Experimental Study of Bandwidth Assurance in a DiffServ Network

Maria-Dolores Cano, Fernando Cerdan, Pablo Lopez-Matencio
Department of Information Technologies and Communications
Polytechnic University of Cartagena
Campus Muralla del Mar s/n 30202 Cartagena
Phone: +34 968 325953 Fax: +34 968 32 59 73
E-mail: {mdolores.cano, fernando.cerdan}@upct.es

ABSTRACT
Inside DiffServ, the Assured Forwarding per hop behavior defines a service that guarantees the contracted target rate to the users and allows consuming more bandwidth if the network load is low. In this paper we perform an experimental study, whose key contribution is that current techniques and commercial equipment do not meet the goals of the AF PHB service in a scalable way.

KEY WORDS
Experimental study, Assured Service, Differentiated Services, QoS.

1 Introduction

Some years ago, the need of finding IP QoS solutions led to the Differentiated Services (DiffServ) approach [1] [2]. Nowadays, most IP-based networks tend to use this architecture to provide end-to-end QoS. The DiffServ architecture is intended to create a simple scheme that provides a range of QoS levels by moving complexity toward the edge of the network. The Type of Service packet header field from IP v.4 is substituted by the DiffServ field and new meanings are conferred to its bits: the six most significant bits compose the DiffServ Code Point (DSCP), while the two less significant bits are currently unused. A group of mechanisms to handle packets of aggregated flows with different priorities according to the information carried in the DSCP is created. 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 traffic conditioning functions (metering, marking, shaping) need only to be implemented at boundary nodes, whereas interior nodes perform a set of forwarding PHBs to aggregates of traffic that have been appropriately marked. There are two PHBs standardized by the IETF, the Expedited Forwarding per-hop behavior (EF PHB) [3] and the Assured Forwarding per-hop behavior (AF PHB) [2] [4].

Due to the interest of providing QoS in IP networks with the use of DiffServ, we are motivated to study DiffServ experimentally. In this paper we analyze different solutions to implement the AF PHB in a DiffServ domain testbed with Cisco routers. We examine the effect of using different scheduling algorithms with or without an active queue management algorithm. We measure the throughput obtained by the aggregates of TCP traffic in heterogeneous scenarios, disregarding retransmitted packets. For completeness, we also consider the distribution of the excess bandwidth among aggregates.

The rest of this paper is organized as follows. Section 2 explains briefly the DiffServ architecture, WRED (Weighted Random Early Detection) and CBWFQ (Class Based Weighted Fair Queuing). Section 3 describes the experimental topology, and testing scenarios. In section 4 we discuss the obtained results. The paper concludes in section 5 with the most important remarks.

2 Differentiated Services

A continuous group of nodes that implements DiffServ composes a DiffServ (DS) domain. The introduction of non-DS-capable nodes in a DiffServ domain can lead to unpredictable results, and few studies have been developed in this way [5]. In a DiffServ domain we find two types of nodes, boundary nodes and interior nodes. Boundary nodes connect the DiffServ domain with other domains that may or may not support DiffServ. These nodes classify and perform conditioning functions to incoming traffic to the DiffServ domain (or outgoing traffic in some cases), assuring that packets in the DiffServ domain are marked with the correct PHB of one of the PHBs implemented in the DiffServ domain. Interior nodes are connected only to other interior nodes or boundary nodes inside the same DiffServ domain. Depending on how packets have been marked, i.e. based on the PHB they belong to, interior nodes treat packets in different ways.

The IETF standardized two PHBs: the EF PHB and the AF PHB. The EF PHB can be seen as a Premium Service with high quality constraints, reserved for traffic with the highest priority. On the other hand, the AF PHB tries to guarantee contracted target rates, 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 [4], 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.

2.1 Congestion Control and Scheduling Techniques

Weighted RED (WRED) [6] is an active queue management technique based on RED [7]. In addition to early congestion detection, WRED allows different dropping profiles for different types of traffic. That is, we can assign diverse thresholds depending on the type of traffic. Hereby, different levels of QoS are offered. When a packet arrives to a queue with WRED the following events occur. First, average queue length is calculated based on the previous and current average lengths. Secondly, if the assessed average is below the minimum threshold the packet is queued. Thirdly, if the assessed average is between the minimum and maximum thresholds, the packet may be queued or dropped depending on the dropping probability. Finally, if the assessed average is above the maximum threshold the packet is dropped.

Class Based Weighted Fair Queuing (CBWFQ) [8] is a variant of the well known scheduling mechanism Weighted Fair Queuing (WFQ). WFQ associates weights to the queues so that each queue gets a specific portion of the available bandwidth. Besides, it tolerates packets of distinct size. CBWFQ has the same characteristics as WFQ, with the new property of scheduling packets in the queues based on traffic classes defined by the administrator.

3 Experimental Topology

Our testbed consists of four Cisco 2600 routers and two PCs with the Linux operating system. Cisco IOS (Cisco Internetworking Operating System) includes tools for DiffServ implementation in network nodes such as the Modular Quality of Service Command Line Interface (MQC). MQC allows configuring Class Based Packet Marking, Class Based Policing, WRED and other DiffServ implementation facilities. The topology is shown in Fig. 1. Each PC has two NICs (Network Interface Card), emulating two different LANs. Each NIC of PC_1 is connected to a boundary router. TCP traffic is generated with Netperf [9]. Destinations are the NICs of PC_2.

We focus on the AF PHB Service with two levels of precedence, in of profile (*in*) packets and out of profile (*out*) packets. Boundary nodes, routers E_1 and E_2, perform traffic conditioning tasks. Routers E_1 and E_2 also apply the AF PHB. Router E_3 only performs the AF PHB, and E_4 only de-multiplexes traffic to its corresponding destination. The bottleneck is placed between E_3 and E_4. Traffic conditioning is done using a token bucket mechanism. Depending on the LAN contracted target rate, the token bucket has different configuration parameters. The token bucket does not drop any packet and only marks them as *in* or *out*. We perform different tests using the following methods for implementing the AF PHB:

- Case 1. AF traffic (*in* and *out* packets) is placed in a single FIFO queue with WRED. See Fig. 2.
- Case 2. AF traffic is buffered using Class Based Weighted Fair Queuing (CBWFQ), so *in* packets are placed in a FIFO queue and *out* packets are placed in another FIFO queue. This is an innovative implementation of the AF PHB introduced in [10], whose aim is to avoid interference between *in* and *out* packets. The scheduler visits the *in* packet queue with a probability ρ_1 (see eq. 1 where i is the aggregate number and n is the number of aggregates) that matches the network load. The *out* packet queue is visited with probability 1- ρ_1. See Fig. 3.

$$\rho_1 = \frac{\sum_i^n \text{contracted target rate}_i}{\text{link capacity}} \qquad (1)$$

- Case 3. AF traffic is buffered using CBWFQ with WRED, so traffic coming from LAN_1 is placed in a queue different from traffic coming from LAN_2. The scheduler visits the LAN_1 queue with a probability that matches its corresponding bandwidth (contracted target rate plus excess bandwidth). Observe Fig. 4.

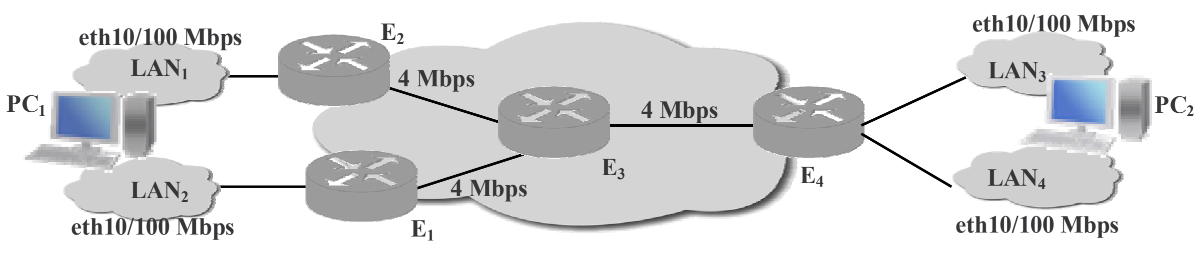

Fig. 1. Experimental topology: 4 routers Cisco 2600 and 2 PCs with Linux (each PC has two NICs)

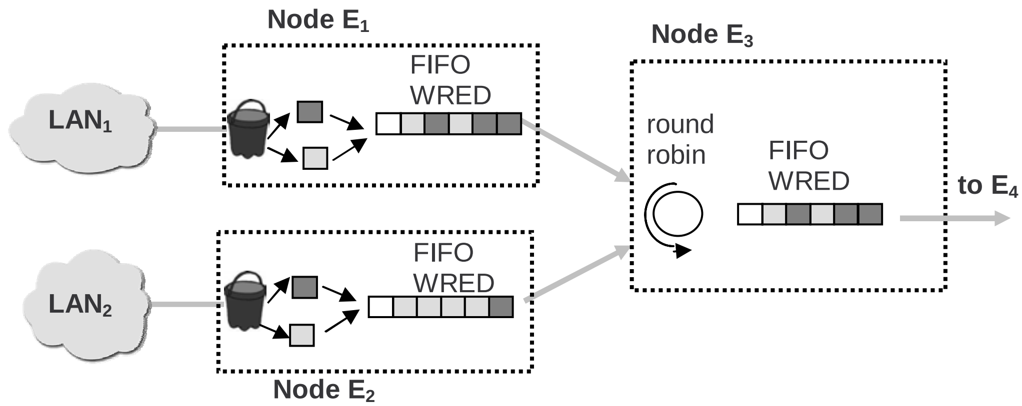

Fig. 2. FIFO queue with WRED for the AF traffic (*in* and *out* packets). Routers E_1 and E_2 employ a token bucket for traffic marking

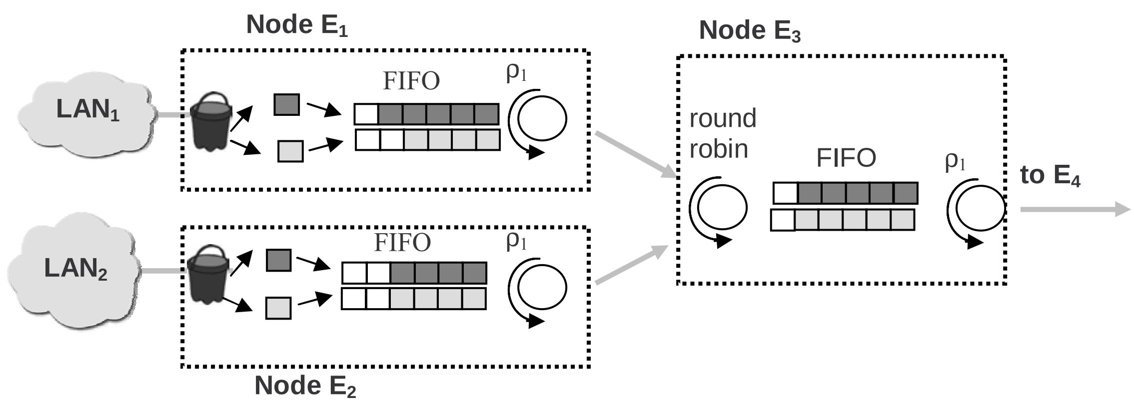

Fig. 3. CBWFQ without WRED for the AF traffic (*in* and *out* packets). In each router there is one queue for *in* packets and another queue for *out* ones. Routers E_1 and E_2 employ a token bucket for traffic marking

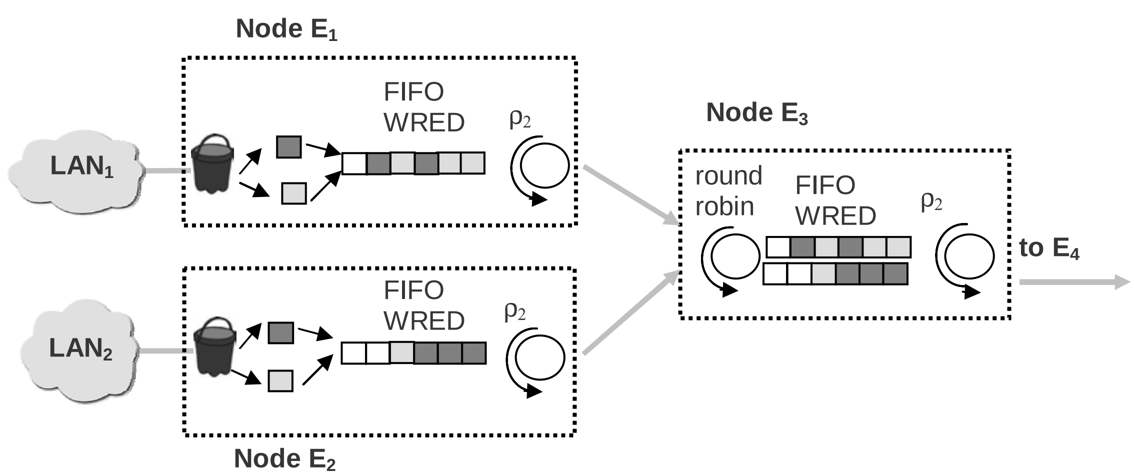

Fig. 4. CBWFQ with WRED for the AF traffic (*in* and *out* packets). Each LAN has its own queue in each node. E_1 only has traffic from LAN_1 thus only has a single queue. The same applies to E_2. E_3 receives traffic from two LANs so it has two queues. Routers E_1 and E_2 employ a token bucket for traffic marking

For instance, if LAN_1 and LAN_2 have contracts of 1 Mbps and 2 Mbps respectively, then the excess bandwidth is 1 Mbps (link capacity 4 Mbps). With an even share of the excess bandwidth LAN_1 should get a total bandwidth of 1.5 Mbps and LAN_2 should obtain 2.5 Mbps. Therefore, queue LAN_1 is visited with a probability of 37.5% (see eq. 2) and queue LAN_2 with a probability of 62.5%.

$$\rho_2 = \frac{\text{contracted target rate} + \text{excess bandwidth portion}}{\text{link capacity}} \quad (2)$$

Notice that scheduling and congestion management mechanisms are applied always at the output of each router. In the inputs, routers use a round robin fashion to serve incoming packets. For each case (1, 2, and 3), scenarios are modified in the following way:

i) Scenario A. Each LAN only has a single source of traffic. Both LANs contract the same target rate. This target is incremented from 250 Kbps (network load 12.5%) to 2 Mbps (network load 100%). See Table 1.

Table 1. Contracted target rates in scenario A

Network load (%)	Contracted target rates of LAN_1 and LAN_2 (Mbps)
12.5	0.250
25.0	0.500
37.5	0.750
50.0	1.000
62.5	1.250
75.0	1.500
87.5	1.750
100	2.000

ii) Scenario B. Each LAN has a single source of traffic. One of them has a fixed contracted target rate of 250 Kbps (LAN_1) and the other (LAN_2) has a contract that ranges from 250 Kbps (network load 12.5%) to 3,750 Kbps (network load 100%). See Table 2.

iii) Scenario C. LAN_1 has four sources of traffic with a total contracted target rate of 1,024 Kbps. LAN_2 has a variable number of sources, from 2 to 16, and each source has a contract of 125 Kbps. This experiment is only performed for CBWFQ with and without WRED (cases 2 and 3). See Table 3.

Table 2. Contracted target rates in scenario B

Network load (%)	Target rate of LAN_1 (Mbps)	Target rate of LAN_2 (Mbps)
12.50	0.250	0.250
18.75	0.250	0.500
25.00	0.250	0.750
31.25	0.250	1.000
37.50	0.250	1.250
43.75	0.250	1.500
50.00	0.250	1.750
56.25	0.250	2.000
62.50	0.250	2.250
68.75	0.250	2.500
75.00	0.250	2.750
81.25	0.250	3.000
87.50	0.250	3.250
93.75	0.250	3.500
100	0.250	3.750

4 Results

In this section we study through experimental tests the performance of the three AF implementations described in section 3: a single FIFO queue with WRED, CBWFQ without WRED (one queue for *in* packets and one queue for *out* packets), and CBWFQ with WRED (as many queues as LAN networks). For each case, we analyze the three scenarios also explain in section 3. The QoS parameters that we measure are the final throughput of the LANs (disregarding retransmitted packets, what is usually called goodput), and the distribution of excess bandwidth among them.

4.1 Case 1: single FIFO queue with WRED

In this section we present the results obtained when AF traffic is placed in a single FIFO queue with WRED. WRED parameters are [40, 70, 0.02] for *in* packets, corresponding to minimum threshold, maximum threshold and dropping probability, and [10, 40, 0.2] for *out* packets. The buffer length is 200 packets.

Fig. 5 shows the throughput achieved for each LAN in scenario A. Contracted target rates are guaranteed for the complete range of network load. In general, both sources get the same portion of excess bandwidth (see Fig. 6). The greater differences in the distribution of excess bandwidth occur for a network load around 50-62.5%. Because both LANs contract the same target rate, contracted bandwidth is assured and excess bandwidth is evenly shared.

Table 3. Contracted target rates in scenario C

Network load (%)	Number of sources in LAN_1 (contract of 0.250 Mbps each)	Contracted target rate of LAN_1 (Mbps)	Number of sources in LAN_2 (contract of 0,125 Mbps each)	Contracted target rate of LAN_2 (Mbps)
31.25	4	1.000	2	0.250
37.50	4	1.000	4	0.500
43.75	4	1.000	6	0.750
50.00	4	1.000	8	1.000
56.25	4	1.000	10	1.250
62.50	4	1.000	12	1.500
68.75	4	1.000	14	1.750
75.00	4	1.000	16	3.000

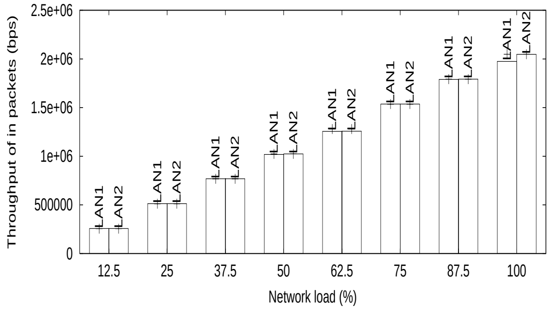

Fig. 5. Throughput of *in* packets in scenario A. AF PHB implemented with a single FIFO queue with WRED. Both LANs have the same contracted target rates from 250 Kbps (12.5% network load) to 2 Mbps (100% network load)

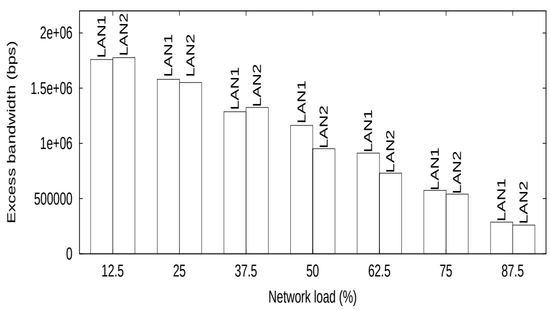

Fig. 6. Excess bandwidth distribution in scenario A. AF PHB implemented with a single FIFO queue with WRED. Both LANs have the same contracted target rates from 250 Kbps (12.5% network load) to 2 Mbps (100% network load)

On the other hand, we examine the effect of a fixed contract for LAN_1 of 250 Kbps and a variable contract for LAN_2 (scenario B). We observe in Fig. 7 how the contract is still assured for both sources. However, there is a lack of fairness in the allotment of spare bandwidth (see Fig. 8). The LAN network whose contracted target rate is smaller (LAN_1) notably obtains more spare bandwidth.

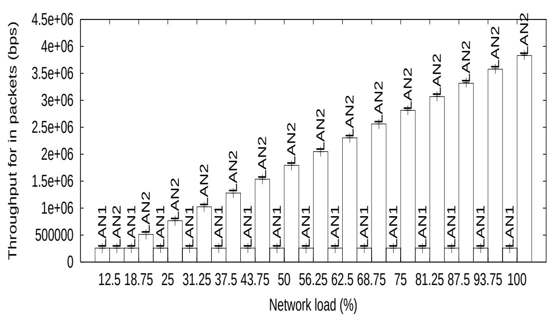

Fig. 7. Throughput of *in* packets in scenario B. AF PHB implemented with a single FIFO queue with WRED. LANs have different target rates

4.2 Case 2: CBWFQ without RED

In this section we present the results obtained when AF traffic is buffered using CBWFQ without WRED. Depending on the type of AF packet (*in* or *out*), this is placed in a different queue. Queues are served with the probability explained in section 3 (see eq. 1). Fig. 9 represents the throughput obtained for each LAN when they contract the same bandwidth (scenario A). We see that contracts are fully satisfied. Regarding the excess bandwidth, the distribution is evenly done; getting each LAN approximately half of the excess bandwidth (see Fig. 10). In fact, it is equal to the one obtained for case 1 (FIFO with WRED). This good behavior is mainly due to the homogeneity of contracted target rates. In a heterogeneous scenario with distinct targets (scenario B), we found that the contracted bandwidths are guaranteed too. However, the distribution of the excess bandwidth is not fair for a network load above 37.5% (again the LAN whose contracted target rate is smaller, LAN_1, gets more resources). See Fig. 11.

In this case, we also study the effect on performance when we increase the number of sources on each LAN (scenario C). On the one hand, we detect that independently of the number of sources, contracted target rates of each LAN are achieved. Inside each LAN bandwidth is equally shared. On the other hand, the LAN whose sources have the smaller contracts (LAN_2) consumes more excess bandwidth (Fig. 12). We observe that the influence of small contracts prevails over the number of flows. For instance, when LAN_1 has four sources (with targets of 256 Kbps) and LAN $_2$ has only two (with targets of 128 Kbps), the latter gets more excess bandwidth even though LAN_1 has more flows. This example corresponds to a network load of 62.5% in Fig. 12.

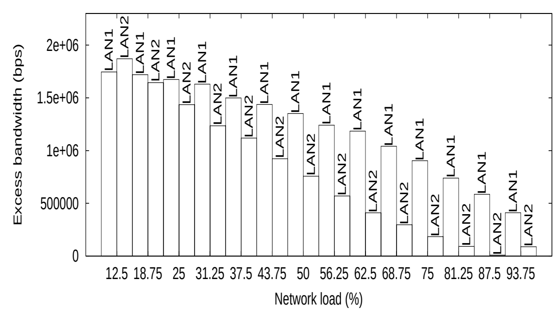

Fig. 8. Excess bandwidth distribution in scenario B. AF PHB implemented with a single FIFO queue with WRED. LANs have different target rates

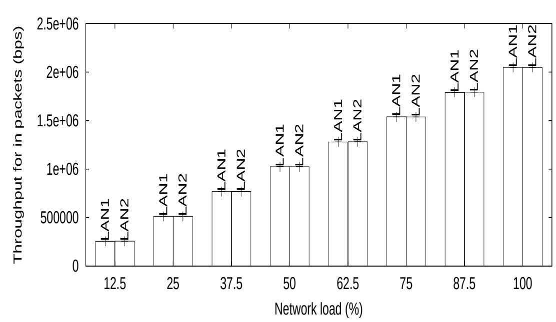

Fig. 9. Throughput of *in* packets in scenario A. AF PHB implemented with CBWFQ without WRED. LANs have the same contracted target rate

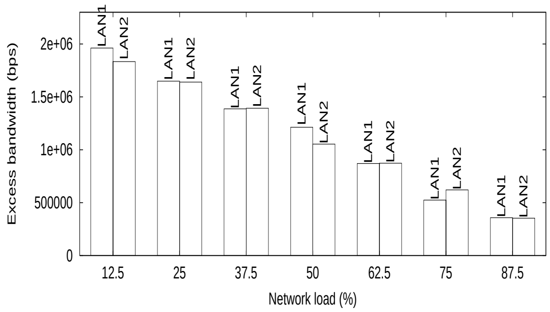

Fig. 10. Excess bandwidth distribution in scenario A. AF PHB implemented with CBWFQ without WRED. Both LANs have the same contracted target rate

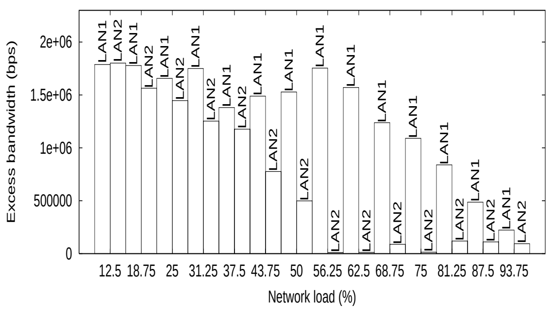

Fig. 11. Excess bandwidth distribution in scenario B. AF PHB implemented with CBWFQ without WRED. LANs have different target rates

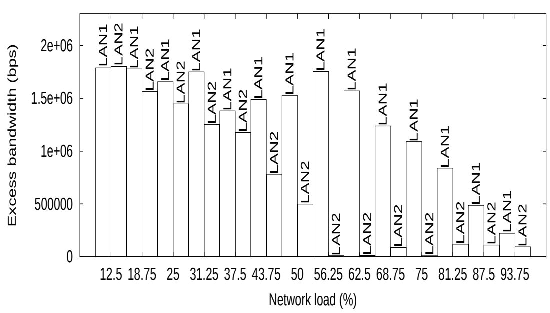

Fig. 12. Excess bandwidth distribution in scenario C. AF PHB implemented with CBWFQ without WRED. LANs have different target rates. LAN_1 has always four sources (contract of 256 Kbps each). LAN_2 has from 2 (network load 31.25%) to 16 (network load 75%) sources (contract of 128 Kbps each)

4.3 Case 3: CBWFQ with WRED

In this section we discuss the results achieved when AF traffic is buffered using CBWFQ with WRED. Traffic arriving from a different LAN is placed in a different queue, and each queue implements WRED with the same parameters as in section 4.1. Queues are served with the probability explained in section 3 (see eq. 2). In scenario A, we observe that contracted target rates are strictly assured and excess bandwidth is strictly shared at 50%. Previous results show the same performance in the homogeneous case, where both LANs have the same targets. In scenario B, when the contract is different for each LAN, this is the only implementation that not only ensures contracted target rates but distributes evenly the spare bandwidth (until a network load of 80 % approximately). See Fig. 13.

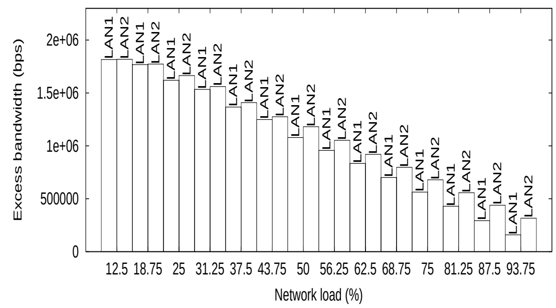

Fig. 13 Excess bandwidth distribution in scenario B. AF PHB implemented with CBWFQ with WRED. LANs have different target rates

Increasing the number of sources does not represent a drawback with this implementation. As shown in experimental results obtained in scenario C, contracts are fully achieved and excess bandwidth is distributed impartially between LANs (see Fig. 14).

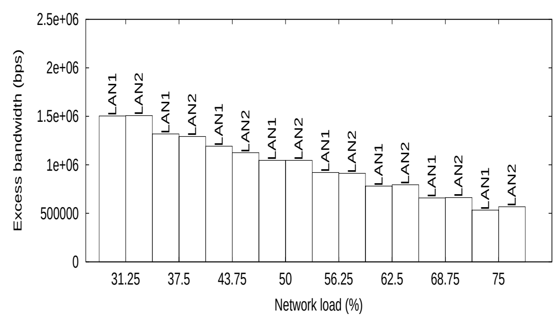

Fig. 14. Excess bandwidth distribution in scenario C. AF PHB implemented with CBWFQ with WRED. LANs have different target rates. LAN_1 has always four sources (contract of 256 Kbps each). LAN_2 has from 2 (network load 31.25%) to 16 (network load 75%) sources (contract of 128 Kbps each)

However, this method presents the following drawbacks:

- The number of queues needed in a router to manage AF traffic is equivalent to the number of LANs meeting at this router. Likely, if there is a great number of LANs the router might not be able to implement so many queues.
- To configure all nodes, boundary and interior ones, it is necessary to know the contracted target rates of all LANs and all link capacities. Hereby, there is a lack of scalability.

5 Conclusions

We performed an experimental study about bandwidth assurance in a DiffServ network. We employed Cisco

2600 routers widely used in current IP networks, and PCs as sources of TCP traffic using the Netperf TCP traffic generator. Traffic conditioning done in boundary nodes is carried out by the token bucket algorithm. We tested various available mechanisms for the AF PHB implementation in DiffServ networks: FIFO with WRED, CBWFQ without WRED, and CBWFQ with RED. For each of them, we analyze different scenarios: homogeneous contracted target rates, heterogeneous contracted target rates and different number of flows in each aggregate.

Results show that all mechanisms are able to guarantee contracts (with more or less accuracy). The problem appears for the excess bandwidth distribution. In general, sources with a small contract get more network resources. The only case in which the two objectives of the AF PHB are achieved is with CBWFQ with WRED. Nevertheless, this implementation requires that all network nodes know all the contracted target rates of the LANs. Moreover, it is necessary to separate traffic depending on the LAN origin. This is not an inconvenient if the number of LANs is reasonable inside the domain, but it makes the system not scalable if the number of LAN grows. Consequently, we conclude that with the current solutions in commercial equipments for DiffServ, specifically for the AF PHB, it is not possible to offer a complete Assured Service; therefore, being decisive to incorporate new techniques in commercial equipment.

Acknowledgements

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

References

[1] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, “An Architecture for Differentiated Services”, RFC 2475, December 1998.

[2] D. Grossman, “New Terminology and Clarifications for DiffServ”, RFC 3260, April 2002.

[3] B. Davie, A.Charny, J. C. R. Bennet, K. Benson, J. Y. Le Boudec, W. Courtney, S. Davari, V. Firoiu, D. Stiliadis, “An expedited forwarding PHB (Per-Hop Behavior)”, RFC 3246, March 2002.

[4] J. Heinanen, F. Baker, W. Weiss, J. Wroclawski, “Assured Forwarding PHB Group”, RFC 2597, June 1999.

[5] 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*, pp. 57-67, 2002.

[6] WRED, “Configuring Weighted Random Early Detection – Cisco IOS Quality of Service Solutions Configuration Guide”, available at <www.cisco.com>.

[7] S. Floyd, V. Jacobson, Random Early Detection Gateways for Congestion Avoidance, *IEEE/ACM Transactions on Networking, Vol. I*, pp. 397-413, August 1993.

[8] CBWFQ, “Class-Based Weighted Fair Queuing – Release 12.0(5)T” available at <www.cisco.com>.

[9] Netperf, available at <www.netperf.org>

[10] Maria-Dolores Cano, Fernando Cerdan, Joan Garcia-Haro, Josemaria Malgosa-Sanahuja. A new proposal for Assuring Services in Internet. *Proceedings of the International Conference on Internet Computing IC’02*, Vol. II pp. 379-384, Las Vegas, USA, June 2002.