A new Three Color Marker for TCP flows

Giovanni Neglia, Giuseppe Bianchi, Marilena Sottile
Dipartimento di Ingegneria Elettrica, Università degli Studi di Palermo
giovanni.neglia@tti.unipa.it, bianchi@elet.polimi.it, marilena.sottile@libero.it

Abstract

In Differentiated Services networks, packets may receive a different treatment according to their Differentiated Services Code Point (DSCP) label. As a consequence, packet marking schemes can be devised to differentiate packets belonging to a same TCP flow, with the goal of improving the experienced performance. This paper presents an extensive performance evaluation of a new adaptive packet marking scheme, which relies on three different service classes, provided by the DiffServ network. The scheme provides protection of vulnerable packets in the TCP flows and adaptivity through "probes" packets. The performance have been evaluted in a traffic scenario composed of TCP flows with different length. The proposed marking scheme provides excellent performance in all utilization conditions.

Keywords: TCP, Marking, RED, RIO.

1. Introduction

Differentiated Services (DiffServ) networks provide the ability to enforce a different forwarding behavior to packets, based on their Differentiated Services Code Point (DSCP) value. A possible way to exploit the DiffServ architecture is to provide differentiated support for flows belonging to different traffic classes, distinguished on the basis of the DSCP employed. However, since it is not required that all packets belonging to a flow are marked with the same DSCP label, another possible way to exploit DiffServ is to identify marking strategies for packets belonging to the same flow.

Several packet marking algorithms have been proposed for TCP flows. The marking strategy is enforced at the ingress node of a DiffServ domain (edge router). Within the DiffServ domain, marked packets are handled in an aggregated manner, and receive a different treatment based on their marked DSCP. Generally, a two-color 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 implemented in active queue management schemes such as RIO - Random Early Discard with IN/OUT packets [1].

The basic idea of the 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, mostly characterized by short-lived TCP flows, and proposes a new scheme able to reduce the completion time of an http session.

In all the above mentioned marking schemes, 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 small window, when packet losses cannot be recovered via the fast retransmission algorithm). As shown in section **2**-C, our marking strategy inherits from [10] a somehow opposite philosophy.

In this paper we propose a new three color algorithm, which merges two proposed approaches: one explicitely developed to reduce the completion time for short-lived http flows [9], the other proposed by us in [10], which uses marking to drive TCP flow control in an adaptive way. In what follows we refer to the first scheme as Web Packet Marking (WPM), to the second as Adaptive Packet Marking and to the new marking scheme as Three Color Packet Marking (3CPM).

The rest of this paper is organized as follows. Section **2** describes the new three color packet marking algorithm, focusing the rationale behind the proposed marking strategy in subsection **2**-C. Section **3** presents the simulation scenario and parameters. The performance evaluation of the proposed algorithm is carried out in section **4**: in particular 3CPM is compared with a no-marker situation and with its "parents schemes" WPM and APM. Finally, conclusive remarks and further research issues are given in section **5**.

2. Packet Marking Algorithm

The three color packet marking algorithm can be implemented at the ingress router of a DiffServ network and acts on a per-flow basis.

In a DiffServ framework, the domain administrator can dedicate an Assured Forwarding class i with three different dropping level to marked TCP traffic: AF$i0$, AF$i1$ and AF$i2$ ordered for increasing dropping probability in core routers [11]. In what follows we refer to AF$i0$, AF$i1$, AF$i2$ packets respectively as green, yellow, red packets.

When the ingress router detects a TCP SYN packet, meaning that a new flow is offered, it reserves a state for the flow. This state is composed of the following variables:

- N_s. This variable stores the number of green packets the flow is allowed to send at the begin of the connection.
- N_a. This variable stores the number of green packets the flow is allowed to send at the begin of Slow Start and Fast Recovery phases.
- 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 arrives at the router. Higher SN is to be intended in a cyclical sense: we

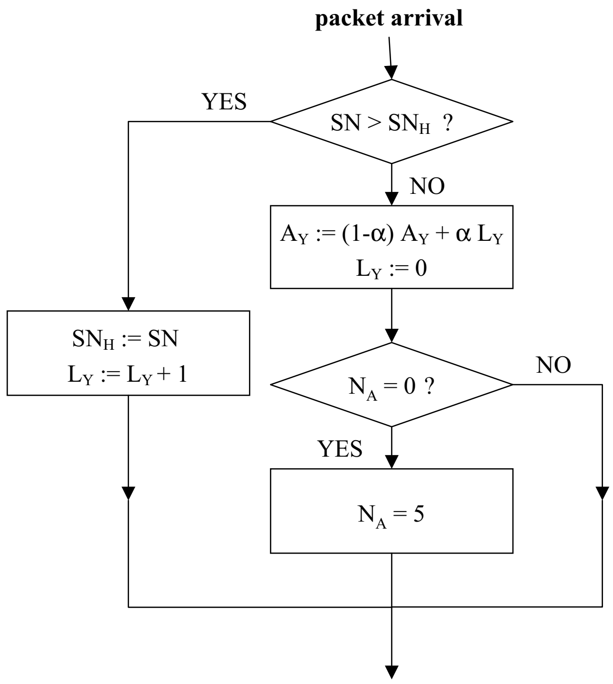

Fig. 1. State update

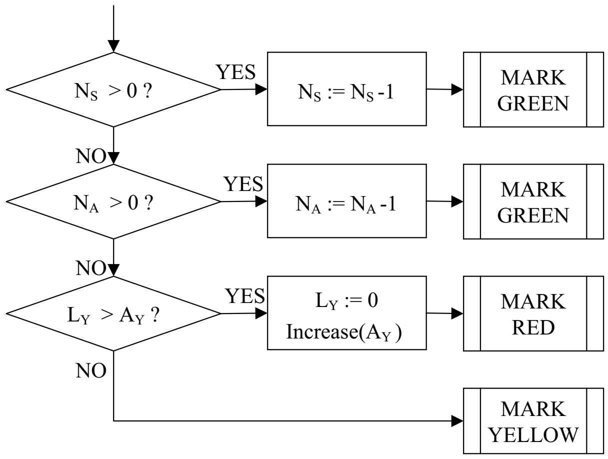

Fig. 2. Marking decision

recall that sequence numbers wrap when the value $2^{32}-1$ is reached.

- L_Y. This counter is initially set to 0. It is reset to 0 when either a packet loss is detected, or a packet marked red is transmitted. It is increased for every transmitted subsequent packet. Hence, the counter L_Y represents the actual length of a burst of green/yellow-marked transmitted packets, i.e. not-probe transmitted packets.
- A_Y. This is an auto regressive filtered value which keeps track of the past values of the counter L_Y (i.e. the size of a burst of successfully transmitted green/yellow 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 red, and it is increased after every red-marked packet to provide adaptivity.

According to our scheme, a few "vulnerable" packets are marked green, a few "probing" packets are marked as red, while the majority of packets are marked as yellow. In the following subsections we specify which packets are marked green and red and then clarify the role of green and red packets. The flow-chart in figure 1 describes how the state is updated when a new packet arrives, while the flow-chart in figure 2 describes how it is determined the packet color.

A. Green packets

Green packets management comes from WPM scheme. The algorithm relies on green packets to allow TCP clients to exit as fast as possible from the undesirable states, i.e. Slow Start phase and the Fast Recovery phase, when the client is more sensitive to packet losses. In [9] the authors propose two variants of their scheme. The first one is tightly integrated with the TCP protocol: the source is allowed to send up to N_s green packets when it starts, then up to $N_a = sstresh$ at the beginning of a Slow Start phase, and up to $N_a = cwnd$ at the beginning of a Fast Recovery phase. The second variants does not require the knowledge of internal TCP variables, hence it can be implemented at the ingress routers. It uses constant values $N_a = N_s = 5$. In order to recognize Fast recovery and Slow Start phase, the ingress router can try to estimate packet losses. Identifying a packet loss can be done looking at the flow sequence number in the TCP header. Indeed, if the sequence number is not greater than the previous one, and the TCP payload is not empty, then the source is retransmitting old data. As a result the router can conclude that the flow enters either the Fast Recovery or the Slow Start phase. However, since the ingress router does not have the possibility to know the internal TCP state, it cannot differentiate between the two phases. 3CPM acts according to this second variant.

B. Red packets

Red packets management comes from APM scheme [10]. 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_Y is updated as the weighted sum of the previous estimate with the current value of L_Y. The L_Y value is then reset to 0, meaning that a new burst of green/yellow-marked packets has begun. The retransmitted packet is delivered marked as green. In the case of a new incoming packet the current green/yellow-marked packet burst size is increased by one. If the current burst L_Y is shorther than the value A_Y, the packet is then marked green or yellow, according to section **2**-A. Conversely, if the actual burst of green/yellow-marked packets has become longer than A_Y, the actual packet is marked as red, and a new burst begins ($L_Y = 0$).
Note that, after the transmission of an red packet, we need to increase the value A_Y. In figure 1, this operation is generically indicated as increase(A_Y). In fact, when congestion conditions occur, several packet losses may be encountered, and thus the value A_Y decreases (right part of figure 1).

To better understand how this increment should be quantitatively accounted, consider the situation in which all packets labelled as green or yellow are successfully received, while all packets labelled as red 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 red marking rate should not vary with time, i.e. an red packet should be marked every $\bar{A}_Y$ green/yellow packets, being $\bar{A}_Y$ a constant[1]
In the assumption of stop&wait TCP operation[2], no green/yellow packet loss, and 100% red packet loss, it is easy to see that A_Y remains constant to an initial value $\bar{A}_Y$ if the increase rule is $A_Y := A_Y/(1-\alpha)$.
The thorough optimization of the algorithm's configuration parameters (namely, α, A_0, and the increase(A_Y) 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 $\alpha = 0.5$, $A_0 = 7$, and $A_Y := 2(A_Y + 1)$ 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.

C. Rationale behind our marking strategy

As regards green packets employment, the purpose is to protect some vulnerability points of TCP protocols, in order to avoid timeouts that would degrade TCP flow perfomance and in particular would lenghten the flow completion time.
More complex is the understanding of red packets role in APM also because usually in marking schemes the most of packets belongs to the lowest priority class (red one according to our scheme). On the contrary our algorithm marks most of the packets as yellow, and interleaves yellow packets with occasional red packets. The length of an yellow-packets burst is adaptively set based on an heuristic estimation of the experienced packet loss ratio.
Since the large majority of packets in the network are of type yellow, by marking a packet as red we dramatically increase the probability that this packet is dropped. In essence, the role of the red packet is that of a *probe*, whose goal is to early discover whether the network is getting congested.
We remark that Random Early Discarding (RED) techniques have been designed with the same philosophy in mind. By randomly dropping packets when the queue size increases above a given threshold, RED provides early feedbacks to the TCP congestion control mechanism running at the network edge. However, RED provides only a "loose" form of control mechanism; dropped packets may belong to a subset of offered flows (which consequently reduce the emission rate), while other TCP flows may not experience packet dropping and thus remain unaware of the congestion situation. Moreover, in the same time, a few TCP flows may even experience multiple packet dropping and therefore be severely penalized.
Conversely, our proposed marking strategy provides a stronger form of control mechanism, as the packets that are likely to experience dropping are not randomly chosen among all the offered packets, but are the ones specifically marked as red.

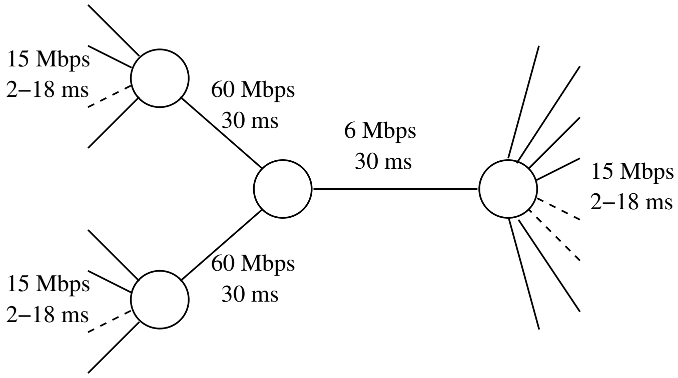

Fig. 3. Network topology

This operation allows to smoothly and fairly drive the TCP congestion control operation and makes the traffic offered to the network more regular, so better performance are achieved. Moreover, as we showed in [10] the higher the dropping rate of red packets versus the yellow 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 red packets but still no loss encountered by yellow packets.
When compared with the schemes proposed in [7], [8], [9], our marking strategy presents two major differences. First, the majority of packets are yellow. Second, the performance takes advantage of a very high red packet loss rate. The fact that our strategy performs well is apparently in conflict with some results presented in [9], which show that interleaving yellow and red packets may have a highly negative impact on the TCP throughput, if the loss rate of the red traffic is much larger than that of the yellow traffic. In particular, a throughput reduction may be encountered as long as the percentage of yellow traffic becomes greater than a given threshold. Indeed, we too have observed performance impairments for both a token-bucket marker and for a marking scheme very similar to the one proposed in [7], [8] (protection of small window and retransmitted packets, an red packet inserted every n yellow packets[3]). However, we remark that these schemes are not designed to be adaptive to the network congestion status, while ours uses some heuristics to provide adaptability.

3. Simulation Scenario

The network topology considered is shown in figure 3. It consists of a single bottleneck link, whose capacity is set equal to 6 Mbps. Propagation delays on the access links are chosen so that Round Trip Times are different (from 124ms to 198ms, the average value is 160ms).
Each router is equipped with RIO (RED with In/Out bit, [1]) as Active Queue Management. For RED operation refer to [12]. 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 was thought to act in a two priority levels scenario (with IN and OUT packets), but its extension to a three priority levels scenario is straightforward: in this situation RIO uses three RED algorithms for dropping packets, one for green,

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

[2] 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.

[3] Neither in [7] nor in [8] the authors indicate the number of available yellow tokens used in the simulations.

TABLE I

FLOW LENGHT DISTRIBUTION

Group	Bytes	Packets	Group	Bytes	Packets
1	119	1	9	1650	2
2	179	1	10	2861	2
3	251	1	11	4706	4
4	334	1	12	8015	6
5	428	1	13	13681	10
6	529	1	14	26641	18
7	658	1	15	284454	190
8	948	1			

one for yellow and one for red packets which share the same physical queue. So RIO is configured with three sets of RED parameters: (min_G,max_G,P_{max_G}), (min_Y,max_Y,P_{max_Y}) and (min_R,max_R,P_{max_R}). RIO discriminates yellow packets versus green ones, and red ones versus yellow ones, essentially in two way: firstly while green dropping probability depends on the average queue occupancy taking into account only green packets, yellow dropping probability depends on the average queue occupancy taking into account green and yellow packets, and red dropping probability depends on the average total queue; secondly dropping probability parameter P_{max} is opportunely chosen for the three 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} are chosen according to [13], i.e. $max = 3min$, $P_{max} = 0.1$. Following [14], the filter coefficient w_q is set to $w_q = 1 - exp(-M/(C * 10 * RTT)) = 0.0012$, 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. In [10] we showed that APM performance is better as the service differentiation among red and yellow packets increases. So here we let the min_Y threshold values going from 6 to 240 packets, while the red traffic settings are fixed to $min_R = 2$, $max_R = 6$ and $P_{max_R} = 0.2$. Min_G threshold is so high that no green packets are dropped.

We compared the proposed marker with a no-marker situation (NM in what follows), where all the packets are treated as yellow packets, and with the two "parents" schemes, i.e. WPM and APM. Coherently with the above descripiton, WPM employs green and yellow packets, while APM employs yellow and red packets.

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 evaluated 3CPM performance in two different traffic scenarii: 1) 10 long-lived TCP flows, which start to transmit randomly in the interval 0-1 s and have always data to transmit, so the throughput is determined only by the network conditions; 2) a more realistic WWW traffic scenario. To simulate WWW-like traffic, a number of TCP-Reno sources are connected to each ingress router. The flows arrival rate is modeled as a Poisson process and the flow lengths are drawn from the distribution given in table I. This distribution has been constructed from data collected at the end of 2002 through the Tstat tool [15] on the Internet access link of Politecnico di Torino, i.e. between the border router of Politecnico and the ingress router of GARR, the Italian research network. Collected flow length data have been ordered from the shortest to the longest and divided in 15 groups, each corresponding to a 6.6% probability. Table I reports, for each group, the average flow length measured both in bytes and in IP packets (for simplicity, we have considered 1500 bytes packets). The flows arrival rate is 21 call/s, corresponding to about 64% of the bottleneck link capacity. One continuous backlogged TCP flow has been added.

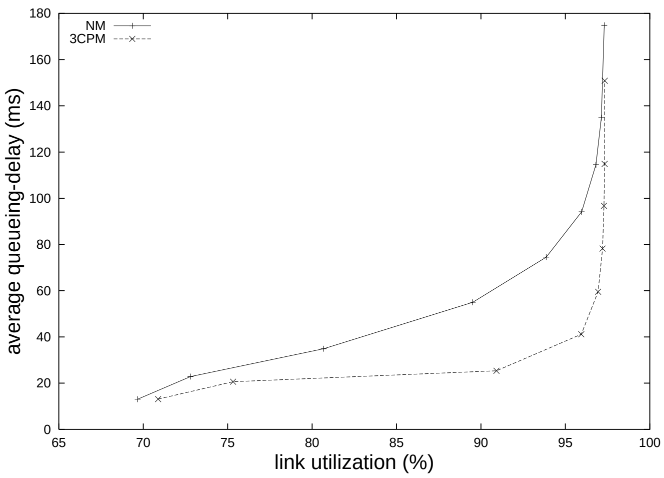

Fig. 4. Delay vs Goodput for long-lived flows

We considered two main performance figures: the average packet delay and, in the second traffic scenario the average flow completion time, i.e. the time from the emission of the SYN packet to the reception of the last data packet (no connection closing has been simulated). These values are plotted versus the average goodput (at the application level) for each of the threshold setting, so we can obtain the "performance frontiers". Simulations were conducted through ns v2.1b9a. We used TCP Reno implementation. For each configuration at least 5 simulations with different random seeds were run. Each simulation lasted 500 simulated seconds, statistics were collected for 500 simulated seconds after discarding the initial 50 seconds. In the figures we present in the following section, standard deviation of goodput is always less than 2% and standard deviation of delay and completion time is always less than 5% of their numerical value.

4. PERFORMANCE EVALUATION

As said in the previous section, throughput and delay performance depend on the RED threshold settings configured by the network administrator. To provide a wider performance evaluation, rather than selecting a given RED configuration, we have run results for several different configurations.

As regards the long-lived flows scenario, figure 4 shows the performance results in terms of the so called "performance frontiers", i.e. delay vs goodput performance. Improvements achieved by 3CPM appear remarkable.

Similar results hold for the second traffic scenario. Nevertheless, in the presence of short-lived flows the average packet delay is not the most significant parameter, but it is preferable to consider average flow completion time. Completion time results are shown in figures 5, for the three cases of i) single packet

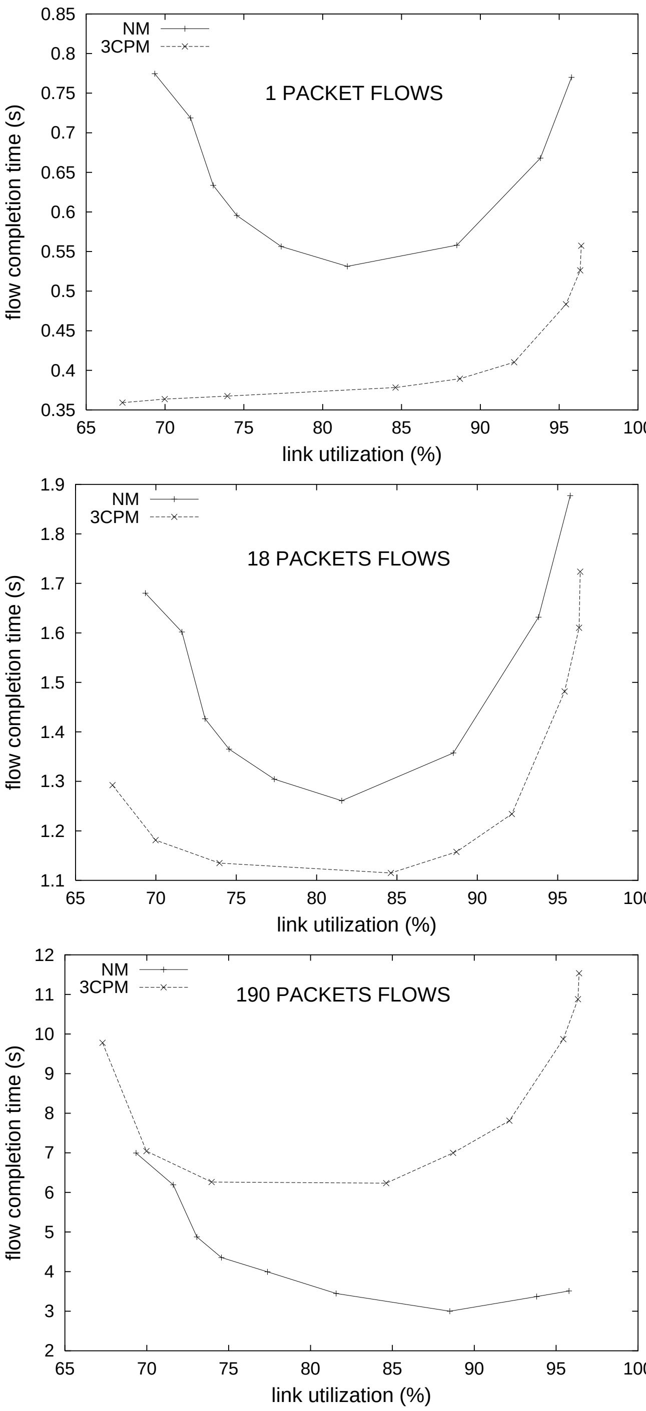

Fig. 5. Flow Completion Time vs GoodPut

flows; ii) 18 packet flows, and iii) 190 packet flows. For all the flow lenghts, 3CPM outperforms NM.

Despite of the 3CPM advantages in comparison to NM, it is also important to evaluate if its higher complexity in comparison to APM and WPM provides real advantages. Figure 6 shows that it is so. As expected, WPM is effective and better than APM when the link utilization is low, essentially because it protects the first packets in the flow, whose loss can be recovered only via retransmission timeout expiration (and not via fast retransmit). At high utilization, this "protection" effect reduces, as packet

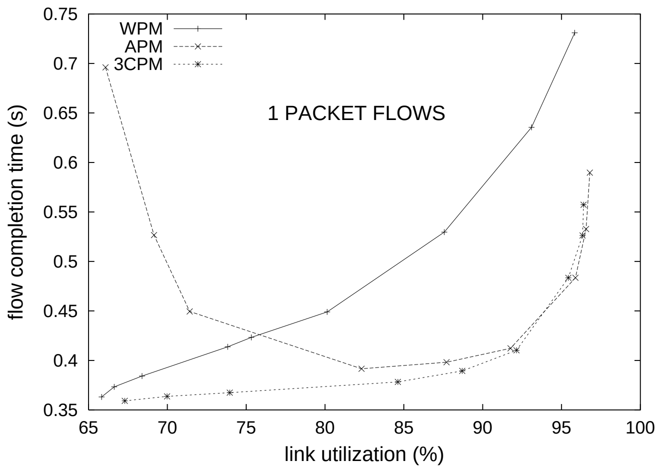

Fig. 6. Flow Completion Time vs GoodPut

TABLE II
SOME PACKETS PERCENTAGE

	NM	WPM	APM	3CPM
green%	–	31-13	–	33-22
yellow%	100	69-87	96-98	63-76
DROP% TOT	5.1-0.1	7.1-0.0	6.8-2.0	6.4-2.0
DROP% green	–	0.0-0.0	–	0.0-0.0
DROP% yellow	5.1-0.1	10-0.0	4.7-0.0	6.0-0.0
DROP% red	–	–	56-99	65-96

loss percentage decreases (results in table II), and queueing delay becomes the main contribute to completion time. For this reason APM provides results consistently better than NM and WPM, because it is the only marking strategy that allows to keep the queue occupancy low in high utilization conditions[4]. Performance evaluation shows that 3CPM inherits good WPM behavior for low link utilization, and good APM behavior for high link utilization, providing the best performance.

Finally, the difference among NM, 3CPM WPM and APM in terms of percentage of green+yellow packets, and percentage of dropped packets appears evident from table II, where their range of variation for the different RIO settings is shown (from lower thresholds to higher one).

5. CONCLUSIONS

In this paper we have presented a new packet marking scheme, which merges two different approaches proposed in literature, and some results of its extensive performance evaluation in an heterogeneous scenario with different TCP flow lengths. Numerical results show that this algorithm provides improved performance in comparison to a no-marker scenario, but also usually better performance to the marking schemes from which it is derived, due to the ortogonality of their respective approaches.

[4] We remark that these insights on the performance of the considered algorithms were made possible only by the choice of presenting results in terms of performance frontiers rather than selecting a specific RED configuration. Indeed, several contrasting results presented in the TCP literature are motivated by different behaviors in different operational conditions - selecting a single RED configuration allows to achieve performance for just a single operational condition.

Since the APM approach is based on some heuristics, 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_Y, and the effect of different initial settings.

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] G. Neglia, G. Bianchi, F. Saitta, D. Lombardo, "Adaptive Low Priority Packet Marking for better TCP performance", Net-con 2002, Paris, October 2002

[11] J. Heinanen, F. Baker, W. Weiss, J. Wroclawski, "Assured Forwarding PHB Group", RFC 2597, June 1999

[12] 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

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

[14] 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

[15] M. Mellia, A. Carpani, R. Lo Cigno, "Measuring IP and TCP behavior on Ege Nodes", Globecom 2002, Taipei, TW, Novemer 2002