“NETWORK MONITORING TOOL”

PROJECT REPORT

Submitted for CAL in B. Tech Networks And Communication (CSE1004)

By

SHIVAM KAPOOR (15BCE1339)
ABHISHEK SINGH (15BCE1009)

Slot: G1

Name of faculty: **Prof. Renuka Devi**

(SCHOOL OF COMPUTING SCIENCE AND ENGINEERING)

VIT®
UNIVERSITY
(Estd. u/s 3 of UGC Act 1956)
VELLORE ■ CHENNAI
www.vit.ac.in

MAY, 2017

CERTIFICATE

This is to certify that the Project work entitled ***"NETWORK MONITORING TOOL"*** that is being submitted by ***"SHIVAM KAPOOR (15BCE1339 and ABHISHEK SINGH (15BCE1009)"*** for CAL in B. Tech Networks And Communication (CSE1004) is a record of bonafide work done under my supervision. The contents of this project work have not been submitted for any other CAL course.

Place : Chennai

Date : 03/05/2016

SIGNATURE OF STUDENTS: SHIVAM KAPOOR (15BCE1339)

ABHISHEK SINGH (15BCE1009)

SIGNATURE OF FACULTY: PROF. RENUKA DEVI

ACKNOWLEDGEMENT

First of all, we would like to express our gratitude to the SCSE Dean & the University Management for giving us an opportunity to carry out this project and prove our worth.

We would also like to thank our families and friends for moral support and constant guidance.

And finally, this project would not have been possible without the constant guidance and support offered by our Professor Renuka Devi. Her suggestions and advices helped us a lot.

SHIVAM KAPOOR (15BCE1339)
ABHISHEK SINGH (15BCE1009)

ABSTRACT

Monitoring and diagnosis of network conditions is a central problem in networking. As such, it has received a lot of attention in the Internet community in general and in the context of overlay networks in particular.

A Network monitoring is the heart of a server. Network Monitoring tool is needed when system analyst needs the data in figures and graphs and to analyze the nodes connected. Through the network monitoring tool, one is able to analyze the whole server and its CPU Usage, Packets deliverance, Connections, Bandwidth, Memory and many more things.

Network monitoring allows network managers to get a better insight in the network traffic transiting in a managed network. In order to make the tasks of a network manager easier, many network monitoring tools are made available for a wide range of purposes (e.g., traffic accounting, performance analysis, and so on) network managers may have. However, most of these tools lack to provide graphical and accurate data output. Through this project, we want to make a state of the art, sophisticated Network Monitoring Tool made for terminals to increase efficiency and computability and making it available on every Linux with a terminal.

In the project, following tools are included –

- Bandwidth Analysis
- CPU Usage Analysis
- MEM Usage Analysis.
- Packet Loss Analysis
- Latency Analysis
- Throughput Analysis
- Jitter Analysis

Etc. Also, references are attached at the end of the document.

CONTENTS

I. DOCUMENTATION

II. APPENDICES

INTRODUCTION

Networks have evolved from being a flat network where there were only a handful of elements. Everything was connected—to a more complex design where there are a lot more technologies, such as cloud, wireless, remote users, VPN, IoT, mobile devices, and so on.

In spite of all the evolution that has occurred, one factor that has been constant is the need for network monitoring. Monitoring allows network admins to know what is going on in their network, be it with their WAN, LAN, VoIP, MPLS, and other connections or the state of various network elements or nodes such as the access, distribution and core switches, routers, firewalls, servers, client systems and so on.

A network monitoring system is capable of detecting and reporting failures of devices or connections. It normally measures the processor (CPU) utilization of hosts, the network bandwidth utilization of links, and other aspects of operation. It will often send messages (sometimes called *watchdog* messages) over the network to each host to verify it is responsive to requests. When failures, unacceptably slow response, or other unexpected behavior is detected, these systems send additional messages called *alerts* to designated locations (such as a management server, an email address, or a phone number) to notify system administrators.

In order to successfully monitor a network or even server and systems, the availability of the below options is necessary:

- Data or information from various elements in the network. Data includes information about the working, current status & performance, and health of the element being monitored.

- An application or monitoring software must be able to collect, process, and present data in a user-friendly format. Software should even alert users about impending problems based on thresholds.

Part - I

DOCUMENTATION

DOC 1
MODULE DESCRIPTION

1.1 Introduction

There are lot of modules included in this project to monitor a node's performance on different matrices. These different modules help us statistically analyze different performance variables and decide future plan or immediate plan of action. The next subtopic contains all the main modules involved.

1.2 Modules

- *Bandwidth Analysis*

Commonly measured in bits/second is the maximum rate that information can be transferred.

In computer networks, bandwidth is used as a synonym for data transfer rate, the amount of data that can be carried from one point to another in a given time period (usually a second). Network bandwidth is usually expressed in bits per second (bps); modern networks typically have speeds measured in the millions of bits per second (megabits per second, or Mbps) or billions of bits per second (gigabits per second, or Gbps).

We have included PHP, Bash Scripts and C program to simulate the same and show the output as graphs. Screenshots and Code is attached in Appendix.

- *CPU Usage*

  CPU time (or process time) is the amount of time for which a central processing unit (CPU) was used for processing instructions of a computer program or operating system, as opposed to, for example, waiting for input/output (I/O) operations or entering low-power (idle) mode. The CPU time is measured in clock ticks or seconds. Often, it is useful to measure CPU time as a percentage of the CPU's capacity, which is called the CPU usage.

  We have implemented C code to simulate the graphs for CPU usage. Screenshots and Code is attached in Appendix.

- *Mem Usage*

  Memory Usage or Mem Usage is the amount of RAM used for processing instructions of a computer program or operating system, as opposed to, A node eats lot of RAM In processing and computing data and this data can be used to analyzed a node's condition and workload. Thus, this matrix is very important.

  We have implemented C code to simulate the graphs for CPU usage. Screenshots and Code is attached in Appendix.

- *Packet Loss Analysis*

  When data is transmitting over computer network, one or more packets may fail to reach their destinations, and this is packet loss.

  Packet loss can be caused by multiple factors including network congestion, the performance or policy of networking devices, and networking hardware faults.

  There can be many causes of packet loss, which can relate to how we get access to the data, the kind of technology used to capture packets, the processing platform, and the application software used to analyze the data. Let's take a look at each of these in turn.

  We have implemented Bash Script to get Packet loss data and analyze the same. Screenshots and Code is attached in Appendix.

- *Latency Analysis*

  It is the delay between the sender and the receiver decoding it, this is mainly a function of the signals travel time, and processing time at any nodes the information traverses.

  In a computer network, it is an expression of how much time it takes for a packet of data to get from one designated point to another. It is sometimes measured as the time required for a packet to be returned to its sender.

  Latency depends on the speed of the transmission medium (e.g., copper wire, optical fiber or radio waves) and the delays in the transmission by devices along the way (e.g., routers and modems). A low latency indicates a high network efficiency.

  We have implemented Bash Script to get Latency data and analyze the same. Screenshots and Code is attached in Appendix.

- *Jitter Analysis*

  It is the variation in packet delay at the receiver of the information.

  Jitter is simply the difference in packet delay. In other words, jitter is measuring time difference in packet inter-arrival time. It is a specific phenomenon that normally exists in bigger packet switched networks. As a time, shift phenomenon, it usually does not cause any communication problems.

  Actually, TCP/IP is responsible for dealing with the jitter impact on communication. On the other hand, when we speak about Voice traffic and VoIP network environment this can be an issue.

  We have implemented Bash Script to get Latency data and analyze the same. Screenshots and Code is attached in Appendix.

- *Throughput Analysis*

It is the actual rate that information is transferred.

Throughput is a measure of how many units of information a system can process in a given amount of time. It is applied broadly to systems ranging from various aspects of computer and network systems to organizations. Related measures of system productivity include, the speed with which some specific workload can be completed, and response time, the amount of time between a single interactive user request and receipt of the response.

We have implemented Bash Script to get Throughput data and analyze the same. Screenshots and Code is attached in Appendix.

- *Connected Nodes*

In data communication, a physical network node may either be a data communication equipment (DCE) such as a modem, hub, bridge or switch; or a data terminal equipment (DTE) such as a digital telephone handset, a printer or a host computer, for example a router, a workstation or a server.

It is very important to analyze how many nodes are connected to a server to keep surveillance on the network. We have implemented Bash Script to get Connected Nodes data and analyze the same. Screenshots and Code is attached in Appendix.

DOC 2
HARDWARE AND SOFTWARE USED

2.1 Hardware List

The list of hardware used in this project is –

- Laptop (1 No.)
- Mobile (2 No.) – For Testing and Hotspot

2.2 Software List

The list of software used in this project is –

- Sublime Text: – Stable Channel, Build 3126
- GCC Compiler: - GCC 7.1
- Linux Terminal: - Stable pre-installed release.
- Guake: - version 0.7.0, Stable Release
- Python Packages
- Various C Libraries

DOC 3
RESULT AND CONCLUSION

3.1 Result

There are lot of modules included in this project to monitor a node's performance on different matrices. These different modules help us statistically analyze different performance variables.

By implementing different modules like Bandwidth Analysis, CPU Usage, Mem Usage, Packet Loss analysis, Jitter, Throughput and Latency analysis, we can easily conclude that our network monitoring tool gives basic statistical advice to the administrator to manage the network. It contains all the basic tool working on a shell environment, thus it carries a very small size and increased computability.

3.2 Conclusion

Network monitoring tools play a vital role in every network and it is a must have if an organization is to achieve its ROI. The benefits inherent in using network monitoring tools cannot be over emphasized. It provides that valuable network monitoring information needed for the management of any network. It enhances network stability, reliability, performance and allows for the controlling of the complexities in modern day networks.

Analysis of the survey on the evaluation of the network monitoring tools implemented showed that the research achieved its objectives to an appreciable level. Furthermore, even after implementing all these tools, there is still a need to reduce the computational cost and increasing of efficiency.

Future work also includes getting the output graphs on a browser to make it more colorful and user friendly.

Part - II

APPENDICES

CODES IMPLEMENTATION

- ## Trial and Division

Python Implementation for this algorithm:-

```
"""
Python Program implementation to find the factors of a number
by trial and division method.
"""

num = int(input("Enter the number to be factorized: "))
print("The factors of",num,"are: ")
test_factor = 1
while( num > 1 and test_factor<= num ):
        if (num % test_factor == 0):
            print(test_factor)
    test_factor=test_factor+1
```

REFERENCES

- Combs, G. “The Ethereal Network Analyzer”, “http://www.wireshark.org”. Accessed on January 12, 2009.
- Jacobson, V., Leres, C., McCanne, S. “Tcpdump”, available at: “ftp://ftp.ee.lbl.gov/”.
- Deri, L., Suin, S., Carbone, R. “Ntop – Network Top”, available at: “http://www.ntop.org”.
- “NfSen – Netflow Sensor”, available at: http://nfsen.sourceforge.net”.
- Burke, J. R. (2004) Network Management: concepts and practice, a hands-on approach, Prentice Hall, New Jersey, ISBN: 0-13-032950-9
- Cheswick, W. R. and Bellovin, S. M. (1994) Firewalls and Internet Security: Repelling the Wily Hacker, Addison-Wesley, Reading, ISBN: 0-201-63357-4
- ComLab Website (2006) ‘Tools for modeling the user-traffic’ [Online], Available: ttp://www.comlab.uni-rostock.de/research/tools.html [Accessed August 2006]
- Gaglio, S., Gatani, L., Lo Re, G. and Urso, A. (2006) ‘A Logical Architecture for Active Network management’ Journal of Network and Systems Management, vol. 14, No. 1, pp127- 146
- Held, G. (2000) Managing TCP/IP Networks: Techniques, tools and security considerations, Wiley, Chichester, ISBN: 0-471-80003-1
- IEPM Website (1999) ‘Monitoring with tcpdump’ [Online], Available: http://wwwiepm.slac.stanford.edu/monitoring/passive/tcpdump.html [Accessed August 2006]

- Leinward, A. and Fang, K. (1993) Network Management: a practical perspective, Addison-Wesley, Reading, Mass, ISBN: 0-201-52771-5
- Liska, A. (2003) The Practice of Network Security: Deployment Strategies for Production environments, Prentice Hall, New Jersey, ISBN: 0-13-046223-3
- Miller, M. A. (1999) Managing Internetworks with SNMP, M & T Books, Foster City, ISBN: 0-7645-7518-X
- Subramanian, M. (2000) Network Management: principles and practice, Addison-Wesley, Reading, Mass, ISBN: 0-201-35742-9
- WildPackets Web Site (2006) ‘WildPackets- Etherpeek’ [Online], Available: http://www.wildpackets.com/products/etherpeek/overview [Accessed August 2006]
- Windump Website (2006) ‘Windump: tcpdump for Windows’ [Online], Available: http://www.winpcap.org/windump/ [Accessed August 2006]
- Wisniewski, S. (2003) Advanced Network Administration, Prentice Hall, New Jersey, ISBN: 0-13-097048-4
- https://en.wikipedia.org/wiki/Network_monitoring
- http://www.solarwinds.com/basics-of-network-monitoring
- http://www.webopedia.com/TERM/N/network_monitoring.html

NETWORK MONITORING TOOL

VIT

UNIVERSITY

(Estd. u/s 3 of UGC Act 1956)

VELLORE ■ CHENNAI

www.vit.ac.in