Fair Stateless Aggregate Traffic Marking Using Active Queue Management Techniques

A. Das, D. Dutta, and A. Helmy

University of Southern California
abhimand@usc.edu, ddutta@isi.edu, helmy@usc.edu

Abstract. The differentiated services architecture promises to provide QoS guarantees through scalable service differentiation among multimedia and best-effort flows in the Internet. Traffic marking is an important component of this Diffserv framework. In this paper, we propose two new stateless, scalable and fair aggregate markers for TCP aggregates and UDP multimedia aggregates. We leverage stateless Active Queue Management (AQM) algorithms to design markers that ensure fair and efficient token distribution among individual flows of an aggregate. We present the Probabilistic Aggregate Marker (*PAM*), that uses the token bucket burst size to probabilistically mark incoming packets to achieve TCP-friendly and fair marking, and the Stateless Aggregate Fair Marker (*F-SAM*) that approximates fair queueing techniques to isolate flows while marking packets of the aggregate. Our simulation results show that our marking strategies show upto 30% improvement over other commonly used markers while marking flow aggregates. When applied to aggregate TCP flows consisting of long-lived flows(elephants) and short lived web flows(mice), our F-SAM marker prevents any bias against short flows and helps the mice to win the war against elephants.

1 Introduction

The Differentiated Service architecture *(Diffserv)* [2, 16] aims to provide statistical QoS guarantees within IP. Unlike Intserv [22], Diffserv is stateless, and, hence, scalable. It uses fine grained, per-flow marking at the network edges to classify flows into the various classes, and, then, applies coarser per-class service differentiation (Per-Hop Behavior) at the network core. There are two main types of Per-hop Behaviors(PHB) currently defined, the Expedited Forwarding(EF) PHB, and the Assured Forwarding(AF) PHB.

In this paper, we design packet markers for an Assured Forwarding PHB domain [13]. Traffic is marked at the edge into different drop-priority classes, according to a traffic profile based on service level agreements (SLA) between two domains. The network core uses simple AQM techniques to provide preferential packet dropping among the classes. Since the SLA at an edge domain applies to the total egress traffic, we need to perform traffic marking on the aggregated traffic at the edge. Simple token-bucket based markers however cannot distinguish between flows within an aggregate. In particular, misbehaving,

The original version of this chapter was revised: The copyright line was incorrect.This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-3-540-45812-8_28

K.C. Almeroth and M. Hasan (Eds.): MMNS 2002, LNCS 2496, pp. 211–223, 2002.

non-responsive flows can get more tokens. Also there is no provision for preventing long TCP flows from squeezing out short duration web traffic. Token-bucket markers aggravate bursty behavior, encourage misbehaving flows, and increase packet drops; this can severely affect both TCP and multimedia performance.

Thus there is a need to mark a traffic aggregate fairly and efficiently according to a given traffic specification. We need to ensure that we protect the TCP traffic in an aggregate from misbehaving flows. Also, we need to ensure that within an aggregate of multimedia UDP flows, the congestion-controlled well-behaved flows do not suffer at the cost of the other non congestion-controlled UDP flows. Again, if we consider just TCP traffic, we do not want the short term TCP flows (mice) to lose out to the long lived TCP flows (elephants). Hence we need to look at traffic markers which can provide fairness among any aggregate without maintaining per-flow state, thus ensuring higher scalability.

Traffic belonging to a single aggregate is marked into conformant (IN) and non-conformant (OUT) packets. The issues in Diffserv traffic marking are also similar to those in Active Queue Management [7] schemes. Well established work in active queue management such as RED [9], CHOKE [17] and CSFQ [18] provide certain degrees of fairness to flows and/or better throughput for flows by probabilistic dropping of packets from the queue, without maintaining per-flow state. We leverage AQM techniques to design two novel stateless, fair, aggregate markers. We show through extensive simulations that our methods achieve fairness in token allocation between flows without maintaining per-flow state in a variety of scenarios containing UDP, TCP and mixed traffic. We are not aware of any other work that utilizes AQM techniques to mark aggregate traffic in a fair and stateless fashion. Our techniques hold promise to control well behaved multimedia traffic and TCP traffic from misbehaving traffic within the Diffserv framework.

The rest of this document is outlined as follows. Section 2 presents related work on markers. A brief background on traffic marking is given in Section 3 and our first scheme, PAM, is describes in Section 4. Section 5 details our second scheme, F-SAM. Our simulation results are discussed in Section 6. Section 7 outlines future work and concludes.

2 Related Work

Packet marking is a well visited topic. Many current markers perform sophisticated per-flow marking [1] [21]. Others mark packet aggregates without maintaining any per-flow states. Stateless markers are simpler to implement, and, hence, desirable. There has been some work in stateless, aggregate marking. They typically rely on metering the average rates and comparing them against a token traffic specification [11], [12]. However these markers do not address fairness issues within an aggregate.

A sophisticated aggregated marker can be found in Fang. et. al. [8]. It does probabilistic marking based on the average aggregate rates, and in this respect, is similar to PAM. However instead of a RED-like transfer function based on the

average token-burst size, as in our case, they rely on the average incoming rate. Thus, their marker cannot handle bursty flows or the bias against short lived TCP flows while marking in a congested network.

Maintaining per-flow state can solve the problem of fair aggregate traffic marking. However, these schemes are not scalable when the number of flows entering a marker is varying and dynamic. Yeom and Reddy [21] address fair aggregate marking issues, but they assume that the individual flows within the aggregate have already been pre-allocated individual contract rate from the aggregate marker tokens. Also their algorithm is per flow based, and entails calculating individual flow throughput information using a TCP throughput model which requires Round Trip Time and packet size information for each flow. Unlike our markers, they only handle TCP flows.

In [19], the authors look specifically at the issue of sharing excess network bandwidth among traffic aggregates. However, they do not fairly marking individual flows within an aggregate, while we do. So their aggregate marker looks at inter-aggregate fairness as opposed to intra-aggregate fairness, as in our case.

In [1] the authors give preferential treatment to short-lived TCP flows in an aggregate, and perform a max-min fair allocation (of tokens) on the remaining TCP flows. But they assume that their marker has state about all the component TCP flows in the aggregate, which might not be practical or scalable. Also, their scheme works only for TCP.

In [14], Lloyd et. al. use the FRED algorithm to create a fair aggregate traffic marker. However this requires per-active flow computation at the marker, and is also more complex to implement than either of our markers. Also, they do not address the issues in fairness between long-lived and short-lived TCP flows.

Protection of TCP traffic and well-behaved multimedia traffic from each other and from misbehaving UDP traffic has been a great concern. In [3], the authors define thresholds for different classes of traffic. They use a dynamic threshold based scheme on top of RED to provide inter-class and intra class fairness between TCP and well behaved *tagged* multimedia traffic. Thus, they ensure that the well behaved UDP flows are not punished just due to their non-TCP friendly properties. However, their scheme does not use the Diffserv framework. Within a Diffserv framework, our markers will achieve the same goal if we mark TCP and UDP traffic with separate marker instances and decide to put the TCP and the UDP flows in different Diffserv classes. Also, unlike them, we present a class of fair markers that are based on AQM techniques.

We are not aware of any previous work in traffic marking that solves the problem of fair token distribution among flows within an aggregate without maintaining per-flow state, within the Diffserv framework. Besides, we are not aware of any marking techniques based on AQM techniques such as RED [9] and CSFQ [18].

3 Traffic Marking

Traffic marking is an essential component of the Diffserv architecture. The traffic marker looks at the incoming packets and compares it with a given traffic profile (for example, a token bucket(TB) characterization of a SLA between two domains). In-profile(IN) packets are marked with an identifier that indicates a higher priority. If packets are out of profile (OUT), it is *marked* with a lower priority.

The main challenge in marking is to efficiently and fairly distribute the allocated IN tokens (specified by the SLA) among individual flows of the edge network entering the marker. One solution is to maintain states of all flows and allocate IN-tokens according to the SLA. Clearly this is not scalable. The other alternative is to mark aggregates using a stateless algorithm. This is our approach.

The problem of aggregate marking is quite similar to the issue of queue management and scheduling. The queue receives packets from various flows and has to decide which packets from the incoming aggregate to buffer, which to drop, and how to allocated bandwidth fairly among flows. We can view a token bucket specification as a queue with the token bucket (TB) burst size as the queue size and the token arrival rate as the queue's link bandwidth. Marking an arriving packet as IN is same as queueing a packet for transmission, and marking it as OUT is equivalent to dropping an incoming packet from the queue. Both the average queue size and the average token bucket size give an indication of the congestion level at the queue/TB. However a small average queue size is equivalent to a large average token bucket size and vice versa. The problem of fair token distribution at a marker (using a Token Bucket traffic specification) among packets of various incoming flows, is equivalent to efficient buffer management and scheduling of incoming packets at a queue in a fair manner.

Active Queue Management techniques such as Random Early Detection(RED) [9] pro-actively drop packets before a queue overflow event, and allow for higher throughput and better fairness by reducing synchronization effects, bursty packet losses and biases against low bandwidth-bursty TCP flows. In addition, queue management techniques like Core Stateless Fair Queueing(CSFQ) [18], Flow Random Early Detection(FRED) [15], CHOKE [17] and Fair Queueing [6] try to distribute link bandwidth at the queue fairly among all flows of the aggregate.

We apply two algorithms from sample AQM techniques - CSFQ and RED, to aggregate Diffserv packet marking. We observe that the use of these techniques yields superior marker performance, in terms of throughput and fairness. In the next two section, we introduce our two stateless AQM based marking schemes. The first one, PAM, is based on RED and the second one, F-SAM, is based on CSFQ.

4 Probabilistic Aggregate Marker

Our Probabilistic Aggregate Marker (PAM) is based on *RED (Random Early Detection)* [9]. The aggregate traffic tries to consume tokens from the token

bucket at a certain rate. We maintain an exponentially weighed moving average (EWMA) of the number of tokens in the token bucket. On every incoming packet, we look at this average token bucket size. If this bucket size falls below a certain threshold min_{th}, all packets are to be marked with a lower priority. If the bucket size varies between min_{th} and max_{th}, we mark the packet as OUT based on a probability function that depends on the size of the token bucket. If the token bucket size exceeds max_{th}, we mark the packet as IN. More formally, our probability function of marking as OUT $P(x)$ can be written as follows.

$$P(x) = \begin{cases} 1 & : \quad x < min_{th} \\ \frac{P_{max} - P_{min}}{max_{th} - min_{th}} \times (max_{th} - x) & : \quad min_{th} < x < max_{th} \\ 0 & : \quad x > max_{th} \end{cases} \tag{1}$$

where x is the average size of the token bucket. Our probability function for marking a packet as OUT, therefore a modification of the RED probability function for accepting a packet into the queue.

This marking scheme ensures that the ratio of IN packets to OUT packets is flow-independent. Hence, the flow that pushes more traffic into the edge will, on the average, have a greater number of OUT packets; hence it will have more of its packets will be dropped if the core is congested. We show in Section 6 show that this scheme will be fairer to TCP flows in the presence of a misbehaving non-responsive flow, than simple token bucket schemes. The main advantages of PAM include its simplicity in implementation and deployment, and its stateless property.

5 Fair Stateless Aggregate Marker

In this section, we propose a very efficient, stateless aggregate fair marker *F-SAM* that employs algorithms from Core Stateless Fair Queueing (CSFQ) [18] like algorithm. CSFQ provides approximate max-min fairness while buffering incoming packets at a queue, by using a probabilistic dropping function based on the average rate of the flow to which the packet belongs. This rate information, instead of being calculated at the queue using per-flow techniques is calculated near the source of the flow and inserted in every packet header. Similarly, we maintain the rate information of a flow in the packet header itself, and use this to calculate a token probability, on a per-packet basis in the edge marker. Packets which can get through the probabilistic dropping function based on their rate information to get tokens from the token bucket are marked as IN, while others are marked as OUT. The queue, the output link speed, and drop probability in [18] are replaced by the token bucket, token bucket rate and $(1 - P[in\ token])$.

In F-SAM, the rate information in each packet header is calculated and filled by the ingress node when the flow enters the edge domain. Since each ingress node is responsible for maintaining the rate of only the flows originating from it, this per-flow rate calculation is simple and scalable. At the egress edge router, the edge marker needs to calculate the fair rate, f, allocated to the flows, and then calculates the token allocation probability of a packet as $min(1, \frac{f}{r})$, where r is the rate of the corresponding flow. So the expected token allocation rate to

a packet belonging to a flow of rate r, is then $min(r, f)$, which corresponds to the max-min fair rate of fair-queuing. As in [18], the fair-rate f, is calculated by an iterative method based on the average aggregate arrival rate of packets A, aggregate token allocation rate of packets R', and the token bucket rate C. Note that the fair-rate calculation algorithm does not need any per-flow measurements. Due to lack of space, we do not go into the specific algorithm details (see [18] or [4] for more details.)

Thus, F-SAM can distribute tokens among various flows in an approximate max-min fair manner like CSFQ. Note that fair token distribution among flow aggregates does not translate into a exactly fair division of throughput at the network core among TCP and UDP flows flows, if the core only implements simple RIO queueing. This is because the loss of any packet of a TCP flow would result in its halving its sending rate, while it does not affect the sending rate of the UDP flow. However, preventing large UDP flows from getting more IN tokens than TCP flows would significantly increase the throughput obtained by the TCP flows in presence of misbehaving traffic. Also, since this is a probabilistic marker, it will be not encourage bursty losses. Thus, even in the case of all-TCP aggregate traffic, the throughput and fairness obtained by individual TCP flows will be much higher than simple token bucket markers.

Note that even in case of aggregates of multimedia UDP flows, F-SAM will ensure fair token distribution to all the individual flows within the aggregate. Thus, given separate traffic specifications for UDP and TCP aggregates at a diff-serv marker node, one could have two instances of the F-SAM marker operating on TCP and UDP aggregates separately, while maintaining fair token allocation within each aggregate.

Another interesting property of F-SAM is that it can eliminate bias against short TCP flows in aggregate traffic consisting of both long and short TCP flows. Short lived flows do not get enough time to ramp up their sending rate and RTT estimation, and any packet loss in short flows can be lethal. In [1] and [10], the authors look at ways to remove this bias against short TCP flows in the marking and queueing domains. However these papers use per-flow mechanisms to reduce this bias, while F-SAM is a completely stateless marker and reduces this bias by the very nature of its probabilistic marking based on average rate of each flow. To provide a theoretical explanation of how F-SAM preferentially treats short flows, one needs to keep in mind that in the CSFQ algorithm the dropping probability is inversely proportional to the flow rate specified in the packet header. Now, if we use an EWMA based rate estimation algorithm, then for short flows, the rate estimation calculation is lower than the actual sending rate of the flow, since the initial rate of the short flow is small and the EWMA does not have time to ramp up before the flow ends. Since packets belonging to short flows arriving at the F-SAM marker have a lower rate in their header than their actual sending rate, the short flows gets a slightly bigger proportion of IN packets than what they are actually entitled to in a fair queueing scheme. This translates into better throughput for the short flows, as we shall see in the next.

6 Experimental Results

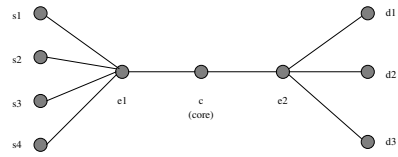


Fig. 1. Simulation scenario: A simple core with two edges. The s_i inject traffic of different classes into the network while the d_is are the sinks.

In this section we evaluate our two AQM-strategy based markers, PAM and F-SAM in progressively complex scenarios, using packet level network simulation, and compare their performance with aggregate markers such as the Token-Bucket marker(TB) and the TSW2CM marker [8].

6.1 Experimental Setup

We have used the *ns-2* [20] network simulator. The topology used in our experiments is depicted in Figure 1. The source nodes are s_i and the destination or sink nodes are d_i. The source nodes are connected to the edge node $e1$ which inject traffic into the network core, c. The core is connected to the edge $2e$ which is further connected to the sink nodes d_is. The source node s_3 is used to generate background traffic in form of many TCP flows carrying bulk traffic. This background traffic is marked with a separate DSCP (e.g., 20) and is absorbed at d_2.

6.2 Validating the Marking Process

	TCP Flow 1		TCP Flow 2		Misbehaving UDP Flow	
Marker	IN pkts	OUT pkts	IN pkts	OUT pkts	IN pkts	OUT pkts
TokenBucket	67	1261	76	1267	2423	7215
PAM	236	935	378	1456	2011	7607
F-SAM	740	882	748	835	1042	8254

Fig. 2. Detailed Packet Marking results with 2 TCP flows and 1 misbehaving UDP flow. For every marker, we give the number of IN and OUT packets obtained by each flow at the marker. The UDP flow has a rate of 2Mb/s.

First, we evaluate our PAM and F-SAM markers and check the amount of tokens obtained by individual flows in an aggregate. We conduct simulations for

	UDP Flow 1 (1Mb/s)		UDP Flow 2 (4Mb/s)		UDP Flow 3 (2Mb/s)	
Marker	IN pkts	OUT pkts	IN pkts	OUT pkts	IN pkts	OUT pkts
TokenBucket	13	748	805	2869	387	1593
PAM	245	695	695	2883	387	1593
F-SAM	497	620	413	2952	456	1571

Fig. 3. Detailed Packet Marking Results with three UDP flows of varying bandwidths. For every marker, we give the number of IN and OUT packets obtained by each flow at the marker.

a mix of UDP and TCP flows and for a mix of UDP flows of varying bandwidth. We verify that the proportion of packets marked for each flow by our marker conforms to our claims. This result helps us to estimate the end-to-end performance results. Due to lack of space, we present a very brief summary. A detailed description can be found in [4].

Consider the table in Figure 2. This experiment had 2 TCP flows multiplexed with 1 misbehaving UDP flow(2Mb/s) injected into the core. For our markers, we use a committed information rate (CIR) of 500kb/s and a burst size of 100kb and ran the simulation for 40 seconds in virtual time. We also use background TCP traffic in the topology to prevent flow synchronization. We calculate the number of IN and OUT packets marked for each flow and we find that TokenBucket markers allows the misbehaving UDP flow to squeeze the elastic TCP connections of IN tokens. However, in case of PAM, as we increase the sending rate, PAM marks IN packets proportionately i.e. the misbehaving flow gets penalized more (in terms of getting lesser IN packets than in the TB case). In other words, with PAM the ratio $\frac{IN\ packets}{total\ packets\ transmitted}$ is approximately the same for all flows. F-SAM demonstrates a fairer token allocation. We see that, in F-SAM, all the flows get approximately similar share of tokens from the marker, in spite of the large misbehaving UDP flow (which misappropriated most of the tokens from the TCP flows in the case of the TokenBucket marker). Thus we see, that the F-SAM marker effectively segregates flows from each other, and does not let misbehaving flows take away tokens from other well-behaved flows. In the next experiment, we consider only three UDP flows carrying CBR traffic, with varying transmission rates (1Mb/s, 4Mb/s, and 2Mb/s).

Looking at Figure 3, one can see a similar improvement of PAM and F-SAM over a TokenBucket Marker with respect to the fairness of token allocation among the three flows. With F-SAM, the number of tokens allocated to each flow is nearly the same, in spite of widely varying individual flows rates. Comparing with the token-bucket marker results , where the larger flows gather most of the tokens, we see that the F-SAM marker effectively isolates flows among the aggregates. It also evenly distributes tokens among the individual flows in a max-min fair manner, without maintaining any per-flow state.

6.3 End-to-End Performance

In this subsection, we compare the end-to-end performance of PAM and F-SAM with the TB (TokenBucket marker) and the sophisticated TSW2CM marker.

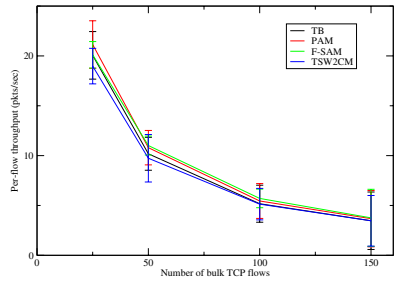
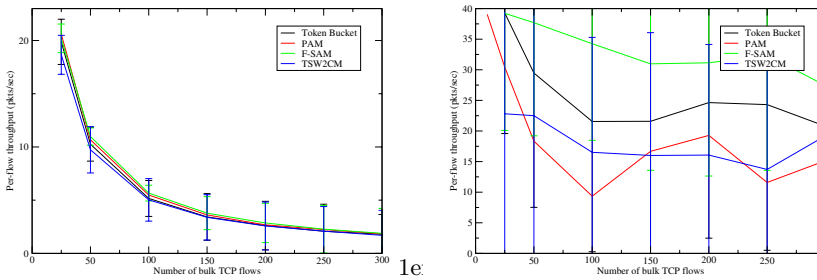


Fig. 4. Average throughput of flows, when only bulk TCP flows are used between $s1$ and $d1$



(i) Elephants (ii) Mice

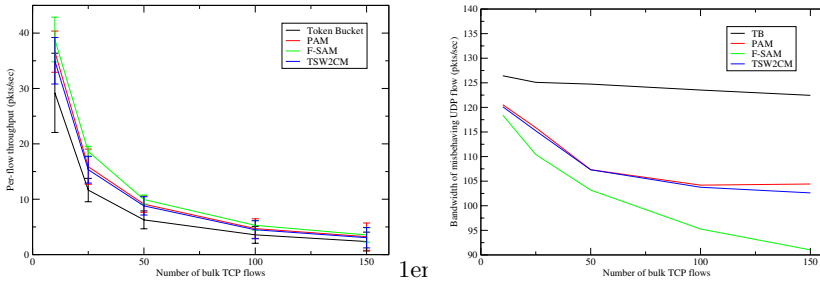
Fig. 5. Average throughput of flows when both bulk (elephants) as well as short lived TCP flows (mice) are used between $s1$ and $d1$

In all the experiments in this subsection, the CIR was 1Mb/s, the TB burst size was 500Kb/s and there was no separate background traffic as in the previous set of experiments. All the links except the one between *core* and $e2$(the bottleneck link) had a bandwidth of 10Mb/s, while the bottleneck had 5Mb/s. The delays in all the links were 5ms. We have used TCP Reno for all TCP agents and the short lived flows were generated by the HTTP 1.0 model in *ns-2*. The parameters for the RIO queues were the default ones in *ns-2*. We have got better results by tweaking the RIO parameters but we have not presented those results here.

First, we test our markers with long lived bulk TCP flows from $s1$ to $d1$ shown in Figure 4. We see that F-SAM is around 10% better than TB. PAM too performs better than TB (by 2-5%). It is interesting to note that both the throughputs due to PAM as well as F-SAM have a lesser standard deviation compared to TB, implying that our markers are more fair. The clear winner is F-SAM which has a much lesser standard deviation as it uses a max-min fairness criteria to distribute the token at the marker. Note that TSW2CM performs worse than TB in this scenario.

In the next experiment, we take several bulk TCP flows along with several short lived TCP flows (with an average arrival rate of 1 second and a payload size of 20KB) marked from the same token bucket. We illustrate the results for the long flow and short flows separately. Again, the average per-flow throughput for long flows with PAM and F-SAM yielded higher throughput and less standard deviation (Figure 5(i)). We must note that the F-SAM results have very low standard deviation (almost half of that of the TB marker), which demonstrates the fairness of F-SAM marking.

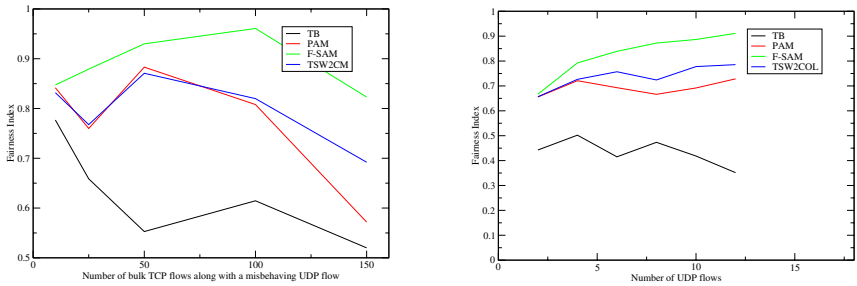
Figure 5(ii) depicts the bandwidth obtained by the short flow aggregates. Again, F-SAM beats the competition by a large margin. This is important as we can use F-SAM to win the war between mice and elephants[10] without maintaining per-flow state. Note that we do not need to detect which flows are short as our max-min fairness based F-SAM ensures fair token distribution. Also note that, the bandwidth obtained by mice is almost constant unlike the other markers. Surprisingly, for the short flows, we see that TB is better than PAM.



(i) TCP flows (ii) Misbehaving flow

Fig. 6. Bandwidth obtained by flows when bulk TCP flows are present between $s1$ and $d1$ along with a single misbehaving UDP flow with a rate of 1Mb/s between $s2$ and $d1$

In the next set of experiments, we use a single misbehaving UDP flow generating CBR traffic at 1 Mb/s apart from many bulk TCP flows. The throughput results for the TCP flows are depicted in Figures 6(i), (ii). As a result of our marking, we see that the average bandwidth received by the bulk TCP flows in F-SAM is the highest among all the markers(around 50% more than TB). PAM too is higher than TB by around 30% and is marginally better than TSW2CM. We note that the performance of TSW2CM is closer to PAM since probabilistic rate based marking in TSW2CM too helps to check misbehaving UDP flows. The better TCP performance of F-SAM is clearly due to an evenly distributed out-profile packets (packets marked OUT) compared to bursty out-profile packets in the case of the TB marker in presence of misbehaving UDP flows. This is also demonstrated by Figure 6(i) which shows the bandwidth obtained by the misbehaving UDP flow in the previous scenario. Clearly this result correlates with



(i) AllTCP + 1UDP (ii) All UDP flows

Fig. 7. Comparison of the Fairness Index of markers when only (i) many TCP flows and 1 misbehaving UDP flow is present and (ii) When only UDP flows are present.

Figure 6(ii), as we see F-SAM (and PAM to a lesser extent) penalizing the UDP flow much more than a TB marker. The fairer token distribution ensures that bandwidth is taken away from the misbehaving flow and is distributed among the TCP flows.

Figure 7(i) plots the fairness index for the previous experiment using the throughput of the TCP flows and the UDP flow. The fairness index [5] can be written as

$$FI = \frac{(\sum_i^N x_i)^2}{N \times \sum_i^N x_i^2} \tag{2}$$

where x_i is the per-flow throughput. We see a marked increase in the fairness index of F-SAM over all the other markers. PAM too exhibits a much greater fairness index than TB. The performance improvement for both F-SAM and PAM continues as the bandwidth of the misbehaving flow is increased. This demonstrates the efficacy of our marking schemes. We should note that no special tuning of queue parameters were required to get our performance improvement. In fact, we have obtained better performance when we tuned the parameters of the PAM marker as well as the RIO queues.

Next, we consider a scenario with only UDP flows. This is important for multimedia transmission, where we want well behaved multimedia streams to be protected by misbehaving flows. In this experiment, we used an aggregate consisting of a varying number of UDP flows, each with a different bandwidth varying from 2Mb/s to 10Mb/s. Our results show that while TB shows a high degree of unfairness in the allocation of tokens to the different UDP flows, F-SAM allocates approximately the same number of tokens to all the UDP flows irrespective of their bandwidth. PAM (and TSW2CM) try and allocate tokens in proportion to the incoming rate and also show greater fairness than TB. Figure 7(ii) plots the fairness index among UDP flows within the aggregate. The figure clearly shows that F-SAM treats flows fairly irrespective of their bandwidth and hence protects well behaved multimedia flows from large misbehaving

flows. We see that while TB performs the worst, PAM and TSW2CM perform better than TB and behave similar to each other.

Thus, we demonstrate that F-SAM does well to isolate misbehaving flows. We conjecture that techniques such as F-SAM perform close to the upper bound for fairness in aggregate marking, without per-flow calculations. PAM is a much less sophisticated marker, has less dramatic gains but its easier to implement than F-SAM. But they are both very easily deployable since they are stateless. Their performance shows a marked improvement over simplistic TokenBucket aggregate markers, and in fact come closer to that of many sophisticated per-flow marking schemes discussed in Section 2.

7 Conclusion and Future Work

In this paper, we have proposed two AQM based aggregate markers that ensure fairness among different flows in the aggregate without maintaining per-flow state. This makes our approach unique. Probabilistic Aggregate Marking (PAM) ensures fairness in a proportional fashion and allocates tokens to packets with a probability transfer function that looks similar to the transfer function of RED. We also presented a more sophisticated marker called Fair Stateless Aggregate Marker (F-SAM) which is based on fair queueing using *max-min* fairness criteria. The promising aspect of our work is F-SAM which can boost the performance of short lived flows and also ensure fairer bandwidth allocation between TCP and UDP flows of an aggregate, and within TCP aggregates and UDP aggregates separately.

The above markers are scalable and readily deployable. Our hope is that markers like PAM and F-SAM will enable a much more fairer and scalable solution to the problem of traffic marking in the differentiated services framework.

Future work in this area involves adapting other AQM techniques like CHOKe to develop fair aggregate markers. Additionally, for PAM, we plan to incorporate marking schemes based on PI controllers and try to dynamically auto-configure the RED-like parameters based on the traffic pattern. In F-SAM, we have not yet looked at the idea of weighted fair queueing while distributing tokens among the flows. Allowing for different weights to individual flows of the aggregate is the next logical step after approximating fair queueing. Finally, we want to explore how our markers can be used to implement good, fair pricing schemes in a diffserv QoS architecture.

References

1. A.Feroz, A. Rao, and S. Kalyanaraman. A tcp-friendly traffic marker for ip differentiated services. *Proc. of the IEEE/IFIP Eighth International Workshop on Quality of Service - IWQoS*, 2000. [212, 213, 216]
2. S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An architecture for differentiated services. *RFC 2475*, 1998. [211]

3. Mark Claypool and Jae Chung. Dynamic-cbt and chips - router support for improved multimedia performance on the internet. *ACM Multimedia*, 2001. 213
4. A. Das, D. Dutta, and A. Helmy. Fair stateless aggregate marking techniques for differentiated services networks, university of southern california technical report usc-cs-tr-02-752, 2002. 216, 218
5. Bruce S. Davie and Larry L. Peterson. *Computer Networks: A Systems Approach*. Morgan Kauffmann, second edition, 2000. 221
6. A. Demers, S. Keshav, and S.J. Shenker. Analysis and simulation of a fair queueing algorithm. *Sigcomm*, 1989. 214
7. B. Braden et al. Recommendations on queue management and congestion avoidance in the internet. *RFC 2309*, 1998. 212
8. W. Fang, N. Seddigh, and B. Nandy. A time sliding window three colour marker (tswtcm), 2000. 212, 217
9. S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking, V.1 N.4*, 1993. 212, 213, 214, 214
10. L. Guo and I. Matta. The war between mice and elephants. *ICNP*, 2001. 216, 220
11. J. Heinanen and R. Guerin. A single rate three color marker. *RFC 2697*, 1999. 212
12. J. Heinanen and R. Guerin. A two rate three color marker. *RFC 2698*, 1999. 212
13. J. Heinehan, T. Finner, F. Baker, W. Weiss, and J. Wroclawski. Assured forwarding phb group. *RFC 2597*, 1999. 211
14. Lloyd Wood Ilias Andrikopoulos and George Pavlou. A fair traffic conditioner for the assured service in a differentiated services internet. *Proceedings of ICC 2000, vol. 2 pp. 806-810*, 2000. 213
15. Dong Lin and Robert Morris. Dynamics of random early detection. *SIGCOMM '97*, pages 127–137, september 1997. 214
16. K. Nichols, V. Jacobson, and L. Zhang. A twobit differentiated services architecture for the internet. *RFC 2638*, 1999. 211
17. R. Pan, B. Prabhakar, and K. Psounis. A stateless active queue management scheme for approximating fair bandwidth allocation. *IEEE INFOCOM 2000*, 2000. 212, 214
18. Ion Stoica, Scott Shenker, and Hui Zhang. Core-stateless fair queueing: Achieving approximately fair bandwidth allocations in high speed networks. *Sigcomm*, 1998. 212, 213, 214, 215, 215, 216, 216
19. H. Su and Mohammed Atiquzzaman. Itswtcm: A new aggregate marker to improve fairness in difserv. *Globecomm*, 2001. 213
20. UCB/LBNL/VINT. The NS2 network simulator, available at `http://www.isi.edu/nsnam/ns/`. 217
21. Ikjun Yeom and A. L. Narasimha Reddy. Adaptive marking for aggregated flows. *Globecomm*, 2001. 212, 213
22. Lixia Zhang, Steve Deering, Deborah Estrin, Scott Shenker, and Daniel Zappala. Rsvp: A new resource reservation protocol. *IEEE Network Magazine*, September 1993. 211