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 5368
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. Many research papers have focused on finding the best traffic conditioner able to assure contracted target rates and to fairly distribute the excess bandwidth among competing sources. Nevertheless, none of the proposals presented so far accomplishes simultaneously both features. 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 Reno sources. The ability of strictly providing the inbound bandwidth is inherited from its predecessor the Counters-Based algorithm, and 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, the amount of excess bandwidth and the average RTT of all connections in the traffic conditioner have to be known. Although this fact implies using some sort of signaling, 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. Simulation results presented in this paper lead us to suggest it as a feasible election for the traffic conditioner device implementation in DiffServ.

Keywords: traffic conditioner, RIO, Assured Service, DiffServ, QoS

1 Introduction

Differentiated Services (DiffServ) paradigm [1] has been standardized as one of the promising solutions for providing QoS in IP networks. The DiffServ architecture tries to create a simple scheme that provides a range of QoS levels by moving complexity toward the edge of the network. A group of mechanisms to treat packets of aggregated flows with different priorities according to the information carried in the DiffServ field of the IP packet header is conceived; thus, packets are classified and marked to receive a particular treatment on the nodes along their path. This treatment is known as per-hop behavior (PHB). Complex classification and conditioning functions (metering, marking, shaping) need only to be implemented at boundary nodes. Meanwhile, interior nodes perform a set of forwarding PHBs to aggregates of traffic that have been appropriately marked.

Two PHBs are currently being standardized by the IETF, the Expedited Forwarding per-hop behavior (EF PHB) [2] and the Assured Forwarding per-hop behavior (AF PHB) [3]. The idea behind AF PHB is to assure a minimum throughput (the 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. As detailed in [3], within each AF class an IP packet can be assigned one of three different levels of drop precedence. In case of congestion, the drop precedence of a packet determines the relative importance of the packet within the AF class. A congested DiffServ node tries 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 or out-profile bandwidth along this paper.

A significant step in providing the Internet Assured Service was the introduction of RIO (RED (Random Early Detection) In and Out) in [4]. Once a packet is marked as in of profile (IN) or out of profile (OUT), the aggregate composed of different flows reaches the router device. In this device, the RIO buffer management scheme is applied. RIO is the combination of two RED [5] algorithms with different drop probability curves, so that OUT packets are more likely to be discarded. RIO uses a single FIFO queue to service both IN and OUT packets. The probability of dropping an OUT packet depends on the total

number of packets arriving to the node, while the probability of dropping an IN packet depends exclusively on the buffer occupancy of IN packets.

Together with an adequate buffer management scheme choice, traffic conditioners play a key role in implementing the Assured Service approach. Abundant literature has been written about the couple traffic conditioner-buffer management scheme that better accomplish the Assured Service task. Basically, to guarantee the contracted rates and to fairly distribute the excess bandwidth. 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.

One of the most studied traffic conditioners is the Time Sliding Window (TSW) introduced in [4]. It consists of a rate estimator and a marker. The couple TSW-RIO manifests some design difficulties like favoring those connections with lower Round Trip Time (RTT). Furthermore, the assured bandwidth with TSW-RIO is well guaranteed for TCP Sack, which is an unusual TCP implementation in Internet, but not TCP Reno. Nevertheless, the authors do not present results about distribution of outbound bandwidth among sources.

Token-based traffic conditioner mechanisms have been also used for Assured Service based on DiffServ implementations. Differing from the conclusions illustrated in [4], Ibanez and Nichols stated in [6] that token bucket based mechanisms are superior to the TSW due to the capability of transmitting deterministic bursts of IN packets. The paper, which was presented as a first step in studying Assured Service applicability over the entire Internet, also states that inbound bandwidth assurance is too dependent of network parameters such as RTT or target rates to ensure the expected QoS. Consequently, an Assured Service cannot offer a quantifiable service to TCP traffic. Simulations were performed using RIO and the token bucket algorithm.

At this point, two tendencies were followed for the traffic conditioner mechanism designs. The first one based on rate estimator algorithms, and the second one based on token mechanisms. As an example of the first trend, W. Lin *et al.* introduced in [7] an enhanced version of the TSW traffic conditioner (ETSW). When compared with the TSW-RIO performance, the ETSW-RIO behaves better. However, the in-profile bandwidth is worse than the obtained when using the token bucket. Regarding the fair distribution of the excess bandwidth, the paper presents two modifications of RIO, the adaptive-RIO and the dynamic-RIO. The former does not work properly when sources have different RTT. The latter needs to keep information about the flows that have OUT packets in the buffer, hence, requiring some additional memory in the router, and likely causing scalability problems.

Another example following the same trend is the *intelligent* traffic conditioner introduced in [8] by B. Nandy *et al*. The paper suggests using a mechanism that calculates the dropping probability of a packet as a function of the measured RTT of the aggregate related to the minimum RTT of the DiffServ domain. The traffic conditioner is used combined with the three-color version of RIO studied in [9]. With the aim of providing fairness in the distribution of the excess bandwidth, another parameter is introduced in the *intelligent* traffic conditioner aware of the minimum target rate of the aggregate as well as the aggregate target rate. Nevertheless, this knowledge requires communication between devices. In addition, the set of assumptions for this scheme is too restrictive (TCP flows operate in congestion avoidance, all flows in the aggregate have the same RTT, etc.).

Following the second trend, it was proposed the use of more than two drop precedences in AF-PHB [10] [11]. By coloring TCP packets with red, yellow and green, and UDP packets with only red and green, it is possible to achieve a fair allocation of excess network bandwidth between responsive and non responsive sources. Using the RED proposal for colored traffic as buffer management scheme, simulation results in [12] show some improvements in outbound bandwidth fair sharing when applying this new scheme, but the problem remains when only TCP sources coexist.

Likewise, H. Kim presented in [13] a *fair* marker. This marker belongs to the per-aggregation flow aware type, i.e., it is based on partial knowledge of individual flows. Simulation results of this traffic conditioner combined with FRED [14] are presented in [15]. This work shows that with an adequate tune of parameters, fairness is achieved in terms of in-profile bandwidth portion, but it cannot provide fairness in the allocation of excess bandwidth. To overcome the problem of sharing excess bandwidth, [15] and [16] propose a very similar modification of the *fair* marker. Simulation results in both studies show that the

modifications over perform comparing with the initial Kim's *fair* marker or the traditional token bucket mechanism. Note that simulations with different RTT for each source were not conducted. Also, if there is a great number of active flows the scalability problem shows up again.

Finally, the Counters-Based (CB) traffic conditioner developed in [17] 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. 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*. From the comparative simulation study carried out in [17], this traffic conditioner together with RIO over performs the 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 presents same 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 [17], 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 RTT of all connections. Although it implies using some sort of signaling, it is more feasible than other proposed traffic conditioners such as [7][8][13][15][16]. 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 Time Sliding Window (TSW) and Leaky Bucket (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 details the characteristics of the CBM traffic conditioner implementation. In Section 3, we present the scenario and assumptions for carrying out simulations. In Section 4, simulation results are presented and discussed. The paper concludes in Section 5 summarizing the most important facts.

2 The Counters-Based Modified Traffic Conditioner

As commented above, the CB traffic conditioner does not alleviate the fair share of the outbound bandwidth among TCP connections. In consideration, it presents the advantage of not having any configuration parameter and strictly assuring contracted rates in miscellaneous scenarios. In this paper, we modify the CB algorithm, fulfilling expectations for the fair distribution of the excess bandwidth among competing sources, keeping their contracted rates assured.

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. Measurements are taken every time a packet is tagged IN or OUT. 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).

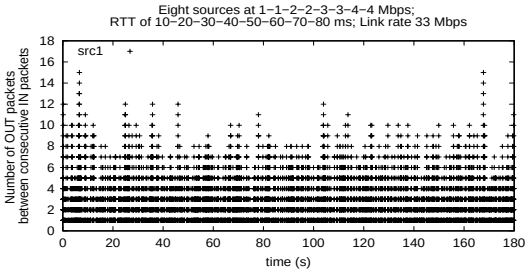

Fig. 1. OUT packets between IN packet tagging events for source 1 (6,031 OUT packets).

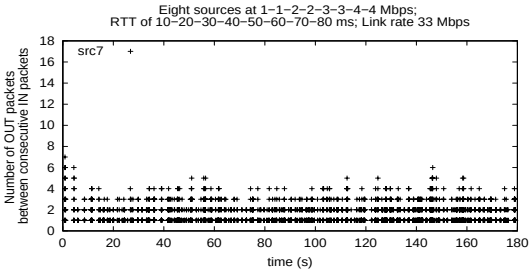

Fig. 2. OUT packets between IN packet tagging events for source 7 (2,331 OUT packets).

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*.

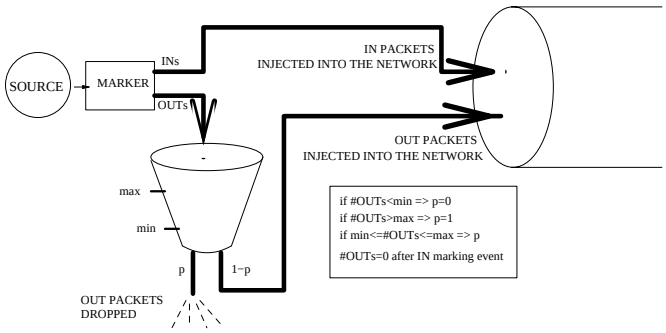

Fig.3. How to drop OUT packets in the CBM traffic conditioner (# means *number of*).

To apply this mechanism three questions need to be answered. The first one is how to set the *max* and *min* threshold parameters, the second one is what equation determines the dropping probability *p*, and the third one is when to start the dropping process.

Firstly, to tune the *max* and *min* parameters we follow equations (1) and (2), where MSS stands for Maximum Segment Size. The excess bandwidth 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 source can 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.

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

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

The 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 [7] [14] and [15].

On the other hand, 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 evaluate 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-lineal 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. All equations are plotted in Figure 4. Notice that equation (3) is only applied when the number of OUT packets is in the interval (*min*, *max*).

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

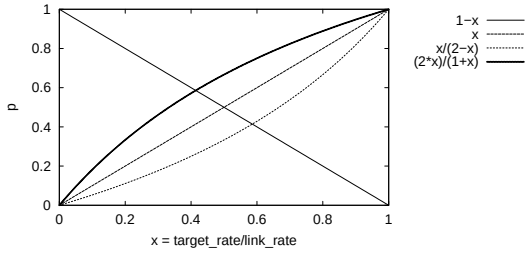

Fig.4. Equations tested to find the most adequate dropping probability functions.

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. The simplified pseudo-code of the entire CBM algorithm is written in Figure 5.

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

Fig.5. Simplified pseudo-code of the CBM traffic conditioner algorithm.

In summary, from equations (1), (2) and (3) the only additional information required by the traffic conditioner is the ratio *target_rate/link_rate*, the excess amount of bandwidth, and an estimation of the RTT average delay of all connections. In the next sections, we will observe that with a proper election of the *max* and *min* parameters is possible to balance the traffic connections to get a fair excess bandwidth distribution and to assure target rates.

3 Scenario for Simulations

The topology selected for our simulations is illustrated in Figure 6. TCP traffic is generated by eight TCP Reno sources transmitting at the link rate, which has been set to 33 Mbps. To verify the impact of target rates, different values are used along the simulations. We also measure the influence of different RTTs. In the TCP homogeneous scenario (same RTT for all connections), round trip delay between sources and destinations is set to 50 ms. In the TCP heterogeneous scenario, this value varies from 10 ms to 80 ms at increments of 10 ms.

The simulation tool used in this work for the sliding window protocol of TCP Reno sources was developed in [18], and has been extensively utilized in [19] and [20]. In addition, it was applied to validate the analytical study carried out in [21]. Some characteristics of this simulation tool are: TCP sources have been selected as greedy for a worst case to achieve a relative high network congestion state, destinations only send acknowledgements, which are never lost or delayed, and the maximum window size equals the product bandwidth delay as usual for WAN environments.

As a first insight, we employ a large packet size of 9,188 bytes, which corresponds to classical IP over ATM. Other packet sizes are used along simulations, as it is indicated in the text. The reason to use different packet sizes is to study miscellaneous work environments. For instance, a simulation configuration with large packets could represent Differentiated Services over MPLS, where the use of the ATM technology seems inherent. Likewise, a packet size of 1,500 bytes represents a typical packet size for Ethernet-attached hosts [22].

A router located inside the network, buffers and forwards the aggregated traffic. The queue management employs RIO, i.e., twin RED algorithms to preferentially drop OUT packets. The RIO parameters are

[40/70/0.02] for IN packets and [10/40/0.2][1] for OUT packets. Weight_in and Weight_out RED parameters used to calculate the average queue size have been chosen equal to 0.002 as recommended in [5].

We consider five different scenarios in an undersubscribed situation (traffic load ≤ 60%). 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.

Scenario A. All connections have same RTT and same contracted rates, which makes this situation both ideal and infrequent in real frameworks. Simulations have been carried out with target rates of 2.4 Mbps and RTT of 50 ms for all connections. The limits *max* and *min* have been set to 9 and 5 packets respectively, which have been calculated using equations (1) and (2) ($Bandwidth_{excess}$ = 13.8 Mbps; $RTT_{average}$ = 50 ms; MSS = 9,188 bytes). It is expected to obtain the best simulation results in this scenario, which has been usually studied in most papers.

Scenario B. All connections have same RTT and different contracted rates. With the introduction of different target rates we try to be closer to a real environment with QoS [24]. Simulations have been conducted under target rates of 1, 1, 2, 2, 3, 3, 4 and 4 Mbps respectively, and RTT of 50 ms for all connections. From equations (1) and (2) ($Bandwidth_{excess}$ = 13 Mbps; $RTT_{average}$ = 50 ms; MSS = 9,188 bytes), the limits *max* and *min* have been set to 8 and 4 packets respectively.

Scenario C. All connections have different RTT and same contracted rates. This scenario is the opposite of scenario B, hence we can analyze the effect of the RTT on the CBM traffic conditioner performance. Simulations have been done with target rates of 2.4 Mbps for all connections, and RTT from 10 to 80 ms at increments of 10 ms. The limits *max* and *min* have been set to 8 and 4 packets, respectively ($Bandwidth_{excess}$ = 13.8 Mbps; $RTT_{average}$ = 45 ms; MSS = 9,188 bytes).

Scenario D. All connections have different RTT and different contracted rates (sources with small targets have small RTT). This is the worst and most complex case under study, because connections with small contracted rates also have small round trip times, which implies these TCP connections being favored as reflected in [7], [17] and [23]. Simulations have been carried out with target rates of 1, 1, 2, 2, 3, 3, 4 and 4 Mbps, and RTT from 10 to 80 ms at increments of 10 ms. The limits *max* and *min* have been set to 7 and 4 packets respectively ($Bandwidth_{excess}$ = 13 Mbps; $RTT_{average}$ = 45 ms; MSS = 9,188 bytes).

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. Simulations have been run with target rates of 4, 4, 3, 3, 2, 2, 1 and 1 Mbps, and RTT of 10 to 80 ms at increments of 10 ms. The limits *max* and *min* have been set to 7 and 4 packets respectively ($Bandwidth_{excess}$ = 13 Mbps; $RTT_{average}$ = 45 ms; MSS = 9,188 bytes).

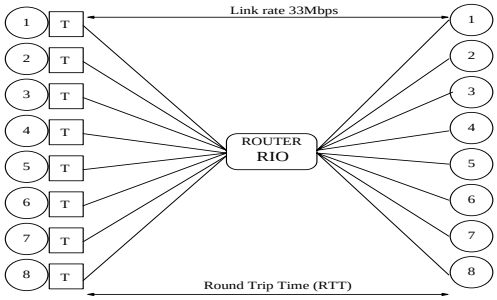

Fig. 6. Topology used in simulations where the bottleneck is the router device (T≡Traffic Conditioner).

[1] [minth, maxth, maxp]

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 new scheme does not affect the CB performance presented in [17] 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. Finally, we have studied the effect of having different packet sizes in the aggregate.

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 7 to 10. Simulations to obtain these figures have been done in scenario D, which is a usual environment in Internet because of the miscellaneous characteristics of each connection.

On one hand, Figures 7 and 9 represent 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. Figure 7 corresponds to source number 1 with a target rate of 1 Mbps and a RTT of 20 ms; whereas Figure 9 corresponds to source number 7 with a contracted rate of 4 Mbps and RTT of 80 ms.

On the other hand, Figures 8 and 10 illustrate the improvement concerning Figures 7 and 9 in controlling the number of OUT packets introduced in the network when the proposed CBM algorithm is adopted. In this scenario, the available excess bandwidth is 13 Mbps, the average RTT is 45 ms and the MSS is set to 9,188 bytes. Consequently, the limits *max* and *min* are set to 7 and 4 (see equations (1) and (2)). The dropping probability *p* is 0.059 for source number 1 (Figure 8), and 0.216 for source number 7 (Figure 10).

Comparing Figures 7 and 8, 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 (e.g. source number 7), as plotted in Figures 9 and 10. 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.

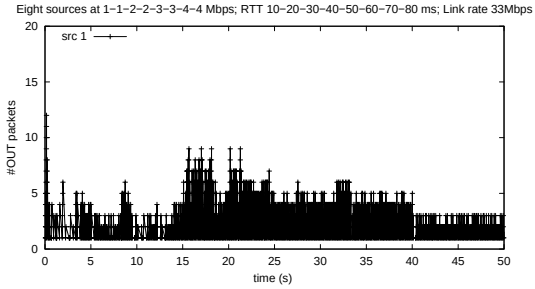

Fig. 7. OUT packets between two consecutive IN packets **without** OUT packet dropping in the traffic conditioner for source 1 (total OUTs=7,183 packets).

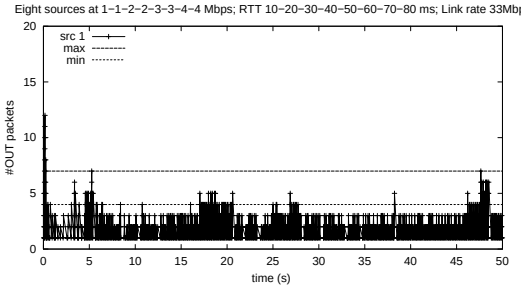

Fig. 8. OUT packets between two consecutive IN packets **with** OUT packet dropping in the traffic conditioner for source 1 (total OUTs=5,674 packets).

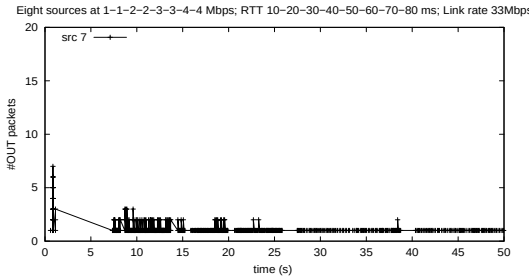

Fig.9. OUT packets between two consecutive IN packets **without** OUT packet dropping in the traffic conditioner for source 7 (total OUTs=435 packets).

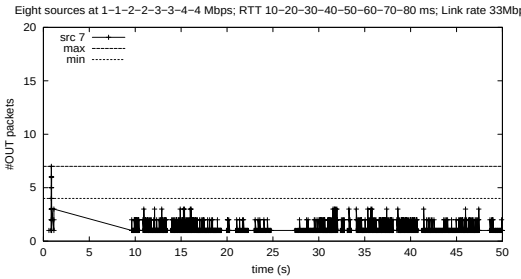

Fig. 10. OUT packets between two consecutive IN packets **with** OUT packet dropping in the traffic conditioner for source 7 (total OUTs=2,781 packets).

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 [25]. The closer to 1 in the *f* value, the more the fairness obtained. We use the term throughput meaning *goodput* in calculations of the fairness index.

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

Table 1 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 traffic conditioner have been carried out taking into consideration the performance evaluation study from [17]. 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 parameters would cause relevant modifications in simulation results. The LB has been also configured following recommendations from the same paper [17]. Experimental results got with this algorithm are also sensitive to variations in its configuration values.

Fairness indexes included in Table 1 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 remembered 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 1. Fairness index in five different scenarios.

Traffic conditioner – 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

In addition, Table 2 is included to confirm that the dropping of OUT packets in the CBM traffic conditioner along with its interaction with the RIO buffer management scheme, does not affect the inbound bandwidth assurance. The CB traffic conditioner presented in [17] hard guarantees contracted rates, but it performs an unfair distribution of the outbound bandwidth. However, its simplicity makes it reasonable for a traffic conditioner implementation. The CB algorithm plus the modification introduced in this paper originate the CBM traffic conditioner, which inherits the attribute of strictly guaranteeing target rates to the TCP connections as shown in Table 2. As it is illustrated in this table, the contracted rates are assured for all connections, with variations that do not exceed 1% of the target rates.

Table 2.Achieved rate in Mbps for IN packets in five different scenarios using CBM-RIO.

	Achieved rate for IN packets (Mbps)				
Source	**Scenario A**	**Scenario B**	**Scenario C**	**Scenario D**	**Scenario E**
0	2.39	1.00	2.39	1.00	3.99
1	2.39	1.00	2.39	0.99	4.00
2	2.40	1.99	2.40	1.99	2.99
3	2.40	1.99	2.39	2.00	2.99
4	2.39	3.00	2.40	2.99	1.99
5	2.39	2.99	2.40	2.99	2.00
6	2.40	3.99	2.39	4.00	1.00
7	2.39	3.99	2.39	3.99	0.98

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. The topology used for this set of simulations consists of eight TCP Reno connections, 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) ($Bandwidth_{excess}$ = 13 Mbps; $RTT_{average}$ = 50ms; MSS = 9,188 bytes), the limits *max* and *min* are 9 and 4 respectively. Ideally, each connection should get 1.625 Mbps of the excess bandwidth. Figure 11 depicts 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 equal to 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 ($Bandwidth_{excess}$ = 11 Mbps; $RTT_{average}$ = 50 ms; MSS = 9,188 bytes), the *f* value is nearly 0.87.

In scenario C, the AS sources have a target rate of 5 Mbps. The RTT value ranges from 10 ms to 40 ms at increments of 10 ms for the AS connections. 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 respectively ($Bandwidth_{excess}$ = 13 Mbps; $RTT_{average}$ = 45 ms; MSS = 9,188 bytes). Figure 12 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 scenarios ($Bandwidth_{excess}$ = 11 Mbps; $RTT_{average}$ = 45 ms; MSS = 9,188 bytes). Figure 13 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 farther from the ideal value than case C (Figure 12).

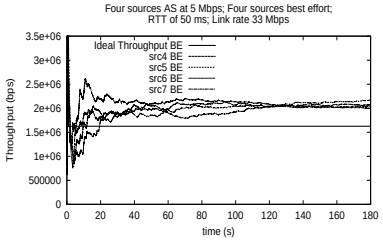

Fig. 11. *Goodput* (bps) for BE sources in Scenario A.

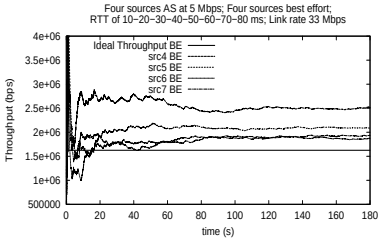

Fig. 12. *Goodput* (bps) for BE sources in Scenario C.

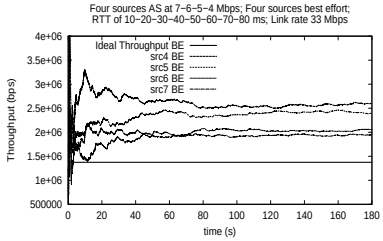

Fig. 13. *Goodput* (bps) for BE sources in Scenario E.

Figure 14 illustrates the good performance accomplished with the CBM traffic conditioner in the distribution of the excess bandwidth when AS connections and best-effort connections coexist. Figure 15 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. 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. Even in heterogeneous scenarios (AS sources with best-effort sources, different target sizes and distinct RTT), the CBM manages to fulfill contracted rates and to fairly allot the outbound bandwidth among sources.

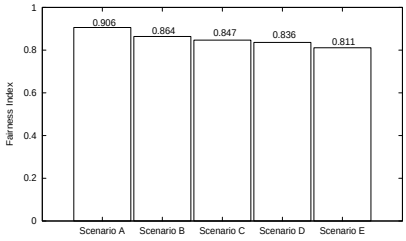

Fig. 14. Fairness index for different scenarios where AS and BE sources coexist.

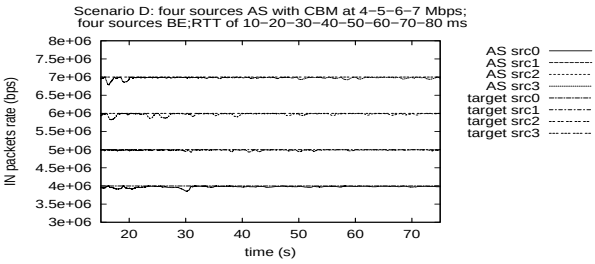

Fig. 15. Achieved rates for IN packets in scenario D where AS and BE sources coexist.

4.4 Packet Size Dependency

To observe how can affect variations in the packet size over the performance of our traffic conditioner proposal in terms of biasing in the excess bandwidth distribution, we have carried out a set of simulations for scenarios A through E, and packet sizes of 1,500, 5,300 and 9,188 bytes. The use of DiffServ over MPLS has become a hot research topic in recent years, thus we use a large packet size of 9,188 to represent IP over ATM technology that is usually utilized in combination with MPLS. The 1,500 bytes packet size stands for typical Ethernet hosts. The 5,300 byte-size does not belong to any known service, though it is used to study the trend of the fairness index as the packet size decreases. As observed in Table 3, the index fairness value remains over 0.8 for nearly all cases with the CBM mechanism. Reduction in the *f* value for a packet size of 1,500 bytes could be fixed readjusting the *max* and *min* thresholds, and it would require further study.

Table 3. Fairness index in CBM with packet size variations and RIO.

Packet size (bytes)	Scenario A	Scenario B	Scenario C	Scenario D	Scenario E
1,500	0.821	0.823	0.700	0.601	0.795
5,300	0.905	0.878	0.981	0.937	0.981
9,188	0.997	0.969	0.942	0.899	0.923

In the description of the CBM algorithm, we assumed that all packets have the same size, and we applied this assumption all along this paper. Thereby, it could be interesting to observe variations in the outbound bandwidth share having different packet sizes. We have employed three packet sizes, 9,188 bytes (sources 0, 3 and 6), 5,300 bytes (sources 1, 4 and 7) and 1,500 bytes (sources 2 and 5). Simulations results obtained for the fairness index are shown in Table 4. It is seen a reduction around 0.2 in the CBM fairness index comparing these values to previous results, which means that simultaneous different packet sizes in the network produce a worse allocation of the excess bandwidth among sources. These results agree with the ones obtained in [23]. One possible reason to explain this misbehavior is the configuration of CBM-RIO parameters that need a more detailed study out of the scope of this paper. Nevertheless, the fairness index stays over 0.7 despite of the miscellaneous situation that confirms the robustness of the CBM traffic conditioner.

Table 4. Fairness index under simultaneous different packet sizes and RIO.

Type of traffic conditioner – RIO	Scenario A	Scenario B	Scenario C	Scenario D	Scenario E
CBM	0.713	0.731	0.770	0.775	0.701

5 Conclusions

In this paper, we introduce a modification to the Counters-Based traffic conditioner that fulfills a fair distribution of the outbound bandwidth, called Counters-Based Modified (CBM). The CBM mechanism, used together with the RIO buffer management scheme, smoothes the effect of different contracted target rates, round trip times, or packet sizes on TCP connections, guaranteeing target rates and allocating excess bandwidth equitably among competing sources. In addition, we have shown that in situations where Assured Service sources and best-effort sources coexist the couple CBM-RIO avoids misbehaviors of possible best-effort users trying to get more network resources than allowed.

The 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, different packet sizes 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. 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. S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, “An Architecture for Differentiated Services”, RFC 2475, December 1998.
2. V. Jacobson, K. Nichols, K. Poduri, “An Expedited Forwarding PHB”, RFC 2598, June 1999.
3. J. Heinanen, F. Baker, W. Weiss, J. Wroclawski, “Assured Forwarding PHB Group”, RFC 2597, June 1999.
4. 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.
5. 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.
6. J. Ibañez, K. Nichols, “Preliminary simulation evaluation of an assured service”, Internet draft, work in progress, draft-ibanez-diffserv-assured-eval-00.txt, August 1998.
7. 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.
8. 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.
9. Elloumi O, De Cnodder S, Pauwels K, “Usefulness of the three drop precedences in Assured Forwarding Service”, Internet draft, work in progress, July 1999
10. J. Heinanen, R. Guerin, “A single rate three color marker”, RFC 2698, September 1999.
11. J. Heinanen, R. Guerin, “A two rate three color marker”, RFC 2698, September 1999.
12. 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.
13. H. Kim, “A Fair Marker”, Internet draft, work in progress, April 1999.
14. D.Lin, R. Morris, “Dynamics of Random Early Detection”, Proceedings of ACM SIGCOMM’97, pp. 127-137, Cannes, France, September 1997.
15. 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.
16. 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.
17. 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.
18. 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.

19. F.Cerdan, O.Casals, “Mapping an Internet Assured Service on the GFR ATM Service”, Lecture Notes in Computer Science 1815 (Networking 2000), pp. 398-409, Springer-Verlag.
20. V. Bonin, F. Cerdan, O.Casals, “A simulation study of Differential Buffer Allocation”, Proceedings of 3rd International Conference on ATM, ICATM’2000, Germany, June 2000.
21. 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.
22. K. Thompson, G Miller, R Wilder, “Wide-Area Internet Traffic Patterns and Characteristics”, IEEE Network Magazine, Vol. 11 No. 6, pp. 10-23, November/December 1997.
23. N. Seddigh, B. Nandy, P.Pieda, “Bandwidth Assurance Issues for TCP flows in a Differentiated Services Network”, Proceedings of IEEE Globecom’99, Rio de Janeiro, Brazil, Vol. 3, pp. 1792-1798, December 1999.
24. F. Cerdan, J. Malgosa-Sanahuja, J. Garcia-Haro, F. Burrull, F. Monzo-Sanchez, “Quality of Service for TCP/IP Traffic: An overview”, Proceedings of PROMS’02, pp.91-99, Cracow 2002.
25. R. Jain, “The Art of Computer Systems Performance Analysis”, John Wiley and Sons Inc., 1991.