END-TO-END TCP PERFORMANCE OF THE COUPLE CBM TRAFFIC CONDITIONER AND RIO BUFFER MANAGEMENT IN A THREE-NODE TOPOLOGY

Maria-Dolores Cano, Fernando Cerdan
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 325953 Fax +34 968 325338
{mdolores.cano, fernando.cerdan}@upct.es

ABSTRACT

Despite the abundant literature written about the AF PHB, no solution has been found to efficiently face up its two goals, assuring a minimum rate to the users and offering a fair distribution of the excess bandwidth if available. The Counters Based Modified (CBM) traffic conditioner, presented in a previous work, is able to achieve these objectives in single-node topologies. This paper raises issues with providing bandwidth assurance and spare bandwidth distribution for TCP flows in more complex topologies than usual. Simulation results explore the effect of target rates, round trip times, and efficiency of CBM when up to three network nodes implement service differentiation, including in some cases the coexistence of assured service and best-effort traffics.

KEY WORDS

Traffic Conditioner, Assured Service, Differentiated Services, QoS

1. Introduction

The Assured Forwarding Per-Hop Behavior (AF-PHB) [1] is one of the IETF PHBs for Differentiated Services (DiffServ) with the status of proposed standard. The idea behind AF-PHB is to ensure a minimum throughput (usually the contracted target 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. It should be remarked that DiffServ mechanisms are not implemented to provide an end-to-end service. However, from many points of view (engineering, users, etc.) it is more interesting the study of end-to-end performance of TCP connections in terms of throughput (excluding retransmitted packets, which is usually called *goodput*).

Despite the abundant literature written about the AF PHB (e.g. [2] to [9]), no solution has been found to efficiently face up its two goals, assuring the contracted bandwidth and offering a fair distribution of the excess bandwidth if available. Two different concepts can be understood as fairness in the excess bandwidth sharing. The first considers fairness as the even distribution of spare bandwidth among all connections that compose the aggregate. The second defines fairness as a proportional distribution of the spare bandwidth with respect to the contracted rate. In this paper we adopt the first definition, which has been mostly used. Although current trends try to implement mechanism with proportional fairness, we should consider that a proportional fairness might not be seen as “fair” by the user if the difference among contracts is significant (e.g. $user_1$ contracts 1 Mbps and $user_2$ contracts 20 Mbps). Since in this case, users with very small contracts hardly benefit from excess bandwidth.

Traffic conditioners play an important role in DiffServ. Algorithms used to condition the traffic that enters the network determine the treatment that these packets will receive in future. We demonstrate in a previous work [10] that when conditioning traffic with the Counters Based Modified algorithm, users obtain their contracted target rates. In addition, the available excess bandwidth was distributed evenly among all the sources that compose the aggregate. We focused that work in a single bottleneck topology with TCP Reno sources.

In this paper, we present a performance analysis of the Counters-Based Modified traffic conditioner in miscellaneous three-node topologies that try to represent more realistic situations, which is a notably more complex and heterogeneous topology than the usual one employed in the related literature. Considering that a number of six nodes is big enough for a DiffServ domain, in our study we use three nodes whose results and conclusions can be extended to a bigger domain.

The rest of this paper is organized as follows. In Section 2 we analyze previous work and explain the Counters Based Modified operation. Section 3 describes the topology, 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. Previous Work

The Counters-Based (CB) traffic conditioner developed in [11] performs comparatively better than other traffic conditioners. This mechanism based on counters guarantees the users' contracted target rates in scenarios with variable round trip times (RTT) and different target rates. Its easy configuration and high accuracy makes it suitable for general use. Only two counters are needed to implement this algorithm, 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 timer expires. From the comparative simulation study carried out in [11], this traffic conditioner together with RIO (RED (Random Early Detection) *In* and *Out*) [2] works better than the two classical mechanism Time Sliding Window [2] (TSW)-RIO and Leaky Bucket (LB)-RIO in terms of guaranteeing contracted target rates. Nevertheless, it also presents problems regarding the excess bandwidth sharing among sources as previous proposals.

The Counters Based Modified (CBM) traffic conditioner was introduced in [10] to overcome the lack of fairness in the excess bandwidth distribution. The starting point of this modification is based on the idea that if all sources introduce the same number of out-of-profile (*out*) packets into the network (assuming all packets have a similar size), 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 round trip times or target rates among others, and the interaction with the RIO buffer management scheme in the routers. To confront these influences, it was suggested that connections that are sending *out* packets beyond their ideal fair quota should be penalized. This penalty was based on probabilistically dropping *out* packets in the traffic conditioner.

In [10] it is shown that connections with small target rates and low delays generate more *out* packets between consecutive in-profile (*in*) packets than other connections, thus consuming more network resources. From these observations, the CBM was developed to work as follows. Placed next to the TCP source (out of the reach of the final user), a variable 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.

Accordingly, to employ CBM it is necessary to configure the *max* and *min* thresholds as well as to calculate the dropping probability p. It is explained in [10] how *max* and *min* are obtained from equations (1) and (2), where MSS stands for Maximum Segment Size, BW_{excess} is the excess bandwidth, and $RTT_{average}$ is the average of the RTT of all connections that join in the boundary node. Although some type of signaling is needed to calculate $RTT_{average}$, 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], [7], [8] or [9]. The dropping probability p for CBM is shown in equation (3), and a more detailed description can be found in [10]. The simplified pseudo-code of the entire CBM algorithm is written in Fig. 1.

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

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

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

Simulation results included in [10] showed that with CBM it is possible to control the number of *out* packets that each source injects into the network. Results obtained in miscellaneous TCP environments (single bottleneck topologies with different target rates, different round trip times and share of resources with best-effort connections), indicated that CBM can assure fairness in the excess bandwidth sharing. Results with CBM were also compared with other traffic conditioner implementations such as Time Sliding Window and Leaky Bucket (taken as classical references), where CBM got a comparatively better accomplishment.

3. Scenarios for Simulation

Simulation topology includes three routers, and TCP traffic is generated by eight long-lived TCP Reno sources transmitting at the link rate, which is set to 33 Mbps. To verify the impact of target rates, different values are used in simulations. We also measure the influence of different RTTs. In the TCP homogeneous scenario (same RTT for all connections), the RTT 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 [12], and was applied to validate the analytical study carried out in [13].

```
Initially:
 counter1=1
 counter2=link_rate/target_rate
 counter3=0
 calculate probability p, 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
     packet is marked as in
     counter3=0
     counter1--
   else
     packet is marked as out
     counter3++
     if time>start_dropping_time
       if counter3>max
        out packet dropped
       else if counter3>min
        out dropped with probability p
       otherwise out is accepted
```

Fig. 1. Simplified pseudo-code of the CBM traffic conditioner algorithm

We employ a large packet size of 9,188 bytes, which corresponds to classical IP over ATM (Asynchronous Transfer Mode) that could represent DiffServ over MPLS (Multi Protocol Label Switching), where the use of the ATM technology seems inherent. The routers located inside the network, store and forward 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] ([minimum threshold, maximum threshold, maximum probability]) 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 [14].

We consider five different scenarios, described in Table 1 for an under-subscribed situation (traffic load ~ 60%), since the excess bandwidth in an oversubscribed scenario represents a very small portion of the total available bandwidth. In scenario A all connections have the same RTT and the same contracted rates, which makes this situation both ideal and infrequent in real networks. In scenario B all connections have the same RTT and different contracted rates. With the introduction of different target rates we try to be closer to a real environment with QoS. Scenario C is the opposite of scenario B (different RTT); hence, we can analyze the effect of the RTT on the CBM traffic conditioner performance. In 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 [3], [10], [11] and [15]. 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.

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.

4. Performance Evaluation of CBM in Three-Node Topologies

In this section, we evaluate the performance of TCP flows crossing two routers in terms of *goodput* and fairness in the excess bandwidth sharing. Early works in this direction commonly found unfeasible to hard guarantee a quantifiable service to TCP traffic [16] [17]. Although recent research presents more favorable results (e. g. [4]), a feasible implementation seems not clear. To evaluate fairness we use the fairness index *f* shown in equation (4), where x_i is the excess goodput of source *i*, and *n* is the number of sources that compose the aggregate [18]. The closer to 1 in the *f* value, the higher the obtained fairness.

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

We study the cases shown in Fig. 2, 3 and 6. In these figures, nodes tagged as *router_1* and *router_2* implement the RIO mechanism with parameters [40/70/0.02] for *in* packets and [10/40/0.2] for *out* packets, or the RED algorithm with parameters [10/40/0.2]. *Router_3* receives traffic from *router_1* and *router_2*, and executes RIO with the same mentioned parameters.Aggregates from each router are either entirely composed of assured service traffic or a mixture of best-effort and assured service traffic, depending on the case under study, so that we can evaluate the robustness of the CBM algorithm in combination with RIO.

CBM traffic conditioners are placed beside the TCP sources if an Assured Service is contracted, otherwise sources belong to the best-effort class and their packets are treated as *out* without performing traffic conditioning. The *max* and *min* values (eq. 1 and 2) employed in the CBM mechanism are shown in Table 1. For *router_1* and *router_2*, we calculate these values as mentioned in section 2 for only one router device, i.e., assuming that *router_1* does not know about the existence of *router_2* and vice versa. Simulation results have a confidence interval of 95% that has been calculated as indicated in section 3.

Table 1. Target rates, round trip times and *max-min* limits for the TCP Reno sources in cases 1 and 2 (from source#0 to source#7 respectively)

Scenarios	Target Rate (Mbps)	RTT (ms)	*Router_1* *Sources# 0-3*		*Router_2* *Sources# 4-7*	
			max	*min*	*max*	*min*
A	2.5	50	16	8	16	7
B	1-1-2-2-3-3-4-4	50	19	10	13	7
C	2.5	10 to 80	8	4	21	11
D	1-1-2-2-3-3-4-4	10 to 80	10	5	17	9
E	4-4-3-3-2-2-1-1	10 to 80	7	4	24	12

4.1. Three RIO Nodes Handling Assured Service Traffic

This first case (case 1) is composed of three RIO routers and 8 TCP Reno sources with a contracted Assured Service. All sources generate traffic at link rate, set to 33 Mbps (see Fig. 2). The different characteristics of scenarios A, B, C, D and E depicted in section 3 are included in Table 1 as well as the *max* and *min* thresholds.

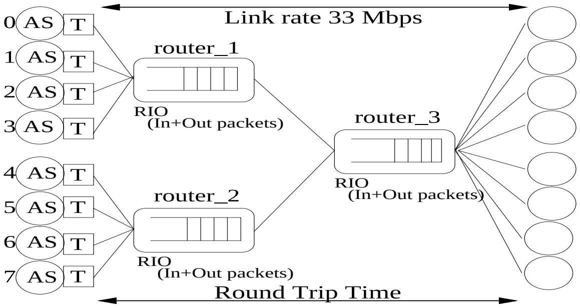

Fig. 2. Three-node topology for case 1 (T≡Traffic conditioner)

As observed in simulation results (see Table 2), CBM and RIO allow users to obtain their contracted target rates despite of the miscellaneous situations. The fact of dropping *out* packets at the traffic conditioner before entering the aggregate makes TCP sources to adapt to the network characteristics regarding the available bandwidth and their respective contracted rates. Once the target rates are achieved, each connection gets a similar portion of excess bandwidth as indicated in last row of Table 2, where the fairness index is over 0.8 for all scenarios except scenario D.

The odd distribution of excess bandwidth in scenario D can be explained as follows. *Router_1* manages connections with small targets and low RTT, while *router_2* has to deal with large target rates and high round trip delays. In a single bottleneck topology, this does not represent a problem due to the good interaction between CBM and RIO [10]. However, in this case, the task of distributing excess bandwidth is done mostly by *router_3*. This router uniquely makes use of RIO, and consequently we get a worse but reasonable fair excess bandwidth allocation (f=0.623). Therefore, a more efficient buffer management would be required to improve the fairness index, if necessary.

Table 2. Achieved rate in Mbps for *in* packets and fairness index *f* in case 1 and the five scenarios A to E

Source	A	B	C	D	E
0	2.50	0.99	2.50	0.99	3.85
1	2.49	1.00	2.50	0.99	3.99
2	2.49	1.99	2.49	2.00	3.00
3	2.49	1.99	2.49	2.00	2.99
4	2.49	2.99	2.50	2.99	1.99
5	2.49	2.99	2.50	2.99	1.99
6	2.49	3.99	2.49	3.95	1.00
7	2.50	3.99	2.48	3.70	1.00
Fairness	0.998	0.857	0.907	0.623	0.803

4.2. Two RIO Nodes and One RED Node Handling Assured Service Traffic

In this situation, eight TCP Reno sources have an Assured Service as depicted in Fig. 3, where connections 0 to 3 join in the RIO router *router_1* and connections 4 to 7 join in the RED router *router_2* (case 2a). We conduct simulations with the same different scenarios written in Table 1. The case we present in this section is interesting for an ISP (Internet Service Provider). Mainly, because an Assured Service could be guaranteed with a simpler implementation (i.e. RED, that is basically a FIFO scheme avoiding global synchronization), since not all routing equipment can configure a RIO scheme. It could be also attractive from the point of view of reconfiguring network resources, being able to face up some network failure when one node has to be temporarily replaced with different hardware that does not implement service differentiation.

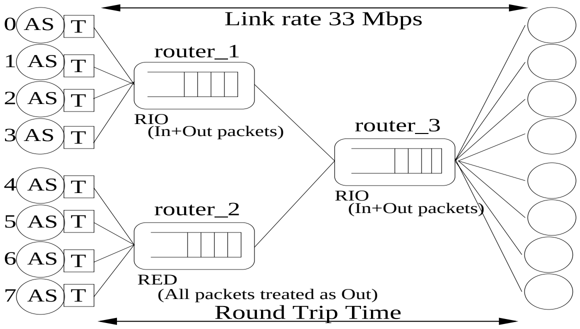

Fig. 3. Three-node topology for case 2 with RED in *router_2* (case 2a)

Results exhibit that the users' contracted target rates are still guaranteed after a transient interval (see Fig. 4, where we show the first 180 seconds of simulation time). Notice that the transient in the *goodput* is not relevant to the final performance. This transient interval principally affects to scenario D (see Fig. 5), the worst situation because *router_2* endures more assured traffic (*in* packets) since sources 4 to 7 have the greatest contracted rates together with the largest RTT, and without implementing packet differentiation (only RED mechanism). For the rest of scenarios (see Table 3), target rates are fulfilled and the transient period is almost negligible considering the final result. The fairness index is over 0.8 except for scenario D (see last row of Table 3).

4.3. Three Nodes Handling Assured Service and Best-effort Traffics

We have performed new simulations considering the option of having the RED scheme in *router_1* and RIO in *router_2* (case 2b). In this situation, the RED router *router_1* does not have to face connections with larger RTT and high contracted rates simultaneously. Results improve slightly as observed in Table 4. Contracted rates are ensured, and in terms of sharing the excess bandwidth, the fairness index is better than case 2a (see last row of Table 4).

Topology of case 2 allows for a more flexible or easier network reconfiguration. At this point, we consider two network failures: first, we see the case traffic conditioner failures, and secondly, we add a network node failure. We assume that if the traffic conditioner fails the source behaves as a best-effort one. In this way, we want to foresee the robustness of the CBM traffic conditioner.

This situation may be also explained from another point of view. DiffServ implementations do not usually mix best-effort and assured traffics in the same queue, but locate them in different queues that belong to different Assured Forwarding classes. Accordingly, it is not the goal of this case of study to suggest the mixing of AF and BE traffics in the same queue for real networks, but to analyze if it is feasible for an ISP to react to network failures reconfiguring. In such a way that both type of traffics have to share the same queue inside the router device without losing network performance. In other words, still guaranteeing contracted target rates and offering a fair share of the spare bandwidth.

In this case (Fig. 6), traffic is generated at link rate by twelve TCP Reno sources. Sources 0 to 3 and 6 to 9 belong to an Assured Service, whereas sources 4 and 5 from *router_1* and sources 10 and 11 from *router_2* are best-effort. We conduct simulations with miscellaneous attributes as shown in Table 5. Packets from best-effort connections are treated as *out*, and these sources do not have contracted target rates, thereby trying to get as much excess bandwidth as possible.

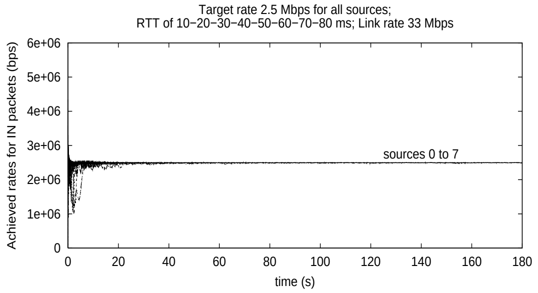

Fig. 4. Guaranteed contracted target rates of all sources with CBM in case 2a and scenario C (all sources have contracted 2.5 Mbps and have RTT from 10 ms to 80 ms respectively at increments of 10 ms)

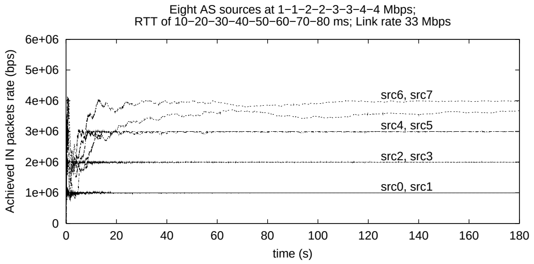

Fig. 5. Contracted target rates are guaranteed for all sources with CBM in case 2a and scenario D (sources have contracted 1-1-2-2-3-4 and 4 Mbps and have RTT from 10 ms to 80 ms respectively)

Table 3. Achieved rate in Mbps for *in* packets in case 2 with RED in *Router_2* (case 2a)

Source	A	B	C	D	E
0	2.49	1.00	2.49	1.00	3.99
1	2.50	1.00	2.50	0.99	3.99
2	2.49	1.99	2.50	1.99	2.99
3	2.50	1.99	2.49	2.00	3.00
4	2.49	2.99	2.49	2.99	1.99
5	2.49	2.99	2.49	2.99	1.99
6	2.49	4.00	2.49	3.99	1.00
7	2.50	3.99	2.49	3.44	1.00
Fairness	0.997	0.848	0.906	0.612	0.803

Table 4. Achieved rate in Mbps for *in* packets in case 2 with RED in *Router_1* (case 2b)

Source	A	B	C	D	E
0	2.49	0.99	2.50	1.00	3.83
1	2.50	1.00	2.50	1.00	3.97
2	2.49	1.99	2.50	2.00	2.99
3	2.50	1.99	2.49	1.99	3.00
4	2.49	2.99	2.50	2.99	1.99
5	2.49	2.99	2.49	2.99	1.99
6	2.49	3.99	2.49	3.96	1.00
7	2.50	3.99	2.49	3.73	1.00
Fairness	0.997	0.862	0.913	0.695	0.794

Table 5. Target rates, round trip times (increments of 10 ms), and *max-min* limits for the TCP Reno sources in case 3 (from source#0 to source#11). Best-effort sources do not have contracted target rates

Source	Target rates (Mbps)	RTT (ms)	*Router_1 Sources# 0-3*		*Router_2 Sources# 6-9*	
			max	*min*	*max*	*min*
A	2.5 (except best-effort)	50	16	8	16	8
B	1-1-2-2-0-0-3-3-4-4-0-0	50	19	10	13	7
C	2.5 (except best-effort)	10 to120	11	6	30	15
D	1-1-2-2-0-0-3-3-4-4-0-0	10 to120	13	7	25	13
E	4-4-3-3-0-0-2-2-1-1-0-0	10 to 120	10	5	35	18

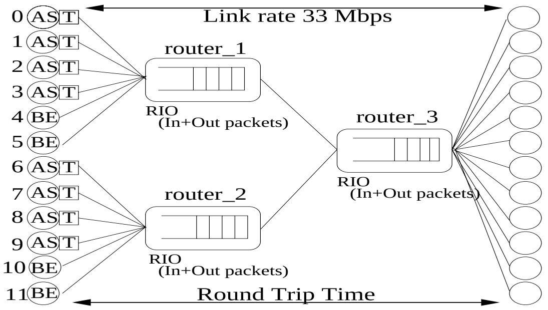

Fig. 6. Three-node topology for case 3 with RIO in all routers (case 3a)

Using the topology of Fig. 6, where the three routers implement RIO (case 3a), there are three connections that would not fully achieve their targets (see Table 6, where logically best-effort sources are not included). We refer to sources number 8 and 9 in scenario D, which have a target of 4 Mbps and a RTT of 90 and 100 ms, and source number 0 in scenario E, which has a target of 4 Mbps and RTT of 10 ms. Due to the substantial differences in delays and targets among connections, and the effect of best-effort traffic, it is not possible to strictly guarantee contracted rates. Additional improvements should be necessary if the ISP considers that having best-effort and assured traffics in the same queue is not a short-live situation, but a constraint in its network configuration. Nevertheless, these results try to adapt to real situations with as much accuracy as possible, hence, the fact of ensuring targets with variations that in the worst case achieve 70 % of target rates should be thought of as a general advance in preserving service differentiation with the assured service approach.

Table 6. Achieved rate (Mbps) for *in* packets in case 3 with RIO in all routers (case 3a)

Source	A	B	C	D	E
0	2.49	1.00	2.49	1.00	3.16
1	2.50	1.00	2.50	1.00	4.00
2	2.49	2.00	2.49	2.00	2.99
3	2.49	1.99	2.49	2.00	2.99
6	2.50	2.99	2.49	2.96	1.99
7	2.49	2.99	2.49	2.93	1.99
8	2.49	3.98	2.49	2.99	1.00
9	2.49	3.99	2.49	2.60	1.00
Fairness	0.843	0.732	0.710	0.622	0.784

In this case, an even worse situation would appear if one of the RIO nodes fails and has to be replaced by a RED one. Table 7 shows that we get similar results. Therefore, as in the previous case we still provide an acceptable assurance level.

Regarding the excess bandwidth, the fact of having mixed assured and best-effort traffics in the same routers (*router_1* and *router_2*) favors the generation of less *out* packets from best-effort connections, therefore the fairness index is kept above 0.75 excluding scenario D (see last rows of Tables 6 and 7). Despite of not splitting best-effort and assured traffics in different AF queues in the routers, the best-effort sources do not consume the whole bandwidth due to the good interaction between CBM and RIO. Table 8 includes the values of the fairness index for all cases and scenarios. Although it is not shown, TSW and LB present worst values for the fairness index performance.

Table 7. Achieved rate in Mbps for *in* packets in case 3 with RED in *Router_2* (case 3b)

Source	A	B	C	D	E
0	2.50	1.00	2.49	1.00	3.30
1	2.49	0.99	2.50	1.00	3.99
2	2.49	1.99	2.49	1.99	2.99
3	2.49	1.99	2.50	2.00	2.99
6	2.49	2.99	2.49	2.90	1.99
7	2.49	2.99	2.49	2.70	1.99
8	2.50	3.97	2.50	2.70	1.00
9	2.49	3.99	2.49	2.60	1.00
Fairness	0.843	0.732	0.710	0.615	0.784

Table 8. Fairness indexes with CBM in three-node topologies. Case 1: three RIO nodes handling Assured Service traffic. Case 2: two RIO nodes and one RED node handling Assured Service traffic (in case 2a *Router_2* is RED and in case 2b *Router_1* is RED). Case 3a: three RIO nodes handling Assured Service and best-effort traffic. Case 3b: two RIO nodes and one RED node handling Assured Service and best-effort traffic

Case\Scenario	A	B	C	D	E
Case 1a	0.998	0.857	0.907	0.623	0.803
Case 2a	0.997	0.848	0.906	0.612	0.803
Case 2b	0.997	0.862	0.913	0.695	0.794
Case 3a	0.843	0.732	0.710	0.622	0.784
Case 3b	0.843	0.732	0.710	0.615	0.784

5. Conclusion

In this paper, we present a performance analysis of the Counters Based Modified (CBM) traffic conditioner in a three-node topology. The underneath idea in CBM is discarding *out* packets in the traffic conditioners before these packets join the aggregate. This dropping process is done with a probability that depends on the target rate, the excess bandwidth and a simple estimation of the RTT. The study done in this paper with three-nodes implementing RIO or RED under different characteristics (different contracted target rates, different round trip times and share of resources with best-effort connections) reflects that if only Assured Service sources are involved, target rates are guaranteed, even if one of the first routers in the topology does not implement any service differentiation (i.e. using RED). Meanwhile, the obtained fairness index shows an even distribution of excess bandwidth for all scenarios, with a slight reduction in scenario D.

We also test in this paper, the behavior of the TCP sources when Assured Service traffic and best-effort traffic compete for network resources. Notice that merging these two different types of traffic in the same queue is not a usual practice. In consequence, obtaining good results under these circumstances would represent the advantage of using a small number of queues when implementing DiffServ in real environments or being able to face up transitory situations in which the ISP cannot reallocate best-effort traffic. In spite of the fact that simulation results reveal that contracted target rates are achieved in most situations, the general conclusion is that the ISP can strictly guarantee only a 70% of contracted target rates for the worst scenarios, what can be considered a good result in case of failure and network reconfiguration for a short period of time.

Given a multi-node topology, if the number of best-effort sources increases in relation to the number of Assured Service sources the architecture based on RIO routers tends to be unfair in the sharing of resources (target rates and spare bandwidth). Therefore, if we want to consider the presence of many best-effort connections along with Assured Service connections it seems clear that new mechanisms should be included to provide a fair end-to-end TCP performance in Internet. With these results we conclude that CBM provides a high accuracy in guaranteeing the users' contracted target rates, introduces less complexity than other proposals, and supplies a good value for the fairness index in heterogeneous topologies. For that reason, being a feasible election in the Assured Forwarding implementation for DiffServ.

4. Acknowledgements

This work was supported by the Spanish Research Council under projects TEC2004-05622-C04-02/TCM and TIC2001-3339-C02-02.

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*, 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,* 182-191, 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, LNCS 1815*, 540-554, France, May 2000.

[5] Elloumi O, De Cnodder S, Pauwels K, Usefulness of the three drop precedences in Assured Forwarding Service. *Internet draft*, 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*, Brazil, Vol. 1(A), 188-193, December 1999.

[7] H. Kim, A Fair Marker, *Internet draft*, April 1999.

[8] Alves, J. De Rezende, L. De Moraes, Evaluating Fairness in Aggregated Traffic Marking. *Proceedings of IEEE Globecom'2000*, 445-449, USA, November 2000.

[9] 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*, New Orleans, LA, Vol. 2, 806-810, June 2000.

[10] Maria-Dolores Cano, Fernando Cerdan, Joan Garcia-Haro, Josemaria Malgosa-Sanahuja, Counters-Based Modified Traffic Conditioner. *Lecture Notes in Computer Science (QofIS'02), Vol. 2511*, 57-67, 2002.

[11] Maria-Dolores Cano, Fernando Cerdan, Joan Garcia-Haro, Josemaria Malgosa-Sanahuja, Performance Evaluation of Traffic conditioner Mechanisms for the Internet Assured Service. *Quality of Service over Next-Generation Data Networks, Proceedings of SPIE Vol. 4524*, 182-193, 2001.

[12] 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,* 273-286, Baltzer Science.

[13] V. Bonin, O. Casals, B. Van Houdt, C. Blondia. Performance Modeling of Differentiated Fair Buffer Allocation. *Proceedings of the 9th International Conference on Telecommunications Systems*, USA, 2001.

[14] S. Floyd and V. Jacobson, Random Early Detection Gateways for Congestion Avoidance, *IEEE/ACM Transactions on Networking, Vol. 1 No.4*, 397-413, August 1993.

[15] N. Seddigh, B. Nandy, P. Pieda, Bandwidth Assurance Issues for TCP flows in a Differentiated Services Network. *Proceedings of IEEE Globecom'99*, Vol. 3, 1792-1798, Brazil, December 1999.

[16] J. Ibañez, K. Nichols, Preliminary Simulation Evaluation of an Assured Service. *Internet draft*, draft-ibanez-diffserv-assured-eval-00.txt, August 1998.

[17] J. F. De Rezende, Assured Service Evaluation. *Proceedings of IEEE Globecom'99*, Brazil, December 1999.

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