An Enhanced Bandwidth Management Scheme for Improved Quality of Service in Network Communication System

Munir Kolapo Yahya-Imam
Malaysia University of Science and Technology, Selangor, Malaysia
Email: mk.yahya2007@yahoo.com

Sellapan Palaniappan and Vathsala Devi
School of Science and Engineering, Malaysia University of Science and Technology, Selangor, Malaysia
Email: {sell, vathsala}@must.edu.my

Abstract—Bandwidth resource management has been adjudged a powerful scheme in controlling communications on a network to avoid overfilling the link to capacity which always results in network congestion and poor quality of service. Bandwidth throttling, a technique widely used in many universities is been considered in this work. It is a process of blocking some links on the network over a period of time in order to conserve the bandwidth. However, it has these limitations: denial of service, slow loading, skipping and stuttering experiences jeopardizing efforts of researchers and general usage of the network which are great challenge in an academic environment. Hence, needs for an enhanced bandwidth management mechanism to overcome these limitations and to improve quality of network service. Effort in this paper is to design and implement an enhanced bandwidth management mechanism to improve efficiency and quality of service in an academic environment by integrating two intelligent tools (SQUID and CBQ) in a prioritization scheme. The enhanced scheme presented will provide platform for users to use the network at will and at different rate, thereby improving efficiency and quality of service. Consequently, this will enhance research activities in no small measure in our citadel of learning.

Index Terms—bandwidth management, quality of service, class based queuing, squid delay pools, tertiary institution network performance

I. INTRODUCTION

Bandwidth management can be described as the process of measuring and controlling communications on a network link, to avoid overfilling the link to capacity which always results to network congestion and poor quality of service (QoS) "Ref. [1]". Bandwidth management has been a great concern especially in an academic environment where research and collaborative learning are regular process. With the high costs of bandwidth many institutions, ISPs and businesses managers have been finding it difficult to shove a router

on their network, connecting to the internet and hope for high quality of service "Ref. [2]". "Ref. [3]" has defined bandwidth management as a general term given to collection of tools and techniques that an institution can use to reduce demand on critical segment of their network. Research has shown severally that bandwidth management have posed many challenges to the tertiary institution network designers as a result of limited and unavailable bandwidth resources to extend her services "Ref. [4]". This coupled with the fact that students tends to use lot of peer-to-peer applications that suck up the overall bandwidth thereby jeopardizing efforts of researchers and normal users activities on the network. Hence, the need to design an enhanced bandwidth management technique for effective and efficient network communication and quality of service.

Effect in this paper is therefore directed at designing and implementing an enhanced bandwidth management technique combining intelligent tools in Linux camp to manage the limited network bandwidth resources that are available in the most efficient way, which will consequently improve network quality of service.

II. LITERATURE REVIEW

Higher institutions of learning are faced with major obstacles in their use of networked information resources simply because the price of bandwidth is disproportionately high and it is difficult to improve international network connectivity "Ref. [5]". Some research works had been carried out in this realm by some network researchers in order to find lasting solution to these problems. Among them are:

Dynamic and Adaptive Bandwidth Management "Ref. [6]" which was proposed to provide a satisfactory QoS which guarantees possible wireless multimedia services under variable traffic conditions. The proposed scheme was designed to provide high bandwidth utilization efficiency and low call blocking probability with minimum QoS fluctuation. The technique works fine because it reduces the probability of frequent quality

changes and provides more satisfactory services for higher priority flows but unfortunately, it only improves the quality of service of high priority flows.

Weighted Round "Ref. [7]" was used to classified packets into various service classes and then assigned a queue of different percentage of bandwidth and they are serviced in round robin order. The technique works well because it ensures that all service classes have access to at least some configured amount of network bandwidth to avoid bandwidth starvation and not compromising the quality of service. The major drawback in this work is that it cannot guarantee fair link sharing. When configured to handle high bandwidth service applications like VoIP and IPTV, it may induce unwanted latency in the transmission.

Fair queuing "Ref. [8]" was used to allocate network bandwidth based upon usage by individual flows. It looks to be more intelligent in making choices, and it is based on relative allocations rather than priority. The problem with fair queuing is that in most cases it manages fairness between flows instead of users or customer. Although it works well in an environment where there are small number of users and small number of definable flows.

In order to effectively manage a network connection of any size, there is need for multifaceted approach that includes effective network monitoring, a sensible policy that defines acceptable behaviour, and a solid implementation that enforces these rules "Ref. [9]". Effective management and optimization of bandwidth are critical to research and education and there is urgent demand for training skills and knowledge developed within this area. Research "Ref. [10]" has shown that majority of higher Institutions undertake little or no monitoring of their network bandwidth. The same research also recommended that improving bandwidth management is probably the easiest way for universities to improve the quality and quantity of their bandwidth resources for educational purposes. Moreover, internet connectivity and access to networked information resources are increasingly essential requirement for any research or educational institution and to achieve this, capacity development within the area of bandwidth management is an essential element.

III. SYSTEM DESIGN

The aim and objective of this research work is to proffer solution to the identified weaknesses of the existing technique by designing and implementing an enhanced network bandwidth management system. In this work, a prioritization technique is presented in which some network traffic will be given a higher priority over the other, based on consumption or utilization rate and this will be carried out on the network main server using Linux OS. Implementing the new bandwidth management configuration inside a Linux sever will highly improve the network quality of service and also proffer solution to the issue of insecurity or threat on the university network. The network bandwidth is proposed to be shared at two levels inside the main server. Any network traffic that falls into any of these levels will be

given a specified speed rate that is assigned to that level. A simple scenario of this is assuming the total data subscribed by a university is 1GB and the speed is 5Mbps, therefore any user trying to access html link (email, online conference papers, online journals etc.) access will be given at a speed rate of 128kbps while user accessing a website that has to do with videos tutorials, online game for students, downloading of software etc will be given a speed rate of 64kbps and no matter the type of downloading software any user might be using, with the help of the tools involve in the design of the improved system, they will have access but not exceeding the specified speed limit.

Fig. 1 depicts the architecture of the new technique. The components of the proposed system comprises of PREROUTING chain, SQUID, CBQ and OUTPUT chain. The PREROUTING chain inspects the packets on arrival and determines which one to be allowed first based on predefined priority rules. SQUID is to fine tune the rate at which each traffic transmits while CBQ will be used to limit any traffic that try to bypass the SQUID configurations. OUTPUT chain will be used to determine which packet goes out first based on some sets of priority rules.
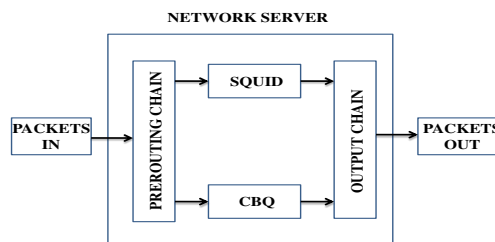

Figure 1. Model of the new prioritization technique

Tools for the implementation

The design will involve both software and hardware tools in which they are:

Software tools

Linux OS: It provides us with a kernel where the configuration will be plugged in.

CBQ Configuration script: It will be used for classification of network traffics and setting up some rules.

SQUID: It will be used to fine tune the speed rate of individuals on the network using the delay pool feature.

Hardware tools

Complete Computer System: This will be used as a network server where all the aforementioned software will be installed and configured.

Network switch: This will serve as AP (Access Point) Bridge for workstations on the network.

UTP CAT-5 Network Cable: This will carry the physical connection between server, switch and workstations.

Network Card (NIC): This will act as an interface between a terminal connector and our main sever.

RJ-45 Terminal Connector: This will be a terminator at each end of the CAT-5 network cable.

IV. IMPLEMENTATION

We will start the implementation by checking the network card information of the system by typing the command "**sudo ifconfig**" this will give us all the necessary information that will be needed for the next stage of the configuration. After we have gotten the idea of the IP address for the network card and other necessary information about it, the next step is to give the server system a static IP address. This will allow the workstations on the network to get connected to the server easily. To do this, we need to open the network interface and plug in the necessary information that will enable the system to use a static IP address. The next stage is to install squid with the command "**sudo apt-get install squid3**" After the installation, we navigate to the directory where the squid configuration file is located using the command "**cd /etc/squid3**" then we open the configuration file. There are over 5000 lines of codes on this configuration file, we are going to add and edit the ones that we need base on how we want the network to behave. The first step here is to scroll down to the "**acl**" (access control list) section to enable and add some codes (Fig. 2) after which we grant permission to the network and also we assign a port number to http port where we want squid to listen on (Fig. 3). We also need to setup some administrative parameters. This will show the network user where to lodge there complain if need be.

Figure 2.   Creating some acl lists

Figure 3.   Granting permission to the network parameters

Delay pools configurations

Configuring the delay pool is one of the most important aspects of this work. Here we need to create couple of access control list as shown in Fig. 4. The first access control list (acl) that was created is "**munir_net**" this is the network where the configuration is going to be applied. It contains the network IP address and "**24**" to show that it is a class C IP address. The alphabet letter "**-i**" is to show that the address will accept any case be it lower or upper case. "**acl**" and "**url_regex**" are both reserve words. "**bad_extensions**" is the second acl and this is where all the file extensions that we wish to limit bandwidth on are attached. The third acl is "**munir_browsing**" which is used to limit bandwidth on individual page that falls on any of the attached extensions. "**work_times**" is to specify the period when user can actually have access to the network, "**time**" is also a reserve word and since there are three pools (munir_net, bad_extensions and munir_browsing) in the acl list that was created, then we assign number "**3**" to our delay pool.

Next is to assign classes to these pools, there are five (5) different classes at this point therefore, based on how we want our network to behave, class 2 was assigned. This will allow us to set some parameters for an individual that falls on any of the categories of the acl lists that was created earlier. We also grant permissions to our network as shown in Fig. 5, this will allow the acl list that was created to run on the network. When setting up the parameters, we gave the network (delay_parameters 1) unlimited access "**-1/-1**", on bad extensions (delay_parameters 2), we limit individuals that falls into this category to 64kbps "**8000/8000**" finally, we limit individual page on the network (delay_parameters 3) to 128kbps "**8000/16000**". All users on the network cannot exceed these specified limits. Other applications that are been commonly used among students that can bypass this limit will be considered using cbq. After this configurations, we save the file by pressing "**Esc**" and "**:wq**" keys from the keyboard. "**Esc**" will take us out of insert mode and "**:wq**" will save and quit the file. Once all these have been done, the next step is to restart the squid service using the command "**sudo service squid3 restart**". This will load the new configurations and once it restarts successfully you will see something like Fig. 6

Figure 4.   Creating access control lists

Figure 5.   Assigning classes, permissions and parameters.

Figure 6.   Restarting squid service

CBQ configurations

In a situation whereby some users try to bypass the squid configuration probably with the help of some applications like napster, audio galaxy, window media, realaudio etc that are commonly used by many students, therefore we configured additional script known as cbq.init version **0.7.3**. Making use of this script will restrict their speed to 50kbps. To start the configuration here, first step is to create a script and plug in some configurations using the command "**sudo vim cbq-50.test-network**". Where "**cbq**" shows that it is a cbq script, "**50**" is the class id and "**test-network**" is the user given name. Once the script had been created, some configurations were plugged in, as shown in Fig. 7.

Figure 7.   CBQ configuration script

After all these have been put together, we save the file just as we did previously in squid.conf file and then "**compile**" the script using the command "**sudo /etc/rc0.d/cbq.init-v0.7.3   compile**". Once the compilation is successful, next is to start the cbq service with the command "**sudo /etc/rc0.d/cbq.init-v0.7.3 start**".

Iptables configurations

The final stage of the whole configuration is to set priorities for traffics that will be passing through the network. This will be achieved with the help of a tool

know as "**iptables**" which can be found inside Linux kernel. In doing this, we make use of PREROUTING and OUTPUT chains. PREROUTING chain was used to determine which traffic should be allowed into the network first by assigning a level of priority to some port numbers and OUTPUT chain on the other hand is used to determine which traffic should be send out of the network first as shown in Fig. 8 and Fig. 9 respectively. Note that ports with high priority number have lower level of priority. After setting up all these priority rules, the next step is to save them using the command "**sudo iptables-save > /etc/network/iptables.rules**" then restart the network service with the command "**sudo /etc/init.d/networking restart**".

Figure 8.   PREROUTING chain configuration

Figure 9.   OUTPUT chain configuration

V.   RESULTS

After putting all the codes and configuration together on our Linux Machine using ubuntulinux, some testing were made to ensure that it met with the paper goal. Among the test was to check if the quality of service has improved or not. In doing this, we connect some workstations to our LAN network. After the connection has been established, we try to download an "iso" file from www.ubuntu.com/download and check the speed. The downloading started automatically and the speed was within the specified rate i.e. 64Kbps. Recall that "**.iso**" extension is a member of bad_extensions acl on the squid delay pool configuration. In the 51% of the downloading, the speed was within the specified rate as shown in Fig. 11. Similarly, a test was carried out on the existing technique before the configuration and Fig. 10 shows the downloading speed. Moreover, despite the fact that few users were on the network at the time of testing, the speed was still poor. The fact that our download speed

rate on the improved system did not exceed the specified speed rate has proved that users on the network are using the squid configuration and also the network behaviour is in line with the configurations that were made. Comparing the improved system with existing one has shown that the level of quality of service of the network has improved.
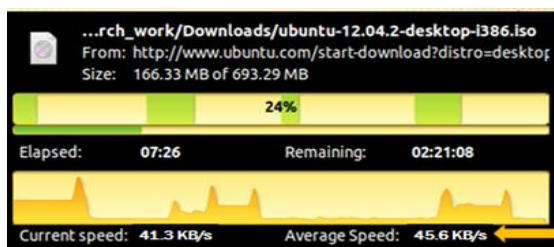

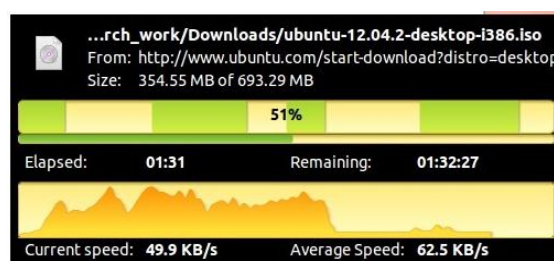Figure 10. Downloading an iso file (Previous System)

Figure 11. Downloading an iso file (Improved System)

VI. CONTRIBUTION

The improved technique provide a stable platform for users to use the network at will and consequently will enhanced research activities in no small measure in our universities and other tertiary institutions if fully implemented.

VII. CONCLUSION

In conclusion, designing an improved bandwidth management technique is an elusive task. It involves full understanding of the strengths and weaknesses of the existing bandwidth management technique. It also involves consummate technical skills in this area in order to have a better understanding of the problems that network users encounter. In the improved technique, solution were proffer to the identified problems that user encountered in the existing technique. The output generated show that the limited bandwidth available had been fully optimized through the use of the ultimate conditioner script which created a stable platform for the prioritization of the scarce bandwidth and allow the network traffic to pass through the proper chains. With the new technique, individual users now have access to the network at different speed.

REFERENCES

[1] J. Evans and C. Filsfils, *Deploying IP and MPLS QoS for Multiservice Networks: Theory and Practice,* Morgan Kaufmann, 2007.

[2] D. G. Balan and D. A Potorac, "Linux HTB queuing discipline implementations," in *Proc. 1st International Conference on Networked Digital Technologies,* Electr. Eng. & Comput. Sci. Fac., Stefan cel Mare Univ. of Suceava, Suceava, Romania, NDT '09. 28-31 July 2009, pp. 122-126.

[3] K. Mochizuki, S. Shimazaki, D. Hanawa, and K. Oguchi, "Proposal of new traffic control method in the next generation home network," in *Proc. Telecommunications and Signal Processing, 35th International Conference on Sci. & Technol.*, Seikei Univ., Musashino, Japan, 2012.

[4] C. J. Taylor, *Bandwidth and the Wireless*, 4th Ed, Compton Press, Los Angeles, 2010.

[5] L. Chitanana, "Bandwidth management in universities in Zimbabwe: Towards a responsible user base through effective policy implementation," *International Journal of Education and Development Using Information and Communication Technology*, vol. 8, no. 2, pp. 62-76, 2012.

[6] H. Guo and G. Kuo, "A dynamic and adaptive bandwidth management scheme for QoS support in wireless multimedia network," presented at Vehicular Technology Conference, VTC 2005-Spring. IEEE 61st, China, 2005.

[7] A. H. Rashwan, H. M. Elbadawy, and H. W. Ali, "Comparative assessments for different WiMAX scheduling algorithms," in *Proc. the World Congress on Engineering and Computer Science*, San Francisco, USA, vol. I, October 20-22, 2009.

[8] M. Mirza, G. Rashed, and M. Kabir, "A comparative study of different Queuing techniques in VOIP, video conferencing and file transfer," *Daffodil International University Journal of Science and Technology*, vol. 5, no. 1, pp. 37-47, January 2010.

[9] L. M. Patnaik and K. R. Venugopal, "Bandwidth limiter," in *Proc. 2nd International Conference on Information Processing*, India: I. K International Pvt limited, Jan., 2008.

[10] C. George and F. Kizumba, "Bandwidth management issues in African universities," presented at the *AAU Conference*, Lome Togo, 2005.

Munir K. Yahya-Imam is currently pursuing Masters of Science degree in information technology from Malaysia University of Science and Technology (MUST), Selangor, Malaysia. He holds a Bachelor degree in computer science major from Al-Hikmah University, Ilorin, Nigeria in the year 2009. Research area of interest includes Network Bandwidth Management and Data communication.

Dr. P. Sellappan is currently an Associate Professor at Malaysia University of Science and Technology (MUST). Prior to joining MUST, he was holding a similar position at the Faculty of Computer Science and Information Technology, University of Malaya. He holds a Bachelor degree in Economics with Statistics major from the University of Malaya, a Master in Computer Science from the University of London and a PhD in Interdisciplinary Information Science from the University of Pittsburgh. He has been in the teaching profession for over 25 years and has taught courses such as Programming, System Analysis and Design, Software Engineering, Object-Oriented Modelling, Human Computer Interaction, Database Systems, E-Commerce, Operating Systems, Management Information Systems, Mathematics and Statistics. He has published several journal and research papers on computer-related topics. He is also the author of several IT books. He has served as an IT Consultant for several local and international agencies such as the Asian Development Bank, the United Nations Development Program, the World Bank and the Government of Malaysia. He is also a recipient of several national grants to undertake IT-related research. He was a member of IEEE (USA), Chartered Engineering Council (UK) and British Computer Society (UK), and is currently a member of the Malaysian National Computer Confederation (MNCC).

Vathsala Devi is currently a lecturer at the Department of Information Technology under the School of Science and Engineering, Malaysia University of Science and Technology (MUST), Selangor, Malaysia. She holds a Bachelor degree in computer science from De Montfort University, Leicester UK, a Master of computer science degree from University of Malaya, Malaysia. She has been in the teaching profession for over 11 years and she has taught courses such as Computer Networking and Computer Programming (C, C++, Java, C#, JavaScript, HTML, PERL, Network Programming, and VB.Net). She has published several journal and research papers on computer-related topics.