A Transparent Packet Pre-marking Method for DiffServ

*Ce-Kuen Shieh , *Chih-Heng Ke,
shieh@eembox.ee.ncku.edu.tw , smallko@hpds.ee.ncku.edu.tw
**Wen-Shyang Hwang, **Ji-Feng Chiu
wshwang@mail.ee.kuas.edu.tw, gary@wshlab2.ee.kuas.edu.tw
*Department of Electrical Engineering,
National Cheng Kung University, Taiwan R.O.C.
**Department of Electrical Engineering,
National Kaohsiung University of Applied Sciences, Taiwan R.O.C.

Abstract — This paper proposes a transparent packet pre-marking method named QLR (QoS Library Redirection) for DiffServ. This method can upgrade legacy applications to obtain packet pre-marking ability without any modification of source codes. In this case, users are allowed to purchase high or low service quality according to their requirements. The major techniques of QLR include the dynamic linking of shared libraries and the function call redirection.

We have upgraded the vic application by using QLR in our FreeBSD testing plateform. The experimental results can show the effectiveness of the QLR.

Keywords: DiffServ, QoS, DSCP, QLR

I. Introduction

Current Internet architecture offers the best-effort delivery service model. All user packets compete equally for network resources. But this kind of service model will not longer meet the requirement of emerging multimedia applications, such as video-on-demand, video teleconferencing and IP telephony, since 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 different users. Basically, the scheme must be accompanied with the billing strategy that discourages users from continually requesting the highest level of service to avoid the waste of network resources. 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 solutions to this problem is to design QoS-aware applications from scratch. But this will cost a lot of effort. Another solution is to modify the source codes of legacy Internet applications to add the QoS ability. However, this needs the source code available.

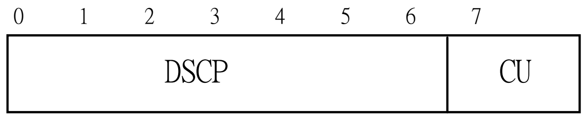
In this paper, a transparent packet pre-marking method named QLR (QoS library Redirection) is proposed. This method can easily upgrade legacy applications to own packet pre-marking ability without any modification of source code. The only thing user need to do before application running is to choose preferred service. Then all traffic sent by this application will be marked according to user's choice.

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

II. Background

Differentiated services are intended to provide a framework and building blocks to enable deployment of scalable service discrimination in the Internet. One reason for developing the differentiated service model is the drawback of per flow-state maintenance in all routers in the integrated service [2][3]. The differentiated service architecture is based on a simple model where traffic entering a network is classified and possibly conditioned at the boundaries of the network, and assigned to different behavior aggregates that are a collection of packets with common characteristics. Each behavior aggregate is identified by a single DSCP (Differentiated Services CodePoint). Within the core of the network, packets are forwarded according to the Per-Hop Behavior (PHB) associated with the DSCP.

DSCP, a replacement header field, is intended to supersede the existing definitions of the IPv4 TOS octet and IPv6 Traffic Class octet. Figure 1 shows DS field structure. Six bits of the DS field are used as a codepoint (DSCP) to select the PHB a packet experiences at each node. Two-bit Currently Unused (CU) field is reserved [1].



DSCP : differentiated services codepoint
CU : currently unused

Fig.1 DS field structure

Figure 2 shows the DS traffic classification and conditioning model. The classifier is used to classify the input traffic according to DSCP or multi-field in the packet's header. A conditioner consists of meter, marker, shaper, and dropper. Traffic meter measures the temporal properties of the packet stream selected by the classifier. Packet marker sets the DS field of the packet to a particular codepoint. Shaper delays some or all of the packets in a traffic stream in order to bring the stream into compliance with the profile. Dropper simply drops packets if they are out of profile [4].

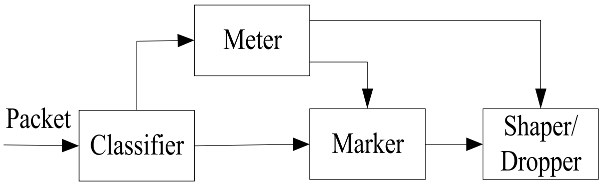


Fig.2 DS traffic classification and conditioning model

III. QoS Library Redirection (QLR) Method

QLR is a method using the dynamic linking of shared libraries based on the redirection of function calls [5]. 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. Dynamic linking is a scheme that postpones the function linking until execution time, and shared libraries are designed to share code segments among programs. So if we put a packet pre-marking function (setsockopt() function which can change the behavior of the socket in most UNIX platform) 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.

In this method, when a program makes a load-and-call service request to the operating system (Figure 3(a)), the operating system will check the environment variable to determine whether the tag library or system libraries are used. If the environment variable is set and the corresponding symbolic name of the routine is in the tag library, the tag library will be loaded if necessary and called. (See Figure 3(b) and Figure 3(c)) In the routine, the TOS field of IP header will be set, afterward the request will be passed to the system libraries, as shown in Figure 3(d). Once the this request is completed in the system libraries, it will be immediately returned back to the tag library then back to operating system. Finally the operating system returns the control to the program that issued the request. This process is illustrated in Figure 3(e). If the corresponding symbolic name of the routine is not in the tag library, control may simply be passed from the dynamic loader to the called routine in system libraries, as shown in Figure 3(f).

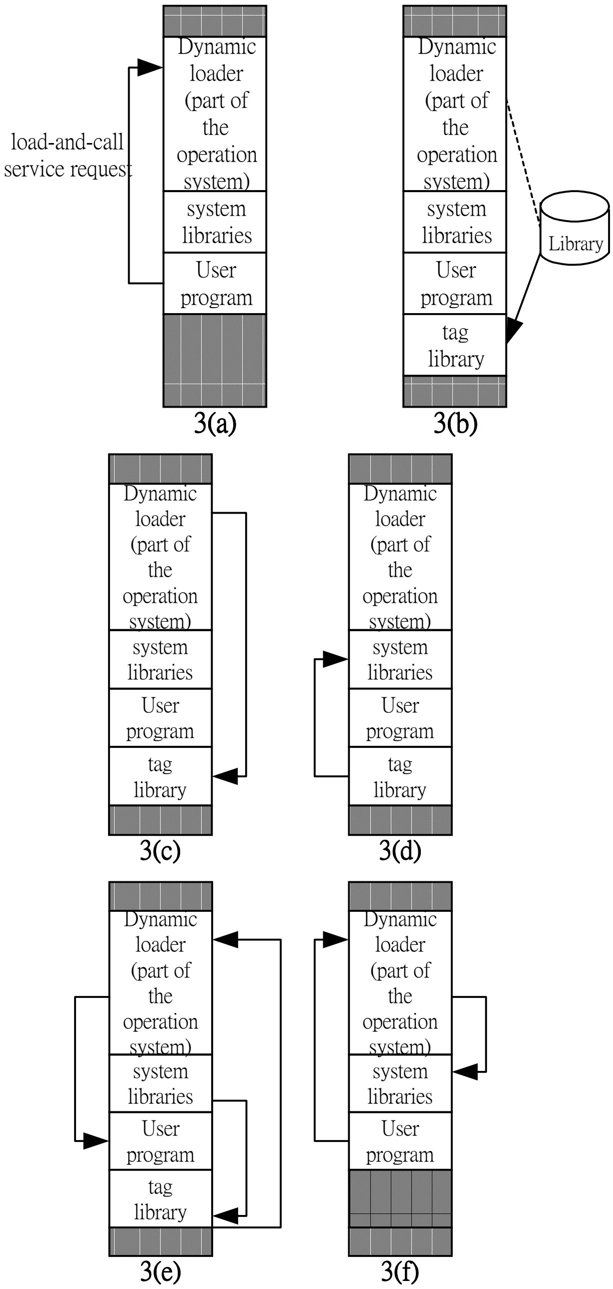


Fig. 3 QLR procedure

Subsequently, we give an example to show how QLR works. We choose FreeBSD 3.2 as the platform, vic as the tested program, and socket() as the intercepted function. First, we should prepare tag library that contains a routine named socket().This routine will execute the setsockopt() function to modify TOS field in the IP header, then use the dlopen() and dlsym() functions to call the socket() routine in the system library. Second, LD_PRELOAD is used to set the environment variable that can be used to specify the alternate shared library [6]. After the above operations, the service request will be redirected to the tag library whenever the program vic executes socket(), and all traffic sent by vic is marked. It is worthy to notice that the other function calls in vic will directly link to the system libraries, hence the overhead of QLR is very light.

IV. Implementation Results

To verify our QLR method, we construct a diffserv test platform. The test platform is mainly consisted of CBQ, 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 our case, the CBQ link-sharing structure is shown in Figure 4. **Diffserv daemon** provides the ability to classify and police the traffic in the input interface and shape the traffic in the output interface.

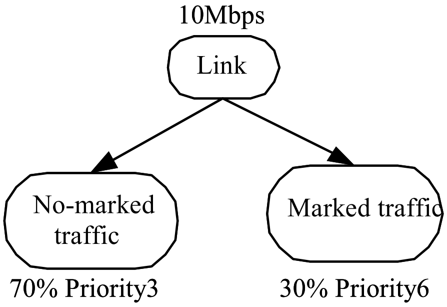


Fig. 4 Link-sharing structure

The testing platform we used is shown in Figure 5. There are three hosts and one router. The two sending hosts are named Henry and Liza, and 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 sending the best effort traffic. All interfaces of Grace are 10 Mbps point-to-point Ethernet links.

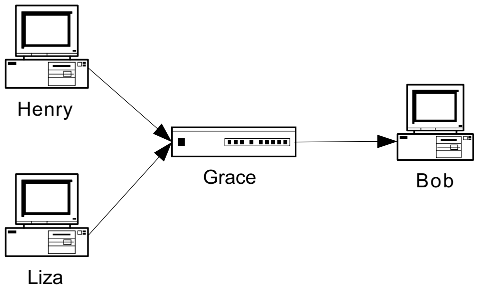


Fig. 5 Testing Platform

Fig. 6 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 measure the performance of IP network [9]. **TTT** (Tele Traffic Tapper) is the kind of tcpdump but capable of real-time, graphical, and remote traffic monitoring [10]. Vic is a real-time multimedia application for video conferencing.

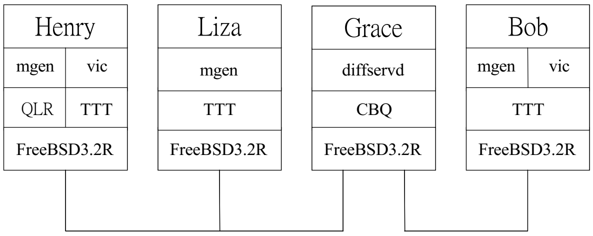
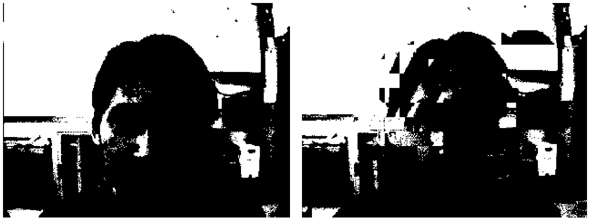


Fig. 6 system configuration

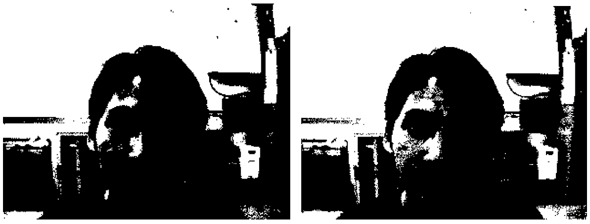
A. vic traffics on an overloaded network

In this experiment, we generate background noise traffic to cause the network overloaded. Figure 7 and Figure 8 shows the vic windows of sender and receiver before/after the DSCP is set. It is obvious that the image is distorted badly when the DSCP is not set. Therefore, we assert the pre-marked traffic will provide a better QoS to applications.



(a) Vic on diff-sender1 (b) Vic on diff-receiver

Fig. 7 Vic With DSCP unset



(a) Vic on diff-sender1 (b) Vic on diff-receiver

Fig. 8 Vic With DSCP set

B. pre-marked traffic and best-effort traffic

To understand the variation of traffic, TTT is used to monitor the incoming and output traffics of router. Figure 9 and Figure10 shows the pre-marked and best-effort incoming traffics respectively, and Figure 11 shows the traffic at Grace's output. The pre-marked incoming traffic is arrived in a rate of 4.8 Mbps, however, it will be shaped in a rate of 3 Mbps as shown in Figure 11 at time 110. At time=125, the best effort traffic is joined this total traffic too, but we can clearly see that the pre-marked traffic flow is not affected by it.

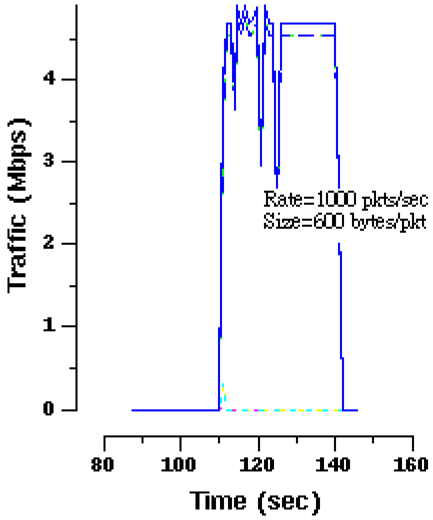


Fig.9 Incoming pre-marked traffic

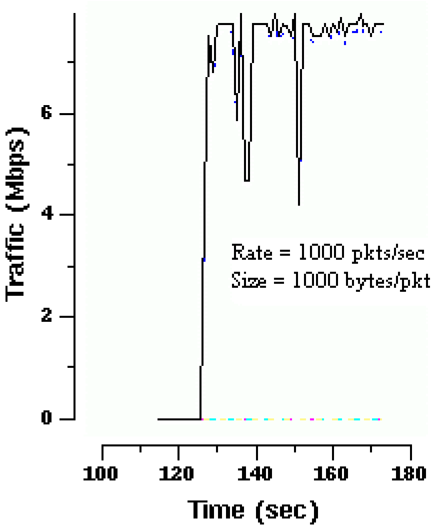


Fig.10 Incoming best effort traffic

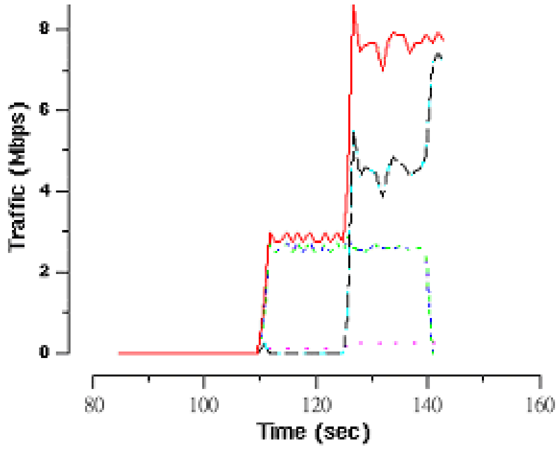


Fig.11 Outgoing Traffic

V. Conclusion and Future Work

In this paper, we have proposed a QoS Library Redirection method that can transparently set the TOS field in the IP header. Hence the existing Internet applications on UNIX platform can be pre-marked without modification or recompilation of source code.

In the future, we plan to add an adaptive packet-marking mechanism in the QLR method. In the method, no packet marking is necessary, as the network load is light. When network load is heavy, marked packets will get the better service than unmarked ones.

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] R.Barden, Ed, ”Resource ReSerVation Protocol（RSVP）-Version 1 Functional Specification,“ Internet RFC2205, September 1997.

[3] Paul P.White,”RSVP and Integrated Service in the Internet : A Tutorial,” IEEE Communications Magazine, May 1997.

[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.