A Memory-Based Approach for a TCP-Friendly Traffic Conditioner 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 TCP dynamics make the design of a marker difficult in many respects. We list out the issues related to designing a TCP-friendly marker and propose an intelligent marker, namely, Memory-Based Marker (MBM) to address those issues. We illustrate the benefits of the proposed marker over some existing markers in a DiffServ network. The marker was implemented in NS simulator and extensive simulations were done to study the behaviour of MBM. Our results show significant improvement in TCP performance, especially in achieving fairness among priority flows with distinct round trip times, windows, and target rates. The marker is capable of protecting TCP flows in cases of congestion caused by the unruly UDP flows. We also investigate the impact of coexisting assured service UDP flows on the assurance to the TCP flows. The major benefits of MBM are its simplicity, least sensitivity to parameters and transparency to the end hosts.

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

1. Introduction

The Differentiated Services (DiffServ) architecture [2], a scalable solution for providing service differentiation among flows, proposed by the IETF DiffServ Working Group supports two important services called *Premium* and *Assured* beyond the current Internet's *Best Effort* service. The class of Assured Services (AS) [16] 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.

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. The TC that is used at the edge router 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.

TB-based marking 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. 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).

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. Also, the TCP parameters like send and receive window sizes if not tuned appropriately might affect the flow throughputs. Hence, providing AS to TCP flows has 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.

In this paper we propose an intelligent TCP friendly marking algorithm for the *TSW-TC*. The rest of the paper is organized as follows: Section 2 gives an overview of the related work on markers for the average rate estimator based TC. Section 3 explains the design issues and algorithm for MBM in detail. Section 4 discusses the assumptions and simulation setup. Section 5 presents the results and their analysis for different cases. Section 6 suggests the deployment scenarios. We conclude with our inferences and suggestions for future work in this area, in Section 7.

2. Related work

Clark and Fang [8] reported one of the early simulation studies on RIO-based scheme with a marking policer that utilized an average TSW rate estimator and intelligent marker. 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. We have mentioned in the Introduction that there are two approaches for the marker: in the first approach, the profile meter 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). They used the second approach – the profile meter looks for the peak of a TCP saw tooth when the TCP exceeds *1.33*target*, at which point, it marks the packet as *out* with the probability *P=(avg_rate-target) /(avg_rate)*. All the packets are marked as *in* otherwise.

In [13] the authors raise issues with providing bandwidth assurance for TCP flows in a RIO-enabled DiffServ network equipped with remarking policer that utilizes the *TSW-TC*. They study the impact of five different factors on offering predictable bandwidth assurance services to customers: Round Trip Time (RTT), size of target rate, presence of non-responsive UDP flows, number of micro flows in a target aggregate, and packet size. Their study demonstrates that the above factors can cause different throughput rates for end users in spite of having contracted identical service agreements. One solution for this problem is to perform intelligent marking that take into account these factors in order to mitigate the impact of these factors [3]. However, the applicability of the marking algorithms proposed by Nandy et al [3] are limited due to the underlying assumptions of those algorithms; e.g., the RTT-aware algorithm assumes that the RTT for each flow is known at the edge and minimum RTT of the network is known to all edge devices. Still another assumption is that the TCP flows are operating in congestion avoidance. Also, these solutions are not feasible for flows that pass through multiple edge devices as it necessitates communication between edge devices, which in turn raises scalability issues. Further, these solutions are not applicable for a one-to-any network topology.

Other researchers [1], [14] have reported different approaches to mitigate the biasing effects of some of the factors outlined in [13]. Lin et al [1] have proposed enhanced TSW-TC and enhanced RIO-based AQM algorithms. However, the proposed solutions face scalability issues due to the usage of state information at the core of the network. The marking algorithm proposed by Yeom and Reddy [14] to mitigate the impact of RTT maintains per-flow information at the edge of the network.

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 marking algorithms discussed so far, not all 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. In order to minimize the chances of triggering such instability in the network, they proposed a TCP-like algorithm for the PME to update the marking probabilities in a more network friendly manner. However, the impact of the various factors such as RTT, size of target rate, etc., in providing the assurance were not studied. They also proposed an alternative solution for the PME not to mark more packets than required and to minimize the instability problem. This solution 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. Memory-based marker

In this section we describe the major design issues that were of concern for us and the algorithm that we propose.

3.1. Design issues

TCP performance is highly influenced by two parameters, namely RTT, and window size. Hence, one of the challenges was to design a marker which understands the TCP dynamics and which helps in reducing the influence of RTT and window size on the performance achieved by the TCP flows. Since markers are mostly deployed at the edge routers, which cannot easily decide the window size and RTT of the various TCP connections passing through, our effort was to have a marker, which can indirectly sense the changes in these parameters and mark accordingly. Another issue was to develop a marker, which is least sensitive to its own parameters unlike the existing markers mentioned in section 1 and 2. For example, *TB-TCs* are very much sensitive to the bucket parameters and the *TSW-TCs* are very much sensitive to the time window (i.e., the past history that the marker remembers). Still another concern was to reduce the burstiness of the marked and unmarked packets, to avoid the potential instability problem reported in [7]. Our marking algorithm details clearly show how the first two issues are dealt with. The burstiness problem is resolved by means of probabilistic marking while each flow (or aggregate) is both in-profile and out-of-profile, and also by adaptively changing these marking probabilities.

Some of the other issues of importance were to have a simple algorithm which requires no support from the end hosts and hence be transparent to the end hosts, and to see that marking is optimal in the sense that while maintaining the observed rate close to the target rate, it should not mark more packets than required. That is, the assured service classes should obtain their fair share of the best effort bandwidth.

3.2. The marking algorithm

Taking the above issues into consideration, we came up with the algorithm for MBM. As mentioned earlier, it is a *TSW-TC* and hence has the rate estimator which calculates the average rate as in [8] and the marker, which marks the packets, based on this average rate. The MBM marking algorithm is described as follows:

```
For each packet arrival

If $avg\_rate \le cir$
 then
  $mp=mp+(1-avg\_rate/cir)+(par-avg\_rate)/avg\_rate$;
$par = avg\_rate$;

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

else if $avg\_rate > cir$
then
$mp= mp + (par - avg\_rate)/avg\_rate$;
$par=avg\_rate$;

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

where,
avg_rate= the rate estimate upon each packet arrival
mp = marking probability (≤ 1)
cir = committed information rate (i.e., the target rate)
par = previous average rate
cp denotes 'codepoint' and *w.p.* denotes 'with probability'.

Next we discuss the basis of our algorithm and the reason why we call it the *memory-based marker*. The TCP window size W and the round trip time RTT are related to the throughput by the equation [11],

$BW = \frac{3}{4}*(MSS*W)/(RTT)$ where W is expressed in number of segments.

Any variation in W or RTT is reflected as subsequent changes in BW, i.e., in our case, the avg_rate. This is our basis of introducing the parameter *previous* *average rate (par), which* is compared with the present average rate to track any change in the rate of flow and thus indirectly extract the variations in RTT or W. We call this the memory-based approach, because the *par* is used to take into consideration any instantaneous change in the average rate of the flow.

During the period when TCP flows experience congestion, either or both of the following occurs:

a) The cwnd reduces reducing the value of W
b) The RTT increases

In the expression for the marking probability *mp*, *(par – avg_rate)/avg_rate* tracks the variations in the above factors and thus increases or decreases the marking probability according to the changes in the flow rate, whereas *(1- avg_rate)/cir* constantly compares the average rate observed with the target rate to keep the rate closer to the target. Thus, when the *avg_rate* is below *cir* but increasing, the factor *(1- avg_rate/cir)* tries to increase the marking probability to reach the target, whereas the factor *(par- avg_rate)/avg_rate* tries to reduce the marking probability though at a lower rate. When the *avg_rate* is below *cir*, and still falling down, both the factors increase the marking probability. Similarly, it takes care of the instantaneous changes in the flow rate while *avg_rate* is above *cir*. This behaviour of the marker plays a major role in improving the performance of TCP. We refer to packets with codepoint 11 as *marked packets* and those with codepoint 00 as *unmarked packets* in later sections of this paper.

4. Simulation details

The studies in this paper were performed using NS simulator [15] on Red hat Linux 7.0. We used Nortel's DiffServ module for implementing it in NS, which we modified to incorporate our marking algorithm.

4.1. The scenario

In this section we outline the topology and basic assumptions used for all our experiments described in this paper. We consider a scenario where a main office has multiple sources sending traffic from the main office domain to the receivers at branch office domain. The whole traffic passes through a diffserv domain. We assume that all the intermediate routers have RIO based active queue management mechanism. The RIO parameters and buffer size are suitably set in order to avoid any kind of bottleneck. The typical values used to get the results reported in this paper are shown in Table 1. The topology is as shown in Fig 1. All links from R1 to R5 are of the same bandwidth, which is mentioned later with the respective experiments. The MBM 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.

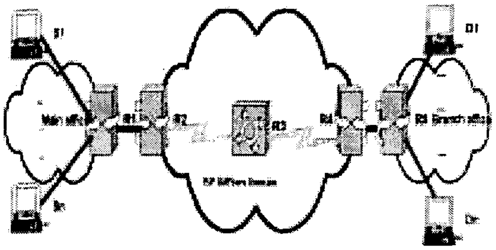

Fig 1. The Topology.

4.2 Simulation parameters

We used FTP bulk data transfer for the TCP traffic in all our experiments. Table 1 shows the values of common simulation parameters for all the experiments. Any deviation from the values specified in Table 1 would be mentioned in the respective experiments.

TCP segment size	536 bytes
RTT	100 ms
simulation time	210 s
TSW window length	1 s

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

Table 1. Simulation parameters

5. Results and analysis

We conducted a series of experiments to analyse the effectiveness of our marker. It is to be noted that for

Expt #	Target Rates (Mbps)		Achieved Rates (Mbps)					
	Rt 1	Rt 2	Ral		Ra2		BE TCP flow	Link goodput
			Total	Marked	Total	Marked	(Mbps)	(Mbps)
1	1	1	2.85	1.45	3.35	1.97	2.94	9.14
2	1	2	2.93	1.76	3.6	2.7	2.64	9.17
3	1	3	2.93	2.08	4.08	3.44	2.2	9.21
4	1	4	2.93	2.21	4.29	3.84	1.93	9.15
5	1	5	2.8	2.32	4.89	4.64	1.51	9.2
6	2	2	3.4	2.58	3.56	2.73	2.49	9.45
7	3	3	3.75	3.34	3.53	3.08	1.85	9.13
8	4	4	3.88	3.7	3.94	3.7	1.31	9.13
9	5	5	4.38	4.38	4.35	4.35	0.42	9.15
10	6	6	4.35	4.35	4.5	4.5	0.34	9.19
	Average link utilization = 92% (approx.)							**9.192**

Table 2. Achieved Rates (Ra) for different Target Rates (Rt).

RTT (ms)	Achieved Rates (Mbps)		
	Ra 1	Ra 2	per source pair goodput(Mbps)
60	1.82	3.81	5.63
80	1.49	3.74	5.23
100	1.52	3.52	5.04
120	1.38	3.58	4.96
140	1.43	3.45	4.88
Total link goodput			**25.74**

Table 3. Achieved Rates (Ra) for different RTT values

all our experiments, we have measured the goodput, whereas the rate estimator calculates the sending rate as the avg_rate. We account this as the possible reason for some of the achieved rates being slightly less than the assured rate.

5.1 Assured service for aggregates with different target rates.

We did a set of experiments with different combinations of target rates to analyse the behaviour of MBM in the cases of under-, over-, and well-subscribed networks. The aim of these experiments was to study the capability of the MBM to assure the target rate for priority (AS) flows. We 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. The bandwidth of all the links were set to 10 Mbps. Table 2 summarises the results obtained for various combinations of the target rates. The target rates of the two AS aggregates are indicated in columns 2 and 3. Next four columns show the achieved rates for these two aggregates. In addition to the total rates, we also show the component due to the marked packets in order to verify that the marking is optimal and the excess, i.e., best effort bandwidth, is equally shared among all the flows.

Analysis:

The results clearly show that the flows achieve the target rates in the under- and well-subscribed cases quite convincingly, and reach quite close to the targets in the over-subscribed case. As mentioned before, it is to be noted that we are measuring the goodput at the receiver whereas the marker uses the sending rate estimated by the TSW rate estimator. The results show that in the under-subscribed scenario, all the flows share approximately equal amount of the excess bandwidth. But in the over-subscribed regions, we see the priority flows getting a lesser share of the best effort. This is due to the fact that as the target requirement increases we see an increase in the marked packet rate in order to reach the target rates, which leaves very less amount of the unmarked packets for the AS TCP flows.

window size(KB)	Without MBM	With MBM
	Achieved Rates	Achieved Rates
	Ra (Mbps)	Ra (Mbps)
384	0.58	1.88
768	3.1	3.06
1125	3.21	2.87
1536	2.76	3.07
1920	1.25	2.93
Total Link utilization	10.90	13.81

Table 4. Achieved Rates (Ra) for different window sizes.

Target Rate (Mbps)	Achieved Rates (Mbps)			
Rt	Ra(tcp_prio)		Ra(udp_be)	Ra(tcp_be)
	Total	Marked		
2	3.83	2.03	2.95	2.6
4	4.85	4.13	2.91	1.66
6	5.76	5.6	2.84	0.81
8	7.13	7.13	2.22	0.04
10	7.94	7.94	1.4	0

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

5.2. Effect of different RTTs

We next studied the effect of different RTTs on MBM. TCP shows an unfair bias against long RTT flows during congestion [12]. Our aim in this experiment was to see if the MBM helps in reducing this bias. The experiment was performed with 5 pairs of flow aggregates, with different RTTs ranging from 60ms to 140ms. Each flow aggregate had 6 micro flows in it. The link bandwidths from R1 to R5 (as shown in Fig. 1) were all set to 28Mbps. The two aggregates of each pair had distinct target requirements of 1 and 4 Mbps and all flows in a pair had the same RTT. We set appropriate window sizes to avoid any bottlenecks due to it. We summarise the results in Table 3.

Analysis:

From the above results, it is evident that MBM does manage to reduce the TCP bias against long RTTs. The difference in goodputs achieved by the low latency flow (60 ms RTT) and the long latency flow (140 ms RTT) is only 0.75Mbps or 13%. The flows with a target rate 1 Mbps achieve their targets and are unaffected by this difference. The overall link utilization in this case is 92%.

5.3. Effect of different window sizes

Different users may use different TCP implementations, which have different advertised window sizes by default. Next, we studied the behaviour of MBM to TCP flows with different advertised window sizes. TCP is known to perform poorly if the window is not set to a value equivalent to the bandwidth-delay product [12]. The objective of this experiment was to see the effectiveness of MBM in providing the assurance to the priority TCP flows with different windows sizes, ranging from a low value to a higher than the bandwidth-delay product. The set up had 5 assured TCP flows having the same RTT (500ms) but different window sizes ranging from 384 to 1920 KB. The flows had a target rate of 3 Mbps. The link bandwidth from R1 to R5 (as shown in Fig. 1) was all set to 18 Mbps. We ran experiments with and without MBM (using the same setup) to compare the performance. The optimum window size for an RTT= 500ms and link bandwidth=18 Mbps is 125 KB. The results of this experiment are summarized in Table 4.

Analysis:

Based on the results achieved in Table 4, we note that without MBM, the flows with window values closer to the optimum value receives a greater share of the link bandwidth, whereas the flows with lower window values suffer. However, the goodputs achieved using MBM shows that the flows with a lower window (384 KB) gets a better share of the total bandwidth

Target Rate (Mbps)	Achieved Rates (Mbps)			
	Ra(tcp_prio)		Ra(udp_prio)	Ra(tcp_be)
	Total	Marked		
2	3.73	1.83	2.98	2.63
4	4.73	4.04	2.98	1.64
6	5.66	5.58	2.98	0.73
8	6.08	6.08	2.98	0.32

Table 6. Achieved Rates in presence of AS UDP and BE TCP.

compared to the situation when there was no MBM. The overall link utilization with MBM (76.7%) is also higher than without MBM (60.5%). We believe that using TCP extensions like SACK would help in achieving even better results with MBM.

5.4 Protection from best effort UDP flows

The interaction between TCP and UDP flows may cause the unresponsive UDP traffic to impact the TCP traffic in an adverse manner. In this experiment, we investigated the capability of MBM to provide an assured service to TCP in the presence of unruly UDP flows. Here we had a set of priority TCP flows along with a set of BE UDP and TCP flows. The sending rate of UDP flows was 3 Mbps. The bandwidths of all the links were 10 Mbps. The experiments were run with the priority TCP flows requiring a target rate ranging from 2 Mbps to 10 Mbps in order to simulate under-, over- and well-subscribed scenarios. The results are shown in Table 5.

Analysis:
Here, we observe that in the under-subscribed scenario, the priority TCP flows achieve their target easily, mostly by taking the BE TCP's share, whereas UDP flows are less affected. As we move on from well subscribed to the over-subscribed scenario, UDP BE flows too are affected and the priority TCP flows take on the share of both the BE flows. Thus MBM tries to achieve the target rate in all conditions.

5.5. Effect of UDP flows with target rates.

There is a need to protect certain UDP flows, which require the same fair treatment as TCP due to multimedia demands. This experiment was run to understand the behaviour of priority TCP flows in presence of an AS UDP flow with a target rate of 3 Mbps. The setup was similar to experiment D except that the UDP had a target rate of 3 Mbps. The results are shown in Table 6.

Analysis:
The priority TCP flow succeeds in achieving the target rates in the well- and under- subscribed scenarios. As we approach the over-subscribed region, the AS TCP flow fails to achieve its target rate whereas the assured UDP flow continues to enjoy its target rate. This bias in favour of UDP is expected as both AS TCP and AS UDP share the same logical queue in the RIO based routers. To guarantee the assurance to TCP, the AS TCP and AS UDP traffic should be assigned to different logical queues.

6. Deployment

The simplicity and least sensitivity to both TCP and marker parameters are the prominent advantages of MBM as has been illustrated in the above experiments. Notice that the marker has followed the TCP dynamics closely in spite of the large TSW window size of 1 sec unlike the other *TSW-TCs* mentioned in section 2. Hence, we suggest the possible deployment of an MBM marker at any edge routers used for traffic conditioning in a DiffServ network. We claim that system administrators would find it much easier to deploy in such routers without being concerned about setting up the right parameters for the marker. Also, a better performance of TCP flows with less influence of different values of RTT and window sizes certainly makes MBM a suitable candidate as a marker anywhere.

7. Inferences and future work

There is a growing need for intelligent TCP friendly markers in present day Internet. In this paper, we presented a memory-based approach in providing better quality of service especially for TCP flows. MBM stands out from other markers in its transparency from the end hosts, simplicity, and least sensitivity to parameters of both TCP as well as its own parameters. These claims have been substantiated in our experiments, which shows that MBM helps in achieving the target rate, with a better fairness in terms of sharing the excess bandwidth among flows. It also provides the TCP flows, a greater degree of insulation from differences in RTT and window sizes, which is one of the major causes of worry today. The overall link utilization also seems to be much better. The memory based approach plays a major part in establishing these results as has been explained in the previous sections. In our experiments, we used NewReno TCP implementation. We believe that by using the TCP extensions such as SACK, our marker would provide even better results. Future work would include extending the present algorithm of the marker to take into consideration the congestion in the network based on a feedback architecture. Experiments are also planned to study the behaviour of MBM with multiple congestion points.

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 D., 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] Mathis M., Semske J., Mahdavi J., Ott T., "The macroscopic behaviour of the TCP congestion avoidance algorithm.", ACM Computer Communication Review, vol. 27 , pp. 67-82, July 1997.

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

[13] Seddigh N., Nandy B., Pieda P., "Bandwidth Assurance Issues for TCP flows in a Differentiated Services Network", Globecom, March 1999.

[14] Yeom I., and Reddy N., "Impact of marking strategy on aggregated flows in a differentiated-services network", International Workshop on QOS, June 1999.

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

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