An Application Based Differentiated Service Model

Fugui Wang and Prasant Mohapatra
Dept. of Computer Science and Engineering
Michigan State University
East Lansing, MI 48824
{wangfugu, prasant}@cse.msu.edu

Sarit Mukherjee
Panasonic Technologies Inc.
2 Research Way
Princeton, NJ 08540
sarit@research.panasonic.com

Abstract

In this paper, we propose a flexible differentiated service model, which is based on the application layer so that applications could indicate their network service requirements directly. We classify Internet applications into four categories and provide different schemes for each of the categories to indicate its request. The model hides the complexity of the network services from the end-users but could still make an efficient use of the negotiated network resources. We implement our model in the ns2 network simulator [1] and show that the performance meets our design goal.

1. Introduction

The current Internet is based on the best-effort model where all data packets are treated equally and the network tries its best to ensure reliable data delivery. However, as the Internet is getting more and more popular and commercial, it has become essential to provide different service qualities to different users [2].

Differentiated services (DiffServ) is proposed as an alternative to the best-effort service model. The basic idea of differentiated services is as follows. The core routers only support several (typically 2 to 3) classes of services for simplicity in implementation. When a packet passes through a core router, the type of service (ToS) byte of the IP header is checked. Based on the tagged bits of ToS, a packet is forwarded or dropped using some forwarding or dropping policy. An important task of the edge router is to mark one or two bits of the ToS in the IP header based on the Service Level Agreement (SLA) between the local network domain and the Internet Service Provider (ISP).

DiffServ model provides a soft bandwidth guarantee for a certain network domain or a host. However, the problem still remains as how to allocate adequate bandwidth to different applications, which are competing for resources. A study of the current Internet applications reveals that different Internet applications have quite different bandwidth and delay requirements. For example, EMAIL applications are generally not sensitive to either bandwidth or individual packet delay. Some real-time applications, such as the IP telephony, are sensitive to both bandwidth and individual packet delay. TELNET does not need high overall bandwidth but would not tolerate long individual packet delay, FTP only cares about the overall throughput instead of individual packet delay. So, it is very important to involve the applications in the differentiated services directly. We propose an application-based differentiated service model in which, we classify the existing Internet applications into four categories and provide different marking schemes for different categories. We implement our model in the *ns2* simulator [1] and show the performance of our model.

The rest of this paper is organized as follows. Section 2 introduces the architecture of the DiffServ model. In Section 3, we propose an application-based DiffServ model. We study the performance of our model in Section 4, followed by the conclusions in Section 5.

2. Differentiated Service Architecture

Figure 1 shows a typical differentiated service architecture. The egress and ingress edge router works as

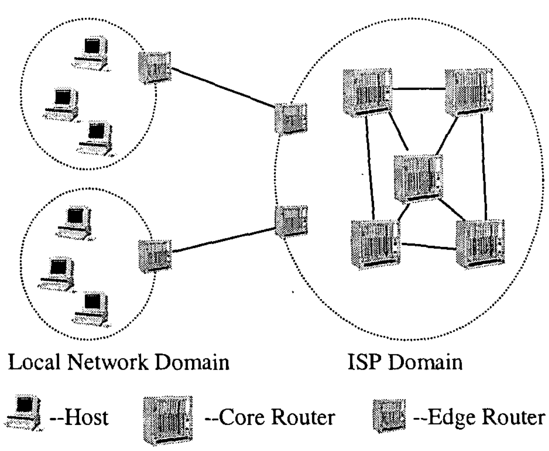

Figure 1: A typical differentiated service model.

0-7695-0777-8/00 $10.00 © 2000 IEEE

the interface between the local network domain and the core network. A marker is implemented in the egress edge router, which marks the packets based on the SLA. The SLA could be negotiated dynamically or statically.

There have been several proposals for differentiated services over Internet. Clark introduced the concept of assured service over Internet [3]. Clark and Fang proposed a 1-bit scheme to support the assured service [4]. Jacobson introduced the premium service [5]. Nichols et al. proposed a 2-bit scheme to support both assured and premium services [6]. According to these proposals, there are generally two types of "high priority" services that could be supported by the Internet core routers. One is the premium service, which is a used for delay sensitive real-time applications. The other is the assured service, which uses a RIO [4] scheme so that the IN profile packets will have a much lower drop probability than the OUT of profile packets. If the network resource is not over provisioned, the IN packets should almost never get dropped. The OUT packets (i.e. best-effort packets) will only compete for the rest of the bandwidth. As the two-bit scheme contains both of these two high priority services, we use it to describe the queuing model of the core router. A typical queuing model for the core router is shown in Figure 2. A premium packet is enqueued in the premium queue, an assured/untagged packet is enqueued in the RIO queue which is a Random Early Detection (RED) [7] queue with different drop preference for assured packet (marked as IN) and untagged packet (marked as OUT). It is expected that the premium packets will incur short delay, the IN packets have low drop rate and the OUT packets get dropped in case of network congestion.

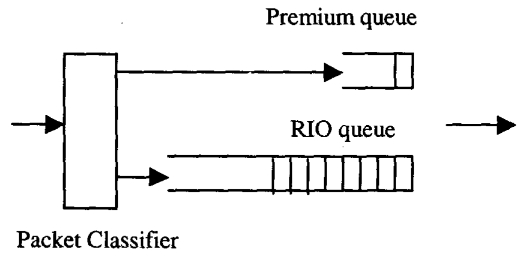

Figure 2: A typical queue model in the core router.

3. Application-Based Differentiated Services

Previously proposed differentiated service models have not considered the detailed properties of Internet applications. Our work is focused on how applications could submit their bandwidth and delay requirement in a feasible way to the edge router and how the edge router can efficiently allocate the negotiated SLA to applications based on the request from applications.

3.1 Previous Work

Before presenting our application-based differentiated service model, we briefly discuss two existing models: application-independent model and flow-specific differentiated service model.

In the application-independent model proposed by Clark and Fang [4], the edge router negotiates an SLA with the ISP for the local domain. Each host of the local domain has an expected service profile in the edge router. If the traffic from a user to Internet does not exceed the bandwidth assigned in the host's service profile, all the packets are marked as IN. Otherwise, the exceeded part are marked as OUT. This model is simple, but has the following disadvantages:

(1) As we have argued in Section 2, allocating bandwidth to each user statically could not make high utilization of the negotiated bandwidth.
(2) Host is the atomic element in the model. Services cannot be differentiated between different applications. However, different applications may have varying bandwidth requirements.

In the flow-specific differentiated service model, the edge router negotiates an SLA with the ISP for the local network domain. Each flow will apply for bandwidth dynamically from the edge router. The edge router does an admission control for each flow and accepts a flow if it can allocate enough bandwidth for that flow. Otherwise, the flow is rejected or degraded to best-effort service. The RSVP protocol [8] has been proposed for this bandwidth reservation procedure in the local domain. Compared to the first model, this model could make better utilization of the negotiated bandwidth because it is dynamically allocated to each flow. Different applications could also get different service qualities by specifying different bandwidth requirements. However, there are some limitations of this model as follows:

(1) Resource reservation is only useful for some real-time applications. It is almost impossible and also not necessary to specify a peak rate, average rate for most of the TCP applications. Although it is reasonable to specify the traffic rate for some real-time streams such as the Internet telephony or Internet video, it is very hard for a user to specify the transfer rate for an FTP or a web user, where the user will ideally need as much bandwidth as available.
(2) It is non-preemptive. Once an application has submitted a rate and get admitted, it will not yield any resource to others. This is desirable for real-time applications. But for most of the non-real-time TCP applications, bandwidth requirement is not strict.

3.2 Application-Based DiffServ Model

Involving applications in the service differentiation process is necessary since it could improve the overall resource utilization and distribute the negotiated bandwidth to different applications more efficiently. Furthermore, most of the current Internet applications are TCP-based. So we need to design a marking model which could make both the real-time applications and the more popular TCP-based applications use the differentiated services efficiently.

We first need to analyze different properties of current Internet applications. Based on bandwidth and delay requirements, the Internet applications could be classified into four categories:

(1) The first category is the real-time applications, which need guaranteed bandwidth and would not tolerate long individual packet delay. The bandwidth could be specified by the sender. Examples of this kind of applications include Internet video or audio.

(2) The second category is the interactive Internet applications such as TELNET. Normally, these kind of applications do not need much bandwidth, but cannot tolerate long delay. So it should not get dropped while being transferred from the source to the destination. Dropping a packet may cause the sender to spend a Round Trip Time (RTT) to send that packet to the destination again. It is hard to specify a traffic rate for this kind of flows. It sends a packet or a burst of packets intermittently.

(3) The third category is the popular TCP-based applications such as web browsing or FTP. The users of these applications mainly care about the overall throughput instead of individual packet delay. It is also hard to specify a transfer rate for this kind of flow. If the network is relatively light-loaded, the application can get high bandwidth. However if the network is relatively heavily loaded, it may be acceptable to get a moderate bandwidth.

(4) The last category is the non-urgent applications such as EMAIL. These kinds of applications are not sensitive about how long the data takes to reach the destination. Generally, it doesn't matter whether the message is sent to the destination immediately or several seconds later or even several minutes later. So it should not compete for the assured bandwidth with other applications. Specifying an expected bandwidth for these applications is meaningless since they do not care about it.

3.2.1 Marking Model in the Edge Router

In order to serve different applications efficiently, the edge router should treat each category in a different way. In order to process both the delay sensitive and throughput sensitive applications, the core router should support a premium service and an assured service. The core router structure is shown in Figure 2. The premium service and the assured service have been described in detail in [6]. The premium packets will always be forwarded with almost no queuing delay and the assured IN packets rarely get dropped even in case of network congestion. The untagged OUT packets get dropped in case of network congestion. The aggregate premium traffic should not exceed a certain percentage of the total bandwidth so that the assured and best-effort traffic do not get starved. We propose that applications belonging to category 1 should be marked as the premium service. Applications belonging to category 2 should only be marked as assured IN. Applications belonging to category 3 should use the remaining assured service and could also compete for the best-effort service. Applications belonging to category 4 should only use the best-effort service. The marking model in the edge router is shown in Figure 3.

In Figure 3, C1 to C4 denote applications belonging to category 1 to category 4 respectively. C1 uses the premium service. C2 uses assured service. C3 uses both assured and best-effort services. C4 only uses best-effort service.

For the premium service, we should implement an admission control in the edge router. Each flow should submit a peak rate to the edge router. If the flow is admitted, the edge router will keep a queue and a leaky bucket for that flow. In Figure 3, q1, q2, ..., q_n are the queues for the premium service micro-flows. The depth for the leaky bucket should be just 1 or 2 packets to avoid bursts. This can ensure that each flow will not exceed the submitted peak rate. The packets from each queue will aggregate in q(2). A leaky bucket called *bucket 2* is located here to reshape the aggregated flow. The leak rate of this bucket is the negotiated premium service peak rate of the local domain. The depth of the leaky bucket is also 1 to 2 packets. The queue length of q_1 to q_n is limited by the tolerable local delay divided by the peak rate. In our model, it is set to a small number. A packet that has captured a token from bucket 2 gets its premium service bit marked and forwarded to the next hop.

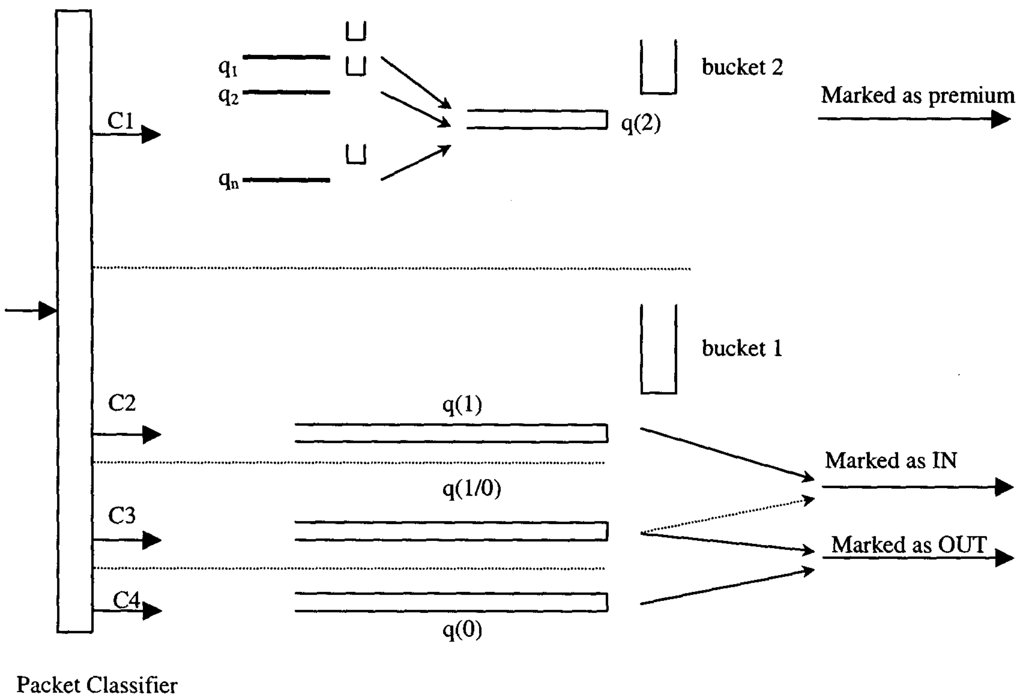

Figure 3: Application-Based marking model.

The assured bandwidth is shared by the applications belonging to category 2 and category 3. Normally applications from category 2 do not consume a lot of bandwidth. So we can assume that it will not exceed the negotiated assured bandwidth. We queue all of the flows of C2 in q(1). Packets in q(1) could only be marked as assured service and forwarded. This ensures that all of the packets will rarely get dropped when they are forwarded in the core routers. Flows belonging to C3 are queued in q(1/0). Packets in q(1/0) will get their assured service bit marked IN if there are tokens available in bucket 1 and q(1) is empty, otherwise, the packet will be marked as OUT. Thus, the flows belonging to C2 will have higher priority to be marked as assured service. Flows belonging to C3 will try to use the remaining negotiated assured bandwidth. However, it could also compete for the best-effort service. The leaking rate of *bucket 1* corresponds to the negotiated assured bandwidth of the local domain. The depth of the leaky bucket is the burst size of the assured service.

Applications belonging to category 4 will not compete for the assured service. They are queued in q(0) and will be marked as OUT and forwarded. So they have the lowest priority and only consume the remaining bandwidth.

3.2.2 Communication Protocol between Applications and the Edge Router

The other implementation issue is how the applications can talk to the edge router to specify their bandwidth and delay request. Applications belonging to category 1 generally will last for a relatively long time once set up. Before setting up the connection, the application could send a signaling message to the edge router to specify the peak rate. The edge router will check the remaining premium service bandwidth and decide whether it should be admitted or rejected. This information will be sent back to the application. If the flow is admitted the flow information such as the source and destination IP and port number, the peak rate, will be kept in the premium service flow table in the edge router. After the flow gets finished, the application will send a signaling message to the edge router to inform the edge router to delete it from the flow table. This process could use the RSVP protocol. Since the connection will last a relatively long period, the overhead of the signaling message is negligible.

Similarly, we could also keep a table for flows in C2 and C3. All the other flows will be deemed as C4. However, most of the TCP flows are relatively short, thus sending the signaling message to the edge router may be a considerable overhead. Also, the edge router needs to spend time to update the tables. In order to save the overhead, the edge router could use the default port number to identify a flow. Most of the current Internet applications are based on the Client/Server model. Generally, the server has a well-known port number which is between 0 to 1023. For example, TELNET uses 23, EMAIL uses 25, HTTP uses 80. The edge router could check the source or destination port number of the packet to decide which category the packet should belong to. Only those applications that do not have or do not use the well-known port numbers should use the signaling message to notify the edge router to add it to the flow table. This, of course, will greatly reduce the overhead of setting up a connection. The other advantage of using default port number to classify packets is that the currently existing Internet applications need not support a signaling protocol before they could join the DiffServ. This will make DiffServ easier to implement in the current Internet architecture.

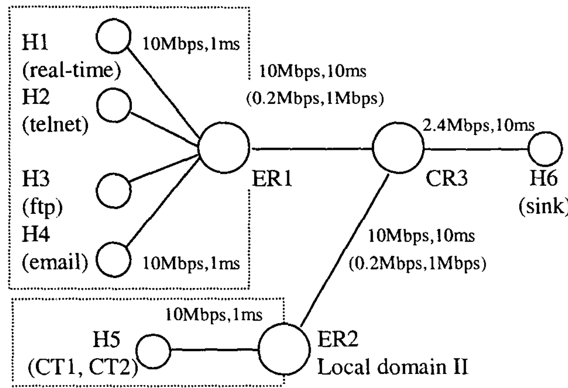

Figure 4: Simulation Topology.

4 Performance Study

We modified *ns2* simulator to study the performance of our model. The simulation topology is shown in Figure 4. ER1 and ER2 are edge routers for two local domains, and CR3 is the core router. Local domain I includes host H1, H2, H3 and H4. Each of them is connected to ER1 with a 10Mbps bandwidth, 1ms delay link. Assume that local domain I negotiated 200kbps for premium service and 1Mbps for assured service. Host H6 is connected to CR3 with a 2.4Mbps, 10ms link, which is also the bottleneck of the network. *Real-time, telnet, ftp* and *email* are four sources representing applications from Category 1, Category 2, Category 3, and Category 4 respectively. We make them coexist in a local domain to see how they will share the negotiated bandwidth. *CT1* and *CT2* are two additional sources that are used to insert a bursty traffic to the network. The detail of each source is described here:

(1) *Real-time* is a constant bit rate (CBR) UDP traffic with a rate of 150kbps. It is used to represent a real-time source of category 1.
(2) *Telnet* is used to represent a typical application of category 2. The size of each data block and the idle interval between neighboring blocks are exponentially distributed. The average block size is 10kb and the average time interval between two neighboring blocks is 0.1 second.
(3) *Ftp* is used to represent a typical application of category 3. It is based on the TCP transport protocol.
(4) *Email* is used to represent a typical category 4 application. It is also based on the TCP transport protocol.
(5) *CT1* is an Exponential On/Off UDP real-time source with 150kbps peak rate. The average on and off interval is 0.5s. It is used to represent a bursty category 1 traffic.
(6) *CT2* is an Exponential On/Off UDP source of category 3, with 800kbps peak rate. The average on and off interval is also 0.5s. It is used to represent a bursty category 3 traffic.

The queuing model shown in Figure 3 is implemented in the two edge routers ER1 and ER2. The queuing model shown in Figure 2 is implemented in the core router CR3. For the RIO implementation, the RED has values of 40 packets, 60 packets and 0.02 for min_in, max_in, and Pmax_in respectively, and 15 packets, 30 packets and 0.5 for min_out, max_out, and Pmax_out respectively[9] where min_in and max_in represent the upper and lower bounds for the average queue size for IN packets and Pmax_in is the maximum drop probability for an IN packet when the queue size is in the [min_in, max_in] range. The min_out, max_out and Pmax_out are the corresponding parameters for the OUT packets.

4.1 Performance in Absence of Cross Traffic

In this simulation, cross traffic CT1 and CT2 are not added. The bandwidth distribution among the four applications is shown in Figure 5. Although the network traffic is busty, *Real-time* gets a steady 150kbps bandwidth. *Telnet* only consumes a small amount of the negotiated assured bandwidth. *Ftp* consumes most of the negotiated assured bandwidth and also compete for the remaining best-effort bandwidth, so it got a high overall bandwidth. *Email* only competes for the remaining best-effort bandwidth, so it gets a much lower overall bandwidth compared to *ftp*.

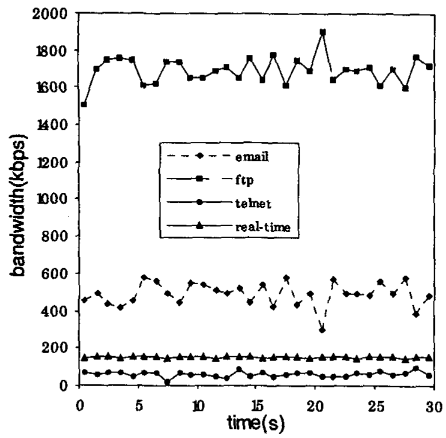

Figure 5: Bandwidth Distribution: Application-Based DiffServ Model, No Cross Traffic.

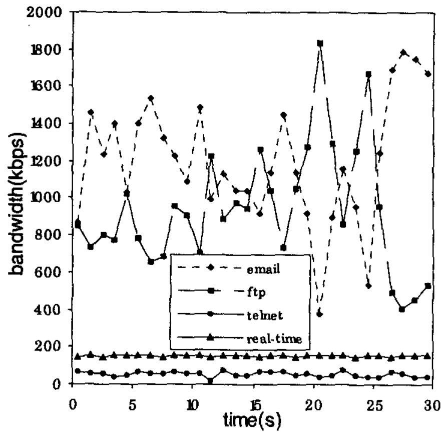

Figure 6: Bandwidth Distribution: Application-Independent Diffserv Model, No Cross Traffic.

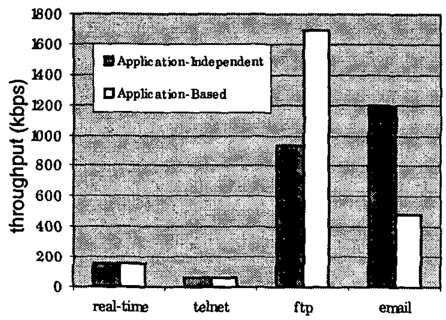

Figure 7: Throughput Comparison of each application: Application-Based vs. Application-Independent Model, No Cross Traffic.

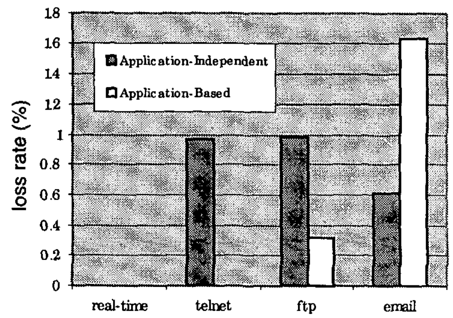

Figure 8: Loss Rate Comparison of each application: Application-Based vs. Application-Independent Model, No Cross Traffic.

We perform a comparative simulation study using the Application-Independent model. We use the same simulation parameters and source models shown in Figure 4. The only difference is, the marker in the Edge Router does not consider different bandwidth and delay requirements of *telnet, ftp* and *email.* They are simply deemed as applications belonging to the same category. If the aggregate bandwidth does not exceed the negotiated assured bandwidth, the packet is marked as IN, otherwise, it is marked as OUT.

The simulation result is shown in Figure 6. According to the simulation result, *real-time*, which uses the premium service, still gets a steady 150kbps bandwidth because it is treated differently even in the application-independent marking model. *Telnet* gets bandwidth similar to the previous experiment because it retransmits those dropped packets during the idle period between two bursts. However, *ftp* and *email* are treated equally now. *Ftp* could not get higher transfer priority over *email* any more.

Figure 7 shows the throughput comparison of each application using the two marking models. The total throughput of the four flows in Application-Independent model is 2.34Mbps. The total throughput of the four flows in Application-Based model is 2.39Mbps. In the application-based marking model, *ftp* gets much higher throughput than *email.* The loss rate comparison under these two marking models is shown in Figure 8. In both cases, the loss rate of *real-time* is 0 because it does not exceed the specified peak rate. In application-independent marking model, the loss rates of *telnet, ftp* and *email* are in the same order. However, in the application-based marking model, loss rate of *telnet* is 0, loss rate of *ftp* is much lower than loss rate of *email.* So, *telnet* gets

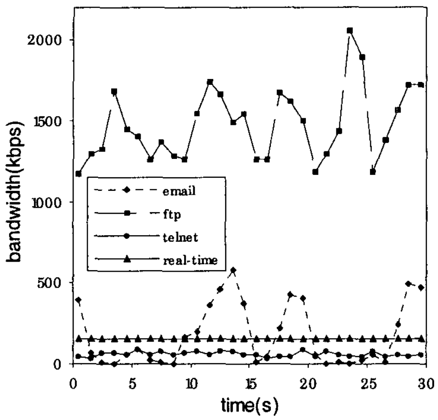

Figure 9: Bandwidth Distribution: Application-Based DiffServ Model, With Cross Traffic.

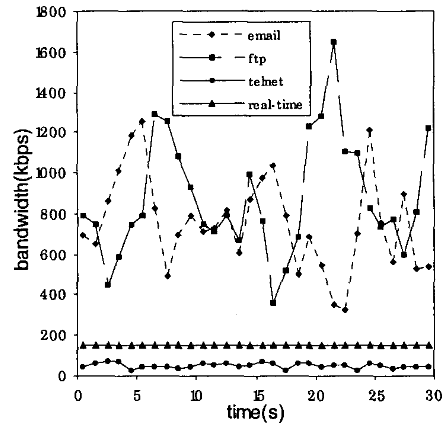

Figure 10: Bandwidth Distribution: Application-Independent Diffserv Model, With Cross Traffic.

smoother responses when using the application-based marking model.

4.2 Performance in Presence of Cross Traffic

We repeat the comparison study in a more realistic network environment with two cross traffic sources *CT1* and *CT2*. The overall average rate is about 75kbps for *CT1* and 400kbps for *CT2*. So the four applications from local domain I could use up to 1.925Mbps. According to the SLA between ER2 and CR3 (200kbps premium and 1Mbps assured), all the packets from *CT1* will be marked as premium and all the packets from *CT2* will be marked as IN.

The bandwidth distributions of both marking models are shown in Figures 9 and 10, respectively. Compared to the former simulation results, the bandwidth distributions for *real-time* and *telnet* are not affected. *Ftp* and *email* are more bursty now because some of the packets from *ftp* and all of the packets from *email* are marked as OUT. They will yield the bandwidth when the burst of the cross traffic comes. However, in the application-based marking model, *ftp* could still get a bandwidth higher than 1Mbps at any time because about 0.95Mbps traffic of *ftp* are marked as IN and is assured.

5 Conclusion

The differentiated service model tries to provide a certain degree of bandwidth or delay guarantees to end-users. However, different applications have different bandwidth and delay requirements. In this paper we suggest that we should involve applications in the differentiated service process directly in order to make use of the negotiated bandwidth efficiently. We also suggested different policies for applications in different categories to indicate their requirements. Based on these considerations, we proposed an application-based differentiated service model, which provides performance matching to the needs of the applications.

References:

[1] http://www-mash.cs.berkeley.edu/ns/

[2] P. Ferguson and G.Huston, *"Quality of Service,"* John Wiley & Sons, 1998.

[3] D.Clark, "Adding Service Discrimination to the Internet," *Tech. Report, MIT Laboratory of Computer Science,* September 1995.

[4] D.Clark, W. Fang, "Explicit Allocation of Best Effort Packet Delivery Service," *Tech. Report, MIT Laboratory of Computer Science,* 1998.

[5] V. Jacobson, "Differentiated Services Architecture," *Talk in the Int-Serv WG at the Munich IETF,* August 1997.

[6] K. Nichols, V. Jacobson and L. Zhang, "A Two-bit Differentiated Services Architecture for the Internet," *Internet Draft,* November, 1997. Available at diffserv.lcs.mit.edu.

[7] S. Floyd, V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Transactions on Networking*, August 1993, pp397-413.

[8] L. Zhang, S. Deering, D. Estrin, S. Shenker, D. Zappala, "RSVP: A New Resource Reservation Protocol." *IEEE Networks*, September 1993.