Computing, Information Systems, Development Informatics & Allied Research Journal
Vol. 7 No. 4, December, 2016 - www.cisdijournal.net

# PH SNIFF: A Packet Sniffer for Network Monitoring and Traffic Analysis

**[1]Obiorah, P. C. [2]Eke, B. O. [3]Oghenekaro, U. L.**
[1,2,3]Department of Computer Science
University of Port Harcourt
Choba, Port Harcourt, Nigeria
[1]philip.obiorah@uniport.edu.ng, [2]bartholomew.eke@uniport.edu.ng, [3]linda.oghenekaro@uniport.edu.ng,

**ABSTRACT**

The daily growth in network infrastructure and significant dependence on computer networks for everyday communication has made it pertinent to monitor, examine and assess network traffic. The data going through networks is a valuable source of proof for system administrators to fish out invaders and abnormal network connections. The need for an application to obtain network information about the source of traffic, the protocols in use, and the destination of the traffic can be crucial in solving congestion problems and network security challenges. This paper presents the use of open source libraries in the development of packet capture and network analysis application. An open source library, Java Packet Capture (JPCAP), was used for capturing and sending network packets with Window Packet Capture (WinPcap), and Live Graph API in the development of PH-SNIFF application. The application written in Java Programming Language can capture packets and display packet statistics in an easy to use graphical user display (GUD). It also occupies less memory space and displays real-time packets flow graph. Results obtained from using the software shows the effectiveness and efficiency of network monitoring and traffic analysis through the utilization the system.

**Keywords-** Packet Sniffer, Jpcap, Winpcap, Network Monitoring, Traffic Monitoring

**CISDI Journal Reference Format**
Obiorah, P. C. Eke, B. O. & Oghenekaro, U. L. (2016): PH SNIFF: A Packet Sniffer for Network Monitoring and Traffic Analysis. Computing, Information Systems, Development Informatics & Allied Research Journal. Vol 7 No 4. Pp 129-138
Available online at www.cisdijournal.net

## 1. INTRODUCTION

The speedy evolution of the Internet has made network management a thought-provoking challenge. The capability of a monitoring system to deliver precise information about the nature and sort of the network traffic can be very indispensable. Information regarding entities producing the maximum traffic, the protocols used, and the source and destination of the traffic can be essential to fixing network glitches. Many network administrators spend much time, trying to know what is degrading the performance of their network (Abiona, 2009). With the advancement in network technology, the management, maintenance and monitoring of network is important to keep the network smooth and improve its safety and economic efficiency (Pallavi & Hemlata, 2012). Thus, for this purpose, a packet sniffer is used. Packet sniffing is necessary for network monitoring to troubleshoot and to keep network logs. Packet sniffers are suitable for analyzing network traffic over wired or wireless networks. System administrators use Packet Sniffers for network management and diagnostics. It can also be used to sniff a packet off from a network to obtain specific information, such as username and password to gain unauthorized access to the network (Su, 2004).

Moreover, there are several other uses of this type of technology such as in the field of computer hacking and network security. A hacker may obtain someone's information maliciously by connecting to their router and viewing their complete information as they pass through his computer. By doing that, sensitive data such as credit card number, username, passwords and any other data of interest can be ripped off and used without the consent of the owner. In contrast, network administrators could utilize this technology to discern network intrusion attempts, observe network usage, and analyze network hitches.Packet sniffers can be useful for both legitimate and unlawful activity. A legitimate packet sniffer is used to provide network security and assist administrators in network management. It is also used as an analytical tool for network backup systems and to scrutinize the network system for any security breaches. An illicit packet sniffer is utilized by a programmer to gain unapproved access to delicate data and subtle information on a system. It is typically introduced without the knowledge of the network administrator and hides away in various areas of the system with the end goal of keeping an eye on sensitive information and taking the data packets that pass over the network (Frieden, 2007).

When a user considers the growing need to secure a network, it has given much impetus to implement adequate network security policies and have consistent network monitoring continuously. Thus, designing and implementing a packet sniffer system for network monitoring and traffic analysis is of the essence. In this paper we present PH–SNIFF, a packet sniffer application developed using an open source libraries which include Java Packet Capture(JPCAP), Window Packet Capture(WinPcap) and Live Graph API. This provides a cheap, efficient and user friendly Graphical User Interface (GUI) system.
The system equally takes less memory space in execution and can easily be impeded in application in other to boost end-user experience and ensure a productive network environment.

## 2. RELATED WORKS

Talitha (2003), offers an outline of the FBI's Carnivore electronic surveillance system. The Carnivore is a surveillance technology, a software program accommodated in a computer unit, which is introduced by legitimately approved FBI operators on a specific Internet Service Provider's (ISP) network. It is utilized together with a tap on the ISP's system to "intercept, filter, seize and decipher digital communications on the Internet". The system is termed as a "specialized network analyzer" that operates by "sniffing" a network and replicating and storing an acceptable section of its traffic. As indicated by the FBI, the "Carnivore chews on all data on the network, but it only actually eats the information authorized by a court order" (Talitha, 2003)]. Numerous individuals regard this software as an invasion of privacy and as such it is highly controversial. Other worries are that granting the government this sort of control would permit it to have eventually the power to seize control of the internet, in dangerous cases. Nevertheless, it could only be utilized for very explicit purposes. Within the confines of the law, before using Carnivore to acquire information concerning an individual, there ought to be misgiving of fraud, internet rivalry, surveillance, child pornography or manipulation, and terrorism. Although Carnivore is not for commercial or private uses, it remains a significant software to comprehend since it has numerous consequences for the future.

Cardiagliano, 2013) developed a network traffic recorder application -, 'n2disk', capable of dumping 10 Gbit traffic to disk using commodity hardware and open- source. It has been designed to write files into disks for very extended periods of time. A maximum number of the distinctive files that will be written for the duration of the execution, have to specified, and if n2disk reaches the maximum number of files, it will start recycling the files from the oldest one. Though this way one can have a comprehensive outlook of the traffic for a static temporal window, knowing in advance the amount of disk space needed. Nevertheless, 'n2disk' is only obtainable in Linux versions. Also, it is not free but rather an expensive commercial software and has an installation on the file size of 7.80 MB. Colasoft LLC on December 15, 2015, announced the availability of Colasoft Capsa Network Analyzer v8.1. Capsa v8.1 created by Colasoft is a network analyzer and packet sniffer that catches unique packets continuously, decipers and break them down, evaluates, and make a diagnosis from captured packets, and then presents the outcomes in straightforward interpretations, visualized diagrams and organized reports, by this means to get the network administrators to be acquainted with the network status completely and rapidly (Capsa, 2016). It sets the network card into promiscuous mode and registers all the packets it gets on the wire. Running counts are presented displaying information about the numerous packets on the network. It is a user-friendly program which displays data in a very easy-to-read way. However, Capsa does not work cross-platform; it is limited to Windows environments. Moreover, it is quite costly. Though a free version is obtainable with limited features, and it has a huge installation file with 49,507,482 bytes.

Consequently, PH-SNIFF: a packet sniffer application is developed to address the limitations cost, memory space, easy to use Graphical User Interface, and platform dependence, to provide improved a tool for analyzing network traffic over wired or wireless networks and enhance network administrators experience in packet sniffing.

## 3. DESIGN AND METHODOLOGY

An object-oriented methodology is employed in the design and implement the system requirements. It elaborates the analysis models to produce the implementation specifications of the system. The program is written in Java Programming Language to leverage its platform independence capability.

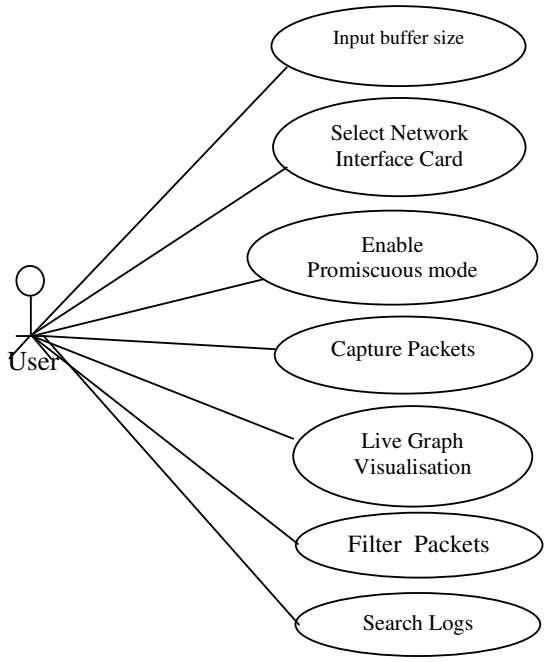


**Fig. 1: Usecase Diagram of the System**

The Usecase diagram of the system is shown in figure 1. The system allows the user to specify the size of the buffer; select a network interface of the intended device to a defined IP address, words or Ports. Also, the users can have a live Graph visualization of live capture.

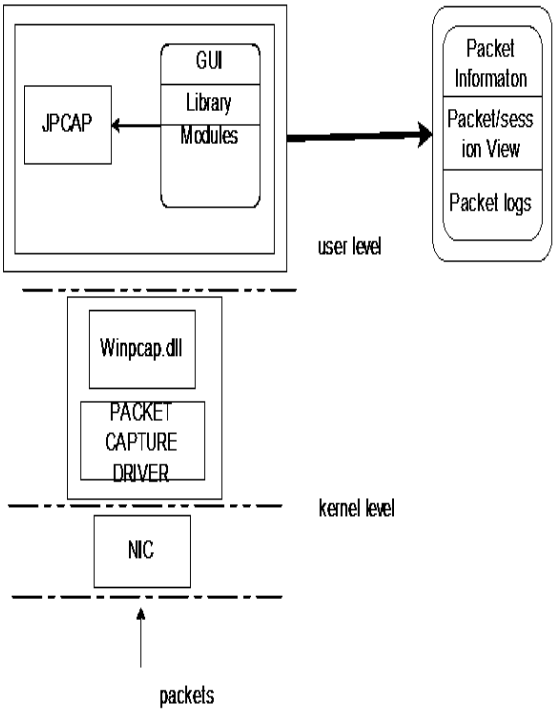


**Fig. 2: The Architecture of the System**

The system architecture is comprise mainly of the user level and the kernel level. The user level is an aggregate of three components, namely, the PS component which mainly provides the graphical user interface (GUI) and the underlying configuration of the system. The java Modules component which essentially handles the packet filter configurations and packet logs; the Library component which makes use of the JPCAP API to capture packets and display packet statistics. At the kernel level the Winpap driver extends the operating system to provide low-level network access via the Network Interface Card (NIC).

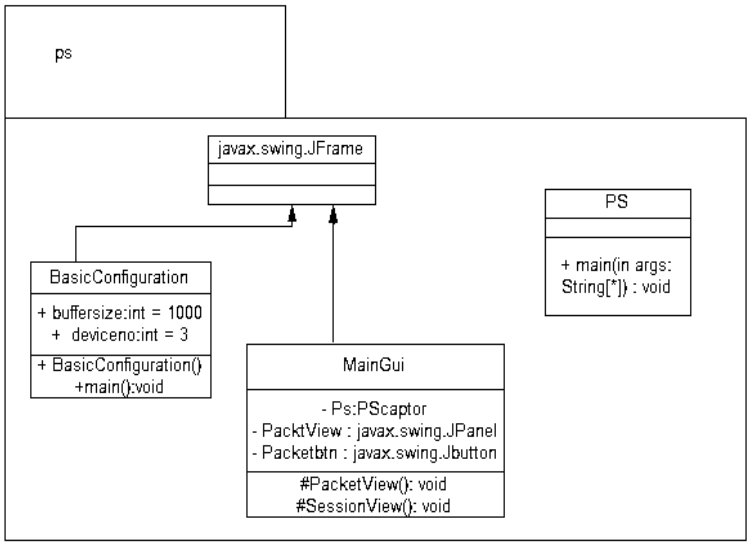


**Fig. 3: Class Diagram of package ps**

Figure 3 shows, the class interaction in the package *ps*. The *MainGui* and *BasicConfiguration* class extends javax.swing.JFrame

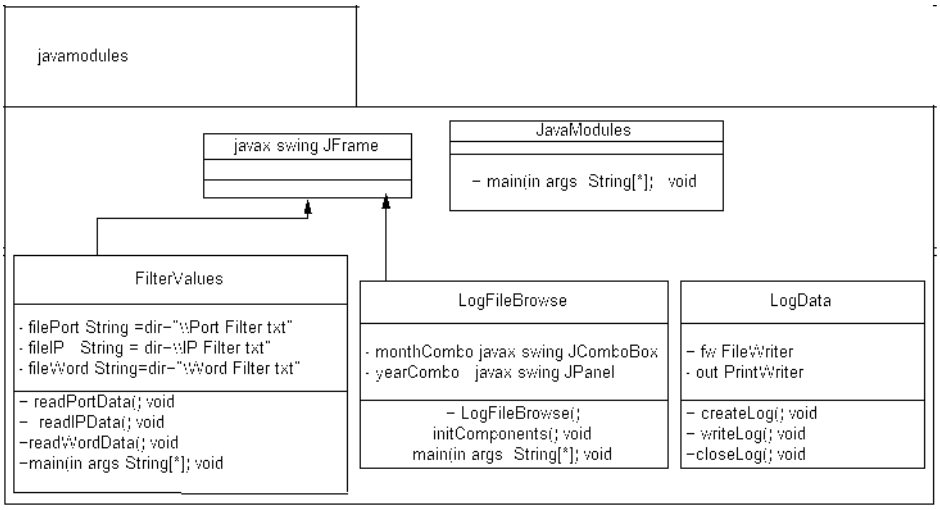


**Fig. 4: Class diagram of javamodules**

Figure 4 shows the interaction within the javamodules package, its various classes and data members.

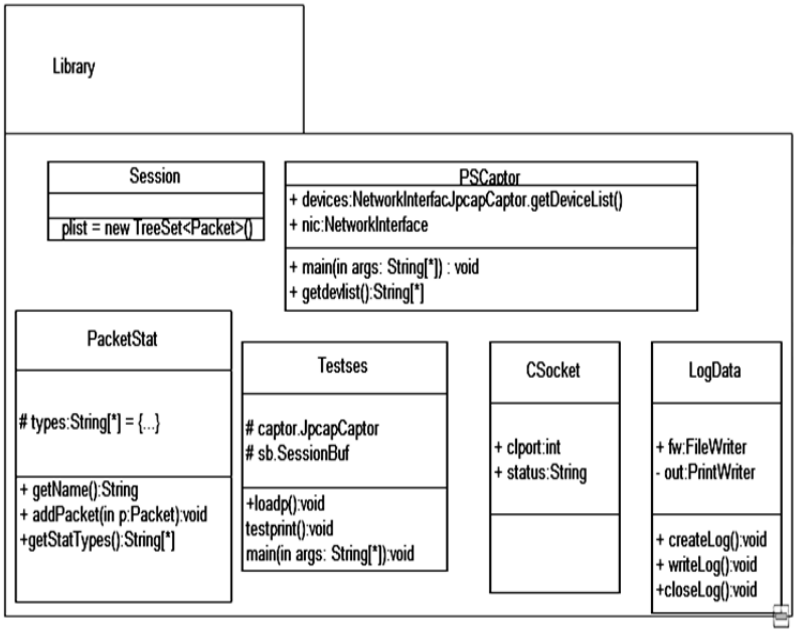


**Fig. 5: Class diagram of javamodules**

Figure 5 shows the interaction within the library package, its various classes and data members.

## 4. EXPERIMENTS AND RESULTS

Experiments were conducted in a small functional shared Internet Network, consisting of three hosts connected via tethering over Wi-Fi, (Mobile Hotspot) using iPhone 4s (MD237LL/A). The three hosts, A and B and C, comprise of two Intel® Core™ i5-2410M @ 2.30GHz with 4.00GB RAM and 64 bit Windows 7 operating system and an AMD E1-1200 APU @ 1.40GHz with installed memory of 2.00GB (RAM) and a 64 bit Windows 8 operating system, with the Host C running the PH-SNIFF packet sniffer.

### 4.1 Basic Requirements

Java Runtime Environment (JRE -7 or higher) and Java Development Kit(JDK 1.7 or greater). JRE could found at http://java.com/en/download/index.jsp.; Windows packet capture library(Winpcap).Download WinPcap from http://www.winpcap.org/; Java Packet Capture (JPCAP) Download JPCAP from http://jpcap.software.informer.com/download/

### 4.2 Execution

The Basic configuration form consists of input fields that allow the user set the buffer size, select a desired network interface card, and then enable the choice promiscuous mode.

a) *Input Buffer Size*: This allows storing the captured data in a specified memory side. Here data may be retained until it is full, or in a rotation method in such a manner that the newest data replaces the oldest data
b) *Selecting Network Interface Card*: The user has a choice to select which interface to capture packet from, either the Ethernet interface or the wireless interface or any other Network Interface card on the system.
c) *Ok Button*: These values form the basic configuration form is passed as values for data processing and packet capturing.
d) *Cancel Button:* This rest in current entries to its default values.

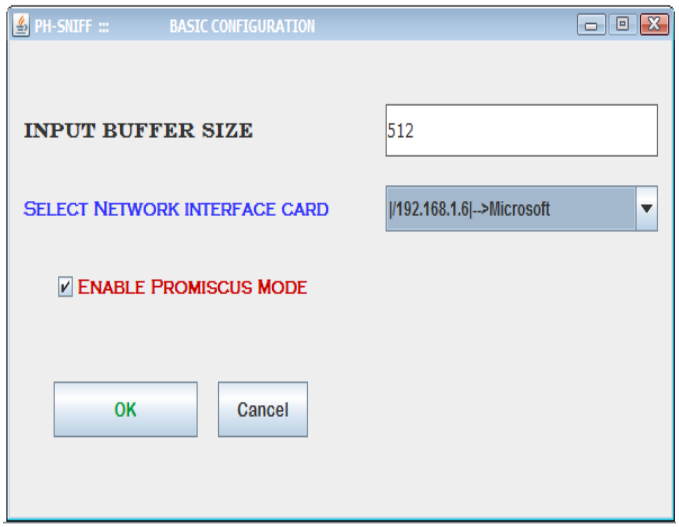


**Fig. 6: Basic Configuration**

Figure 6. Showing the Basic Configuration form which allows the user to input buffer size, select network interface card, & enable promiscuous mode with values

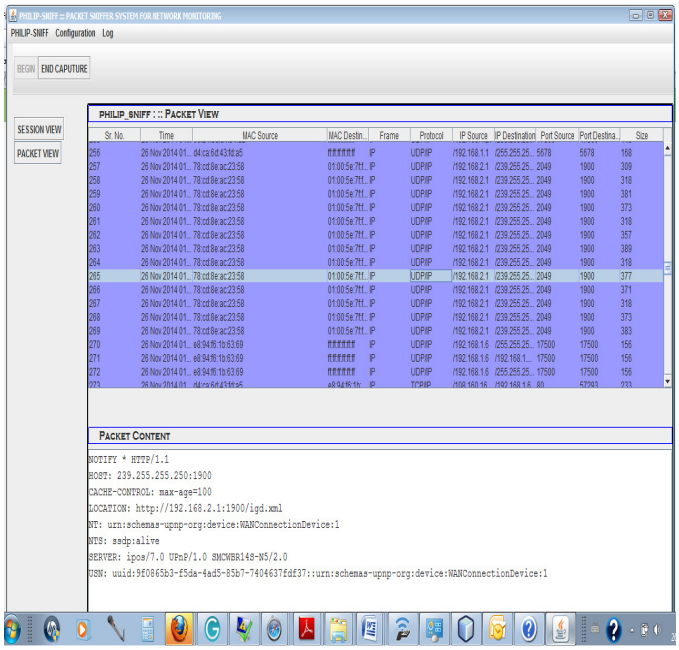


**Fig. 7: Packet view of captured packets**

Figure 7 shows a packet view of captured packets indicating Serial No of packet; the time at which to packet was captured; the MAC address source and destination address of the packet; the frame of the packet; the IP source and destination address; the size of the packet; the port source and destination source of the packet.

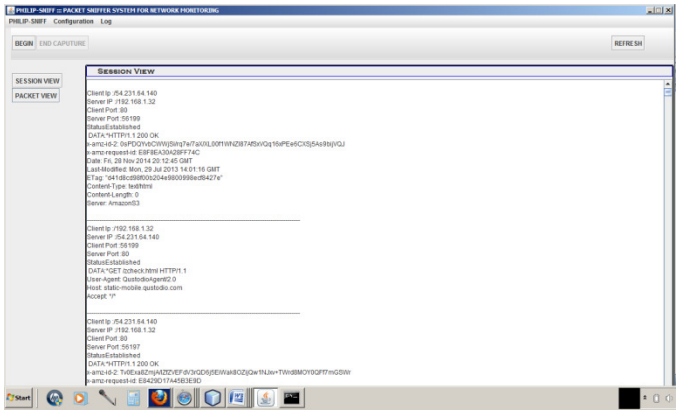


**Fig. 8: Displays a session view of the captured packets**

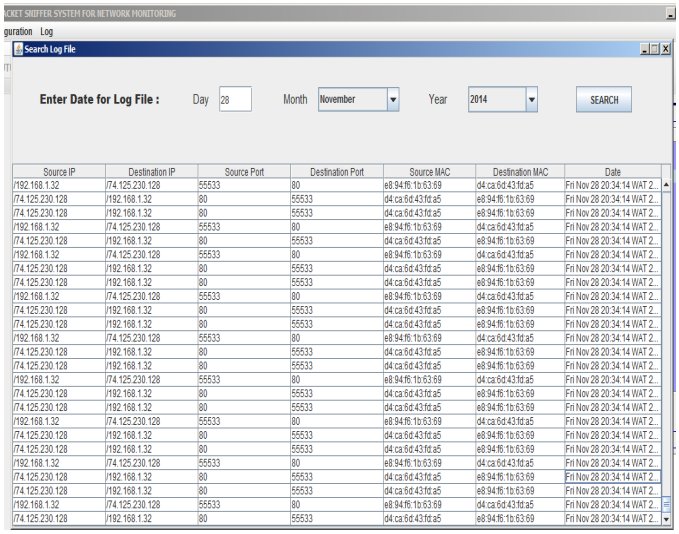


**Fig. 9: Search Log File**

The search Log File displays result from Search logs. Through wich network administrators could monitor activites in the network.

*4.2 LiveGraph API*

A data visualization and analysis framework is integrated to present a graphical view of packet sniffing in real time . The LiveGraph plotter application plots graphs based on data contained in a data stream in real-time.

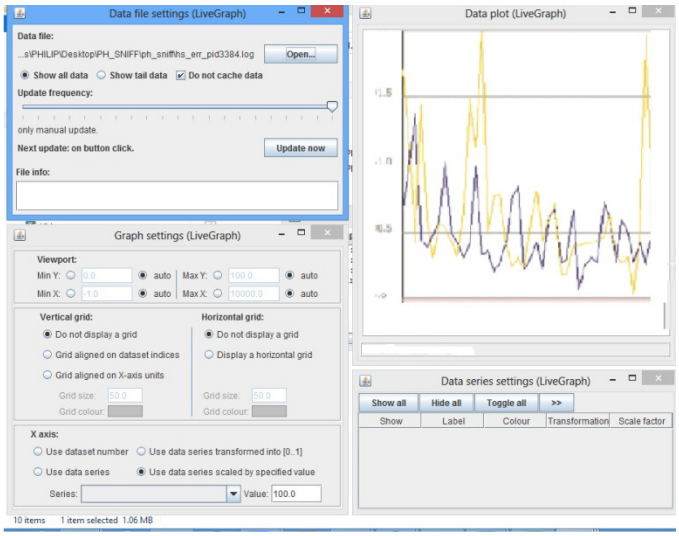


**Fig. 10: Live graph of the packets and protocols**

A live graph of the packets and protocols used on the application layer at the date of packet sniffing.

From the preceding, PH: SNIFF application captures live packet information in promiscuous and non-promiscuous mode; displays all available network interfaces card and enables the user to select one from which data would be captured. It shows logs, saves the statistics of the received packets for future retrieval; and presents real-time packets flow graph.

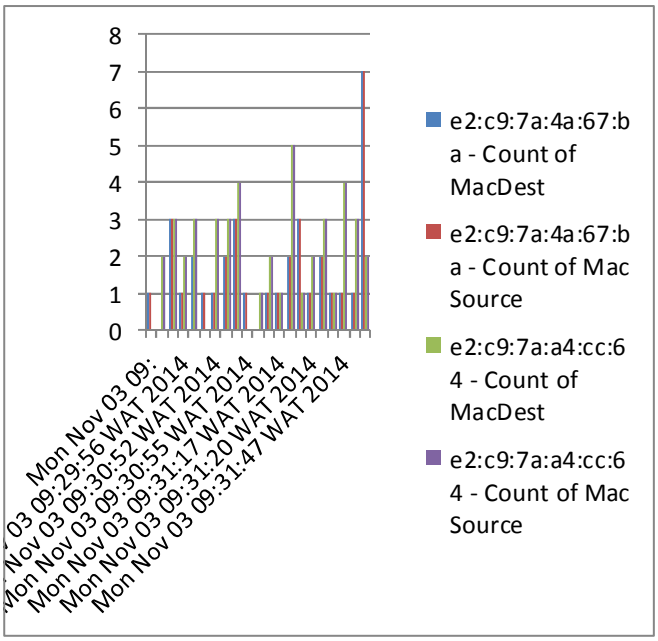


**Fig. 11: A chart showing a count of MAC address source and destination against time and date.**

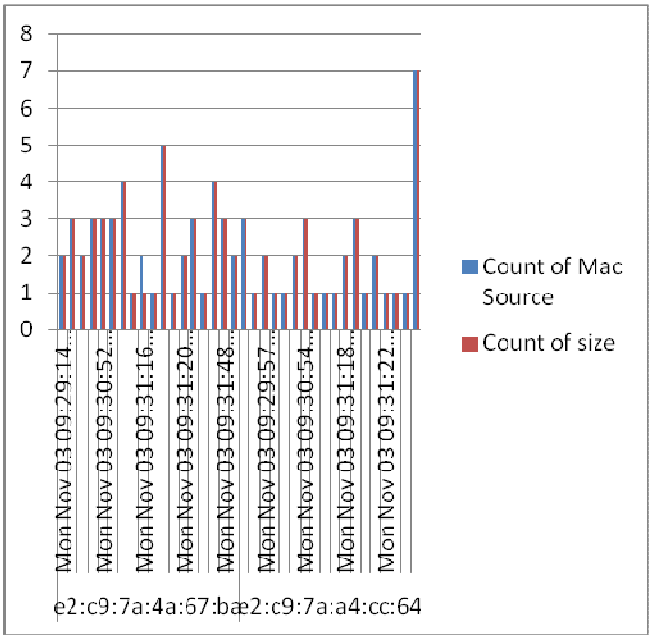


**Fig. 12: A chart indicating a count of MAC address source and packet size utilized against time.**

A Pivot Chat analysis of the sniffer log files presents a graphical view of users activity in the network. The chart below shows a count of MAC address source and destination against time and date.

This application has been used to test the capturing capabilities of JPCAP API with WinPcap. The result of the monitoring is up to mark as it captures all types of packets demanded by the interface. Compared to similar works, this application shows the layer and the proto- cols involved in sniffing in a real time graph. The Live Graph setting would enable network administrators to isolate protocols to visualize, change the color and grid size of the graph

## 4. RECOMMENDATION

In our rapid developing and advancing society, it is vital that people, government organizations and associations explore the full advantages and value of information Technology. Governments and stakeholders should organize training programmes for their network administrator to equip them with knowledge of the trending network technologies. The University authorities should facilitate the improvement of this system through development so that it can be used to monitor the network traffic prompting the network administrators to take affirmative actions only at times it is needed.

## 5. CONCLUSION

The development of a scalable packet capture tool for traffic monitoring and analysis system is presented in this paper. The system is capable of monitoring and analysing the network. The sniffing tool PH SNIFF will listen for packets as they are sent through the network layer and presents real-time packets flow graph. It has a very rich and user-friendly GUI developed in Java Swing Technology. Thus it is entirely easy to use. It is highly economical regarding memory use as an installation file for PH SNIFF is only 413KB. Also, based on its object-oriented design, any further changes to it can be easily adaptable.

## REFERENCES

[1] Abiona, et al. (2009). A Scalable Architecture for Network Traffic Monitoring and Analysis Using Free Open Source Software. International Journal of communications, Network, and System Sciences., 2, 1-2. Retrieved from http://www.scirp.org/journal/Paper Download.aspx?paperID=696
[2] Abiona,et al, (2009) Network Traffic Analysis:A Case Study of ABU Network. Retrieved from http://www.iiste.org/Journals/index.php/CEIS/article/download/5041/5140
[3] Capsa Review by Crunchaear - Colasoft, Retrived from http://www.colasoft.com/capsa/crunchgear-capsa-review.php (accessed August 10, 2016).
[4] Cardigliano, et al, (2013) n2disk ? ntop, http://www.ntop.org/ products/traffic-recordingreplay/n2disk/ (accessed August 08, 2016).
[5] Frieden, R. M. (2007). Internet Packet Sniffing and Its Impact on the Balance of Power. Retrieved March 14, 2014 from Selected Works of Rob Frieden: http://works.bepress.com/rober_frieden/2
general_accessibility/accessibility/CAVI-projects/Speech%20 (accessed August 08, 2016).
[6] Pallavi, A., & Hemlata, P. (2012). Network Traffic Analysis Using Packet Sniffer . International Journal of Engineering Research and Applications , 2(3), 854-856..
[7] Speech Friendly Packet Sniffer - CUCAT, http://www.cucat.org/
[8] SU, C. C. (2004). Speech Friendly Packet Sniffer. Kent S, Bentley WA 6102, Australia: School of Electrical and Computer Engineering Curtin University of Technology
[9] Talitha Nabbali (2003). Going for the Throat: Carnivore in an Echelon World – Part http://papers.ssrn.com/ sol3/papers.cfm?abstract_id=1974304 (accessed August 12, 2016).
[10] The History Of Packet Sniffing Information Technology Essay, https://www.ukessays.com/essays/information-technology/the-history-of-packet-sni (accessed August 08, 2016).
[11] What is a Packet Sniffer and How Does It Work? - Spamlaws, http://www.spamlaws.com/how-packet-sniffers-work.html (accessed August 10, 2016).