IJCSE International Journal of Computer Sciences and Engineering Open Access

Review Paper Volume-5, Issue-5 E-ISSN: 2347-2693

Real-time Packet Behavior Response in Socket Application

Prathap M.[1*], A.S. Thanamani[2]

[1]Research and Development Centre, Bharathiar Univeristy, Coimbatore, India

[2]Dept of Computer Science, NGM College, Pollachi, India

[*]*Corresponding Author prathapcomputer@gmail.com, Tel.: +91-9500428322*

Online Available at: www.ijcseonline.org

Received: 20/Apr/2017, Revised: 28/Apr/2017, Accepted: 21/May/2017, Published: 30/May/2017

Abstract- The administrator has to know the network bandwidth and other resources that are used for accounting and auditing. This creates an emphasis to monitor network traffic and conduct analysis to ensure smooth operations performed routinely. Time is an important factor, which contributes to the variations of packet flow. flexible and allows powerful filters during or after capture to isolate traffic by specific node, protocol, and packet content In this paper, initially, the packet movement in the network was monitored with only hypertext transfers using HTTP. The same movement was analyzed with respect to other file transfer applications. The number of file downloads between the HTTP server and clients are also varied to follow the behavior of data. We also designed and developed multithreaded socket applications using java which was again monitored and analyzed using the above methodology. We compared the reading and the experimental results shows that the developed application outperforms in terms of packet acceleration compared to the previous ones.

Keywords- packet, bandwidth, delay, throughput, hypertext, HTTP, server, client, multithread and socket

I. INTRODUCTION

Traffic shaping is used to guarantee performance, optimize , improve latency, and/or increase usable bandwidth. A lot of investigations had been made in bandwidth allocation but gaps in studies presented in the paper show that there is still strong scope for improvement [1]. Whereas Packet analyzer consists of a packet analyzing PC program and sniffer device . The sniffer device of the packet analyzer monitors captures MAC packet frames, and transmits packet frames to the packet analyzing computer [2]. Packet analysis could be done by capturing and analyzing traffic passing by the machine where the tool is installed with results displayed. It also decodes all major and frequently used protocols including TCP/IP, UDP, HTTP, etc. In the research work conducted to compare TCP and UDP packet analysis, they concluded that frame length, frame no. and bytes captured during sending a mail more of TCP than UDP [3].

Traffic flow models are based on observations. Traffic flow prediction is an important task for traffic managers and it allows performance assessment of major traffic infrastructure [4].It comprises user friendly interfaces to display technical information. It is possible to filter the network traffic to focus on the specified needed information. It is flexible and allows powerful filters during or after capture to isolate traffic by specific node, protocol, and packet content. Hence the tool was initialized as per the settings and HTTP server response time was configured as shown in fig 1.

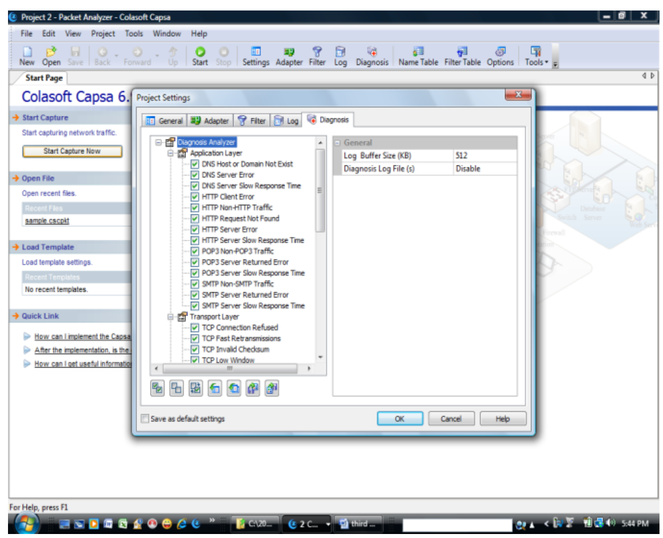

Fig 1: Initial seeting showing buffer size of 512 KB.

Network performance and security prevent network problems, conduct effective troubleshooting and take actions quickly to solve possible problems. The network bandwidth and other resources are used for accounting, auditing or for network planning purposes and analysis of the packets passing through the network. Network Packet Analyzer CAPSA is an advanced network traffic monitoring, analysis and reporting tool, based on Windows operating systems. It captures and analyzes all traffic transport over both Ethernet and WLAN networks and decodes all major TCP/IP and application protocols. Its advanced application analysis modules allows to view and log key communication applications such as emails, http traffic, instant messages and DNS queries. The comprehensive reports and graphic views enable to understand network performance and bandwidth usage quickly, to check network health. Initial HTTP readings were noted by varying the requests as shown in fig 4. Hypertext transfer protocol was designed to be quick, simple, and non instructive. A distinguishing feature of the methods is that they quickly made real-life impact, addressing problems that need to be solved by interactive systems at a large scale[5]. To increase the efficiency and precision of large-scale road network traffic flow prediction, a GA-SVM model wais proposed [6]. The connection between a server and a client program is temporary and must be reestablished for every data transfer. A URL is always a single, unbroken line of text with no spaces. Web browser generally displays the URL of the Web page currently being viewed near the top of the windows.

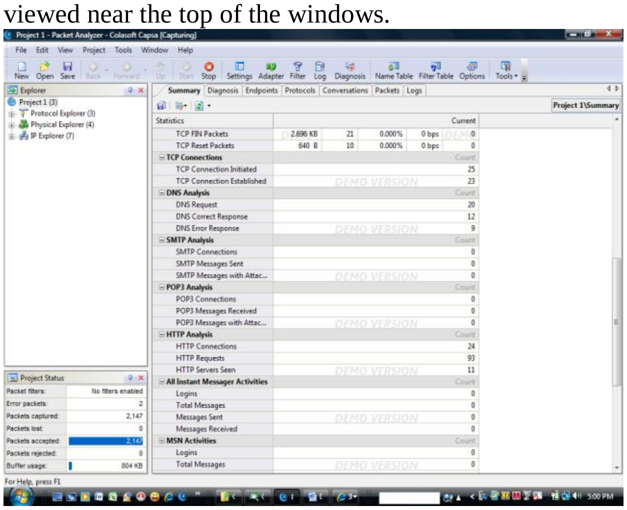

Fig 2: Initial analysis of HTTP at the start

The packets used were noted and the readings for corresponding protocols are shown in fig 3 and the initial HTTP requests were shown in fig 4.

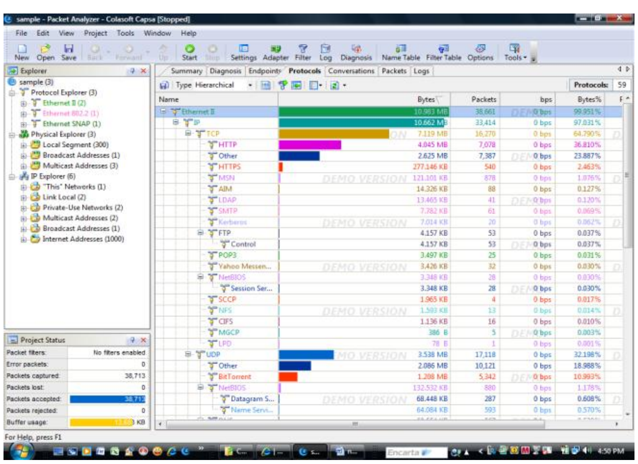

Fig 3 packets used by protocols

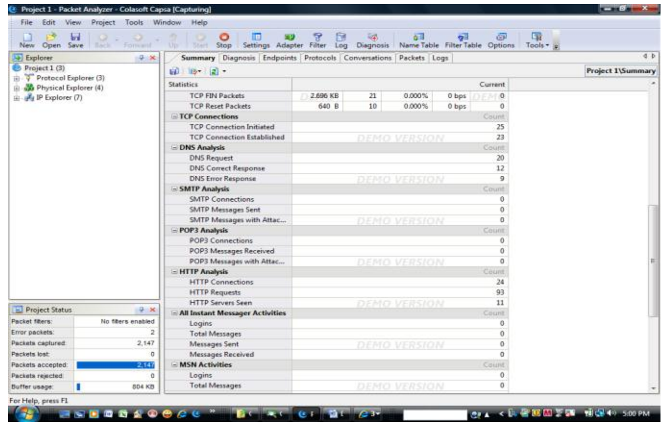

Fig 4 Buffer usage after the HTTP requests made without download

The two sites of same URL were requested and the same file of same size was downloaded. Now in order to invoke downloads of same size file single HTTP were noted and two HTTP requests were done for the same URL. The readings were noted for the same file of same size but as different requests as shown in fig 5. From the figure we can view that the buffer usage is almost nearing its capacity indicated in red. This is due to the fact that when more hypertext transfers are made the allocated buffer usage increases.

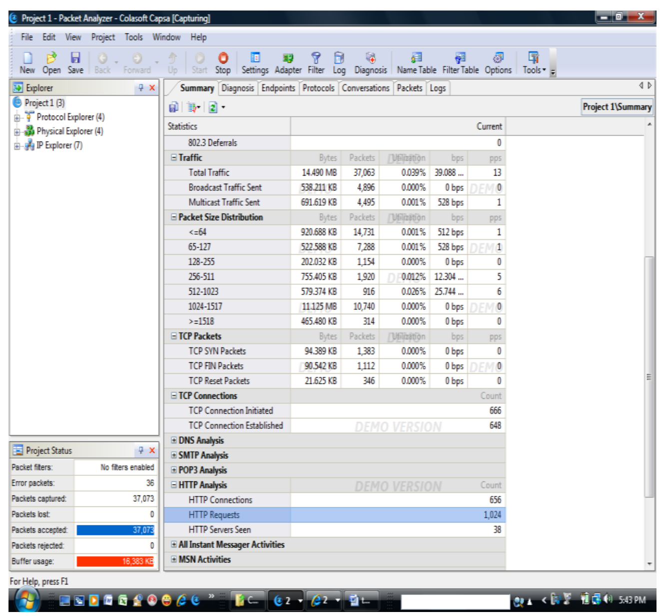

Fig 5 Buffer usage of HTTP with two downloads

II. RESULTS OF SOCKET APPLICATION PERFORMANCE

A socket is a software endpoint that establishes bidirectional communication between a server program and one or more client programs. The socket associates the server program with a specific hardware port on the machine where it runs so any client program anywhere in the network with a socket associated with that same port can communicate with the server program. Typical system configuration places the server on one machine, with the clients on other machines. The clients connect to the server, exchange information, and then disconnect. Moreover, the processes that use a socket can reside on the same system or on different systems on different networks. Initially, client and server create sockets. The server binds to an address and port and it is automatically done in Java. The client knows server's address and port whereas the server listens on that port. The client connects to the address and port server accepts the connection. Hence the client and server read from and write to their sockets.

Client Side Program:

```
import java.io.*;
import java.net.*;

public class myClient {
   public static void main(String[] args) throws IOException {
        Socket echoSocket = null;
        PrintWriter out = null;
        BufferedReader in = null;
        try { // connect to local host at port 4444
echoSocket = new Socket ("10.0.0.1", 4444);
                      ……………………..
                       ……………………..
      }
}
```

The Server side program:

```
import java.net.*;
import java.io.*;
public class myServer {
   public static void main(String[] args) throws IOException {
        ServerSocket serverSocket = null;
        try {
            serverSocket = new ServerSocket (4444); // binds, listens automatically
                      ……………………..
                       ……………………..
       }

}
```

The application of the client side and server side execution

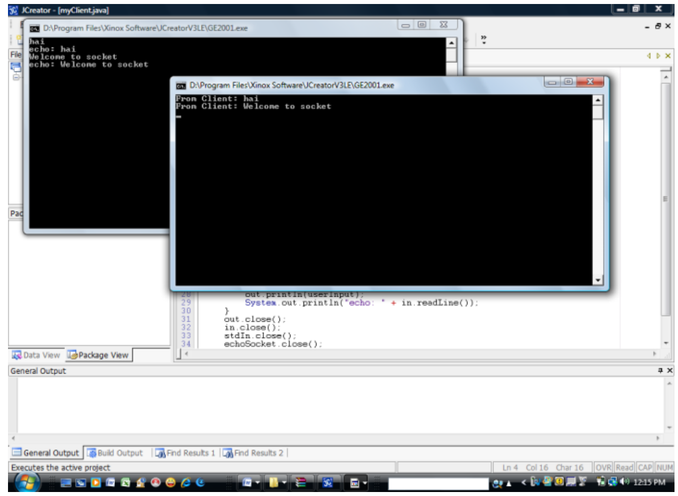

Fig 6 Interaction of server and client programs

Table 1. Performance measurement of socket application

Time	Received	Sent	R.Diff	S.Diff	S-R
4:03:00	2,534,091	474,244	178	172	-6
4:04:00	2,534,269	474,429	108	113	5
4:05:00	2,534,377	474,543	136	144	8
4:06:00	2,534,513	474,687	-	-	-
		Total	422	429	

III. CONCLUSION

The network performance factors delay and throughput were discussed and measured. Real-time packet capture and analysis had been done using packet analysis technique. Also the network comprises Netest which helps in monitoring the bad performing hardware and provides way for better performance. The observation include the load of a large user population among multiple servers were not spread. Reduce network usage and improve server performance when there is a significant user population in distributed locations. Though there were five servers functioning but the configurations are inefficient. The better solution could be the application of same time server which the network lacks currently. The tool had been tested under various HTTP requests and the increased requests are shown. Moreover, the analysis was done with variable number of downloads. The socket application was designed and implemented. The server side and client side communicated effectively and the measurements as well as the performance of this application was also measured repeatedly and shown in table 1. From the above table 8 the average of sent and received calculated are 2.34 packets per second for receiving and average of sent 2.38 packets per second. Finally, the behavior of the system was compared against the previous reports. Average of received is 2.34 packets per second the average of sent is 2.38 packets per second compared to 2.19 and 2.20 respectively. The results obtained shows the application sent and received bytes for the application increases compared to that of the previous throughput.

REFERENCES

[1] Akash Deepa, K.S. Kahlonb, Harish Kumara, "*Survey of Scheduling Algorithms in IEEE 802.16 PMP Networks*", Egyptian Informatics Journal Vol.15, Issue.1, pp.25-36, 2014.

[2] Sol Lim, KJ. Lee, SY. Kim, CS. Chae, "*Intae Hwang and Dae Jin Kim* Implementation of IR-UWB MAC Development Tools Based on IEEE 802.15.4a*", International Journal of Control and Automation Vol. 8, No. 4 , pp.275-286, 2015.

[3] Mahesh Kumar, Rakhi Yadav, "*TCP and UDP Packets Analysis Using Wire shark",* International Journal of Science, Engineering and Technology Research (IJSETR), Vol.4, Issue.7, 2015.

[4] Niu Xiaoguang, Zhu Ying, Cao Qingqing, Zhang Xining, Xie Wei, Zheng Kun, "*An online-traffic prediction based route finding mechanism for smart city*", International Journal of Distributed Sensor Networks, Vol.11, Issue.8, pp.1-12, 2015.

[5] Bast Hannah, Delling Daniel, Goldberg Andrew, Muller Hannemann, Matthias Pajor, Thomas Sanders, Peter Wagner, Doro Thea, Werneck Renato, "*Route planning in transportation networks*", Technical Report MSRTR-2014, US, pp-1-68, 2014.

[6] Yang Zhaosheng, Mei Duo, Yang Qingfang, Zhou Huxing, Li Xiaowen, "*Traffic flow prediction model for large-scale road network based on cloud computing*", Mathematical Problems in Engineering, Vol.14, Issue.7, pp;23-29, 2014.

AUTHORS PROFILE

Prathap M has teaching experience of about 17 years. His research interests networks, operating systems. He has published about many books and 19 papers in national and international journals.

Dr. Antony Selvadoss Davamani is working as a Reader in NGM college with a teaching experience of about 28 years. His research interests includes knowledge management, web mining, networks, mobile computing, telecommunication. He has published many books and papers in national and international journals.

