Development of a graphical based traffic monitoring system for a Communication Network

Oyinloye O.E.1, Jegede Olubunmi1
1Computer science, department of mathematical science, Ekiti state university Ado-Ekiti, Ekiti state, Nigeria

ABSTRACT: The application of packet sniffer has gone a long way in proving that it is needed on all networking systems. Network monitoring is a vital part of modern network infrastructure management. Existing techniques either present a restricted view of network behavior and state, or do not efficiently scale to higher network speeds and heavier monitoring workloads. The network sniffer developed in this research was based on the type of capture library used by tcpdump; libpcap and the significance of having a monitoring system are better emphasized.

KEYWORD- Packet capture (pcap), graphical interface, Traffic analysis, Network Monitoring, Network analyzer, Packet sniffer,handler

1. INTRODUCTION

The network monitoring systems implemented on a communication network can provide essential data for network research and management. Traffic monitoring systems consists of equipment and gadgets that constantly monitors a communication network for slow or failing components and notifies the administrator (via email, sms or other alarms) in case of outages. It involves watching for problems frequently. Tools and services are as numerous and varied as the environments they guard and analyze.

Network monitoring for a corporate network is a critical information technology function that can save money in network performance, employee productivity and infrastructure cost overruns. A network monitoring system monitors an internal network for problems. A network monitoring system lets you know how well the network is running during the course of ordinary operation.

Network analysis is a process of capturing network traffic and inspecting it closely to determine what is happening in the network [1]. The traffic data encompasses the time and duration of the communication, the intricate shape of the communication streams, the identities of the groups communicating, and their location. The analysis of network traffic provides information about the user behavioral patterns thereby enabling network operators to understand the underlying traffic phenomenon. Various parameters are studied or closely monitored based on this phenomenon. Maximum utilization must be obtained from the capital.

Deciding specifically what to monitor on a network is as important as giving network monitoring a general thumbs up. A user must be sure that his corporate network topology map is up to date. That map should accurately lay out the different types of networks to be monitored, which servers are running which applications on which operating system, how many desktops need to be counted into the mix and what kind of remote devices have access for each network. Network traffic is a extremely filled with data and the approaches involved in measuring, analyzing, predicting and modeling solely depends on the end use of the data.

There are lots of monitoring systems currently existing but in this research the monitoring tool of focus is the tcpdump, this is because its output is in a dos environment and not easily understood by learners of networking. The

purpose of this research is to develop a more efficient system that has a graphical interface compared to the environment in which tcpdump is functional. This software did not get it source from tcpdump but it provides a better option for users who need to monitor their network.

This approach incorporates the use of simple freely available tools which put together helps to accomplish interesting results. This research is expected to presents the ways of utilizing available resources to analyze traffic using standard free packet capture and analysis software. The analysis of network traffic provides information about the user behavioral patterns thereby enabling network operators to understand the underlying traffic phenomenon. This research is significant in the area of network monitoring by providing a better way for both network administrators and networks instructors to easily teach TCP/IP related issues to students.

2. EXISTING SYSTEM

2.1. WIRESHARK

Wireshark is a free and open-source packet analyzer. It is used for network troubleshooting, analysis, software and communications protocol development, and education. Originally named Ethereal, in May 2006 the project was renamed Wireshark due to trademark issues. Wireshark is cross-platform using pcap to capture packets; it runs on various Unix-like operating systems and on Microsoft Windows [2]. Although wireshark uses a graphical interface to show its output but still it has facilities that cannot be fully understood by a someone just learning networking and it does not fully reach the objectives set out in this work. The key limitations of wireshark are:

i. Wireshark is packet-centric not data-centric.
ii. Wireshark doesn't work well with large network capture files [3].
iii. Wireshark does not capture what happens between frames such as problems in the inter-frame gap and does not show the start of the packet and preamble data or termination of the frame. These can result in problems wherby the engineer simply has no idea where the errors are happening [4].

During the course of my research it was noted that most of the existing application does not provide a platform allowing a novice in networking to understand fully what is going on in a network. For example some users have to undergo series of certification to be able to fully manage wireshark

2.2. SNIFF

Is a text mode wrapper written in Perl which takes the output from tcpdump and colour codes and formats it to aid readability . Aside from slightly refined output, similar display can be achieved through use of the standard tcpdump options. So it still has the same characteristics found in tcpdump whereby a clearer output is needed [5].

2.3. TCPFILTER

Provides a number of further formatting options for tcpdump output. Most usefully is a decode option which adds human readable packet analysis to the output but with these information further analysis can still be done [6]. Several advantages that can be gotten from the implementation of a monitoring system on a network. Due to this all necessary adjustment needed to make the packet analyzer work better should be made. Therefore in a university environment provision has to be made to enable upcoming networking student know what is happening on a network without necessarily going for the certification related to the packet analyzer in use.

2.4 NETDUDE

It combines the functionality of tcpdump and tcpfilter in a graphical user interface format (GUI) format. Layout and operation is similar to that of Ethereal with a listing of captured segments which may be selected to view further details. But it does not permit real time capture

[5]. NETwork Dump data Displayer and editor for tcpdump trace files. It is a graphical user [6] interface based tool that allows you to make detailed changes to packets in tcpdump trace file. It combines the functionality of tcpdump and tcpdfilter in a GUI format.

3. TCPDUMP COMMAND OUTPUT

Assuming that the system is in place and ready for capturing of packets and tcpdump is functioning, the first step will be to create a script that will run tcpdump with the arguments necessary to capture data. Making the tcpdump command and arguments part of a script will allow for changes to be a bit easier [7].Table 4.1 contains an example of the syntax of the tcpdump command that would capture all traffic on the given interface and a brief description of what each switch does.[8]

tcpdump -s0 -nn $EXCLUDED -G60 -w "%Y-%m-%d-%H%M.pcap" -i $IFACE -z pcap_parsing_script.sh"	
Switch explanation:	
-s0	Will capture the entire packet
-nn	Do not resolve host names or port names
-G60	Rollover to new capture file every 60 seconds
-w	Write capture to file, in this case the date in year, month, day, Hour, minute with .pcap extension (for consistency and to aid in clean up)
-i	Interface that tcpdump listens on, in this case, a script variable is used
-e	Interface the capture listens on (variable in script or network device)
-z	Post rotate command - for operation on the packet after capture

Table 4.1

The '–s' switch captures "snaplen" bytes of data from each packet. Giving it a value of '0' makes it capture the default value of 65535. This effectively means that the entire packet will be captured [9].

The '–nn' switch will prevent resolution of host names and port names.Preventing the resolution of domain names is important as it will reduce network traffic on the network.

The '–G' switch rotates the capture file specified by the '–w' switch. The number value is the amount of seconds for the capture to go before a new file is written. So a value of '60' will result in a new file every sixty seconds.

The '–w' switch writes the captured packets to a file and should include in the file name statement syntax for the time and date otherwise, the capture file will be overwritten .

The '–z' switch is used to make tcpdump run a command on the file it just finished writing to disk. This is commonly used to compress pcap files after capture, but can run any program on the system [10].

Figure 4.1 shows the output during a capture session.

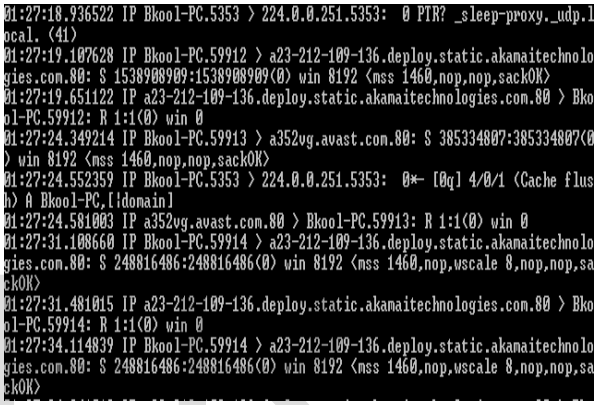

Figure 4.1

4. DESIGN

Packet capture is the process of intercepting and logging traffic. The packet analyzer (also known as a network analyzer, protocol analyzer or packet sniffer, or for particular types of networks, an Ethernet sniffer or wireless sniffer) developed for these research is a computer program that can intercept and log traffic passing over a digital network or part of a network. As data streams flow across the network, the sniffer captures each packet and decodes the packet's raw data. Figure 5.1 shows the model of the system.

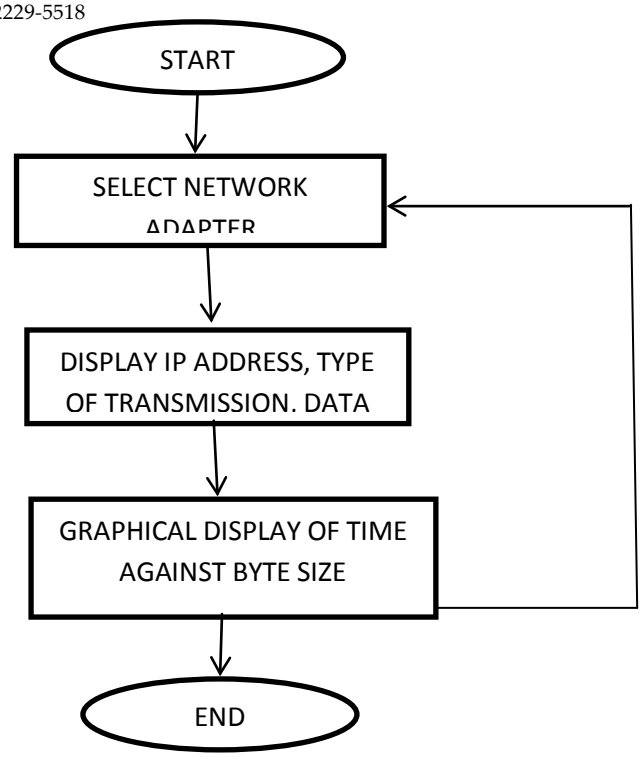

Figure 5.1

The captured information is decoded from raw digital form into a human-readable format that permits users of the protocol analyzer to easily review the exchanged information. Protocol analyzers vary in their abilities to display data in multiple views, automatically detect errors, determine the root causes of errors, generate timing diagrams, reconstruct TCP and UDP data streams, etc. The system is being developed using java as the main tool. The application is programmed to listen on several interfaces like USB port, wireless and Bluetooth. This program takes the form of a GUI which supports the completion of a number of options supported by tcpdump. The design of the system features an interface which can be easily understood by a network administrator. It is effective in removing the need for the end user to be familiar with the command line syntax for tcpdump and provides a point of reference for the implementation of relevant options.

5. ARCHITECTURE

The program consists of three major modules and five sub-modules (figure 6.1). These are Packetsniffer, MainGui and Handler. They are driven from a control class. It was important to separate each of these modules because this allows for easy maintenance and simplifies future development of the system.

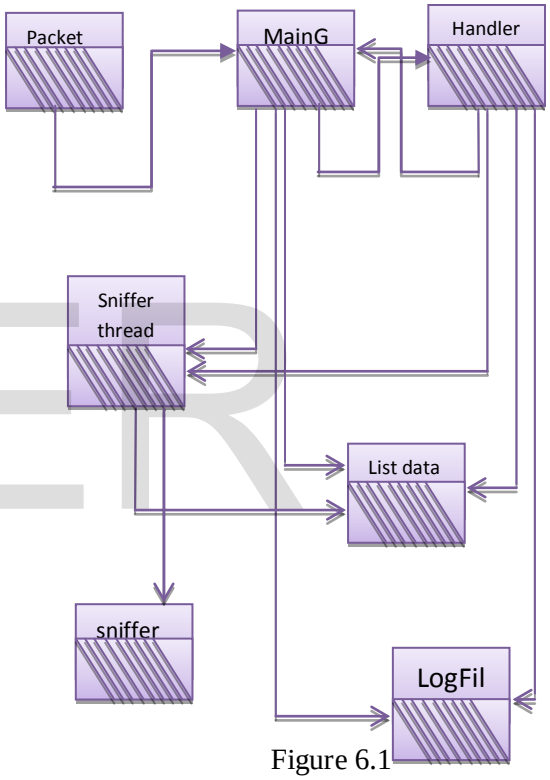

Figure 6.1

- **PACKETSNIFFER**
  This part of the system brings the entire software into play. Its runs the MainGui and starts up the necessary tools needed for the packet capture.
- **MAINGUI**
  It controls the display on the screen. It is also works along with the handler in order to give a proper display and to determine the actions to be performed when a specific menu is clicked.
- **HANDLER**

The handler object takes log messages from a logger and exports them. It can write them to a file or can send it to the main interface for it to be displayed. It controls and brings to play the menus on the display screen.

6. SYSTEM OUTPUT

The system design has being tested using the connection between a host computer and a device that has internet connection. A wireless connection was used to connect the two devices together. The initial capture interface is shown below.

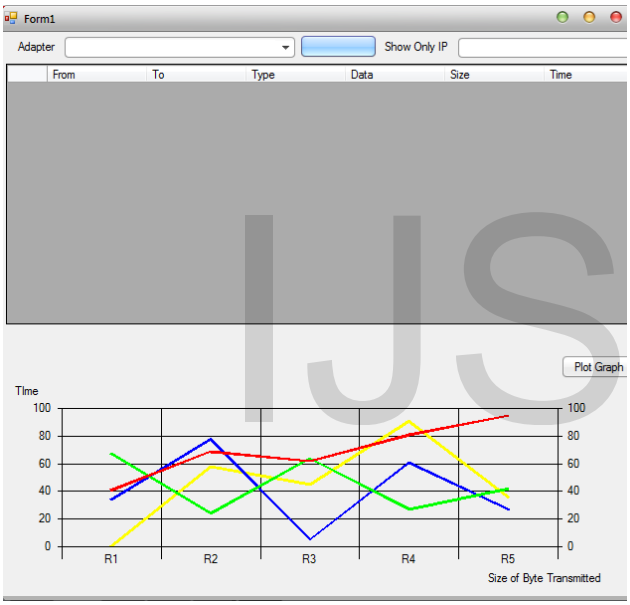

Figure 7.1 before capturing

Figure 7.1 shows the initial interface of the program before a capture session. The interface features series of options that allows the users to fully understand what is going on during a packet capture.

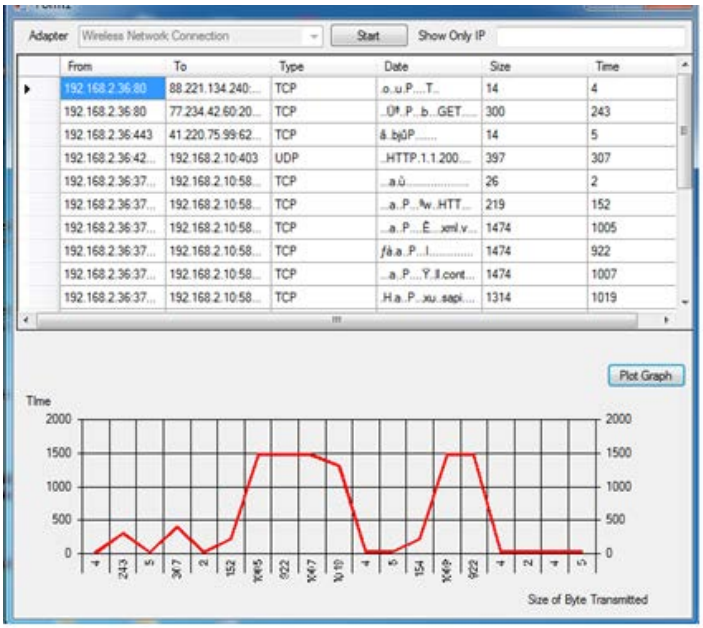

Figure 7.2 during capture session

7. CONCLUSION

The deployment of the final system design on a communication network was campared alonside tcpdump. The advantages were obvious due to the graphical interface that was used. Also there was no need for the syntax associated with tcpdump, which makes the application easier to use. Although the system is fully functional, there is need of developing the system to be able to decrypt information sent in a packet.

We have shown that efficient network monitoring is possible, to some extent, without introducing complex new infrastructure. This is accomplished using freely available online software that allows the filter mechanism to perform monitoring functions. In this way, monitoring functions can safely and efficiently execute in the system. Packet sniffer can be used to support existing and new applications while allowing commodity systems to satisfy the growing demand for network monitoring at increasing network speeds.

Other areas that can be studied and developed included the diagrammatic display of captured packet on a network, in

real time which can show the information passed from the sender to the reciever.

REFERENCES

[1] B. R. Chang, and H. F. Tsai, *"Improving network traffic analysis by foreseeing data packet-Flow with hybrid fuzzy-based model prediction,"* Expert Systems with Applications, vol. 36, 2009.

[2] Pallavi Asrodia, Hemlata Patel, Network Traffic Analysis Using Packet Sniffer, *Pallavi Asrodia, Hemlata Patel / International Journal of Engineering Research and Applications, Vol. 2, Issue 3, May-Jun 2012, pp.854-856.*

[3] Forensicswiki (2011), www.forensicswiki.org/wiki/Wirreshark. Retrieved on October 2, 2014.

[4] Absolute analysis, (2011). "*Is your wireshark trace missing critical data?*" Retrieved from www.absoluteanalysis.com/blog on October 2, 2014

[5] Stew Benedict. *Making sense of tcpdump with add-on enhancements*. Builder.com, March 2002. http://builder.com.com/5100-6387-1045521.html.

[6] Felix Fuentes and Dulal C. Kar. *Ethereal vs. tcpdump: a comparative study on packet sniffing tools for educational purpose.* Journal of Computing Sciences in Colleges, Volume 20(Issue 4), 2005.

[7] Derek Banks, *"Custom Full Packet Capture System"*,February 27, 2013

[8] Tcpdump/Libpcap Web page. http://www.tcpdump.org/.

[9] Jacobson, L. (2009). *Tcpdump*. Retrieved from http://www.tcpdump.org/tcpdump_man.html on august 12, 2013.