Comparative Analysis of two Open Source Network Monitoring Systems: Nagios & OpenNMS

Muhammad Qadir
Muhammad Adnan
This thesis is submitted to the School of Computing at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Computer Science. The thesis is equivalent to 20 weeks of full time studies.

Contact Information:
Author(s):
Muhammad Qadir
Folkparksvagen 18:03
Ronneby 372 40, Sweden
mabidoon@hotmail.com

Muhammad Adnan
Folkparksvagen 14:11
Ronneby 372 40, Sweden
madnanmoghal@gmail.com

University advisor(s):
Jeff Winter
School of Computing

School of Computing
Blekinge Institute of Technology
SE – 371 79 Karlskrona
Sweden

Internet : www.bth.se/com
Phone : +46 455 38 50 00
Fax : +46 455 38 50 57
ABSTRACT

Context: Extensive and rapid continuous growth in Internet Protocol (IP) based networks and as the result of increasing dependencies on these networks makes them extremely challenging to manage and keep them running all the time. 24/7 non-stop monitoring is important to minimize the down time of the network. For this reason dependency on automated network monitoring has been increased.

Objectives: There are many tools and systems available for network monitoring. This includes expensive commercial solutions to open source products. Nagios and OpenNMS are two of the most popular systems and they are considered to be close competitors. Comparison discussions about them are very common at different forums on internet. But no empirical study for comparison analysis has been done. In this thesis the comparison study between Nagios and OpenNMS was conducted.

Methods: Network monitoring functionalities are listed down from literature followed by industrial interviews with the networking professionals. Then taking these functionalities as a base, to evaluate, survey studies for both systems were conducted separately and comparison analysis of the results was performed. Usability evaluation of both systems was done by conducting usability testing and comparison analysis of the results was performed.

Results: Besides providing the comparison of both systems this study also can help to find-out the strengths and weaknesses of both systems separately. And in the end we suggested a list of functionalities and features which might help in selection of a monitoring system and also might be helpful for improvement of the monitoring systems.

Keywords: Network monitoring, Nagios, OpenNMS, NMS, Fault monitoring, Performance monitoring.
ACKNOWLEDGMENTS

We are thankful to Allah Almighty who gave us everything that we have in life, including the opportunity, health and strength needed to complete this study.

This thesis would not have been possible without the guidance of our thesis supervisor Jeff Winter, he always did accommodated us to provide important feedbacks. We are very grateful to you Sir.

We are thankful to the following BTH students Kamran Talib, Yasir, Sajjad, Mansoor, Aamir, Falak, Zulifqar, Adnan, Hassan and Ali for participating in usability testing as test users.

We would like to thank Jawwad Ahmad (NetSol, Lahore, Pakistan) and Muhammad Sonain (Network consultant, Canada) to provide technical reviews and insight when needed. They also provided important references for Interviews and Surveys.

We are thankful to our friend Shahid Hussain (PhD student, Blekinge Institute of Technology) he was always there trying to motivate us and helped us to recover from downtimes. We are also thankful to all the friends for providing their companionship during these years while living in Ronneby. Without having them with us it would not have been possible to stay here this long, away from our homes and families.

It is not possible for us, to express in words, our thankfulness to our parents and families. We know during this period there were many moments when they suffered our absence very deeply but they always encouraged us to achieve our goals.

We are thankful to Blekinge Institute of Technology and Sweden for providing us this opportunity to obtain quality education. Also very thankful to friendly people in Ronneby specially the staff at the store we used to shop at.
# Table of Contents

ABSTRACT .................................................................................................................................................. 2  
ACKNOWLEDGMENT .................................................................................................................................... 3  

1 INTRODUCTION ........................................................................................................................................ 7  
1.1 NAGIOS: .............................................................................................................................................. 8  
1.2 OPENNMS: .......................................................................................................................................... 8  
1.3 NETWORK MONITORING, UNDER THE HOOD OF NETWORK MANAGEMENT SYSTEM: ........... 9  
1.3.1 Functional areas: .............................................................................................................................. 9  
1.4 USABILITY: .......................................................................................................................................... 10  
1.5 STRUCTURE OF THESIS REPORT: ..................................................................................................... 11  

2 PROBLEM STATEMENT AND RESEARCH METHOD ................................................................. 12  
2.1 AIMS AND OBJECTIVE ....................................................................................................................... 13  
2.2 RESEARCH QUESTIONS ..................................................................................................................... 13  
2.3 EXPECTED OUTCOMES ...................................................................................................................... 14  
2.4 RESEARCH METHODOLOGY ............................................................................................................ 15  
2.4.1 Overview: ....................................................................................................................................... 15  
2.4.2 Interviews: ..................................................................................................................................... 15  
2.4.3 Survey: .......................................................................................................................................... 16  
2.4.4 Usability testing: .............................................................................................................................. 16  
2.5 VALIDITY THREATS RELATED TO INTERVIEWS ......................................................................... 18  
2.5.1 Credibility: ..................................................................................................................................... 18  
2.5.2 Transferability/ Generalization: .................................................................................................... 19  
2.5.3 Dependability: ............................................................................................................................... 19  
2.6 VALIDITY THREATS RELATED TO SURVEY AND USABILITY TESTING ........................................ 19  
2.6.1 Conclusion validity threat: ........................................................................................................... 19  
2.6.2 Construct validity threats: ............................................................................................................. 20  
2.6.3 Internal validity threats: ............................................................................................................... 20  
2.6.4 External validity threats: .............................................................................................................. 20  

3 FUNCTIONAL AREAS OF NETWORK MONITORING SYSTEMS .................................................. 21  
3.1 BROADER PICTURE: ............................................................................................................................. 21  
3.2 MONITORING WITH RESPECT TO FCAPS: .................................................................................. 21  
3.2.1 Monitoring for fault management: ................................................................................................. 22  
3.2.2 Monitoring for configuration management: .................................................................................... 23  
3.2.3 Monitoring for accounting management: ....................................................................................... 24  
3.2.4 Monitoring for performance management: .................................................................................... 24  
3.2.5 Monitoring for security management: ............................................................................................ 26  
3.3 OUTCOME; NETWORK MONITORING TASKS: .............................................................................. 26  
3.3.1 Analysis: .......................................................................................................................................... 27  
3.4 INDUSTRIAL INTERVIEWS: ................................................................................................................. 28  
3.4.1 Objective: ........................................................................................................................................ 28  
3.4.2 Selection of interviewee (subjects): ................................................................................................. 28  
3.4.3 Data Collection: ............................................................................................................................. 28  
3.4.4 Transcription of Interviews: .......................................................................................................... 29  
3.4.5 Analysis process: ............................................................................................................................ 29  
3.4.6 Outcome: ....................................................................................................................................... 30  
3.5 CONCLUDING THE CHAPTER: ........................................................................................................ 32  
3.5.1 Network monitoring functionalities: ............................................................................................... 32  

4 EVALUATION AND COMPARISON USING QUESTIONNAIRE SURVEY ........................................ 33  
4.1 SURVEY AS METHOD: ....................................................................................................................... 33  
4.2 PURPOSE: .......................................................................................................................................... 33  
4.3 TARGET PARTICIPANTS: ..................................................................................................................... 33
5 USABILITY COMPARISON

5.1 PLANNING AND CONDUCTING THE TESTS:
5.1.1 Goal, objectives and measurements: ................................................................. 46
5.1.2 Test users’ selection: ......................................................................................... 46
5.1.3 Tasks: ............................................................................................................... 47
5.1.4 Evaluator: ......................................................................................................... 48
5.1.5 Lab (Test instrument) ....................................................................................... 48
5.1.6 Pre-Test: ......................................................................................................... 50
5.1.7 Conducting the Tests: ...................................................................................... 50
5.1.8 Post test questionnaire (PTQ): ......................................................................... 51
5.2 RESULTS AND ANALYSIS: ................................................................................ 51
5.2.1 Nagios: ........................................................................................................... 51
5.2.2 OpenNMS: ..................................................................................................... 55
5.3 COMPARISON ANALYSIS: ................................................................................ 59
5.4 SUMMING UP THE CHAPTER: ......................................................................... 61

6 CONCLUSION

6.1 FINDINGS: ......................................................................................................... 63
6.2 SUGGESTIONS: ................................................................................................ 64
6.3 FUTURE WORK: ................................................................................................ 67

7 LIST OF REFERENCES .......................................................................................... 69

APPENDICES ........................................................................................................ 74
APPENDIX A: INTERVIEWEES’ INFORMATION....................................................... 74
APPENDIX B: INTERVIEW QUESTIONS ................................................................. 75
APPENDIX C: SURVEY INSTRUMENT ................................................................. 77
FOR Nagios: ......................................................................................................... 77
FOR OpenNMS: .................................................................................................... 80
APPENDIX D: LAB DESIGN .................................................................................. 83
APPENDIX E: TASK DESCRIPTIONS FOR Nagios AND OpenNMS ................. 84
NAGIOS TASK: .................................................................................................... 84
OPENNMS TASK: ............................................................................................... 85
APPENDIX F: POST TEST QUESTIONNAIRE ....................................................... 86
NAGIOS: ............................................................................................................... 87
OPENNMS: .......................................................................................................... 88
Table 1: Research question with methodology to answer them ................................................ 15
Table 2: Network monitoring tasks and interdependencies ...................................................... 27
Table 3: Results of functional section of the questionnaire for Nagios ........................................ 36
Table 4: Results of usability section of the questionnaire for Nagios ........................................ 37
Table 5: Results of functional section of the questionnaire for OpenNMS ................................. 39
Table 6: Results of usability section of the questionnaire for OpenNMS .................................... 40
Table 7: Comparison of Nagios and OpenNMS based Q2 – Q11 .............................................. 42
Table 8: Comparison of Nagios and OpenNMS based on AM, AL, TH and HM ....................... 42
Table 9: Comparison of Nagios and OpenNMS based Q12 – Q16 ........................................... 44
Table 10: Hardware components for lab .................................................................................. 48
Table 11: Software used in lab .................................................................................................. 49
Table 12: Nagios average test results of first attempt ............................................................... 51
Table 13: Nagios average test results of second attempt ........................................................... 51
Table 14: Nagios Results for learnability, memorability and error rate ...................................... 54
Table 15: Nagios results of effectiveness ................................................................................. 54
Table 16: Nagios results of efficiency ...................................................................................... 55
Table 17: Nagios satisfaction; results of satisfaction from PTQ of Nagios .................................. 55
Table 18: OpenNMS average test readings of first attempt ...................................................... 56
Table 19: OpenNMS average test readings of second attempt ................................................ 56
Table 20: OpenNMS results for learnability, memorability and error rate ................................. 57
Table 21: OpenNMS results of effectiveness .......................................................................... 57
Table 22: OpenNMS results of efficiency .............................................................................. 58
Table 23: OpenNMS results of satisfaction from PTQ of OpenNMS ........................................ 58
Table 24: Comparison based on Nielsen’s definition .................................................................. 59
Table 25: Comparison according to ISO’s defined attributes + summative usability ................. 60
Table 26: Summative usability comparison of Nagios and OpenNMS ........................................ 61
Table 27: Mapping of suggestions to the part of study, on which they are based on .................. 67

Figure 1: Research methodology flow ...................................................................................... 18
Figure 2: Seidel model of QDA .............................................................................................. 29
Figure 3: Functional section of the questionnaire for Nagios .................................................. 37
Figure 4: Results of usability section of the questionnaire for Nagios ...................................... 38
Figure 5: functional section of the questionnaire for OpenNMS .............................................. 40
Figure 6: of usability section of the questionnaire for OpenNMS ........................................... 41
Figure 7: Comparison based on functionalities ....................................................................... 43
Figure 8: Comparison of Nagios and OpenNMS based on real-time monitoring functions .......... 43
Figure 9: Comparison of Nagios and OpenNMS based on Usability part of survey ................. 45
Figure 10: Physical Network .................................................................................................... 49
Figure 11: Virtual Network ...................................................................................................... 50
Figure 12: Comparison based on Nielsen’s defined attributes ............................................... 60
Figure 13: Comparison according to ISO’s defined attributes + summative usability ......... 61
1 Introduction

In 1985 Denning described computer science as "the body of knowledge dealing with the design, analysis, implementation, efficiency, and application of processes that transform information" and suggested that "What can be automated?" is the fundamental question underlying all of computing [1], [2]. But since 1980s areas of computing research have been gradually increasing from machine and automation towards how and where computers are used, the activities of end users and how end users collaborate and interact [2]. And now there are a range of topics that falls under the discipline of computer science. CSAB (formerly known as Computing Science Accreditation Board) includes following areas in computer science domain: software engineering, artificial intelligence, computer networking and communication, database systems, parallel computation, distributed computation, computer-human interaction, computer graphics, operating systems, and numerical and symbolic computation [3]. Denning has described the following subareas of computer science: algorithms & data structures, programming languages, architecture, operating systems and networks, software engineering, databases & information retrieval, artificial intelligence & robotics, graphics, human computer interaction, computational science and organizational informatics [4]. Computer-communication networking is an important field of computer science, in ACM (Association of Computing Machinery) classification system this field is appeared in section c.2 with further subcategories, section c.2.3 contain network monitoring and network management both [5], these are the concern fields of computer science in this study.

More-or-less all current computing and IT (Information Technology) is dependent on IP (Internet Protocol) Networks. Computers no longer exist as standalone machines, computing technology is used to communicate, there is now almost no place on earth that you cannot communicate with via this technology [6]. Network communication is the base, whether setting up organizations having offices around the globe and working as a single unit, sharing and exchanging data within or with other organizations, setting up IT services like data-centers and network-storage. Technologies like grid computing, cloud computing and virtualization and many more are based on IP networks. And not limited to computers, several other services and devices are based on these networks for example using IP Networks as carrier for voice, this changed dramatically the way of telephonic communication and now it is the most feasible solution for long distance calls due to cost benefits over traditional circuit switched networks, gaming consoles using IP Networks for online gaming. IPTV is another example.

As a result of this rapid boom in networked technologies the networks are getting big, complex and more and more heterogeneous. Consequently this bring challenges to the field of Network Management because the services for which organizations and/or individuals are paying can’t be compromised at committed quality of service, and needs 0% (virtually) down time. Therefore it is now near to impossible to maintain/manage network(s) without Network Management System. A Network Management System may consist of a combination of different software and hardware depending on the vendors and the requirements of your network. Network Monitoring System (NMS) is one of the most important modules [7-10]. Network Monitoring Systems help us to address fully or partially almost all functional areas of a network management system [11], [12] that are; according to International
Organization for Standardization (ISO): Fault management, Configuration management, Performance management, Security management and Accounting management [11], [13-15]. A monitoring system can help the network managers by detecting faults before any complaint from user, performing proactive fault monitoring, doing performance monitoring for optimization, assist in efficient application of resources [15], monitoring the resources accessed; for security reasons, etc [8].

The purpose of this thesis work is to perform evaluation and comparison of two open source network monitoring systems: Nagios and OpenNMS. Above discussed developments increased the competition between different NMS, including the systems developed by market giants like Oracle and HP, but the systems developed by companies are very expensive and also restricted in use according to the license agreements. As a result the systems developed and contributed by open source communities are also getting popular. Here we are analyzing two among most popular open source systems Nagios [16] and OpenNMS [17].

The reason to select Nagios and OpenNMS is that they both are among best available open source systems and known to be enterprise level monitoring solutions, they both considered to be highly scalable and also used in different research studies [18-23]. It is very common debate on different web forums and on network & open source communities about different comparisons between them. Both are winners of multiple awards [24], [25]. Even competing for open source awards like Nagios won Infoworld’s Best of Open Source Software (BOSSIE) in 2008 in server monitoring category [26] and in 2009 it is among 3 winners of BOSSIE award in the category Networking and network management, OpenNMS also was the winner of same award along with Nagios and Cacti [27]. OpenNMS won BOSSIE again in 2010.

1.1 Nagios:

Launched in 1999, initially known as Netsaint, Nagios is a powerful open source monitoring system that helps organizations to discover and resolve IT infrastructure problems before they can affect critical business processes. When something wrong happens, Nagios alert technical staff with necessary information regarding problem, allowing them to resolve the problem before it affect business processes, end-users, or customers [28].

Nagios compete well against most of the commercial applications, and in the opinion of Schubert et al. in most cases Nagios have a lower cost with a higher level of effectiveness than many commercial applications available for monitoring purpose [29]. Nagios uses modular approach, the core Nagios application works with different modules for different functionalities. This makes it very flexible and scalable.

1.2 OpenNMS:

OpenNMS started in 1999 claim its self as an enterprise level, highly scalable, open source, network management system. The claim is that OpenNMS is currently managing a network with nearly 70000 devices [30]. Known for monitoring IT infrastructure and perform auto device discovery, performance measurement, service monitoring and event management.
The back-end (management server) is implemented in Java as a multithreaded application and front-end (user interface) consists of number of Servlets and Java server pages (JSPs), and perform network monitoring, service availability, generates performance reports and provide assets management capacity [31].

1.3 Network monitoring, under the hood of network management system:

Network Management has become a very complex task to perform due to the rapid growth of networks in size and complexity [32]. Complexity involves the heterogeneous nature of today’s network; which is not only means sharing and transmitting data but we are using these networks for voice and video services like IP telephony, videos streaming and now IP TVs [33] are getting popular and replacing satellite dishes. Moreover, the mesh of transmitting media that can be involved in a network makes it more complex. For example, different cable technologies, point to point radio links, Wireless hotspots and mobile services with 3G all are working together in networks. IP networks are replacing many technologies and adding the facilities in combination with other IT and software technologies one never could think about a decade or two back. These days IT services are not luxury or limited to some big organizations these are the part of daily life activities like learning management systems, health management systems and online banking. So networks are everywhere from small businesses to large companies, public or private organizations to government institutions, even in homes from kids playing games over the networks or house wife surfing internet for a recipe to cook. When dependencies on these services are that much increased, it results in providing better and better quality. And now the standards in the mind of end users are set, they need zero downtime. As a result of these developments Network Management becomes a highly technical, complex and demanding field [34], and it is not possible without using high end automated tools and systems.

Network management is not expected without Network monitoring because monitoring is the most important part of network management [9][10] which provide necessary data about network, this data reveals the information about network’s infrastructure, health and also used for most of the network management tasks [11][12]. You can’t wait for an end user to send you a complaint about any malfunction.

1.3.1 Functional areas:

The functionalities or tasks that a Network Monitoring System should perform can be outlined and categorized according to the five functional areas of Network Management also known as FCAPS. International Organization for Standardization (ISO) has defined and standardized a model containing five functional areas for Network Management: Fault Management, Configuration Management, Accounting Management, Performance Management, and Security Management [11],[13], [14]:

- **Fault Management (FM):** is related to detecting, isolating, reporting, fixing and logging the errors in the network.

- **Configuration Management (CM):** deals with controlled-configuration of network and network resources. For examples defining and following proper
way of configuring, changing, adding and removing of network devices and services.

- **Accounting Management (AM):** is concerned to user management i.e. allocating the resources to the end users e.g. assigning storage space at sever or defining limit on network bandwidth. It also deals with accounting and billing of network resources and services usage.

- **Performance Management (PM):** is concerned to keep the network performance at maximum according to requirements. For example managing Quality of Services (QoS), resources utilization and avoiding problems such as congestion, jitter and packet loss.

- **Security Management (SM):** aims to protect entire networks from malicious activities and to make sure the security of resources (devices and services) and data (stored in machines or traveling over the network).

### 1.4 Usability:

Usability is an important attribute to study when evaluating and comparing any software products. It is desirable to understand usability in context before further going into the study. So we can learn about the components of usability on which we can base the usability part of our study.

A simple way of defining the usability of a software product is: how easy it is to learn and use, what is the level of productivity (efficiency) achieved by the user using the software and how much support is required by the user [35]. It is important to find a well established standard usability definition. Generally speaking, to define a standard, means setting some rules and guidelines by an agreed-upon organization to perform (design, develop, deploy, test, etc) something. Standards help to maintain consistency [36]. So it is important to look for a standard which defined usability and is well established; to evaluate and compare understudy systems accordingly. Some attempt has been made to deliver standard by ISO/IES standards, specifically for interface components but mostly were ignored against industrial standards but the ISO 9241 standards are the one that have had impact on industry [36]. Usability defined in ISO 9241-11 is: The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use [37], [38]. According to ISO 9241-11:1998, it is necessary to identify goals and decompose effectiveness and satisfaction and components of the use context into subcomponents with measurable and verifiable attributes when specifying or measuring usability [38]. The goal, tasks, users and context of use is discussed in later section 5.3. Here our purpose is to establish the components of usability that can be measured. There is general agreement from the standards boards that the dimensions of usability are effectiveness, efficiency and satisfaction [39]. As described above it is necessary to decompose effectiveness and satisfaction so these components can be measurable. Effectiveness includes measures of the completion rate and errors, efficiency is measured from time on task and satisfaction can be concluded by using satisfaction related questionnaire either collected on task-by-task basis or at the end of a test session [39]. Completion depends on learnability and memorability of the software. When a new user using a system it depends on learnability of the system to complete the task and when an old user using a system it depends on memorability of the system to complete the task. If memorability of a
system lacks users may go through learning again and again. According to Rubin and Chisnell Learnability, memorability and error rate are the components of effectiveness [40]. Jacob Nielsen also describes learnability and memorability as components of usability, according to him usability is associated with five elements: learnability, memorability, error rate, efficiency and satisfaction [41]. Different studies such as [11], [42] discussed above mentioned attributes to understand usability.

1.5 Structure of thesis report:

This section is about the structure of this report, the chapters, their arrangement (sequence) and a brief overview of each of them, starting from chapter 2.

Chapter 2 – Problem statement and research method

The discussion is to identify the problem, present motivation to take it as a research study. Define goals and objectives towards the solution of given problem, in the process the research questions are developed. Overview of research methodology is presented. Later in the chapter validity threats are discussed.

Chapter 3 – Functional areas of network monitoring tools

To find out that which functionalities a network monitoring systems should perform, first a literature study is presented followed by interviews with the experienced network management and monitoring experts.

Chapter 4 – Comparison using questionnaire survey

To evaluate both systems: Nagios and OpenNMS, on the basis of outlined functionalities from chapter 3, survey study has presented in this chapter with a comparison analysis.

Chapter 5 – Usability comparison

This chapter contains usability performance measurement testing to evaluate and compare usability of Nagios and OpenNMS.

Chapter 6 – Conclusion

Overall results and finding with suggestions are presented.

Chapter 7 – List of References

Appendices
2 Problem Statement and Research Method

Data-communication networks is an important field in computer science [3], [4]. Computers no longer exist as standalone machines, computing technology is used to communicate, there is now almost no place on earth that you cannot communicate with via this technology [6]. Latest computing technologies like cloud computing, grid computing, virtualization, multiprocessing and many more are dependent on data networks. ACM classification system mentioned computer-communication networks in section c.2 and subsection c.2.3 contains network monitoring [5].

Networks are getting bigger, complex and heterogeneous due to increased dependency of latest computing and technologies on them. These network based technologies and services are big part of our routine life like learning management systems, e-health, online banking and online ticketing are few names. The users don’t want to compromise on the quality and availability of the services they are using. Due to this evolution, the complexity of networks has increased; therefore network management requires a great deal of attention [7]. Network management refers to activities associated with running a network, and a big part of which is monitoring [7]. There are many things which can cause a network to fail it doesn’t matter whether the reason is malicious or accidental only two things can save the system from downtime: redundancy and monitoring [8]. Network monitoring is the important part of network management [9][10] which provide necessary data about the network, this data reveals the information about the network’s infrastructure, health and is also used for most of the network management tasks [11][12]. Network monitoring is known to be a major building block for many domains in communication networks and has become an important research area since the development of the first computer network [43].

Due to discussed importance of network monitoring it is understandable that how important it is to select a network monitoring system that can fulfill your network’s requirements. When it comes to implementing a network monitoring system (NMS), there are multiple options available to choose between. Due to the importance of network monitoring it’s easy to see why big corporations and governments purchase monitoring software with six-figure (US Dollars) price tags [8]. Small- to medium-sized businesses and universities can have complex networks even more complex then big corporation e.g. with their geographically dispersed campuses and satellite offices, network monitoring can be a challenge, but they don’t have the luxury of high-priced monitoring system [8]. Fortunately, to fulfill their network monitoring requirements there are some well developed open source network monitoring systems available. Some time big corporations also prefer open source system due to the flexibility, these system can integrate other open source module and also gives you the option of developing your own modules or plug-ins by making available the source code.

Implementing a network monitoring system is not an easy job. If implemented correctly it can be your best friend and help you uphold your service level agreement (SLA) by helping to solve problem before any one knows about or effected by the problem [8]. It takes time and effort to implement monitoring system that can bring stability to the environment. This is why it is very important to select a system that is most suitable for your requirements from beginning. In this study we’ll perform an empirical evaluation of two widely used OS monitoring systems: Nagios [16] and
OpenNMS [17] and comparison analysis will be performed based on empirical results. The reason to select these systems is that they both are among best available open source systems and known to be enterprise level monitoring solutions, they both considered to be highly scalable and also used in different research studies [18-23]. It is very common debate on different web forums and network communities about different comparisons between them. But the current available information is not enough to help one to decide which system out of these two would be most fit for any particular network needs. The available material is about what functionalities they can perform and how to use (apply) them in different network conditions but nothing about how well they are performing these functionalities, users’ satisfaction regarding these tasks.

2.1 Aims and objective

This study is to evaluate two network monitoring systems Nagios and OpenNMS against each other on the basis of how well they can perform the network monitoring, plus usability evaluation and comparison is also performed. While performing comparison study the goal is to make suggestions which can serve as a guideline to improve the functionalities and usability of network monitoring systems and also can help in selection between available monitoring systems. To achieve this aim following are the objectives:

- Identification of functionalities (tasks) which should be performed by a Network Monitoring System performs. This is done by literature review.
- Crosscheck, validate and update the outcome of above said objective by conducting interviews with the experts which are performing network monitoring related jobs in industry.
- Investigation and comparison of how well Nagios and OpenNMS address the functionalities by conducting and analyzing close ended survey. Survey designed with same question for both systems and performed separately with the experts of respected system, means survey for Nagios is conducted with the experts using Nagios and survey for OpenNMS is conducted with the experts using OpenNMS.
- Designing lab environment using virtualization (Virtual Box [44], Vyatta [45]) for usability testing.
- Usability comparison by conducting the tests, comparing and analyzing the results.
- Outline the suggestions based on the analysis of all empirical results of the study.

2.2 Research questions

Following Research questions (RQ) will be answered:

RQ1: What are the functionalities a Network Monitoring System should perform?
RQ2: How effectively Nagios and OpenNMS perform the network monitoring functionalities?
RQ2.1: What is the comparison of the network monitoring functionalities of these systems?
RQ3: Which of the above systems is better in terms of usability?
RQ4: Which functions and features should be taken under consideration while selecting a network monitoring system?

By answering RQ1 we will try to outline all important network monitoring functionalities, this is important because there is no standard monitoring functionality list available at the moment like functional areas FCAPS [11], [13] of network management from ISO. This list can provide initial understanding about what is important in network monitoring to the people who need to implement the monitoring system.

While answering RQ2, it is not only the comparison but one can learn from the evaluations of both systems how well these two OS systems can perform monitoring tasks, and not every organization needs to spend six figure (US Dollars) for their monitoring requirement.

It is very important to answer RQ3, because when you evaluate a computer system/software one essential thing to look at is how the user interacts with the system and how comfortable and satisfied he is while performing the tasks. As Shneiderman has argued that computing has changed its approach, the old computing was about what computers can do but new computing is about what users can do [46]. The focus of computing research has broadened from machine and automation towards how and where computers are used and how the end users interact and collaborate [2].

By analyzing the outcomes while trying to answer these questions the goal is to make suggestions which can serve as a guideline to improve the functionalities and usability of network monitoring systems and also can help in selection between available monitoring systems and this will be the answer to RQ4.

2.3 Expected outcomes

Following are the expected outcomes (EO) of this study:

EO1: A list of network monitoring functionalities included their inter-dependencies on each other.
EO2: Update and validate EO1 by conducting interviews with network management experts working in industry.
EO3: Survey results and comparison shown as graphs for both systems. Graph can be helpful for visualizing and analyzing the results.
EO4: Virtual lab environment design for usability testing.
EO5: Results and comparison analysis report of usability testing for both systems.
EO6: List of suggestions that can serve as guideline to improve the functionalities and usability of network monitoring system, this guideline also can help in selection between available monitoring systems.
2.4 Research methodology

2.4.1 Overview:

According to Creswell [47] there are three types of research methodologies i.e. qualitative, quantitative and mixed method. To answer the research questions and achieve the final goal of our research, we employed both research types i.e. qualitative and quantitative at different phases according to requirements.

Qualitative approaches were used, first literature review to extract the required information from books, papers, article and web sources and then conducting interviews with the experienced professionals of the domain working in industry related to network management jobs, to verify and crosscheck the finding of literature review. In combination qualitative method was used to answer RQ1.

Structured survey was designed on the basis of outcome of RQ1. The survey study was performed for both monitoring systems separately, by using same survey instrument for both. The same questions were asked to the networking professionals using Nagios about Nagios and to the professionals using OpenNMS about OpenNMS. Qualitative analysis of the results was performed, compared and displayed using graphs. This process helped us to find that how well the outlined functionalities (finding of RQ1) are performed by the both NMS, comparison analysis of the results were performed. This process helped us in answering RQ2.

We had some data to answer RQ3 as a result of survey part but we preferred to perform usability testing, trying to find out unbiased results. For example, when asking a professional how effectively or efficiently he can use the system that he is using for long time, he might answer towards positive because he is used to work with that system. For this purpose a lab environment was developed, performance measurement usability test [41] (for details of performance measurement testing reference [41] page 192) was designed and conducted for both NMS. During this phase first we had performed literature review to find the appropriate usability testing method and what is needed to be test.

Below table 1 is mapping the methodology to answer each research question:

<table>
<thead>
<tr>
<th>Research Question</th>
<th>Method(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RQ1</td>
<td>Literature review + Interviews with the professionals in network management domain</td>
</tr>
<tr>
<td>RQ2</td>
<td>Quantitative survey (Instrument designed based on results of RQ1)</td>
</tr>
<tr>
<td>RQ3</td>
<td>Performance measurement usability testing (quantitative analysis)</td>
</tr>
<tr>
<td>RQ4</td>
<td>List of suggestions based on literature review, interviews, survey and usability testing</td>
</tr>
</tbody>
</table>

2.4.2 Interviews:

There can be different types of interviews, in [48], [49] three types of interviews are discussed:
1. Structured interviews: In this type all the questions are well prepared by the interviewer and are very specific.

2. Unstructured interviews: In contrast to structured, not very specific questions are asked, instead a theme is suggested by the interviewer with very few questions. This results into very open ended questions.

3. Semi-structured interviews: These are the combination of other discussed two types.

For our study, semi-structured interviews were conducted for the data collection about network monitoring tasks from the professionals in this field. This inquiry method helped the researchers to have a full detailed answers and opinions from the interviewees by allowing open ended questions, while focusing and following the required subject matter closely with specific questions, as semi-structured interviews are also called [49] focus interviews. More details are presented in section 3.4.3.

2.4.3 Survey:

The methodology is selected because to test both systems in lab experiment needs huge resources in form of equipment and time. And by using survey method we can investigate from professional users. And for this we need to get information from as many professionals as possible so questionnaire based survey is better option than interview. According to Wohlin et al. selection of empirical strategy depends upon the purpose of evaluation and conditions for the empirical investigation, a survey is often used for investigation when a tool or technique has been in use for a while [50]. As the systems, OpenNMS and Nagios are well established and have been in use for years by the network management professionals, for this reason, to evaluate both systems, survey study was performed. For each system separate survey was performed with users of the concerning system. Both surveys were consisted of the same questions.

2.4.4 Usability testing:

Usability evaluation methods can be categorized generally into two types: 1) Analytic evaluation involves modeling and analysis of system’s features and their implication of use 2) Empirical evaluation involves observation or other data collection from system users [51]. Both categories contain many different methods for usability evaluation. According to Nielsen [52] four basic ways are: automatically by using some computer program, empirically by involving the real users, formally by using exact formulas or models to calculate usability and last is informally by using rules of thumb and skills of the evaluator.

For this study we decided to do empirical evaluation. To evaluate user interfaces empirical evaluation is considered to be one of the main procedure with user testing most frequently used method [52]. The fundamental usability method is User testing with real users and it can be considered irreplaceable in some sense because it provides direct information about how people use computers and what are problems with the interface being tested [41]. Dumas and Redish describe five characteristics of a usability test: 1) primary goal to improve the usability of a product (more specific goals can be achieved e.g. how easy to use for a user) 2) The participants represent real users 3) real tasks should be done by the participants 4) you observe and record what participants do or say 5) analyze the data and diagnose the problems [53]. After usability testing and analyzing the results of Nagios and OpenNMS, we
need to conclude a comparison between them with respect to effectiveness, efficiency and satisfaction. One reason to select usability testing approach as discussed by Dumas and Redis usability testing can be used to understand strength and weaknesses of a product separately or in the comparison to another product [53].

There are multiple techniques available for usability testing. Some are: coaching method, co-discovery, performance measurement, retrospective testing and thinking aloud [41]. We used performance measurement as the testing technique for this study and followed by questionnaire. This selection is discussed in following subsections.

2.4.4.1 Performance measurement:

In performance measurement users’ performance is measured while performing the given set of experimental tasks by a group of test users and collecting time and error data [41]. According to Preece et al. [54] (p. 341) usability testing involves measuring typical users' performance on carefully prepared tasks that are typical of those for which the system was designed while users' performance is generally measured in terms of number of errors and time to complete the task.

To compare different designs, measuring users’ performance has been the base of usability testing [54], similarly according to Nielsen measurement studies has its importance for comparing usability of competing products [41]. Here the broader goal of this usability study is to find out: which of Nagios and OpenNMS is better in usability in comparison to other. This testing method helped in measuring effectiveness, efficiency and satisfaction which are required parameters of usability, of both systems and then these results helped to perform the comparison.

The following are some attribute of performance-measurement-usability-testing, found from above mentioned studies: performed under controlled environment, results in quantitative data, interaction between users and evaluator should be minimized (because it can affect the quantifiable data), performed either in lab environment with sample users or measure the required task by observing the users at work.

Usability testing, performance measurement technique is used to measure two usability components i.e. effectiveness and efficiency of both systems Nagios and OpenNMS.

2.4.4.2 Questionnaire:

To measure user satisfaction, after performance measurement test users were asked to answer a questionnaire to know about their subjective satisfaction. From the viewpoint of a single user these question may result into subjective output but when replies from multiple users are averaged together, results into an objective measurement [41].

Flow of our research process can be seen in figure 1 on the following page.
2.5 Validity threats related to Interviews

To improve the accuracy of research, it is important to find out the possible threats to the research methodology so they can be contained. Egon G. Guba discussed following four aspects of trustworthiness in naturalistic inquiry (in brackets their scientific counterparts): Credibility (Internal validity), Transferability (external validity), Dependability (reliability), and Conformability (Objectivity) [55].

2.5.1 Credibility:

Credibility of a study can be affected by personal bias or perception of researcher as well of respondents. To minimize this threat we had tried to prolong our engagement with the interviewee, started slowly with some back ground information and asked them general questions initially to develop a comfort level between us and interview. We tried to prolong the discussion on every question so that we can minimize our own perception by having enough output to understand interviewee answer, also we tried to avoid over involvement. Another threat that can hurt the credibility of the study is that different perspective of regarding different data sources can be made by different inquirers. To minimize this threat we have recorded the interviews and then
both investigators analyzed them separately, then we compared and discussed our finding towards end results.

2.5.2 Transferability/ Generalization:

As described by Guba [55] that Naturalists avoid generalization that virtually all social/behavior phenomena are context bound ad it is not possible to find a truth statement that have general applicable. But our study is not the case to studying any typical social/behavior phenomena but our interview study is context bound that is to find out network monitoring functionalities and the output can be used in other words can be generalized to this context. To minimize the threat to incorrect generalization the subjects were selected carefully. We contacted the well known companies with good IT infrastructure as a result all interviewee were network professional with at least 4 years of experience and working in big organization.

2.5.3 Dependability:

To increase dependability and minimize the instability threat, which may occur as a result of changing realities we have used two methods to find the results: Interviews and Literature studies. Guba describe this approach as overlap methods. And because we got the results from these methods validating each other so we can argue that we had stability in our results [55].

2.6 Validity threats related to Survey and Usability testing

For these studies, four types of validity threats are conclusion validity, construct validity, internal validity and external validity [50].

2.6.1 Conclusion validity threat:

These threats are concerned with the issues that affect the ability to draw the correct conclusion [50]. One important threat in this category is reliability of measures it depends upon different factors like poor question wording, bad instrumentation or bad instrument layout [50]. To restrict this threat we cross check our survey questionnaire with two professionals of the under study domain. These two were experienced network managers and performing network monitoring as a part of their jobs. Both were not included in original survey study. For usability testing two pilot tests were performed before original testing and appropriate changes were made in instruments. These pilot tests were performed with two students of MS-CS both had studied TCP/IP and had some knowledge of networks. These two participants were not included in final testing.

To avoid ‘random irrelevancies in experimental setting’ [50], during experiments we manage a controlled environment without permitting any disturbance from inside or outside the lab.

Another threat can be, if the participant group for a study is very heterogeneous, there is risk of variations in the results [50]. To avoid this risk, for interviews and survey questionnaires we selected subjects having above 4 years of experience in Network management domain.

As discussed in [50] selection of the subjects should be close representation of the population. In our usability testing it was not possible to hire network management...
experts so to make sure that the participants should be as close representative of population as possible we selected students of Master’s having studied these two subjects: Network Management and TCP/IP subjects. This also serves us better because we need to test these systems with new users but with knowledge of networks and network management.

2.6.2 Construct validity threats:

Construct validity threat caused by using inadequate definitions and measures [47]. During our literature study we selected only reliable database and standards.

According to Claes Wohlin evaluation apprehension is a validity threat, it means some people are afraid of being evaluated, and tend to answer towards better [50]. To avoid this threat during our survey studies, cover letter containing information about study and assuring them that we are not evaluating the users but this study is to evaluate systems. And their demographic information will not be disclosed.

Another potential threat in this category was restricted generalizability across the construct [50], it means if in comparison one product is better in productivity may be it is lacking in maintainability. To strict this threat we tried to measure almost all aspects of the systems.

2.6.3 Internal validity threats:

Internal validity threats are based on some problem in defining experimental procedures, treatments or experience of the participants [47].

Internal validity threats include improper selection of the sample or study material, incompetency of the researchers. To reduce this kind of threats we studied material properly for every phase of the study e.g. survey designing, proper subjects were selected for the interviews (details are discussed in section 3.4.2) and survey (details are discussed in section 4.3). Experiment was designed properly according to the usability testing guidelines presented in [41], [53], [54], [56] (details are discussed in section 2.4.4 and 5.1). Pilot tests were performed for survey and experiments, as discussed in 2.5.1.

2.6.4 External validity threats:

External validity threats are the conditions that which can lead researcher to incorrect generalization of the study [47], [50]. Means the study performed on sample which is not real representative of actual population to which the results will be generalized or in case of an experiment the settings and material is not relevant to industrial practices [50].

To reduce these threats during our study subjects were carefully selected. The survey participants were network professionals plus working on one of under study systems. Setting for experiments were performed carefully, by using state of art technology of virtualization a replica of real wide area network was created.
3 FUNCTIONAL AREAS OF NETWORK MONITORING SYSTEMS

The purpose of this chapter is to find-out the functionalities of network monitoring. In other words outline the tasks which should be addressed by a network monitoring system/tool. These outcomes will serve as the base to develop criteria to perform the comparison study of both tools; Nagios and OpenNMS. This purpose is solved by conducting literature study and then by conducting interviews with the experts performing network monitoring duties in different companies.

3.1 Broader picture:

The objective of monitoring the network is obvious i.e. to help keeping the network at optimized level. Network Monitoring System (NMS) acquires the required up-to-date information of network for Network Manager/Administrator. This information can be the statistical data about network traffic, network links, devices and hosts and can be reported in different way as configured, like on monitoring dashboard, to an email address to a pager or mobile phone [57][58]. These monitoring reports can be of two types:

1) Statistical data of resources’ utilization: for example the bandwidth utilization, CPU utilization, access rate of specific services, devices, etc over a period of time or on a specific time. These reports are not of emergency type and so usually displayed on monitoring station. These reports are used for efficient future planning and configuration of the network. One example can be, if the bandwidth of a link is reported never to be reached more that 25% then why paying for that much bandwidth on contrary if the link is always about 80% plus in use then we know it can be a problem in future, specially if users at that link are expected to increase.

2) Alerts: these are the messages sent by the NMS when immediate attention is needed to a matter for running the network and all its services smoothly. According to the configuration the NMS sends specific messages to the resources (hosts, switches, routers and other) over the network at a regular interval of time set by admin, to check the status. If any of these resources are to be found down or running in critical stage it returns an Alarm message and the NMS sends the Alert messages to multiple destinations beside monitoring stations e.g. Alert message to email, mobile phone, and pager. Examples of this situation can be physical link disconnected, switch or router or server down, overheating of CPU, Bandwidth bottleneck.

3.2 Monitoring with respect to FCAPS:

The implementation of a Network Monitoring System provides the required data and reports that help to perform Network Management functions efficiently and effectively [59]. Below, narrowing down to identifying the tasks of a Monitoring System under the functional areas, this may also help to categorize them at the same time. And the reader can see that to what extend these areas are addressed by the
Network Monitoring System. Obviously some are more dependent than others on Network Monitoring System.

3.2.1 Monitoring for fault management:

Fault, generally speaking, is an unpleasant attribute to have in anything even in our daily life. Now with the rapid growing dependencies on IT systems and services network manager or administrator can’t wait for a call from a user to tell him about some kind of service failure or defect and then he or she start looking for what is wrong. This brings need of automated system that can monitor the Network for identifying and reporting the defect. This will allow recovering even before the users experience trouble or if they do experience then helps to keep it for minimum duration [7]. In literature at many places like [57], [60], [61] the reader can find monitoring is all about fault detection and resolution, therefore this can be arguable that fault detection and resolution has the biggest share of a Network Monitoring System.

Following are the five major responsibilities of a Monitoring System toward Fault Management (FM) in the light of these studies [11], [7], [60], [62]:

1. **Alarms:** Continues monitoring should be performed in such a way that can detect immediately whenever there is any occurrence of malfunctioning in the network. To achieve this functionality Alarms are used. Alarms are the type of event messages that have been triggered from the network to the Monitoring station. As the name suggests these event messages need most immediate attention. Alarm messages can be about anything that can affect badly the services provided by the network at a critical level. Some examples can be: a router or switch detects that suddenly a link has been down, over heating of a device, unexpected shutdown of a machine, drop down in a link quality to a critical level, suspecting an unauthorized access, etc. Some problems in the network like packet loss and bandwidth utilization should not set off the Alarms unless reached to a defined critical state, these kinds of problems should be reported to monitoring station in other ways and by taking appropriate measures about them will help the performance maximization. Setting up Alarms is a complex matter and if not rightly configured can result into false Alarms.

After receiving the Alarm(s), Monitoring Station sends Alert massages to concerned personnel like Network Admin or Manager, on their phones, mobiles communication devices or emails according to the configuration. This means the problem don’t have to wait for working hours or availability of network admin in the office premises.

Some time Alarm management is referred as an interchangeable term to fault management [7], this shows the importance of Alarms in the overall process. Rest of the fault monitoring mainly depends upon Alarms too.

2. **Diagnosis:** Is the process to find the root cause of the fault. Multiple Alarms can be triggered due to a single problem e.g. if a link is down and more than one devices or services are not being able to be accessed this can lead us into multiple Alarms, at-least one for each of them.

Monitoring System can help in this regard by creating interlinks and dependencies (mapping) between different connections, services and Alarms.
Viewing and analyzing on Monitoring station with the highlighted areas where the problems are indicated by the Alarms on the topology map of the network, can be helpful in diagnoses.

3. **Trouble Ticket:** This process is used to organize and keep track of the problems occurred in network towards their resolution. A trouble ticket is opened after detecting the Alarm. By opening it, the concerned person acknowledges that he received the Alarm, insert problem description and attributes like ID of related equipment, services effected, users effected and level of impact and then assigned to the operators who are responsible to solve the problem. Tracking can be done about which problems are solved and which are still under process by monitoring the tickets. Automated messages can be sent if a ticket is not closed within its due time. Moreover, the system helps by communicating a problem with its history and all the information, between multiple professionals.

4. **Logging:** Effective logging is also important, these log reports can maintain a history of every event and can be helpful to find the reason for any problem by relating it to same kind of problems occurred previously.

5. **Proactive detection:** As the name suggests, an automated Monitoring System can predict faults before their occurrence by using some algorithms. This functionality is limited because normally the monitoring or management stations have not much power to do computing for large network and large amount of data. Therefore this process usually depends on your performance monitoring and configuration monitoring. For example lack in performance of a critical resource can grow into a fault or changing in the configuration of a device without using recommended procedure can result in a fault or security laps.

3.2.2 **Monitoring for configuration management:**

In the above section as we discussed that FM its self sometimes known as monitoring, likewise when talking about configuration management (CM) this may be the most important part in Network Management overall [7], it starts right from the beginning to setup of the network. Without proper CM the tracking of what is deployed and how it is working can’t be done efficiently. And it is also crucial in performing some other Network Management Functions e.g. in FM it is very important to have knowledge of configuration of the network to diagnose the problem [7]. CM is also to assure the strict control over the process of configuring the services or devices [60], this will provide the mechanism that will only allow changes to be made by authorized persons and also maintains track of the changes [62] as well the previous version of configuration can be restored if needed.

CM can be divided in following sub sections [11], [7], [62]:

- **Configuration of managed resources:** How to configure devices or services in the network. This is about how to send instructions or commands to the managed equipment to update the configurations. This may be configuring a single device or interface or setting up the configuration of multiple devices at once in a big network environment.
• **Inventory management:** This includes discovering (auto-discovery) the devices in the network and maintaining the topology map. This process is helpful to optimize performance and fault management.

• **Back-ups and versioning of configuration:** By follow a controlled procedure of changing configuration of network and devices, it helps to maintain the previous versions of the configuration and keep track of the changes. This allows in case of a bad configuration to restore the network at a previous point.

• **Software management:** This task is about controlling the procedure of installation, updating and configuration of software or services. Like scheduling the updates for; operating systems, antivirus and other patches that must be updated regularly, can’t be left to the end users.

From this sub division most of them do not fall in the monitoring system’s domain. Network management has two parts: one is monitoring and the other is controlling [11]. Major portion of the CM belongs to the controlling part and for this purpose needs a specialized tool for CM. Although, few functionalities can be fulfilled using NMS like Inventory Management. NMS generates topology maps and display them in different ways, also do auto discovery of the devices.

### 3.2.3 Monitoring for accounting management:

Services provided over the network mean that the service provider needs to generate revenue from the users. This means the service provider needs to put limits on usage according to billing and the amount of service purchased [11]. These kinds of user and billing related issues are managed by an Accounting Management (AM) module. Even the service provided is not for market but for the employees, the IT department can perform AM to avoid resource exploitation. AM includes e.g. bandwidth checks, storage limitation, time specific limitations, user’s billing, resource consumption records [7]. In case of shared resource this helps in fair distribution between users and maintains the Service Level Agreements (SLAs), function (fair distribution) depend on performance monitoring to conclude which users are accountable for the specific activity [13].

AM also mainly consists of controlling functionalities and normally is performed by specialized tool. Some data can be gathered by the monitoring system e.g. bandwidth and storage utilization.

### 3.2.4 Monitoring for performance management:

Performance is not about what functionalities or services are provided but what qualities these services are provided at. The objective of Performance Management (PM) is to maintain the performance level as required, and PM attains this goal by doing Traffic & QoS management, which helps to maintain the Service Level Agreements (SLAs) between service provider and client [11]. PM can be divide in two parts first is monitoring the network and second is tuning (controlling part) the network, for performance [7].

PM is heavily dependent on monitoring, and performance monitoring should measure at-least the following:

• **Throughput:**

---

24
Throughput can be defined as measuring the traffic units to be transmitted in unit time. While these traffic units are to be measured different units can be used at different Layer levels. For example, the transmission units can be bytes or octets transmitted per second at data link layer. Similarly, at network layer number of packets routed per second can be employed, whereas at application level it can vary depending upon the network service e.g. for web services, number of web requests that are processed [7]. This process helps to calculate the actual network usage. The service provider can obtain the snapshot of the whole process during any point in time for carrying out the analysis to plan the network changes accordingly.

- **Delay:** Delay is to measure the time against a unit of transmitted data. Considering the example of data link layer, how much time it takes to reach an octet to the destination. While, at network layer how much time an IP packet takes to the targeted destination. Likewise, at the application layer it again depends upon the service too, such as for a web service how much time it takes for a request to reach the destination (hosting of the service) [7].

- **Quality & Packet Loss:** Measuring the quality and packet loss is of prime importance in order to enhance the performance of a network. Although the transmission mechanism automatically retransmits the lost packets but it can have some negative affects over the bandwidth due to consumption of resources. Packet loss can also considerably affect the performance of a wide range of internet protocols and services [63]. Besides, it can also be dependant over some application and services that are more critical towards packet loss as compared to other applications and services. For example, long range data transmission, audio video transmission and grid computing [64].

- **Thresholds:** Thresholds are defined on the basis of the availability of resources. There are certain limitations of different resources that are to be considered while defining the thresholds based over these baselines. So, if a threshold is to be reached, alarms can be generated and different actions are taken as a result to avoid a potential fault [11], [60]. Some examples can be: implementing threshold on CPU usage, memory usage and bandwidth usage.

- **Resource utilization:** Another monitoring function is to keep an eye on resources for efficient utilization; means obtain required performance and avoiding any unnecessary resource wastage (under utilization) [7]. For example if a link in your network with 10 MB bandwidth never reached to even 50 % then why to pay for 10 MB instead of paying 5 or 6 MB or using rest of the bandwidth for some other traffic.

By monitoring the above mentioned parameters, beside taking some quick action like in case of thresholds, we can generate reports over a period of time and by taking measurements of different attributes at some regular intervals, these intervals can be defined as required, from seconds to hours. Viewing the values presented in these report over a period of time can help in future planning e.g. potential bottle necks can be identified early or underutilized resources can be identified, and decisions for capacity planning can be done like bandwidth should be rolled in or out, at which link [7], [65].
3.2.5 Monitoring for security management:

Security Management (SM) deals with maintaining the total security of the network. It includes defending against all kinds of attacks to the network resources that can result in performance sufferance, service unavailability or configuration corruption, also to ensure the safety, privacy and integrity of the transmitted data [11]. The first thing is to control access to the network resources by mapping the critical resource and users and monitoring them against any suspicious and unauthorized access [66].

SM can be understood as a two part process. One is monitoring and the other is controlling. Monitoring tools are to detect unintentional and intentional security threats. They also detect unwanted access and traffic, and not allowing the users to run unauthorized application or try to change configuration [67].

3.3 Outcome; network monitoring tasks:

After discussing the Network Management Functions i.e. FCAPS we continued the study to identify and discuss all the monitoring functionalities for each of network management functions. As the result of section 3.2 below the monitoring functions (tasks a NMS should perform) are listed down in tabular form and will help in further study by acting as criteria to design the survey instrument. Now we know that the involvement and importance of monitoring in different areas of FCAPS varies so accordingly is the arrangement is done in the table e.g. Fault and Performance are at the top most priority in monitoring as above discussion showed. And their sub tasks are also dependent on each other.
### Table 2: Network monitoring tasks and interdependencies

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Dependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Fault Monitoring</strong></td>
<td></td>
</tr>
<tr>
<td>Alarms</td>
<td>Alarm, Topology map</td>
</tr>
<tr>
<td>Diagnosis</td>
<td>Alarm, Topology map</td>
</tr>
<tr>
<td>Trouble Ticket</td>
<td>Alarm, User complaint</td>
</tr>
<tr>
<td>Logging</td>
<td>Alarms, Throughput, Delay, Packet Loss, Thresholds reached, Resource utilization, Configuration changes</td>
</tr>
<tr>
<td>Proactive Detection</td>
<td>Throughput, Delay, Packet Loss, Thresholds reached, Resource utilization, Configuration changes</td>
</tr>
<tr>
<td><strong>Performance Monitoring</strong></td>
<td></td>
</tr>
<tr>
<td>Throughput</td>
<td>Throughput, Packet loss, Resources’ utilization,</td>
</tr>
<tr>
<td>Delays</td>
<td>Delay, Resources’ utilization</td>
</tr>
<tr>
<td>Packet loss</td>
<td>Resources’ utilization, Throughput</td>
</tr>
<tr>
<td>Thresholds</td>
<td>Resources’ utilization, Throughput</td>
</tr>
<tr>
<td>Resources Utilization</td>
<td>Throughput</td>
</tr>
<tr>
<td><strong>Security Monitoring</strong></td>
<td></td>
</tr>
<tr>
<td>Unauthorized access</td>
<td></td>
</tr>
<tr>
<td>Sudden traffic change</td>
<td></td>
</tr>
<tr>
<td>Illegal Applications</td>
<td></td>
</tr>
<tr>
<td><strong>Configuration Monitoring</strong></td>
<td>Topology mapping, Auto discovery</td>
</tr>
<tr>
<td><strong>Accounting Monitoring</strong></td>
<td>Resource utilization by users</td>
</tr>
</tbody>
</table>

#### 3.3.1 Analysis:

In table 2 it is presented that as a result of literature study the major monitoring tasks are about fault and performance, and most of them are interlinked and dependant on each other.

Security management is also dependent on monitoring, but mostly the monitoring tasks for security purpose are independent in nature and less linked with general fault and performance monitoring. Because of the independent nature and importance, security monitoring is done by specialized tools e.g. Intrusion Detection System (IDS).

In comparison to fault and performance, configuration and accounting management has fewer things related to monitoring. They are mostly related to controlling part of network management. And therefore usually have separate modules in network management station, whereas a monitoring system should provide functionalities like
topology mapping and auto discovery for configuration. And resource utilization details for accounting management.

3.4 Industrial interviews:

To verify and crosscheck the finding of literature review and also to minimize the probability of ignoring any important factor interviews with experienced professionals of the domain were conducted.

3.4.1 Objective:

Interviews help the researchers to understand the topic from the subjects’ point of view based on their experience, by providing insight into their world, their thoughts and feeling [49]. To answer RQ1 and before the comparison studies it is important to fully find out the tasks identified as Network Monitoring Tasks i.e. the functionalities which should be performed by an automated NMS. Industrial interviews were conducted with professionals doing network monitoring as the part of their regular job duties. Analyzing and relating results from literature review and interviews helped us to validate the results and also minimized the probability of ignoring any important factor.

3.4.2 Selection of interviewee (subjects):

As needed, to learn and validate functionalities of a NMS one best way was to ask the professionals of the field. We had contacted different well know companies using different contacts like our seniors from BTH, former colleagues and asked them if they can refer us to the concerning department and person of their organization. As a result we interviewed seven well experienced professionals of network management and administration domain. These experts are from different parts of the globe, working in technology industry, and network monitoring is the part of their day-to-day activities. All subjects have more than 4 years of experience of deploying, running and managing the data communication networks. The information including contact and experience of interviewees is presented in Appendix A.

3.4.3 Data Collection:

Structure is discussed in methodology chapter in section 2.4.2. We followed an interview protocol as described by Creswell [47] i.e. by including the following during our interviews; heading, opening statements, questions and probes. Heading includes title, interviewee information (e.g. name, job experience, designation etc), interviewer, date and place. Opening statements contain the interviewers’ introduction, giving a brief description of the study and its purpose to the subject (interviewee) etc. Questions, start with few basic questions to develop a comfort level then followed by the core questions. Probes, based on the reply of interviewee to the questions, sometimes required to clarify or get more information. In transition message, indication of changing the theme is delivered to the interviewee, e.g. in case of interviewee irrelevant answer or you needs to switch to another theme immediately.

After introduction phase followed by brief description about study and role of this interview in the study, a few questions were asked first of broader and a bit general
in nature, this helps interviewee to be get comfortable and in-sync with the topic, then followed by specific questions.

Instrument (questionnaire) created for the interviews is presented in Appendix B. Five interviews were conducted using Skype due to the geographical locations of the interviewees. Due to time availability constraints two interviewees were asked to fill the questionnaire, they were given enough time and full description in writing so the quality can be maintained. This helped to have opinion from different parts of the world. As we were interviewing for technological information about I.T. network which has nothing to do with internal or business processes of their organization so there was no hesitation from them and we don’t had to take observations etc. Specific Information (like name and versions) about the systems or tools they are using to manage the network in current organization was not asked (in some interviews discussed, but we didn’t mention about them in this report) due to the security reasons as they all are connected to public networks (internet). Average duration for an interview was of fifty minutes. All interviews were recorded with the consent of interviewees, and we also had manual notes during interviews this helped to cross question in response of interviewees answer if required later, plus during transcription of interviews.

3.4.4 Transcription of Interviews:

All interviews were recorded plus some manual notes were taken, later while listening recordings each interview noted into written form by keeping the information that is relevant to the study in a categorized manner, and discarded irrelevant discussion.

3.4.5 Analysis process:

After carefully writing the relevant information from the interviews by listening the recorded audios and consulting the notes, data was analyzed using Qualitative Data Analysis (QDA) approach. We used QDA as discussed by Seidel [68] this process has three phases noticing, collecting and thinking.

![Seidel model of QDA](image-url)
As shown in figure 2, the QDA process is not linear but has following characteristics [68]: Iterative and Progressive, means it is a cycle keeps repeating; recursive, means one part can call you back to previous part; holographic, means each step in the process contains the entire process.

We used these three steps to analysis the interviews. The purpose was to identify the monitoring functionalities that should be performed by a monitoring system, and their importance and interrelation (dependencies on each other). Noticing involve finding out interesting things from interviews and codes them, in our case these interesting things are network monitoring functionalities. Collecting, involves categorizing the noticed things, we manage different monitoring task under different heads e.g. tasks under the category of fault management or performance management. And in thinking you examine things which were noticed and collected towards the goals, examining and thinking was done to find out the relations, importance and dependencies of noticed monitoring functionalities.

3.4.6 Outcome:

Following subsections present the resulting output of the interviews after performing all described steps and analysis process. The subsections are based on the network management functionalities and their interdependence for network monitoring.

3.4.6.1 Fault and Performance:

As the result of analysis of the interviews it is observed that most important parts of network monitoring are fault and performance monitoring. The reason to discuss them here together is because we found that the boundaries between them are very blurred and everyone agreed that they have greater impact on each other and more related to each other than any other functionality e.g. a fault will result into poor performance or vice versa a performance issue can be developed as fault.

Our initial two questions during interviews were broader in nature and not about any particular type of functionality. When we asked about the importance of monitoring in overall network management, according to all seven replies to this question monitoring attributes fall into fault and performance domain, for example monitoring is important: to check service availability, for capacity management, for efficient resources utilization and for fault tolerance. For example, a response of an interviewee is “network monitoring provides a firsthand knowledge of services availability, can be use to formalize systematic approach to ensure performance, capacity management and trend analysis of resources’ utilization that help in acquisition and upgrade required resources in time”

Every one discussed these functionalities as a reply. The other broader question was to outline monitoring functionalities (tasks), the replies to this inquiry was also the list of the tasks which can be categorized under the head of fault and performance monitoring.

Two separate questions were asked in each interview to outline the functionalities for fault detection and for performance management. The answers were over-lapping. Even in the same interviews the functionalities for both were mostly replied as the same and in different interviews also were overlapping. For example responses to these two questions by an interviewee were, first when we asked about the functionalities for fault detection “alerts, reporting, utilization, delay, jitter, and packet loss” when we asked about performance monitoring functions “should
provide information like link utilization, delay, jitter, and packet loss”. In the end it was agreed that these can’t be separated and if one is talking about fault monitoring it automatically means performance as well and vice versa.

Following are the functionalities outlined by the interviewees related to fault and performance monitoring:

Alarms, Alerts, Fault recovering, Performance measurements, Reporting & logging, Thresholds and Resource utilization.

3.4.6.2 Security:

All the participants of interviews were convinced that when talking about monitoring, security monitoring is considered to be one of the most important components. Some discussed it during answering the first two questions as well along with fault and performance attribute. Correlation of security with fault and performance was also discussed by the interviewees. But when they were asked about how should security monitoring been done all strongly believed that security monitoring should be taken care of using special equipment and software (like firewalls, Intrusion detection systems, authentication systems) and it is out of the scope of a general monitoring tool. For example, an interviewee responds to security question “security in itself is a complete domain, it is preferred to have dedicated hardware and software to handle the security monitoring”

3.4.6.3 Configuration:

There is no doubt about the importance of a properly configured network, because poor configuration can make everything worse. But mostly configurations were performed using either built-in functionality using the interfaces provided by the vendors by the experts, or by using software tools specially made for this purpose. For example, a response was “pretty much depends on the hardware itself, some devices like Juniper support a lot of features like candidate configuration to perform managed configurations. I am not using any tools”. Usually, to make this secure and manageable and to avoid poor configuration this process is only handled by a limited number of experts (assigned to do this process) not by everyone working in the department.

Apart from its importance, this process is not part of the monitoring system. Only one interviewee was using a monitoring system (Ciscoworks) that contains this functionality. Basically the system is a commercial application not containing only monitoring functionalities but also can include management components.

3.4.6.4 Accounting:

The use of accounting management is limited these days, as the resources are available in large capacity, financially well in reach to the companies, as said by an interviewee “we don’t do this part of accounting in our organization” so usually they don’t put limitations to the employees in utilizing these resources like internet usage or disk space restrictions. This process is in maximum use where the services are provided to external clients and billing is an essential thing. During the interviews we found that most of the interviewees were not using accounting management. Two interviewees explained that their companies have deployed accounting management using active directory services of windows server and it has no relation to the monitoring system.
3.4.6.5 Summary of outcomes:

The sum up of interview process is discussed here. We came to the conclusion that fault monitoring and performance monitoring are the major and compulsory functionalities an NMS should perform. Furthermore these are so inter-related that it is very difficult to keep them separate from each other while performing monitoring tasks.

Security monitoring also has high importance and it is not possible to ignore it. But this is known as an independent domain to the experts, which should be handled separately from performance monitoring and fault monitoring by using its specific tool.

Configuration and accounting management has nothing much to do with regular monitoring functionalities these are also handled using built in functionalities from vendors or in some cases by using other software.

3.5 Concluding the chapter:

By analyzing the outcome of literature review and industrial interviews we have found that both are validating each other and the outcomes are almost the same. During literature study as presented in table 2 (in section 3.3) that the two most interdependent components of network management system are: fault and performance. Similarly the results of interviews show that when we talked about network monitoring the respondents’ reply was about fault, performance and security. But all agreed that security monitoring is an independent process and should be performed separately. Both results also show that configuration and accounting management are functionalities separate from monitoring and normally performed using separate tools.

3.5.1 Network monitoring functionalities:

Give below are the outcome as functionalities a NMS should perform:

- Alarm (event) management
- Alerts
- Device discovery (Topology map)
- Fault recovering mechanism (detection, trouble ticket, fault diagnose and location)
- Performance measurements (delays, throughput, packet loss, bandwidth utilization)
- Logging
- Capacity planning

This list served as baseline for the survey study that is presented in next chapter.
4 EVALUATION AND COMPARISON USING QUESTIONNAIRE SURVEY

This chapter presents the survey study we used for the evaluation purpose of both under study systems separately, and the comparison analyses of the results are performed to answer RQ2. The outcome of chapter 3 acted as the main criteria to defining survey questionnaire.

4.1 Survey as method:

The methodology is selected because to test both systems in lab experiment needs huge resources in form of equipment and time. And by using survey method we can investigate from professional users. And for this we need to get information from as many professionals as possible so questionnaire based survey is better option than interview. According to Wohlin et al. selection of empirical strategy depends upon the purpose of evaluation and conditions for the empirical investigation, a survey is often used for investigation when a tool or technique has been in use for a while [50]. As the systems, OpenNMS and Nagios are well established and have been in use for years by the network management professionals, for this reason, to evaluate both systems, survey study was performed. For each system separate survey was performed with the users of the concerning system. Both surveys were consisted of the same questions.

4.2 Purpose:

The main purpose was to evaluate and compare how well each system performs network monitoring functionalities (tasks), which are already discussed as the outcome of literature review and industrial interviews in the previous chapter. For this purpose using survey study, following steps were performed:

- Evaluation of Nagios.
- Evaluation of OpenNMS.
- Comparison analysis of the results from both evaluations.

4.3 Target participants:

The target respondents were the network professionals using either of these systems in their organizations. The experts using Nagios were asked to take survey for Nagios and similarly the experts using OpenNMS were surveyed for OpenNMS. Both survey questionnaires were similar, only the names of the monitoring systems were changed accordingly. The complete responses with proper demographic information were included in this study. Any response missing answers to some questions or with incomplete demographic information was not considered for the study.
4.4 Data collection, instrument design:

Web based questionnaire was used as the instrument to collect the required information from the experts. Close ended structured instrument was designed (except question 1 and 17) and used for this study, each item of the questionnaire contains a ‘stem’ (question or statement) and ‘response format’ (framework for respondents’ answer) [69]. We followed a four step approach to design our instrument [70]:

1. Establish the questions to be asked: This step is about to decide what kind of information needed for the study. There can be four types of questions which can be used in survey study: attributes, attitude, beliefs and behavior [70]. Attribute questions are used to inquire personal or demographic information. We asked for attributes information in our question number 1. Attitude questions are used to inquire from respondents that as a result of years of experience about topic, whether they have positive or negative feedback about the topic. Questions 2 to 11 were asked about attitude based on the functionalities of network monitoring. Belief questions are about what respondent think or what is his opinion about a particular subject at a particular time. Questions 12 to 17 were of type beliefs, as these are about respondent’s satisfaction, comfort and easy of learn and use about the system.

2. Selection of the question type/format for each question and specify wording: Researchers carefully rewrote the questions designed during the step 1 keeping in mind the target audience (i.e. Network professionals not researchers), further more these questions were verified by two network professionals. As the purpose of the study is to evaluate and compare two systems for this structured format was followed. The response-format used was rating scale five (except from question 1 and 17, where users need to fill their demographic information and comments subsequently). Question 2 to 11 used the Likert rating scale five (1 to 5) containing ‘very poor’ at 1 to ‘excellent’ at 5. And for questions 12 to 16 Likert scale was used with strongly disagree at 1 and strongly agree at 5. And then qualitative analysis was performed using graphs and tables. Although we have taken the averages of Likert scale reading for showing the combine results in graphs.

3. Design the question sequence and overall questionnaire layout: We followed the natural sequence of the questions according to the domain’s (network monitoring) requirement. Means the sequence of functionalities performed in actual environment of network monitoring was followed. We designed all questions on one web page with submit button, may be some arguments can be made in favor of dividing them on multiple pages but the reasons we choose this because 1) the questions (statements) were kept short as possible and the answering format was scale based to click one option (all options were presented in scale format in single line) 2) we motivated the target participant by including the information that questionnaire is of only one page to answer and will not take long. This can help to motivate potential participant to take the survey, he or she can know before starting that how long is it the survey by scrolling down a single page.

4. Develop ancillary documents, we developed a cover latter (explaining the study and requesting to take the questionnaire) and a thank you note.
Appendix C contains the questionnaire.

### 4.5 Pilot test the questionnaire:

To find out any ambiguity the pilot testing of the questionnaire was conducted with two experienced network professionals. These two professionals were using some other monitoring systems so they were not the potential participants of our original study. As the result based upon their feedback we had to drop one question, rest of the things were approved in the result of this pilot testing. We found this pilot testing enough because the questionnaire was aimed to the experts of these systems and can be best tested with them. Before this pilot test we also asked four students of BTH with the background of network management studies to explain us what they understand after reading each questions so we made sure it is actually the same what we wanted to convey.

### 4.6 Administer the questionnaire:

Questionnaire was web based and launched at [http://www.kwiksurveys.com/](http://www.kwiksurveys.com/). We emailed invitation to take this survey to the industrial contacts and communities containing the expert users of these systems, the experts replied from these communities are well known for their expertise regarding the system. Following the guidelines presented in [69], [70] we included the cover letter in the invitation email containing: purpose of the survey, why he or she was chosen, why the participation of the person is important, confidentiality was assured, estimated time to take the survey and last date. The survey was remained open to answer for 10 days. The survey response with proper demographic information were considered only, the importance was given to the position (title) and experience in this regard.

### 4.7 Results and analysis:

This discussion is divided in follow two subsections 4.7.1 presents the results of Nagios and 4.7.2 presents the results for OpenNMS.

#### 4.7.1 Nagios:

Total number of responses for Nagios questionnaire received was 32. To maintain the quality of study as suggested [50] before analysis we checked every response for any casual reply such as a reply that took very less time to complete or deviate to a great extent from the answers of other replies and looks like random selection. As the result, 5 out 32 responses were discarded due to incomplete answers and improper demographic information. 3 of them were discarded because we were not able to find the respondents in the related community with any strong history. We looked for the information of the respondents which are not directly contacted in related communities, and for these 3 there was not a significant record of their participation present. Other 2 were with incomplete responses. Total 27 responses were considered for the analysis.

#### 4.7.1.1 Respondents’ information:

All respondents were I.T. experts doing network management and monitoring tasks as their job responsibilities. These respondents were from different countries around
the world like mostly from USA, Pakistan and India. The reason of this split was, USA is the biggest market for networks as a result we were able to find much contacts in USA. Most important thing about respondents was the job experience. Because with this experience they can understand the strengths and weaknesses of the monitoring system they are using. All the responses taken under consideration were from the experts with more than 5 years of experience in networks and it goes to 15+ years.

4.7.1.2 Results:
The questions of the survey were broadly categorized into two categories:

1. Functional (based on results of RQ1), table 3 presents the result for questions 2 to 11
2. Usability, table 4 presents the results for questions 12 to 16.

Table 3: Results of functional section of the questionnaire for Nagios (*)

<table>
<thead>
<tr>
<th>Question #</th>
<th>Scale 1 to 5 (number and percentage of users reply at the scale)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 Very Poor</td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Overall Average</td>
</tr>
</tbody>
</table>

(*) Most left column represents question number. Numeric value and percentage value against each question in next five columns represents the number of respondent(s) out of 27, which selected that answer at the scale of 1 to 5 for the question. Most right column presents the average selection of the scale for the question.

For more understanding the average rating of each question along with overall average is shown as a graph in figure 3. The overall average rating of the functionalities performed by Nagios is 3.67. This shows that the monitoring functionalities performed by Nagios is in between average and good, a little incline towards good. More analysis is done in later section 4.8 while comparing Nagios and OpenNMS based of these results.
Figure 3: Functional section of the questionnaire for Nagios

Table 4: Results of usability section of the questionnaire for Nagios (*)

<table>
<thead>
<tr>
<th>Question #</th>
<th>1: Strongly disagree</th>
<th>2: Disagree</th>
<th>3: Neutral</th>
<th>4: Agree</th>
<th>5: Strongly Agree</th>
<th>Average of replied scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>3</td>
<td>8</td>
<td>16</td>
<td></td>
<td></td>
<td>3.48</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4.22</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4.22</td>
</tr>
<tr>
<td>15</td>
<td>7</td>
<td>11</td>
<td>9</td>
<td></td>
<td></td>
<td>3.07</td>
</tr>
<tr>
<td>16</td>
<td>3</td>
<td>14</td>
<td>10</td>
<td></td>
<td></td>
<td>4.25</td>
</tr>
</tbody>
</table>

(*) Most left column represents question number. Numeric value and percentage value against each question in next five columns represents the number of respondent(s) out of 27, which selected that answer at the scale of 1 to 5 for the question. Most right column presents the average selection of the scale for the question.
As whole this rating is showing the almost the agreement is on that system is good in usability. As shown in the table 4 average result is 3.85 that is near to 4, and 4 means agree, but not strongly agree. The results are also presented in graphical form in figure 4.

![Graph showing usability section results for Nagios](image)

**Figure 4: Results of usability section of the questionnaire for Nagios**

### 4.7.2 OpenNMS:

Total number of responses for OpenNMS questionnaire received was 35. 6 out 35 responses were discarded due to incomplete answers and improper demographic information. 3 of them was not completed and there was unanswered demographic information and 3 respondents were from community with no history of active participation. Total 29 responses were considered for the analysis.

#### 4.7.2.1 Respondents’ information:

Respondent groups for both systems were of the same profile, i.e. just like respondent for Nagios the participant were the professionals with extensive job experience of network management and monitoring. The difference is they were using OpenNMS for network monitoring. These respondents were from different countries around the world like mostly from USA, Pakistan and India, reason was discussed in section 4.7.1.1. Most important thing about them is the job experience. Because with this experience they can understand the strengths and weaknesses of the monitoring system they are using. The responses taken under consideration were from the experts with more than 5 years of experience in networks and it goes to 20 years.
4.7.2.2 Results:

The questions of the survey were broadly categorized into two categories:

1. Functional (based on results of RQ1), table 5 presents the result for questions 2 to 11
2. Usability, table 6 presents the results for questions 12 to 16.

Table 5: Results of functional section of the questionnaire for OpenNMS (*)

<table>
<thead>
<tr>
<th>Question #</th>
<th>1 Very poor</th>
<th>2 Poor</th>
<th>3 Average</th>
<th>4 Good</th>
<th>5 Excellent</th>
<th>Average of replied scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td>4</td>
<td>4</td>
<td>21</td>
<td></td>
<td>4.58</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>1</td>
<td>16</td>
<td>12</td>
<td></td>
<td>4.37</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>2</td>
<td>4</td>
<td>13</td>
<td>10</td>
<td>4.06</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>12</td>
<td>11</td>
<td>2</td>
<td></td>
<td>2.37</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>4</td>
<td>10</td>
<td>15</td>
<td></td>
<td>4.37</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>5</td>
<td>9</td>
<td>15</td>
<td></td>
<td>4.34</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>4</td>
<td>11</td>
<td>13</td>
<td></td>
<td>4.20</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>5</td>
<td>12</td>
<td>12</td>
<td></td>
<td>4.24</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>5</td>
<td>9</td>
<td>15</td>
<td></td>
<td>4.34</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td>8</td>
<td>21</td>
<td></td>
<td>4.72</td>
</tr>
</tbody>
</table>

(*') numbers in most left column represents questions, numeric value against each question’s number in every column represents the number of respondent(s) out of 29 which selected that answer for the question, also presented the percentage of respondents select that answer.

The average rating of each question along with overall average is shown as a graph in figure 5. The overall average rating of the functionalities performed by OpenNMS is 4.16. This shows that the monitoring functionalities performed by OpenNMS is a little above the good. Detailed analysis is done in later section 4.8 while comparing Nagios and OpenNMS based of these results.
Figure 5: functional section of the questionnaire for OpenNMS

Table 6: Results of usability section of the questionnaire for OpenNMS (*)

<table>
<thead>
<tr>
<th>Question #</th>
<th>1 Strongly Disagree</th>
<th>2 Poor</th>
<th>3 Neutral</th>
<th>4 Agree</th>
<th>5 Strongly Agree</th>
<th>Average of replied scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>12</td>
<td>4</td>
<td>3.24</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>4</td>
<td>10</td>
<td>15</td>
<td></td>
<td>4.37</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>12</td>
<td>9</td>
<td>3.86</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>6</td>
<td>16</td>
<td>6</td>
<td></td>
<td>3.93</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td>3</td>
<td>10</td>
<td>16</td>
<td></td>
<td>4.44</td>
</tr>
<tr>
<td><strong>Overall</strong></td>
<td><strong>Average</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>3.97</strong></td>
</tr>
</tbody>
</table>

(*) numbers in most left column represents questions, numeric value against each question’s number in every column represents the number of respondent(s) out of 29 which selected that answer for the question, also presented the percentage of respondents select that answer.

As whole this rating shows that the agreement is on system is good in usability. As shown in the table 6 average ration is almost 4, and 4 means agree, but not strongly agree. The results are also presented below in graphical form in figure 6.
4.8 Comparison analysis:

In this section survey results of Nagios and OpenNMS are compared against each other. Subsection 4.8.1 presents the comparison based on the functionalities and 4.8.2 presents the comparison that which system is easy to use.

4.8.1 Functionality based comparison:

Using the results of the surveys for Nagios and OpenNMS from section 4.7.1.2 and 4.7.2.2 the comparison is performed. Below table 7 presents the average results of the survey for both systems side by side.
Table 7: Comparison of Nagios and OpenNMS based Q2 – Q11(*)

<table>
<thead>
<tr>
<th>Systems</th>
<th>Question numbers along with their functionality attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2 AM</td>
</tr>
<tr>
<td>OpenNMS</td>
<td>4.58</td>
</tr>
<tr>
<td>Nagios</td>
<td>4.14</td>
</tr>
</tbody>
</table>


By observing these readings in table 7, differences in ratings for all functionalities between OpenNMS and Nagios can be seen. This table shows us that in every functionality, OpenNMS was rated better except for trouble ticketing where Nagios has a better rating and health monitoring where both has same rating level. When we calculated the average rating for both the difference in clear i.e. OpenNMS has 4.16 and Nagios has 3.67. Means OpenNMS rated above good and Nagios rated between average and good little bit more towards good.

Further we can create a subset of the functionalities presented in table 7 which can be considered more important than remaining. During the literature review it was noticed that Alarm management which includes alerts as a mechanism to inform instantly to the concerned staff members is most important part of fault monitoring, as some time alarm management itself known as an interchangeable term for fault management [7], [62]. Major part of network management functions are concerned with managing events and alarm and by effective management of these fault in the network should be fixed or prevented before end user raise fault ticket [60]. Alarms & alerts depend upon thresholds and health monitoring of the network devices. These are the monitoring functions which need to be performed real time. Remaining functionalities like fault diagnose, trouble ticket, resource utilization and logging are not real time activities. So the functions alarms, alerts, thresholds and health monitoring are considered to be more important. And many small scale network monitoring software in market are based on only these functionalities. Due to their importance the values for these functions are presented below in table 8 separately.

Table 8: Comparison of Nagios and OpenNMS based on AM, AL, TH and HM(*)

<table>
<thead>
<tr>
<th>Systems</th>
<th>Functionality attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AM</td>
</tr>
<tr>
<td>OpenNMS</td>
<td>4.58</td>
</tr>
<tr>
<td>Nagios</td>
<td>4.14</td>
</tr>
</tbody>
</table>


The comparison based on all monitoring functionalities is presented in graphical form below in figure 7. And an individual graph comparison is given in figure 8 for the separated functions which needed to be performed real-time.
Figure 7: Comparison based on functionalities

Figure 8: Comparison of Nagios and OpenNMS based on real-time monitoring functions.
This can be observed that in most of the functionalities and when combining all the monitoring functionalities OpenNMS is doing better than Nagios. Only in performing trouble ticket functionality Nagios is better, but overall trouble ticket management is seems to be done poorly in both systems when comparing to other functionalities. In case of health monitoring of the networked devices both are rated equal.

4.8.2 Which one is better in usability, a comparison:

For the purpose of this comparison the results for Q12 to Q16 of the surveys for Nagios and OpenNMS presented in section 4.7.1.2 and 4.7.2.2 are used. The average results of these questions are presented below in table 9 for both systems.

Table 9: Comparison of Nagios and OpenNMS based Q12 – Q16 (*)

<table>
<thead>
<tr>
<th>Systems</th>
<th>Question numbers along with their usage attributes</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>AVG</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>EL</td>
<td>EF</td>
<td>HM</td>
<td>IC</td>
<td>OS</td>
<td></td>
</tr>
<tr>
<td>OpenNMS</td>
<td></td>
<td>3.24</td>
<td>4.37</td>
<td>3.86</td>
<td>3.93</td>
<td>4.44</td>
<td>3.97</td>
</tr>
<tr>
<td>Nagios</td>
<td></td>
<td>3.48</td>
<td>4.22</td>
<td>4.22</td>
<td>3.07</td>
<td>4.25</td>
<td>3.85</td>
</tr>
</tbody>
</table>

(*) Where EL: easy to learn, EF: Efficient to use, HM: availability & quality of helping material, IC: Interface comfort, OS: overall satisfaction, AVG: average rating value of questions 12 to 16.

These results showing us almost all the results are round about rating scale 4, which is agree. And in all questions the rating difference is very low between Nagios and OpenNMS. Easy to learn and helping material category is a little bit better for Nagios. Efficiency and overall satisfaction has very little difference towards OpenNMS. The only some significant difference can be seen in interface comfort where OpenNMS scores more than Nagios.
As this can be observed that there is not much difference in the results related to usability section of survey and we are looking the answer for RQ3 i.e. which one is better in usability. We had noticed that most of the replies of the survey for both systems where we asked the questions about usability functions, the mostly answers were showing that they were agree to the statements. The reason for this probably is that all the participants were experienced professionals and using these systems for long. Due to which the commands and controls were at their finger tips and now they feel it easy to do required functionality with them. To strengthen our argument we can look at the results, when we asked them to remember and answer that how easy it was to learn the system when they start using it, this is the only result that has low values.

As the result of this discussion the need for performing further evaluation of usability appears, to fully resolve RQ3. And this usability evaluation is performed in next chapter.

Figure 9: Comparison of Nagios and OpenNMS based on Usability part of survey
5 Usability Comparison

In this chapter we conducted the usability comparison of Nagios and OpenNMS to find the answer to RQ3. To achieve this goal, understanding of usability is discussed in introduction chapter section 1.2. Methodology overview is presented in section 2.4.4 under methodology chapter. Below planning and conducting of evaluation method, results and comparison analysis of the results are performed.

5.1 Planning and conducting the tests:

5.1.1 Goal, objectives and measurements:

According to ISO 9241-11:1998, it is one of the necessary part to identify goals when specifying or measuring usability [38]. For a usability test you have to start by looking at what you want to learn that is by defining goals and concerns. It makes rest of the planning much easier [53].

Goal of this testing is to find the usability strengths and weaknesses of these systems and on that basis later we can compare them. To achieve this goal the objectives are measuring the usability of both systems (how effectively and efficiently user use the system and how satisfied is the user after completing the tasks) separately and compare the results.

Following are the measurements taken for both systems while tested by the users performing test tasks:

- How long it takes user to complete the task (and all tasks)?
- How many errors committed by user? (measured by testers, as the testers understand and know the right way to perform these tasks)
- What is the total time spent over the help material (documents and manuals)?
- How many times user showed a clear emotion (frustration or joy)? (this was noted while observing the users performing their tasks)
- How much satisfy the user with the system? (measured in debriefing session after the test by answering the quantitative questionnaire)

5.1.2 Test users’ selection:

While selecting test users the consideration was that the users should as representative as possible [41] of intended users of the system. First of all it was not possible for us to have a number of professionals for the usability testing of Nagios and OpenNMS secondly when tested with new users we were intended to measure attributes include learnability, memorability and satisfaction. To meet the above mentioned requirement we selected the ten volunteer students at Blekinge Institute of Technology studying in the programs ‘MSC in Computer Science’ and ‘MSC in Electrical Engineering with emphasis on Telecommunication’ and to be more relevant not any students from these program but all of them have studied the two related subjects 1) Networks Management 2) TCP/IP.

We found number of this sample is sufficient to fulfill our requirements, as the main objective to compare the results to identify which of two systems is relatively better in usability. As Lewis (2005) [56] concluded that it depends upon the goal a
practitioner wants to achieve, in cases it may be enough to test with five users and in another case 50 users may be participated, both can be true according to requirements. Here we are more depending upon the limited numbers of students but with network management and TCP/IP knowledge (means the users should be as close representative of intended audience as possible) not on quantity, as there are many available users studying in related MSC programs but not all have studied the subjects TCP/IP and especially Networks Management. Due to these criteria the test users know what functionalities they were asked to perform and have the knowledge about Network Monitoring. No one among them have any previous experience of using these kinds of systems.

When usability testing is conducted to compare the usability of two or more systems, two methods to employing the test users are 1) between-subject: when different test users for different systems 2) within-subject: when all test users get to test all systems [41]. We used within-subject, this helps to prevent against the natural individual variability of users (some individuals may have ability to do things faster than average and some may have tendency to be slower than average). The short come of this way is that user uses the learning from the first system during testing second one. But in our case, how to perform the tasks on either system were completely different and we had communicated this thing at the start of the test session (to make user aware that don’t waste time trying to do things in same way). Furthermore we divided users into two equal groups of five and users from one group performed the task using Nagios first and then OpenNMS, reverse the order for second group. While dividing the users another thing taken under consideration was to divide equal numbers in between both groups from the faculties of computer science and electrical engineering.

5.1.3 Tasks:

These systems are specialized for Network Management professionals not for the use of general users, therefore in some sense bit complex to use. Three tasks were carefully designed addressing the functionalities which are most important in a running monitoring system. This process was completed with the help of two network management experts.

Task 1: Discover the devices (computers, router) that are connected in the network and monitor their status.

Discovery of networked devices is one of the fundamental functions of network management [71]. It is the first step in network monitoring, if you need to monitor a network the first thing is that you should able to see a list of all networked devices and how they are mapped. Device discovery can be challenging and difficult in large and changing networks so it should be configured as auto-discovery using the monitoring system [8]. After configuring the device discovery test users had to report the status of the devices. This includes which devices (computers and router) are up and running and what services on the computers are running.

Task 2: Monitor the following parameters running on a remote Windows machine: Explorer status, Disk usage and CPU usage.

To monitor these parameters is very important because these are the part of real-time health monitoring of a remote computer. For example you are running any number of server(s) which are providing important services (e.g. active directory, www, mailing server, cloud computing and network storage) and can’t afford decrease in quality or
downtime, then it is important to keep under observation these parameters of the concerning server(s). When reaching a certain defined limit they can generate alarms. Task 3: Configure to send email Alert against any critical event (Alarm). Here test users were asked to configure automated alert, send to an email if a device’s status found to be down.

5.1.4 Evaluator:

One researcher (evaluator) of this study stayed during all the tests which were conducted separately for each user. There was almost no direct communication between evaluator and users during the tests, besides evaluator observing the users and taking notes to have the required measurements discussed in section 5.1.1.

5.1.5 Lab (Test instrument)

The requirement for testing these two systems is a networked environment that can resemble real world network. This should include communication between multiple networks as well. This required router(s), many computers and other connectivity equipment. To fulfill these requirements we used virtualization. Thanks to the open source market we found free virtual solutions to create a network(s) for our lab. VirtualBox [44] is an open source x86 and AMD64/Intel64 virtualization product. Using it you can create any number of computers (depends upon hardware power of your physical computer) with any operating system (known as guest operating system) on your physical computer. This helped us to create more machines for our network. We used Vyatta as router, Vyatta [45] is an open source networking solution that provides routing and is known as Cisco’s replacement. Vyatta can run as a virtual machine over the x86 platform.

The hardware used to design the lab is listed below in table 10.

<table>
<thead>
<tr>
<th>Component</th>
<th>Specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer A, Desktop</td>
<td>Intel Pentium 4 processor 1.5 GHZ, 1 GB RAM</td>
</tr>
<tr>
<td>Computer B, Desktop</td>
<td>Intel Pentium 4 processor 1.8 GHZ, 1.5 GM RAM</td>
</tr>
<tr>
<td>Computer C, Desktop</td>
<td>Intel Pentium 4 processor 1.8 GHZ, 1.5 GM RAM</td>
</tr>
<tr>
<td>Computer D, Laptop</td>
<td>Intel Core 2 Duo processor 2.26 GHZ, 3 GB RAM</td>
</tr>
<tr>
<td>Computer E, Laptop</td>
<td>Intel Dual Core processor 1.6 GHZ, 2GB RAM</td>
</tr>
<tr>
<td>LAN Switch</td>
<td>D-link 8 port</td>
</tr>
</tbody>
</table>

Software used is listed below in table 11
Table 11: Software used in lab

<table>
<thead>
<tr>
<th>Component</th>
<th>Specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating System</td>
<td>Microsoft Windows Xp</td>
</tr>
<tr>
<td>Operating System</td>
<td>Microsoft Windows Vista</td>
</tr>
<tr>
<td>Operating System</td>
<td>Ubuntu version 9.10 [72]</td>
</tr>
<tr>
<td>Operating System</td>
<td>Xbuntu version 9.10 [72]</td>
</tr>
<tr>
<td>Virtual Router</td>
<td>Vyatta Core, version 5.0.2 [45]</td>
</tr>
<tr>
<td>Virtual Machine (VM) (virtual computer)</td>
<td>Sun Virtual Box (Vbox), version 3.1.0 [44]</td>
</tr>
<tr>
<td>Monitoring System</td>
<td>Nagios, version 3.2.1 [16]</td>
</tr>
<tr>
<td>Monitoring System</td>
<td>OpenNMS, version 1.7.9 [17]</td>
</tr>
</tbody>
</table>

Using all hardware and software components listed in table 10 and table 11, results in network as shown in figure 10.

We kept the lab environment without any disturbance, during tests no one was allowed to interrupt or enter the lab.

![Physical Network](image)

**Figure 10: Physical Network**

The Network was expanded using virtualization with the help of Virtualbox to create Virtual Machines (computers) and Vyatta as a virtual router (replacement of Cisco router) to create WAN (Wide Area Network) environment. The detail description of creating this lab is given in Appendix D. Figure 11 shows the resulting Network.
5.1.6 Pre-Test:

Two pre-tests were conducted for both systems to go through from all the procedures. This made us ready and confident for real testing. For this purpose two volunteer students were selected studying in MSC Computer Science with experience and knowledge of computer networks, as we did not want to consume potential test users (who have studied TCP/IP and Network Management courses). These users were not the part of actual testing.

5.1.7 Conducting the Tests:

Tests were conducted at different times as suited to the test user one user at a time. After welcoming, the researcher briefed the user about the purpose of the test, how to perform the test, including information about the help material, role of the researcher during test i.e. he will be taking notes of the activities performed by user and will not interrupt in between and user have to do it his self with the help or documents provided and then explanation of the tasks. Furthermore they were assured that individual results with the user’s name will not be discussed, the purpose of the study is to test systems not the users. This initial session took about 10 to 15 minutes with each user depending upon the full understanding of the tasks.

As already mentioned every user had to test both systems, five users performed the task on Nagios first and then on OpenNMS and the other five users, vice-versa to this.
Printed lists of tasks with overview of steps to follow for completion of each task were provided to every user, separately for both systems. This task lists can be found in Appendix E. Tests were completed fully two times by every user for both systems. Means when a user completed tasks 1, 2 and 3 was asked to complete them again, after a refreshment break of 15 minutes (drinks and snacks were provided and access to wash room) for each system. These recompletions of the tasks were performed to check memorability attribute of the systems. Right after completion of the test, user was given the questionnaire to answer, questionnaire can be seen in appendix F.

5.1.8 Post test questionnaire (PTQ):

Every user was given the PTQ to answer shortly after completing the test for a system. Taking guidance from [73] questions were designed and following four scores can be obtained: overall satisfaction score, system usefulness, information quality and interface quality.

The questionnaire was designed to get answers on a Likert scale, 1 to 5, where 1 is strongly disagree and 5 is strongly agree.

5.2 Results and analysis:

5.2.1 Nagios:

All tasks were completed 100% by all users. The average results (measurements taken by evaluator during tests) of Nagios testing for all the users are presented below in tabular form. Every user completed the test two times. Table 12 shows the results of first attempt and table 13 shows the results of second attempt. These all tasks are fundamental for network monitoring are equally often to perform and also are interdependent on each other.

Table 12: Nagios average test results of first attempt

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Total time</th>
<th>Time spent reading documents</th>
<th>Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 1</td>
<td>44</td>
<td>21</td>
<td>5.4</td>
</tr>
<tr>
<td>Task 2</td>
<td>41</td>
<td>19</td>
<td>6.2</td>
</tr>
<tr>
<td>Task 3</td>
<td>16</td>
<td>11</td>
<td>0.6</td>
</tr>
<tr>
<td>Cumulative</td>
<td>101</td>
<td>51</td>
<td>12.2</td>
</tr>
</tbody>
</table>

Table 13: Nagios average test results of second attempt

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Total time</th>
<th>Time spent reading documents</th>
<th>Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 1</td>
<td>20</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>Task 2</td>
<td>17</td>
<td>5</td>
<td>1.6</td>
</tr>
<tr>
<td>Task 3</td>
<td>7</td>
<td>3</td>
<td>0.1</td>
</tr>
<tr>
<td>Cumulative</td>
<td>44</td>
<td>18</td>
<td>3.7</td>
</tr>
</tbody>
</table>
For Nagios, using these statistics, in following subsection 5.2.1.1 usability attribute effectiveness and in subsection 5.2.1.2 efficiency are evaluated.

5.2.1.1 Effectiveness:

According to ISO 9241-11:1998, it is necessary to decompose effectiveness and satisfaction so these components can be measureable [38]. Effectiveness includes measures of the completion rate and errors [39]. Completion depends on learnability and memorability of the software. When a new user using a system it depends on learnability of the system to complete the task and when an old user using a system it depends on memorability of the system to complete the task. According to Rubin and Chisnell Learnability, memorability and error rate are the components of effectiveness [40]. To measure effectiveness here we had measured sub components of effectiveness that are learnability, memorability and error rate.

Learnability, as according to Nielsen [41] system should be easy to learn so that user can start doing some work using that system. Learnability is users’ ability to use the system to an extent after a specific period of learning [40].

Usually learnability is measured as time consumed by a user to learn the system (interface) to complete the task. Folmer and Bosch wrote “Learnability, for example, can be measured by measuring the time it takes to learn a specific task” [74]. Similarly Preece et al. referred learnability as “time to lean a task” [54]. Nielsen also mentioned that “When analyzing learnability, one should keep in mind that users normally do not take the time to learn a complete interface fully before starting to use it. On the contrary, users often start using a system as soon as they have learned a part of the interface.” [41]. By considering these guidelines we can calculate learnability using the data gathered during usability testing. That is completion of tasks – time spent on learning to perform these tasks in percent of total time spent on completion of tasks.

Learnability = Percentage to which tasks completed – Learning time in percentage of total task time

In this study all tasks were completed 100 % so the equation for us is:

\[
\text{Learnability} = 100 - \left( \frac{\text{Learning time}}{\text{Total task time}} \right) \times 100 \%
\]

The values from table 12 are used to find out learnability of the tasks because these values represent the actual learning period when users were performing these tasks first time. Learnability of the tasks is as follows:

Task 1 Learnability = 100 - (21/44) x 100 (%) = 52.27 %
Task 2 Learnability = 100 – (19/41) x 100 (%) = 53.65%
Task 3 Learnability = 100 – (11/16) x 100 (%) = 31.25%
Cumulative Learnability = 100 – (51/101) x 100 (%) = 49.50%

(Cumulative means, after completing all tasks)

52
Memorability means system should be easy to remember. Nielsen explained that “Systems should be easy to remember, making it possible for casual users to return to the system after some period of not using the system, without having to learn everything all over again” [41]. It is one of the reasons to made users test the systems twice. This helped us to see how much they remember how to perform given tasks while performing second time. In our case the boundary between learnability and memorability is not very clear. Because the memorability is much concerned with the factor that when a user is using a system after some time. But for this study we don’t had the means and luxury of testing these systems with same users after a long time. So we asked them to perform the task again after a refreshment break. This served us to have some results regarding memorability (even it is short term memory).

So for this purpose value given in table 12 are used in relation to the values given in Table 14, on a scale of 100%, means if total memorability should be 100 percent then our tasks’ memorability will be difference between 100 and (time-spend-on-learning-second-time) divided by (time-spend-on-learning-first-time) (%).

Accordingly the equation is:

\[
\text{Memorability} = 100 - \left( \frac{\text{time spend on docs, second testing}}{\text{time spend on docs, first testing}} \right) \times 100 \%
\]

Task 1 Memorability = 100 - (10/21) x100 (%) = 52.38%
Task 2 Memorability = 100 – (5/19) x 100 (%) = 73.68%
Task 3 Memorability = 100 – (3/11) x 100 (%) = 72.72%
Cumulative Memorability = 100 – (18/51) x 100 (%) = 64.70%
(Cumulative means, after completing all tasks)

Error Rate, for a system it should be as less as possible errors to commit by users while performing tasks. The error rate is measured by counting the number of errors committed by user while performing some specific tasks [41]. Error rate is related to the duration of time a user using system, as long as he or she performing some tasks on the system the chances of committing errors are there. As Preece et al. describes that number of errors committed when performing a given task over time [54]. So when calculating error rate in percentage the equation will be:

\[
\text{Error Rate} = \left( \frac{\text{numbers of errors}}{\text{total time performing task}} \right) \times 100 \%
\]

Ideally zero errors should be committed that’s mean 100 % error free performance. Therefore calculation for error free percentage will:

\[
\text{Error Free} = 100 \% - \text{Error Rate} \%
\]

Task 1 Error Rate = (5.4/44) x 100 (%) = 12.27%
Task 1 Error Free = 100 – 12.27 = 87.73%
Task 2 Error Rate = (6.2/41) x 100 (%) = 15.12%
Task2 Error Free = 100 – 15.12 = 84.88%
Task 3 Error Rate = (.6/16) x 100 (%) = 3.75%
Task3 Error Free = 100 – 3.75 = 96.25 %
Cumulative Error Rate = (12/101) x 100 (%) = 11.88%
Cumulative Error Free = 100 – 11.88 = 88.12%
The results of these calculations are presented below in table 14.

Table 14: Nagios Results for learnability, memorability and error rate (in percentage)

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Learnability</th>
<th>Memorability</th>
<th>Error Free Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 1</td>
<td>52.27</td>
<td>52.38</td>
<td>87.73</td>
</tr>
<tr>
<td>Task 2</td>
<td>53.65</td>
<td>73.68</td>
<td>84.88</td>
</tr>
<tr>
<td>Task 3</td>
<td>31.25</td>
<td>72.72</td>
<td>96.25</td>
</tr>
<tr>
<td>Cumulative</td>
<td>49.50</td>
<td>64.70</td>
<td>88.12</td>
</tr>
</tbody>
</table>

As discussed earlier learnability, memorability and error rate are the components of effectiveness [40].

Effectiveness = (Learnability + Memorability + Error Free Ratio) / 3

Effectiveness Task 1 = (52.27 + 52.38 + 87.73) / 3 = 64
Effectiveness Task 2 = (53.65 + 73.68 + 84.88) / 3 = 70.73
Effectiveness Task 3 = (31.25 + 72.72 + 96.25) / 3 = 66.74
Effectiveness Cumulative = (49.50 + 64.70 + 88.12) / 3 = 67.44
(Cumulative means, after completing all tasks)

Table 15: Nagios results of effectiveness (in percentage)

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Task 1</th>
<th>Task 2</th>
<th>Task 3</th>
<th>Cumulative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Effectiveness</td>
<td>64</td>
<td>70.73</td>
<td>66.74</td>
<td>67.44</td>
</tr>
</tbody>
</table>

5.2.1.2 Efficiency:
Efficiency is usually measurement of time, how quickly the user’s goal can be achieved completely [40]. Efficiency of a system is the measurement to completing tasks, it should not include learning [41]. To calculate efficiency we used values from table 13 because at second attempt users had some experience to doing these tasks and spent less time in learning systems. Plus, the learning time from total time is subtracted, following the efficiency measurements of the tasks are:

Efficiency = Total time (spent on the task) – Learning time

Task 1 Efficiency = 20 – 10 = 10 minutes
Task 2 Efficiency = 17 – 5 = 12 minutes
Task 3 Efficiency = 7 – 3 = 4 minutes
Cumulative Efficiency = 44 – 18 = 26 minutes
We don’t have any benchmark values of efficiency for these tasks to compare. So we asked three experts (these are from the participants of interviews) to perform these tasks using their network monitoring system. All three were using different network monitoring systems. Two of them completed these tasks in 16 minutes and the last one completed in 17 minutes. We considered the least time taken (most efficient) as a comparison value, that is 16. Now we have 16 minutes as most efficient time. Percent efficiency can be calculated against 16 minutes for both systems for comparison purpose.

Efficiency (percent) = \((\text{most efficient possible time}/\text{time taken by this system}) \times 100\) (%)

**Efficiency = \((16/26) \times 100 = 61\%\)**

### 5.2.1.3 Users’ Satisfaction:

To find out this attribute PTQ were given to users, motivated in section 2.4.4.2 and defined in section 5.1.8 above in this chapter. Instrument is presented in Appendix F. The average result for each user is given in table 17

Table 17: Nagios satisfaction; results of satisfaction from PTQ of Nagios, scale of 5.

<table>
<thead>
<tr>
<th>Users</th>
<th>U1</th>
<th>U2</th>
<th>U3</th>
<th>U4</th>
<th>U5</th>
<th>U6</th>
<th>U7</th>
<th>U8</th>
<th>U9</th>
<th>U10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Satisfaction</td>
<td>2.87</td>
<td>2.62</td>
<td>3.12</td>
<td>3.12</td>
<td>4.12</td>
<td>3.12</td>
<td>2.00</td>
<td>2.12</td>
<td>3.00</td>
<td>3.37</td>
</tr>
</tbody>
</table>

By taking average of the values in table 17, cumulative measurement of satisfaction while using Nagios is calculated.

**Satisfaction = 2.94 (scale is 5)**

Satisfaction in percentage = \((2.94/5) \times 100\)

**Satisfaction = 58 \%**

### 5.2.2 OpenNMS:

All tasks were completed 100% by all users. The average results (measurements taken by evaluator during tests) of OpenNMS testing for all the users are presented below in tabular form. Every user completed the test two times. Table 18 shows the results of first attempt and table 19 shows the results of second attempt.
Table 18: OpenNMS average test readings of first attempt

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Total time</th>
<th>Time on docs</th>
<th>Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 1</td>
<td>14</td>
<td>8</td>
<td>0.3</td>
</tr>
<tr>
<td>Task 2</td>
<td>65</td>
<td>40</td>
<td>7</td>
</tr>
<tr>
<td>Task 3</td>
<td>9</td>
<td>4</td>
<td>0.2</td>
</tr>
</tbody>
</table>

| Cumulative | 88         | 52           | 7.5    |

Table 19: OpenNMS average test readings of second attempt

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Total time</th>
<th>Time on docs</th>
<th>Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 1</td>
<td>5</td>
<td>2</td>
<td>0.1</td>
</tr>
<tr>
<td>Task 2</td>
<td>24</td>
<td>12</td>
<td>4</td>
</tr>
<tr>
<td>Task 3</td>
<td>4</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

| Cumulative | 33         | 15           | 4.1    |

These statistics are used in following subsections 5.2.2.1 and 5.2.2.2 to evaluate effectiveness and efficiency while performing the tasks using OpenNMS.

5.2.2.1 Effectiveness:

It is already discussed in section 5.2.1.1 to evaluate effectiveness we need to evaluate Learnability, Memorability and Error Rate first. And relations are also already discussed in same section. So here in case of OpenNMS we used the same equations to calculate these attributes.

**Learnability**, calculated using values from table 18 in following equation (reasoning can be found in section 5.2.1.1):

\[
\text{Learnability} = 100 - \left( \frac{\text{Learning time}}{\text{Total task time}} \right) \times 100 \ (\%)
\]

Task 1 Learnability = 100 - (8/14) x 100 (%) = 42.85%
Task 2 Learnability = 100 – (40/65) x 100 (%) = 38.46%
Task 3 Learnability = 100 – (4/9) x 100 (%) = 55.55%
Cumulative Learnability = 100 – (52/88) x 100 (%) 40.90%
(Cumulative means, after completing all tasks)

**Memorability**, calculated using values from table 18 and table 19 in following equation (reasoning can be seen in section 5.2.1.1):

\[
\text{Memorability} = 100 - \left( \frac{\text{Time spend on docs, second testing}}{\text{time spend on docs, first testing}} \right) \times 100 \ (\%)
\]

Task 1 Memorability = 100 - (2/8) x 100 (%) = 75%
Task 2 Memorability = 100 – (12/40) x 100 (%) = 70%
Task 3 Memorability = 100 – (1/4) x 100 (%) = 75%
Cumulative Memorability = 100 – (15/52) x 100 (%) = 71.15%

**Error Rate**, calculated using values from table 18 in following equation:

\[
\text{Error Rate} = \frac{\text{numbers of errors}}{\text{total time performing task}} \times 100 \text{ (%)}
\]

Ideally zero errors should be committed that’s mean 100 % error free performance. Therefore calculation for error free percentage will:

\[
\text{Error Free} = 100 \% - \text{Error Rate} \%
\]

Task 1 Error Rate = (0.3/14) x 100 (%) = 2.25%
Task 1 Error Free = 100 – 2.15 = 97.85 %
Task 2 Error Rate = (7/65) x 100 (%) = 10.76%
Task2 Error Free = 100 – 10.76 = 89.23 %
Task 3 Error Rate = (0.2/9) x 100 (%) = 2.22%
Task3 Error Free = 100 – 2.22 = 97.77 %
Cumulative Error Rate = (7.5/88) x 100 (%) = 8.52%
Cumulative Error Free = 100 – 8.52 = 91.47%

The results of these calculations are presented below in table 20.

Table 20: OpenNMS results for learnability, memorability and error rate (in percentage)

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Learnability</th>
<th>Memorability</th>
<th>Error Free Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 1</td>
<td>42.86</td>
<td>75</td>
<td>97.85</td>
</tr>
<tr>
<td>Task 2</td>
<td>38.46</td>
<td>70</td>
<td>89.23</td>
</tr>
<tr>
<td>Task 3</td>
<td>55.55</td>
<td>75</td>
<td>97.77</td>
</tr>
<tr>
<td>Cumulative</td>
<td>40.90</td>
<td>71.15</td>
<td>91.47</td>
</tr>
</tbody>
</table>

As discussed earlier learnability, memorability and error rate are the components of effectiveness [40].

\[
\text{Effectiveness} = \frac{\text{Learnability} + \text{Memorability} + \text{Error Free Ratio}}{3}
\]

Effectiveness Task 1 = (42.85+75 +97.85 ) / 3 = 71.90
Effectiveness Task 2 = (38.46+70 +89.23) / 3 = 65.89
Effectiveness Task 3 = (55.55+75 +97.77) / 3 = 76.10
Effectiveness Cumulative = (40.90+71.15 +91.47) / 3 = 67.84

Table 21: OpenNMS results of effectiveness (in percentage)

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Task 1</th>
<th>Task 2</th>
<th>Task 3</th>
<th>Cumulative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Effectiveness</td>
<td>71.90</td>
<td>65.89</td>
<td>76.10</td>
<td>67.84</td>
</tr>
</tbody>
</table>
5.2.2.2 Efficiency:
According to the discussion in section 5.2.1.2, the formula to calculate efficiency is:

\[ \text{Efficiency} = \text{Total time} - \text{Learning time} \]

By using the values from table 20 in above mentioned equation following are the efficiency measurement for the tasks done using OpenNMS:
- Task 1 Efficiency = 5 - 2 = 3 minutes
- Task 2 Efficiency = 24 – 12 = 12 minutes
- Task 3 Efficiency = 4 – 1 = 3 minutes
- Cumulative Efficiency = 33 – 15 = 18 minutes

Table 22: OpenNMS results of efficiency (in minutes)

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Task 1</th>
<th>Task 2</th>
<th>Task 3</th>
<th>Cumulative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Efficiency</td>
<td>3</td>
<td>12</td>
<td>3</td>
<td>18</td>
</tr>
</tbody>
</table>

As discussed in section 5.2.1.2, we had most efficient time to perform these tasks is 16 minutes.
Efficiency (percent) = (most efficient possible time / time taken by this system) x 100 (%)

\[ \text{Efficiency} = \frac{16}{18} \times 100 = 88\% \]

5.2.2.3 Users’ Satisfaction:
To find out this attribute PTQ were given to users, motivated in section 2.4.4.2 and defined in section 5.1.8 above in this chapter. Instrument is presented in Appendix F.
The average result for each user is given in table 23.

Table 23: OpenNMS; results of satisfaction from PTQ of OpenNMS, scale of 5.

<table>
<thead>
<tr>
<th>Users</th>
<th>U1</th>
<th>U2</th>
<th>U3</th>
<th>U4</th>
<th>U5</th>
<th>U6</th>
<th>U7</th>
<th>U8</th>
<th>U9</th>
<th>U10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Satisfaction</td>
<td>4</td>
<td>3.5</td>
<td>3.25</td>
<td>3.75</td>
<td>3.5</td>
<td>3.62</td>
<td>3</td>
<td>3</td>
<td>3.75</td>
<td>2.75</td>
</tr>
</tbody>
</table>

By taking average of the value in table 23, cumulative measurement of satisfaction after using OpenNMS is calculated.

Satisfaction (Cumulative) = 3.41 (scale is 5)
Satisfaction in percentage = (3.41/5) x 100
Satisfaction = 68 %
5.3 Comparison Analysis:

In this section usability test results of both systems are compared against each other which give the answer to the RQ3 of this study.

Three types of comparisons are performed:

1. According to Nielsens defined subdivision of usability: learnability, memorability, error rate, efficiency and satisfaction.
2. According to ISO’s defined components of usability.
3. As summative usability defined by Sauro and Kindlund.

Nielsen [41] defined learnability, memorability, efficiency, error rate and satisfaction as the components of usability. In following table 24 these components for both systems are presented side by side.

<table>
<thead>
<tr>
<th></th>
<th>Nagios</th>
<th>OpenNMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learnability</td>
<td>49.50</td>
<td>40.90</td>
</tr>
<tr>
<td>Memorability</td>
<td>64.70</td>
<td>71.15</td>
</tr>
<tr>
<td>Error Rate (Error free)</td>
<td>88.12</td>
<td>91.47</td>
</tr>
<tr>
<td>Efficiency</td>
<td>26 minutes (61%)*</td>
<td>18 minutes (88%)*</td>
</tr>
<tr>
<td>Satisfaction</td>
<td>58</td>
<td>68</td>
</tr>
</tbody>
</table>

* See section 5.2.1 and 5.2.2 for description.

Learnability of Nagios is found better with a difference of 8.6 percent than OpenNMS, whereas memorability, efficiency and satisfaction is clearly more in case of OpenNMS. OpenNMS performed these function with less errors than Nagios thought the difference is very low. This means it takes more time initially to learn OpenNMS than Nagios but after learning it is easy to remember how to perform the functions, and also can be performed more quickly. Users are also better satisfied with OpenNMS.

Below in figure 12 this comparison is presented in graphical form.
ISO’s [37] defined components of usability are Effectiveness, Efficiency and Satisfaction. Given below is the table 25 presenting the values of these components side by side for Nagios and OpenNMS.

Table 25: Comparison based on ISO’s usability definition

<table>
<thead>
<tr>
<th></th>
<th>Nagios</th>
<th>OpenNMS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Effectiveness</strong></td>
<td>67.44</td>
<td>67.84</td>
</tr>
<tr>
<td><strong>Efficiency</strong></td>
<td>26 minutes (61%)*</td>
<td>18 minutes (88%)*</td>
</tr>
<tr>
<td><strong>Satisfaction</strong></td>
<td>58</td>
<td>68</td>
</tr>
</tbody>
</table>

* See section 5.2.1 and 5.2.2 for description.

In this definition we calculated quality attribute (learnability, memorability and error rate) together. As we discussed earlier in this section that learnability of Nagios has better value and memorability is better in case of OpenNMS and error rate (less error) has not much difference but a bit positive towards OpenNMS. And when we calculate them together to measure Effectiveness the overall value have almost no difference, i.e. the difference of 0.40 in favor of OpenNMS. Remaining two components are already discussed above in same section.

Below in figure 13 this comparison is presented in graphical form.
Usability, calculated as whole (summative), according to Sauro and Kindlund usability can be calculated as a single summative metric, even though it can’t be a replacement for component metrics but can provide high-level summary information [39]. It can provide an idea how useful a product can be, also can be used for hypothesis analysis [39]. For this reason presenting usability summative is useful here in our scenario where we are testing the hypothesis that one system is more usable than other. They purpose the summative equation as follows:

\[
\text{Usability} = \text{Efficiency} + \text{Effectiveness} + \text{Satisfaction (avg. satisfaction)}
\]

Results for Nagios and OpenNMS based on this equation are presented in Table 26 and also display in above given figure 13.

Table 26: Summative usability comparison of Nagios and OpenNMS

<table>
<thead>
<tr>
<th></th>
<th>Nagios</th>
<th>OpenNMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Usability</td>
<td>62.14</td>
<td>74.61</td>
</tr>
</tbody>
</table>

When looking at summative usability results, OpenNMS has a clear upper hand over Nagios. Discussing details and component level results earlier in this sub section we were already predicting this type of reading.

5.4 Summing up the chapter:

Here usability comparisons with respect to Nielsen’s definition, ISO’s definition and summative results are presented briefly and this provides the answer to RQ3. Means if you are looking for the answer with respect to some attribute you can check with the attributes presented by Nielsen or ISO. Or if required this answer at collective level, summative usability results is the answer for that.

Nielsen’s usability attributes (see table 24 in section 5.3 for details):
When we take Nielsen’s description of usability following are the attribute usability of a product is based upon:

Learnability: Users spent more time on learning the system when they were performing the tasks first time using OpenNMS in comparison with Nagios the percentage of time spent on learning system was 8.6 % less.

Memorability: After users had performed the tasks once, they were asked to perform same tasks again after taking a few minutes refreshment break to see how much they were able to memorize the systems. In this case users were more comfortable with OpenNMS. They had spent less 6.45 percent of time while reading documents and getting help for OpenNMS then they had spent for Nagios.

Error Rate: It was noticed that while performing the tasks with OpenNMS the ratio of committing errors by the users is less 3.35 % in comparison while performing same tasks with Nagios.

Efficiency: For efficiency we used the results taken while performing the tasks second time. At that point users had learned the systems to an extent. The overall efficiency to complete the tasks was better by 27% for OpenNMS than Nagios.

Satisfaction: To measure users’ satisfaction post test questionnaires were conducted with the users after performing the test for each system. The average result at a scale of 5 for Nagios was 2.94 and average result for OpenNMS is 3.41. The difference shows that with OpenNMS users were 10 % more satisfied in comparison to Nagios.

**ISO’s usability attributes** (see table 25 in section 5.3 for details):

Effectiveness: When measuring overall effectiveness that is composed of learnability, memorability and error rate the results for both systems are almost the same and have difference of only 0.4% positive towards OpenNMS.

Efficiency: For efficiency we used the results taken while performing the tasks second time. At that point users had learned the systems to an extent. The overall efficiency to complete the tasks was better by 27% for OpenNMS than Nagios.

Satisfaction: To measure users’ satisfaction post test questionnaires were conducted with the users after performing the test for each system. The average result at a scale of 5 for Nagios was 2.94 and average result for OpenNMS is 3.41. The difference shows that with OpenNMS users were 10 % more satisfied in comparison to Nagios.

**Summative usability** (see table 26 in section 5.3 for details):

Summative usability means measuring usability as a single core. When we calculated the results of effectiveness, efficiency and satisfaction together the value for OpenNMS is better than Nagios and difference is 12.47%.
6 CONCLUSION

The purpose of this thesis study is to evaluate two Network Monitoring Systems: OpenNMS and Nagios, and performing the comparison analysis based on the evaluation results. And this help us to deliver some suggestion which can be used as guideline while selecting between any available network monitoring solutions or also can be helpful while developing or improving a network monitoring system.

A literature study followed by interviews with the experienced network management experts is performed to find out the network monitoring functionalities that served as the criteria at which both systems are evaluated. For evaluation of the systems, using outcome of the literature review and interview studies as a base, a survey study is conducted with the network management experts which are using either of these systems. In later part of the thesis usability evaluation is done by performing usability testing. The results of these evaluations is compared and analyzed to answer the research questions.

This final chapter contains a brief discussion of findings (include answers to research questions), suggestions and future work.

6.1 Findings:

RQ1: What are the functionalities a Network Monitoring System should perform?

Section 3.3 and 3.5 present the functionalities which should be performed by a network monitoring system. The first step in our work was to find out the functionalities a network monitoring system should perform. It is done by performing literature review and interviews with domain professionals (the whole process in presented in chapter 3). By analyzing the outcome of literature review and industrial interviews we can see that both are validating each other and the outcomes are almost the same. During literature study (sections 3.2 and 3.3) we observed that the major network monitoring functionalities are related to fault management and performance management and these both areas are highly interdependent. Similarly the results of interviews (section 3.5) show that when we talked about network monitoring the respondents’ reply was about fault, performance and security, but all agreed that security monitoring is an independent process and should be performed separately. For fault management and performance management following monitoring tasks is identified: Alarm (event) management, Alerts, Device discovery, Fault recovering mechanism (detection, trouble ticket, fault diagnose and location), Performance measurements (delays, throughput, packet loss, bandwidth utilization), Logging and Capacity planning.

RQ2: How effectively Nagios and OpenNMS perform the network monitoring functionalities?

Based on the identified monitoring tasks from literature and interviews a survey study is conducted to evaluate and compare Nagios and OpenNMS (see chapter 4). The evaluation results to answer RQ2 are presented in section 4.7. These results indicate that both monitoring systems are performing most of these tasks, but lacking
in few. Trouble ticketing is not handled properly. As we mentioned before that monitoring is about fault and performance management so it is important that a monitoring system should provide all required possible mechanism to address these two areas and trouble ticketing is a part of fault management. For trouble ticket management these both systems are dependent on external modules.

RQ2.1: What is the comparison of the network monitoring functionalities of these systems?

The comparison analysis based on the survey study is presented in section 4.8.

RQ3: Which of the above systems is better in terms of usability?

To answer RQ3 usability comparison analysis is presented in sections 5.3 and 5.4. The usability evaluation of both systems is performed empirically by conducting usability testing and post test questionnaire (chapter 5 for details). In usability testing and also when asked in post test questionnaire one main reason that make most of the difference between the test results of the two monitoring systems is the way user can interact with the system. In case of OpenNMS users can use mouse (point and click) with less need of entering commands from command line interface in comparison to Nagios. Due to this it is easy to memorize and required tasks can be performed with less error rate and efficiently. Users were also more satisfied while using mouse to interact the system.

RQ4: Which functions and features should be taken under consideration when select a network monitoring system?

The answer to this question is presented below in section 6.2, which is based upon all parts of this study.

By presenting the detailed evaluation results based on monitoring functionalities and usability, it is now possible to find positives and negatives of both systems and one can select according to his or her requirements. Plus this study can work as a framework to evaluate different network monitoring products.

6.2 Suggestions:

Based on this thesis work we suggest following functions/ features that should be taken under consideration when selecting a network monitoring system. This list regarding functionalities and usability features also might be helpful in improving the systems.

- Alarms and Alerts should be flexible to implement and should not be complex to configure.

Alarm management and Alerts: The purpose of automated network monitoring is to get informed timely regarding any event that is important to run network and its services. A good configured network monitoring system works as a watch dog, it should generate alarms when ever any event occurs that is configured to be critical. Wrongly configured system generate false alarms (this can cause you trouble even in mid night). Although almost all network monitoring systems perform alarm management, one thing might be of concern some systems provide more liberty to
create events like incase of Nagios external censors can be used for different events e.g. alarm can be generated if the room temperature reaches to a specified limit. Based on these alarms the Alerts should be configurable in flexible manner such as send to different devices like email, mobile phone and may be different recipients in different timings (according to duty schedule).

- Automated device discovery should be performed by the monitoring systems.

**Device discovery:** It includes the information like operating system, running services and IP addresses of the devices. In case of OpenNMS it discovered all the devices by its self, whereas Nagios needs some range of IP addresses to be entered, it may cause problem if you have a big network and you can miss some IP addresses. Also should be able to create topology maps and relationship between devices, it helps analyzing the network and root cause of a problem.

- A range of options for real time health monitoring of network devices should be available.

**Health monitoring:** In real time health monitoring of network it is important to check which statistics a monitoring system is able to provide for the polled devices such as Status, CPU (utilization, heating, etc), memory, storage, bandwidth, response time, uptime/ downtime, services and software. This network’s health information enables network managers to set the thresholds and reaching a threshold results to generate the Alarm. As a result an Alert message with required information from the Alarm, will be sent to the configured device for example an alert is received to the network manager at home with the information that a server’s storage is at critical point, the manager can login to the server from his laptop and can resolve this problem by running disk clean up or deleting temp files. These statistics also provide help in performance optimization and resource utilization. Beside emergency alerts it is important to provide easy to understand reports including graphical representation of performance related statistics.

- Trouble ticket module should be a part of a monitoring system.

**Trouble ticket:** It is an important functionality that helps in fault resolution. It was noticed that the understudy systems were not providing this functionality and need third party trouble ticket modules. It is discussed that monitoring is mostly about fault and performance management so a monitoring system should help in addressing these areas as much as possible. Trouble ticket module should be a part of a monitoring system.

- Network monitoring system should be compatible and easy to integrate with other systems.

**Integration with other systems:** Literature, interview and survey studies show that a network monitoring system is a part of network management, and should be capable of integration with other software systems and modules such as configuration management, accounting and security systems. Some time the monitoring solutions provided by big names lack this capability on purpose because they want you to buy their modules for other functionalities too.

- Most of the operations should be completed with the help of mouse instead of entering long commands.

**User’s interaction with system:** The way users can interact with a monitoring system to perform the required tasks should be simple. In the usability tests we
observed that most of the differences in effectiveness, efficiency and users’ satisfaction were due to how a user can interact with the systems. The system provides the option of performing its configuration using mouse results better in the usability testing than the system that needs more commands to be entered using keyboard.

- Meaningful error messages should be generated by the system when user commits an error.

**Error messages:** During testing it was noticed that Nagios and OpenNMS both lacking in giving error messages. Most of the time when a user entered an incorrect command or configuration, the systems just save that setting and due to error in the command didn’t performed the required functionality, no error message was generated. And in few cases when they give an error message it was not that precise to help a user to locate immediately where the error was committed. Proper error massages and feedback is very important for any software system and need to be improved in these cases.

- Visualization or display of available information by a system is an important feature to consider.

**Visualization on monitoring station:** It is very important to present the information in a way that is simple and easy to understand. Sometimes information is available but being not been presented properly it is hard to find. Although the network monitoring systems provide important real time statistics in tabular form and also graphically but still need more improvement. As during usability testing some users after completing their task had taken time to find the results. This should be an area where continues improvement is needed to simplify and distribute the reports in multiple categories, as networks can be consist of thousands of nodes and there is many thing to watch on a single screen of monitoring station.

- It is important to check how much support (include peer and community support, paid support agreement), documents (guides, training material) and books are available.

**Support and helping material:** Especially when it comes to open source systems important factors to consider is community support, quality & availability of helping material (training docs, manuals). The system from commercial companies usually comes with support contract and good helping documents (which can include training videos). It was observed that the quality of documents was batter at Nagios community site, which was the reason users spent more time on reading documents while performing the tasks first time with OpenNMS.

- Open source vs. free, availability of source code and rights to change the code and develop and integrate new modules.

**Open source:** Selecting an open source system, reason may be not only money, open source provide you with source code of the product which gives you the flexibility to write your own code according to your company’s unique requirements. Source code should follow some general framework so can be easy to understand by others. Availability of source code is the difference between free software and open source software.

These function and features along with the suggestions are presented below in table 27.
Table 27: Mapping of suggestions to the part of study, on which they are based on.

<table>
<thead>
<tr>
<th>Functions/ features</th>
<th>Based On</th>
<th>How to do</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alarm management and Alerts</td>
<td>As an output of Literature review and Interview study.</td>
<td>Alarms and Alerts should be flexible to implement and should not be complex to configure.</td>
</tr>
<tr>
<td>Device discovery</td>
<td>Based on literature and interview studies and also verified during usability testing.</td>
<td>Automated device discovery should be performed by the monitoring systems.</td>
</tr>
<tr>
<td>Health monitoring</td>
<td>Literature and Interview studies</td>
<td>A range of options for real time health monitoring of network devices should be available.</td>
</tr>
<tr>
<td>Trouble ticket</td>
<td>Finding from survey study</td>
<td>Trouble ticket module should be a part of a monitoring system.</td>
</tr>
<tr>
<td>Integration with other systems</td>
<td>Literature and Interview studies</td>
<td>Network monitoring system should be compatible and easy to integrate with other systems.</td>
</tr>
<tr>
<td>User’s interaction with system</td>
<td>Finding of Usability testing</td>
<td>Most of the operations should be completed with the help of mouse instead of entering long commands.</td>
</tr>
<tr>
<td>Error messages</td>
<td>Based on usability testing</td>
<td>Meaningful error messages should be generated by the system when user commits an error.</td>
</tr>
<tr>
<td>Visualization on monitoring station</td>
<td>Literature and Usability testing studies</td>
<td>Visualization or display of available information by a system is an important feature to consider</td>
</tr>
<tr>
<td>Support and helping material</td>
<td>Literature and Usability testing</td>
<td>It is important to check how much support (include peer and community support, paid support agreement), documents (guides, training material) and books are available.</td>
</tr>
<tr>
<td>Open source</td>
<td>Literature study</td>
<td>Open source vs. free, availability of source code and rights to change the code and develop and integrate new modules.</td>
</tr>
</tbody>
</table>

6.3 Future work:

After conducting this study we feel that following work should be done in the area of network monitoring and management systems.

We have performed functional evaluation of the network monitoring systems by conducting survey study. It should be interesting if these evaluations and comparison will be performed by conducting experiments using real network. And these
evaluations can be performed for other monitoring systems including commercially available systems from big companies such as HP OpenView and Tivoli from IBM.

The usability testing was performed using Master’s students of Computer Science at Blekinge Technical Institute. This usability testing can be conducted in future by including experienced users. The usability testing using experts can give us some interesting results especially for efficiency although learnability, memorability and error rate will not be of much interest.
7 LIST OF REFERENCES

[38] “Measuring usability - balancing agility and formality : for stakeholders’ needs in software development (Licentiate by Jeff Winter) - Electronic Research


[58] M. Nair and V. Gopalakrishna, “‘CloudCop’: Putting network-admin on cloud nine towards Cloud Computing for Network Monitoring,” in *Internet*


## APPENDICES

### Appendix A: Interviewees’ Information

<table>
<thead>
<tr>
<th>ID</th>
<th>Experience (years)</th>
<th>Position</th>
<th>Firm Description</th>
<th>Technical Qualification</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>10</td>
<td>Network Manager</td>
<td>IBM, Canada</td>
<td>CCIE Routing and Switching and CCIE Service Provider</td>
</tr>
<tr>
<td>B</td>
<td>15</td>
<td>IT technician</td>
<td>Redcloud IT, Sweden</td>
<td>Several Years to maintain and support Novell Netware Servers And Windows Servers</td>
</tr>
<tr>
<td>C</td>
<td>10</td>
<td>Asst. Manager</td>
<td>NetSol Technologies, Pakistan</td>
<td>MCSE, CCNA, CCNP</td>
</tr>
<tr>
<td>D</td>
<td>5</td>
<td>Data Network Engineer</td>
<td>RIM, Canada.</td>
<td>CCIE</td>
</tr>
<tr>
<td>E</td>
<td>4</td>
<td>System Administrator</td>
<td>Mobicom Co Ltd. Mongolian</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>6</td>
<td>Network Engineer</td>
<td>Higher Education Commission, Pakistan</td>
<td>CCNA Certified</td>
</tr>
<tr>
<td>G</td>
<td>5</td>
<td>Linux Administrator</td>
<td>Horizon Medical World US private limited, USA</td>
<td>RHCE, CCNA</td>
</tr>
</tbody>
</table>
Appendix B: Interview Questions

Name:
Designation and Company:
Previous companies and designations (optional):
Total experience (in years):
Education:
Technical qualifications:
Email address:

Questions

1. How would you explain the importance of monitoring in overall Network Management?

2. Please outline monitoring functionalities which should be performed to keep Network running and optimized.

3. What functionalities should a monitoring system perform to detect and recover from a fault?

4. What functionalities a monitoring system should provide for performance optimization?

5. What about security monitoring, if it needs separate specialized tools, or general monitoring tools (like OpenNMS, Nagios, CiscoWorks, HP OpenView) should handle it as well? Please explain. Is your organization using specialized security systems, like IDS?

6. Even if security monitoring should have its specialized tools, what minimum we should have from general monitoring tools regarding security?

7. How configurations of networked devices should be performed managed and monitored? Any specialized tools you are using in your organization for this purpose?

8. Do you need accounting management in your organization (like resource allocation to the users (e.g. disk space), time restriction for file transfer over internet, etc), if yes how it is performed?
9. Optional, can you tell us the name of monitoring system/ tool you people are using in your organization?

10. The reason to select the monitoring system/ tool you are using now in your organization? Explain the factors taken under consideration at that time.

11. Which one are the most inter-related domains with respect to monitoring functionalities among: Fault, Performance, Security, Configuration and Accounting?

12. Name most important area(s) which should be addressed (almost fully) by a basic monitoring system, from: Fault, Performance, Security, Configuration and Accounting point.
Appendix C: Survey Instrument

For Nagios:

Nagios Evaluation

We are studying in MSC Computer Science at Blekinge Institute of Technology, Sweden. We are performing a comparison study between two open source network monitoring systems i.e. Nagios and OpenNMS. As a part of this study we need to have feedback from the professionals about the system (Nagios or OpenNMS) they are using.

It would be a great help for us if you fill out this short survey (about five minutes) based upon your insight and experience about network monitoring and Nagios.

- For questions 2 to 11 you may choose quality from very poor to excellent.
- For questions 12 to 16 please choose your level of agreement to the statement from Strongly Disagree to Strongly Agree.

The demographic information will be kept confidential and only be used for statistical purpose.

1. Please fill out following demographic information:
   * Position: 

   Company: 

   * Job experience (in years): 

   Country: 

   Email Address (if interested in results of this survey): 

2. Alarm (event) management & control in Nagios is (if it provides a range of options e.g. services’ status, nodes’ status, etc):

<table>
<thead>
<tr>
<th>Very Poor</th>
<th>Poor</th>
<th>Average</th>
<th>Good</th>
<th>Excellent</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
3. Capability of setting and managing Alerts in Nagios:

<table>
<thead>
<tr>
<th>Very Poor</th>
<th>Poor</th>
<th>Average</th>
<th>Good</th>
<th>Excellent</th>
</tr>
</thead>
</table>

4. Assistance provided by Nagios to diagnose (locate) actual fault(s):

<table>
<thead>
<tr>
<th>Very Poor</th>
<th>Poor</th>
<th>Average</th>
<th>Good</th>
<th>Excellent</th>
</tr>
</thead>
</table>

5. Trouble Ticket management in Nagios:

<table>
<thead>
<tr>
<th>Very Poor</th>
<th>Poor</th>
<th>Average</th>
<th>Good</th>
<th>Excellent</th>
</tr>
</thead>
</table>

6. Monitoring and reporting health of the devices in the network (e.g. CPU Heating, Server room temp, etc):

<table>
<thead>
<tr>
<th>Very Poor</th>
<th>Poor</th>
<th>Average</th>
<th>Good</th>
<th>Excellent</th>
</tr>
</thead>
</table>

7. Thresholds’ management using Nagios:

<table>
<thead>
<tr>
<th>Very Poor</th>
<th>Poor</th>
<th>Average</th>
<th>Good</th>
<th>Excellent</th>
</tr>
</thead>
</table>

8. Capability to measure and report different performance related attributes like throughput, delays, and packet loss:

<table>
<thead>
<tr>
<th>Very Poor</th>
<th>Poor</th>
<th>Average</th>
<th>Good</th>
<th>Excellent</th>
</tr>
</thead>
</table>

9. Capacity planning assistance by providing statistics of resource utilization:

<table>
<thead>
<tr>
<th>Very Poor</th>
<th>Poor</th>
<th>Average</th>
<th>Good</th>
<th>Excellent</th>
</tr>
</thead>
</table>

10. Affective Logging of all required activities:

<table>
<thead>
<tr>
<th>Very Poor</th>
<th>Poor</th>
<th>Average</th>
<th>Good</th>
<th>Excellent</th>
</tr>
</thead>
</table>
11. In your opinion, overall fault and performance management capability of Nagios is:

<table>
<thead>
<tr>
<th>Very Poor</th>
<th>Poor</th>
<th>Average</th>
<th>Good</th>
<th>Excellent</th>
</tr>
</thead>
</table>

12. It was easy to learn Nagios when I started using it:

<table>
<thead>
<tr>
<th>Strongly Disagree</th>
<th>Disagree</th>
<th>Neutral</th>
<th>Agree</th>
<th>Strongly Agree</th>
</tr>
</thead>
</table>

13. I perform monitoring tasks efficiently (quickly) using Nagios:

<table>
<thead>
<tr>
<th>Strongly Disagree</th>
<th>Disagree</th>
<th>Neutral</th>
<th>Agree</th>
<th>Strongly Agree</th>
</tr>
</thead>
</table>

14. Sufficient documentation (helping material) and community support is available for Nagios:

<table>
<thead>
<tr>
<th>Strongly Disagree</th>
<th>Disagree</th>
<th>Neutral</th>
<th>Agree</th>
<th>Strongly Agree</th>
</tr>
</thead>
</table>

15. The interface of Nagios (including how to interact e.g. using keyboard, mouse, and commands) is pleasant:

<table>
<thead>
<tr>
<th>Strongly Disagree</th>
<th>Disagree</th>
<th>Neutral</th>
<th>Agree</th>
<th>Strongly Agree</th>
</tr>
</thead>
</table>

16. Overall, I am happy and satisfied with Nagios, providing me what I need from this kind of system:

<table>
<thead>
<tr>
<th>Strongly Disagree</th>
<th>Disagree</th>
<th>Neutral</th>
<th>Agree</th>
<th>Strongly Agree</th>
</tr>
</thead>
</table>

17. Any comments, like what do you like most or what you don’t like about Nagios, or any thing you want to share.
OpenNMS Evaluation

We are studying in MSC Computer Science at Blekinge Institute of Technology, Sweden. We are performing a comparison study between two open source network monitoring systems i.e. Nagios and OpenNMS. As a part of this study we need to have feedback from the professionals about the system (Nagios or OpenNMS) they are using.

It would be a great help for us if you fill out this short survey (about five minutes) based upon your insight and experience about network monitoring and OpenNMS.

- For questions 2 to 11 you may choose quality attribute from very poor to excellent.
- For question 12 to 16 please choose your level of agreement to the statement from strongly disagree to strongly agree.

The demographic information will be kept confidential and only be used for statistical purpose.

1. Please fill out following demographic information(optional):
   - Position
   - Company
   - Experience (in years)
   - Country
   - Email Address (if you are interested in result of this survey)

2. Alarm (event) management & control in OpenNMS is (if it provides a range of options e.g. services' status, nodes' status, etc):
   - Very Poor
   - Poor
   - Average
   - Good
   - Excellent
3. **Capability of setting and managing Alerts in OpenNMS:**

<table>
<thead>
<tr>
<th>Very Poor</th>
<th>Poor</th>
<th>Average</th>
<th>Good</th>
<th>Excellent</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4. **Assistance provided by OpenNMS to diagnose (locate) actual fault:**

<table>
<thead>
<tr>
<th>Very Poor</th>
<th>Poor</th>
<th>Average</th>
<th>Good</th>
<th>Excellent</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5. **Trouble Ticket management in OpenNMS:**

<table>
<thead>
<tr>
<th>Very Poor</th>
<th>Poor</th>
<th>Average</th>
<th>Good</th>
<th>Excellent</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6. **Monitoring and reporting health of the devices in the network (e.g. CPU Heating, Server room temp, etc.):**

<table>
<thead>
<tr>
<th>Very Poor</th>
<th>Poor</th>
<th>Average</th>
<th>Good</th>
<th>Excellent</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

7. **Thresholds' management using OpenNMS:**

<table>
<thead>
<tr>
<th>Very Poor</th>
<th>Poor</th>
<th>Average</th>
<th>Good</th>
<th>Excellent</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

8. **Capability to measure and report different performance related attribute like throughput, delays and packet loss:**

<table>
<thead>
<tr>
<th>Very Poor</th>
<th>Poor</th>
<th>Average</th>
<th>Good</th>
<th>Excellent</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

9. **Capacity planning assistance by providing statistics of resource utilization:**

<table>
<thead>
<tr>
<th>Very Poor</th>
<th>Poor</th>
<th>Average</th>
<th>Good</th>
<th>Excellent</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

10. **Affective Logging of all required activities:**

<table>
<thead>
<tr>
<th>Very Poor</th>
<th>Poor</th>
<th>Average</th>
<th>Good</th>
<th>Excellent</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
11. In your opinion, overall fault and performance management capability of OpenNMS is:

<table>
<thead>
<tr>
<th>Very Poor</th>
<th>Poor</th>
<th>Average</th>
<th>Good</th>
<th>Excellent</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

12. It was easy to learn OpenNMS when I started using it:

<table>
<thead>
<tr>
<th>Strongly Disagree</th>
<th>Disagree</th>
<th>Neutral</th>
<th>Agree</th>
<th>Strongly Agree</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

13. I perform monitoring tasks efficiently (quickly) using OpenNMS:

<table>
<thead>
<tr>
<th>Strongly Disagree</th>
<th>Disagree</th>
<th>Neutral</th>
<th>Agree</th>
<th>Strongly Agree</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

14. Sufficient documentation (helping material) and community support is available for OpenNMS:

<table>
<thead>
<tr>
<th>Strongly Disagree</th>
<th>Disagree</th>
<th>Neutral</th>
<th>Agree</th>
<th>Strongly Agree</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

15. The interface of OpenNMS (including how to interact e.g. using keyboard, mouse, and commands) is pleasant:

<table>
<thead>
<tr>
<th>Strongly Disagree</th>
<th>Disagree</th>
<th>Neutral</th>
<th>Agree</th>
<th>Strongly Agree</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

16. Overall, I am happy and satisfied with OpenNMS, providing me what I need from this kind of system:

<table>
<thead>
<tr>
<th>Strongly Disagree</th>
<th>Disagree</th>
<th>Neutral</th>
<th>Agree</th>
<th>Strongly Agree</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

17. Any comments, like what do you like most or what you don't like about OpenNMS, or anything you want to share:

<p>| |</p>
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>
Appendix D: Lab design

Lab environment: Virtual Box (Vbox) is an open source virtualization tool supported by Sun Microsystems. Vbox was used to create virtual machines (virtual computers) over original computers e.g. if one is using a computer with any operating system installed, by using Vbox over that operating system give the power to create an environment (virtual environment) by sharing hard disk space, CPU, memory and other resources of the computer and using that environment just like another computer. The original machine is known as host machine with host OS and the virtual machine as guest machine with guest OS. Host and guest OS works just like two separate machines and can be different from each other e.g. Windows XP host with Ubuntu guest or Ubuntu host with Windows guest. The number of VM (guest OS) can be more than one, it depends upon the CPU power (Speed + number of cores), RAM and Disk space, you need to share these resources for every guest.

Vyatta is an open source system can work as a VM over x86 platforms to perform routing functionalities. We used Vyatta as a replacement of Cisco router to create the network lab.

Physically all computers were connected to a single LAN switch using their 100 Mbps Ethernet cards, UTP 5 cable and RJ45 connectors. The operating systems installed on the computers were: Computer A Ubuntu, B Windows XP, Computer C Xubuntu, Computer D and E Windows Vista.

Using these computers as host machines we deployed Vyatta on Computer C using Vbox as a guest system this gave us a Computer + a Router on same physical machine. And using Vbox a guest Ubuntu was deployed on three of the remaining four computers of computers as guest operating systems. As the result now we had 5 host machines + 3 guest computers + 1 router. One host which was without any guest OS used as the management station and Nagios and OpenNMS were installed on it.

Now we had logically 8 Computer for networking, 1 computer as management station and 1 router. We developed a WAN network using these systems. And assign systems in different network subsets. Direct communication between 2 systems was not possible even if both are on same physical machine and directly connected to same switch. Vyatta was configured as a router that performed the communication between these systems just like an original environment.

We installed configured OpenNMS and Nagios on management station and checked the functionalities to discover network devices and all important issues before conducting the tests with users.
Appendix E: Task descriptions for Nagios and OpenNMS

Nagios task:

Task 1: Discover connected devices and their status.

Step 1:- Create a host.cfg file in ‘/etc/nagios3/conf.d’ directory.
Step 2:- Insert commands in file.
Step 3:- Create a service.cfg file in ‘/etc/nagios3/conf.d’ directory.
Step 4:- Insert commands to define ping service.
Step 5:- Restart the Nagios server.

Task 2: Monitor the remote Window machine for Explorer, Disk Drive, and CPU load (using port number 12489 and password 123).

Step 1:- Open the file ‘/etc/nagios3/conf.d/service’.
Step 2:- Enter required code to monitor the Explorer.
Step 3:- Enter required code to monitor the Disk Drive.
Step 4:- Enter required code to monitor the CPU load
Step 5:- Open file ‘/etc/Nagios-plugins/config/nt.cfg’ verify port number and password in “check_nt”.
Step 6 :- Restart the Nagios server.

Task 3: Enter an email address to receive the alerts in case of any critical problem.

Step 1:- open the file ‘/etc/nagios3/conf.d/contacts_nagios2.cfg’.
Step 2:- Enter email address test@localhost.
Step 3:- Assign this email to group admin.
Step 3:- Restart the Nagios server.
OpenNMS task:

Task 1: Discover connected devices and their status.
Step 1: Click the admin tab
Step 2: Click the configure discovery tab (new screen is opened).
Step 3: Perform required changes.
Step 4: Save changes.
Step 5: Click home tab and check the status.

Task 2: Monitor the remote Window machine for Explorer, Disk Drive, and CPU load (using port number 12489 and password 123).
Step 1: Open file /etc/opennms/nsclient-config.xml
Step 2: Enter necessary code to access port 12489 with timeout 3000ms.
Step 3: Open file /etc/opennms/collected-configuration.xml
Step 4: Enter required code to filter and services definition for NSclient where IP address of Window machine is 192.168.1.40 and port address 12489.
Step 5: Open file /etc/opennms/nsclient-datacollection-config.xml
Step 6: Enter necessary code to monitor the Explorer.
Step 7: Enter necessary code to monitor the Disk Drive.
Step 8: Enter necessary code to monitor the CPU load.
Step 9: Force reload the OpenNMS file.
Step 10: Start OpenNMS.

Task 3: Enter an email address to receive the alerts in case of any critical problem.
Step 1: Click admin tab.
Step 2: Click user and groups tab (new screen is opened).
Step 3: Click on add new user.
Step 4: Set an email notification.
Step 5: Save changes.
Appendix F: Post test questionnaire

Post Test Questionnaire

The following questionnaire (on next page) will reflect your reaction and satisfaction, about the system you used to perform the given tasks. Your reply will help us to find the level of satisfaction after using the system.

Please think about the tasks carefully you have performed while answering.

Read each statement and answer by circling the number on the scale 1 to 5, to show the level of agreement towards the statement.

1: Strongly disagree
2: Disagree
3: Neither agree nor disagree
4: Agree
5: Strongly agree

Thank you very much!
Nagios:

1. I find Nagios easy to use towards achieving given tasks.
   Strongly Disagree 1 2 3 4 5 Strongly Agree

2. I was comfortable using Nagios.
   Strongly Disagree 1 2 3 4 5 Strongly Agree

3. It was easy to learn Nagios.
   Strongly Disagree 1 2 3 4 5 Strongly Agree

4. I think I can use Nagios to do Network Monitoring task productively in future.
   Strongly Disagree 1 2 3 4 5 Strongly Agree

5. Error messages were clear to understand and help to recover from errors.
   Strongly Disagree 1 2 3 4 5 Strongly Agree

6. The helping information (documentation, online helping material) was clear and it was easy to find required information to complete the tasks.
   Strongly Disagree 1 2 3 4 5 Strongly Agree

7. The way to interact the Nagios (include using mouse, keyboard and user interface) is pleasant and I like it
   Strongly Disagree 1 2 3 4 5 Strongly Agree

8. Overall, I am satisfy about achieving the goals (tasks) using Nagios.
   Strongly Disagree 1 2 3 4 5 Strongly Agree

Any comments, your overall feelings, anything you want to share after performing this test:
OpenNMS:

ID:                  Date:

1. I find OpenNMS very easy to use towards achieving given tasks.
   Strongly Disagree  1  2  3  4  5  Strongly Agree

2. I was comfortable using OpenNMS.
   Strongly Disagree  1  2  3  4  5  Strongly Agree

3. It was easy to learn OpenNMS.
   Strongly Disagree  1  2  3  4  5  Strongly Agree

4. I think I can use OpenNMS to do Network Monitoring task productively in future.
   Strongly Disagree  1  2  3  4  5  Strongly Agree

5. Error messages were clear to understand and help to recover from errors.
   Strongly Disagree  1  2  3  4  5  Strongly Agree

6. The helping information (documentation, online helping material) was clear and it was easy to find required information to complete the tasks.
   Strongly Disagree  1  2  3  4  5  Strongly Agree

7. The way to interact the OpenNMS (include using mouse, keyboard and user interface) is pleasant and I like it
   Strongly Disagree  1  2  3  4  5  Strongly Agree

8. Overall, I am satisfy about achieving the goals (tasks) using OpenNMS.
   Strongly Disagree  1  2  3  4  5  Strongly Agree

Any comments, your overall feelings, anything you want to share after performing this test: