ELSEVIER

Computer Networks 38 (2002) 731–743

COMPUTER NETWORKS

www.elsevier.com/locate/comnet

TCP-friendly traffic conditioning in DiffServ networks: a memory-based approach ☆

K.R. Renjish Kumar, A.L. Ananda *, Lillykutty Jacob

Department of Computer Science, Centre for Internet Research, School of Computing, National University of Singapore, Lower Kent Ridge Road, Singapore 119260, Singapore

Received 12 October 2001; accepted 30 October 2001

Responsible Editor: I.F. Akyildiz

Abstract

Recently, there has been a considerable research interest in designing intelligent markers, tailored for TCP traffic. Markers, one of the building blocks of a traffic conditioner play a major role for resource allocation in a Differentiated Services (DiffServ) network. The TCP dynamics make the design of a marker difficult in many respects. In this paper, we list out the issues related to designing a TCP-friendly marker and propose an intelligent two-colour marker, namely, memory-based marker (MBM) to address those issues. We then extend this concept for a three-colour marker, memory-based three-colour marker (MBTCM), to be deployed for the assured forwarding per-hop behaviour in a DiffServ network. We illustrate the benefits of the MBTCM over time sliding window three-colour marker. The markers were implemented in NS simulator and extensive simulations were done to study their behaviours. 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 markers are 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 our markers are its simplicity, least sensitivity to parameters and transparency to the end hosts.

Keywords: QoS; Assured services; TCP-friendliness; Traffic conditioner; DiffServ networks; TSWTCM

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

☆ A version of this paper was presented in the IEEE ICNP 2001 conference, Riverside, California.

* Corresponding author.

E-mail addresses: kaleelaz@comp.nus.edu.sg (K.R. Renjish Kumar), ananda@comp.nus.edu.sg (A.L. Ananda), jacobl@comp.nus.edu.sg (L. Jacob).

1389-1286/02/$ - see front matter
PII: S1389-1286(01)00281-X

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 TCP-friendly markers done so far. Section 3 explains the design issues and algorithm for memory-based marker (MBM) in detail. Section 4 discusses the assumptions and simulation set-up for MBM. Section 5 presents the results and their analysis for different cases. Section 6 explains the marking algorithm of memory-based three-colour marker (MBTCM). In Section 7 we compare MBTCM with time sliding window three-colour marker (TSWTCM) and present the analysis of results. Section 8 suggests the deployment scenarios. We conclude with our inferences and suggestions for future work in this area, in Section 9.

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 $(\text{avg\_rate} \times T_w + \text{pkt\_size})/(T_w + \text{pkt\_interval})$, 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 Section 1 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 $1.33 \times$ target, at which point, it marks the packet as *out* with the probability $P = (\text{avg\_rate} - \text{target})/(\text{avg\_rate})$. 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.

In [17], the authors address the problem of serious performance issues in TCP due to bursty packet loss behaviour over DiffServ. They propose a series of TCP-friendly components to solve this problem. However, some of the components viz. the TCP rate increase dampers are implemented at the TCP source and thus are not transparent to the end hosts. Also the results show little improvements in the average goodput.

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 Sections 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 ⩽ 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 is the rate estimate upon each packet arrival; mp, the marking probability ($\leqslant 1$); cir, the committed information rate (i.e., the target rate); par, the 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 MBM. The TCP window size W and the RTT are related to the throughput by the equation [11]

$$\mathrm{BW} = 3/4(\mathrm{MSS} \times W)/(\mathrm{RTT})$$

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, $(\text{par} - \text{avg\_rate})/\text{avg\_rate}$ tracks the variations in the above factors and thus increases or decreases the marking probability according to the changes in the flow rate, whereas $(1 - \text{avg\_rate})/\text{cir}$ 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 $(1 - \text{avg\_rate}/\text{cir})$ tries to increase the marking probability to reach the target, whereas the factor $(\text{par} - \text{avg\_rate})/\text{avg\_rate}$ 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 traffic flows between two corporate networks (CNs) via an ISP network, which is DiffServ enabled. We assume that all the intermediate routers have RIO-based AQM 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 CN1. 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.

Table 1
Simulation parameters

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

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.

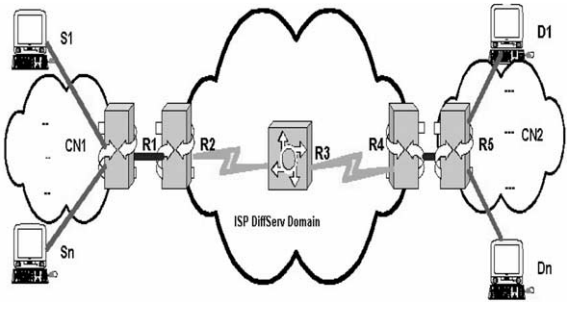

Fig. 1. The topology.

5. Results and analysis

We conducted a series of experiments to analyse the effectiveness of our marker. It is to be noted that for 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 six micro-flows), with aggregate target requirements, along with a set of nine best effort (BE) TCP micro-flows. The bandwidth of all the links were set to 10 Mbps. Table 2 summarizes 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.

5.1.1. 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.

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

Table 2
Achieved rates (Ra) for different target rates (Rt)

Expt. #	Target rates (Mbps)		Achieved rates (Mbps)				BE TCP flow (Mbps)	Link goodput (Mbps)
	Rt 1	Rt 2	Ra 1		Ra 2			
			Total	Marked	Total	Marked		
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 3
Achieved rates (Ra) for different RTT values

RTT (ms)	Achieved rates (Mbps)		Per source pair goodput (Mbps)
	Ra 1	Ra 2	
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 4
Achieved rates (Ra) for different window sizes

Window size (KB)	Achieved rates (Mbps)	
	Without MBM	With MBM
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

with five pairs of flow aggregates, with different RTTs ranging from 60 to 140 ms. Each flow aggregate had six micro-flows in it. The link bandwidths from R1 to R5 (as shown in Fig. 1) were all set to 28 Mbps. 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 summarize the results in Table 3.

5.2.1. 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 0.75 Mbps 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 window sizes, ranging from a low value to a higher than the bandwidth–delay product. The set-up had five assured TCP flows having the same RTT (500 ms) 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 set-up) to compare the performance. The optimum window size for an RTT = 500 ms and link bandwidth = 18 Mbps is 1125 KB. The results of this experiment are summarized in Table 4.

5.3.1. 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 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

Table 5
Achieved rates in presence of BE UDP and TCP

Target rate (Mbps)	Achieved rates (Mbps)			
	Ra (tcp_prio)		Ra (udp_be)	TRa (tcp_be)
	Total	TMarked		
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 6
Achieved rates in presence of AS UDP and BE TCP

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

priority TCP flows requiring a target rate ranging from 2 to 10 Mbps in order to simulate under-, over- and well-subscribed scenarios. The results are shown in Table 5.

5.4.1. 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 set-up was similar to experiment D except that the UDP had a target rate of 3 Mbps. The results are shown in Table 6.

5.5.1. 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. Memory-based three colour marker

In this section, we describe an extension of MBM, called the memory-based three colour marker and the improvements it has over MBM.

6.1. The marking algorithm

MBTCM is an extension of MBM and hence is again a TSW–TC. As in MBM, the MBTCM consists of a TSW rate estimator, which calculates the sending rate of the traffic as in [8] and a marker to mark packets based on our algorithm. In the MBM algorithm mentioned in Section 3.2, the marking probability, mp for the AS UDP flow will remain constant when the observed rate exceeds the target. This problem is tackled here with the MBTCM algorithm. Unlike the MBM, an MBTCM being a three-colour marker marks packets into green (code point 10), yellow (code point 11) and red (code point 00). The colours green, yellow and red represent drop precedences 0, 1, 2 respectively of a single AF class. The marking algorithm is explained as follows:

```
For each packet arrival
If avg_rate ⩽ cir
   then
      mp = mp + (1 − avg_rate/cir)
           + (par − avg_rate)/avg_rate;
      par = avg_rate;
```

```
    mark the packet using:
    cp 10 w.p. mp
    cp 11 w.p. (1 − mp)
  else if (avg_rate > cir) && (avg_rate ⩽ pir)
    then
      mp = mp + (par − avg_rate)/
           avg_rate(avg_rate − cir)/pir;
      par = avg_rate;
    mark the packet using:
    cp 11 w.p. mp
    cp 00 w.p. (1 − mp)
  else
    mark the packet using:
    cp 00 w.p. 1
```

where avg_rate, is the rate estimate upon each packet arrival; mp, the marking probability ($\leqslant 1$); cir, the committed information rate (i.e., the target rate); par, the previous average rate; pir, the peak information rate; cp denotes 'codepoint' and w.p. denotes 'with probability'.

The marking algorithm is designed based on the MBM algorithm and hence tracks the TCP dynamics based on the explanations provided in Section 3.2. The MBTCM meters and marks the packets of the traffic stream to green, yellow or red based on the measured throughput (the sending rate in our case) and three other rates: committed information rate (cir), peak information rate (pir) and the previous average rate (par). The MBTCM algorithm works as follows:

- If the estimated average rate (avg_rate) is less than or equal to cir, packets are marked as green (codepoint 10) with probability mp and marked as yellow (codepoint 11) with probability (1 − mp). The value of mp is calculated as mentioned in the algorithm. The component (1 − avg_rate/cir) helps in increasing or decreasing mp as the avg_rate varies with respect to cir, whereas (par − avg_rate)/avg_rate helps in tracking the instantaneous variations in the avg _rate and thus follow the TCP dynamics closely as described in the previous section.
- If the estimated avg_rate is greater than the cir but less than or equal to the pir, then the packets are marked yellow (codepoint 11) with probability mp and marked with red (codepoint 00) with probability (1 − mp). mp in this case is calculated as shown in the algorithm. Here, the component (par − avg_rate)/avg_rate continues to track the TCP dynamics and adjust the marking probability accordingly whereas (avg_rate − cir)/pir acts as the reduction factor for reducing the probability as the avg_rate increases towards pir. This component is particularly useful when the traffic stream has a constant avg_rate (e.g., UDP traffic) and is above cir. In such a scenario, mp does not remain constant but reduces to zero.
- If the estimated avg_rate is greater than pir, then the packets are marked as red (codepoint 00) with probability 1. Since the value of pir would be always higher than cir, which is the required rate, we believe that marking all the packets as best-effort in this case should not affect the quality of service.

7. TSWTCM vs. MBTCM—a comparison

TSWTCM [18] has been widely accepted as the marker for providing three different priority services. Like the MBTCM, it is a TSW–TC based marker. The marking is performed based on measured throughput and two parameters—committed target rate (ctr), and peak target rate (ptr). In this section we perform a comparative study of our marker with TSWTCM and provide an analysis of the results.

7.1. Simulation details

The topology and basic assumptions used for all our experiments are similar to the one described for MBM in Section 4 (see Fig. 1). We have set the value of pir and ptr *as* 1 Mps greater than the cir and ctr values.

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 7. The MBTCM/TSWTCM is placed only at the egress edge router R1 of the main office. S1 to S*n* represent the sources and D1 to D*n* represent the receivers for

Table 7
Simulation parameters

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
Green	900	1400	0.02
Yellow	600	1000	0.05
Red	400	700	0.1

the experiments. R2 and R4 are the edge routers and R3 is the core router of the DiffServ domain.

7.2. Results and analysis

In this section we present the results of simulations run for both MBTCM and TSWTCM and do a comparison of the results achieved.

7.2.1. Assured service for aggregates with different target rates

The aim of this experiment was to compare the capability of the MBTCM and TSWTCM to assure the target rate for priority (AS) flows. We had two sets of priority TCP flows (each having six micro-flows), with aggregate target requirements, along with a set of nine BE TCP micro-flows. The bandwidth of all the links were set to 10 Mbps. Tables 8 and 9 summarize the results obtained for various combinations of the target rates for MBTCM and TSWTCM, respectively. 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. We also specify the marked packet rates in order to show that the rate at which packets are marked is optimal. Here the marked packets include both the green and yellow colour packets.

7.2.1.1. Analysis. The results show that both MBTCM and TSWTCM help the flows in achieving the target rates in the under- and well-subscribed cases, and reach quite close to the targets in the over-subscribed case. However, we notice that in the under-subscribed regions, MBTCM achieves the assured rate with lower rate of marking compared to TSWTCM. This proves that our algorithm achieves the assured rates by maintaining an optimum level of marking. This could be seen as an economical advantage as well, since the customer can achieve his required target rate by paying less. As the target rates reach the well- and over-subscribed regions, we see a greater number of packets getting marked to maintain the QoS in the case of MBTCM. However it still

Table 8
Achieved rates (Ra) for different target rates (Rt) for MBTCM

Expt. #	Target rates (Mbps)		Achieved rates (Mbps)				BE TCP flow (Mbps)	Link goodput (Mbps)
	Rt 1	Rt 2	Ra 1		Ra 2			
			Total	Marked	Total	Marked		
1	1	1	2.55	0.01	2.64	0.01	3.99	9.2
2	1	2	2.57	0.02	2.9	0.74	3.67	9.14
3	1	3	2.28	0.13	3.53	1.95	3.33	9.14
4	1	4	2.17	0.19	3.89	3.41	3.19	9.25
5	1	5	2.29	0.31	3.93	3.76	3.25	9.47
6	2	2	3.13	0.95	3.14	0.73	3.4	9.67
7	3	3	3.41	2.62	3.2	2.12	2.45	9.06
8	4	4	3.31	2.92	3.58	3.2	2.64	9.53
9	5	5	3.13	3.07	4.6	4.3	2.2	9.93
10	6	6	3.83	3.82	4	3.99	1.8	9.63
	Average link utilization = 94% (approx.)							9.402

Table 9
Achieved rates (Ra) for different target rates (Rt) for TSWTCM

Expt. #	Target rates (Mbps)		Achieved rates (Mbps)				BE TCP flow (Mbps)	Link goodput (Mbps)
	Rt 1	Rt 2	Ra 1		Ra 2			
			Total	Marked	Total	Marked		
1	1	1	2.58	1.96	2.57	2.57	3.98	9.13
2	1	2	2.53	2.1	2.9	2.9	3.62	9.05
3	1	3	2.52	1.97	3.14	3.14	3.36	9.02
4	1	4	2.7	1.97	3.06	3.06	3.31	9.07
5	1	5	2.41	1.96	3.85	3.85	3.17	9.43
6	2	2	2.91	2.87	2.92	2.92	3.1	8.93
7	3	3	3.57	3.57	3.5	3.5	2.7	9.77
8	4	4	3.52	3.52	3.12	3.12	2.56	9.2
9	5	5	3.84	3.84	3.53	3.53	1.75	9.12
10	6	6	3.11	3.11	4.25	4.25	1.65	9.01
	Average link utilization = 92% (approx.)							9.173

maintains optimum marking. The total link utilization remains consistent in under-, well- and over-subscribed regions for both MBTCM as well as TSWTCM. The MBTCM helps in achieving better average link utilization of 94% compared to 92% with TSWTCM.

7.2.2. Effect of different window sizes

Here, we compared the behaviour of MBTCM and TSWTCM to TCP flows with different advertised window sizes. The set-up was similar to the one mentioned in Section 5.3. We ran experiments with and without MBTCM (using the same set-up) and TSWTCM to compare the performance. The optimum window size for an RTT = 500 ms and link bandwidth = 18 Mbps is 1125 KB. The results of this experiment are summarized in Table 10.

Table 10
Achieved rates (Ra) for different window sizes

Window size (KB)	Achieved rates (Mbps)		
	Without MBTCM	With MBTCM	With TSWTCM
384	0.58	2.07	3.23
768	3.1	3.12	1.4
1125	3.21	3.52	0.01
1536	2.76	2.58	4.27
1920	1.25	3.85	4.47
Total link goodput	10.90	15.14	13.38

7.2.2.1. Analysis. The results show that MBTCM helps in providing fairness to TCP flows with window sizes not equal to the bandwidth–delay product. We can see from Table 10 that the achieved rates with MBTCM are fairly consistent irrespective of difference in window sizes among the flows and is close to the required target rate of 3 Mbps. On the other hand, the flows with TSWTCM or without any marker show a bias towards the flows with different window sizes. We also notice the throughput of flows with TSWTCM to be inconsistent. Besides that, the total link goodput also seems to be much better at 15.14 Mbps with our marker compared to 13.38 Mbps with TSWTCM.

8. Deployment

The simplicity and least sensitivity to both TCP and marker parameters are the prominent advantages of MBM and MBTCM as has been illustrated in the above experiments. Notice that the markers have followed the TCP dynamics closely in spite of the large TSW window size of 1 s unlike the other TSW–TCs mentioned in Section 2. Hence, we suggest the possible deployment of these markers 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 our markers suitable candidates as markers anywhere.

9. 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. Both MBM and MBTCM stand 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 our markers help 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 markers to take into consideration the congestion in the network based on a feedback architecture. Experiments are also planned to study the behaviour of MBM and MBTCM with multiple congestion points.

Acknowledgements

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

References

[1] W. Lin, R. Zheng, J. Hou, How to make assured services more assured, in: Proceedings of the 7th International Conference on Network Protocols (ICNP'99), Toronto, Canada, October 1999, pp. 182–191.
[2] S. Blake, D.L. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, An architecture for differentiated services, RFC 2475, December 1998.
[3] B. Nandy, N. Seddigh, P. Pieda, 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] D. Floyd, V. Jacobson, Random early detection gateways for congestion avoidance, IEEE/ACM Transactions on Networking 1 (4) (1993) 397–413.
[5] J. Heinanen, T. Finland, R. Guerin, A two rate three color marker, Internet draft, May 1999 (work in progress).
[6] H. Kim, A fair marker, Internet draft, April 1999 (work in progress).
[7] W. Feng, D. Kandlur, D. Saha, K. Shin, Adaptive packet marking for providing differentiated services in the Internet, IEEE/ACM Transactions on Networking 7 (5) (1999) 685–697.
[8] D. Clark, W. Fang, Explicit allocation of best effort packet delivery service, IEEE/ACM Transactions on Networking 6 (4) (1998) 362–373.
[9] S. Sahu, P. Nain, D. Towsley, C. Diot, V. Firoiu, On achievable service differentiation with token bucket marking for TCP, ACM SIGMETRICS 2000, August 2000.
[10] L. Simon, UDP vs. TCP distribution, 01 March, 2001, end2end-interest mailing list, available from http://www.postel.org/pipermail/end2end-interest/2001-March/000218.html.
[11] M. Mathis, J. Semske, J. Mahdavi, T. Ott, The macroscopic behaviour of the TCP congestion avoidance algorithm, ACM Computer Communication Review 27 (1997) 67–82.
[12] W.R. Stevens, TCP/IP Illustrated, vol. 1, Addisson-Wesley, Reading, MA, 1994.
[13] N. Seddigh, B. Nandy, P. Pieda, Bandwidth assurance issues for TCP flows in a differentiated services network, Globecom, March 1999.
[14] I. Yeom, N. Reddy, Impact of marking strategy on aggregated flows in a differentiated-services network, International Workshop on QoS, June 1999.
[15] NS simulator, available from http://www.isi.edu/nsnam/ns.
[16] J. Heinanen, F. Baker, W. Weiss, J. Wroclawski, Assured forwarding PHB group, RFC 2597, June 1999.
[17] F. Azeem, A. Rao, X. Lu, S. Kalyanaraman, TCP-friendly traffic conditioners for differentiated services, IETF Internet Draft, March 1999.
[18] W. Fang, N. Seddigh, B. Nandy, A time sliding window three color marker (TSWTCM), RFC 2859, June 2000.

K.R. Renjish Kumar received the Bachelor of Engineering degree in Electronics and Communications from the Regional Engineering College, Suratkal, India in 1997. He is currently pursuing Masters in Computer Science in the area of quality of services in networks at the Centre for Internet Research, National University of Singapore. He was with Cognizant Technology Solutions for over two years and is currently working with Siemens ICM, Singapore as R&D Engineer. His research interests are IP QoS, TCP performance issues, wireless networks, mobile communication.

Akkihebbal L. Ananda is an Associate Professor in the Computer Science Department of the School of Computing at the National University of Singapore. He is also the Director of the Centre for Internet Research. He is actively associated with Singapore Advanced Research and Education Project (SingAREN) and has involved in network research and connectivity issues relating to Internet2. He is one of the key players in developing the NUS's campus secure plug-and-play network which has around 12,000 points campus wide. His research areas of interest include High-speed computer networks, transport protocols, collaborative applications, and distributed systems. He is a member of the IEEE Computer and Communications Societies. Ananda obtained his B.E. degree in electronics from the University of Bangalore, India, in 1971; his M.Tech degree in Electrical Engineering from the Indian Institute of Technology, Kanpur in 1973, and the M.Sc and Ph.D degrees in computer science from the University of Manchester, UK, in 1981 and 1983 respectively. From 1974 to 1980 he worked as a system software engineer in India.

Lillykutty Jacob obtained her M. Tech. degree in Electrical Engineering (Communication Engineering) from the Indian Institute of Technology at Madras in 1985, and Ph.D. degree in Electrical Communication Engineering (computer networks) from the Indian Institute of Science, Bangalore in 1993. She was a research fellow in the Department of Computer Science, Korea Advanced Institute of Science and Technology, S. Korea, during 1996–1997. Since 1985 she has been with the Regional Engineering College at Calicut, India. Currently, she is with the School of Computing, National University of Singapore, where she is a visiting academic fellow. Her research interests include quality-of-service and resource management in Internet, network protocols, and performance modelling and analysis. She is a member of IEEE.