An Approach of QoS Library Redirection Method for DiffServ in Microsoft Windows Systems

*Wen-Shyang Hwang, Yu-Ben Miao, Chih-Heng Ke, Ji-Feng Chiu and Ce-Kuen Shieh
*Department of Electrical Engineering,
National Kaohsiung University of Applied Sciences, Taiwan R.O.C.
Department of Electrical Engineering,
National Cheng Kung University, Taiwan R.O.C.

Abstract — **The paper presents an implementation of QoS Library Redirection (QLR) for DiffServ in Microsoft Windows systems. In QLR method, the legacy application obtains packet pre-marking ability without any modification of source codes. We have upgraded the FTP and vic applications by using QLR in our Microsoft Windows testing platform. Our experimental results show that this approach can use really effectively and provide high performance of the QLR in Microsoft Windows systems.**

I. Introduction

The Internet architecture only offers the best-effort delivery service model. All user packets compete equally for network resources. But this kind of service model will no longer meet the delivery needs of time- and performance-critical applications, such as video-on-demand, video teleconferencing and IP telephony. These applications are very sensitive to the quality of service they receive from network. In the future, Internet will not only provide the best-effort service, but also support the differential Quality of Service (QoS) to meet the various requirements. The Differentiated Services (DiffServ) architecture has been recently become the preferred method to address QoS issues in IP networks. It can support multiple levels of services while preserving the scalability and simplicity of the current Internet. Packets are classified and marked with appropriate type-of-service (ToS) value at the source host or the edges of the network [1]. Routers in the core of the network treat packets based on their classes. Per-flow state does not need to be maintained in the core routers, which leads to increased scalability. In the scheme, internet applications should give users the option of purchasing high or low service quality by pre-marking its traffic via setting the TOS field in the IP header [1]. However, legacy applications are QoS-unaware which means that these applications do not allow users to choose preferred service. One of the solutions to this problem is to modify the source codes of legacy Internet applications to add the QoS ability. However, this needs the source code available.

In the [15], we had proposed a transparent packet pre-marking method named QoS Library Redirection (QLR). This method can transparently upgrade legacy applications to own packet pre-marking ability without any modification of their source codes. But this approach depends on the library redirection mechanism supported only in UNIX operating system. To realize QLR approach in Microsoft Windows systems, one of the most popular operating system, the library redirection method should be revised. This paper proposes a method to apply QLR in Microsoft Windows systems.

The rest of this paper is organized as follows: Section II gives an overview of differential service and QLR. Section III presents the QLR method for Microsoft windows systems, and Section IV describes our experimental descriptions and results. Finally, a conclusion is presented.

II. Background

In DiffServ, A single Differentiated Services CodePoint (DSCP) identifies each behavior aggregate. DS field can be pre-marked at the host end and remarked at the ingress router of differentiated service domain. To accommodate this scheme, QoS-aware applications call *setsockopt()* function to pre-marked proper DSCP for its data flow. QLR is used to transform legacy Internet applications into QoS-aware by means of library redirection. Its main purpose is to intercept the function calls issued by applications and then to set DSCP in the IP header. After that, real functions in the system libraries are called to finish original jobs.

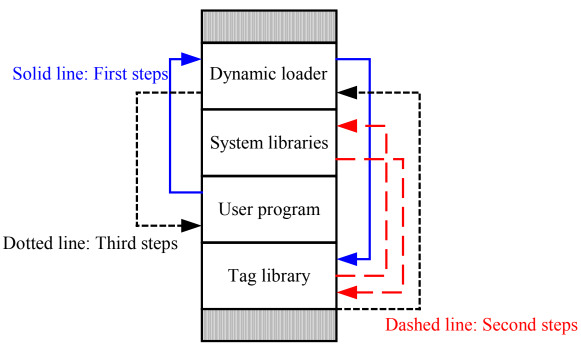

Fig.1 QLR procedure

0-7803-7208-5/01/$17.00 (C) 2001 IEEE

In UNIX platform, if we put a packet pre-marking function in the shared library which we will call tag library in the subsequence and setup the environment variables, applications can transparently pre-mark their traffic to get better service. Figure 1 shows QLR procedure. First, when a program makes a load-and-call service request to the operating system, the tag library will be loaded if necessary and called. Second, the originally invoked *socket()* routine continues to complete and setting DSCP included in the tag library if necessary. Finally the operating system returns the control flow to the program that issued the request.

III. QLR Method for Microsoft Windows Systems

Although the QLR method is based on the redirection of dynamic linking for shared library, its implementation varies with the operating system. These dynamic linking libraries are called DLLs in Microsoft Windows systems.

To add the packet pre-marking function into the target application, the invocation of *socket()* routine need to be intercepted and the *setsockopt()* routine is inserted before the control flow returns to the application. To intercept the invocations of *socket()* routine in Microsoft Windows systems, we have to modify the Import Address Table (IAT) in programs. IAT is an address table that stores the addresses of functions imported from a DLL. Each application maintains IATs for DLLs. When the application invokes an imported routine, the real address of that routine is obtained by looking up IAT and the control flow transfer to that address [16].

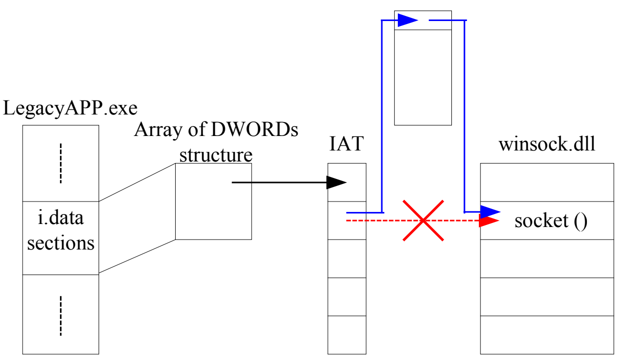

Fig.2 QLR procedure in Microsoft windows systems

In QLR method, we prepare a stub routine to set the DSCP field of IP header. Since every program has its own address space, a small loader named QLR loader is needed to load our stub routine into the address space of our target program. Fig.2 shows QLR procedure for Microsoft windows systems. The steps as follows: First, QLR loader loads the target program and the stub routine into memory. This loader also changes the *socket()* routine address stored in IAT with the address of the stub routine. Second, when the application invokes *socket()* routine, the control flow is redirected to the stub routine. Third, the control passed directly to the original *socket()* routine and the originally invoked *socket()* routine continues to complete. Fourth, the control flow return to the stub routine and *setsockopt()* is called to set a proper DSCP for that socket. Fifth, the control flow returns to the target program. Consequently, application can transparently pre-mark its traffic to get better service.

IV. Experimental Descriptions and Results

To verify the QLR method in Microsoft Windows systems, there is not only a DiffServ test platform was constructed but also two different type applications of transport layer protocol (TCP/UDP) are implemented on this platform. In TCP environment, popular client-server applications CuteFTP and Serv-U are chosen to examine QLR mechanism. In UDP environment, the most widely video conferencing tool on Mbone vic is chosen. This tool is currently used in many teleteaching projects. The test platform is mainly consisted of CBQ (Class-Based Queueing), DiffServ daemon and several test tools.

CBQ is a mechanism that allows a hierarchy of arbitrarily defined traffic classes to manage resource on a link in a controlled fashion [7][8]. In this case, the CBQ link-sharing structure is shown in Figure 3. **Diffserv daemon** is used to classify and police the traffic in the input interface and shape the traffic in the output interface.

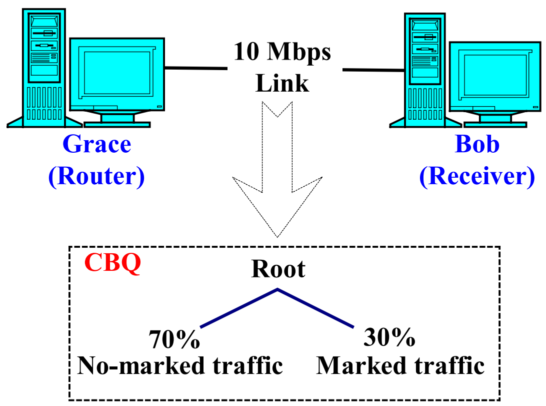

Fig 3 Link-sharing structure

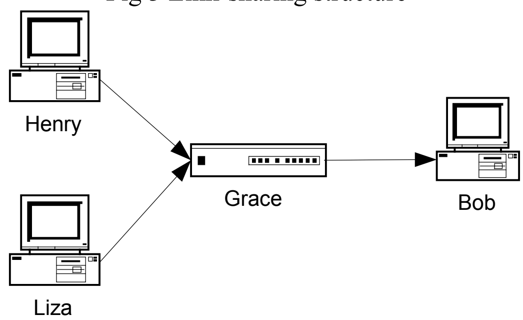

Fig. 4 Testing Platform

The testing platform we used is shown in Figure 4. There are three hosts and one router. The two sending hosts are named Henry and Liza. Bob is the name of the received host that is the destination for both traffic streams. Henry is used to send the pre-marked traffic, and Liza is always sending the best effort traffic. All interfaces of Grace are 10 Mbps point-to-point Ethernet links.

Fig. 5 shows the details of the system configuration and features. There are CBQ, DiffServ daemon, and other tools such as Mgen, TTT, and vic. **Mgen** provides the statistics ability to generate traffic in the IP network [9]. **TTT** (Tele Traffic Tapper) is the kind of tcpdump but capable of real-time, graphical, and remote traffic monitoring [10]. CuteFTP is a Windows based File Transfer Protocol (FTP) in client, it allows users to utilize the capabilities of FTP without having to know all the details of the protocol itself [17]. Serv-U is a flexible and easy to set up FTP server (or Daemon) for Windows platform. It can upgrade a network PC to an FTP site on the Internet [18]. vic is a real-time multimedia application for video conferencing.

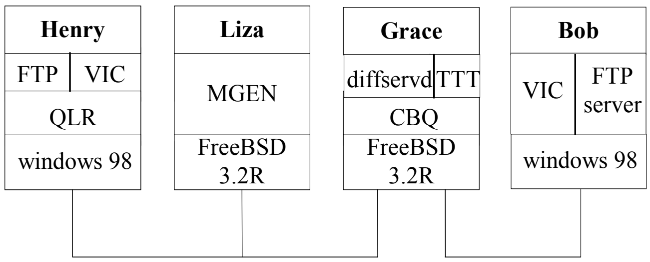

Fig. 5 System configuration

A. TCP experiment

In this experiment, we send a file (24M bytes) to Bob and generate different levels of background traffic from Liza to Bob, then observe the throughput of pre-marked and non pre-marked traffic, illustrated in Figure 6. To understand the variation of traffic, TTT is used to monitor the output traffic of router. Figure 7 and 8 show the non pre-marked and pre-marked traffic at Grace's output on an overloaded network respectively. Clearly, non pre-marked traffic is oscillated heavily, but pre-marked traffic is still smoothly.

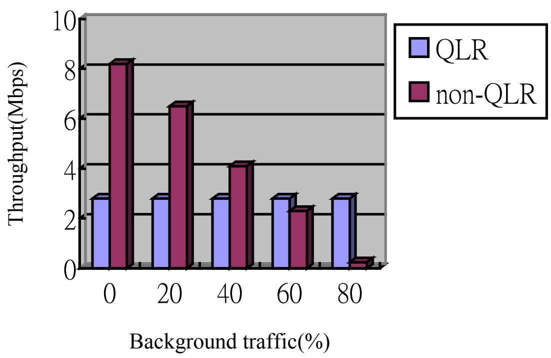

Fig. 6 Compare throughput of pre-marked and non pre-marked

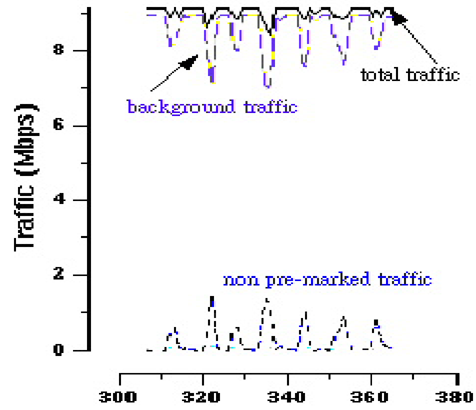

Fig 7 Non pre-marked traffic on an overloaded network

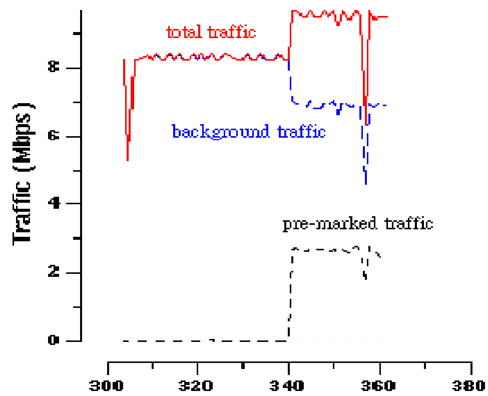

Fig 8 Pre-marked traffic on an overloaded network

B. UDP experiment

In UDP experiment, Henry plays and sends a movie to Bob for an hour. At the same time, we generate different levels of background traffic to cause the network congest. Figure 9 shows the packet loss rate at Bob affected by these different levels of background traffic. From these results, the packet loss rate of non pre-marked will increase when background traffic is increase. Conversely, the packet loss rate of QLR vic traffic does not be affected by increasing background traffic.

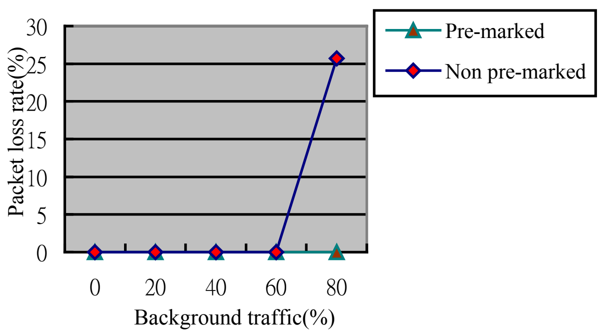

Fig. 9 Packet loss rate of Bob (Receiver)

V. Conclusion and Future Work

In this paper, we have proved the feasibility of QoS Library Redirection method in Microsoft Windows systems. QLR can transparently set the DSCP field in the IP header, hence the existing Internet applications in Microsoft Windows systems can be pre-marked without modification or recompilation of source code.

In the future, we plan to add an adaptive packet pre-marking mechanism in the QLR method. In the method, the client program with the packet pre-marking engine built in is used to study different requested transfer rate and the relationship between the sending rate and receiving rate.

References

[1] K. Nichols, S. Blake, F. Baker and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers," RFC 2474, December 1998.

[2] Wu-Chang Feng, Dilip D. Kandlur, Debanjan Saha, and Kang G. Shin, "Adaptive Packet Marking for Maintaining End-to-End Throughput in a Differentiated-Services Internet," IEEE/ACM Transactions on Networking Vol. 7, No. 5, October 1999.

[3] David D. Clark, and Wenjia Fang, "Explicit allocation of best-effort packet delivery service," IEEE/ACM Transactions on Networking, Vol. 6, No. 4, August 1998.

[4] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang and W. Weiss, "An Architecture for Differentiated Services," RFC 2475, December 1998.

[5] Beck, Leland L., "System software," second edition, Addison Wesley, 1990

[6] W. Richard Stevens, "UNIX Network Programming", Volume1, Networking APIs：Sockets and XTI, second edition, Prentice Hall Inc., 1998

[7] Sally Floyd and Michael Francis Speer, "Experimental Results for Class-Based Queuing," http://www-nrg.ee.lbl.gov/floyd/cbq/notes.html, Jan 1998.

[8] S.Floyd, V.Jacobson, "Link-sharing and Resource Management Models for Packet Networks," IEEE/ACM Transactions on Networking Vol.3 No.4, pp365-386, August 1995.

[9] http://manimac.itd.nrl.navy.mil/MGEN

[10] http://www.csl.sony.co.jp/person/kjc/software.html

[11] Juha Heinanen, Fred Baker, Walter Weiss and John Wroclawski, "Assured Forwarding PHB Group," INTERNET DRAFT, November, 1998.

[12] Steven Blake, "Some Issues and Applications of Packet Marking for Differentiated Services," INTERNET DRAFT, December 1997.

[13] Juha Heinanen, "Use of the IPv4 TOS Octet to Support Differential Services," INTERNET DRAFT, November 1997.

[14] S.Blake, et al (1998), "A Framework for Differentiated Services," INTERNET DRAFT, October 1998.

[15] Ce-Kuen Shieh, Chih-Heng Ke, Wen-Shyang Hwang and Ji-Feng Chiu, "A Transparent packet pre-marking method for DiffServ", ICC2001, Volume 9, pp. 2897-2900, 2001.

[16] Matt Pietrek, "Windows 95 system programming secrets", IDG Books, 1996.

[17] http://www.globalscape.com/

[18] http://www.ftpserv-u.com/