A Fair Traffic Conditioner for the Assured Service in a Differentiated Services Internet

Ilias Andrikopoulos, Lloyd Wood and George Pavlou
Centre for Communication Systems Research (CCSR), University of Surrey
Guildford, Surrey, GU2 5XH, United Kingdom
Email: {I.Andrikopoulos, L.Wood, G.Pavlou}@ee.surrey.ac.uk

Abstract— A research issue under investigation in the context of differentiated services (DiffServ) is the fair distribution of bandwidth between aggregates sharing the same Assured Forwarding (AF) class. Multiplexing both responsive and unresponsive flows, e.g. TCP and UDP respectively, leads to unfair sharing of the available bandwidth in over-provisioned networks. To date, much effort has concentrated on experiments using different methods for mapping TCP and UDP flows of the same AF class to the three possible drop precedences of the AF specification. Although this approach may protect responsive from unresponsive flows, it has not been shown to provide adequate fairness. In this paper we present a traffic conditioner able to provide fairness between responsive and unresponsive flows originating from the same customer network, using a Fair Two-Rate Three-Color Marker. Its capability for fairness is based on the use of the FRED fair active buffer algorithm to control the token allocation of the token buckets residing in the traffic conditioner. We also show that by employing Fair Multiple RED (FMRED) at the DiffServ domain ingress node, the overall fairness of the customer network aggregates is improved when compared to the case where the vanilla MRED algorithm is used.

Index Terms— Fair traffic conditioner, differentiated services, assured forwarding, multiple RED, FRED.

I. INTRODUCTION

Research in the differentiated services (DiffServ) area [1] has focused, among other issues, on the evaluation and applicability of the services that can be defined from the use of the Expedited Forwarding (EF) [2] and Assured Forwarding (AF) [3] Per-Hop Behaviors (PHB). Although the former PHB is relatively easy to quantify and to define in terms of a service level specification (SLS), the latter is more challenging as the definition of the AF PHB allows for the specification of a wider variety of possible services. Mechanisms able to provide the AF PHB are currently being studied and constitute a significant research topic.

The AF PHB group allows a provider DiffServ domain to offer different levels of forwarding assurances to IP packets received from a customer DiffServ domain. The AF PHB provides for the delivery of IP packets in four independent delivery classes (AF classes), where each class is allocated a certain amount of resources, such as buffers and bandwidth, in each DiffServ node. Within each AF class, IP packets are marked either by the customer or the provider DS domain with one of three possible drop precedence values. In the case of congestion at a node, the drop precedence of a packet determines the relative importance of the packet within the AF class. A congested DS node tries to prevent packets with a lower drop precedence value from being lost, by preferably discarding packets with a higher drop precedence.

Recent studies [4, 5, 6] have shown that when responsive TCP flows share the same AF class with non-responsive UDP ones, there is unfair bandwidth distribution for aggregate flows. Given conservative TCP congestion control algorithms and the lack of similar control for UDP flows, UDP will tend to dominate the capacity available to the AF class, unfairly starving TCP of throughput capacity. The solution that has been followed until now is to map the TCP and UDP in-profile and out-of-profile flows to the three drop precedences of the AF class in a variety of ways. However, there have been differing conclusions as some authors [4, 5] claim that non-responsive flows should be penalised when congestion occurs, while others [6] even propose the use of two separate queues, one for each type of flow.

We believe that fairness should be strived for, both within the customer network and within the core DiffServ network. This can be achieved by using a fair traffic conditioner at the edge (egress) of each customer network to control the "local" fairness, and a fair version of RED, such as FRED, in the core provider network. FRED will be responsible for fairness among the aggregates originating from the customer networks. In particular, as we are dealing with the three drop precedences of the AF class, we propose that Fair Multiple RED (FMRED) should be used in the DiffServ ingress network nodes. This paper presents this approach, followed by simulation results demonstrating its utility.

The organisation of the paper is as follows: Section II gives a brief overview of relevant background information. Section III presents the architecture of the proposed fair traffic conditioner. Section IV describes the simulation configuration used for the evaluation of the Fair Traffic Conditioner and corresponding simulation parameter settings. In Section V, the results from the simulation experiments that demonstrate the efficiency of the proposed scheme are analysed and discussed. The conclusions of this work, together with proposed extensions of the current work, are given in Section VI. Finally, the Appendix presents the graphs of results as discussed in Section V.

II. Background

In this section we present some theoretical background of the basic components needed in the architecture of the fair traffic conditioner.

A. RED and RIO

Whether the assurance level of an AF class can be supported or not depends on the following factors:

- the amount of resources allocated to the AF class,
- the current class traffic load,
- and in the case of congestion, the drop precedence value of the IP packet.

The implementation of the AF PHB requires the existence of an active queue management mechanism that will be capable of minimising long-term congestion, while permitting short-term congestion in order to accommodate traffic bursts.

The most common mechanism employed today able to support the above requirements is the Random Early Detection (RED) active queue management mechanism. The goal of RED is to drop packets from each incoming flow in proportion to the amount of bandwidth that each flow uses on the output link. RED detects incipient congestion by estimating the average queue size at each packet arrival. RED calculates the average queue size using a low-pass filter on the instantaneous queue size, which permits transient bursts in the gateway. If this average value exceeds a minimum threshold *min_th*, RED begins dropping incoming packets with a dynamically computed probability. This drop probability increases with the average queue length *avg_queue* and with the number of packets accepted into the buffer since the last time a packet was dropped. The resulting high drop probability will detect and limit congestion by discarding packets early. If *avg_queue* exceeds a second threshold, *max_th*, then every incoming packet will be dropped until the queue size falls below *max_th*. A detailed description of RED and its associated parameters can be found in [7].

The RIO (RED with *In/Out bit*) scheme was initially proposed as a basis for providing two-tier service differentiation [8]. Packets from a flow complying with the contracted service profile are marked as *In* (in profile) and those packets that do not comply are marked as *Out* (out of profile), thus being assigned one of two drop precedences. RIO essentially comprises two REDs: the one used to control the *In* packets, the other the *Out* packets. The latter is more aggressive than the former as the *Out* packets must be dropped first in case congestion occurs. This should happen so that the *In* packets experience virtually no loss in a well-provisioned network. Although each RED in the RIO mechanism is configured with its own set of parameters, the calculation of *avg_queue* for the *Out* RED is based on the total queue occupancy, in contrast to *avg_queue* of the *In* RED. The RIO scheme can be extended to support multiple drop precedences, in which case it is known as Multiple-RED (MRED) – RIO is actually MRED with two drop precedences.
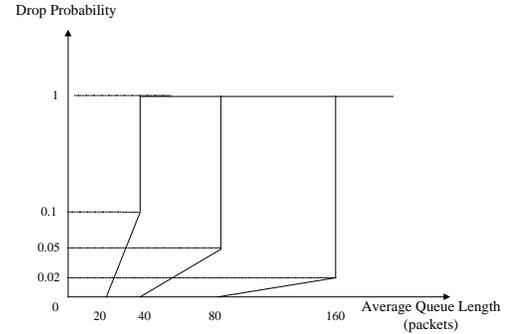

Fig. 1. An example of the drop probability as a function of the average queue length in MRED with three drop precedences (i.e. non-overlapping MRED).

An example of the drop probability as a function of the average queue length in MRED with three drop precedences is depicted in Fig. 1.

B. FRED

FRED is a modified version of RED that improves fairness between competing traffic flows when different types of traffic share a gateway. FRED is more effective in isolating unresponsive flows, provides better protection for bursty and for low speed flows, and is as fair as RED in handling identical robust flows such as bulk-data transfers. FRED provides these benefits by keeping state for just those flows that have packets buffered in the gateway, i.e. active flows. The cost of this per-active-flow accounting is proportional to the buffer size and is independent of the total number of flows, except to the extent that buffer use may depend on the number of active flows. An extensive description of the FRED algorithm can be found in [9].

C. The Traffic Conditioner in Differentiated Services

The traffic conditioner forms a key part of a differentiated services network. Its purpose is to apply conditioning functions on previously-classified packets according to a predefined profile, i.e. a traffic-conditioning specification (TCS). A traffic conditioner consists of one or more components, as follows:

Classifier: A device which measures the temporal properties of a traffic stream selected by a classifier.

Marker: A device that sets the DS Codepoint (DSCP) in a packet based on well-defined rules.

Shaper: A device that delays packets within a traffic stream to cause the stream to conform to some defined traffic profile.

Dropper/Policer: A device that discards packets based on specified rules (e.g. when the traffic stream does not conform to its TCS).

A typical arrangement of the above-mentioned components is illustrated in Fig. 2. For the purpose of this paper, no shaper or dropper has been used in our simulation experiments, in order to identify easily the effects of the proposed conditioning scheme.
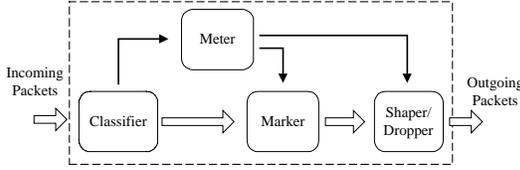
Fig. 2. Functional schematic of a Traffic Conditioner.

D. Two-Rate Three-Color Marker

In accordance with the definition of the AF PHB group, the Two-Rate Three-Color Marker (trTCM), a marker applicable to three drop precedences, was proposed as a component of the traffic conditioner [10]. The trTCM meters an IP packet stream and marks its packets based on two rates, the Peak Information Rate (PIR) and the Committed Information Rate (CIR), and on their associated burst sizes, the Peak Burst Size (PBS) and the Committed Burst Size (CBS) respectively. A packet is marked red if it exceeds the PIR, otherwise it is marked yellow or green depending on whether it exceeds or doesn't exceed the CIR. The two meters of the trTCM use token buckets with parameters (CIR, CBS) and (PIR, PBS).

Although this marker is suitable for identifying conforming and non-conforming packets and allocating each of them one of three drop precedences according to a given profile, it does not address the fairness problem when there are both responsive and non-responsive flows competing for the CIR and for any available bandwidth in excess of the CIR.

III. A FAIR TRAFFIC CONDITIONER

In this section we present a traffic conditioner with the capability to provide fairness among responsive and unresponsive flows sharing the same AF class and originating from the same customer network.

A simple fair marker was proposed in [11] to control the token distribution from the token bucket of the marker to the flows originating from the same subscriber network, in order to enforce fairness among them. However, the utility of this scheme was demonstrated only for a scenario where individual TCP and UDP sources are directly connected to a bottleneck link, thus having a one-to-one LAN configuration.

We extend this scheme so that the proposed implementation is based on the trTCM, where the FRED active buffer management algorithm has been employed to provide fair marking. We call this traffic conditioner FairTC (Fair Traffic Conditioner). Its fairness capability is based on the use of the FRED fair active buffer algorithm to control the token allocation of the token buckets residing in the traffic conditioner. Furthermore, we propose the use of Fair MRED (FMRED) at the DiffServ border node. Its presence is needed to improve the overall fairness between the customer network aggregates when compared to the case where vanilla MRED is used.

The rationale behind this proposal is to provide fairness between all sources belonging to a particular customer network, not only for their respective aggregate reserved rate but also for any excess bandwidth which may also be available. This means that packets originating from different sources within a customer network, and destined to be coloured either green or yellow at the edge router, should be marked in a fair manner. In other words, each source should be allocated a fair share of the green and yellow rates. Therefore, all green tokens are shared fairly between the sources, the remaining packets, i.e. those not marked green, are given fair share of the yellow tokens, and finally all the unmarked packets are marked red.

The FairTC is actually composed of two identical parts connected in series, which we call FairTC_PIR and FairTC_CIR due to their correspondence to the first and second token buckets of the trTCM respectively, as shown in Fig. 3. Associated with each part and each token bucket are:

A *trace queue*, which is a queue of records or traces of packets, belonging to active flows that have consumed tokens. This queue is emptied with rate PIR or CIR depending on the part of the FairTC.

A *state table*, which contains information on each individual active packet flow as the tuple (*Flow_ID*, n), where *Flow_ID* uniquely identifies the flow (e.g. is a hash of the source IP address and the port number) and n is the number of packets in that flow already in the trace queue that have consumed tokens.

We will now show how FairTC works by describing in detail what happens when a packet arrives at the edge node.

When a packet belonging to a particular flow arrives at the FairTC_PIR, the FRED algorithm uses the contents of the trace queue and the state table to make a decision on the packet. If FRED decides not to accept the new packet, the packet is marked red. Whether the packet is accepted or not depends on whether the flow to which the packet belongs has entirely used up its fair share of tokens or not.

If the packet is accepted, the token bucket is checked to determine if there are enough tokens available for the packet to consume – the tokens consumed are in proportion to the packet's size. If there are enough tokens available, the packet trace is queued in the trace queue. If not enough tokens are available, the packet is considered out-of-profile ("failed"), is marked as red, and exits the FairTC. Otherwise the packet is passed to the FairTC_CIR, where the same procedure takes place. The packets "failed" by FairTC_CIR are marked yellow and exit, whereas the successful ones are marked green and exit.
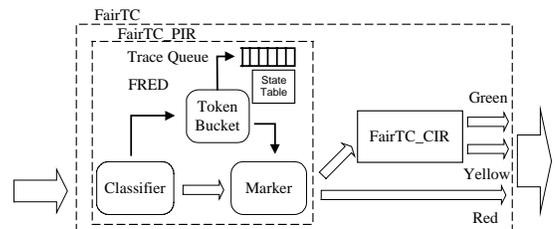

Fig. 3. The design of the FairTC architecture. The FairTC_CIR component is internally identical to the FairTC_PIR.
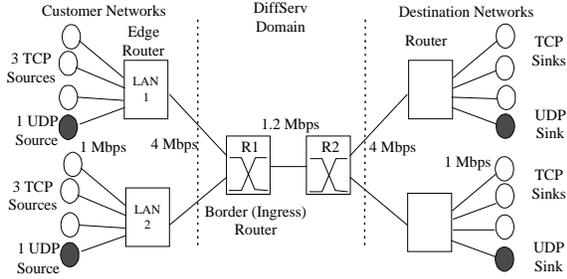
Fig. 4. Simulation network topology.

IV. SIMULATION CONFIGURATION

In order to study the performance of the FairTC, simulations were conducted using the topology shown in Fig. 4. With this topology we test the effectiveness of the FairTC and also demonstrate the advantage of using the FMRED at the ingress router of a DiffServ-capable network. The simulation configuration parameters are listed in Table I. Two switched LANs, comprising three Reno TCP sources and one UDP source each, perform unidirectional data transmissions across the bottleneck link (R1-R2) to two corresponding destination LANs. Only TCP ACKs are sent in the opposite direction, for which it is assumed that there is no loss. Each LAN is connected through an edge router to the border router R1. Every edge router includes either a vanilla Traffic Conditioner or our FairTC depending on the test scenario under consideration. Border Router R1 is equipped with a 'non-overlapping' three-drop precedence MRED, either the vanilla version or the fair version (i.e. FMRED) depending on the test scenario.

TABLE I
SIMULATION ASSUMPTIONS AND SYSTEM PARAMETERS

Parameter		Value
TCP MSS		512 bytes
Maximum Window Size		64 Kbytes
LAN Capacity		1 Mbps
LAN Propagation Delay		0.1 msec
Access Link Capacity		4 Mbps
Edge-to-Network Propagation Delay		1 msec
Bottleneck Link Capacity		1.2 Mbps
Bottleneck Link Propagation Delay		10 msec
Border Router R1 Buffer Size		300 packets
trTCM	CIR	256 Kbps
	CBS	5 Kbytes
	PIR	512 Kbps
	PBS	5 Kbytes
MRED / FMRED	DP 0 min_th	64 packets
	DP 0 max_th	128 packets
	DP 0 max Pb	0.02
	DP 1 min_th	32 packets
	DP 1 max_th	64 packets
	DP 1 max Pb	0.25
	DP 2 min_th	16 packets
	DP 2 max_th	32 packets
	DP 2 max Pb	0.5
	Weight_q	0.002

The values that were used throughout the simulations do not necessarily correspond to an optimal configuration. However, from simulation runs not shown here, using a wide range of parameters, this configuration was found to be suitable for testing the FairTC. The variation of the various parameters and their impact on the performance of the FairTC under a number of different conditions, which will allow us to create an adaptive version of the FairTC, is left as further work.

In order to evaluate the performance of the FairTC, we executed simulation experiments according to the following test scenarios:

1) trTCM in edge routers and MRED at the border.
2) FairTC in edge routers and MRED at the border.
3) FairTC in edge routers and FMRED at the border.

In the last scenario, FMRED is actually an MRED where RED is replaced by FRED. Instead of using per-active-flow accounting, as is normally the case in FRED, we use per-active-aggregate accounting, where each aggregate originates from each customer LAN. In the topology used for our simulations, each FRED in router R1 (i.e. one for each drop precedence) will see two aggregate flows in total and will try to allocate them fairly in the buffer according to the original FRED algorithm.

V. SIMULATION RESULTS

In this section we present and discuss the results obtained from the simulation experiments described in the previous section.

To evaluate fairness between the traffic sources of a specific LAN as well as between the aggregate throughputs between the LANs themselves, we use the following formula [12]:

$$FI = \frac{\left(\sum_i x_i\right)^2}{N \cdot \sum_i x_i^2} \tag{1}$$

where FI is the fairness index $(0 < FI < 1)$, x_i is the mean throughput of traffic source i, and N is the total number of sources under consideration. The closer the fairness index is to 1, the fairer the bandwidth distribution between sources.

Graphs illustrating the results of our simulations can be found in the Appendix at the end of this paper (Fig. 5 and Fig. 6).

In the first scenario, where trTCM was used together with MRED, each LAN managed to achieve its *CIR* and also share the remaining bandwidth fairly, but the total bandwidth of the bottleneck link was occupied by the UDP microflows of the two LANs. This was expected, as the UDP sources in our simulations are non-responsive, and will therefore overtake the TCP sources, which back off after their packets are dropped and thus are never able to increase their throughput to the desired level.

In the second scenario, where FairTC is introduced in the edge routers, the *CIR* for each LAN is achieved and is also

more fairly shared between the competing TCP and UDP sources than in the first scenario. On the other hand, the excess bandwidth is not shared well between the TCP and UDP sources, though the TCPs manage to have much higher throughput than the zero throughput of the previous scenario.

The results from the third scenario prove the utility of the proposed scheme. The combination of FairTC in the edge routers and FMRED in the border node not only provides both TCP and UDP sources much fairer access to the assured bandwidth, but also provides fairer access to the excess bandwidth.

From extensive experimentation – we lack space to show all the simulation results graphically – it was found that the parameters used for the FairTC and FMRED are sensitive to the traffic source characteristics and link rates. The settings of RED-related threshold values and queue sizes are especially affected. The choice of *PIR* was also found to impact the fairness results in our simulations. The closer *PIR* is set to *CIR*, the higher the fairness index. The reason for this is that the greater the difference (*PIR–CIR*) is, the more bandwidth the UDP microflow occupies from the available excess bandwidth. It should be noted that the overall fairness between the two LANs remains 0.999 in all test scenarios, which indicates that the two LANs always share the bandwidth equally. The results for LAN 2, not shown in the graphs, are similar to those of LAN 1.

VI. CONCLUSIONS AND FURTHER WORK

We have presented the FairTC traffic conditioner, and examined it in the scenario of a DiffServ-capable service provider's border node handling ingress traffic from customer networks subscribed to an Assured Forwarding service provided by the service provider. Our simulations have shown that FairTC, when combined with the FMRED active buffer management at the border of the DiffServ domain, is capable not only of enabling the customer networks to achieve their committed rate, but also of providing fair access to reserved and available excess unreserved bandwidth for multiple hosts within each customer network.

The work presented here can be extended further. The next step would be to evaluate the proposed scheme under more vigorous and realistic conditions. Testing in both over-provisioned (multiple levels of reservation) and under-provisioned network configurations, using both short- (e.g. HTTP traffic) and long-lived TCP traffic flows with diverse round-trip times per source, as well as using a network topology with multiple bottleneck links, is worth consideration. Moreover, the effect of a shaper in the traffic conditioner and the impact of variation in parameters for FairTC and FMRED also need investigation. Finally, based on our results, we expect that using an adaptive version of RED with self-configuring parameters based on the incoming traffic load [13], will allow for more stable behaviour that is insensitive to the initial RED parameter settings.

REFERENCES

[1] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang and W. Weiss, "An architecture for Differentiated Services", RFC2475, informational, December 1998.

[2] V. Jacobson, K. Nichols, K. Poduri, "An Expedited Forwarding PHB", RFC2598, proposed standard, June 1999.

[3] J. Heinanen, F. Baker, W. Weiss and J. Wroclawski, "Assured Forwarding PHB Group", RFC2597, proposed standard, June 1999.

[4] M. Goyal, A. Durresi, R. Jain and Chunlei Liu, "Effect of Number of Drop Precedences in Assured Forwarding", submitted as individual Internet Draft, July 1999.

[5] O. Elloumi, S. De Cnodder and K. Pauwels, "Usefullness of three drop precedences in Assured Forwarding Service", submitted as individual Internet Draft, June 1999.

[6] N. Seddigh, B. Nandy and P. Pieda, "Study of TCP and UDP Interaction for the AF PHB", submitted as individual Internet Draft, June 1999.

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

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

[9] D. Lin and R. Morris, "Dynamics of Random Early Detection", Proceedings of *ACM SIGCOMM*, pp. 127-137, September 1997.

[10] J. Heinanen and R. Guerin, "A Two Rate Three Color Marker", Internet Draft, May 1999.

[11] H. Kim, "A Fair Marker", submitted as individual Internet Draft, April 1999.

[12] R. Jain, *The Art of Computer Systems Performance Analysis*, John Wiley and Sons Inc., 1991.

[13] W. Feng, D. D. Kandlur, D. Saha, K. G. Shin, "A Self-Configuring RED Gateway", Proceedings of *IEEE INFOCOM*, March 1999.

APPENDIX
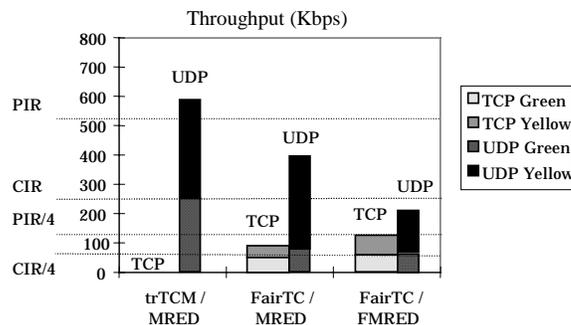## SIMULATION RESULTS

Fig. 5. Mean throughputs within LAN 1 for green and yellow packets of the TCP and UDP microflows in the case of the three test scenarios (for the TCPs, the throughput is shown averaged over the three TCP sources).
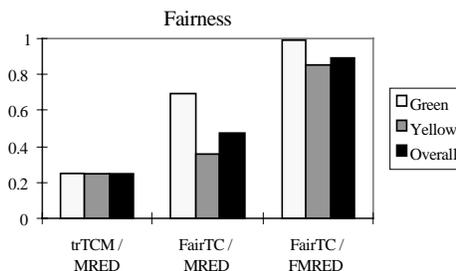

Fig. 6. Fairness within LAN 1 for green and yellow packets fort combined TCP and UDP microflows in the case of the three test scenarios.