An Analytical Model of a new Packet Marking Algorithm for TCP flows - preliminary insights

Giovanni Neglia*, Giuseppe Bianchi§ and Vincenzo Falletta*
*Dipartimento di Ingeneria Elettrica, DIE
Università degli Studi di Palermo
Palermo, Italy
Email: {giovanni.neglia, vincenzo.falletta}@tti.unipa.it
§Dipartimento di Ingegneria Elettronica
Università degli Studi di Roma - Tor Vergata
Rome, Italy
Email: bianchi@elet.polimi.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 analytical model for an adaptive packet marking scheme proposed in our previous work. The model combines three specific sub-models aimed at describing $i)$ the TCP sources aggregate $ii)$ the marker, and $iii)$ the network status. Some preliminary simulative results seem to validate the model.**

I. 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-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 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 II, our marking strategy is based on a somehow opposite philosophy.

In this paper we slightly modify the mechanism proposed in [10], and we describe an analytical model to evaluate the network performance.

The rest of this paper is organized as follows. Section II describes our adaptive packet marking algorithm, focusing on some changes to the previous version. Section

III presents the analytical model, the three submodels are detailed respectively in sections III-A, III-B, III-C. Finally, conclusive remarks and further research issues are given in section VI.

II. The Packet Marking Algorithm (PMA)

In [10], [11] we proposed a new marking algorithm, able to achieve better performance in terms of average queueing delay and flow completion time versus link utilization. According to this marking scheme "long" IN-packets bursts are interleaved with a single OUT packet. The OUT packet is thence employed as a *probe* to early reveal a possible seed of congestion in the network. The algorithm dynamically updates the length of IN-packets bursts by a heuristic estimation of the experienced packet loss ratio.

The idea of marking the majority of packets as IN seems to be in contrast with some results found with other marking scheme [7], [8], [9], but the intrinsic adaptivity of our algorithm is something all these models lack.

If we think about Active Queue Management (AQM) techniques such as Random Early Detection (RED) we observe the same idea of dropping some packets when signals of an incoming congestion are received. Our algorithm moves further: it reallocates losses among the OUT packets, so it spaces them as much as possible, avoiding consecutive losses for a flow and assuring a more regular TCP adaptation behavior.

By simulative evaluation we found better performance when OUT-packets dropping probability is near 100%, while IN packets are not dropped at all.

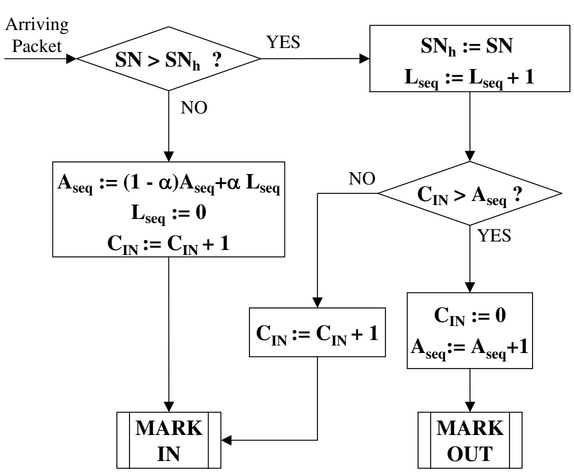

Fig. 1. PMA Flow diagram.

The algorithm flowchart is shown in Fig. 1. Now we will explain how this procedure works. Each time a new SYN packet arrives at the edge router a new state vector is set, containing the following variables:

SN_h: This counter stores the highest Sequence Number (SN) encountered in the flow. It is initially set to the ISN (Initial Sequence Number) value. It is updated whenever a non-empty packet (i.e. non ACK) arrives with a higher SN.

L_{seq}: It is initially set to zero. It is increased by one unit for each new arrived packet (i.e. in-sequence packet), while is reset to zero every time an out-of-sequence packet arrives.

A_{seq}: It stores the average length of in-sequence packet burst between two consecutive losses, using an auto-regressive filter on the previous values of L_{seq}.

C_{IN}: It counts the number of IN-packets in the burst. It is reset to zero when it exceeds A_{seq} and an OUT packet is sent.

The algorithm has been slightly changed in comparison to the version presented in [10], [11]. In the previous algorithm a single variable (L_{IN}) was taking into account the number of in-sequence packets (as L_{seq} actually does) and the number of IN packets of the actual IN-packets burst (as C_{IN} actually does). This coupling required an artificial increase of the variable A_{IN} after marking an OUT packet, we chose $A_{IN} := 2A_{IN} + 1$ but its correct amount was dependant from network condition as it is discussed in [10], [11]. After the introduction of the new variable C_{IN}, a small increase of A_{IN} has been left: it assures better fairness among the flows, allowing flows with underestimated A_{IN} values to faster reach the correct estimate.

III. The analytical model

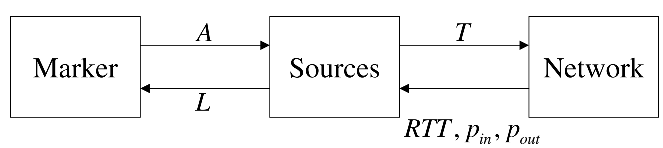

Fig. 2. The three-block model.

The algorithm has shown good performance, but it essentially relies on a heuristic. In order to achieve a deeper understanding and to establish RIO setting criteria, we have developed an analytical model.

The actual model assumes n long-lived homogeneous flows sharing a common bottleneck, whose capacity is c. The model is based on three submodels, which describe respectively the TCP sources, the marker and the network status. Fig. 2 shows the relation among these elements. The number of in-sequence packets L_{seq}

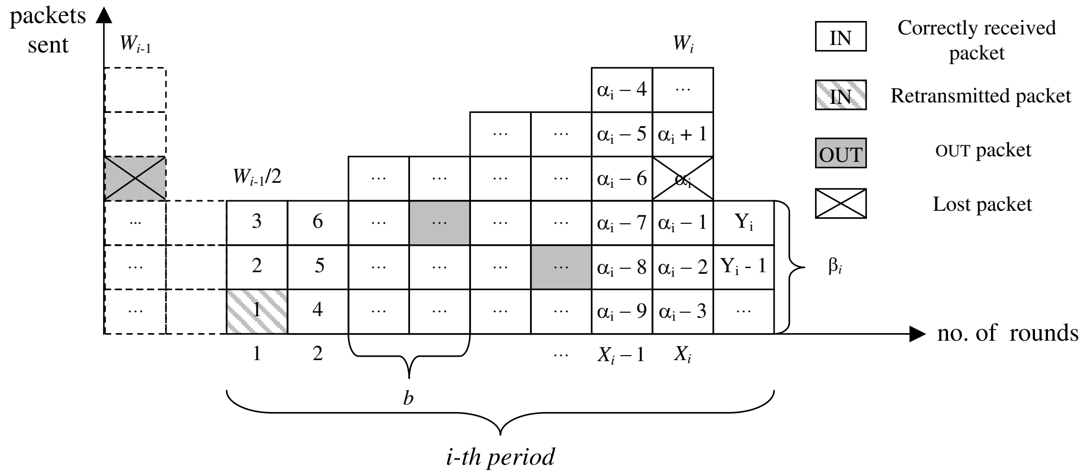

Fig. 3. Timeline and transmitted packets.

is employed by the marker to calculate A_{seq}. The TCP flows are marked according to A_{seq}, hence A_{seq} affects TCP throughput. The TCP traffic congests the network, and produces not-empty queues. Increase of the Round Trip Time and packet losses act as feedback signals for the TCP sources. According to the fixed point approach the influence of each block on the others is considered constant, equal to the average value. The average values for the throughput, A_{seq}, L_{seq} and the Round Trip Time are respectively indicated in Fig. 2 as T, A, L and RTT. At the same time the losses are taken into consideration only through the average dropping probabilities for IN and OUT packets, p_{in} and p_{out}.

We discuss the three submodels in the following subsections. Each of them could be replaced by a more sophisticated one.

A. The Sources Model

According to the previous description, we aim to obtain an expression of the average TCP throughput (T, the input to the Network block) and of the average length of the in-sequence packet burst (L, the input to the Marker block), given the marking profile (A) and the network status (RTT, p_{in}, p_{out}). We have conjectured a regenerative process for TCP congestion window ($cwnd$), thus extending the arguments in [12] to include two different service classes, with different priority levels.

We only consider loss indications due to triple duplicated acks, which turn on TCP fast retransmit mechanism. We don't consider in our analysis the fast recovery mechanism neither the time-out loss events, for the sake of simplicity. As regards time-out neglecting, this approximation appears to be not critical because PMA spaces OUT packets and hence loss events. For this reason errors are usually recovered by fast retransmission, not by time-out. Such intuition is confirmed by our simulation results, where the number of time-outs appear to be significantly reduced in comparison to a no-marker scenario.

A period of our regenerative process starts when the sender congestion window is halved due to a loss indication. Figure 3 shows $cwnd$ trend as rounds succeed. W_{i-1} is the $cwnd$ value at the end of the $(i-1)$-th period, hence in the i-th period $cwnd$ starts from $W_{i-1}/2$ and it is incremented by one every b rounds (b is equal to 2 or 1, respectively if the receiver supports or not the delayed ack algorithm). Notice that, due to neglecting fast recovery and timeouts, each period starts with an IN retransmitted packet, hence the number of packets sent in the period (Y_i) is equal to $L_{seq} + 1$, according to the marker description in section II.

In the i-th period we define also the following random variables: Y_i is the number of packets sent in the period; I_i is the length of the period; β_i is the number of packets transmitted in the last round; α_i is the number of the first lost packet since the beginning of the period, while γ_i is the number of packets transmitted between the two losses occurred in the $(i-1)$-th and in the i-th period. We get $Y_i = \alpha_i + W_i - 1$ and $\alpha_i = \gamma_i - (W_{i-1} - 1)$.

Due to the renewal-reward theorem we can obtain the expression for the average throughput as

$$T(A, RTT, p_{in}, p_{out}) = \frac{E[Y_i]}{E[I_i]}$$

We first compute $E[Y_i]$. The relation between α_i and γ_i allows us to explicit $E[Y_i]$ as a function of the marking profile (A) and the network status (in particular p_{in}, p_{out}). In general $Y_i \neq \gamma_i$, however if we consider their mean values, it holds:

$$\begin{aligned} E[Y_i] &= E[\alpha_i] + E[W_i] - 1 = \\ &= E[\gamma_i] - (E[W_{i-1}] - 1) + E[W_i] - 1 = \\ &= E[\gamma_i] \end{aligned}$$

Let us denote by N the expected value $E[\gamma_i]$. We compute N as:

$$N = \sum_{n=0}^{\infty} np(n) = \sum_{n=0}^{\infty} (1 - P(n)) = \sum_{n=0}^{\infty} Q(n)$$

where $p(n)$ is the probability of losing the n-th packet after $(n-1)$-th successful transmission, $P(n) = \sum_{l=0}^{n} p(l)$ is cumulative distribution function, and so $Q(n) = 1 - P(n)$ represents the probability of *not* losing any packet among these n. If we put n as $n = k(A+1) + h$, with $0 \le h < (A+1)$ we can write $Q(n)$ as

$$Q(n) = s_{in}^{kA+h} s_{out}^{k}$$

where $s_{in} = 1 - p_{in}$, $s_{out} = 1 - p_{out}$. The expression of N can be rewritten as

$$N = \sum_{k=0}^{\infty} \sum_{h=0}^{A} s_{in}^{kA+h} s_{out}^{k}$$

and can be solved in a close form:

$$N = \frac{s_{in}^{A+1} - 1}{s_{in} - 1} \frac{1}{1 - s_{in}^{A} s_{out}} \quad (1)$$

Now we compute $E[I_i]$. Denoting with X_i the round in the i-th period when a packet is lost, we obtain the period length as $I_i = \sum_{j=1}^{X_i+1} r_{ij}$, where $r_{i,j}$ is the j-th round trip time length. Supposing r_{ij} independent of the round number j (i.e. independent of $cwnd$ size), taking expectation we find

$$E[I_i] = (E[X] + 1)E[r]$$

where $E[r] = RTT$ is average round trip time.

In the i-th period $cwnd$ size grows from $W_{i-1}/2$ to W_i with linear slope $1/b$, so[1]

$$W_i = \frac{W_{i-1}}{2} + \frac{X_i}{b} - 1$$

and taking expectation we get

$$E[W] = \frac{2}{b}\left(E[X] - b\right)$$

[1]There are actually different ways to represent $cwnd$ linear growth above the i-th period in the continuous; period bounds are chosen respectively at the beginning of the first round and at the end of the last round, but while in [12] $cwnd$ starts from $W_{i-1}/2$ at the beginning of the period, in our analysis it reaches $W_{i-1}/2$ only after $b/2$ rounds.

To simplify our computations we assume $W_{i-1}/2$ and X_i/b to be integers. Now let us count up all the packets:

$$\begin{aligned} Y_i &= \sum_{k=0}^{X_i/b-1} \left(\frac{W_{i-1}}{2} + k\right) b + \beta_i \\ &= \frac{X_i W_{i-1}}{2} + \frac{X_i}{2}\left(\frac{X_i}{b} - 1\right) + \beta_i \\ &= \frac{X_i}{2}\left(W_{i-1} + \frac{X_i}{b} - 1\right) + \beta_i \\ &= \frac{X_i}{2}\left(W_i + \frac{W_{i-1}}{2}\right) + \beta_i \end{aligned}$$

and taking again expectation it follows

$$N = \frac{E[X]}{2}\left(E[W] + \frac{E[W]}{2}\right) + E[\beta]$$

Assuming β identically distributed between 1 and $W_i - 1$ we can write $E[\beta] = E[W]/2$; therefore, solving for $E[X]$:

$$\begin{aligned} E[X] &= \frac{b}{2}\left(-\frac{2+3b}{3b} + \sqrt{\frac{8N}{3b} + \left(\frac{2+3b}{3b}\right)^2} + 2\right) \\ &= \frac{3b-2}{6} + \sqrt{\frac{2bN}{3} + \left(\frac{2+3b}{6}\right)^2} \end{aligned}$$

then it follows

$$E[I_i] = RTT\left(\frac{3b-2}{6} + \sqrt{\frac{2bN}{3} + \left(\frac{2+3b}{6}\right)^2} + 1\right)$$

Now we can write down the throughput formula:

$$\begin{aligned} T(N, RTT) &= \frac{N}{RTT(E[X]+1)} \\ &= \frac{N}{RTT} \quad \frac{1}{\frac{3b-2}{6} + \sqrt{\frac{2b(N)}{3} + \left(\frac{2+3b}{6}\right)^2} + 1} \end{aligned} \quad (2)$$

Throughput dependance from A, p_{in} and p_{out} is included in N through eq.(1).

Note that if $A_{seq} = A = 0$ (i.e. there is only one class of packets) and $p_{out} = p \to 0$ we get the well-known formula [12]:

$$T(p, RTT) \simeq \frac{1}{RTT}\sqrt{\frac{3}{2bp}}$$

.

Finally, as regards the average length of the in-sequence packet burst (L), from previous remarks it simply follows:

$$L = E[Y_i] - 1 = N - 1 \quad (3)$$

B. The Marker Model

We have discussed before about PMA in this paper, and we have seen how the procedure acts marking one packet OUT every A_{seq} IN, where A_{seq} is obtained filtering L_{seq} with an autoregressive unitary-gain filter. Hence, given A and L respectively the average values of A_{seq} and L_{seq}, they are tied by the relation $A = L^2$. The relation between A_{seq} and L_{seq} has been chosen according to the rationale discussed in section II. Anyway the relation between A and L can be considered a project choice:

$$A = a(L) \tag{4}$$

A change of the $a()$ law leads to a different marking algorithm, for example pursuing a different target.

As regards the fixed-point approach approximation, we observe that the previous relation looks more suitable as long as the system reaches the state where $p_{in} \simeq 0$ and $p_{out} \simeq 1$. In fact, in the case of $p_{in} = 0, p_{out} = 1$ we would have $A_{seq} = L_{seq}$, not simply $A = L$. In [10] and [11] we have shown that the algorithm exhibits optimal performance under *hard differentiation* setting, which leads to $p_{in} \simeq 0$ and $p_{out} \simeq 1$. Hence fixed-point approximation appears justified for PMA.

C. The Network Model

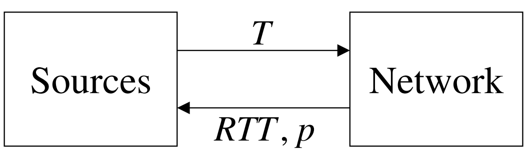

Fig. 4. Interaction between the Network Model and the Sources Model.

The network model has been developed following the approach proposed in [13], which presents a fixed-point model for a best-effort scenario with long-lived flows. The system diagram in Fig. 2 reduces to that in Fig. 4, where no marker appears and there is only one dropping probability p. The dropping probability p and the Round Trip Time RTT can be immediately derived from the queue size. In facts:

$$RTT = R_0 + q/c \tag{5}$$

where c is the bottleneck link capacity and R_0 is the propagation and transmission, and

$$p = H(q) \tag{6}$$

[2]A closer look to the algorithm reveals that this is an approximation due to the update $A := A + 1$ after each OUT-packet transmission.

where $H()$ is referred in [13] as "control function" and depends from the drop module, for example it can be the RED dropping function.

As regards q the authors assume that TCP sources achieve full bottleneck utilization, then for each flow

$$T(p, RTT) = c/n$$

where n is the number of TCP flows. If we denote by $T^{-1}_{RTT}(p, y)$ the inverse function of $T(p, RTT)$ in RTT, then

$$RTT = T^{-1}_{RTT}(p, c/n)$$

From eq.(5), if we consider that q is greater equal than 0 and less equal than the maximum buffer size q_x,

$$q = \max\Big(\min\Big(c\left(T^{-1}_{RTT}(p, c/n) - R_0\right), q_x\Big), 0\Big) \tag{7}$$

This relation is referred in [13] as the 'queue law'. The value of q can be obtained from eq.(5) and eq.(6). In Fig.5 the solution of the two equations is shown as the intersection of the curves $q = G(p)$ and $p = H(q)$.

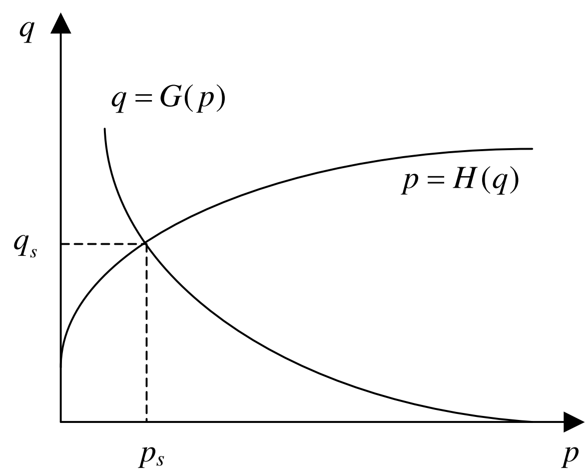

Fig. 5. Steady state (p_s, q_s) as intersection of queue and control laws.

Now we are going to present our extension to this model. In our DiffServ scenario we have two virtual queue q_{in} and q_{out}, and hence two control law H_{in} and H_{out} for IN and OUT packets respectively. According to RIO behavior:

$$\begin{cases} p_{in} = H_{in}(q_{in}), & (i) \\ p_{out} = H_{out}(q_{in} + q_{out}), & (ii) \end{cases} \tag{8}$$

The same arguments of [13] lead to the following relation:

$$\begin{aligned} q_{tot} =& q_{in} + q_{out} = \\ =& \max\Big(\min\Big(q_x, c\left(T^{-1}_{RTT}(N, c/n) - R_0\right)\Big), 0\Big) \end{aligned} \tag{9}$$

where $T^{-1}_{RTT}(N, y)$ the inverse function in RTT of the eq.(2). Note that N depends from A, p_{in}, p_{out}.

The model has 7 variables (q_{in},q_{out},p_{in},p_{out},N,A,L) and 6 equations (1),(2),(3),(4),(8) and (9). We need an equation relating q_{in} and q_{out}, given the traffic offered to the network. If there are not other sources apart from the TCP ones (as we are assuming), a simple relation can be $q_{out} = q_{in}/A$. Usually it holds $A \gg 1$, for this reason we considered $q_{out} \approx 0$.

A further simplification allows us to get again the simple two-variables model in [13]. In fact if H_{in} is invertible, p_{out} is univocally individuated by p_{in}: $p_{out} = H_{out}(H_{in}^{-1}(p_{in}))$. The network is now characterized by the following equations:

$$p_{in}{=}H_{in}(q_{in}) \tag{10}$$

$$\begin{aligned} q_{in}{=}&G(p_{in}) = \\ {=}&\max\Big(\min\Big(q_x, c\left(T_{RTT}^{-1}(N, c/n) - R_0\right)\Big), 0\Big) \end{aligned} \tag{11}$$

Given A, the operation point (q_{in}, p_{in}) can be found setting up an iterative procedure which can be implemented numerically.

As regards the assumption for a RED law of being invertible, we know there are some intervals where this inversion cannot be accomplished (see Fig.6):

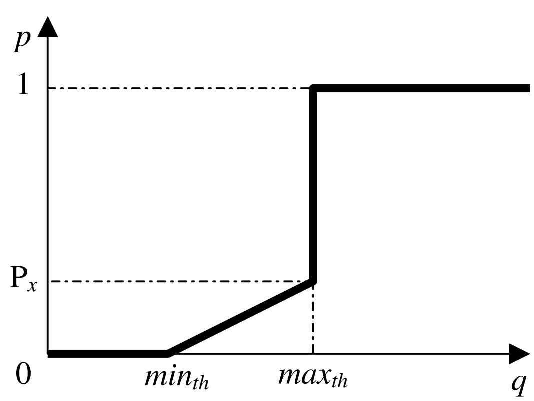

Fig. 6. RED law.

For $0 \leq q \leq min_{th}$ and $max_{th} \leq q \leq q_x$ is not possible to define the inverse function $q = H^{-1}(p)$; we need to introduce a slight slope to eliminate flats, creating a new "RED$_{inv}$" law for IN class, which is invertible.

IV. About the solutions of the system

Summarizing, our model relies on equations (1), (2), (3), (4), (10) and (11). In this section we afford existence and uniqueness of solutions for this system. Let us focus on the expression of the throughput (2). We can express the throughput as a function of q_{in} and q_{out}, it appears that it is a not-increasing monotone function of the queues values. In fact it is immediate to note that RTT is an increasing function of the queues and that N, given A, is a not-increasing function of the dropping probabilities and hence of the queues (1). N is also a not-decreasing function of A if $p_{in} < p_{out}$. The dependance of A from the queues is more complex. From equations (1) and (4), we obtain

$$A + 1 = \frac{s_{in}^{A+1} - 1}{s_{in} - 1} \frac{1}{1 - s_{in}^{A} s_{out}} \tag{12}$$

Remember that our PMA is described by $A = L$, i.e. $A = N - 1$. A is solution of the above equation. It can be shown that if $p_{in} < p_{out}$ A is a decreasing function of p_{in} and p_{out}, hence a not-increasing function of q_{in} and q_{out}.

According to the considerations in the previous section, we can consider only dependance from q_{in}.

The following results hold:

$$\lim_{q_{in} \to 0} T(q_{in}) = +\infty$$

$$\lim_{q_{in} \to +\infty} T(q_{in}) = 0$$

If H_{in} and H_{out} are continuous functions, also $T(q_{in})$ is a continuous functions.

From the previous considerations and hypotheses it follows that the system admits at least one solution, i.e. it exists always a value q_{in}, such that $T(q_{in}) = c/n$ and all the equations are satisfied. One has only to verify that $q_{in} < q_x$. Being the throughput a not-increasing monotone function of the queues values, the solutions set is an interval (eventually reducing to a single point). Finally if H_{in} is a strictly increasing function of q_{in} and $0 < p_{in}, p_{out} < 1$, the throughput is a strictly decreasing function of the queue and hence the solution is unique. It is possible to set up an iterative procedure to find numerically this solution, and this is just what we did using MATLAB.

Now we want to address the solutions in a particular context. Let us remove the previous hypothesis about invertibility and consider equation (12). We consider the RIO settings in Fig.7, where $max_{out} < min_{in}$. In the range $[max_{out}, min_{in}]$, $s_{in} = 1$ and $s_{out} = 0$, hence equation (12) reduces to an identity, and the system admits as solution the whole set of values $[max_{out}, min_{in}]$. In Fig.7 we have put in evidence this interval.

As we said, we have introduced a slight slope to RED, in order to make it invertible. Hence the previous range should reduce to a point near the value max_{out}. Despite of this, the MATLAB procedure is affected by numerical approximations, in particular p_{in} can be undistinguishable from 0 in the range $[max_{out}, min_{in}]$, so also the MATLAB procedure can find different solutions in this range, depending on the initial conditions chosen. In particular, unless we start from a point inside the range, the system will converge to the left or right extremity of the interval.

When the model predicts a range of solutions, the dynamics of the system play a fundamental role to determine the final solution. Such dynamics are not considered in a fixed point approach. According to a preliminary study it appears that the model exhibits a higher sensitivity to state perturbations for higher values of the queues, this could justify the simulation results and it suggests that the system dynamics could be recovered by inserting in the numerical procedure a sort of model noise.

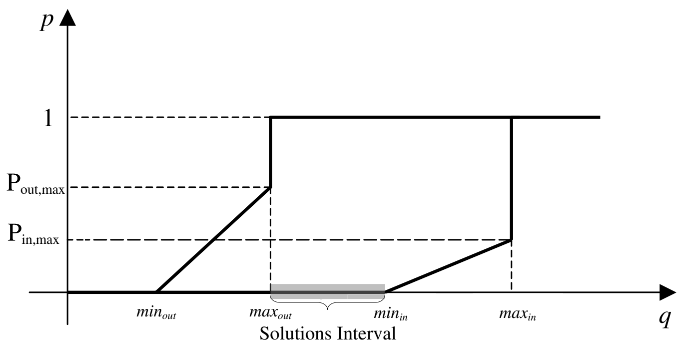

Fig. 7. Solutions Interval.

V. Model validation

To validate our model we considered the network topology showed in Fig.8, which is the same encountered in [10], consisting of a single bottleneck link with capacity equal to 6Mbps. The Round Trip Time goes from

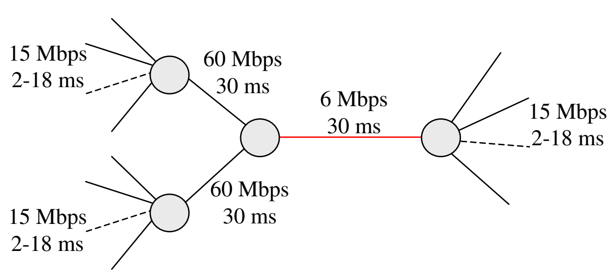

Fig. 8. Network topology.

128ms to 192ms, for an average value of R_0=160ms. The IP packet size is chosen to be 1500 Bytes, for a bottleneck link capacity of c = 500packets/s. We started three different simulation sets, each one related to a different way of configuring RIO thresholds.

We denote as *overlapping* a RIO configuration in which $min_{in} < max_{out}$. We set $max_{out} = 4min_{out}$, $max_{in} = 4min_{in}$ and $min_{in} = 3min_{out}$. We denote as *contiguous* a configuration where RIO thresholds are set so that $max_{out} = min_{in}$. We set $max_{out} = 3min_{out}$, $max_{in} = 3min_{in}$. Finally we denote as *non overlapping* a RIO configuration in which $max_{out} < min_{in}$ (i.e. a configuration in which a GSI exist), more precisely we choose $max_{out} = 3min_{out}$, $max_{in} = 3min_{in}$ and $min_{in} = 4max_{out}$. For each of this settings criteria, we have tested five different configurations, varying min_{out} from 2 up to 32.

We ran our simulations using ns v2.1b9a, with the Reno version of TCP. In Table I we report the results of our analysis in terms of queue occupation for all tested configurations, while in Fig.9 , 10 and 11 we can see the same results in a more readable form.

As regards the *overlapping* and *contiguous* RIO configurations, the analytical model predicts a unique solution, according to the considerations in section IV. Figures 9 and 10 show that model results are quite accurate if compared to simulation results.

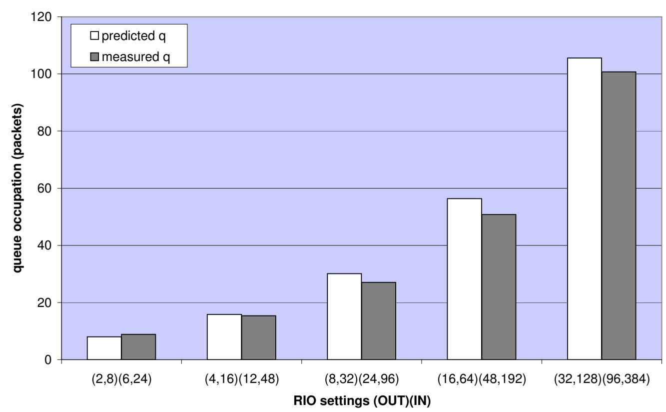

Fig. 9. Queue occupation vs RIO settings (*overlapping*).

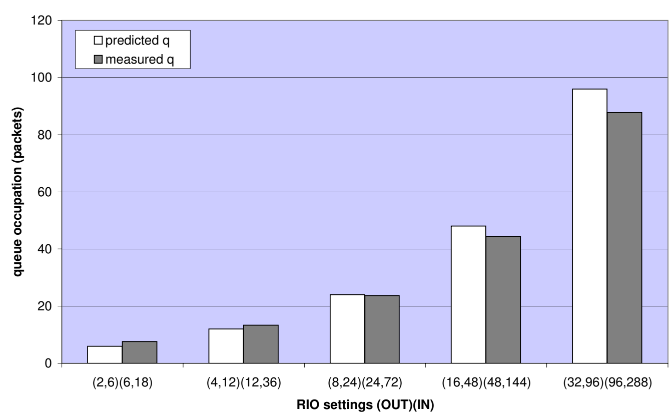

Fig. 10. Queue occupation vs RIO settings (*contiguous*).

As regards the *non overlapping* RIO settings, due to the small slope of the "RED$_{inv}$" curve, the solution

TABLE I

QUEUE OCCUPATION VS RIO SETTINGS

RIO (*non overlapping*)	predicted q	measured q	RIO (*contiguous*)	predicted q	measured q	RIO (*overlapping*)	predicted q	measured q
(2,6)(8,24)	{6;8}	8,29	(2,6)(6,18)	6	7,59	(2,8)(6,24)	8,00	8,87
(4,12)(16,48)	{12;16}	12,90	(4,12)(12,36)	12	13,32	(4,16)(12,48)	15,84	15,39
(8,24)(32,96)	{24;32}	27,77	(8,24)(24,72)	24	23,68	(8,32)(24,96)	30,07	27,01
(16,48)(64,192)	{48;64}	54,87	(16,48)(48,144)	48	44,39	(16,64)(48,192)	56,36	50,76
(32,96)(128,384)	{96;128}	108,85	(32,96)(96,288)	96	87,77	(32,128)(96,384)	105,61	100,69

interval coincides with the range $[max_{out}, min_{in}]$, as it is shown in Table I. According to the initial value of A our iterative procedure converges to the lower or to the higher value of the solution interval. Actually, with a starting value of $A_0 = 7$ the procedure converges to the lower value, while with $A_0 = 300$ the higher value is held. In Fig.11 we reported for each setting the measured queue occupation (got from ns simulations) and both lower and higher predictions (respectively horizontal and vertical lines in the bars). The fact that the measured

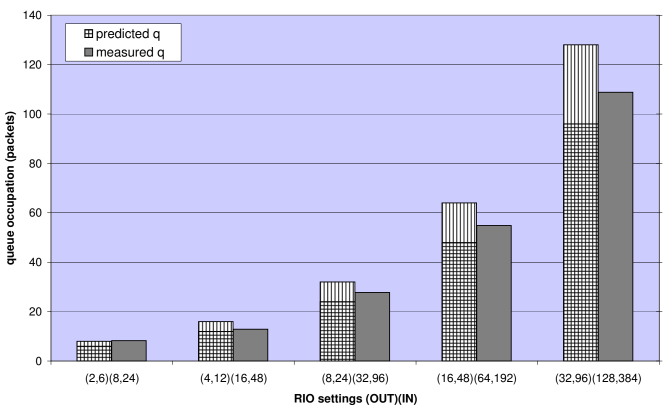

Fig. 11. Queue occupation vs RIO settings (*non overlapping*).

solutions stay between the predicted values is a proof that our speculations are correct. Yet we cannot say anything about the real adaptive dynamics, whose inclusion in the model will be one of our future issues to investigate.

VI. CONCLUSIONS AND FURTHER RESEARCH ISSUES

In this paper we have presented an analytical model for our adaptive packet marking scheme proposed in previous works. Preliminary simulative results seem to validate model predictions about average queue occupancy. We are going to extend simulative evaluation.

Future research will address the extension of the model in order to include throughput predictions. Besides the model will employed to establish optimal RIO settings according to different performance criteria (throughput, fairness, delay and flow completion time) and to improve the same marking algorithm.

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] G. Neglia, G. Bianchi, M. Sottile, “Performance Evaluation of a new Adaptive Packet Marking Scheme for TCP over DiffServ Networks” Globecom 2003, dicembre 2003, San Francisco, USA

[12] J. Padhye, V. Firoiu, D. Towsley, J. Kurose. “Modeling TCP Throughput: A Simple Model and its Empirical Validation”, May 1998

[13] V. Firoiu, M. Borden, “A Study of Active Queue Management for Congestion Control”, 2000