Counters-Based Modified Traffic Conditioner

Maria-Dolores Cano, Fernando Cerdan, Joan Garcia-Haro, Josemaria Malgosa-Sanahuja
Department of Information Technologies and Communications
Polytechnic University of Cartagena
Campus Muralla del Mar s/n (Ed. Hospital de Marina)
30202 Cartagena, Spain
Ph. +34 +968 32 5953
Fax +34 +968 32 5338
{mdolores.cano, fernando.cerdan, joang.haro, josem.malgosa}@upct.es

Abstract. Traffic conditioners play a key role in implementing the Assured Service in the framework of the DiffServ approach. In this paper, we propose a traffic conditioner for the Internet Assured Service called Counters-Based Modified (CBM) that strictly guarantees target rates and performs a fair share of the excess bandwidth among TCP sources. The fairness in the outbound bandwidth distribution is met by probabilistically dropping OUT packets in the traffic conditioner. To determine the dropping probability of an OUT packet, some sort of signaling is needed. Although, it results more feasible than other proposed intelligent traffic conditioners. The CBM traffic conditioner is evaluated under different conditions by simulation using TCP Reno sources.

1 Introduction

The Assured Forwarding Per-Hop Behavior (AF-PHB) [1] is one of the IETF PHBs for Differentiated Services with the status of proposed standards. The idea behind AF-PHB is to ensure a minimum throughput (target rate or contracted rate) to a connection, while enabling consuming more bandwidth if the network load is low. To achieve this goal, packets of individual flows are marked belonging to one of the four independently forwarded AF classes. Within each AF class an IP packet can be assigned one of three different levels of drop precedence. In case of congestion, DiffServ nodes try to protect packets with a lower drop precedence value from being lost by preferably discarding packets with a higher drop precedence value. Note that minimum throughput is also called in-profile bandwidth or inbound bandwidth, and excess bandwidth can be also referred as outbound bandwidth along this study.

Despite of the abundant literature written about the AF-PHB (e.g. [2] to [9]), none solution has been found to face up its two goals, assuring the inbound bandwidth and offering a fair distribution of the excess bandwidth if available. Two different concepts can be understood as fairness in the outbound bandwidth sharing. The first considers fairness as the even distribution of excess bandwidth among all connections that compose the aggregate. The second defines fairness as a proportional distribution of the outbound bandwidth with respect to the contracted rate. In this paper we adopt the first definition. It should be remarked that despite DiffServ mechanisms are not implemented to provide and end-to-end service, it has sense to study the performance of TCP connections in terms of throughput excluding retransmitted packets, which is usually called *goodput*.

The Counters-Based (CB) traffic conditioner developed in [10] has been demonstrated to perform comparatively better than other traffic conditioners. This mechanism based on counters guarantees the in-profile bandwidth allocation in scenarios with variable round trip times and different target rates. Its easy configuration and high accuracy make it suitable for general use. Only two counters are needed to implement this algorithm, C1 and C2, and no parameter configuration is required. It also includes a simple mechanism to avoid accumulation of "credits" when a source stops transmitting data, for instance when a time out expires. From the comparative simulation study carried out in [10], this traffic conditioner together with RIO [2] over performs the two classical Time Sliding Window [2] (TSW)-RIO and Leaky Bucket (LB)-RIO mechanisms in terms of guaranteeing inbound bandwidth, with fluctuations in the achieved rate that do not exceed 1% of the connection target rate. Nevertheless, it also presents problems regarding the excess bandwidth sharing among sources as previous proposals.

In this paper, we introduce an alternative approach for achieving fairness in the excess bandwidth distribution among TCP sources for the Internet Assured Service. Starting from a high accuracy in assuring inbound bandwidth provided by the CB algorithm [10], we meet the fairness in the outbound bandwidth distribution adding a probabilistically dropping of OUT packets in the traffic conditioner. We call this new version of the CB algorithm the Counters-Based Modified (CBM) traffic conditioner. With this modification, complexity remains at the assured service capable host before the RIO buffer management scheme. To determine the dropping probability of an OUT packet, it is assumed that the traffic conditioner knows the amount of excess bandwidth and the average Round Trip Time (RTT) of all connections. Although it implies using some sort of signaling, it is more feasible than other proposed traffic conditioners such as [3], [4], [7], [8] or [9]. Along this paper we describe the CBM characteristics and study its performance with the RIO buffer management scheme throughout simulations. In addition, CBM accomplishment is compared with its precursor, the CB mechanism, and the two deeply studied algorithms TSW and LB. As we show later in simulation results, it is possible to afford fairness in the excess bandwidth sharing by using the CBM traffic conditioner without losing accuracy in assuring contracted rates.

The rest of this paper is organized as follows. Section 2 describes the CBM traffic conditioner implementation. Section 3 presents the scenarios and assumptions for simulations. In Section 4, simulation results are shown and discussed. The paper concludes in Section 5 summarizing the most important facts.

2 The Counters-Based Modified Traffic Conditioner

Assuming that all packets have a similar size, if all sources introduce the same number of out-of-profile packets into the network, then each source can get the same portion of excess bandwidth. This ideal behavior is affected by the odd characteristics of each TCP connection, like different RTT or target rates among others, and the interaction with the RIO buffer management scheme in the router. To overcome these influences, we suggest that connections that are sending OUT packets beyond their ideal fair quota should be penalized. This penalty is based on probabilistically dropping OUT packets in the traffic conditioner. The arising question is how to select what OUT packets should be dropped, and what ones should be added to the aggregate.

To solve it, we have studied the behavior of the excess bandwidth distribution from a different perspective. In simulation results, we have observed the number of out-of-profile packets generated between consecutive in-of-profile packet arrivals. Hence, we can state that:

- (i) A source with small target rate generates more OUT packets between two consecutive IN packets than a source with higher target. TCP sources transmit at link rate, so the smaller the target the more OUT packets are injected into the network. For this reason, these sources can get more network resources.
- (ii) The faster time response of the TCP sources with small RTT makes them inject more traffic, i.e., more OUT packets. Therefore, a source with small RTT is able to generate more OUT packets between two consecutive IN packets than a source with higher RTT.

An example illustrating this fact is depicted in Figures 1 and 2. Simulations to obtain these figures have been done using the CB algorithm and RIO. Eight sources generate TCP traffic at link rate, where each source has a contracted rate of 1-1-2-2-3-3-4 and 4 Mbps respectively. The RTT for each connection ranges from 10 to 80 ms at increments of 10 ms. The x-axis represents the time in seconds, and the y-axis the number of OUT packets between two consecutive IN packets. Observing both figures, the source with lower RTT and smaller target is injecting more OUT packets into the network. Furthermore, total number of OUT packets generated by source number 1, with target rate of 1 Mbps and RTT of 20 ms, is 6,031 packets; whereas for source number 7, with target rate of 4 Mbps and RTT of 80 ms, it is nearly the third part 2,331 packets.

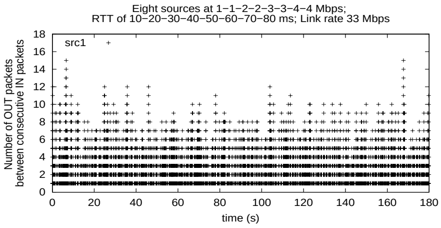

Fig. 1. OUT packets between IN packet tagging events for source 1 (6,031 OUT packets)

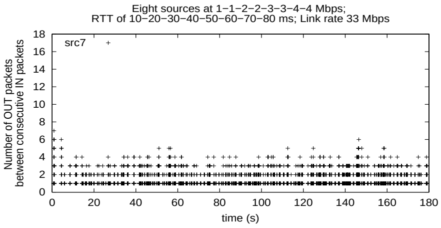

Fig. 2. OUT packets between IN packet tagging events for source 7 (2,331 OUT packets)

From these observations, the idea suggested in this paper is explained as follows (see Figure 3). The Counters-Based Modified (CBM) traffic conditioner, which is placed next to the TCP source (out of the reach of the final user), has a variable that counts the number of packets that have been marked as OUT between two consecutive IN packets. Every time a packet is marked as OUT, the CBM traffic conditioner checks this variable. If the variable does not exceed a minimum value *min*, then the OUT packet is injected into the network. If it exceeds a maximum value *max*, then the OUT packet is dropped. Finally, if the variable remains between *min* and *max*, the OUT packet is dropped with probability *p*.

To tune the *max* and *min* parameters we follow equations (1) and (2), where MSS stands for Maximum Segment Size. The excess bandwidth (BW_{excess}) could be seen as another TCP source whose maximum TCP window size is determined by the product BW_{excess} times $RTT_{average}$. Therefore we set the *max* limit to this value. A source that injects a number of OUT packets close to this limit would consume almost the entire excess bandwidth. In addition, if this limit is exceeded the source could even steal part of the guaranteed bandwidth, therefore the source cannot inject OUT packets beyond this *max* value. Another characteristic of this limit is that allows sources to increase the consumed excess bandwidth in case other sources finish their connections. In the extreme situation where only one source remains active it could use almost all the excess bandwidth, which is a reasonable behavior. It is well known that in TCP/IP, a simple additive increase and multiplicative decrease algorithm satisfies the sufficient conditions for convergence to an efficient state of the network, and it is used to implement congestion avoidance schemes. For this reason, a practical *min* value is half the *max* value.

```
Initially:
  Counter1=1
  Counter2=link_rate/target_rate
  Counter3=0
  Calculate the values for the
   probability p and the limits max
   and min

For each unit of time:
  Counter2--
  if counter2 <= 0
    Counter1++
    counter2=link_rate/target_rate
  if there is a packet arrival
                        (continue)

                              (continue)
if counter1>0
  packet marked as IN
  counter3=0
  counter1--
else
  packet marked as OUT
  counter3++
  if time>start_dropping_time
    if counter3>max
      OUT packet is dropped
    else if counter3>min
      OUT packet is dropped
      with probability p
    otherwise OUT packet
    accepted
```

Fig.3. Simplified pseudo-code of the CBM traffic conditioner algorithm

The estimation of RTT can be obtained by periodically signaling from the router device. The TCP protocol implements an algorithm that estimates the RTT of the current connection. This estimation is periodically sent to the router device, which calculates the average RTT. This value is then returned to the traffic conditioner, where packets are marked and/or dropped. Notice that per-flow state monitoring in the router is not required, in the sense that the router does not contain information on each individual active packet flow. It only has to periodically assess the RTT average with the information that receives from the TCP connections, and once performed, these values are not stored anywhere unlike traffic conditioner implementations from [3], [4], [8] and [9].

$$max = \left\lceil \frac{Bandwidth_{excess} \cdot RTT_{average}}{MSS} \right\rceil \tag{1}$$

$$min = \left\lceil \frac{max}{2} \right\rceil \tag{2}$$

The dropping probability *p* is shown in equation (3). Each source has a different value of *p*, between 0 and 1, based on its contracted rate. From statements (i) and (ii), it is intuitive to apply an equation in the form $p=1-x$, where x is *target_rate/link_rate*, thus connections with small target rates drop more OUT packets. However, once the *max* threshold is established, the traffic conditioner causes the lost of all OUT packets over the *max* limit. The fact of dropping a packet makes the source to slow down, so other sources can introduce more traffic into the network, that is more OUT packets.

If an equation that favors sources with large target rates is employed, then we are penalizing sources with small targets in excess; thus, when they recover from the lost, buffer resources are being consumed by sources with high targets. This situation causes new losses and makes sources with small targets to slow down again, originating the opposite effect stated in (i) and (ii), which is not desirable either. Therefore, we should use an equation for the dropping probability that gives a little more preference to connections with small targets.

We first evaluated a lineal equation such as $p=x$, and simulations showed that the CBM performed a fairer distribution of the excess bandwidth than the CB, albeit still away from the ideal behavior. To observe how the shape of the equation could influence the CBM performance, we conducted simulations with $p=2*x/(1+x)$ and $p=x/(2-x)$, two curves that give preference to connections with small targets over connections with high targets but in a non-linear way. It is important to state that small differences in p value may cause big performance differences because of the TCP congestion algorithm. From these results, we experienced that expression (3) is the most adequate equation in the performance of the CBM traffic conditioner. Notice that equation (3) is only applied when the number of OUT packets is in the interval (*min*, *max*).

$$p = 2 \cdot \frac{target\_rate/link\_rate}{1 + target\_rate/link\_rate} \qquad (3)$$

Finally, if the dropping process starts at the same time for all connections, then connections with larger RTT are adversely affected because of the slower time response. As a result, each traffic conditioner starts the process when a random multiple of its RTT has elapsed.

3 Scenario for Simulations

The topology selected for our simulations is illustrated in Figure 4. TCP traffic is generated by eight TCP Reno sources transmitting at the link rate, which has been set to 33 Mbps. The simulation tool used in this work for the sliding window protocol of TCP Reno sources was developed in [11], and was applied to validate the analytical study carried out in [12]. As a first insight, we employ a large packet size of 9,188 bytes, which corresponds to classical IP over ATM (e.g. Differentiated Services over MPLS, where the use of the ATM technology seems inherent).

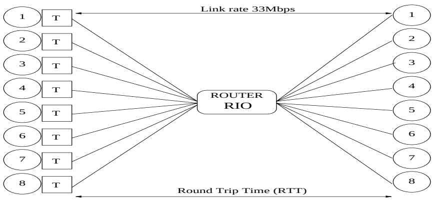

Fig. 4. Topology for simulations, the bottleneck is the router device (T≡Traffic Conditioner)

A router located inside the network, buffers and forwards the aggregated traffic. The queue management employs RIO with parameters [40/70/0.02] for IN packets and [10/40/0.2][1] for OUT packets. Weight_in and Weight_out RED parameters to calculate the average queue size were chosen equal to 0.002 as recommended in [13].

---

[1] [minth, maxth, maxp]

We consider five different scenarios in an undersubscribed situation (traffic load ≤ 60%), whose characteristics are included in Table 1. The oversubscribed scenario (traffic load > 60%) is less interesting in this study since the excess bandwidth represents a very small portion of the total available bandwidth. Simulation results have a confidence interval of 95% that has been calculated with a normal distribution function using 30 samples, with an approximate value of ±0.002 for all fairness calculations, and ±0.01 for the achieved target rates.

Table 1. Description of scenarios A through E employed in simulations (TR≡Target Rate)

	A	**B**	**C**	**D**	**E**
Link rate (Mbps)	33	33	33	33	33
TR src#0 to 7 (Mbps)	2.4	1-1-2-2-3-3-4-4	2.4	1-1-2-2-3-3-4-4	4-4-3-3-2-2-1-1
RTT src#0 to 7 (ms)	50	50	10-20-30-40-50-70-80	10-20-30-40-50-70-80	10-20-30-40-50-70-80
Σ target rates (Mbps)	19.2	20	19.2	20	20
BW_{excess} (Mbps)	13.8	13	13.8	13	13
$RTT_{average}$ (ms)	50	50	45	45	45
max (#OUT packets)	9	8	8	7	7
min (#OUT packets)	5	4	4	4	4

In scenario A, all connections have same RTT and same contracted rates, which makes this situation both ideal and infrequent in real frameworks. It is expected to obtain the best simulation results in this scenario, which has been usually studied in most papers. In scenario B, all connections have same RTT but different contracted rates. With the introduction of different target rates we try to be closer to a real environment with Quality of Service. Scenario C is the opposite of scenario B, since all connections have different RTT but same contracted rates, hence we can analyze the effect of the RTT on the CBM traffic conditioner performance.

Scenario D is the worst and most complex case under study. All connections have different RTT and different contracted rates, where sources with small targets have small RTT. Therefore, it implies that these TCP connections are favored as reflected in [3], [10] and [14]. Finally, in scenario E all connections have different RTT and different contracted rates (sources with small targets have large RTT). This is also a representative case, however assigning large round trip times to connections with small target rates avoids favoritism, as it occurs in scenario D.

4 Simulation Results

In this section, we present and discuss simulation results carried out in the scenarios described earlier. Firstly, it is shown how the new mechanism can control the number of OUT packets transmitted over the network leading to a fair share of the excess bandwidth. Moreover, we demonstrate that our scheme does not affect the CB performance presented in [10] regarding the in profile bandwidth assurance. We also present results of the interaction of Assured Service connections with Best-Effort connections competing for the outbound bandwidth.

4.1 OUT Packets Dropping

As indicated in Section 2, by setting the thresholds *max* (eq. 1) and *min* (eq. 2), and making use of the dropping probability *p* (eq. 3), the CBM traffic conditioner controls the number of out-of-profile packets injected into the network by each source. This effect can be observed in Figures 5 and 6. Simulations to obtain these figures have been done in scenario D (see Table 1), which is a usual environment in Internet because of the miscellaneous characteristics of each connection.

Figure 5 represents the number of OUT packets between consecutive IN packets arrivals using the CB traffic conditioner; that is, without applying the probabilistically dropping of out of profile packets. It corresponds to source number 1 with a target rate of 1 Mbps and a RTT of 20 ms. Figure 6 illustrates the improvement concerning Figure 5 in controlling the number of OUT packets introduced in the network when the proposed CBM algorithm is adopted.

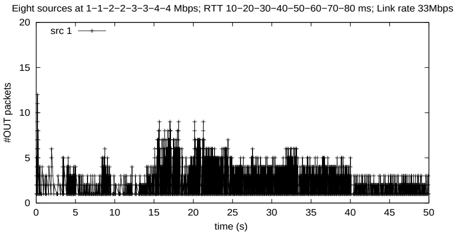

Fig. 5. OUT packets between two consecutive IN packets **without** OUT packet dropping in the traffic conditioner for source 1 (total OUTs=7,183 packets)

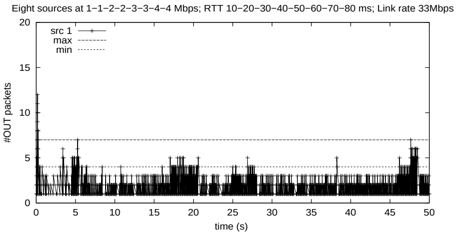

Fig. 6. OUT packets between two consecutive IN packets **with** OUT packet dropping in the traffic conditioner for source 1 (total OUTs=5,674 packets)

Comparing Figures 5 and 6, we observe that using the CBM algorithm we are able to obey sources with small targets and small RTT (e.g. source number 1) to generate less OUT packets. Likewise, with this mechanism we can increase the number of OUT packets injected into the network by connections with high target rates and large RTT. When the *max* number of OUT packets between two consecutive IN packets is exceeded, these packets are dropped. TCP connections reflect these drops slowing down, so more excess traffic (OUT packets) from other sources can be added to the aggregate. From these results, it can be presumed that the CBM traffic conditioner controls the number of out of profile packets that join the aggregate; hence, it can manage the sharing of excess bandwidth with the aim of providing a fair distribution as we illustrate in next section.

4.2 Fairness Index

To evaluate fairness we use the fairness index *f* shown in (4), where x_i is the excess throughput of source *i*, and *n* is the number of sources that compose the aggregate [15]. The closer to 1 in the *f* value, the more the fairness obtained.

$$f = \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n \cdot \sum_{i=1}^{n} x_i^2}\,;\ f \le 1 \tag{4}$$

Table 2 depicts the different *f* values obtained from simulations, and compares them in the same scenarios to other traffic conditioners that do not implement probabilistic OUT packets dropping (CB, TSW and LB). Simulations for the TSW and LB traffic conditioners have been carried out taking into consideration the performance evaluation study from [10]. This research includes a TSW configuration guide, since one of the disadvantages of the TSW algorithm is the difficulty in adjusting all the parameters involved on it. Slight variations in the values of the TSW or LB parameters cause relevant differences in simulation results.

Fairness indexes included in Table 2 reveal that it is possible to assure fairness in the excess bandwidth sharing with the CBM traffic conditioner, achieving an *f* value close to 0.95. Although the LB and TSW algorithms attain a high *f* value in scenarios A and B respectively, it should be noted that using these mechanisms inbound bandwidths are not guaranteed. Therefore, the underlying idea of keeping all connections sending a similar number of OUT packets is presented as a comparatively improvement in the development of traffic conditioners for the Internet Assured Service.

Table 2. Fairness index in the five different scenarios (TC≡Traffic Conditioner)

TC – RIO	Scenario A	Scenario B	Scenario C	Scenario D	Scenario E
CBM	0.997	0.969	0.942	0.899	0.923
CB	0.854	0.855	0.781	0.708	0.836
TSW	0.582	0.807	0.631	0.489	0.562
LB	0.853	0.687	0.740	0.817	0.832

4.3 Interaction of Assured Service Sources with Best-Effort Sources using CBM

In this subsection, best-effort (BE) sources compete with Assured Service (AS) sources for the available excess bandwidth. We use the topology shown in Figure 4, where the first four connections have an Assured Service and the last four connections belong to the best-effort class. The fact of being best-effort implies that all packets generated by these sources are considered as out of profile, and they do not have contracted target rates. We have conducted simulations for the five scenarios explained in Section 3 with slight modifications commented below. Link rate is kept at 33 Mbps.

In scenario A, the AS sources have a target rate of 5 Mbps, and all sources (included the BE ones) have a RTT of 50 ms. From equations (2) and (3), the limits *max* and *min* are 9 and 4 respectively. Ideally, each connection should get 1.625 Mbps of the excess bandwidth. Figure 7 depicts the achieved *goodput* of BE connections, where it is seen how these sources obtain nearly the same portion of the excess bandwidth after a transient interval. In this environment, we reach a fairness index of 0.906. Scenario B is like scenario A, but the four AS connections have contracted rates of 4-5-6 and 7 Mbps each. In this case, where thresholds *max* and *min* are 7 and 4 packets, the *f* value is nearly 0.87.

In scenario C, the AS sources have a target rate of 5 Mbps with RTT values that range from 10 ms to 40 ms at increments of 10 ms. Moreover, the BE connections have a RTT that varies from 50 ms to 80 ms at intervals of 10 ms. The limits *max* and *min* take a value of 7 and 4 packets. Figure 8 shows the *goodput* of BE sources in scenario C. In this situation, the excess bandwidth is 13 Mbps, thus the ideal *goodput* for BE connections is 1.625 Mbps. As depicted in this figure, BE sources achieve a *goodput* close to the ideal value with a difference of 0.5 Mbps between the maximum and minimum reached *goodputs*. The effect of having different values of RTT is hardly noticeable in the distribution of the outbound bandwidth, which is reflected in a fairness index of 0.847.

Finally, the most complex scenarios D and E also present an *f* value over 0.8. In scenario D, the four AS sources have contracted rates of 4-5-6 and 7 Mbps, and a

RTT that goes from 10 ms to 40 ms at intervals of 10 ms. The RTT for the BE sources ranges from 50 ms to 80 ms in increments of 10 ms. Scenario E only differs from D in the target rates of the AS connections, being in this case 7-6-5 and 4 Mbps. The limits *max* and *min* take a value of 6 and 3 packets in both cases. Figure 9 shows the *goodput* of BE sources in scenario E. In this case, the difference between the maximum and minimum reached *goodput* is about 0.5 Mbps, but worse than case C (Figure 8).

Figure 10 displays the achieved rates for IN packets in scenario D to remark that the existence of best-effort sources does not influence the AS sources regarding the contracted rates with CBM. When you want to offer higher-quality connections for some customers, you need tools to limit the effect of malicious users within the best-effort class. This type of users only generates out of control OUT packets that difficult the provisioning of a consistent network service. We show that the robustness of the couple CBM-RIO makes the entire service structure resistant to malicious users who try to maximize the bandwidth they attain from the network, since all AS sources get their target rates and also benefit from the excess bandwidth quite closely to the ideal behavior.

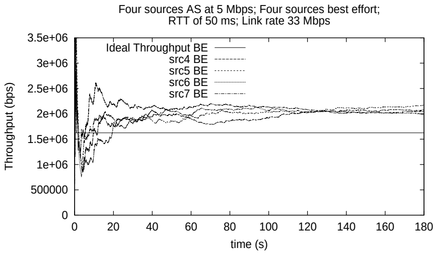

Fig. 7. *Goodput* (bps) for BE sources in scenario A

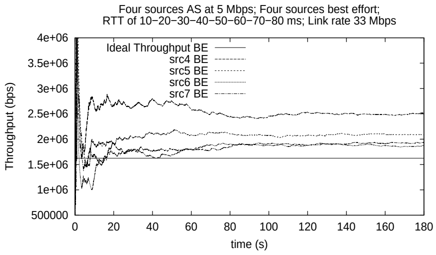

Fig. 8. *Goodput* (bps) for BE sources in scenario C

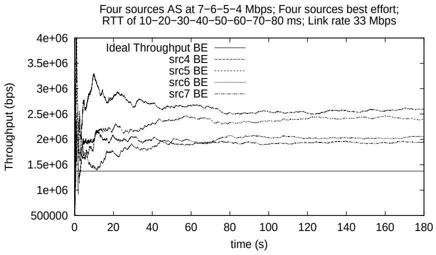

Fig. 9. *Goodput* (bps) for BE sources in scenario E

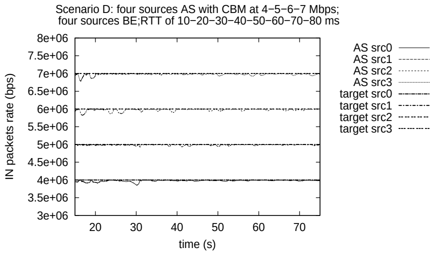

Fig. 10. Achieved rates for IN packets in scenario D, where AS and BE sources coexist

5 Conclusions

In this paper, we introduce a modification to the Counters-Based traffic conditioner that fulfills a fair distribution of the outbound bandwidth and guarantees target rates, called Counters-Based Modified (CBM). The CBM ability of controlling the number of out-of-profile packets that each source introduces in the aggregate helps to fair distribute outbound bandwidth, since excess bandwidth is occupied with this type of packets. The CBM traffic conditioner reaches this objective discarding out-of-profile packets before joining the aggregated, with a probability that depends on the target rate, the excess bandwidth and an estimation of the average RTT of all connections. We present simulation results in miscellaneous TCP environments (different target rates, different round trip times and share of resources with best-effort connections), showing that CBM can assure fairness in excess bandwidth sharing achieving a fairness index over 0.9. Results with CBM are also compared with other traffic conditioner implementations such as Time Sliding Window and Leaky Bucket, being illustrated that the CBM gets a comparatively better accomplishment. In addition, we have shown that in situations where Assured Service sources and best-effort sources coexist, the couple CBM-RIO is robust enough when possible best-effort users try to get more network resources than allowed. The high accuracy in guaranteeing the inbound bandwidth, the low complexity introduced, and the good value of the fairness index obtained in simulation results, lead us to believe that it is a feasible election in the Assured Service implementation with DiffServ.

Acknowledgements

This work was supported by the Spanish Research Council under grant FAR-IP TIC2000-1734-C03-03.

References

1. J. Heinanen, F. Baker, W. Weiss, J. Wroclawski, “Assured Forwarding PHB Group”, RFC 2597, June 1999.
2. 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.
3. W. Lin, R. Zheng, J. Hou, “How to make assured service more assured”, Proceedings of the 7th International Conference on Network Protocols (ICNP'99), pp. 182-191, Toronto, Canada, October 1999.
4. B. Nandy, N. Seddigh, P. Pieda, J. Ethridge, “Intelligent Traffic Conditioners for Assured Forwarding Based Differentiated Services Networks”, Proceedings of Networking 2000, Paris, France, pp.540-554, May 2000.
5. Elloumi O, De Cnodder S, Pauwels K, “Usefulness of the three drop precedences in Assured Forwarding Service”, Internet draft, work in progress, July 1999
6. M. Goyal, A. Durresi, P. Misra, C. Liu, R. Jain, “Effect of number of drop precedences in assured forwarding”, Proceedings of Globecom 1999, Rio de Janeiro, Brazil, Vol. 1(A), pp. 188-193, December 1999.
7. H. Kim, “A Fair Marker”, Internet draft, work in progress, April 1999.
8. I. Alves, J. De Rezende, L. De Moraes, “Evaluating Fairness in Aggregated Traffic Marking”, Proceedings of IEEE Globecom’2000, San Francisco, USA, pp. 445-449, November 2000.
9. I. Andrikopoulos, L. Wood, G. Pavlou, “A fair traffic conditioner for the assured service in a differentiated services internet”, Proceedings of IEEE International Conference on Communications ICC2000, New Orleans, LA, Vol. 2, pp. 806-810, June 2000.
10. Maria-Dolores Cano, Fernando Cerdan, Joan Garcia-Haro, Josemaria Malgosa-Sanahuja, “Performance Evaluation of Traffic Conditioner Mechanisms for the Internet Assured Service”, in Quality of Service over Next-Generation Data Networks, Proceedings of SPIE Vol. 4524, pp. 182-193, 2001.
11. F. Cerdan, O.Casals, “Performance of Different TCP Implementations over the GFR Service Category”, ICON Journal, Special Issue on QoS Management in Wired &Wireless Multimedia Communications Network, Vol.2, pp.273-286, Baltzer Science.
12. V. Bonin, O.Casals, B. Van Houdt, C. Blondi, “Performance Modeling of Differentiated Fair Buffer Allocation”, Proceedings of the 9th International Conference on Telecommunications Systems, Dallas, USA.
13. S. Floyd and V. Jacobson, “Random Early Detection Gateways for Congestion Avoidance”, IEEE/ACM Transactions on Networking, Vol. 6 No 4, pp. 397-413, August 1993.
14. N. Seddigh, B. Nandy, P.Pieda, “Bandwidth Assurance Issues for TCP flows in a Differentiated Services Network”, Proceedings of IEEE Globecom’99, Vol. 3, pp. 1792-1798,Rio de Janeiro, Brazil, December 1999.
15. R. Jain, “The Art of Computer Systems Performance Analysis”, John Wiley and Sons Inc., 1991.