Adaptive-Weighted Packet Scheduling for Premium Service

Haining Wang† Chia Shen‡ Kang G. Shin†

†The University of Michigan
Ann Arbor, MI 48109
{hxw,kgshin}@eecs.umich.edu

‡Mitsubishi Electric Research Laboratory
Cambridge, MA 02139
shen@merl.com

Abstract—**This paper presents a new scheduling scheme to support premium service in the Differentiated Service (DiffServ) architecture. It is based on weighted packet scheduling policies such as weighted round robin or fair queueing. The key feature of the new scheduling scheme is to change the scheduling weights of Behavior Aggregates adaptively. By adaptively adjusting the weights according to the dynamics of the average queue size of premium service, the proposed scheme can achieve low loss rate, low delay and delay jitter for the premium service. Moreover, it requires neither rigid admission control nor accurate traffic conditioning to support premium service in the DiffServ architecture. This adaptive packet scheduling is shown to absorb the transient burstiness of the Expedited Forwarding (EF) aggregate — which is caused by the traffic distortion inside the network — without incurring packet loss or increasing the queueing delay.**

I. INTRODUCTION

Differentiated Service (DiffServ) [1], [2] has been proposed as a scalable method for providing the Quality of Service (QoS) over IP networks. In the DiffServ architecture, per-flow states and signalling are not required at core routers; traffic conditioning and per-flow management are done at edge routers only. Based on the DS field in the IP header, IP flows are classified into different aggregates, and services are provided for aggregates, instead of individual flows, and defined by a small set of Per-Hop Behaviors (PHBs). PHBs are the forwarding behaviors applied to aggregates at core routers.

Currently, three types of PHBs are specified in the DiffServ architecture: Expedited Forwarding (EF) PHB [7], Assured Forwarding (AF) PHB [8] and Best-Effort PHB. EF is to support premium service [10] in the DiffServ, which has been proposed as a virtual leased line. Providing low loss rate, low delay, low delay jitter and an assured throughput is the main goal of premium service. AF only provides low loss rate without any guarantee on delay and delay jitter.

To implement premium service in IP networks, the packet scheduler at a router must meet the EF goals. Among the various proposed packet scheduling schemes, priority queueing and weighted round robin have attracted a great deal of attention as the means of realizing EF due mainly to their simplicity. They have been evaluated by simulation experiments [7]. The simulation results show that priority queueing can provide lower delay and lower delay jitter for an EF flow than weighted round robin. This is expected, since with a priority scheduler the priority queue is always serviced before any other queue to guarantee timely delivery of packets. However, priority queueing can cause greater burstiness since the EF packets do not get interleaved with any other packets that belong to a different behavior aggregate (BA).[1] The aggregation of EF flows leads to the cluster of EF packets, and the EF burstiness increases with the number of EF flows aggregated at core routers. The side effects of priority queueing could cause the EF packet arrival rate to exceed the reserved service rate at core routers, thereby resulting in packet losses. Recent work has confirmed that priority queueing leads to increased burstiness and bursty packet loss [4].

[1] A behavior aggregate is a set of packets with the same DS field in a forwarding path.

The weighted round robin (or weighted fair queueing [3]) scheduling does not have such drawbacks, but the traffic distortion inside the network and the dynamic flow aggregation make it difficult to use static weights at routers. To provide no (or very small) queueing delays, the premium service requires that at every transit node the EF aggregate's maximum arrival rate should always be less than the aggregate's minimum departure rate. There are two prerequisites to meet this requirement: (1) the EF aggregate has a well-defined minimum departure rate, which is independent of the dynamic state of the router; and (2) the EF aggregate is conditioned, which includes policing and shaping, to ensure that its arrival rate at any router is less than the router's configured minimum departure rate.

Unfortunately, traffic conditioning is only performed at edge routers. Traffic distortion inside the network such as packet clustering could violate the promised traffic specification. Furthermore, in each router the number of flows in the EF aggregate changes with the addition or departure of an individual EF flow, and hence the minimum departure rate for the EF aggregate should be dynamically adjusted to reflect the change of traffic profile. Without the support of rigid admission control and accurate traffic conditioning, the static setting of weights could cause bursty packet losses.

In this paper, we propose an adaptive-weighted packet scheduling scheme to support delay-sensitive and loss-sensitive traffic in the DiffServ architecture, which can apply to weighted round robin and weighted fair queueing. The proposed scheme not only guarantees low loss rate but also achieves low queueing delay and delay jitter for EF flows. A slightly larger buffer space for EF aggregates is used to absorb the burstiness caused by traffic distortion inside the network, and reduces the loss rate of EF aggregates. However, a larger buffer space could cause longer queueing delay and larger delay jitter to EF packets, which should be avoided. To solve this problem, we use EWMA (Exponentially Weighted Moving Average) to estimate the average queue size of premium service. By adaptively adjusting the weights, we keep the average queue size small, guaranteeing a small average queueing delay. Also, we use a low-pass filter to estimate the average queue size, which makes the instantaneous queue size slightly fluctuate with time, resulting in a small delay jitter.

Although the deployment of bandwidth broker [11] could make dynamic resource provision a possibility, and the traffic conditioning at edge routers shapes the incoming traffic as their traffic specification, there are still many factors that could cause traffic distortion inside the network:

- the transient effect caused by the dynamic flow aggregation;
- inaccurate traffic shaping at edge routers, and no traffic conditioning at core routers;
- packet clustering caused by cascaded queueing effects; and
- the path changes caused by route flip.

It is therefore important to make the packet scheduler at a core router adaptive to absorb the traffic distortion inside the network. The performance of the proposed scheme is evaluated by simulation. The simulation results have shown the proposed scheme to reduce the loss rate significantly without degrading the delay and delay jitter.

The rest of this paper is organized as follows. Section 2 briefly reviews the background and related work. The proposed scheduling scheme is detailed in Section 3. Section 4 presents the performance evaluation of the proposed scheme. Finally, Section 5 concludes the paper.

II. Background and Related Work

To support end-to-end QoS in the Internet, the IETF has defined two major architectures for augmenting the single-class best-effort service: Integrated Services (IntServ) [12] and Differentiated Services (DiffServ). In the network data plane of the IntServ architecture, scheduling schemes such as Weighted Fair Queueing (WFQ) [3], Virtual Clock (VC) [17] and Rate-Controlled Earliest Deadline First (RC-EDF) [15] have been proposed to support guaranteed QoS. In the control plane, a signaling protocol RSVP [16] is required for admission control and resource reservation. While IntServ provides QoS guarantees, it requires per-flow management at core routers, which places an unbearable burden on core routers. Due to its poor scalability of the IntServ architecture, DiffServ has been proposed as an alternative.

In the network data plane of the DiffServ architecture, the need for per-flow state management at core routers has been eliminated. A core router implements a simple scheduling and buffering mechanism to serve the aggregated flows based on the DS field in the IP header. By pushing the complexity to the edge routers, DiffServ's data plane is much simpler and hence more scalable than IntServ. While DiffServ is more scalable, it still requires the support of admission control, resource provisioning, and service-level agreement on the control plane. A novel bandwidth broker architecture [18] has been proposed for admission control and resource provisioning in each network domain, which decouples QoS control from core routers. Core routers do not maintain any reservation state; all reservation states are stored in, and managed by, bandwidth brokers.

For packet scheduling in the data plane, a number of mechanisms are available to implement coarse-grain QoS support. Besides priority queueing and a weighted round robin scheduler, Class-Based Queueing (CBQ) [6] can be implemented to meet the requirements of forwarding behaviors in the DiffServ architecture, in which the EF packets are given priority up to the configured EF rate.

III. The Proposed Scheme

To deal with the traffic distortion and dynamics of flow aggregation, we propose an adaptive-weighted packet scheduling scheme, which can be applied to weighted round robin or fair queueing. The features of adaptive-weighted scheme include:

- A slightly larger buffer space for premium service is used to accommodate transient bursts. In the current IETF proposals, the buffer space for premium service can only contain 1 or 2 packets in order to achieve low delay and low delay jitter;
- Exponential weight moving average (EWMA) is employed to estimate the average queue size of premium service, which is the index used for calibrating the weights;
- The weight of premium service is adaptively adjusted, according to the dynamics of average queue size. However, there is an upper limit by which the weight of premium service should be bounded; and
- By maintaining a very small average queue size, low queueing delay is achieved. Also, a low-pass filter is used to reduce the fluctuation of instantaneous queue size, achieving low delay jitter.

To provide different packet-forwarding services, in the DiffServ architecture each behavior aggregate has its own buffer space at core routers, instead of a common shared buffer. The "queue size" mentioned in this paper refers to the queue for premium service. In the following subsection, the proposed scheme is detailed.

A. *Adaptive Weight Calibration*

As with Random Early Detection (RED) [5], we employ the estimated average queue size of premium service as the index to adaptively adjust the weights. The average queue size of premium service is calculated by using a low-pass filter with an exponential weighted moving average. Assuming avg is the average queue size, q is the instantaneous queue size and f_l is the low-pass filter, the average queue size of premium service is estimated as:

$$avg \leftarrow (1 - f_l) \cdot avg + f_l \cdot q$$

To reduce the fluctuation of instantaneous queue size, the low-pass filter f_l is set to 0.01 in the proposed scheme, which results in a low delay jitter.

To adaptively calibrate the weight of premium service, two thresholds (minimum and maximum) are introduced. The minimum threshold represents the desired queueing delay, and the maximum threshold represents the acceptable queueing delay. By keeping average queue size below the maximum threshold, a low queueing delay is achieved. To accomplish this, the weight of premium service should be proportionally increased once the average queue size exceeds the minimum threshold. However, the weight of premium service cannot exceed an upper limit after the average queue size reaches max_{th}; otherwise, the proposed scheme would temporarily degrade to priority queueing and lead to packet clustering.

In our proposed scheme, there is a linear relationship between the weight of premium service and the average queue size. Assume the original weight of premium service is w_p, then the weight function of premium service is given by:

$$f(avg) = \begin{cases} w_p, & avg \in [0, 0.5) \\ \frac{(upper - w_p)\cdot(avg - min_{th})}{max_{th} - min_{th}} + w_p, & avg \in [0.5, 2) \\ upper, & avg \in [2, full] \end{cases}$$

where the $upper$ is the upper limit that the weight of premium service can reach, and avg is the average queue size of premium service. If the total weight is 1, then $EF_w + AF_w + BE_w = 1$, where EF_w is the weight of premium service, AF_w is the weight of assured service and BE_w is the weight of best-effort. We suggest the upper limit of EF_w to be set to 0.7, and the rest of weight to be used by assured-forwarding (AF) and best-effort services.

Since the total weight for a shared link is fixed, the increase of premium service's weight must cause the same amount of decrease in the best-effort's weight or AF's weight. The rule we applied here is: first shift the weight of best-effort to premium service, and if this is not enough and the weight of premium service has not reached its upper limit, then part of AF's weight will be shifted to premium service. However, once the average queue size of premium service backs down below max_{th}, the weights taken from best-effort or AF will be returned.

To meet the requirement of no or a very small queueing delay of premium service, we set the minimum threshold to 0.5 and the maximum threshold to 2, measured in packets instead of bytes. Figure 1 illustrates the dynamics of the weights calibration, in which the initial weights are 0.3, 0.3 and 0.4 for premium service, AF, and best-effort, respectively. Since the upper limit for premium service is 0.7, no need to shift the weight from assured service to premium service in this case.

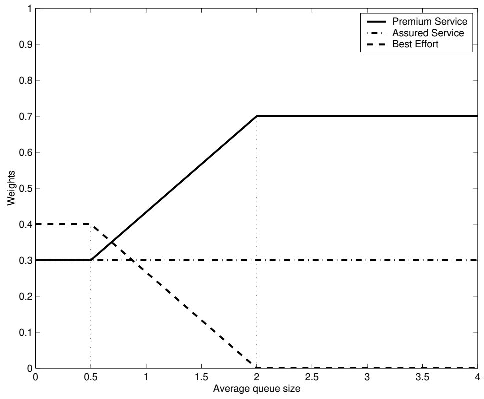

Fig. 1. The Dynamics of Weights

IV. Performance Evaluation

Simulation is used to evaluate the proposed scheduling scheme. To characterize the EF behavior, three QoS metrics are included: packet loss rate, one-way end-to-end delay, and one way end-to-end delay jitter. The definition of delay jitter follows the one given in [7], which is based on the one-way end-to-end delay and defined as the absolute difference between the delays of two consecutive packets. Assume D_i is the one-way end-to-end delay of the i_{th} packet, then the one-way end-to-end jitter is given as:

$$Jitter = |D_{i+1} - D_i|$$

To evaluate the effect of weight changes on assured and best-effort services, we measure effective throughput, a.k.a. goodput, which does not include dropped or duplicate data packets.

A. Simulation Setup

Our simulations are done in ns-2 [14] with DiffServ additions [9]. A relatively simple, yet sufficiently representative simulation topology is used, which is shown in Figure 2. All nodes are in a single DS domain. Each end-host is connected to its respective edge router, which does per-flow traffic shaping and conditioning. The edge routers are connected via two core routers. The link capacity and the one-way propagation delay between an end-host and an edge router are 10 Mbps and 1 ms, respectively. However, the bandwidth and the link delay between routers [2] are set to 3 Mbps and 10 ms, respectively.

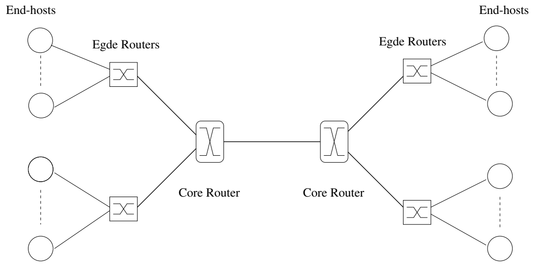

Fig. 2. The network topology used for simulation experiments

The packet size is set to 256 bytes since the average packet size measured on WAN links is reported to be about 250 bytes [13]. In all simulation experiments the packet size is fixed, and hence, the comparison between adaptive and static weights can also be applied to weighted fair queueing, although weighted round robin is employed in our simulation experiments. The buffer space in our simulation is measured in number of packets. For AF and best-effort services, their buffer sizes at routers are set to 100. For premium service, according to the recommendations in the IETF proposals, the buffer size at routers is set to 2 in both cases of static weighted round robin and priority queueing. However, in the adaptive-weighted scheme, a slightly larger buffer — which is set to 6 in our simulation — is used for premium service.

TABLE I
Initial Weight Setting

	Edge Routers	Core Routers
Premium Service	0.1667	0.3334
Assured Service	0.3333	0.3666
Best-Effort	0.5	0.3

The traffic type in our simulation is UDP. The background traffic includes AF and best-effort aggregates, whose source transmission rates are 1 Mbps and 2 Mbps, respectively. They are kept unchanged for all simulation experiments. For EF aggregates, the minimum packet inter-arrival time is varied for different simulation experiments, categorizing the simulation into

[2] It does not matter if it is an edge or core router.

different scenarios. The initial weight settings at edge routers and core routers are listed in Table I.

B. Simulation Results and Analysis

We now present the results obtained from the different simulation scenarios. According to the minimum packet inter-arrival time of an EF flow, three simulation scenarios are tested: under-provisioning, on-provisioning and over-provisioning.

Under-provisioning: is mainly caused by the lack of rigid admission control and the dynamic flow aggregation. In this case, the minimum packet inter-arrival time is set to 3.5 msec.

On-provisioning: rigid or dynamic admission control is assumed so that the effect of dynamic flow aggregation has been eliminated. Only traffic distortion inside the network caused by packet clustering is simulated, and the minimum packet inter-arrival time is set to 4 msec.

Over-provisioning: the resources at routers are over-booked for premium service. The minimum packet inter-arrival time is set to 4.5 msec in this scenario.

The goal of our simulation is to evaluate the adaptive weighted round robin in terms of packet loss rate, delay and delay jitter, and compare its performance with those achieved by using static weighted round robin, and priority queueing in these simulation scenarios.

Figure 3 illustrates the packet loss rate of EF aggregate, showing that the proposed scheme achieves no packet loss in all simulation scenarios. In contrast, the static weighted round robin and priority queueing have unacceptably high packet-loss rates in case of under-provisioning, and experience packet loss in case of on-provisioning.

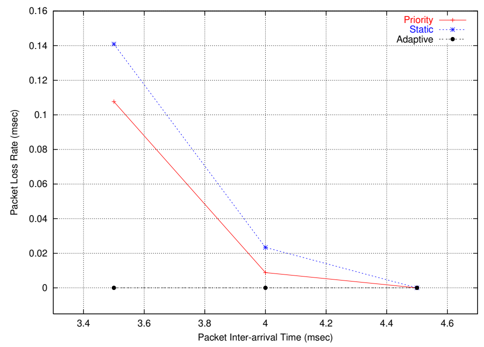

Fig. 3. Packet-loss rate

In comparison with static weighted round robin and priority queueing, the adaptive-weighted scheme significantly reduces the packet-loss rate of premium service. However, this is due partly to the deployment of a larger buffer for premium service. So, it is very important that this reduction of packet loss should not be at the expense of longer end-to-end delay and larger delay jitter.

The average one-way end-to-end delay experienced by EF packets is plotted in Figure 4. As expected, priority queueing has the smallest average end-to-end delay. However, as compared with static weighted round robin, the proposed scheme does not cause a longer delay even though it uses a larger buffer for the EF aggregate at routers. In the case of on-provisioning, the adaptive-weighted scheme even achieves a slightly lower end-to-end delay than static weighted round robin.

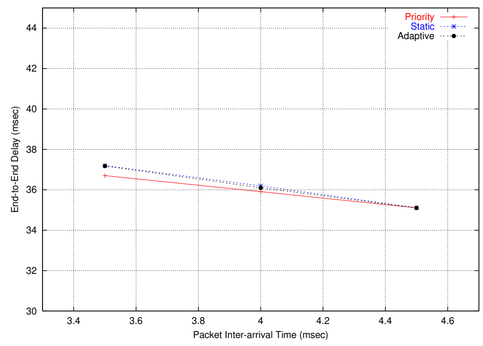

Fig. 4. Average end-to-end delay

For real-time audio/video applications, delay jitter is the key metric that affects the quality of service. To illustrate the delay jitter of different schedulers, the cumulative distribution of end-to-end delay jitter experienced by the EF packets is plotted for each simulation scenario. Figures 5 and 6 plot the one-way delay jitter in the under-provisioning and on-provisioning cases, respectively. The proposed scheme also achieves a smaller delay jitter than static weighted round robin in both cases. Figure 7 plots the delay jitter in the over-provisioning case, where the proposed scheme and the static weighted round robin provide similar delay variations due mainly to less demanding traffic sources.

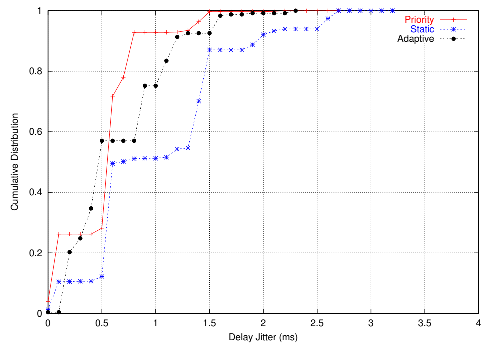

Fig. 5. Delay jitter in under-provisioning scenario

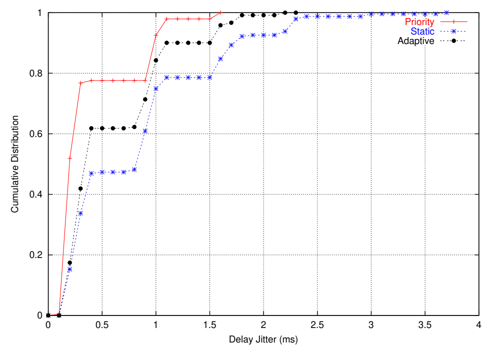

Fig. 6. Delay jitter in on-provisioning scenario

We conclude that in the over-provisioning case, there is no performance difference between the adaptive-weighted scheme and the static one. However, in the on-provisioning and under-provisioning cases, the adaptive-weighted scheme significantly reduces the packet-loss rate without enlarging the end-to-end delay. More importantly, it achieves a smaller delay jitter than the static-weighted scheme.

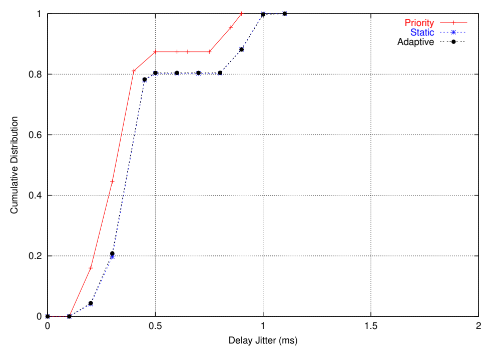

Fig. 7. Delay jitter in over-provisioning scenario

Now, we evaluate the side-effect of the proposed scheme on AF and best-effort services, since the weights of AF and best-effort services are reduced by increasing the premium service's weight. Here we deal only with the effective throughput since AF service does not give any bound on end-to-end delay and delay jitter, and best-effort service does not provide any guarantee at all. Figure 8 shows the goodput of AF service, in which the proposed scheme does a better job than priority queueing as expected. Figure 9 plots the goodput of best-effort service. Unsurprisingly, the proposed scheme provides a lower goodput for best-effort service in the cases of under-provisioning and on-provision, since its weight has been frequently shifted to premium service according to the dynamics of average queue size of premium service. Especially in the under-provisioning case, because the remaining weight is mostly taken by assured service, the proposed scheme has the lowest goodput for best-effort. However, since best-effort provides no guarantee to service, we believe that this trade-off is the right choice.

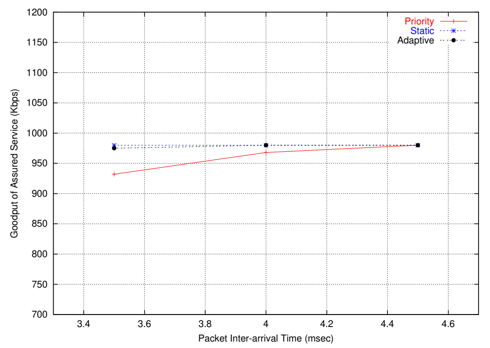

Fig. 8. Goodput of AF service

V. Conclusion

We proposed an adaptive-weighted scheduling scheme for supporting premium service in which the scheduling weights of behavior aggregates are adaptively changed with the dynamics of average queue size of premium service. It is able to absorb the traffic distortion inside the network without degrading delay or delay jitter. Moreover, it makes rigid admission control and accurate traffic conditioning not imperative requirements for supporting premium service in the DiffServ architecture. Our simulation results show that the proposed scheme can achieve low loss rate, low delay and low delay jitter for the premium service.

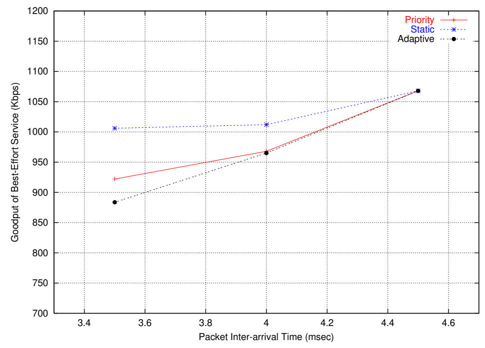

Fig. 9. Goodput of best-effort service

References

[1] Y. Bernet *et al.*, "A Framework for Differentiated Services", *IETF Internet Draft*, February, 1999.

[2] S. Blake *et al.*, "An Architecture for Differentiated Services", *RFC 2475*, December 1998.

[3] A. Demers, S. Keshav, and S. Shenker, "Analysis and Simulation of a Fair Queueing Algorithm", *Proceedings of ACM SIGCOMM'89*, September, 1989.

[4] T. Ferrari and P. F. Chimento, "A Measurement-based Analysis of Expedited Forwarding PHB Mechanisms", *Proceedings of IWQoS'2000*, Pittsburgh, June 2000.

[5] S. Floyd and V. Jacobson, "Random Early Detection gateways for Congestion Avoidance", *IEEE/ACM Transactions on Networking*, Vol. 1, No. 4, August 1993.

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

[7] V. Jacobson, K. Nichols, and K. Poduri, "An Expedited Forwarding PHB", *Internet Draft*, June 1999.

[8] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, "Assured Forwarding PHB Group", *RFC 2597*, June, 1999.

[9] S. Murphy, "DiffServ Additions to ns-2", May 2000, http://www.teltec.duc.ie/ murphys/ns-work/diffserv.

[10] K. Nichols, V. Jacobson, and L. Zhang, "An Approach to Service Allocation in the Internet", *Internet Draft*, November 1997.

[11] K. Nichols, V. Jacobson, and L. Zhang, "A Two-bit Differentiated Services Architecture for the Internet", *RFC 2638*, July 1999.

[12] S. Shenker, C. Patridge, and R. Guerin, "Specification of Guaranteed Quality of Service", *RFC 2212*, September 1996.

[13] K. Thompson, G. J. Miller, and R. Wilder, "Wide-Area Internet Traffic Patterns and Characteristics", *IEEE Network*, Vol. 11, No. 6, pp. 10-23, November/December 1997.

[14] UCB/LBNL/VINT, "Network Simulator", *ns-2*, 1997. http://www.isi.edu/nsnam/ns/

[15] H. Zhang and D. Ferrari, "Rate-controlled Static-priority Queueing", *Proceedings of IEEE INFOCOM'93*, April, 1993.

[16] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala, "RSVP: A New Resource ReSerVation Protocol", *IEEE Network*, 7(5):8-18, September 1993.

[17] L. Zhang, "Virtual Clock: A New Traffic Control Algorithm for Packet Switching Networks", *Proceedings of ACM SIGCOMM'90*, September, 1990.

[18] Z. Zhang, Z. Duan, L. Gao, and Y. T. Hou, "Decoupling QoS Control from Core Routers: A Novel Bandwidth Broker Architecture for Scalable Support of Guaranteed Services" *Proceedings of ACM SIGCOMM'2000*, September, 2000