International Journal of Future Generation Communication and Networking
Vol. 2, No. 1, March, 2009

Apportioning Bandwidth to Clients as per Privileges

Mohammed A. Qadeer[1], Shahid Habib[2], Ahmad Yazdan Javaid[3]
[1]*Dept. of Computer Engineering, Aligarh Muslim University, Aligarh- 202002, India*
[2]*Engineer-PV, Tejas Networks Limited, Bangalore-560078, India,*
[3]*Assoc.App. Developer, CSC India Pvt. Ltd.,NOIDA- 201301, India*
[1]*maqadeer@zhcet.ac.in,* [2]*shahid@tejasnetworks.com,* [3]*ajavaid@csc.com*

Abstract

In this paper we present a technique to allocate bandwidth to the clients as per the privileges. It means basically that each user can buy bandwidth which will be available as a premium, high and normal class bandwidth. As per the class a user buys he will be given a username and password. Using this username and password he will authenticate himself to the server. The server then provides bandwidth to that user for running the applications which fall within the bandwidth range of that class. The server queues all the packets coming to it from the internet connection and then sends them out on a priority basis which is defined by the class each user falls into.

Keywords: *Bandwidth, Clients as per Privileges, Authentication*

1. Introduction

In the fields of networking and communications, there is a continual concern over poor bandwidth utilization. It is sometimes the case that host computers can employ applications or processes that are very bandwidth intensive, limit the amount of bandwidth that other hosts have at their disposal [1], [2], [3]. Sometimes, the case is that of host employing applications or processes having high bandwidth requirement, and thus, limit the bandwidth available to other hosts, which in turn causes limitation of working capacity (task completions per unit time) of the host computer as well as other hosts. This in turn limits the amount of work that can be done in a given amount of time. In order to alleviate this problem, we believe that we can develop a system wherein the amount of bandwidth that each application or process uses on a network can be measured in real-time.

In order to solve such kind of problems, we have developed a system which can allocate, and/or guarantee, the bandwidth to any user and measure the bandwidth it has been allocated/consuming. The allocation/guarantee can be based on the service discipline designed by the network administrator.

1.1. System Features

We have developed a product that will help control and measure the distribution of bandwidth in a network. There are several key goals of developing the product (apart from bandwidth control and management) and features that the product needed to have, in order to fulfill all the requirements of a bandwidth controlling system of its type. The system can also be modified to be used as a module/service of the operating system; preferably, a UNIX/Linux based central system. Some key design goals are stated here.

1.1.1. Username and Password based Authentication: The bandwidth provided to each user should be based on the username and password.

1.1.2. Low Bandwidth Consumption by the Server: This constraint was also recognized during the designing of the product. Since this product is being created for the reason of utilizing and conserving bandwidth, the system itself must not inhibit the data communications by using the connection bandwidth for itself. Since the design calls for the host machines to continually connect to the database, bandwidth will be used during this transfer. This issue was also taken care of during the module designing and needs not to be monitored later as inner network bandwidth is quite high than the connection bandwidth, and the system is going to use the lightly used inner network bandwidth.

1.1.3. Compatibility to non-UNIX systems: The user interface of the system is windows-based. Our product also needs to be compatible with software that is already widely being used in existing servers and servers currently under operation in the market, so as to reach as much market as possible. Specifically, our software has to meet and be compatible with non-UNIX standards to be able to reach a higher volume of customers. Furthermore, an interactive and windows-based user interface will be simpler to use and provide training for operating the system (if necessary).

1.1.4. System Throughput: The total data transferred through the central system i.e., total bytes per second of data sent and received through the system will be monitored and controlled by the system. To make it simpler for the network administrator, a feature can added in the future using which one will be able to create rules and filters that could regulate how certain programs transmit or receive data in a much more efficient way.

1.1.5. Non-Hardware Control: Using a software control in the system is the most important objective of our system because of the most important reason behind designing the system i.e., to manage bandwidth consumptions of users/programs based on their importance in a work environment, e.g., in an advertising company, where high-end applications on workstations use bandwidth to download, share and transfer data among themselves, there is a great need to have a certain amount of bandwidth available so as to have some bandwidth available for fulfillment of requests from users outside the network. In recent times, it has been found that the use of applications like Kazaa, Grokster, WinMX, etc, is growing among employees of organizations. These applications may seem good to employees as individuals, but they sure take valuable resources away from the company, which is investing its money in a fast internet connection.

1.1.6. Centralized operation Centre: Obviously, the system needs to have a centralized control and thus, a centralized place of operations. This means that it will run on a server of the network. Since, servers provide a high performance environment with the added protection against potential security breaches and network attacks; it is more beneficial for the system to be in such a secure and controlled environment to utilize system resources optimally.

1.2. The Approach

Today in the modern communication world, the traffic that exists in the Internet is becoming more and more abnormal. This was mainly due to increase in number of users day by day which results in bandwidth congestion, poor response time for end user's etc. The most efficient solution to this problem is to manage and allocate the existing bandwidth proportionally using suitable queuing disciplines. It is a full featured technology which may reduce the cost and improve network performance. If there was a way to control bandwidth that a service already has, then it can create a smarter network for all users [4]. This is the main objective of our project. We explored ways to keep a real-time log of which applications and hosts are using the most bandwidth and divert or restrain the bandwidth in order to make the network more efficient. This will be done by creating an application to sit on server. This application will serve three primary purposes. It will be a firewall, bandwidth allocator, and active application monitor. The data collected will be the basis in creating a smarter self-sustaining network. An improved condition of this project is that it will be fully self-sustaining. The database on the server will keep track of the running applications and prioritize those that are running. There will be no need for a system administrator because the database will decide based on priority to close or limit bandwidth. Bandwidth can be either allocated on committed bandwidth allocation basis to the users or on burstable bandwidth allocation basis where extra bandwidth is allocated to the current users. However we will deal with the first one

2. Design

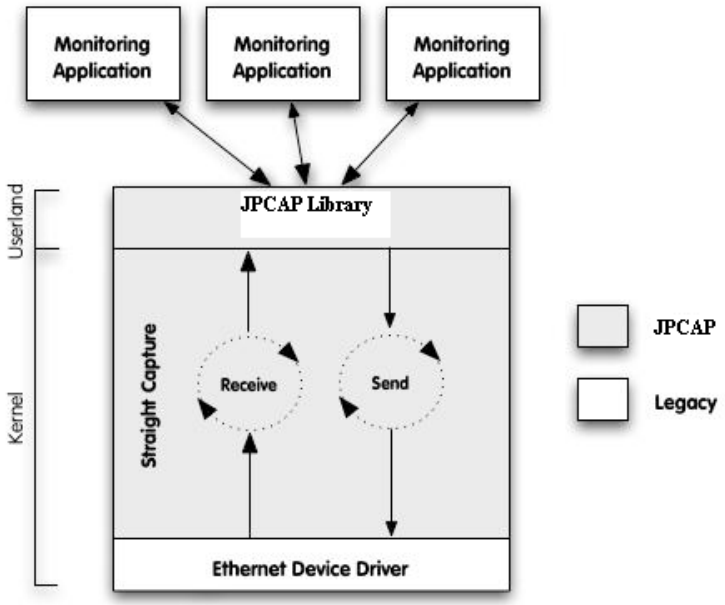

Figure 1. Packet Capture using JPCAP

2.1. Core of the Design

2.1.1. Fair Queuing: Fair queuing is a technique that allows each flow passing through a network device to have a fair share of network resources [5], [6]. Users or processes having paid for higher bandwidth will be guaranteed that amount of bandwidth at all times.

2.1.2. Username and Password based Authentication: The bandwidth provided to each user will be based on the username and password. Each user who pays for bandwidth will have a unique username and password. Whenever a user becomes active it sends a request message to the server to grant it the requisite bandwidth. The server first matches the username and password from the stored database. If they match it sends an o.k. message to the client else sends a connection denied message is send. Now the server maps the username and password to the bandwidth allocated to that username and password. It then reads the header of the request packet received to determine MAC address of the client. It then makes an entry in its database of the MAC address of the packet in the tuple containing that username and password and the bandwidth. All the incoming packets will be queued on the basis of MAC address.

2.2. System Design

The implementation idea was to simply restrict bandwidth based on application priority because of the reason of the requirement of such bandwidth limiting in any industrial, technical or institutional setup.

Basic implementation is a five phase process. Starting from authenticating the user to finally controlling every packet passing through the server, all phases require a powerful and stable server, being the reason behind selecting the platform for our application as a UNIX platform.

The phases are described below.

2.2.1. Authentication: The bandwidth provided to each user will be based on the username and password. Each user can buy bandwidth of premium, gold or silver class. The class premium has the maximum bandwidth available and it would have bandwidth to support all sorts of application like IPTV, Internet Telephony as well as basic applications like mails etc while gold and silver classes will have lesser bandwidth available. Silver class has the lowest bandwidth and one can only run basic applications like mail, ftp etc. Each user who pays for bandwidth will have a unique username and password as per the class. Whenever a user becomes active it sends a request message to the server to grant it the requisite bandwidth. The server first matches the username and password from the stored database. If they match it sends an o.k. message to the client else sends a connection denied message.. Now the server maps the username and password to the bandwidth allocated to that username and password. It then reads the header of the request packet received to determine MAC address of the client. It then makes an entry in its database of the MAC address of the packet in the tuple containing that username and password and the bandwidth. All the incoming packets will be queued on the basis of MAC address. If the user shifts to some other PC this PC will again send a connection request. The previous connection will be closed and the database entry for this user will be updated with the current MAC address.

2.2.2. Packet Capturing: When the user gets the authentication, it gets access to the internet through the server and can request whatever it wants. Now, to limit bandwidth based on priorities, when the requests of each user is granted and a reply is received, the incoming packets needs to be captured and stored somewhere so as to provide a limited rate to the user.

Packets can be captured either using hardware or a software. Software tools are often preferred often because of their low cost and high versatility [7], [8]. We will be using the libraries like jpcap and winpcap to capture packets.

2.2.2.1. How are packets captured by the NIC?: Modern NICs have a small memory required to enable the receiving and sending packets at the full link speed, independently of the host capabilities .moreover, NICs perform some preliminary checks such as CRC errors, short Ethernet frames, while packets are stored in the on board memory so that invalid frames can be discarded immediately.

After a valid packet has been received by the NIC, this generates a request toward the bus controller for a bus mastering data transfer. At this point the NIC takes control of the bus, transfers the packet to the NIC buffer in the host's memory, releases the bus and generates a hardware interrupt towards the Advanced Programmable Interrupt Controller (APIC) chip. This chip wakes up the OS interrupt handling routine, which triggers the Interrupt Service Routine of the NIC device driver.

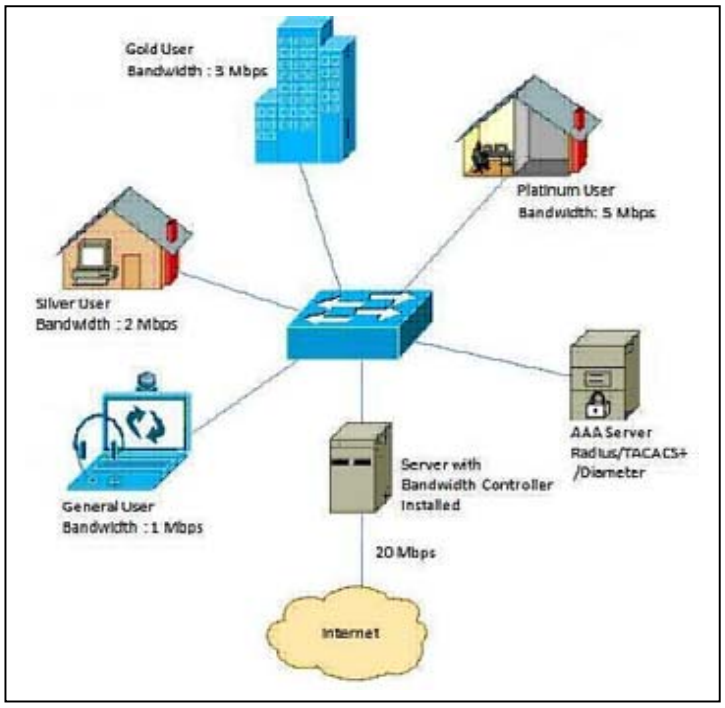

Figure 2. Authentication based QoS using Bandwidth Manager

The ISR of a well written device driver has little to do. Basically it checks if the interrupt to itself and acknowledges it. Then the ISR schedules a lower priority function (called the Deferred Procedure Call or DPC) that will later process the hardware request and notify the upper layers that a packet has been received. The CPU will process the DPC routine when no interrupt requests are pending. Interrupt coming from the NIC are disabled when a NIC

device driver is performing its work because a processing of a packet has to be completed before the next one is serviced. Moreover since interrupt generation is a costly operation, modern NICs allow more than one packet to be transferred in the context of a single interrupt so that an upper layer driver is able to process several packets each time it is activated.

2.2.2.2. What is JPCAP: Traditionally, applications running in user space have been able to view and operate on packets only when the packets originate in user space or once the packets pass up through the protocol stack into the application layer. Recently, however, computer programmers and network administrators have seen a need to be able to monitor packet traffic through the network and analyze in user-space packets that are being processed by the kernel stack code. For this purpose, many operating systems now provide the ability for user-level processes to “sniff” or “capture” network traffic, by employing “packet taps” in kernel space.

Jpcap is an open source library for capturing and sending network packets from Java applications. It provides facilities to:

- Capture raw packets live from the wire
- Save captured packets to an offline file, and read captured packets from an offline file.
- Automatically identify packet types and generate corresponding Java objects (for Ethernet, IPv4, IPv6, ARP/RARP, TCP, UDP, and ICMPv4 packets).
- Filter the packets according to user-specified rules before dispatching them to the application.
- send raw packets to the network

2.2.2.3. Problem using JPCAP/WINPCAP: These packet taps do not provide means for altering the packets in user space or for changing the flow of packets Jpcap captures and sends packets *independently* from the host protocols (e.g., TCP/IP). This means that Jpcap does not (cannot) block, filter or manipulate the traffic generated by other programs on the same machine: it simply "sniffs" the packets that transit on the wire. Therefore, it does not provide the appropriate support for applications like traffic shapers, QoS schedulers and personal firewalls. Packet capture components are usually transparent to other software modules like protocol stacks, thus not influencing the system’s behavior .They just insert a hook in the system so that they can be notified usually through a callback function called tap()-as soon as a new packet arrives from a network. Packet capture components are usually implemented as network protocols drivers in Win32.A packet not destined to the host will be captured by JPCAP if it is running in the promiscuous mode (this mode captures all packets whether destined to the host or not).The NIC upon recognizing that the packet does not belong to it will not send it to the protocol stack but will pass it onto the Ethernet. Our software will queue the packet and send it after a requisite delay. So the destination will receive two packets: One send by our software and one by the NIC. This will result in redundancy as shown in Fig 1 in which two copies are created for every packet.

2.2.2.4. The Solution: One solution may be implemented with a lightweight modification to kernel code, and an associated application programming interface (API). Provided with the collective ability to divert packets from the kernel stack to user space and inject packets back into the kernel stack from user space, a program running in user space may then examine and manipulate packets on their way through the kernel stack. The system facilitates creation of a

special socket for passing packets between kernel space and user space. The system in turn facilitates creation and application of a packet filter associated with the socket, in order to trap incoming or outgoing packets being processed in the kernel at a designated point in a protocol stack. Once a packet is trapped, it is moved through the socket into user space, thereby at least temporarily preventing the protocol stack from further processing the packet. In user space, an application may operate on the packet, for instance, modifying aspects of the packet or deleting the packet altogether. The system in turn facilitates injection of a packet from user space into kernel space, and into a designated point in the protocol stack for desired stack processing.

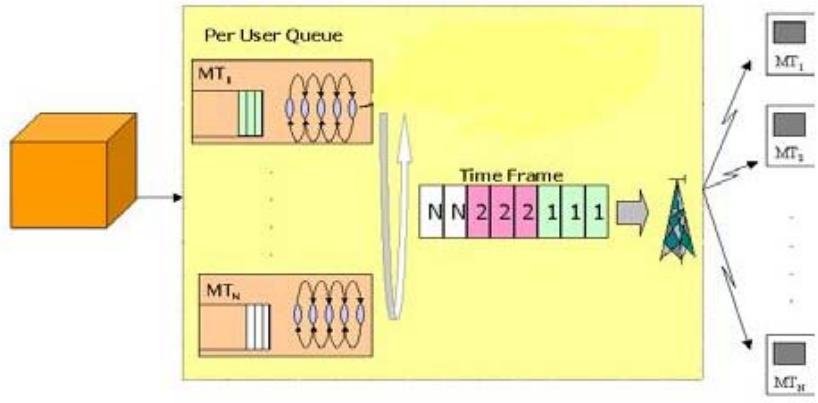

Figure 3. Queuing and transmission as per privileges

Another solution is not to pass copy of the packet to the user but instead pass its address. Since the packet is initially stored in the NIC buffer passing that address will ensure that no copy is created.

Another alternative solution may be modifying the Deferred Procedure Call (DPC) function so that it only notifies the packet capture drivers and not the protocol layer drivers. So packet not destined to the host will only be captured by JPCAP and processed by our software and hence will not result in redundancy. However, we will be using the first solution.

2.2.3. Packet Queuing: When the system sees the packet, it can do one of three things:

1. **Discard the packet:** This allows the system to provide a very robust and granular packet filtering mechanism.

2. **Forward the packet at real time:** This means that the packet bypasses the entire bandwidth management system and is immediately forwarded by the device. The end- result is effectively the same as if bandwidth management was not enabled at all. This will be done if some user is having such privileges.

3. **Prioritize the packet:** This allows the mechanism to provide actual bandwidth management services (applicable to rest of the users).

Packets captured in previous step are now maintained in separate dynamic queues based on their destination address so as to transmit to their respective destination.

2.2.4. Inserting delay: All of the services are then provided the required rate (based on their priority) by queuing the packets and making them wait for their turn. A queue after sending packets will have to wait for its turn again since the packets are sent in a round robin manner. One problem in sending the packets is that the traffic arrives in bursts. A higher priority queue that has to send more no. of packets may not have those much of packets in its queue while the low priority queue may have a higher no. of packets. The solution to this problem is to monitor the flow of incoming packets and to take it as criteria in deciding how much packets each queue will send to guarantee each service its allocated bandwidth. However for practical purpose if the higher priority packet's queue is empty then the lower priority packet's queue can be serviced to avoid resource wastage.

2.2.5. Packet transmitting*:* Finally, after determining the required delays for different users, the packets received are now sent to the respective destinations by the data rate determined by inserting delays between transmissions of each packet. The packets are send by first finding out the Network Interface address and then sending the packets to that NIC. The libraries jpcap and winpcap are used for sending the packets.

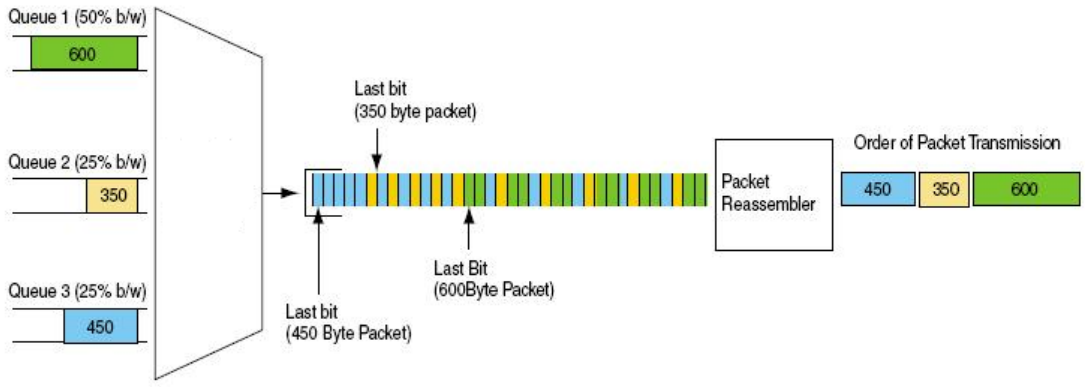

Figure 4. Detailed diagram showing how weighted fair queuing actually happens.

The packets are first put into their respective queues. Then the queue which is having higher priority will be serviced for a greater duration than the lower priority queues. In other words higher priority queues can send more no of packets than the lower priority queues.

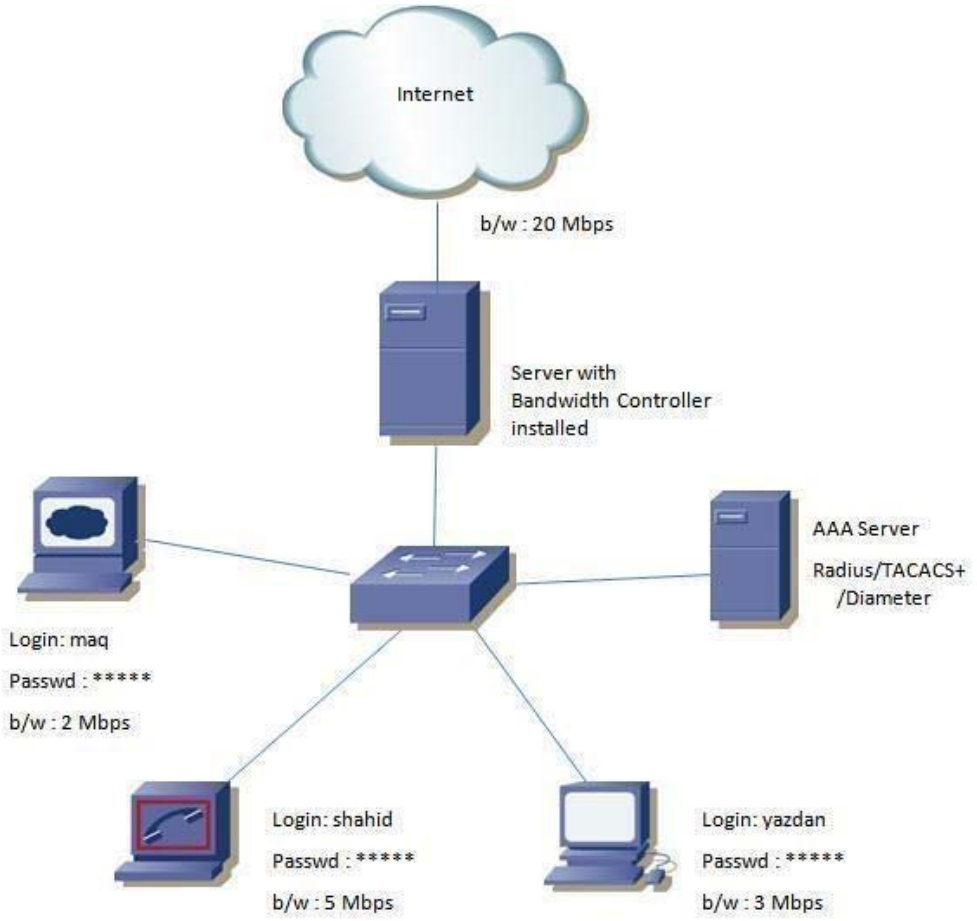

Figure 5. Illustration of Authentication Based Bandwidth Limiter.

How the no. of packets transmitted by each queue is calculated

In case of allocation of bandwidth per host on basis of MAC addresses
There is a table in database:

MAC Address	b/w allocated	No of packets to be transmitted	Username	Password
00:00:00:00:00:01	2 Mbps		x	******
00:00:00:00:00:02	5 Mbps		y	***
00:00:00:00:00:03	3 Mbps		z	**********

Let the maximum no of packets which can be transmitted out=x/second
Let b/w of each mac address=b(i)
Then p(i)=no of packets of host i=b(i)/∑b(i){i=1 to no of hosts}*x

So p(i) packets of host i,p(j) packets of host j and p(k) packets of host k will be transmitted.

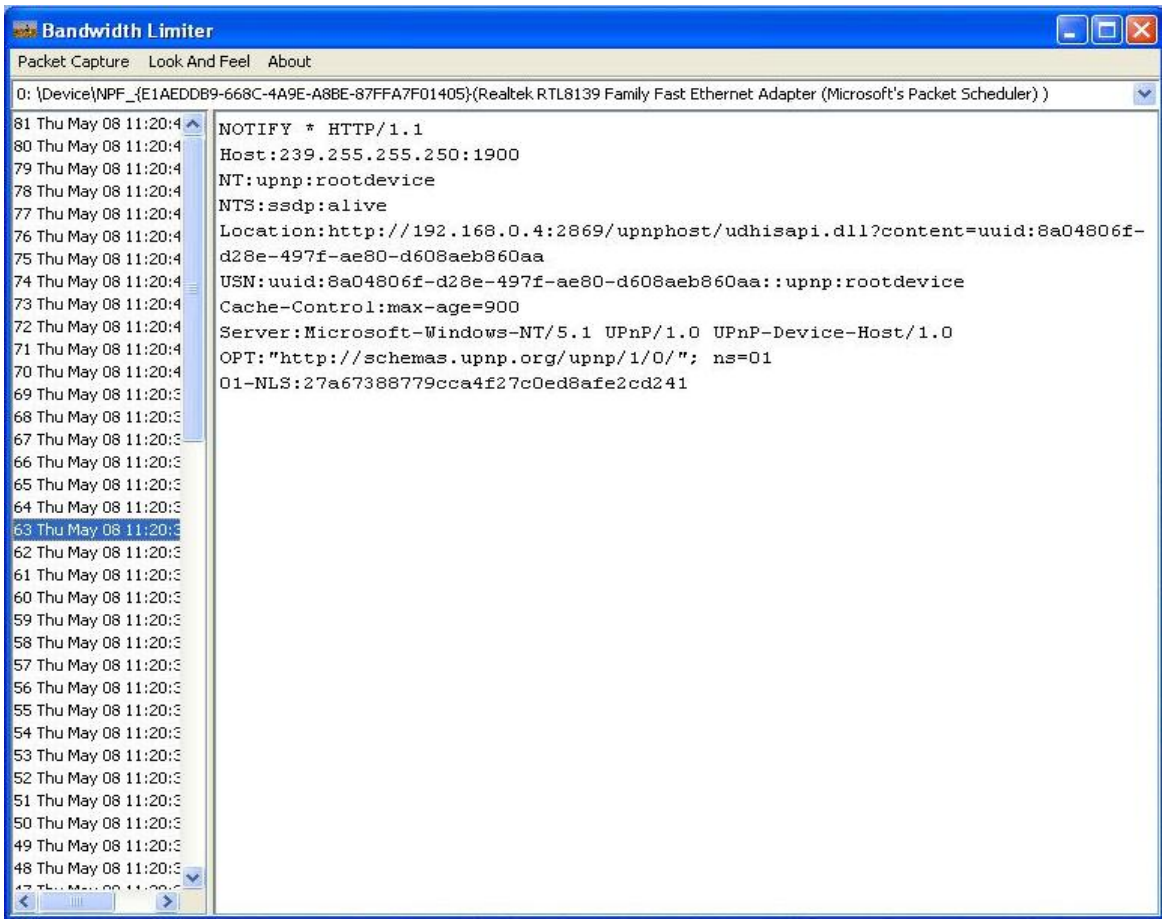

Figure 6. Snapshot of UI of our software

Right side of the frame shows the contents of the packet.

3. Results

The System having a 112Kbps connection gives following results as shown in Table 1.

Pure Bandwidth – Actual available bandwidth to the server.

Bandwidth Actually available – Total bandwidth available (i.e., which can be used) due to some delay in server due to its processing.

4. Conclusion

Primarily, the module can be used to provide following functionalities:

- traffic loggers
- traffic generators
- user-level bridges and routers
- network intrusion detection systems (NIDS)
- network scanners
- Security tools

Table 1

Pure bandwidth (in kbps)	Bandwidth Actually available (in kbps)	System 1 (2/3 of BW) (in kbps) Gold	System 2 (1/3 of BW) (in kbps) Silver	In case of bandwidth reduction, priority to
31	**26**	**18**	**8**	**System 2**
36	**32**	**22**	**10**	**System 2**
39	**36**	**24**	**12**	**System 1**
45	**42**	**28**	**14**	**System 1**
35	**29**	**20**	**9**	**System 2**

Some more specific practical applications of this module are:

- Ensure that critical applications are not impacted by non-priority traffic
- Deliver optimal application performance by allocating more bandwidth for higher priority applications
- Provide flexible bandwidth limits and traffic queuing
- Control rate classes based on any traffic variable
- Enable application bandwidth to be shared across similar priority applications for better resource sharing
- Ensure that specific types of application traffic stay within authorized boundaries

The system's performance can be enhanced by adding a redundant server. It will make the system robust and less vulnerable to failures because the redundant server can always come into effect if the primary server fails. In this a switch(just before the router connected to the internet in Fig 5) monitors both servers. It switches the incoming packets to the primary server. If the primary server fails the switch starts sending and receiving packets from the secondary server.

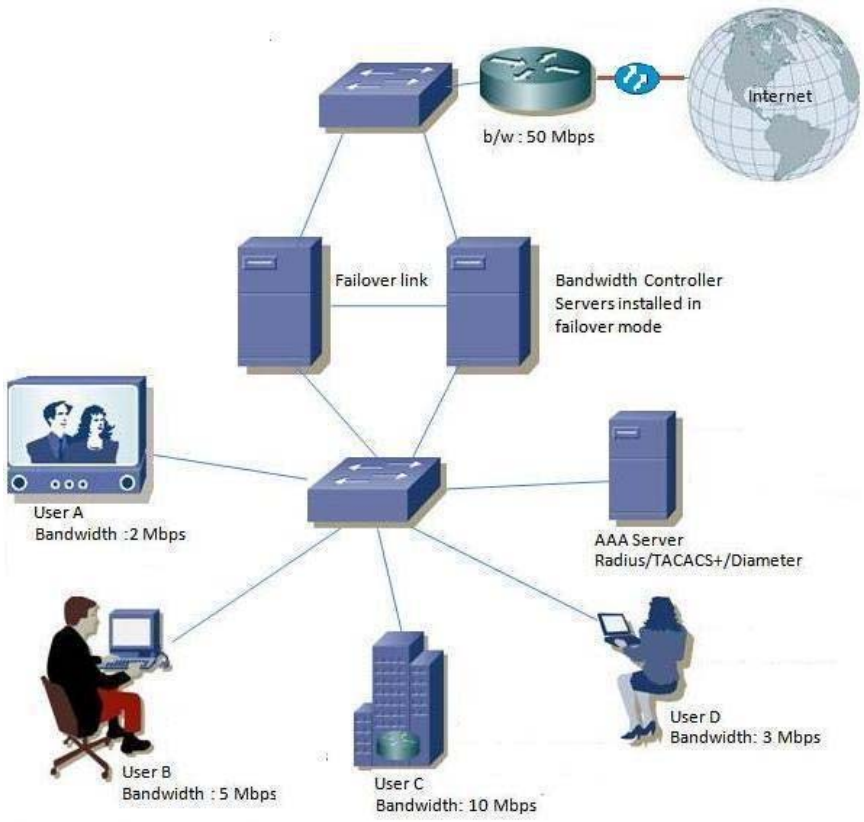

Figure 7. Redundant Authentication based Bandwidth Limiter with Failover mode

Secondly, the performance of our system is quite good in providing the Quality of service required by today's data operators and organizations. The system will be able to provide actual bandwidth allocated to the particular service. Obviously, in any organization, the resources are allocated on the basis of priorities, so same will be done by our system. Since time is spent in capturing the packets, queuing them and then sending them there is a minor delay in the users receiving the packets. Also the OS has to run many other processes and each process receives a quantum of time to execute. However taking into the account the fewer resources our product requires and the freedom of running the application along with other applications and the near to the allocated bandwidth each user receives, it is obviously better than contemporary Bandwidth Limiters.

A short overview of what the implementation of our project might look like can be seen in the **Figure 7.**

5. References

[1]. D.P. Bertsekas, R. Gallager, Data Networks, Prentice-Hall, Englewood Cliffs, NJ, 1992.

[2]. H. Zhang, D. Ferrari, Rate-controlled service disciplines J. High Speed Networks 3 (4) (1994) 389–412 connection admission control for bandwidth management of an Internet access link, Communications Magazine, IEEE , Volume: 38 Issue: 5 , May 2000 Page(s): 160 -167

[3]. A.S. Tanenbaum, Computer Networks, third ed., Prentice-Hall, Inc., Englewood Cliffs, NJ, 1996, pp. 380–381.

[4]. Bolliger R.., Gross T.R., Bandwidth monitoring for network-aware applications, High Performance Distributed Computing, 2001. Proceedings. 10th IEEE International Symposium on , 7-9 Aug. 2001 Page(s): 241 - 251

[5]. Douglas E. Comer. [1995] Internetworking with TCP/IP Vol.1: Principles, Protocols, and Architecture. 4th Edition Upper Saddle River: Prentice Hall.

[6]. Bo Chen, Yaping Zhou, Hongsheng Xi, Bandwidth Allocation Based on Consumers' Demand Information, Proceedings of the 6th World Congress on Intelligent Control and Automation, June 21 - 23, 2006, Dalian, China

[7]. T.-W. Angus Lee, S.-H. Gary Chan, Qian Zhang, Wen-Wu Zhu, and Ya-Qin Zhang, Allocation of Layer Bandwidths and FECs for Video Multicast Over Wired and Wireless Networks, IEEE Transactions On Circuits And Systems For Video Technology, Vol. 12, No. 12, December 2002

[8]. Yi-Hsien Tseng, Eric Hsiao-Kuang Wu, and Gen-Huey Chen, Scene-Change Aware Dynamic Bandwidth Allocation for Real-Time VBR Video Transmission Over IEEE 802.15.3 Wireless Home Networks, IEEE Transactions On Multimedia, Vol. 9, No. 3, April 2007

[9]. I-Shyan Hwang, Bor-Jiunn Hwang, Ling-Feng Ku, Pen-Ming Chang, Adaptive Bandwidth Management and Reservation Scheme in Heterogeneous Wireless Networks, 2008 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing

[10]. Saswati Sarkar, and Leandros Tassiulas, Fair Bandwidth Allocation for Multicasting in Networks with Discrete Feasible Set, IEEE Transactions On Computers, Vol. 53, No. 7, July 2004

[11]. Wei-chih Hong, Zsehong Tsai, Adaptive Bandwidth Allocation Via Dynamic Programming In A Shared Wireless Network, The 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'07)

[12]. Heng-Qing Ye, Stability of Data Networks Under an Optimization-Based Bandwidth Allocation, IEEE Transactions On Automatic Control, Vol. 48, No. 7, July 2003

[13]. Dr Orhan Gemikonakli (2003) "Network Management" http://www.cs.mdx.ac.uk/staffpages/orhan/csy4061/netman1.htm

[14]. Tommy K Paul (1994) "Building Network Bandwidth" Network News - The Network Professional Association monthly publication, http://www.sju.edu/%7Ejhodgson/netw/tpasg4.html

[15]. Bandwidthcontroller.com "Internet sharing guide", http://bandwidthcontroller.com/internet-sharing.html

[16]. The Internet Services Company (2002) "Active Bandwidth Management Device" Networking, http://www.interchannel.net/product/networking/bandwidth.htm

[17]. Juniper.net (1999) "Managing Bandwidth from a Large Traffic Source", http://www.juniper.net/solutions/literature/solutionbriefs/351000.pdf

[18]. SearchNetworking.com (2001) "Data Transfer Rate"

http://searchnetworking.techtarget.com/sDefinition/0,,sid7_gci213492,00.html

[19]. Cristobal Baray and Kyle Wagner (1999) "Where Do Intelligent Agents Come From?" ACM Crossroads Student Magazine, http://www.acm.org/crossroads/xrds5-4/dumbagents.html

Cisco Systems (2001) "Meet the Cisco Intelligent Engine 2100" Service

Authors

Mohammed A Qadeer is a Lecturer with the Department of Computer Engineering, Aligarh Muslim University, India. Earlier, he was working with Cisco Systems Inc. as a Network Consulting Engineer with the Advanced Services division in the APAC region. He received his B.Sc. Engineering (Computer Engineering) from Aligarh Muslim University in 1996. He has an experience of 12 years in the area of computer networks and systems. He has been a TPC reviewer for IEEE CCNC 2008, WCNC 2009, ICC 2009, GCC 2009 and ICLAN 2008. Established global and nationwide setups of Internet Service Providers (ISP), Internet Exchange Points (IXP), Internet Data Centre (IDC) and Content Delivery Networks (CDN)

both from a Networks and Systems perspective. His areas of research are computer networks, wireless networks, mobile computing, next generation networks and QoS.

Shahid Habib is currently working as a Engineer - Product Verification, in Tejas Networks Limited, Bangalore, India. He received his Bachelors from Z H College of Engineering and Technology, Aligarh Muslim University in the year 2008. He has authored 4 papers published in international conferences and proceedings in the field of Networking. His papers have also been accepted in international journals. He was sponsored by his company to present his research paper in an IEEE Conference in Uzbekistan. His research interests are Bandwidth Management, PBT (Provider Backbone Transport) and QoS.

A Y Javaid is currently working as an Associate Application Developer in Computer Sciences Corporation India Limited, NOIDA, India. He received his Bachelors from Z H College of Engineering and Technology, Aligarh Muslim University in the year 2008. He received a special recognition award as Red Hat Scholarships - 2005 for his work in the field of face Recognition. He has authored 4 papers published in international conferences and proceedings in the field of Networking and 1 paper in national conference in the field of Face Recognition. His research interests are Bandwidth Management, Face Recognition and DSMS.