ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

An ISO 3297: 2007 Certified Organization *Vol.5, Special Issue 2, April 2017*

An International Conference on Recent Trends in IT Innovations - Tec'afe 2017

Organized by

Dept. of Computer Science, Garden City University, Bangalore-560049, India

Portable Network Monitoring Probe Using an Internet Embedded Microprocessor

Madhavi Giridhar D, Mamatha S Upadhya

Asst. Professor, Dept. of Computer Science, Garden City College, Bangalore, India

Asst. Professor, Dept. of Computer Science, Garden City College, Bangalore, India

ABSTRACT: Organization of a (Tele-communication Network) network demands the requirement to monitor and control devices on LAN and MAN Networks. Components used to perform this function are referred to as network probes. This Research paper looks into the model of a network probe which uses an embedded internet microprocessor to gather and transfer data to a network monitor. The component uses the TINI Internet Interface evolved by Dallas Semiconductor. This device was selected as the Ethernet driver has a dissipated mode which potentially allows the Seize of all packets transmitted on the Ethernet connection of the embedded internet microprocessor. A framework system of the remote Network Monitoring System is progressed; personal computer and some preliminary results are presented using Java based software for server and client. The program client was not actualize on the TINI because of complications associated with memory restriction of the selected embedded internet microprocessor, with the TINI only having simple functionality of packet seizure demonstrated. Future hardware design requisite for the TINI board are required to provide sufficient resources on the embedded processor to grant for effective packet capture and storage.

KEYWORDS: TINI, Portable, inexpensive, network, monitoring, probe, internet, embedded, Microprocessor.

I. INTRODUCTION

Network survey is essential in automated network administration. The aim of network monitoring is to collect information concerning the status and performance of nodes which are part of a managed network. The ISO describes the key functions of network management as: fault management, accounting management, configuration and name management, performance management and security management. Network survey is typically the heart which supports the five functional areas of the network monitoring framework. According to Stallings [1], the data that should be achieved from network monitoring systems can be categorized into three main groups. The first group is Static data that will not change often, such as the number of ports on a router. The second group is Dynamic data, which is regularly changing, such as transfer of packets on a network. The third group is Statistical data which is imitative from dynamic information such as packets transmitted by a node to another node in the network. The assemblage of static information can be made available to the network monitor through a software agent installed in the monitored element. Alternately it can also be attained from a proxy agent. Dynamic and statistical information is also collected and stored in a similar manner to static information.

The architecture of a NMS (Network Management System) can be explained from a functional perspective [2]. This model indicates the monitoring system as four functional blocks and is shown in Figure 1. These are the monitoring application, manager function, agent function, and managed objects.

The TINIs400 evaluation board was used by prototype system, provided by Dallas Semiconductor and integrated feasible packet capturing libraries such as WinPcap [3] and Jpcap [4] to highlight how such a system could be potentially deployed in a switched network. Section II encompasses an examination of the TINI microprocessor used. Section III considers the software used in the Personal Computer to provide a Remote Network Management System. Section IV describes the developed TINI ethernet driver which was able to confiscation packets.

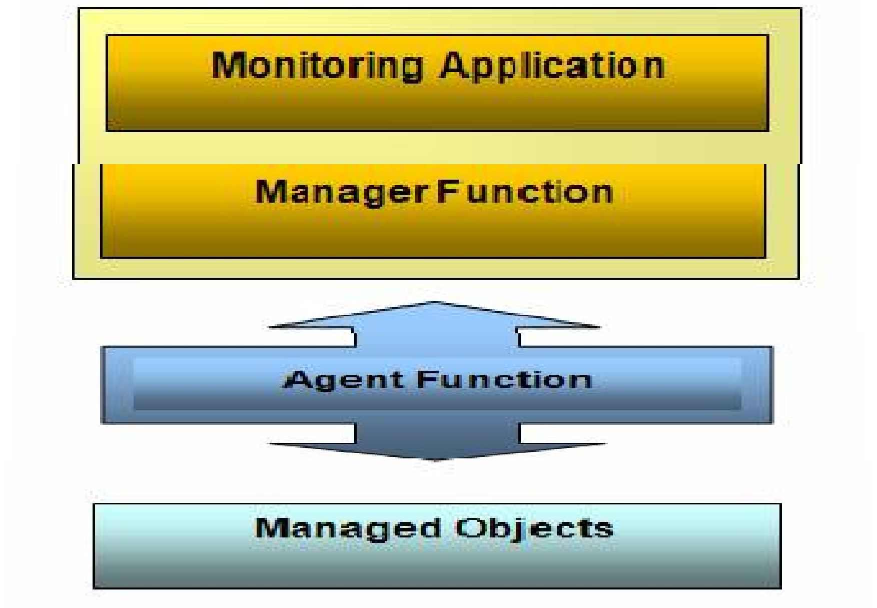

Figure 1: Architecture of a Network Management System [2]

II. THE TINY INTERNET INTERFACE

A. TINIs400, TINIm400, DS80C400

Figure 2 shows the devices of the TINIs400 socket and TINIm400 unit which was installed as a portable internet probe in this prototype.

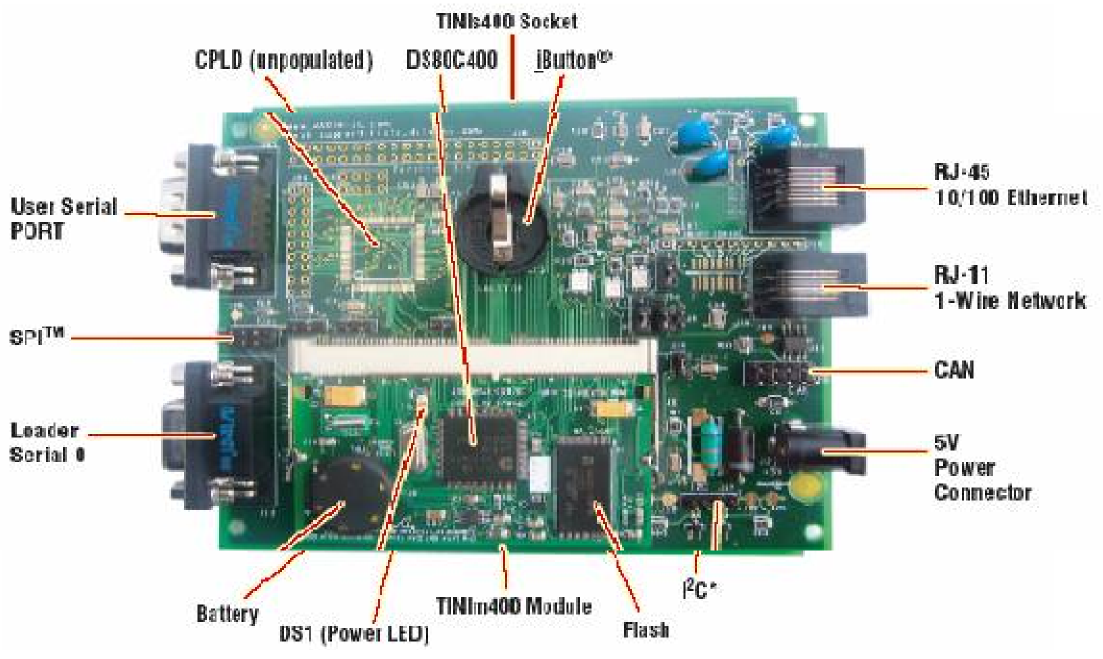

Figure 2: Components of the TINIs400 evaluation board [5]

ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

An ISO 3297: 2007 Certified Organization Vol.5, Special Issue 2, April 2017

An International Conference on Recent Trends in IT Innovations - Tec'afe 2017

Organized by

Dept. of Computer Science, Garden City University, Bangalore-560049, India

DS80C400 microprocessor was designed and manufactured by Dallas Semiconductor used in applications of internet. Microprocessor used is 8051 device and it has 75MHz maximum clock speed. A 24 bit addressing scheme is used to permit the access up to 16MB of adjoining memory, though the board come up with only 1MB of Static RAM and 1MB of flash RAM. The Operating System (TINI) is located in the ROM of the DS80C400 microprocessor and it also includes the functionality of the IPv4/6 network stack. The network stack provides for a maximum of 32 concurrent Transmission Control Protocol connections with a transfer rate of up to 5Mbps through the Ethernet MAC [5].

B. Packet Capture aspect of TINI

The aspect of the TINIs400 is to seizure the ethernet data packets and sends this to the monitoring application which is to be run on a PC through an Ethernet link. The PC would have the analysing unit, the configuration unit and possibly an RMON MIB implemented. As indicated in Figure 3, the consolidation of the TINIs400 and the PC with installed RMON probe software provides a possible implementation of an economical and highly portable RMON probe

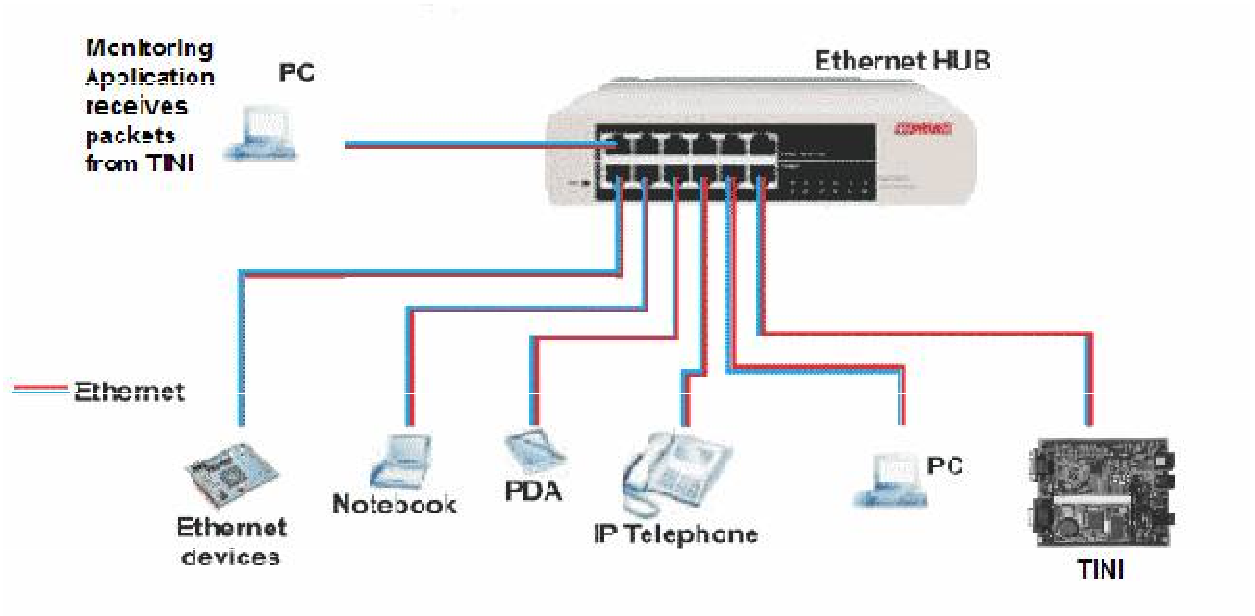

Figure 3: Basic concept of using the TINI and a PC to capture and process packets at the MAC layer

The microprocessor DS80C400 also has a built-in Ethernet (media-access controller) MAC with an standard MII (media independent interface), the media independent interface defines input output lines that allow the microprocessor DS80C400 to communicate with the physical layer interface or NIC. The media independent interface contains two basic blocks. The two blocks are the media independent interface input output Block and the media independent interface Management block. The function of the media independent interface Input output Block is to deliver and receive Ethernet frames to and from an external Network Interface Card. Through the media independent interface Input output the microprocessor DS80C400 is able to support all of the transfer and receive data transactions between the Microprocessor DS80C400 MAC and the external Network Interface Card.

The DSTINIm400- is built with the microprocessor DS80C400 and this microcontroller was ironically designed with Ethernet networking as its first objective or application area [6]. With the support of the media independent interface, the microprocessor DS80C400 permits for the integration of the Ethernet Medium access control and as a result permits for the capacity to access the network stack. Assembly language was used to appliance an Ethernet interrupt handler that would be operated in pro-mises mode so that all passing packets could be seized by the TINI and retransmitted to the PC. Assembly code was also used to send and receive Ethernet packets to the PC with the supervising application.

ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

An ISO 3297: 2007 Certified Organization Vol.5, Special Issue 2, April 2017

An International Conference on Recent Trends in IT Innovations - Tec'afe 2017

Organized by

Dept. of Computer Science, Garden City University, Bangalore-560049, India

The microprocessor DS80C400 has the capacity to seizure the MAC layer packets via a (special function register) SFR called the BCU (buffer control unit). The buffer control unit (BCU) serves as the central controller of all microprocessor DS80C400 Ethernet activity and is culpable for coordinating as well as reporting condition for all data-packet transactions between the Ethernet MAC and the 8kB dual port buffer memory [7].

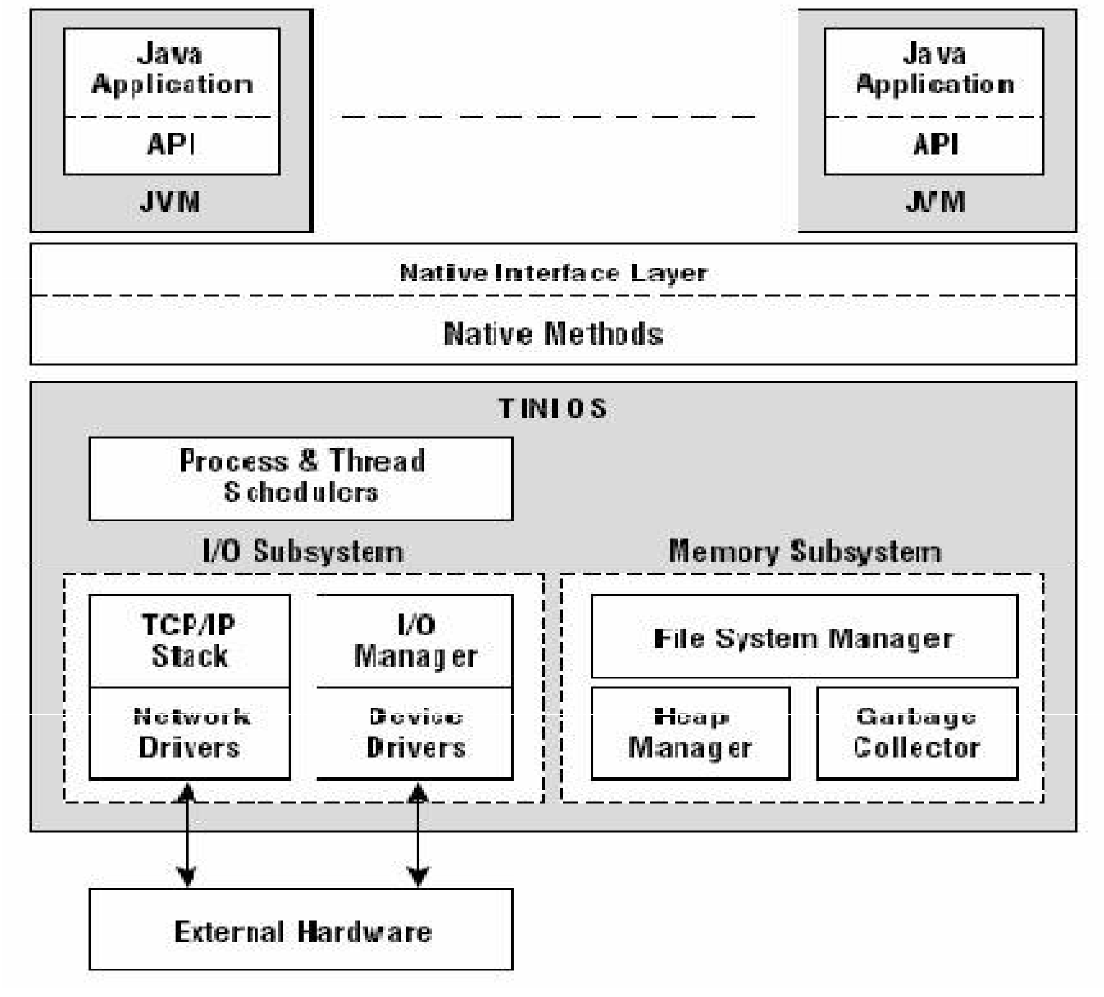

Figure 4: TINI runtime environment [6]

To obtain excellent performance an ISR (Interrupt Service Routine) for handling Ethernet activities must be utilized. The Ethernet ISR can be activated when the Buffer Control Unit reports the status of either transfer or receive packets [8]. When the ISR is triggered the microprocessor DS80C400 will be able to capture and transfer Ethernet packets. The practice of this ISR is practiced in Assembly and the skeleton for the ISR is prevised by - Dallas Semiconductors.

III. THE REMOTE NETWORK MONITORING SYSTEM

The current network protocol analyzers are considered and Remote network monitors in the possible research literature and online sources. The intention of this was to contact the functionalities of current systems to assess the feasibility of incorporating such features in the RNMS (Remote Network Monitoring System).

ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

An ISO 3297: 2007 Certified Organization Vol.5, Special Issue 2, April 2017

An International Conference on Recent Trends in IT Innovations - Tec'afe 2017

Organized by

Dept. of Computer Science, Garden City University, Bangalore-560049, India

A. Ethereal

Ethereal [9] is an open-source, free packet capture packet analyser. Years of the development as a protocol analyser have resulted in a feature-rich product with some excellent analysis options. It uses a library known as libpcap to perform the actual packet capture. Libpcap is a system independent interface for user-level packet capture and provides a portable framework for low-level network monitoring [9]. Libpcap is a free library for developers wanting to write applications for packet capture. The majority of packet captures software interfaces with this library.

B. TCPDump and WinDump

TCPDump is an open-source, command line packet capture tool [3]. It is installed on most Linux systems and is often used by system administrators for a quick overview of network activity. Though a simple application, it has many command line switches which are used to customize its functionality. It again uses libpcap to obtain packets, and though traditionally a Linux application, has been ported to run under Windows and other platforms. WinDump is the Windows version of Tcpdump, WinDump is fully compatible with TCPDump and can be used to watch, diagnose and save to disk network traffic according to various complex rules. WinDump captures using the WinPcap library to capture packets [3].

C. RMONGrabber

RMONGrabber is a proprietary closed-source applications marketed at network administrators. Information regarding system design was not available [10]. However the product specifications for this application seem to be similar to that of Ethereal with the exception that the RMONGrabber software follows the SNMP and RMON 1 standards to interact with remote probes [10].

D. Design considerations of RNMS

libpcap is the most common system for sezing packets. The conclusion was made to use WinPcap which is a windows edition of libpcap to seize packets for the mobile probe due to the fact it was open source and due to the bounty of support for the module. All of the abstracted products had some system of exporting seized data to a file. The facet of being able to save the seize packets in a common format is necessory for the RNMS as it allows the seized data to be analyzed for future use and be used in conjunction with other analysis software. Due to its popularity, libpcap was used as the format for stored seize packets. A facet missing from any of the products which had basic remote capture support was the ability to command more than one capture agent from the client software. Particularly on a larger managed networks where it would be desirable to have agents deployed at strategic points, having to manually keep a list of agent locations would be rather time consuming. One of the major design objectives for the portable TINI probe and management PC was that the developed software should support the ability to control multiple agents concurrently from a single management console. Another design consideration was to develop a graphical client with an intuitive and appealing interface, paying particular attention to Human Computer Interface (HCI) principles to ensure an enjoyable user experience. The broad design goals of the RNMS were that the software needed to interface with WinPcap library using the Java Native Interface. It also needed to establish a means of finding the packet capturing agents in the given network. This was implemented by sending a UDP Broadcast. It also needs to establish a connection between the remote packet capturing agent which is capturing the Ethernet packets and the client which is in essence is the GUI showing the retrieved packet. This was achieved using Java sockets. It also needed the development of a GUI which allows the user to connect to a particular agent, start/stop capturing packets, show the retrieve packets and save the retrieved packets in a pcap formatted file. The design of the Software architecture is provided in Figure5.

The agent module, shown in Figure 5, should ideally be executed on the TINI board but during development the agent module was executed on a PC. Once the agent is initialized, it creates a Java socket and waits for a client to connect to this socket. When the client wants to find and start retrieving the packets from the agent the client sends a UDP broadcast across the network. The agent replies to this broadcast and hence the agent is identified. Once the agent is identified the client has the option to start retrieving packets from the agent. This is initiated when the client prompts the agent to start capturing packets. The captured packets are sent from the agent to the client via the socket which was established when the agent module was executed. The client GUI is populated by the packets received from the agent in a tabular manner showing the source and destination MAC address of the received packet and the packet type. The client has the option of terminating the packet capture, and saving the retrieved packets into a file for further analysis.

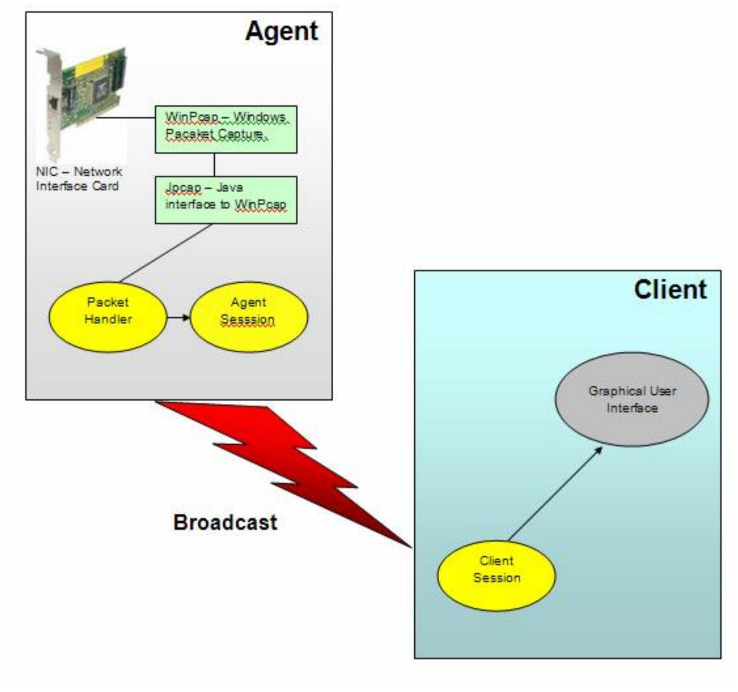
Agent
WinPcap – Windows Pacaket Capture
NIC – Network Interface Card
Jpcap – Java interface to WinPcap
Packet Handler
Agent Session
Client
Graphical User Interface
Broadcast
Client Session

Figure 5: Proposed architecture of RNMS

IV. CONCLUSION

The main goal of the portable embedded microprocessor, based on the TINI, was to capture Ethernet packets via the TINIs400 and this objective was achieved. A PC version of the system showed that in a network with low utilization that just over sixty percent of the traffic could be accumulated by the probe. This may also be a function of the NIC (network interface cards) employed and the processing speed of the general purpose computer. Future work will involve re-designing the TINI hardware (packet capture occurs on the TINI and porting the client Java software to the TINI).

REFERENCES

[1] W. Stallings, SNMP, SNMPv2, SNMPv3, and RMON 1 and 2 , 3 rd ed., Addison-Wesley, 1996.
[2] D. Chiu and R. Sudama, Network Monitoring Explained: Design and Application, Ellis Horwood, 1992.
[3] WinPcap: The Windows Packet Capture Library", visited 29/08/2006, http://www.winpcap.org/
[4] Jpcap-Java package for packet capture", 29/08/2006, http://netresearch.ics.uci.edu/kfujii/jpcap/doc/index.html
[5] Maxim/Dallas Semiconductors - DS80C400 Network Microcontroller data sheet, visited 13/06/2006, http://www.maxim-ic.com/products/tini/pdfs/TINIs400.pdf
[6] Maxim/Dallas Semiconductors", visited 10/05/06 http://www.maxim-ic.com/products/tini/pdfs/TINI_GUIDE.pdf
[7] Maxim/Dallas Semiconductors - DS80C400 Network Microcontroller data sheet, visited 13/06/2006, http://www.maxim-ic.com/products/tini/pdfs/DS80C400.pdf
[8] Maxim/Dallas Semiconductors, Appl. Note 712, pp. 1-12.
[9] "SourceForge.net", visited 18/08/2006, http://sourceforge.net/projects/libpcap/
[10]RMONGrabber,visiited07/29/2008. http://www.wildpackets.com/products/legacy_products/rmongrabber/overview,

ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

An ISO 3297: 2007 Certified Organization Vol.5, Special Issue 2, April 2017

An International Conference on Recent Trends in IT Innovations - Tec'afe 2017

Organized by

Dept. of Computer Science, Garden City University, Bangalore-560049, India

[11] Maxim/Dallas Semiconductors, Appl. Note 712, pp. 1-12. 49
[12] "The Java tutorials". Visited 24/08/2006, http://java.sun.com/docs/books/tutorial/networking/index.html
[13] Y.H Choi, K. H. Lee, J. L. Lee and S. B. Lee, "A method of gathering end-to-end management information", Network Operations and Management Symposium, IEEE , Vol. 3, pp. 849-858, Feb 1998.
[14] K. Terplan, *Communication Networks Management*, Prentice-Hall, 1992.
[15] D. Chiu and R. Sudama, Network Monitoring Explained: Design and Application, Ellis Horwood, 1992.