ADAPTIVE LOW PRIORITY PACKET MARKING FOR BETTER TCP PERFORMANCE*

Giovanni Neglia, Giuseppe Bianchi, Francesco Saitta
Università di Palermo, Dipartimento di Ingegneria Elettrica
giovanni.neglia@tti.unipa.it, bianchi@elet.polimi.it, francesco.saitta@tti.unipa.it

Dario Lombardo
Engineering Ingegneria Informatica
dario.lombardo@eng.it

Abstract This paper proposes a packet marking scheme for TCP traffic. Unlike previous literature work, in our scheme the majority of TCP packets are transmitted as high priority. The role of a low priority packet appears that of a *probe*, whose goal is to early discover network congestion conditions. Low priority packets are marked according to an adaptive marking algorithm. Numerical results show that our scheme provides improved throughput/delay performance.

Keywords: TCP, Packet marking, RIO

1. Introduction

Several packet marking algorithms have been proposed to provide service differentiation among a set of TCP flows that share network resources. All packet marking mechanisms have a common basic approach. Packets of each individual flow are marked based on a suitably chosen profile at an edge router. Then, marked packets are aggregated in the network, and receive a different treatment in the network core routers.

Generally, a two-level marking scheme is adopted, where packets labelled as IN receive better treatment (lower dropping rate) than packets marked as OUT. Within the network, dropping priority mechanisms are

*This research is supported by European Community and MIUR in the frame of the Pollens project (ITEA, if00011a) implemented in active queue management schemes such as RIO - Random Early Discard with IN/OUT packets [1].

The basic idea of proposed algorithms is that a suitable marking profile (e.g. a token bucket which marks IN/OUT profile packets) may provide some form of protection in the case of congestion. A large number of papers [1, 2, 3, 4, 5, 6] have thoroughly studied marking mechanisms for service differentiation, and have evaluated how the service marking parameters influence the achieved rate.

More recently, TCP marking has been proposed as a way to achieve better than best effort performance [7, 8, 9]. The idea is that packet marking can be adopted also in a scenario of homogeneous flows (i.e. all marked according to the same profile), with the goal of increasing the performance of all flows. In particular, [7, 8] consider long lived flows and adopt goodput and loss as performance metrics. Conversely, [9] focuses on WWW traffic and proposes a new scheme able to reduce the completion time of a http session. According to the marking schemes used in these works, most of the packets in the network are of type OUT. Hence, packets marked as IN will be protected against network congestion (indeed [9] relies on this property to protect flows with congestion window lower than 4, when packet losses cannot recovered via fast retransmission).

This paper proposes a new adaptive packet marking mechanism devised to increase the flow performance. The novelty of our approach in comparison to [7, 8, 9] consists in marking the majority of packets as IN, but occasionally transmitting an OUT packet. This leads to a very different network situation with respect to previous works: since the large majority of packets in the network are of type IN, by marking a packet as OUT we dramatically increase the probability that this packet is dropped. In essence, the role of the OUT packet is that of a *probe*, whose goal is to early discover whether the network is getting congested.

The preliminary performance evaluation carried out in this paper shows that our proposed scheme consistently achieves better performance than that of an all-IN packet marking scenario. Moreover, the higher the dropping rate of OUT packets versus the IN dropping rate, the better the performance gain is. Ideally, the optimal operational condition in the network should be that of a 100% loss rate of the OUT packets but still no loss encountered by IN packets.

We recall that some results presented in [9] show that interleaving IN and OUT packets may have a highly negative impact on the TCP throughput, if the loss rate of the OUT traffic is much larger than that of the IN traffic. In particular a throughput reduction may be encountered as the percentage of IN traffic becomes greater than a given threshold.

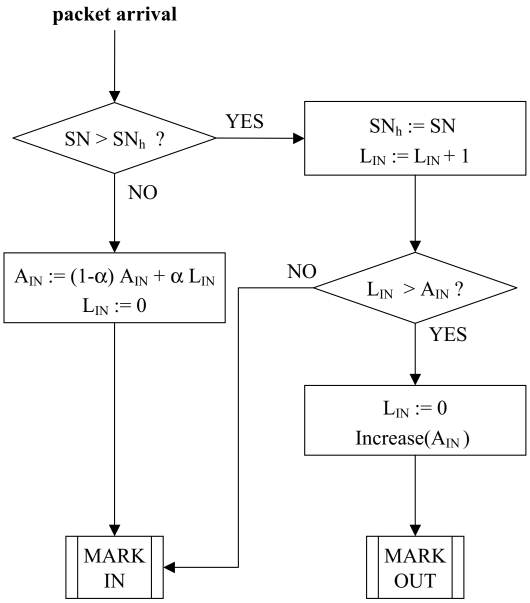

Figure 1. Packet Marking Algorithm

Indeed, we confirm that other marking algorithms may have critical performance in such conditions[1]. Our marking algorithm does not suffer of this problem and, on the contrary, it takes advantage of a very high OUT packet loss rate.

The rest of this paper is organized as follows. Section 2 describes the packet marking algorithm. Section 3 presents the simulation scenario and parameters. The performance evaluation of the proposed algorithm is carried out in section 4. Finally, conclusive remarks and further research issues are given in section 5.

2. Packet Marking Algorithm

The packet marking algorithm proposed in this paper can be implemented at the ingress router. It acts on a per-flow basis. When the ingress router detects a TCP SYN packet, meaning that a new flow is

[1] We have observed this effect with both a token-bucket marker and with a marking scheme very similar to the one proposed in [7, 8] (protection of small window and retransmitted packets, an OUT packet inserted every n IN packets). Neither in [7] nor in [8] the authors indicate the number of available IN token used in the simulations.

offered, it reserves a state for the flow. This state is composed of the following variables:

- SN_h. This variable stores the higher Sequence Sumber (SN) transmitted by the flow. It is initially set to the ISN value (Initial Sequence Number) carried by the SYN packet, and it is updated whenever a non-empty packet (i.e. non a pure ACK packet) with higher SN[2] arrives at the router.
- L_{IN}. This counter is initially set to 0. It is reset to 0 when either a packet loss is detected, or a packet marked as OUT is transmitted. It is increased for every transmitted subsequent packet. Hence, the counter L_{IN} represents the actual length of a burst of IN-marked transmitted packets.
- A_{IN}. This is an Auto Regressive filtered value which keeps track of the past values of the counter L_{IN} (i.e. the size of a burst of successfully transmitted IN packets averaged over a recent period). It is initially set equal to a design parameter A_0. In addition, as shown below, it is used by the marker to determine which packet to mark as OUT, and it is increased after every OUT-marked packet to provide adaptivity.

The algorithm is described in the flow-chart in figure 1. When a non-empty packet arrives at the router, its sequence number SN is read. According to the new SN value, and the recorded highest sequence number encountered before, we face two possible situations. If $SN \leq SN_h$, then the incoming packet is a replica of a previously transmitted packet. This means that such packet has probably been lost. Conversely, if $SN > SN_h$ the incoming packet is a new one.

Our algorithm distinguishes these two cases. In the case of packet loss, the value A_{IN} is updated as the weighted sum of the previous estimate with the current value of L_{IN}. L_{IN} value is then reset to 0, to mean that a new burst of IN-marked packets has begun. The retransmitted packet is delivered marked as IN. In the case of a new incoming packet the current IN-marked packet burst size is increased by one. The packet is then marked as IN if the current burst L_{IN} is shorther than the value A_{IN}. Conversely, if the actual burst of IN-marked packets has become longer than A_{IN}, the actual packet is marked as OUT, and a new burst begins ($L_{IN} = 0$).

Note that, after the transmission of an OUT packet, we need to increase the value A_{IN}. In figure 1, this operation is generically indicated as increase(A_{IN}). In fact, when congestion conditions occur, several

[2] in a cyclical sense - recall that sequence numbers wrap when the value $2^{32} - 1$ is reached

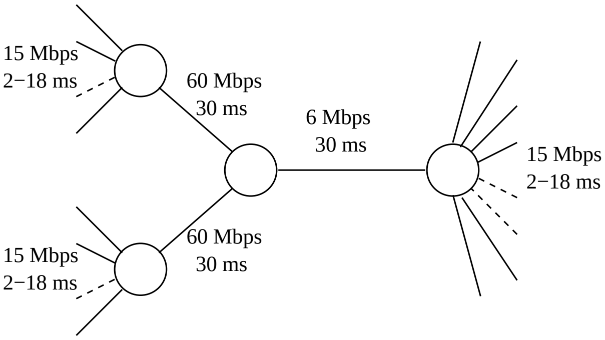

Figure 2. Network topology

packet losses may be encountered, and thus the value A_{IN} decreases (left part of figure 1).

To better understand how this increment should be quantitatively accounted, consider the situation in which all packets labelled as IN are successfully received, while all packets labelled as OUT are discarded. This means that the congestion level in the network has reached a given stationary target value. To remain in such stationary conditions, the OUT marking rate should not vary with time, i.e. an OUT packet should be marked every $\bar{A}_{IN}$ IN packets, being $\bar{A}_{IN}$ a constant[3] In the assumption of stop&wait TCP operation[4], no IN packet loss, and 100% OUT packet loss, it is easy to see that A_{IN} remains constant to an initial value $\bar{A}_{IN}$ if the increase rule is $A_{IN} := A_{IN}/(1-\alpha)$.

The thorough optimization of the algorithm's configuration parameters (namely, α, A_0, and the increase(A_{IN}) rule) is out of the goals of this paper, and is object of current research activity. To obtain numerical results, unless otherwise specified, we have adopted $\alpha = 0.5$, $A_0 = 10$, and $A_{IN} := 2A_{IN} + 1$ as increase rule. It is interesting to remark that even with parameters chosen without any accurate tuning, the performance of the algorithm are very good. This is perhaps an indication of the robustness of the considered algorithm to non optimal settings.

3. The Simulation Scenario

The network topology considered is shown in figure 2, it consists of a single bottleneck link, whose capacity is set equal to 6 Mbps.

[3] It depends (in a non trivial manner) on the RIO configuration at the bottleneck link and on the number of offered flows.

[4] For general values of the contention window, such an analysis is much more complex as it further depends on how many packets have been sent when a triplicate ACK arrives at the sender.

We considered two different load conditions with 10 and 40 TCP long-lived flows. The sources have always data to transmit, so the throughput is determined only by the network conditions.

In order to avoid synchronization among the sources each source starts to transmit randomly in the interval 0-1 s, and propagation delays of the access links are chosen so that Round Trip Time are different (from 124ms to 198ms, the average value is 160ms).

Each router deploys RIO (RED with In/Out bit, [1]) as Active Queue Management. For RED operation refer to [10]. We let min, max be the two thresholds, w_q the weight of the instantaneous queue value in the moving average filter, P_{max} the maximum dropping probability in the region of random discard. RIO uses two twin RED algorithms for dropping packets, one for IN packets and one for OUT packets which share the same physical queue. So RIO is configured with two sets of RED parameters: (min_{in},max_{in},$P_{max_{in}}$) and (min_{out},max_{out},$P_{max_{out}}$). RIO discriminates against OUT packets in times of congestion essentially in two way: firstly IN dropping probability depends on the average queue for the IN packets, while OUT dropping probability on the average total queue; secondly parameter are opportunely chosen for the two kinds of traffic. In [1] the authors suggest the following rules: $min_{out} < min_{in}$, $max_{out} << max_{in}$, $P_{max_{out}} > P_{max_{in}}$, and in the paper they choose $max_{out} < min_{in}$.

As regards RED parameters, the thresholds and $P_{max_{in}}$ are chosen according to [11], the filter coefficient w_q according to [12], i.e. $max = 3min$, $P_{max_{in}} = 0.1$ and $w_q = 1 - exp(-M/(C * 10 * RTT)) = 0.0012$, where C is the link capacity, M is the packet size and RTT is the Round Trip Time.

RIO configuration allows the network provider to trade off link utilization and delay performance: the higher the RED thresholds, the higher link utilization and delay. Different settings were considered. As regards the IN traffic, the min_{in} threshold values goes from 2 to 80 packets. As regards the OUT traffic we considered two different scenarios: in the first the OUT traffic settings vary according to IN traffic parameters, $max_{out} = 3min_{out} = min_{in}$ and $P_{max_{out}} = 0.2$, in the second they are fixed to $min_{out} = 2$, $max_{out} = 6$ and $P_{max_{out}} = 0.2$. In what follows we refer to this two settings respectively as *soft differentiation* and *hard differentiation*.

Lastly queue physical lengths were chosen so that packet losses occurred only in the core router, due to RIO (not to physical queue overflow).

We compared the proposed marker with a no-marker situation, where all the packets are treated as IN packet. For each of the threshold setting

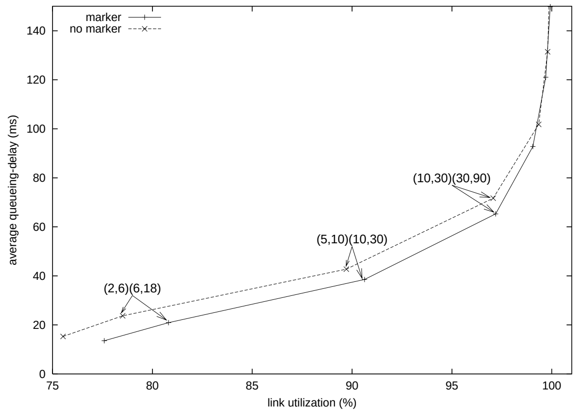

Figure 3. Delay vs link utilization - 10 flows, soft differentiation

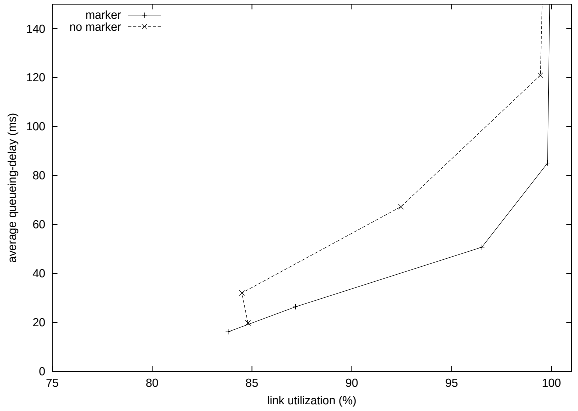

Figure 4. Delay vs link utilization - 40 flows, soft differentiation

we evaluated link utilization (goodput) and average delay, and plotted them as "performance frontiers".

Simulations were conducted through ns v2.1b8. We used TCP Reno implementation. For each configuration at least 5 simulations with different random seeds were run. Each simulation lasted 1000 simulated seconds, statistics were collected after 50 seconds. In the figures we present in the following section, standard deviation of goodput and average delay is always less than 1% of their numerical value.

4. Performance Evaluation

Figures 3 and 4 show the performance frontiers respectively for 10 flows and 40 flows in soft differentiation. In figure 3 RIO threshold settings are reported for three points in the form (min_{out},max_{out})

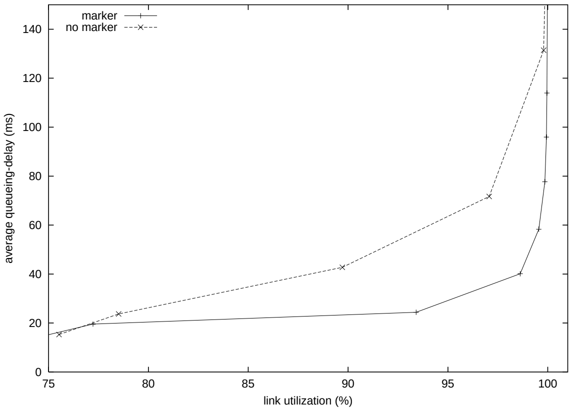

Figure 5. Delay vs link utilization - 10 flows, hard differentiation

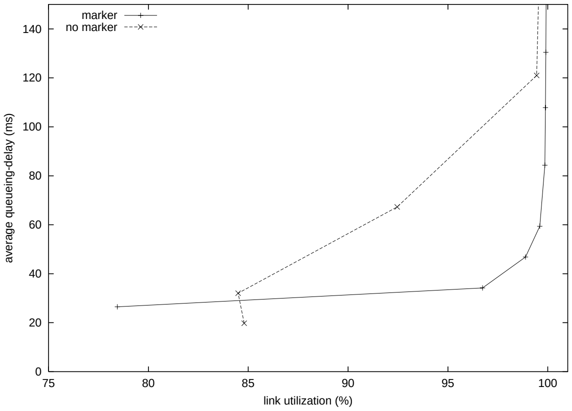

Figure 6. Delay vs link utilization - 40 flows, hard differentiation

(min_{in},max_{in}). Performance improvement provided by the marker employment is remarkable under high load condition.

The improvement is more significant when IN and OUT packets receive more different services from the network, as one can see in figures 5 and 6.

As regards the number of packet marked IN by the algorithm, it increases as thresholds are higher and link utilization increases. For both soft and hard differentiation IN packet percentage varies from about 98% to more than 99% for the tested configuration with 10 sources and from about 94% to 97% with 40 sources (losses increase with the number of flows). Figure 7 shows global, IN and OUT loss percentage for 10 flows. We see that for high goodput values in hard differentiation OUT loss percentage is near 100% while IN loss percentage is very small: source

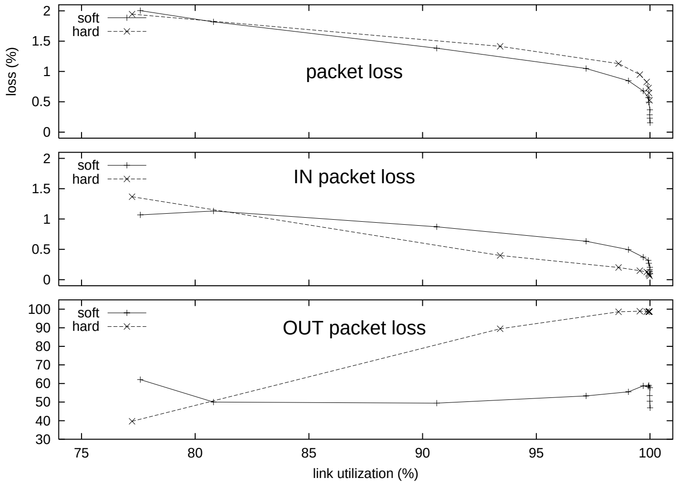

Figure 7. Loss percentage - 10 flows, soft and hard differentiation

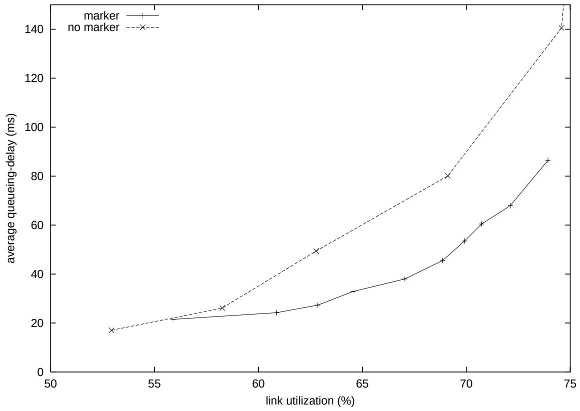

Figure 8. Delay vs link utilization - 10 flows, UDP traffic, 5 s activity time

behavior becomes almost “deterministic”, the variance of the offered load is highly reduced so performance are significantly improved.

We tested also a different -less aggressive- increasing law for the A_{IN} after the OUT-marking of a packet: $A_{IN} := A_{IN} + 1$. This modified marker performs better in soft differentiation: delay decrease achievable by marker employment is almost doubled in comparison to 3. Besides also dropping probability for OUT packets is lower: in the range [15%,30%] for the different configurations tested, versus the range [50%,60%] for the previous simulations. On the contrary performance are poorer in hard differentiation, so in what follows we consider the original increasing law. Nonetheless these results seem to confirm the intuition that when the OUT congestion thresholds are not severe the marker can usefully mark OUT a higher number of packets, allowing the sources to more readily adapt themselves to available resources. This hints that the marker should estimate also losses in the OUT class and should use this information to regulate A_{IN} increase.

4.1. Dynamic Behavior

In order to study the behavior of the proposed mechanism in a changing environment we added an ON/OFF UDP source with a peak rate equal to 3 Mbps. Activity time and idle time are equal and strictly alternating. It is a very hard stress condition, given the 6 Mbps bottleneck bandwidth. We considered 0.05 s, 1 s and 5 s activity time. For 10 flows and in hard differentiation better performance results are achieved through the marker employment. As activity time increases, points on the marker frontier become denser in comparison to points on the no-marker frontier (as one can note for example in figure 8). This indicates that, for a given RIO configuration, sources react slower to the increase of network resources due to the marker employment, so goodput is lower. The increasing law of A_{IN} needs more thorough analysis.

With 40 sources and hard differentiation the same results and consideration hold, except that for very low thresholds performance are worse when the marker is employed. In soft differentiation the advantage of the marker employment is significant for 0.05s activity time with 10 flows, for 0.05s and 1s with 40 flows.

4.2. Interaction with other flows and deployment issues

The employment of the proposed marking algorithm at the edge router relocates dropping from the IN class to the OUT class. Indeed the marker employment reduces losses inside the IN class. This property has important consequences as regards the interaction with other flows and consequently the deployment issues.

Firstly we consider a single domain and two kinds of possible non-cooperative flows: UDP (non reactive) flows and TCP flows not subjected to the marker, in what follows we call them free TCP flows. In both cases if flow packets are marked OUT[5], there is an improvement of the performance of the TCP protected flows, because the increase of OUT traffic intensity determines higher OUT packet losses.

[5] The network provider should evaluate if it is opportune, because, depending on RIO configuration, OUT traffic can experiment very high loss percentage as it has been showed above.

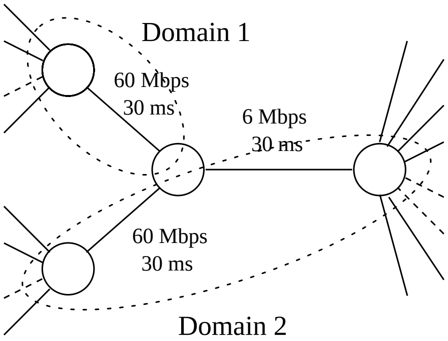

Figure 9. Network topology

As regards UDP flows marked IN, there is still an improvement as regards network resources utilization, but loss decrease inside the IN class advantages moreover UDP flows against TCP flows.

Free TCP flows, whose packets are always marked IN, try to exploit all the available bandwidth at the expense of the other flows, whose throughput starves. For this reason no free TCP source should be admitted to transmit in the IN class.

Let us now consider a multidomain scenario. We examined the situation when flows coming from domain 1 where the marker is exploited compete for network resources against flows in domain 2 where there is a single best effort service. Figure 9 shows this scenario, in order to simplify the simulations there are not two different border routers (one for each domain). Note that it is essentially the same topology of figure 2, but the bottleneck is in a domain which does not support differentiation between IN and OUT packets.

In the simulations we varied the RED queue parameters in domain 2 according to previous simulations to obtain performance frontiers. As regards domain 1, IN packet thresholds are high (100 and 300), and we considered different settings for the OUT triple (min_{out},max_{out},$P_{max_{out}}$): (30,90,0.2), (2,6,0.2), (0.1,0.3,1). We considered 10 and 40 flows. In both load regimes with the first two settings for OUT traffic the performance of marked flows and free flows are almost the same, only with the third pathological setting, the marked flows experiment lower throughput (from two to three times). The explanation is the following: markers in domain 1 note losses due to congestion in domain 2 and mark some packets OUT, but unless the congestion threshold is extremely severe for OUT traffic these packets are not discarded in domain 1 where there is no congestion and have the same dropping probability of all the other packets in domain 2 where there is no distinction between the two classes of traffic.

5. Conclusions and further research

In this paper we have presented a new adaptive marking mechanism for TCP flows, able to increase throughput/delay performance in a wide range of scenarios. Its novelty consists in using OUT packets as probes to early discover network congestion condition.

Exploitation requires that the marker is deployed all over a domain at the edge routers or that a traffic class (the IN class in this paper) is devoted to marked TCP flows.

We think that further improvements of the marking mechanism are possible. In particular current research activity is investigating the adaptation law of the parameter A_{IN} and the performance of the marking scheme in the presence of short-lived TCP flows (e.g. http sessions).

References

[1] D. D.Clark and W. Fang, “Explicit Allocation of Best Effort Packet Delivery Service”, IEEE Transactions on Networking, Vol. 6, No. 4, pp. 362-373, Aug. 1998.

[2] J. Ibanez, K. Nichols, “Preliminary Simulation Evaluation of an Assured Service”, IETF draft, August 1998

[3] N. Seddigh, B. Nandy, P. Piedu, “Bandwith Assurance Issues for TCP flows in a Differentiated Services Network”, IEEE Globecom, Rio de Janeiro, pp. 1792-1798, December 1999

[4] S. Sahu, D. Towsley, J. Kurose, “Quantitative Study of Differentiated Services for the Internet”, IEEE Globecom, Rio de Janeiro, pp. 1808-1817, December 1999

[5] S. Sahu, P. Nain, D. Towsley, C. Diot, V. Firoiu, “On Achievable Service Differentiation with Token Bucket Marking for TCP”, Proc. ACM SIGMETRICS’00, Santa Clara, CA, June 2000

[6] W. Feng, D. Kandlur, D. Saha, K. Shin, “Adaptive Packet Marking for Mantaining End-to-End Throughput in a Differentiated Services Internet”, IEEE/ACM Transactions on Networking, Vol. 7, NO:5, pp. 685-697, April 1999

[7] F. Azeem, A. Rao, S. Kalyanaraman “A TCP-Friendly Traffic Marker for IP Differentiated Services” IwQoS’2000, Pittsburg, PA, June 2000.

[8] G. Lo Monaco, F. Azeem, S. Kalyanaraman, Y.Xia, “TCP-Friendly Marking for Scalable Best-Effort Services on the Internet”, Computer Communication Review (CCR), Volume 31, Number 5, October 2001.

[9] M. Mellia, I. Stoica, H. Zhang, “Packet Marking for Web traffic in Networks with RIO Routers”, Globecom 2001, San Antonio, Texas, November 25-29, 2001

[10] S. Floyd, V. Jacobson, “Random Early Detection gateways for Congestion Avoidance” IEEE/ACM Transactions on Networking V.1 N.4, August 1993, p. 397-413

[11] S. Floyd, “RED: Discussions of Setting Parameters”, email November 1997, http://www.icir.org/floyd/REDparameters.txt

[12] S. Floyd, R. Gummadi, and S. Shenker, “Adaptive RED:An Algorithm for Increasing the Robustness of RED’s Active Queue Management”, August 1, 2001, under submission