Using Edge-To-Edge Feedback Control to Make Assured Service More Assured in DiffServ Networks

K.R.R.Kumar, A.L.Ananda, Lillykutty Jacob
Centre for Internet Research, School of Computing
National University of Singapore
{kaleelaz, ananda, jacobl}@comp.nus.edu.sg

Abstract

Markers, one of the building blocks of a traffic conditioner play a major role for resource allocation in a DiffServ network. Recently, there has been a considerable research interest in designing intelligent markers, tailored for TCP traffic. The transient congestion levels in the network can affect the TCP throughput adversely and have great impact on the effectiveness of the marker. To address this issue, we propose an intelligent marker, which relies on an ECN-like feedback control mechanism between ingress and egress traffic conditioners. We present the control architecture and the marker algorithm, and illustrate the benefits of the proposed marker in a DiffServ network. The marker was implemented in NS simulator and extensive simulations were done to study its effectiveness in the assurance of throughput, especially under the case that AS flows require different target rates, and co-exist with the unruly UDP flows. The ECN-like feedback mechanism that we use for the marker is also suitable in an MPLS domain for providing signaling to the ingress router of a congested label switched path (LSP) to mitigate congestion via dynamic traffic engineering techniques.

Keywords: *QoS, Congestion feedback, Assured services, TCP friendliness, Traffic conditioner, DiffServ networks*

1. Introduction

The exponential growth in the amount of traffic in the Internet has resulted in the deterioration of the quality of service (QoS) available. Although, over provisioning of the networks with more bandwidth and better switching capability could be one of the ways to provide assured services, an intelligent network service with better resource allocation and management methods has been widely accepted to be a more cost effective and efficient alternative.

The Differentiated Services architecture [2], proposed by the IETF DiffServ Working Group has been one of the widely accepted service models mainly because of its simplicity and scalability. DiffServ standardization efforts are on for two of the *per-hop behaviors (PHB)*, namely Expedited Forwarding [EF] [11] and Assured Forwarding [AF] [14]. While the former PHB provides a deterministic QoS, the latter is meant for statistical QoS. DiffServ network, accordingly, supports two important services called *Premium* and *Assured* beyond the current Internet's *Best Effort* service. The class of Assured Services (AS) is intended to give the customer the assurance of a minimum throughput, called the *target rate*, even during periods of congestion, while allowing it to consume, in some fair manner, the remaining bandwidth when the network load is low.

The AS architecture relies on packet marking mechanism, performed by the *Traffic Conditioner (TC)*, at the edge routers, and queue management mechanism at the core routers, to realize the above objectives. The *TC* - comprising a meter plus marker - used at the edge for marking the packets as in-profile and out-of-profile can be classified into two broad categories: Token Bucket (TB) based [5], [6], [9] and average rate estimator based, also called Time Sliding Window (TSW) profile meter [1], [3], [7], [8]. In this paper, we use the terms *profile meter* and *TC* interchangeably. RIO-based [8] schemes have been proposed as simple means of Active Queue Management (AQM) at the core routers. The basis of the RIO (RED with In/Out) mechanism is RED-based [4] differentiated dropping of packets during congestion at the router. The RIO scheme utilizes a single queue. Two sets of RED parameters are maintained, one each for in-profile and out-of-profile packets. The drop probabilities of the in-profile packets are obviously lower than that of the out-of-profile packets.

Recent measurements across the transatlantic links have shown TCP flows being in majority with almost 95%

of the byte share [10]. TCP flows due to its congestion avoidance and slow start mechanisms [12] are much more sensitive to congestion, especially to multiple drops. Hence, providing assured services to TCP flows have always been an active research issue. It assumes more significance in the present day world, with more and more non-TCP flows flooding the networks, which make the TCP flows vulnerable. Thus, there is a need for designing intelligent TCP friendly marking algorithms, which take care of the TCP dynamics as well. The existing markers sense congestion only when the input flow rate reduces, which could be happening after the congestion has occurred. In this paper we propose an edge-to-edge feedback control mechanism and an intelligent marker which is proactive in cases of congestion. The marker uses a *congestion factor (cf)* provided by the control mechanism.

The paper is organized as follows: Section 2 gives an overview of the related work on various marker designs and the factors, which need attention. Section 3 explains the design issues, the edge-to-edge congestion feedback mechanism, and the algorithm of the marker in detail. Section 4 discusses the simulation setup. Section 5 presents the results and their analysis for different cases. Section 6 suggests the areas for deployment of our marker. We conclude with inferences and directions for future work in this area in Section 7.

2. Related work

TB-based marking [5], [6], [9] comprises all strategies that include one or more TB mechanisms measuring the amount of data that individual (or aggregate) flows generate in any time interval. The problem associated with the *TB based TC (TB-TC)* is that it is not easy to decide the optimal value of the bucket size. If it is small, the average rate of packets that are marked as in-profile will be less than the target rate. If the bucket size is large, it may cause unfairness in the sharing of the excess bandwidth. In [9], Sahu et al derive an analytical model for determining the achieved rate of a TCP flow when edge routers use TB -TC and core routers use AQM for preferential dropping. They report three important results: (i) the achieved rate is not proportional to the assured rate, (ii) it is not always possible to achieve the assured rate and, (iii) there exist ranges of values of the achieved rate for which TB parameters have no influence.

TSW profile meters (TSW-TC) [1], [3], [8] have two components: a rate estimator that estimates average sending rate over a time window (T_w), and a marker that tags packets as in-profile or out-of-profile. When a packet arrives, the TSW rate estimator estimates *avg_rate* (i.e., sending rate over a time window T_w) as ($avg\_rate * T_w + pkt\_size) / (T_w + pkt\_interval)$, where *pkt_size* is the packet size of the current packet and *pkt_interval* is the interarrival time between the current and the last packets. There are two approaches to use TSW profile meter: in the first approach, it remembers a relatively long past history (T_w is large); in the second approach, it remembers a relatively short past history ($T_w \cong$ RTT). The problem associated with the first approach is that it cannot reflect well the traffic dynamics of TCP. The drawback of second approach is that the average rate of packets that are marked as in-profile will be much more than the target rate in the under-subscribed scenario (i.e., when the actual throughput attainable is significantly higher than the target rate).

Feng et al [7] also used average rate estimator based TC (which they called *packet marking engine* (PME)) at the edge, and *Enhanced RED* (ERED) based differential dropping (which is same as the RIO scheme) at the core routers. The PME adaptively adjusts the packet-marking rate based on the measured sending rate. Unlike the other marking algorithms, not all of the in-profile packets are marked as priority packets, but in a probabilistic manner only. Also, some of the out-of-profile packets are marked as priority packets, again in a probabilistic manner. This marking probability adaptively changes for the entire range of the observed rate, i.e., for both below and above the target rate. Though this adaptive marking helped to maintain the assurance to TCP traffic in spite of the burstiness of the TCP traffic, Feng et al realized the potential network instability due to large swings in the number of marked (i.e., priority) and unmarked packets. Their solution for not marking more packets than required and for minimizing the instability problem is based on integrating the PME with the source congestion control mechanisms, which in turn modifies the source TCP protocol, and cannot be deployed for the profile meter at the edge routers.

3. Congestion-aware traffic conditioner (CATC)

In this section, first we discuss the major design issues that were of concern for us. Subsequently, we describe the edge-to-edge control architecture and the marker algorithm.

3.1. Design issues

Proper understanding of the transient congestion level in the network would certainly help in better resource management. The edge routers have a better understanding of the amount of total traffic in a domain and hence a feedback mechanism between these routers could help to improve the performance in a network by dynamically changing the priorities of the packets of a flow. An early indication of congestion in a network would help the marker to prioritize the packets of a flow in advance rather than doing it after the input flow rates have been reduced. This kind of proactive measures could help TCP flows to avoid multiple drops and thus reduce timeouts in periods of heavy congestion. None of the markers mentioned in sections 1 and 2 considers this crucial factor. This motivated us to design an adaptive marker similar to the one proposed in [7] but uses in its marking probability computation, the congestion feedback provided by the edge-to-edge control mechanism. The existing feedback mechanisms are end-to-end and available only for the end host to react upon indication of congestion, like the explicit congestion notification (ECN)-capable transport connections [15]. Hence, an edge-to-edge feedback mechanism for the flow aggregates, decoupled from the end-to-end feedback, was our next major concern.

Considering the above mentioned issues we came up with the idea of a *congestion-aware traffic conditioner (CATC)*, which uses the edge-to-edge feedback architecture to get an indication of the transient congestion level in the network.

3.2. Edge-to-Edge feedback architecture

The edge-to-edge feedback architecture consists of the two edge routers as the control sender and control receiver. The control sender sends control packets (short probe packets) at a regular interval of time called *control packet interval (cpi)*. The control packets are given the highest priority in the DiffServ domain. Each intermediate router maintains the *status* of drops, if any, of the best effort packets, until a control packet arrives at the node. For instance, consider an event where a best effort packet gets dropped. The *status* flag at that node is set to 1 to indicate a drop. Henceforth, the flag is reset to 0 only after the next incoming control packet to that node receives this status information. By doing so, we reduce the chances of any information regarding the drops in the best effort queue getting lost. The duration of time for which the information is maintained is, however, limited to a maximum of *cpi* time. On arrival at each of these intermediate nodes, the control packet's *congestion notification (CN) bit is* set or reset based on the *status* flag. In order to avoid any duplication in the drop status being carried over by multiple control packets, the *status* flag is reset to 0 as soon as the control packet gets the information. Upon arrival of the control packet at the control receiver, an acknowledgement is generated. The control receiver responds to incoming control packets that have the *CN bit* set by setting the *congestion echo (CE) bit* of the outgoing acknowledgement. The control sender maintains a parameter known as the *congestion factor (cf)*. The *cf* is set to 1 or 0 based on the status of the *CE bit* in the acknowledgements received.

3.3. The marking algorithm

Our *CATC* belongs to the class of *TSW-TCs* and hence has rate estimators, which calculate the average rate as in [8] and the markers, which mark the packets in a probabilistic way using an adaptive algorithm described as follows:

For each packet arrival

If avg_rate ≤ cir
then
$mp = mp + (1 - avg\_rate/cir) * (1 + cf * (cir/cir\_max));$

mark the packet using :
cp 11 w.p. mp
cp 00 w.p. (1-mp)

else if avg_rate > cir
then
$mp = mp + (1 - avg\_rate/cir) * (1 - cf * (cir/cir\_max));$

mark the packet using :
cp 11 w.p. mp
cp 00 w.p. (1-mp)

where,
avg_rate = the rate estimate on each packet arrival
mp = marking probability (≤ 1)
cir = committed information rate (target rate)
cf = congestion factor
cir_max = maximum committed information rate
also,
cp denotes 'codepoint' and *w.p.* denotes 'with probability'. We refer to packets with codepoint 11 as *marked packets (i.e, priority)* and those with codepoint 00 as *unmarked packets (i.e., best effort)* in later sections of this paper.

Note that the marking probability computation is based on the following four parameters:
i) *committed information rate (cir)*
ii) *average rate (avg_rate)* measured by the TSW rate estimator.
iii) *congestion factor (cf)*
iv) *maximum committed information rate (cir_max) among all the cir*s.
where, the first two parameters are specific to each flow (or aggregate) and the last two parameters are common for all flows (or aggregates) passing through that edge node.

Next, we discuss the effect of these factors in our marking algorithm. In the expression for the marking probability *mp*, we have two components:

i) *Flow component (1- avg_rate/cir)* constantly compares the average rate observed with the target rate to keep the rate closer to the target.

ii) *Network component cf*(cir/cir_max)* provides a dynamic indication of congestion level status in the network. Inorder to mitigate the impact of the target rates on the assurance (due to the longer time for a high target flow to rampup after a packet drop), the marking probability increment is done in proportion to the target rate by multiplying *cf* with a weight factor *cir/cir_max* .

Thus, the marking algorithm takes care of any transient congestion variation in the network and dynamically increases or decreases the marking probability *mp* accordingly.

4. Simulation details

The studies in this paper were performed using NS simulator [13] on Red hat Linux 7.0. We used Nortel's DiffServ module for implementing it in NS, which we modified to incorporate our feedback mechanism and the marking algorithm.

4.1. The scenario

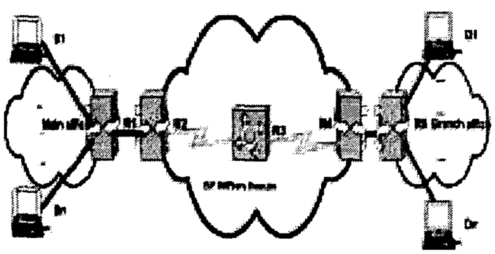

Fig. 1. The Topology

The architecture is for a VPN with an MPLS [16] like implementation. This situation is realistic in a Remote Office Branch Office (ROBO) scenario. The topology is as shown in Fig. 1. The traffic flows from the main office to the branch office. All the intermediate domains are assumed to be DiffServ enabled. The control packet flow always has the highest priority over the other flows in the network. The intermediate routers have RIO like mechanism which enables them to provide differentiated service and are modified to mark the control packets based on the *status* flag as mentioned earlier. Since most of today's routers have the capability of being ECN-enabled, we believe that our modifications would be a minor change to the existing implementations. All links from R1 to R5 are of the same bandwidth, which is mentioned later with the respective experiments. The marker is placed only at the egress edge router R1 of the main office. S1 to Sn represent the sources and D1 to Dn represent the receivers for the experiments. R2 and R4 are the edge routers and R3 is the core router of the DiffServ domain. R1 acts as the control packet sender and R5 the control packet receiver.

4.2 Simulation parameters

We used FTP bulk data transfer for the TCP traffic in all our experiments. Table 1 shows the typical values used for all our simulations. Any deviations from these values would be specified in the respective experiments.

TCP segment size	536 bytes
RTT	100 ms
simulation time	210 s
TSW window length	1 s
Control packet interval	1 ms
Control packet size	41 bytes
Link bandwidth	10 Mbps

	Marked	Unmarked
Min_th(packets)	250	150
Max_th(packets)	500	300
Max_dp	0.02	0.1

Table 1. Simulation parameters

5. Results and analysis

To test the effectiveness of our scheme, we conducted a series of simulations with different setups. For all our simulation studies, we calculated the goodput at the receiver whereas the rate estimator at R1 (Fig. 1) measured the sending rate. We believe this as the possible reason for some of the achieved rates being slightly less than the assured rate.

Expt #	Target Rates(Mbps)		Achieved Rates (Mbps)			
	Rt 1	Rt 2	Ra1	Ra2	BE TCP flow (Mbps)	Link goodput (Mbps)
1	1	1	2.54	2.58	3.76	8.88
2	1	2	2.54	2.58	3.76	8.88
3	1	3	2.41	2.93	3.46	8.8
4	2	3	2.36	2.89	3.58	8.83
5	3	3	2.8	2.8	3.21	8.81
6	3	4	2.73	3.49	2.59	8.81
Average link bandwidth (Mbps)						**8.835**

Table 2. Achieved Rates (Ra) for different Target Rates (Rt) -- under- and well-subscribed cases.

Expt #	Target Rates(Mbps)		Achieved Rates (Mbps)			
	Rt 1	Rt 2	Ra1	Ra2	BE TCP flow (Mbps)	Link goodput (Mbps)
1	2	6	1.83	4.85	2.06	8.74
2	3	5	2.5	4.04	2.05	8.59
3	3	6	2.4	4.6	1.53	8.53
4	1	8	1.2	6	1.28	8.48
5	4	6	3.17	4.5	0.11	7.78
6	2	8	1.55	6.16	0.72	8.43
Average link bandwidth (Mbps)						**8.425**

Table 3. Achieved Rates (Ra) for different Target Rates (Rt) -- over-subscribed cases.

5.1 Assured service for aggregates

The aim of these experiments was to study the capability of the CATC to assure the target rate for priority (AS) flows. Here, we analysed the effect of our marker in an under-, well- as well as over- subscribed scenarios.

i) Assured service in the under- and well- subscribed cases.

The experimental setup consists of two sets of priority TCP flows (each having 6 micro flows), with aggregate target requirements, along with a set of 9 best effort (BE) TCP micro flows. The bandwidth of all the links were set to 10 Mbps. Table 2 summarises the results obtained for various combinations of the target rates. Rt1 and Rt2 refer to the target rates whereas Ra1 and Ra2 are the achieved rates of the priority TCP flows.

The results clearly show that the priority flows are able to achieve their target rates for the under- and well-subscribed cases. As the target rates approach the over-subscribed case, we find that the marker tries to maintain the achieved rate close to its target rate. Also, it is to be noted that the total link utilization remains more or less constant, irrespective of the target rate variations from the under- to well-subscribed values. The average link utilization in this case is 88.35%.

ii) Assured service in the oversubscribed case.

Here, we study the capability of our marker to provide AS in an over-subscribed scenario in the presence of transient congestion. The simulation setup had two sets of priority TCP flows (each having 6 micro flows), with aggregate target requirements, along with a set of 9 best effort (BE) TCP micro flows. One set of priority flows runs throughout the simulation period of 210 s whereas the other set is an on-off source aggregate which remains *on* for a period of 50s each and is *off* for 20s each over the total simulation period of 210 s. We use an on-off source to simulate a condition of transient congestion. Table 3 shows the goodput achieved by the priority flows as well as the BE TCP flows during the periods when the on-off priority flows are in the *on* state.

From the results, we infer that the marker still helps in maintaining the achieved rate quite closer to the target rates. The overall link utilization is also maintained similar to the under- and well-subscribed cases mentioned earlier. The average link utilization is 84.25% in this case.

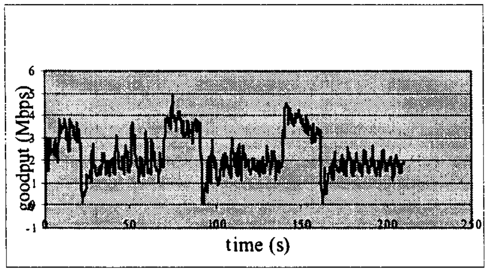

Fig 2. Goodput vs. time graph for continuous priority-flow with 2 Mbps target rate.

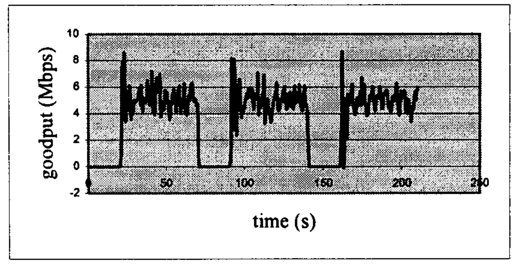

Fig 3. Goodput vs. time graph for on-off priority-flow with 6 Mbps target rate.

Expt #	Target Rates(Mbps)		Achieved Rates (Mbps)				
	Rt 1	Rt 2	Ra1	Ra2	BE TCP flow	BE UDP	Link goodput
					(Mbps)	(Mbps)	(Mbps)
1	2	6	1.52	4.18	0.46	3.54	6.16
2	3	5	2.08	3.41	0.44	2.52	5.93
3	3	6	2	4.42	0.13	2.12	6.55
4	1	8	0.66	6.34	0.01	1.87	7.01
5	4	6	2.65	4.6	0	1.5	7.25
6	2	8	1.21	6	0	1.6	7.21
Average link bandwidth (Mbps)							**6.685**

Table 4. Achieved Rates in presence of BE UDP and TCP.

Expt #	Target Rates(Mbps)		Achieved Rates (Mbps)				
	Rt 1	Rt 2	Ra1	Ra2	BE TCP flow	AS UDP	Link goodput
					(Mbps)	(Mbps)	(Mbps)
1	1	1	1.7	1.77	2.61	2.99	6.08
2	2	2	1.92	1.88	2.27	2.99	6.07
3	3	3	2.37	2.47	1.18	2.99	6.02
4	4	4	2.92	2.98	0.13	2.98	6.03
5	5	5	3.12	2.83	0.1	2.97	6.05
Average link bandwidth (Mbps)							**6.05**

Table 5. Achieved Rates in presence of AS UDP and BE TCP

Figs 2. and 3 show the goodput vs time graph for an over-subscribed case when the target rates of the priority flows are 2 and 6 Mbps respectively. Both the figures clearly illustrate the marker's ability to keep the achieved rates close to the target rates. Fig 2. shows the goodput over a time interval for the set of continuous priority-flows whereas Fig 3. shows goodput for the set of on-off priority-flows.

5.2. Protection from best effort UDP flows

The performance of TCP flows degrade drastically in the presence of unruly BE UDP flows. In this experiment we investigated the capability of our marker to provide an assured service to TCP in the presence of unruly UDP flows. The sending rate of UDP flows was 3Mbps.The bandwidth of all the links were 10 Mbps. The experiments were run with the priority TCP flows requiring different target rates to simulate an over-subscribed scenario. Table 4 summarises the results for this experiment. The results show our marker performing very well in achieving goodput close to the target rates considering the fact that we are having an over-subscribed scenario. Even in this worst-case scenario our marker tries to achieve a goodput close to the target rate by taking the share of the BE TCP and UDP flows. BE TCP flows get no goodput as the target rate requirements reach the maximum link bandwidth.

The results also show a decrease in the goodput of the unruly BE UDP flows as the target requirements of the priority flows increase. Thus our marker protects the priority TCP flows during periods of heavy congestion and achieve goodputs close to the target rates . The average link utilization is 66.85%.

5.3. Effect of UDP flows with assured rates

With an increasing amount of multimedia traffic in the recent years, assurance to UDP flows has become a reality. In such a scenario, performance of TCP flows coexisting with the UDP flows will suffer. Through this experiment, we analyse the effect of such UDP flows with assured rates on priority TCP flows in the presence of our marker. The results are summarized in Table 5. The experimental set up was similar to the one mentioned in the previous experiment with the exception that best effort UDP flow is replaced by an AS UDP flow having a target rate of 3 Mbps.

The results show that AS UDP flows get their assured rate in under-, well- and over-subscribed cases. The priority TCP flows achieve their assured rate in the under- and well-subscribed cases . In the over-subscribed case, the assurance for TCP is not completely met, although the marker tries to achieve a goodput closer to the assured rate. The average link utilization is 60.5%.

6. Deployment

The edge-to-edge feedback mechanism helps the marker to prioritise packets depending on the transient congestion levels in the network. Such an architecture can be used in VPN like environment. One possible deployment scenario could be in a network having an MPLS over DiffServ implementation. The marking scheme along with the feedback mechanism could be integrated with the network for dynamic prioritization of the packets of an aggregate flow. The marking algorithm is less sensitive to its own parameters unlike other markers as mentioned in the introduction. This lack of sensitivity could help the system administrators to implement our marker in any network without being concerned about setting the right marker parameters.

7. Inferences and future work

Feedback from the network has always been considered as an essential factor in improving the quality of service and better resource management. In this paper, we presented an intelligent adaptive traffic conditioner, CATC, which takes care of this factor. We conducted various experiments to show the effectiveness of our scheme. We list out some of the advantages of our scheme.

1) The architecture is transparent to TCP sources and hence doesn't require any modifications at the end hosts.

2) The edge-to-edge feedback control loop helps the marker to take proactive measures in maintaining the assured service effectively, especially during periods of congestion.

3) A single feedback control is used for an aggregated flow. Hence this architecture is scalable to any number of flows between the two edge gateways.

4) The architecture is adaptive to changes in load and network conditions.

5) The marking algorithm takes care of any bursts in the flows.

Future work would include extending the present architecture to take care of any drops in the priority queues as well. Such a feedback of drop status at different priority and non-priority queues along the path, received at the gateway, can be further used to interpret congestion in the network and act efficiently. Our future work would be to devise a new algorithm at the gateway by including the additional information mentioned above.

8. Acknowledgement

The authors would like to acknowledge the National Science and Technology Board (NSTB), Singapore for supporting this work under the Broadband21 (BB21) project grant.

9. References

[1] W. Lin, R. Zheng, and J. Hou, " How to make assured services more assured," in Proceedings of the 7th International Conference on Network Protocols (ICNP'99), Toronto, Canada, Oct. 1999, pp. 182-191.

[2] S. Blake, D.L. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, " An architecture for differentiated services," RFC 2475, December 1998.

[3] B. Nandy, N. Seddigh, P. Pieda, and J. Ethridge, "Intelligent traffic conditioners for assured forwarding based differentiated services networks," in Proceedings of IFIP High Performance Networking (HPN 2000), Paris, France, June 2000.

[4] Floyd S., and Jacobson V., "Random Early Detection gateways for Congestion Avoidance", IEEE/ACM Transactions on Networking, vol.1, no.4, August 1993, pp. 397-413.

[5] Heinanen J., Finland T., and Guerin R., "A Two Rate Three Color Marker," Internet draft, work in progress, May 1999.

[6] Kim H., "A Fair Marker", Internet draft, work in progress, April 1999.

[7] Feng W, Kandlur D, Saha D, Shin K, "Adaptive Packet Marking for Providing Differentiated Services in the Interent," IEEE/ACM Transactions on Networking, vol.7, no.5, pp. 685-697, Oct. 1999.

[8] Clark D. and Fang W., "Explicit Allocation of Best Effort Packet Delivery Service", IEEE/ACM Transactions on Networking, vol.6, no.4, pp. 362-373, Aug. 1998.

[9] Sahu S., Nain P., Towsley D., Diot C., Firoiu V., " On Achievable Service Differentiation with Token Bucket Marking for TCP", ACM SIGMETRICS 2000, August 2000.

[10] Simon L ., "UDP vs. TCP distribution", 01 Mar 2001,end2end-interest mailing list, http://www.postel.org/pipermail/end2end-interest/2001-March/000218.html.

[11] V. Jacobson, K. Nichols, and K. Poduri, "An expedited forwarding PHB," *RFC 2598,* Internet Engineering TaskForce, June 1999.

[12] Richard Stevens W., "TCP/IP Illustrated, Volume 1", Addisson-Wesley, 1994.

[13] NS simulator, http://www.isi.edu/nsnam/ns.

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

[15] Ramakrishnan K.K., Floyd S., " A proposal to add explicit congestion notification (ECN) to IP", RFC 2481, January 1999.

[16] Rosen E., Viswanathan A., Callon R., "Multiprotocol Label Switching Architecture", RFC 3031,January 2001.