the network model will mean updating this in multiple systems, further increasing the likelihood of mistakes.

Novelty Journals

ISSN 2394-7314

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 3, Issue 2, pp: (55-68), Month: May - August 2016, Available at: www.noveltyjournals.com

8. INVESTIGATION

It has become apparent at this point that there are a number of issues with the most popular network monitoring systems used to monitor networks apart from implementation costs. Some of the newer systems investigated may be heading in the right direction but still remain targeted to specialized monitoring and are missing important features such as notification systems and decent dashboards. An example in question is Wireshark.

In this section we will investigate new techniques and updates to current techniques as well as the attitudes of computer network users. This investigation helps us to build a network monitoring tool suited for system administrators and which overcomes some of the limitations we have identified thus far. This section will also endeavor to introduce new methods that can be implemented in a network monitoring tool alongside current methods to improve the configuration and maintenance of the network monitoring tool.

8.1 Data Collection:

This section defines a new set of requirements for a new data collector to replace SNMP. By replacing SNMP we seek to remove the limitations that it adds to the data collection section of a network monitoring system.

Most devices being monitored by a network monitoring system are powerful servers or switches. The requirement to have a very simple protocol that can work on low hardware requirements is therefore unnecessary. The efficient alternative is to use an efficient protocol built for transmitting large amounts of data points regularly without significant load impact on our collection server. A data collector with support for pushing data as a first class citizen allows important data to turn up instantly rather than on the next poll interval. A collector with a fast poll interval, to the point of being able to collect data at real-time without much overhead, will add another technique for a network or systems administrator to use to inspect active faults in greater depth. Using a decentralized system eliminates a single point of failure for our collector and improves scalability.

8.2 Automatic Configuration Discovery:

One of the stated goals of this project is to ensure the configuration is easy to initialize and maintain. The best method of achieving this goal is to limit the amount of configuration required by our network monitoring tool by performing automatic configuration where possible. This section explores three different methods of automatically generating different parts of the configuration for a network monitoring tool.

However automatic configuration does not solve all our configuration issues. Automatic configuration must be maintained through some mechanism. This maintenance may prove to be too expensive to run with every interaction with the network monitoring tool. We need to have an update interval so that when the automatic configuration is performed changes are merged into the current network model. Our tool will keep this network model internally reducing on any storage requirements which may have a negative impact on the network.

8.2.1 Topology:

Topology discovery is a method of scanning a local network or computer and determining the logical topology of that network. This is an important process because the network may have devices hidden in many different logical segments of it. By discovering all network segments and finding the entire network we can ensure that monitoring black holes are not introduced to our network monitoring by leaving out segments from our network model.

8.2.2 Devices:

Once we have a network topology, the next stage of the automatic configuration system is device discovery. In the context of our network monitoring tool, devices refer to adapters on the computer on which automatic configuration is being performed. Therefore this step is used to find adapters on the individual segments of the network that was revealed by the topology discovery. This results into a list of adapters that are to be monitored. There are a number of different techniques available for scanning adapters on a given network segment. We use an iteration technique through the available network adapters to discover all the available ones and list them as devices to be monitored.

8.2.3 Services:

Service discovery is very similar to device discovery. When we have a list of adapters to monitor each adapter can then be independently probed to determine the services that it hosts. This can be as simple as doing a scan of open ports on the

host machine and matching these against the well-known ports that common services use. Open ports can also be queried for a banner to see what service is listening on that port. In the test phase of the project Nessus (Jay Beale, etl, 2004) was used to verify that our network monitoring tool was scanning and identifying the services correctly.

8.3 Network User Attitudes:

In order to make the network monitoring tool relevant to the organizations under investigation we interviewed a sample population of network users and systems administrators at UNZA and TAZAMA. This section summarizes what their views are regarding network usage and monitoring.

8.3.1 Views and Attitudes of Computer Network Users at UNZA and TAZAMA:

The continued growth of the cyber space era has brought with it a permanent change in the way students and workers in industry interact, socialize and do their research or work. A major part of this change is the advent of so called social networking sites on the internet, which have evolved to become virtual communities, where people communicate, share information and perhaps most important share their work or research ideas. The views of network users for both learning and corporate institutions are that network availability is priority number one. Connection speeds also are an issue that users expect to have reasonable enough for their use. Research is an area highlighted as number one for both learning and corporate institutions, therefore the demand for lesser restrictions on the sites to access. Users feel the restrictions that are put in place by administrators are hindering them from achieving their intended goals. They therefore would like to be involved or consulted before restrictions to internet access are effected.

8.3.2 Views of Network and Systems Administrators at UNZA and TAZAMA:

Availability of Network is one of the most critical items an administrator looks at. This is for both corporate and learning institutions. Once the network is available, network security is the second most important issue to be considered. Network administrators always search for the best network monitoring tools, because a system administrator always needs to know about the status of their systems so that he can optimize performance and head off potential problems. Definitely, dealing with the networking system needs proper knowledge and good experience so that one can easily deal with the daily crashes, frequent errors and failures.

To be able to identify potential problems even before users start to complain, the administrator needs to be aware of what is normal in the network. Base lining network behaviour over a couple of weeks or even months will help the administrator understand what normal behaviour in the network is. Once normal or base line behaviour of the various elements and services in the network are understood, the information can be used by the administrator to set threshold values for alerts.

The network always has to be available because learners are always using it. The second most important thing as mentioned is security. Under an educational institution, such an organization is put under the scrutiny of software houses. Meaning that the software that is installed on the network needs to be correctly licensed and the network traffic always needs to be legitimate. Accessing unauthorized tasks should be monitored which brings us to the traffic the tool monitors in addition to the monitoring of availability that the tool conducts.

Learners are usually a notorious bunch and do not adhere to the regulations set out by the institution. It is therefore the province of network administrator to be on top of things in terms of monitoring the network traffic, usage, and availability. This also goes for corporate institutions like Tazama.

9. IMPLEMENTATION

The initial evaluation of existing network monitoring systems has demonstrated that no general purpose and light foot print network monitoring tool exists that meets all the requirements for this project. As a result, new methodologies and technologies are necessary to build a light foot print and refreshed network monitoring tool with systems and network administrators in mind.

Furthermore, this project aims to produce a system that we will call a Network Monitoring Tool, where the emphasis is on simplicity and dynamism, rather than statically, configured network models. Many key design decisions have been made with these goals in mind.

Novelty Journals

ISSN 2394-7314

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 3, Issue 2, pp: (55-68), Month: May - August 2016, Available at: www.noveltyjournals.com

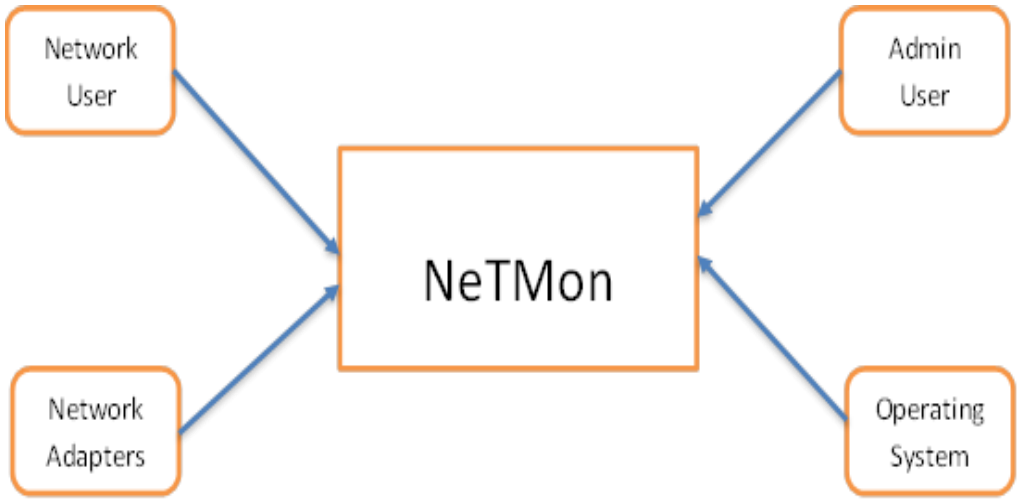
The NetMon Tool has been designed to be a modular tool for two reasons. First, features can be implemented incrementally while maintaining a working system at each development stage. This is important because building a feature complete network monitoring system is outside the scope of this project due to technological and time constraints. The second reason was to allow users to extend the feature set and tailor NetMon for their particular needs and to enable improved versions of modules, such as the data collector module, data storage, to be substituted without having to keep both loaded in the program at the same time. The modularity also helps reduce software bloat because the running version of the software only needs to enable the features required to monitor a particular host computer.

Another key design principal employed in the design of NetMon is to limit the amount of configuration that is required for the system to function. The minimum set of configuration required to monitor network adapters is stored in memory. Anything not required is considered non-essential. This is implemented by attempting automatic configuration where possible and having sensible defaults pre-configured for use if automatic configuration fails. This approach limits the amount of configuration that is required to run NetMon and reduces the likelihood of mistakes. Also because the configuration is easy to maintain, it is more likely to be up to date.

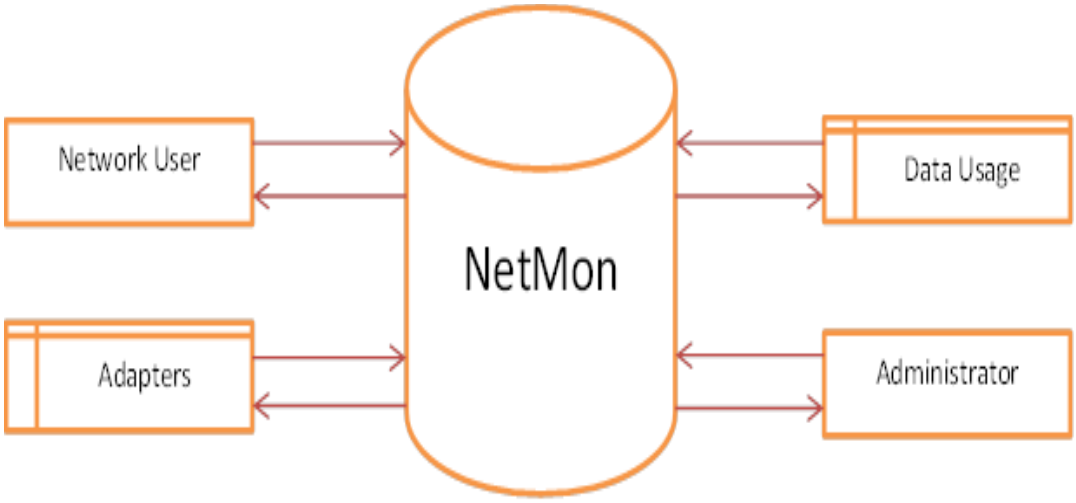
9.1 System Design:

This section presents high level system diagrams of NetMon.

9.1.1 Context Diagram:



9.1.2 Data Flow Diagram:

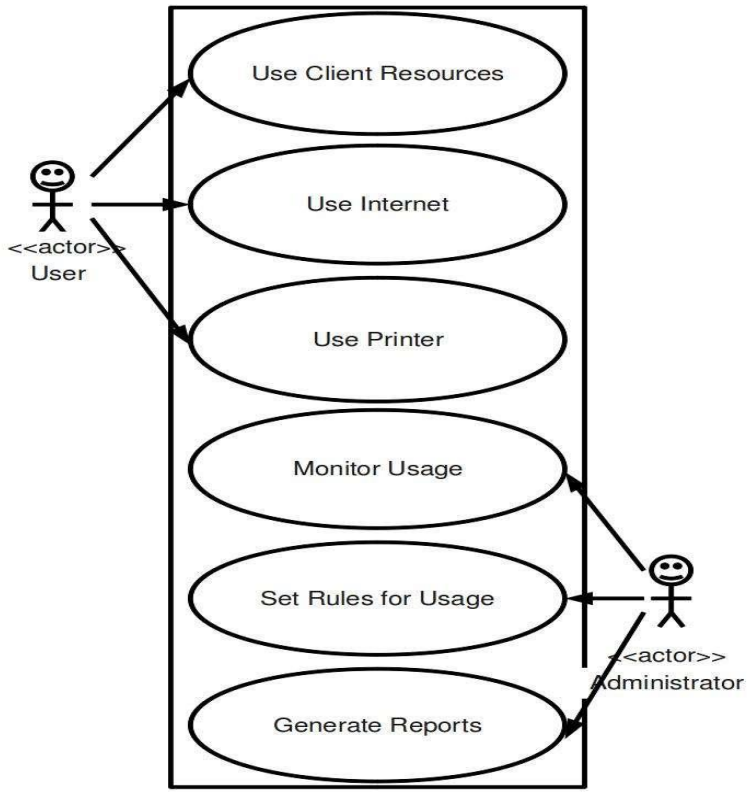


Novelty Journals

ISSN 2394-7314

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 3, Issue 2, pp: (55-68), Month: May - August 2016, Available at: www.noveltyjournals.com

9.1.3 Use Case Diagram:



9.1.4 Development Methodology:

This project followed an Agile development methodology with an emphasis on delivering iterative working system components to stakeholders. The users would then provide feedback on the modules that needed improvement and those would be worked on. The diagram below shows the stages that where followed in the development process.

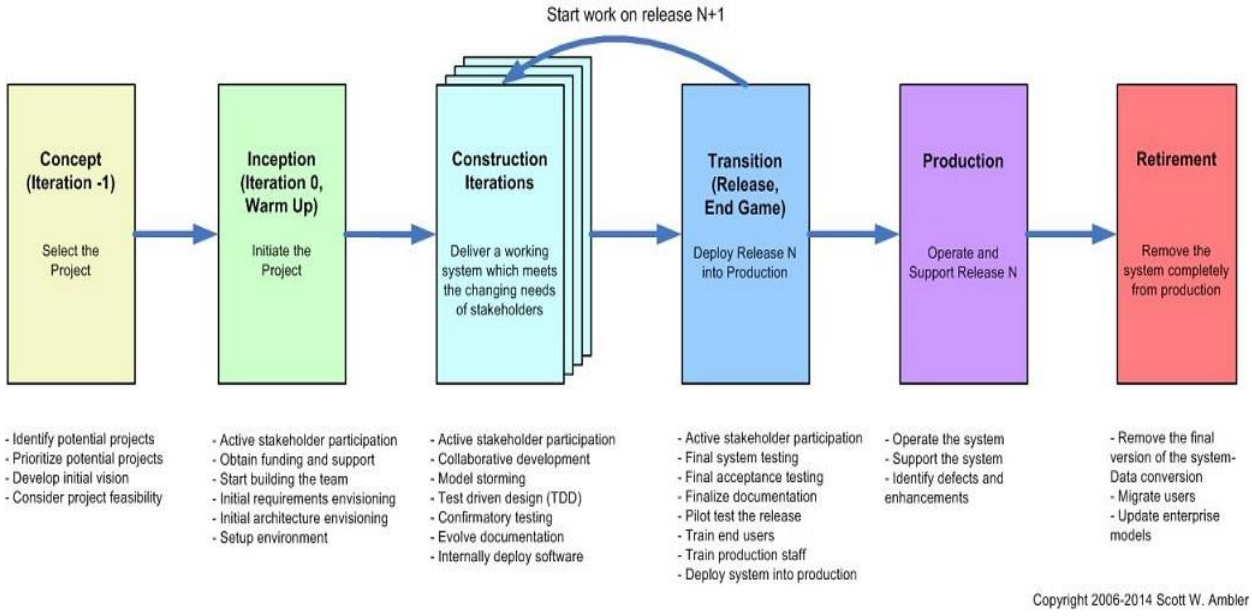


Figure 9.1 The Agile SDLC (high-level)

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 3, Issue 2, pp: (55-68), Month: May - August 2016, Available at: www.noveltyjournals.com

Agile software development is described as a group of software development methods in which requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. It is said to promote adaptive planning, evolutionary development, early delivery, continuous improvement, and encourages rapid and flexible response to change. It also emphasizes just enough documentation as opposed to the traditional methods of software development especially for small projects. It is worth bearing in mind that in software development "no one size fits all" therefore the methodology is adapted to suite the project at hand. The high level stages of the Agile Software development Life Cycle are as follows:

- ***Concept Phase*** – the optimal project is selected from potential projects and an initial project vision is developed with consideration of project feasibility

- ***Inception Phase –*** the project is initiated with active participation of stakeholders. Funding and support are also solicited whilst the team is built and initial project requirements and initial architecture are envisioned. The development environment is also set up during this phase.

- ***Construction Iterations –*** Delivery of a working system which meets the changing needs of stakeholders is performed. This involves active stakeholder participation, collaborative development, model storming, test driven design (TDD), confirmatory testing, document evolution and internal deployment of the software solution

- ***Transition (Release, End Game) Phase–*** Deploy Release N into production. This phase involves active stakeholder participation, final system testing, final acceptance testing, finalizing documentation, pilot testing of the release software, training end users and deploying the release into production

- ***Production Phase –*** Operate and Support Release N. This phase involves operating the system, supporting it and identifying defects and enhancements.

- ***Retirement Phase –*** Remove the system completely from production. Every system eventually reaches obsolescence. This occurs during this phase in which the final version of the system is removed from production, data is converted to a newer system, users are migrated and the enterprise models are updated.

9.2 Technologies Used:

This section outlines the technologies that were used to develop NetMon.

9.2.1 Visual Studio 2013:

Visual Studio is Microsoft's proprietary Integrated Development environment. It has reach features for application development as well as development of web based applications. The rich libraries provide classes that are used to communicate with low level system hardware. This was one of the influencing factors in selecting this environment. Visual Studio supports C++, VB, C# and F# as the back end programming languages.

9.2.2 C-Sharp (C#):

C# was used as the programming language of choice for development of this project. It is an industry standard software development language with its roots in C++. It is however much simpler to learn than C++ and less prone to programming errors. C# offers a rich set of libraries that communicate with low level hardware such as network interface cards. These libraries were used in the implementation of NetMon.

9.2.3 Forms Application:

NetMon is developed as a desktop windows forms application. When packaged into an executable it is able to run on any Windows framework with any need for installation. It is has been developed with simplicity in mind and a very low foot print.

Novelty Journals

ISSN 2394-7314

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 3, Issue 2, pp: (55-68), Month: May - August 2016, Available at: www.noveltyjournals.com

9.3 NetMon Screen Shots:

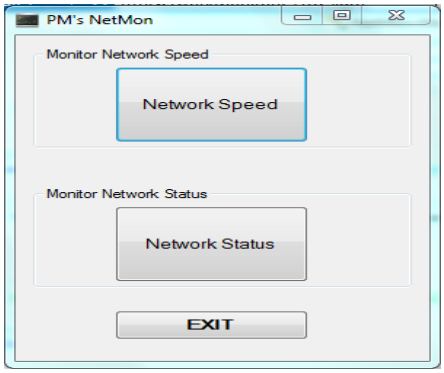


Figure 9.2 NetMon Launch Screen – the launch screen presents two options for monitoring network speed and network status on the host computer

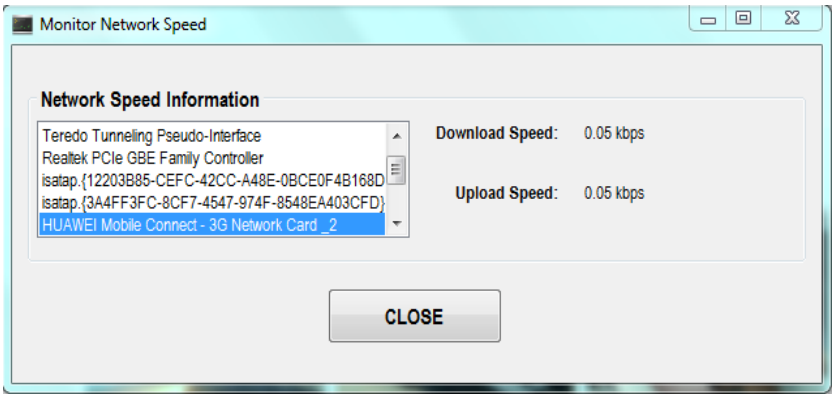


Figure 9.3 NetMon Network Speed – the Network Speed monitor lists available network adapters that NetMon is monitoring. Selecting a network adapter from the list shows the download and upload speed of that adapter.

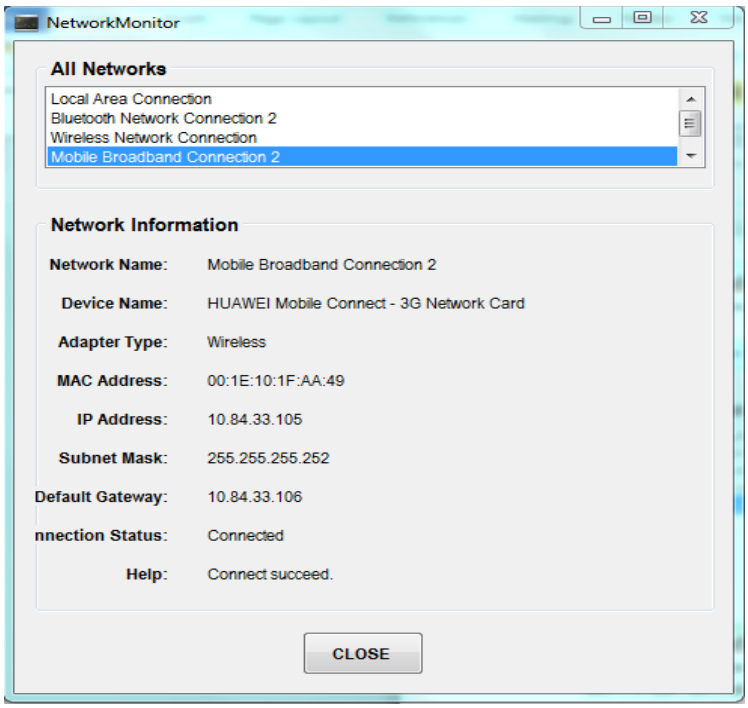


Figure 9.4 Network Monitor – the network monitor monitors the status of the network adapters on the host computer in real-time. Selecting an adapter from the list displays the status of that adapter.

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 3, Issue 2, pp: (55-68), Month: May - August 2016, Available at: www.noveltyjournals.com

10. FUTURE WORK

Network monitoring systems are very large systems and to complete a fully featured network monitoring system is a mammoth task. The NetMon tool has been implemented as the framework to build a feature complete network monitoring system on top of. The core feature of a network monitoring system has been implemented in NetMon that of monitoring network devices these being network adapter monitoring in real-time.

10.1 Event Notification Module:

A major feature that a network monitoring system should have is event notification. When something out of the ordinary occurs on the network, the network monitoring system should be able to send an event notification to the user so that the triggering event is investigated further. Additionally the event notification system should have the ability to send only event notifications that are solvable by the user. Sending event notification that the user is unable to solve would be unintelligent and a waste of resources. The event notification system should also be able to group similar triggering events and send only a single notification for such events as opposed to sending multiple notifications for a variation of the same base event trigger. This feature is an enhancement that would greatly improve NetMon. The notification system would also benefit from being modular and different notification modules could be turned on and off. For example notifications could be made over email, short message service (SMS), pager or Instant Message (IM).

10.2 Data Storage:

NetMon currently stores all its configuration and monitoring data in memory. Once the tool is closed, this data is lost. Having a persistent data store allows network monitoring systems to be able to generate historical data and thereby generate reports based on that data. An ideal implementation of this data storage would be using SQL database with a Relational Database Management System (RDBMS) to store the data. Incorporating this feature in NetMon would mean one of two things: either re-implement the tool so that it is installable on a network server which has a RDBMS or incorporate a RDBMS in NetMon itself for data persistency. Both scenarios have added advantage as an enhancement.

11. CONCLUSION

This project has shown the benefits and established how necessary network monitoring is on a network in academic and corporate organizations to properly administer and maintain it. Without monitoring, a network is a black hold and faults can go unnoticed for extended periods of time.

This project has shown that most popular network monitoring systems that are used today to monitor networks have a number of limitations and issues that prevent them from providing full testing coverage and detecting every fault. We have shown the major issues can be solved by using technology available today that was not available when these systems were originally developed.

Improved techniques for building a better network monitoring system were devised and investigated. A new network monitoring tool was built from the ground up using new technologies. Some of the new techniques devised were integrated into the new network monitoring tool to improve usability and usefulness of a network monitoring tool from the perspective of network users and systems administrators.

Lastly we also investigate the views and attitudes of network users and administrators and how these can be factored into developing network monitoring systems that address the today's network issues.

REFERENCES

[1] Ambysoft: *The Agile System Development Life Cycle (SDLC), http://www.ambysoft.com/essays/agileLifecycle.html*. Accessed June 6 2015.

[2] Colin Pattinson. A study of the behaviour of the simple network management protocol. In Olivier Festor and Aiko Pras, editors, *DSOM*, pages 305–314. INRIA, Rocquencourt, France, 2001.

[3] Jay Beale, Renaud Deraison, Haroon Meer, Roel of Temmingh, and Charl Van Der Walt. Service detection. In *Nessus Network Auditing*, page 248. Syngress Publishing, 2004.

[4] Nagios. The Industry Standard In IT Infrastructure Monitoring. http://www.nagios.org. Accessed June 3, 2015.

[5] Information Sciences Institute, University of Southern California. Internet protocol. RFC 791, RFC Editor, http://www.ietf.org/rfc/rfc791.txt,September 1981.

[6] Stephen Few. What is a dashboard? In *Information Dashboard Design: The Effective Visual Communication of Data*, page 34. O'Reilly Media, Inc, 2006.

[7] The Cacti Group, Inc. Cacti: The complete rrdtool-based graphing solution. http://www.cacti.net. Accessed June 6, 2015.

[8] The Icinga Project.Icinga: Open source monitoring. http://www.icinga.org. Accessed June 3, 2015.