Evaluating Fairness in Aggregated Traffic Marking *

Igor Briglia Habib de Almeida Alves
José F. de Rezende, Luis Felipe M. de Moraes
`igor@ravel.ufrj.br, rezende@gta.ufrj.br, moraes@cos.ufrj.br`
Lab. de Redes de Alta Velocidade (RAVEL) - PESC/COPPE/UFRJ
Grupo de Teleinformática e Automação (GTA) - PEE/COPPE/UFRJ
`http://www.ravel.ufrj.br, http://www.gta.ufrj.br`

Abstract

This article analyses the behavior and evaluates the performance of an implementation of the fair traffic marker proposed in [1]. It aims to enforce fairness among different flows from the same subscriber network in a DS domain. The results show that fairness can be achieved if parameters are set correctly. Well-defined guidelines are established to help configure the fair-marker. It is also shown that it cannot provide fairness in excess bandwidth allocation. An extension to the original proposal is done to overcome this problem. As an additional contribution, marking strategies proposals are discussed and classified according to two distinct criteria.

1 Introduction

The need to offer different service levels in the Internet has encouraged the research in Differentiated Services (diffserv or DS) [2, 3]. This proposal is based on a set of simple mechanisms that treat packets with different *priorities* as function of the marking in the DS field of the IP header. Before entering in a DS domain, this field is marked with a certain value (or codepoint) that determines the treatment that should be applied to this packet inside the domain. Therefore, the complexity associated to flow classification and mapping into flow aggregations is moved from the core to the borders. Inside the DS domain, different service levels are offered to the aggregated traffic instead of to each flow, what makes this architecture very scalable.

In the standardization groups, different treatments (Per-Hop Behaviors or PHBs) are being specified together with the associated codepoints. Two PHBs, now in wide discussion, are the Express Forwarding (EF-PHB) [4] and the Assured Forwarding (AF-PHB) [5].

In order to provide the desired level of service, traffic conditioning is performed by DS boundary nodes. Traffic conditioners may contain markers, meters, droppers and shapers to bring traffic into compliance with an established profile. Several markers were proposed in the literature [6, 7, 8, 9, 1]. The function of these mechanisms is to mark traffic according to the service profile contracted by the user. The behavior of these markers has a great impact on the service level, in terms of bandwidth, obtained by TCP flows that cross a DS domain. This behavior has been studied in several scenarios: TCP flows with different RTTs, with different service expectations (target rates), in presence of congestion-insensitive (non-adaptive) flows. In this last scenario, problems of fairness are observed in excess (non-contracted) bandwidth allocation between TCP and non-adaptive flows, such as UDP flows. However, a problem still few explored is fairness in bandwidth allocation among flows of an aggregate, when marking is performed on the aggregated traffic instead of per-flow.

In this work, we present a classification of traffic markers according to two different criteria; discuss the need of aggregated traffic marking, evidencing the problem of fairness that this sort of marking can bring; and present a solution to this problem, called fair-marker (FM) [1]. The fair-marker explores the duality between packet queuing and consumption of tokens in a bucket. From this duality, we describe a possible implementation of this marker using the algorithm specified in [10]. In order to evaluate the behavior of this marker in different scenarios and comparing it with other traffic markers, it was implemented in the ns-2 simulator. The results show that a correct tuning of the FM can make it guarantee a high degree of fairness in the allocation of the assured bandwidth among traffic flows that compose an aggregate. Important guidelines are provided

---

*This work was supported in part by FUJB, CNPq, CAPES, COFECUB, REENGE e FAPERJ.

to its configuration. Concerning the inefficiency of the FM in providing excess bandwidth sharing, we propose and evaluate an extension to this marker.

This article is organized as follows: Section 2 gives the fundamental concepts involved with the AF service and active queue management. Section 3 discusses and classifies several proposals of traffic markers. Section 4 describes the fair-marker in terms of objectives, operation and chosen implementation. Section 5 presents simulation results and analysis. Finally, section 6 summarizes the conclusions and perspectives of this work.

2 Fundamental Concepts

The AF-PHB provides IP packets delivery in four independent classes, called AFx classes (x = 1, 2, 3 or 4). For each class there is a certain amount of resources, such as buffer and bandwidth, allocated in each DS node. Within each AF class, an IP packet can be assigned, either by the user or by the DS domain, to one of the three levels of loss precedence ($codepoint = AFx1$, $AFx2$ or $AFx3$). In case of congestion within an AF class, a DS node preferentially discards packets with higher loss precedence values. Normally, DS nodes perform active queue management by using RED [11], one for each loss precedence level. Each RED aims to reduce the effects of congestion before it becomes necessary to discard packets with lower loss precedence values.

In [12], the authors present four general categories of RED policies when multiple loss precedence levels (or colors) are used in packet marking. These categories originate from the way one calculates the average queue size and sets drop thresholds for each RED algorithm. In this work, we use the Multiple Average/Multiple Thresholds (MAMT) category. In this policy, one average queue size is calculated for each precedence level, where the number of packets of a certain level is equal to the sum of packets of this and inferior levels (if any). In addition, each precedence level has different drop thresholds. For instance, the RIO queue (RED with *IN* and OUT) [6] belongs to this category. The average queue size for *IN* packets is calculated using solely the number of *IN* packets, while the average queue size for OUT packets is calculated using the number of *IN*+OUT packets. Different drop thresholds are defined for each level.

3 Traffic Markers

In this section, our concern is on the several strategies used to mark packets, in search of classifying them and understanding their differences. Marking can be performed by the provider or the customer. When marking is assigned to the customer, packets can be marked by a DS-compliant host or a customer router/access device. On both cases, the provider may monitor and remark packets to ensure compliance with the contract.

Marking strategies can be classified into three categories based on which knowledge is used to perform this task. Devices can mark packets: (i) based on the state of all individual flows of an aggregate, called *per-flow marking*, (ii) based only on the aggregation state, without any knowledge about individual flows, called *per-aggregation marking* or, (iii) based on a partial knowledge of individual flows, called *flow aware per-aggregation marking.*

When *per-flow marking* is performed in an aggregated traffic, the device responsible for marking packets needs to deal with individual flows states. In this aspect, *per-aggregation marking* is easier to manage and it is more suitable for customers that generate a huge number of individual flows. An example of a customer with this characteristic is a web-server. The number and dynamics of short-term flows generated by this kind of customer can prevent devices from performing *per-flow marking.* The large number of states associated with metering needed to do *per-flow marking* turns this strategy not scalable. Furthermore, giving each flow a fraction of the aggregation target rate can lead to an inefficient utilization of the reserved bandwidth. In this case, "idle" flows would waste their shares while preventing "active" flows from increasing theirs. Consequently, 100% of assured bandwidth utilization wouldn't be achievable. On the other hand, *per-aggregation marking*, despite being scalable, can introduce unfairness within aggregated flows. This unfairness can be caused by different target rates, different link bandwidth, or different levels of congestion experienced by individual TCP flows within the network [13]. In the *flow aware per-aggregation marking* category, devices responsible for marking are not aware of how many flows are being marked, neither any parameter associated to a particular flow is kept. However, the marker maintains a partial state of the flows being marked. This bounded number of states can be a major advantage in certain scenarios.

Most studies on diffserv networks deals with *per-flow marking* [6, 7, 14, 12, 15, 16, 13, 17]. The marking strategies presented in [13] focus on aggregated sources, while providing additional mechanisms to deal with unfairness within aggregated flows. The authors make a comparison between *per-aggregation marking* and *per-flow marking*, and three different strategies are proposed to alleviate the unfairness due to different RTTs and target rates. A similar study is done in [17]. In this work, Nandy *et al.* proposes some strategies in order to mitigate the effect of RTTs, UDP/TCP interactions, and different target rates.

Concerning the mechanism used to check the traffic conformity against the service profile, packet marking can be further classified in two broad categories: *token-bucket based* and *average rate estimator based.* This classification is completely orthogonal to the one described earlier, i.e. all marking strategies can be classified independently according to both criteria.

Token-bucket marking comprises all strategies that include one or more token-bucket mechanisms measuring the amount of data that an individual (or aggregated) flow generates in any time interval. Recent works on diffserv networks mostly use token-bucket marking [8, 9, 7, 14, 12, 15, 18]. To improve fairness in allocation of excess bandwidth between adaptive and non-adaptive traffic inside an AF class, new token-bucket marking strategies were developed [8, 9]. The effectiveness of three loss precedence levels was evaluated in [14, 12]. In [7], the authors evidence some advantages of token-bucket marking in respect to average rate estimator marking. In [18], Sahu *et al.* makes a performance analysis of token-bucket marking for TCP by means of an analytical model. Important findings resulted from this work.

In the *average rate estimator based* category, marking is performed according to the measurement of the average sending rate of individual (or aggregated) flows. The works in [6, 19, 13, 17] study this marking category. In the initial proposal [6], when the estimator measures an average rate that surpasses a certain threshold for a given flow, packets are marked as OUT with a linearly increasing probability. Clark and Fang propose the use of a time sliding window (*TSW*) rate estimator and an intelligent marker. In [19, 13, 17], authors propose some extensions to the *TSW* in order to improve fairness.

4 Fair-Marker

The fair-marker (FM) consists of a token-bucket based marker that performs *flow aware per-aggregation marking.* It focuses on distributing tokens fairly among individual flows from an aggregate. In order to achieve this purpose, it maintains information regarding the consumption of tokens by the monitored flows. Though, to avoid state implosion, FM only keeps states of flows that consumed tokens during the last time interval corresponding to the time needed to fill the token-bucket, denominated $TBFT$ (Token Bucket Fill Time).

FM uses an analogy between a token-bucket and a queue, where maintaining states from flows that consumed tokens during last $TBFT$ is similar to keeping states from flows that have packets in a queue. One can imagine a packet consuming a token as a similar situation of having a trace of this packet replacing tokens in the bucket. In practice, FM keeps a complementary queue (with the same size of the bucket) where these traces are stored. Whenever a token is generated, the queue is consulted to know if the number of tokens in the bucket is enough to remove some traces from the queue. To obtain a *fair* marker, traces are queued according to a fair buffer allocation algorithm.

For each arriving packet, one must determine how many traces can be removed from the complementary queue. This is equal to the number of tokens accumulated since the arrival of the last packet. After removing these traces, each individual flow that still has traces in the queue has its state updated. These flows are the ones that consumed tokens during the last $TBFT$. Next, packets are marked according to the current number of tokens in the bucket. If this number is insufficient, the packet is marked as OUT and its trace is not placed in the complementary queue. Otherwise, the consumption of tokens in last $TBFT$ determines whether the packet can consume tokens or not. The fair algorithm is used to determine if the packet trace is queued or not. In case it can be queued, tokens are consumed and the packet is marked as *IN*. Otherwise, the packet is marked as OUT.

Fairness in token distribution is a function of the fair buffer allocation algorithm used by the FM. In our implementation, we use FRED (Flow Random Early Drop) [10], which is a modified version of RED [11]. Besides the two minimum ($minth$) and maximum ($maxth$) thresholds, FRED introduces two new thresholds corresponding to the minimum ($minq$) and maximum ($maxq$) number of packets that each flow can have in the queue. FRED also controls the instantaneous ($qleni$) and the average ($avgq$) number of packets per flow, favoring the flows that have fewer packets than the average. Further, FRED punishes flows that try to exceed the maximum number of packets allowed per flow. More details about the FRED algorithm can be found in [10].

5 Simulations

The first topology we use in our simulations is illustrated in figure 1. To verify the interaction between TCP and UDP flows composing the same aggregate and the impact of different RTTs, four different scenarios are created: homogeneous TCP (same RTTs) with and without CBR (1 and 3), and heterogeneous TCP (different RTTs) with and without CBR (2 and 4). In all scenarios, the traffic is generated by ten FTP/TCP traffic sources from nodes n to nodes $n+10$, where $n = 1, 2, ..., 10$. The TCP flavor used is the TCP Reno with maximum window flow control of 90 packets. In the homogeneous TCP scenarios, the propagation delay between sources/sinks and routers is 1ms. In the TCP heterogeneous scenarios, this value varies from 1ms to 46ms in an arithmetic progression of ratio 5ms. So, RTTs vary from 44ms (TCP1) to 224ms (TCP10). Scenarios including CBR traffic have only one CBR/UDP traffic source from node 1 to node 11. This source generates traffic with a transmission rate of 2.5Mbps (100% of the bottleneck capacity). All packets are 1000 bytes long. The RIO queue has a capacity of 50 packets ($qlim$), and parameters for *IN* and $IN + OUT$ packets are equal to $[0.3 * qlim, 0.6 * qlim, 0.002, 0.1]$ and $[0.14 * qlim, 0.3 * qlim, 0.002, 0.1]$ [1], respectively.

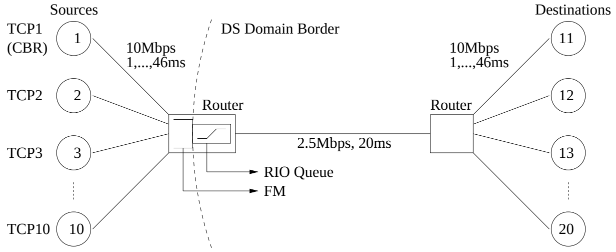

Figure 1: Topology 1.

The total time for all simulations is 55s. The sources start transmitting at a random time uniformly distributed between 0s and 5s. To eliminate the transient phase, all results are computed taking the interval from 10s to 50s. The FM has a bucket size (b) of 50 packets and a token rate (r) of 1Mbps (40% of the bottleneck). To verify the influence of FRED parameters on the FM behavior, we use the ranking method described in [20]. The parameters $minq$, $maxq = minth$ and $maxth$ assume the values 2, 4, 8, 16, 32 and 50, respecting the inequalities $minq < (maxq = minth) < maxth$. The other parameters are maintained constant with values $wq = 0.002$ and $maxp = 0.02$. The twenty configurations obtained are numbered according to the first column of table 1. The following three columns specify the corresponding parameter values ($maxq = minth$).

Table 1: Configurations Levels in the 4 Scenarios

#	$minq$	$minth$	$maxth$	1	2	3	4
1	2	4	8	18	8	1	1
2	2	4	16	19	13	4	3
3	2	4	32	20	18	7	7
4	2	4	50	20	17	7	7
5	2	8	16	17	11	2	2
6	2	8	32	20	18	8	8
7	2	8	50	20	18	13	12
8	2	16	32	17	17	4	2
9	2	16	50	17	12	11	9
10	2	32	50	17	4	10	6
11	4	8	16	17	10	2	1
12	4	8	32	20	17	8	8
13	4	8	50	20	17	12	12
14	4	16	32	17	17	4	2
15	4	16	50	19	13	11	8
16	4	32	50	17	4	10	8
17	8	16	32	19	16	4	2
18	8	16	50	18	12	11	9
19	8	32	50	15	4	7	6
20	16	32	50	17	4	10	5

For each scenario, we run five simulations for each configuration. For each flow, the number of packets marked as *IN* is calculated. Then, we compare the configurations regarding the fairness index (fi) calculated by the equation 1 [21], where x_i is the number of *IN* packets for the flow i and $N = 10, 11$ in the scenarios without and with CBR, respectively.

$$fi = \frac{(\sum_{i=1}^{N} x_i)^2}{N * \sum_{i=1}^{N} (x_i)^2} \tag{1}$$

In each scenario, a level is attributed to the values of fi according to table 2. The higher the fairness index the higher the level and the shorter its range. The classification levels for each configuration and scenario are shown in the columns numbered from 1 to 4 of table 1. Table 3 shows the total sum of the classification levels for the four scenarios and the ranking of each configuration according to this total. From these results, some guidelines can be stated in order to maximize fairness in assured bandwidth allocation:

- lower values of $maxth$ (close to $minth$) degrade the FM performance. In these cases, the maximum average number of IN packets in the FRED trace queue decreases, leading to a lack of space for all connections at the same time. In the scenarios with CBR, this non-adaptive connection will always occupy its share in the trace queue. On the other hand, TCP connections will fight with each other for space due to its bursty nature. In the TCP heterogeneous scenarios, the fairness index

---

[1] [minth,maxth,wq,maxp]

will also decrease since longer RTT TCP connections will be more sensitive to the increased number of packet drops (due to the lower $maxth$). We recommend $maxth$ close or equal to b.

- on one hand, when $maxq$ surpasses a certain limit in comparison with $maxth$, the performance of FM decreases. This is due to the increasing in the number of packets that can be marked as *IN* during a $TBFT$, which reduces the capacity of the FRED algorithm to punish the CBR flow and TCP flows with smaller RTTs. On the other hand, $maxq$ should not be very low so that TCP flows don't become punished for trying to reach this amount of IN packets in the trace queue.
- since the most important issue is that flows don't mark more than their fair share, $minq$ practically doesn't affect the performance of the FM. It is easy to mark at least $minq$ packets during a $TBFT$.

Table 2: Classification Levels

min	*max*	#	*min*	*max*	#
0.5000	0.6999	1	0.9600	0.9699	11
0.7000	0.7999	2	0.9700	0.9799	12
0.8000	0.8499	3	0.9800	0.9899	13
0.8500	0.8999	4	0.9900	0.9919	14
0.9000	0.9099	5	0.9920	0.9939	15
0.9100	0.9199	6	0.9940	0.9959	16
0.9200	0.9299	7	0.9960	0.9979	17
0.9300	0.9399	8	0.9980	0.9989	18
0.9400	0.9499	9	0.9990	0.9994	19
0.9500	0.9599	10	0.9995	1.0000	20

It is clear therefore that a good performance of the FM in terms fairness in distribution of tokens depends on the correct adjustment of its parameters. According to our results, a recommended configuration is $maxth = b$, $minq \leq 10\%b$ and $2 * minq \leq maxq \leq 25\%maxth$.

Next, we compare the FM with a classical token-bucket marker (TB). For this, we use the topology of figure 2, which simulates a more realistic situation. The monitored traffic consists of ten traffic sources of TCP Reno from nodes 1,...,10 to nodes 51,...,60, and a CBR/UDP traffic source from node 1 to node 51 with a transmission rate of 2.5Mbps (100% of the bottleneck). Ten additional TCP traffic sources, from nodes 11,...,20 to nodes 31,...,40 and using a best-effort service, are competing with the monitored traffic. The token rate of both markers (r) varies from 8% to 80% of the bottleneck (200kbps to 2Mbps). The parameters $minq$, $maxq = minth$ and $maxth$ assume the values 2, 8, 8 and 50, respectively (configuration 7). The RIO parameters for *IN* and *IN* + *OUT* packets are equal to $[0.5 * qlim, 0.8 * qlim, 0.002, 0.02]$ and $[0.2 * qlim, 0.5 * qlim, 0.002, 0.1]$ respectively. Five simulations are run for each value of r.

Table 3: Ranking of Configurations

#	*minq*	*minth*	*maxth*	*tot*	*rank*
7	2	8	50	63	1
13	4	8	50	61	2
6	2	8	32	54	3
12	4	8	32	53	4
3	2	4	32	52	5
4	2	4	50	51	6
15	4	16	50	51	6
18	8	16	50	50	8
9	2	16	50	49	9
17	8	16	32	41	10
8	2	16	32	40	11
14	4	16	32	40	11
2	2	4	16	39	13
16	4	32	50	39	13
10	2	32	50	37	15
20	16	32	50	36	16
5	2	8	16	32	17
19	8	32	50	32	17
11	4	8	16	30	19
1	2	4	8	28	20

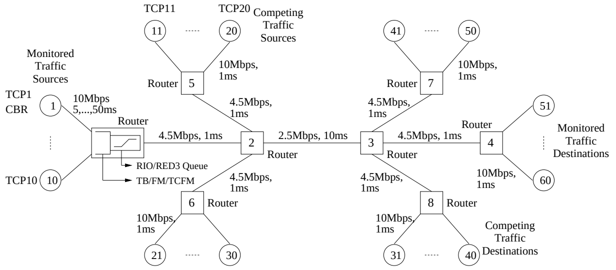

Figure 2: Topology 2.

Figure 3 shows the fairness in assured bandwidth sharing. The bars in each point define a confidence interval of 95%. The FM performs better than the TB, obtaining fairness above 0.9 for values of r up to 50% of the bottleneck capacity. The TB, independently of the value of r, gives a low fairness index, i.e. the CBR flow practically obtains all the assured bandwidth. Figures 4 and 5 show the fairness indexes of the excess (*OUT* packets) and total bandwidth (*IN* + *OUT* packets) sharing. Figure 4 shows that FM and TB have the same performance in terms of excess bandwidth, what can be explained by the fact that both mechanisms treat *OUT* packets the same way, without any action in the sense of guaranteeing fairness. Concerning the fairness in total bandwidth allocation (figure 5), the FM achieves higher fairness indexes as the number of *IN* packets in the network increases, since this sort of packets are fairly divided among flows. However, the performance of TB practically stays the same.

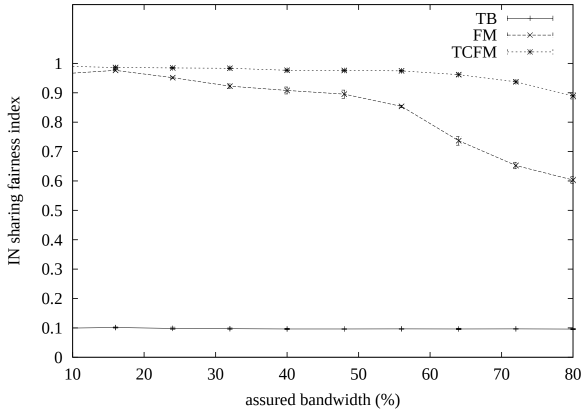

Figure 3: Fairness in the Assured Bandwidth Sharing

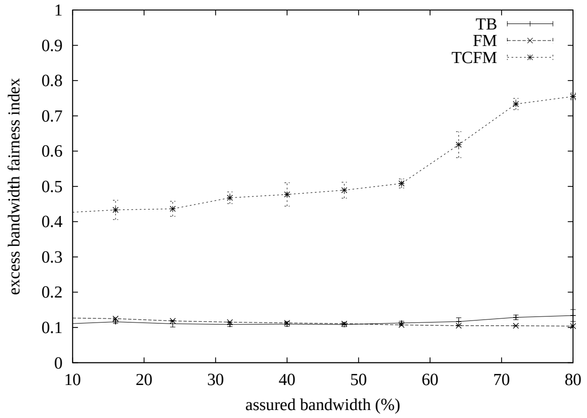

Figure 4: Fairness in the Excess Bandwidth Sharing

Figure 4 shows that FM lacks of an additional mechanism to deal with the problem of excess bandwidth allocation. To overcome this deficiency, we implemented an extension to the FM, called Three Color Fair-Marker (TCFM). This marker consists of two token buckets and two corresponding FRED trace queues. The two (green and yellow) token buckets work independently from each other and may have different bucket sizes, profile rates and trace queue parameters settings. A packet will be marked as *GREEN* if there are enough tokens in the green bucket and it can be queued in the green trace queue. A packet will be marked as *YELLOW* if at least on condition above is not satisfied and the same things happen for the yellow bucket and trace queue. Otherwise, a packet will be marked as *RED*.

For the purpose of evaluating the TCFM we test it under the same situation. However, since we are making use of three loss precedence levels, we replace the RIO queue by a RED3 queue with parameters values of $[0.6 * qlim, 0.8 * qlim, 0.002, 0.025]$ for *GREEN* packets, $[0.4 * qlim, 0.6 * qlim, 0.002, 0.05]$ for *GREEN+YELLOW* packets, and $[0.2 * qlim, 0.4 * qlim, 0.002, 0.1]$ for *GREEN+YELLOW+RED* packets. The green profile rate CIR (Committed Information Rate) is varied from 8% to 80% of the bottleneck capacity (as before) while the yellow profile rate PIR (Peak Information Rate) always correspond to 2.5Mbps - CIR. The green bucket size CBS (Committed Burst Size) and the yellow bucket size EBS (Excess Burst Size) are equal to 50 packets. Both FRED trace queues have the same settings of the FM trace queue for *IN* packets.

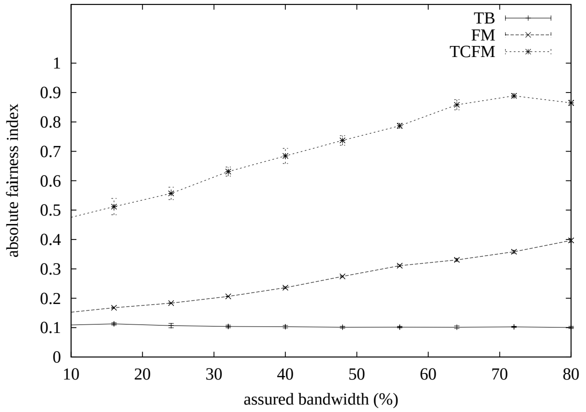

Figure 5: Fairness in the Total Bandwidth Sharing

These preliminary results show that the TCFM provides the same or better performance than the FM in assured bandwidth sharing (figure 3). Furthermore, the TCFM provides a considerable improvement in excess bandwidth sharing among the flows of the same aggregation (figure 4). This improvement can be explained by the impact of having a second FRED trace queue to fairly distribute yellow tokens among the flows. Consequently, this improvement in performance reflects in the total bandwidth sharing (figure 5).

6 Conclusions

In this work, we presented a classification of different types of markers existing in the literature, pointing out the need of marking aggregated traffic in the entry of a DS domain, and the problem of fairness among flows that compose an aggregate. This classification is based on two criteria: (i) if marking is performed using information concerning all flow states, only the aggregate state or a composition of both; (ii) if marking is based on the availability of tokens in a bucket or on the average transmission rate of flows.

For obtaining fairness among flows of an aggregate in the Assured Service, we present an implementation of the traffic marker defined in [1]. This implementation uses the active queue management FRED [10]. Concerning the assured bandwidth sharing among flows of an aggregate, FM outperforms the classic token-bucket. However, it is evidenced that its performance can be degraded as function of an inadequate adjustment of its parameters. Regarding the excess bandwidth sharing, FM is unable to assure fairness since no differentiated treatment is applied to *OUT* packets. The TCFM, an extension to FM, provides significant improvements regarding this issue. Future plans include a more deep analysis of the FM and the TCFM, using formal methods described in [20], regarding the adjustment of their parameters and their behavior in scenarios in which other factors of interest are varied.

References

[1] H. Kim, "A Fair Marker," `draft-kim-fairmarker-diffserv-00`, Apr. 1999.

[2] K. Nichols, S. Blake, F. Baker, and D. L. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers," *RFC 2474*, Dec. 1998.

[3] S. Blake, D. L. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services," *RFC 2475*, Dec. 1998.

[4] V. Jacobson, K. Nichols, and K. Poduri, "An Expedited Forwarding PHB," *RFC 2598*, June 1999.

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

[6] D. D. Clark and W. Fang, "Explicit Allocation of Best Effort Packet Delivery Service," *IEEE/ACM Transactions on Networking*, vol. 6, pp. 362–373, Aug. 1998.

[7] J. Ibanez and K. Nichols, "Preliminary Simulation Evaluation of an Assured Service," `draft-ibanez-diffserv-assured-eval-00`, Aug. 1998.

[8] J. Heinanen, "A Single Rate Three Color Marker," `draft-heinanen-diffserv-srtcm-01`, May 1999.

[9] J. Heinanen, "A Two Rate Three Color Marker," `draft-heinanen-diffserv-trtcm-01`, May 1999.

[10] D. Lin and R. Morris, "Dynamics of Random Early Detection," in *Proceedings of ACM SIGCOMM'97*, (Cannes, France), Sept. 1997.

[11] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, pp. 397–413, Aug. 1993.

[12] M. Goyal, A. Durresi, P. Misra, C. Liu, and R. Jain, "Effect of Number of Drop Precedences in Assured Forwarding," in *IEEE/Globecomm'99*, (Rio de Janeiro), Dec. 1999.

[13] I. Yeom and A. L. N. Reddy, "Marking for QoS Improvement," in *submitted to IEEE INFOCOM'2000*, Mar. 2000.

[14] O. Elloumi, S. D. Cnodder, and K. Pauwels, "Usefullness of Three Drop Precedences in Assured Forwarding Service," `draft-elloumi-diffserv-threevstwo-00`, July 1999.

[15] J. F. de Rezende, "Assured Service Evaluation," in *IEEE/Globecomm'99*, (Rio de Janeiro), Dec. 1999.

[16] S. Sahu and D. Towsley, "A Quantitative Study of Differentiated Services for the Internet," in *IEEE/Globecomm'99*, (Rio de Janeiro), Dec. 1999.

[17] B. Nandy, N. Seddigh, P. Pieda, and J. Ethridge, "Intelligent Traffic Conditioners for Assured Forwarding Based Differentiated Services Networks," *to appear in Proceedings of Netw'2000*, May 2000.

[18] S. Sahu, P. Nain, D. Towsley, C. Diot, and V. Firoiu, "On Achievable Service Differentiation with Token Bucket Marking for TCP," *to appear in ACM SIGMETRICS*, May 2000.

[19] W. Lin, R. Zheng, and J. Hou, "How to Make Assured Services More Assured," in *Proceedings of ICNP'99*, (Toronto, Canada), Oct. 1999.

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

[21] R. K. Jain, D.-M. W. Chiu, and W. R. Hawe, "A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer System," *submitted for publication to ACM Transaction on Computer Systems*, Sept. 1984.