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

Giovanni Neglia, Vincenzo Falletta

Dipartimento di Ingegneria Elettrica, DIE
Università degli Studi di Palermo
Palermo, Italy

Giuseppe Bianchi

Dipartimento di Ingegneria Elettronica, DIE
Università degli Studi di Roma - Tor Vergata
Roma, Italy

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. Preliminary simulative results show quite accurate predictions for throughput and average queue occupancy. Besides, the research suggests new interesting guidelines to model queues fed by TCP traffic.

Key words: TCP Marking, Differentiated Services, Models

1 Introduction

Differentiated Services (DiffServ) networks provide the ability to enforce a different forwarding behavior to packets, based on their Differentiated Services

Email addresses: giovanni.neglia@tti.unipa.it (Giovanni Neglia),
vincenzo.falletta@tti.unipa.it (Vincenzo Falletta),
bianchi@elet.polimi.it (Giuseppe Bianchi).

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 [1–16] 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 [17–19]. 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. Our algorithm was first proposed in [20] and share this aim. An introductory comparison with the other marking algorithms is presented in section 2.

In this paper we slightly modify the mechanism proposed in [20], and we describe an analytical model to evaluate the network performance. This model can be employed to study possible variants of the algorithm. By the way, the network sub-model exhibits some novelty in comparison to previous approaches and could be useful in different network scenarios where TCP traffic is considered.

The rest of this paper is organized as follows. After an overview of proposed marking schemes in section 2, section 3 describes our adaptive packet marking algorithm, focusing on some changes to the previous version. Section 4 presents the analytical model which relies on the Fixed Point Approximation, whose rationale and whose employment in computer networks field are shortly introduced in subsection 4.1. The three submodels are detailed respectively in subsections 4.2, 4.3, 4.4, while in subsection 4.5 existence and uniqueness of a solution are proven. Section 5 deals with validation of the proposed model. A simple application of the model to evaluate the performance of a variant of the algorithm is presented in section 6. Finally, conclusive remarks and further research issues are given in section 7.

2 Related works

The idea to employ marking mechanisms for service differentiation was first introduced in [1], where the authors propose a time-sliding window marker. In [2] token bucket appears to achieve better performance in comparison to time-sliding window. At the same time the authors claim that marking cannot offer a quantifiable service to TCP traffic due to the interaction of TCP dynamics with priority dropping: when IN packets and OUT packets are mixed in a single TCP connection, drops of OUT packets negatively impact the connection's performance. Afterwards token bucket and time-sliding window markers have been extended to three colors [3–5].

Following studies confirm the difficulty of marker configuration. A detailed experimental study of the main factors that impact the throughput of TCP flows in a RIO based DiffServ network is provided in [6]. The article shows that in an over-provisioned network all target rates are achieved, but unfair shares of excess bandwidth are obtained. However, as the network approaches an under-provisioned state, not all target rates can be achieved. In [7] it is shown that it is possible to improve the throughput significantly even when a small portion of traffic is sent as in-profile packets. At the same time the authors observe that, in order to fully utilize the benefit of out-profile packets, the amount of out-profile packets sent in addition to the in-profile packets has to be carefully determined. In [8] a set of experimental measures is presented. The main result is that the differentiation among the transmission rates of TCP flows can be achieved, but it is difficult to provide the required rates with a good approximation. In [9] the limits of token bucket are deeply investigated. It appears that (i) the achieved rate is not proportional to the assured rate, (ii) it is not always possible to achieve the assured rate and, (iii) there exist ranges of values of the achieved rate for which token bucket parameters have no influence.

These results suggested the need to introduce some adaptivity in order to cope with TCP dynamics. In [10] the Packet Marking Engine monitors and sustains the requested level of service by setting the DS-field in the packet headers appropriately. If the observed throughput falls below the minimum target rate the Engine starts prioritizing packets until the desired target rate is reached. Once the target is reached, it strives to reduce the number of priority packets without falling below the minimum requested rate. The Active Rate

Management is proposed in [11] in order to provide minimum throughputs to traffic aggregates. It is a classical, linear, time-invariant controller, which sets the token bucket parameters (specifically the token bucket rate) adapting to changes in the network. The same issue is tackled by [12]. The adaptive dual token bucket in [13] regulates the amount of OUT packets in order to prevent TCP packet losses caused by excess low-priority traffic in the network. This adaptive technique requires a congestion signaling procedure from internal routers to border routers.

The Equation-Based Marking [14] is someway similar to ours because it senses the current network conditions, in particular it estimates the loss probability and the Round Trip Time (RTT) experienced by a TCP flow (without any signaling with core routers), and adapts the packet marking probabilities accordingly. In particular it uses the TCP model in [21] and these estimates in order to identify the target loss probabilities, corresponding to target throughput rates. Then, it uses the current loss probability estimate as well as these target loss probabilities to calculate the packet-marking probabilities. Main targets are fairness among heterogeneous TCP flows and protection against non-assured traffic. Fairness is also the main focus of [15] and [16]: the first concentrates on the effect of different RTTs, the second propose the Direct Congestion Control Scheme to achieve fairness between responsive and unresponsive aggregates.

The proposals described above share the purpose to assure a minimum throughput to TCP individual flows or aggregates. As we said in the previous section, TCP marking has also been proposed as a way to achieve better than best effort performance [17–19]. In particular [17] focuses on WWW traffic and proposes two packet marking schemes. The first one is tightly integrated with the TCP protocol: the source is allowed to send up to N_s IN packets when it starts, and 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 scheme does not require the knowledge of internal TCP variables, but it uses a constant value $N_a = N_s = 5$, hence this scheme can be implemented at ingress router. The rationale behind the schemes in [17] is that packets marked as IN will be protected against network congestion, hence marking can be useful employed to protect flows with small window or retransmitted packets, when packet losses cannot be recovered via the fast retransmission algorithm but trigger timeouts, which reduce TCP source throughput.

The TCP-friendly marker in [18,19] considers long lived flows and adopt goodput and loss as performance metrics. The main guidelines are: 1) to protect small-window flows and retransmitted packets from losses by marking them IN; 2) to avoid, if possible, to mark OUT consecutive packets in order to reduce the possibility of burst loss of packets. Our approach share the purpose to space as much as possible packet losses, at the same time many differences hold. In the TCP-friendly marker a fixed number of IN tokens is available for each time interval and it has to be distributed among the flows, on the contrary our scheme adaptively set the length of IN packets burst (i.e. the number of flow consecutive packets that are marked IN) according to the network status. Besides, all the marking schemes share the idea that packets marked IN will be protected against network congestion, while our algorithm operates according to the someway opposite philosophy to employ OUT packets as *probes* (see section 3). Finally, our approach is much simpler.

We present some further remarks about the previous algorithms in order to stress the novelty of our approach. In a DiffServ Assured Forwarding (AF) scenario, the differentiation between traffic classes is relative. For example usual RIO configuration [1] assures that IN packets dropping probability is lower than OUT packets one, but no bound is guaranteed. For this reason the protection of IN packets in [17] relies on the assumption that most of the packets in the network are of type OUT, hence IN packets will receive a "good-enough" service. In fact in [17] the authors show that a throughput reduction may be encountered as long as the percentage of IN traffic becomes greater than a given threshold. The authors claim that the problem is interleaving IN and OUT packets, when the loss rate of the OUT traffic is much larger than that of the IN traffic. We want to stress that the IN packet protection vanishes as IN traffic increases. 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 [18,19] (protection of small window and retransmitted packets, an OUT packet inserted every n IN packets).

Hence our approach shows two main differences [20]: 1) the majority of packets are IN, 2) the performance takes advantage of a very high OUT packet loss rate. The apparent conflict with results in [17] and with similar results for the marker proposed in [18,19] relies on the adaptivity. These schemes are not designed to be adaptive to the network congestion status, while ours uses some heuristics to provide adaptivity.

3 The Packet Marking Algorithm (PMA)

In [20,22] 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 [18,19,17], 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.

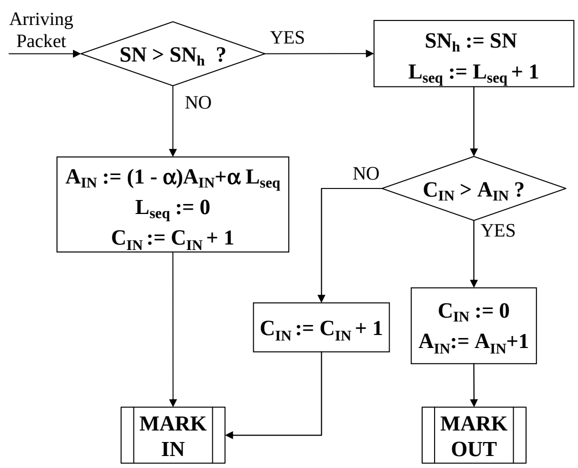

Fig. 1. PMA Flow diagram.

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 [1] .

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.

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

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

A_{IN}: It stores the number of packets which will be marked IN, it tracks the average length of in-sequence packet bursts through autoregressive filtering.

The algorithm has been slightly changed in comparison to the version presented in [20,22]. 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 [20,22]. 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.

4 The analytical 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 model assumes n long-lived homogeneous flows sharing a common bottleneck, whose capacity is C. The model is based on a Fixed Point Approximation (FPA), a modeling technique described in the following subsection. According to FPA the system is divided into its three main components as shown in Fig. 2: the TCP sources, the network and the marker. Each element is modeled separately, taking into account the effects of the others through the parameters shown in figure. For example TCP sources depend on the network by the RTT and the dropping probabilities p_{in} and p_{out}, and on the marker by the length of IN packet bursts (A_{IN}).

After an overview of FPA methods in section. 4.1, the submodels for the TCP sources, for the marker and for the network are respectively presented in sections 4.2, 4.3 and 4.4. Each of them could be replaced by a more sophisticated one.

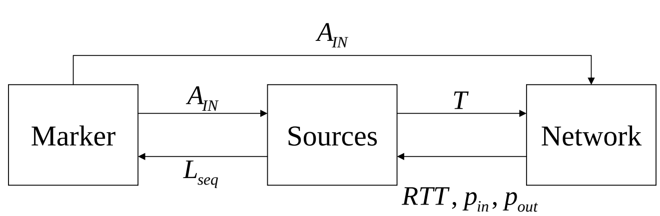

Fig. 2. The three-block model.

4.1 About Fixed Point Approximations

The expression *Fixed Point Approximation* (FPA) refers to a particular modeling technique, which we are going to describe in this section. This name is quite spread in scientific literature [23–25], but also other names appear: fixed point models [26,27], fixed point approach [28], reciprocal model tuning [29]. Other papers [30,31] refer the expression "fixed point" to the specific method employed to solve the model system of equations, rather than to the modeling technique.

This section is organized as it follows. Firstly we introduce the idea of FPA with reference to our specific problem, and we explain the origin of the expression *fixed point*. Secondly we briefly present telecommunications works employing this kind of modeling technique. For a more detailed overview of FPA in the field of computer networks refer to [32].

Let us consider a single bottleneck network, where a single TCP flow is marked at the edge and feed the queue at the bottleneck. Suppose we are interested into some average values, like TCP throughput or queue occupancy. If we know all the parameters characterizing the network (i.e. link capacities, link delays, buffer size) the TCP sender (e.g. the TCP version, the maximum congestion window size, the timer granularity), the TCP receiver and the marker, we are able to describe exactly the behavior of each element of the network and to evaluate the throughput of the TCP sender or the queue occupancy at each time. If we were able to describe the evolution of these quantities in a closed form, we could evaluate their average value, by integrating the analytical expressions, but in general it is not the case.

In order to achieve our purpose we have to sacrifice the exact description of the system. A way to make the problem analytically tractable is to divide the system into three parts (e.g. the TCP source, the queue at the bottleneck and the marker), to assume some simplifying assumptions about their interaction, and then to develop an analytical model for each part.

According to the FPA approach, the main assumptions are that we model each part considering the other in a steady state, and that this state is independent by the behavior of the part we are modeling. In our example we know that the throughput of the TCP source is dependent from the path current RTT, from packet discard at the queue and from the marking pattern (characterized by A_{IN}). At the same time the TCP traffic generates the queue in the network and causes eventually packet discard when the buffer is full. Nevertheless, in order to model the TCP behavior, we assume that the network and the marker are in a steady state: specifically we consider that RTT and A_{IN} are constant ($A_{IN} = A$), and that the packet discard for both IN and OUT packets are bernoullian processes respectively with mean values p_{in} and p_{out}. Different assumptions can be done. Anyway these allows us to derive an expression for the long-term steady-state TCP throughput as a function of RTT, p, A, say:

$$T = f(RTT, p_{in}, p_{out}, A) \tag{1}$$

and an expression for the average number of in-sequence packets as:

$$L = g(p_{in}, p_{out}, A) \tag{2}$$

In the same manner, in order to model the network, we assume the TCP source offers a constant traffic intensity to the network, independently from the present network status (queue occupancy and packet discard probability), with a ratio of IN packets to OUT ones equal to A. If we add some further hypothesis about the statistical characterization of packet arrivals at the buffer and the way IN and OUT packets are interspersed, we are able to derive the mean number of packets in the router and hence the average RTT and the mean dropping probabilities (p_{in},p_{out}), i.e.:

$$p_{in} = h_{in}(T, A) \tag{3}$$
$$p_{out} = h_{out}(T, A) \tag{4}$$
$$RTT = l(T, A) \tag{5}$$

Finally given the average number of in-sequence packet (L), we can derive the average length of IN packet bursts.

$$A = m(L) \tag{6}$$

In order to determine T, RTT, p_{in}, p_{out}, L, A, we need to solve the system of equations (1),(2),(3), (4),(5) and (6). If we define the function $U : \Re^6 \rightarrow \Re^6$ as

$$\begin{aligned} &U(T, L, p_{in}, p_{out}, RTT, A) = \\ &= [f(RTT, p_{in}, p_{out}, A), g(p_{in}, p_{out}, A), h_{in}(T), h_{out}(T), l(T), m(L)] \end{aligned} \tag{7}$$

then we can note that a solution of such system ($[T^*, L^*, p^*_{in}, p^*_{out}, RTT^*, A^*]$), if any, satisfies the following relation:

$$[T^*, L^*, p^*_{in}, p^*_{out}, RTT^*, A^*] = U(T^*, L^*, p^*_{in}, p^*_{out}, RTT^*, A^*), \tag{8}$$

i.e. the point $[T^*, L^*, p^*_{in}, p^*_{out}, RTT^*, A^*]$ is a fixed point for the $\Re^6 \rightarrow \Re^6$

mapping, established by the function U [2]. This remark justifies the name of FPA.

Under some proper conditions about the function U and its definition set, fixed-point theorems can be used to conclude that at least a solution exists, like the Bolzano's theorem, the Brouwer's one and the Kakutani's one (see for example [33] or [34]). The question of uniqueness is more difficult, eventual monotonicity greatly constraints the possible dynamics.

Different methods can be employed in order to solve Eq. (8). In particular repeated substitution takes into account the following relation:

$$[T_{i+1}, L_{i+1}, p_{in,i+1}, p_{out,i+1}, RTT_{i+1}, A_{i+1}] = U(T_i, L_i, p_{in,i}, p_{out,i}, RTT_i, A_i), \quad (9)$$

assuming that

$$\lim_{i->\infty}[T_i, L_i, p_{in,i}, p_{out,i}, RTT_i, A_i] = [T^*, L^*, p^*_{in}, p^*_{out}, RTT^*, A^*].$$

This kind of solution is particularly appealing, because Eq. (9) can be read as a dynamical system, describing the network operation [24,35]: in our example if the network is unloaded ($p_{in} = p_{out} = 0$) and the TCP source starts injecting traffic in the network, the buffer provides new (different) value of p_{in} and p_{out} by dropping packets. The source reacts to this packet loss probability adjusting its sending rate and at the same time the marker changes its marking profile until convergence is reached. Despite this striking interpretation, it is not clear how close Eq. (9) actually describes the network operation. By the way, convergence of Eq. (9) is not guaranteed.

Some other kind of approximations are often employed together with FPA and they cannot often be easily distinguished from a FPA approach extended to all the network elements, i.e. when we divide the network into as many parts as the number of TCP sources plus the number of the network routers. One example is the Mean Field Theory (also known under many names and guises, e.g. Self Consistent Field Theory, Bragg-Williams Approximation, Bethe Approximation, Landau Theory) whose simplifies a many-body interactions problem by replacing all interactions to anyone body with an average or effective interaction. The Mean Field Theory is explicitly referenced in [36,37] as a way to model the interaction of many TCP flows. Another common assumption concerns the networks of queues, and it is known as *Kleinrock's independence approximation* [38]. Also in a network of queues there is a form of interaction, in the sense that a traffic stream departing from one queue enters one or more

[2] Note that in what follows we will introduce for convenience other variables, but nothing changes as regards the idea of FPA described here.

other queues, perhaps after merging with portions of other traffic streams departing from yet other queues. Analytically, this has the unfortunate effect of complicating the character of the arrival process at downstream queues. Kleinrock suggested that merging several packet streams on a transmission line has an effect akin to restoring the independence of interarrival times and packet lengths. It was concluded that it is often appropriate to adopt $M/M/1$ queueing model for each communication link regardless of the interaction of traffic on this link with traffic on other links.

The employment of FPA techniques to model networks is not a novelty. For example there is a considerable body of literature on the application of fixed point methods to estimating blocking probabilities in circuit switched networks (see [39] for some applications). More recently FPA has been widely used to model the interaction of TCP sources with the network (see [26,31,35,40,23] [27,41,29,28,24]). [26] has probably the merit to be the first paper where the FPA approach is clearly stated and presented as a method "which allows the adaptive nature of TCP sources to be accounted for". Regarding network models [31,35,40,23,27,41] do not need stochastic queue model, but they essentially rely on the assumption that long-lived TCP flows are able to achieve full bandwidth utilization. Aside from [27,41] they consider AQM mechanisms relating the dropping probability and the queue occupancy. [27] considers zero buffer queue and [41] considers large delay-bandwidth networks in order to neglect queueing delay. Multi-bottleneck networks are considered in [40,23,27,41], and the existence of a solution is proved in [41] under the above simplification. The hypothesis of full bandwidth utilization is removed in [26,29,28,24], which consider Poisson arrival at the queue. In [28] each buffer is modeled as a $M/M/1/K$ queue or as a $M^{[X]}/M/1/K$ queue with batch arrivals. The paper discusses the admissibility of the Poisson hypothesis and prove the existence and the uniqueness of the solution when the nominal load is less than one for short and long lived TCP flows. A more detailed investigation of the existence, the uniqueness and the stability of equilibrium points appears in [24] for a single-bottleneck scenario and short-lived flows.

As a final remark we note that there has been related work focusing on the development and solution of a set of differential equations describing the transient behavior of TCP flows and queue dynamics [42]. FPA complements this approach. The fixed point approach is much more efficient computationally as the number of unknowns equals the number of links in the network, whereas the differential equations approach requires the solution of a number of equations equal to the number of links plus the number of TCP flows. On the other hand, the differential equations approach can be used to study transient behavior.

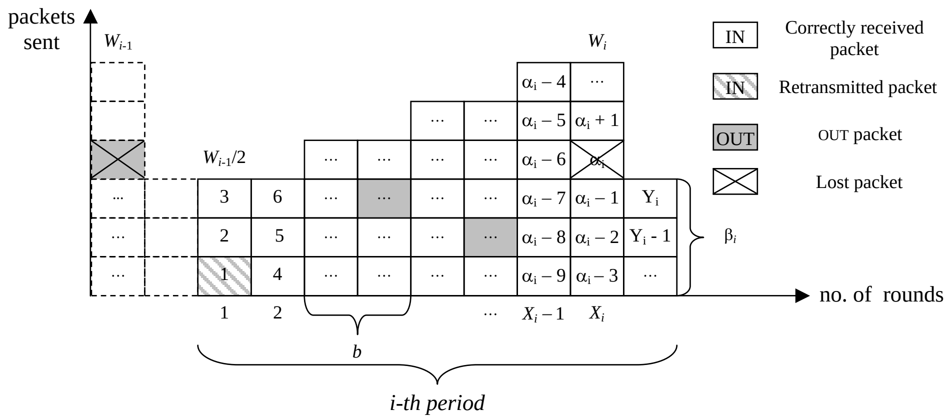

Fig. 3. Timeline and transmitted packets.

4.2 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 [21] to include two different service classes, with different priority levels.

In our analysis we neglect slow start operation and time-out events, we only consider loss indications due to triple duplicated acks, which turn on (always successful) TCP fast retransmit mechanism. 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. Fig. 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 our assumptions on TCP operation, 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 4.3.

In the i-th period we define also the following random variables: 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 of n sources sharing the same path:

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

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:

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

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 $0 \leq h < (A+1)$ 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:

$$N = \frac{s_{in}^{A+1} - 1}{s_{in} - 1} \frac{1}{1 - s_{in}^{A} s_{out}} \tag{10}$$

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 $I_i = \sum_{j=1}^{X_i+1} r_{i,j}$, 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 [3]

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

and taking expectation we get

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

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:

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

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 $E[X]$:
$E[X] = \frac{b}{2}\left(-\frac{2+3b}{3b} + \sqrt{\frac{8N}{3b} + \left(\frac{2+3b}{3b}\right)^2} + 2\right) = \frac{3b-2}{6} + \sqrt{\frac{2bN}{3} + \left(\frac{2+3b}{6}\right)^2}$
then it follows

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

[3] The formula is a linear approximation of the exact relation $W_i = W_{i-1}/2 + \lceil X_i/b \rceil - 1$. In [21] a different approximation has been considered.

Now we can write down the throughput formula:

$$T(N, RTT) = n\frac{N}{RTT(E[X]+1)} = \frac{nN}{RTT}\frac{1}{\frac{3b-2}{6} + \sqrt{\frac{2bN}{3} + \left(\frac{2+3b}{6}\right)^2} + 1} \quad (11)$$

Throughput dependance from A, p_{in} and p_{out} is included in N through Eq. (10).

Note that if $A_{IN} = A = 0$ (i.e. there is only one class of packets) and $p_{out} = p \to 0$ we get the well-known formula [21]:

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

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

$$L = E[Y_i] - 1 = N - 1 \quad (12)$$

4.3 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_{IN} IN packets, where A_{IN} is obtained filtering L_{seq} with an autoregressive unitary-gain filter. Hence, given A and L respectively the average values of A_{IN} and L_{seq}, they are tied by the relation [4] :

$$A = L \quad (13)$$

The relation between A_{IN} and L_{seq} has been chosen according to the rationale discussed in section 3. Anyway the relation between A and L can be considered a project choice:

$$A = m(L) \quad (14)$$

A change of the $m()$ law leads to a different marking algorithm, for example pursuing a different target. We are going to show an example in section 6.

As regards the fixed-point approach approximation, we observe that the previous relation looks more suitable as long as the system reaches the state

[4] 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 $p_{in} \simeq 0$ and $p_{out} \simeq 1$. In fact, in the case of $p_{in} = 0, p_{out} = 1$ we would have $A_{IN} = L_{seq}$, not simply $A = L$. In [20] and [22] 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.

4.4 The Network Model

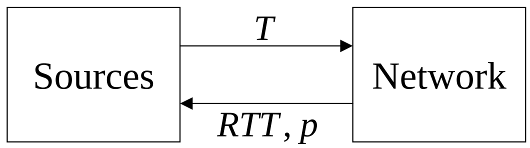

Fig. 4. Interaction between the Network Model and the Sources Model.

In [43] we have proposed a network submodel, extending the approach proposed in [35] for a best-effort scenario to a DiffServ one, where routers deploy RIO (we indicate the configuration parameters as $(min_{out}, max_{out}, Pmax_{out})$ and $(min_{out}, max_{out}, Pmax_{out})$ respectively for OUT and IN packets [1]). A limit of that approach is that TCP sources are intrinsically assumed to achieve full bottleneck utilization (assumption also in [31,35,40,23,27,41]), hence the model is able to predict average queue occupation, not link utilization. Besides the model in [43] predicts a range of solutions when $max_{out} < min_{in}$. These problems could be overcome introducing in the model queue variability.

Anyway in this paper the approach is radically different, we consider that the queue can be modelled as a M/M/1/K queueing system. This allows us to evaluate the stationary distribution of the queue for a given offered load T, and then the average values we are interested in, i.e. RTT, p_{in} and p_{out}.

As regards the assumption of Markovian arrivals, it seems to be justified when the TCP connection rate increases [44]. Anyway M/M/1/K models have been widely employed in literature and have shown good performance [45,26,30,46,47]. In particular our framework is similar to those of [30] and [47], which model respectively Token Bucket and Single Rate Three Color Marker, but it differentiates because it assumes state dependent arrivals, rather than uniform ones.

These models take into account the presence of different class of traffic and the effect of AQM mechanism like RIO, but they assume that dropping probability depends only on the instantaneous queue size, disregarding the effect of filtering.

According to [30], the stationary distribution of the queue can be evaluated as:

$$\pi(i) = \pi(0) \left(\frac{T}{C}\right)^i \prod_{j=0}^{i-1} (1 - p(j)), i = 1, 2, ...max_{in}$$

where C is the bottleneck capacity, $\pi(0)$ is given by the normalization equation

$$\pi(0) = \left(1 + \sum_{i=1}^{max_{in}} \left(\frac{T}{C}\right)^i \prod_{j=0}^{i-1} (1 - p(j))\right)^{-1}$$

and

$$p(i) = \frac{T_{in} p_{in}(i) + T_{out} p_{out}(i)}{T_{in} + T_{out}} = \frac{A p_{in}(i) + p_{out}(i)}{A + 1}$$

Note that we assumed $max_{out} < max_{in}$, and that it is useless considering queue values greater than max_{in} because RIO drops all the incoming packets when the instantaneous queue is equal to max_{in}.

Once $\pi(i)$ has been obtained RTT, p_{in} and p_{out} can be evaluated as

$$RTT = R_0 + q/C = R_0 + \frac{1}{C} \sum_{i=0}^{max_{in}} i\pi(i) \tag{15}$$

$$p_{in} = \sum_{i=0}^{max_{in}} p_{in}(i)\pi(i) \tag{16}$$

$$p_{out} = \sum_{i=0}^{max_{out}} p_{out}(i)\pi(i) \tag{17}$$

where R_0 is the sum of propagation and transmission delays.

We have followed such approach, but results are unsatisfactory. The physical explanation appears from figures 5(b) and 5(a), which show the empirical distribution coming from simulations and the queue distribution predicted by the model given the same average load, for three different configurations. The RIO settings in the legend are given in the form $(min_{out}, max_{out}, Pmax_{out}) - (min_{in}, max_{in}, Pmax_{in})$. According to the model the queue should exhibit a spread distribution, with high probability for low queue values (in particular the probability density is strictly decreasing if $T < C$), while the empirical distribution looks like a gaussian one: the dynamic adaptive throughput of the TCP sources, which increase their throughput when RTT decreases and vice versa, appear to be able to create a sort of "constant bias".

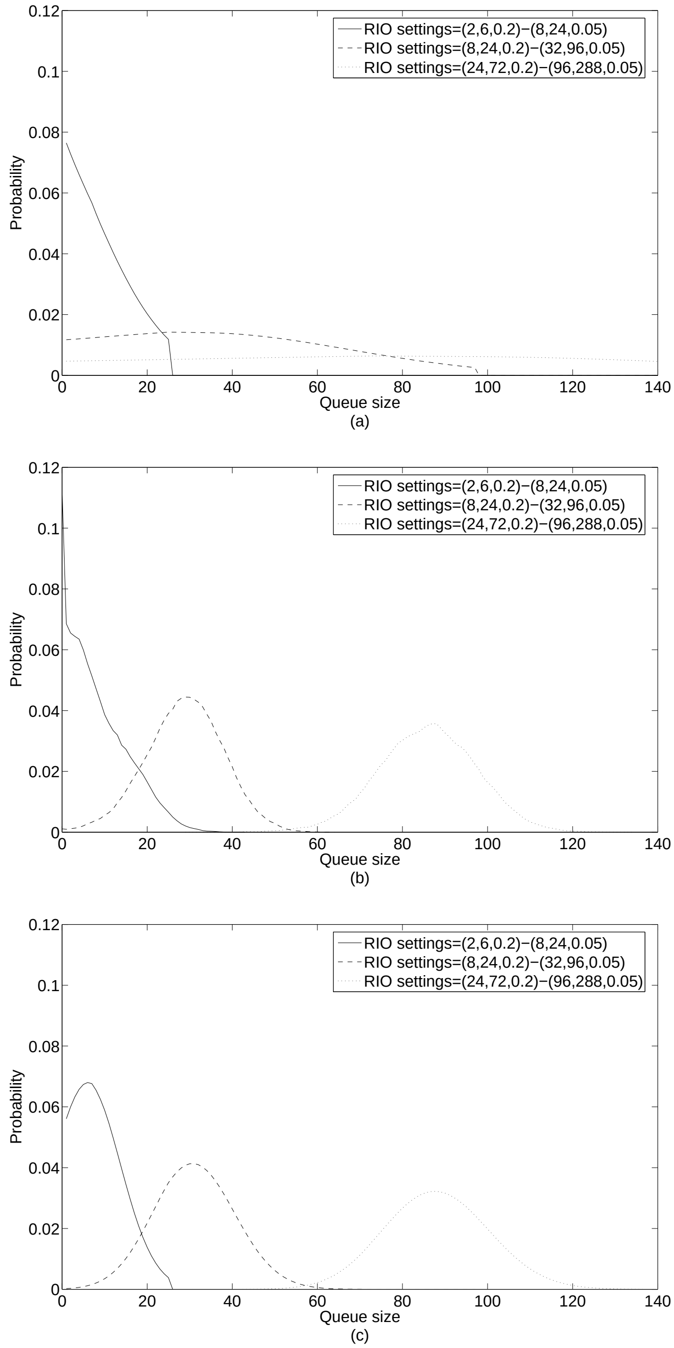

Fig. 5. (*a*) Queue distribution predicted by the model with uniform arrivals. (*b*) Queue distribution obtained by simulations. (*c*) Queue distribution predicted by the model with state dependant arrivals.

In order to capture this behavior, we have modified the model in [30], by introducing arrival dependence from the network status. The input to the sub-model is

$$F(N) = T * RTT = \frac{N}{\frac{3b-2}{6} + \sqrt{\frac{2bN}{3} + \left(\frac{2+3b}{6}\right)^2} + 1} \tag{18}$$

and the arrival rate when the there are j packets in the queue is:

$$T(j) = \frac{F}{R_0 + \frac{j}{C}}$$

Now the stationary distribution can be evaluated as:

$$\pi(i) = \pi(0) \prod_{j=0}^{i-1} \frac{T(j)}{C} (1 - p(j)), i = 1, 2, ...max_{in}$$

Fig. 5(c) shows the queue distribution evaluated by the new model. The similarity with figure 5(c) is impressive, the only difference is for the first configuration $((2, 6, 0.2) - (8, 24, 0.05))$, as regards low queue occupancy. The peak for $q = 0$ is probably due to timeouts, which are more common with low RIO settings, and make TCP throughput less uniform and hence the markovian arrival assumption less accurate.

4.5 About the solutions of the system

Summarizing, our model has 8 variables (N,A,T,L,RTT,p_{in},p_{out},F) and 8 equations (10), (11), (12), (13), (15) (16), (17) and (18). In this section we afford existence and uniqueness of solutions for this system. We are going to reduce the system to a simpler one with two variables (F and q).

First, let us note that F can be expressed as an increasing function of A by equations (18), (12) and (13). Besides it can be proven that $\pi(i+1)/\pi(i)$ increases with F and A alike, being $p_{in}(i) < p_{out}(i)$. Hence q, p_{in} and p_{out} are continuous increasing function of F and A and by the relation between F and A, we can express them as increasing function of F (e.g. from $q = q(A, F)$ and $A = A(F)$, $q = q(A(F), F) = q(F)$). Being these function invertible, we can express p_{in} and p_{out} as (increasing) functions of q.

Besides, the following results hold:

$$q(F = 0) = 0$$

$$\lim_{F \to +\infty} q(F) = max_{in}$$

As regards A, from equations (10) and (13) ($A = L$), we obtain that A is the solution (if any) of the following equation

$$A + 1 = \frac{s_{in}^{A+1} - 1}{s_{in} - 1} \frac{1}{1 - s_{in}^{A} s_{out}} \tag{19}$$

The right member of equation (19) (i.e. N) is an increasing function of A, s_{in} and s_{out} as it appears immediately from the same definition of N, taking into account that $p_{in} < p_{out}$. Being $1/p_{out} \leq N \leq 1/p_{in}$ the curve represented by the right member always intersects the line represented by the left member only in one point (because N increases with A). Hence equation (19) admits one and only one solution A and this solution increases with s_{in} and s_{out} (because N increases with s_{in} and s_{out}). From the relation between p_{in}, p_{out} and q it follows that A is a decreasing function of q. Besides when q converges to zero p_{out} converges to zero and $A > 1/p_{out}$ diverges, i.e.

$$\lim_{q \to 0} A(q) = +\infty$$

Let us focus on the expression of F (18). From the relation between F and A and the relations established above, it appears that F is a decreasing function of q and:

$$\lim_{q \to 0} F(q) = +\infty$$

$$\lim_{q \to +\infty} F(q) = 0$$

From the previous considerations and hypotheses it follows that the simplified system in F and q admits one and only one solution, as it is qualitatively shown in figure 6. Being all the function monotone, the original system admits only one solution.

It is possible to set up an iterative procedure to find numerically this solution, and this is just what we did using MATLAB.

5 Model validation

To validate our model we considered the network topology showed in Fig. 7, consisting of a single bottleneck link with capacity equal to 6Mbps. Considering both the transmission and the propagation delay of packets and acks in the network, the average Round Trip Time is $R_0 \cong 138$ms. The IP packet size is chosen to be 1500 Bytes, for a bottleneck link capacity of $c = 500$packets/s.

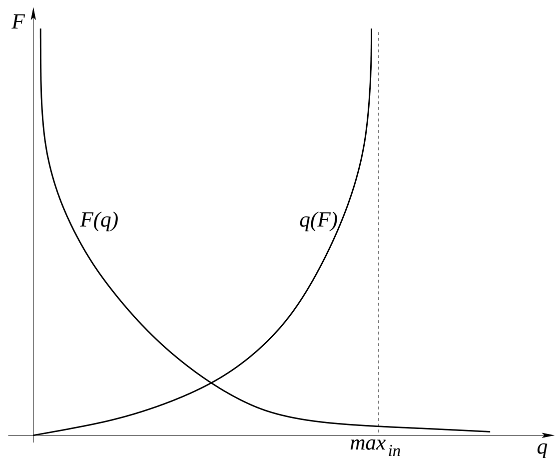

Fig. 6. Existence and uniqueness of the solution.

Table 1
Model vs Simulation with 10 flows

RIO	T (pkt/s)		G (pkt/s)		q (pkt)		A (pkt)	
	mod	sim	mod	sim	mod	sim	mod	sim
(2,6)(8,24)	478.02	474.87	467.57	467.82	7.86	8.54	51.51	57.35
(3,9)(12,36)	494.22	492.59	485.02	486.29	12.49	12.13	60,42	67.70
(4,12)(16,48)	499.25	499.76	490.94	494.27	16.72	14.90	67.17	79.50
(6,18)(24,72)	501.29	504.15	494.25	499.27	24.52	22.17	78.83	90.76
(8,24)(32,96)	501.10	504.39	495.02	499.88	32.04	29.24	90.33	97.96
(12,36)(48,144)	500.39	503.83	495.68	499.98	46.63	44.01	114.59	113.96
(16,48)(64,192)	499.93	503.23	496.15	499.99	60.89	58.57	141.06	132.07
(24,72)(96,288)	499.44	502.37	496.84	500.00	89.03	86.67	201.26	169.65
mean error (%)	-0.28		-0.65		5.00		-3.92	
max error (%)	-0.68		-1.00		12.22		18.63	

RIO	$Pdrop$ (%)		$Pdrop_{in}$ (%)		$Pdrop_{out}$ (%)		
	mod	sim	mod	sim	mod	sim	
(2,6)(8,24)	2.185	1.455	1.009	0.194	62.791	75.099	
(3,9)(12,36)	1.860	1.288	0.731	0.131	70.108	80.599	
(4,12)(16,48)	1.665	1.135	0.580	0.090	74.590	84.961	
(6,18)(24,72)	1.405	1.017	0.408	0.051	79.978	89.252	
(8,24)(32,96)	1.214	0.939	0.303	0.037	83.494	89.453	
(12,36)(48,144)	0.942	0.792	0.184	0.017	87.805	89.628	
(16,48)(64,192)	0.757	0.670	0.121	0.012	90.452	88.069	
(24,72)(96,288)	0.522	0.506	0.060	0.011	93.531	85.179	
mean error (%)	30.52		648.84		-6.02		
mean error (%)	50.24		991.51		-16.39		

As regards RIO configurations we considered *non overlapping* the ones in which $max_{out} < min_{in}$, more precisely we choose $max_{out} = 3min_{out}$, $max_{in} = 3min_{in}$ and $min_{in} = 4max_{out}$. In previous performance evaluation this kind of settings showed better results in comparison with a *overlapping* RIO configuration in which, $max_{out} >= min_{in}$. We tested seven different configurations, varying min_{out} from 2 up to 24, and for each configuration we gathered statistics from 10 trials of 1000 seconds each. We chose $P_{max_{out}} = 0.2$ and $P_{max_{in}} = 0.05$. We ran our simulations using ns v2.1b9a, with the Reno ver-

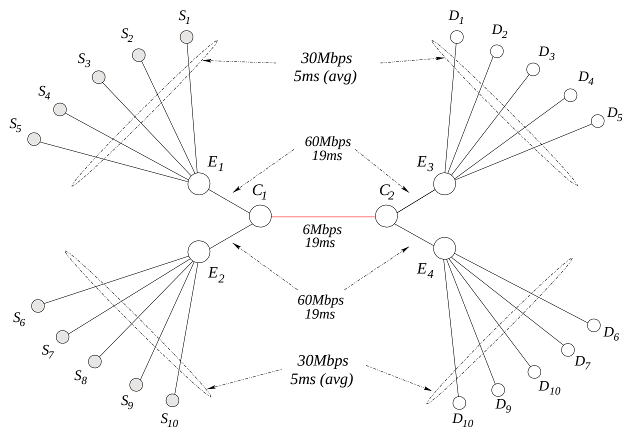

Fig. 7. Network topology.

sion of TCP.

Table 1 compares model predictions with simulation results when the number of flows is equal to $n = 10$, as regards throughput (T), goodput [5] (G), queue occupancy (q), the dropping probability for the generic packet, for IN packets and for OUT packets (respectively $Pdrop$,$Pdrop_{in}$, $Pdrop_{out}$), and the average length of IN packets bursts. The average mean error over the different settings and the maximum error are shown in the last two rows. The model appears to be able to predict with significant accuracy throughput, goodput and queue occupancy, which are the most relevant performance indexes when we consider TCP long lived performance flows. On the contrary dropping probability estimates are very inaccurate, in particular as regards $Pdrop_{in}$. We think the reason is that the model neglects the effect of filtering on dropping probability calculation from RIO routers. In fact some preliminary results which take into account filtering seem to suggest that filtering: i) can be neglected in order to evaluate the dynamic of the instantaneous queue, ii) it is significant for the evaluation of the dropping probabilities. In particular probabilities estimates look better. At the moment we have introduced the effect of filtering by considering a two dimensional Markov chain where the status is the pair of instantaneous queue and filtered queue (whose values have been quantized). This approach is particularly heavy from the computational point of view, for this reason, at the moment, we have not adopted it.

The goodput/delay tradeoff is presented in Fig. 8, where each point corresponds to a different threshold setting.

We evaluated also the model with the same network topology with a different number of flows ($n = 6$,$n = 20$). The differences between model predictions and simulation results are similar to those observed for $n = 10$ flows. The relative errors for these two scenarios are shown in Table 2.

[5] The goodput is estimated as $G = T(1 - Pdrop)$.

Table 2

Model vs Simulation with 6 and 20 flows

$n = 6$	T (pkt/s)	G (pkt/s)	q (pkt)	$Pdrop$ (%)	$Pdrop_{in}$ (%)	$Pdrop_{out}$ (%)	A (pkt)
mean error (%)	-0.65	-0.74	-1.01	12.12	649.54	-1.60	25.98
max error (%)	-0.86	-0.95	-22.46	23.71	1141.30	28.58	58.46
$n = 20$	T (pkt/s)	G (pkt/s)	q (pkt)	$Pdrop$ (%)	$Pdrop_{in}$ (%)	$Pdrop_{out}$ (%)	A (pkt)
mean error (%)	1.25	-0.27	2.17	83.70	598.05	-3.23	-36.48
max error (%)	4.43	1.58	-18.42	120.97	729.45	-13.51	-46.13

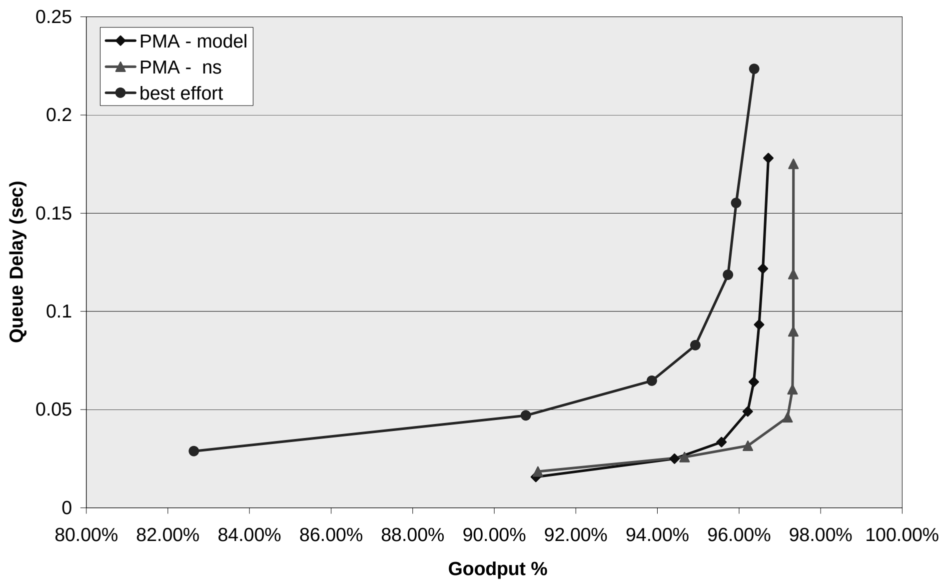

Fig. 8. Queue Delay vs Goodput for the PMA model and the corresponding $ns2$ simulations, together with simulation results of a standard best effort service (without marking)

6 A model application

Here we want to show a possible application of our model. In particular we want to evaluate a new variant of the algorithm where a higher number of packet is marked OUT. Intuitively this new version should be able to react more quickly to traffic changes, by allowing more *probes*. With reference to the algorithm flowchart in Fig. 1, in the new marking scheme a packet is marked OUT every time C_{IN} exceeds $A_{IN}/2$. From the modeling point of view we have only to change the marking law $m()$ (see section 4.3) as it follows:

$$A = \frac{1}{2}L$$

The model predictions and the simulation results are shown in Fig. 9 as performance frontiers. The same RIO configurations have been considered for both the original algorithm and the variant, with min_{out} ranging from 2 to 24 while the other parameters have been chosen according to section 5. It appears that the model is able to capture the main change: the curve of the new variant

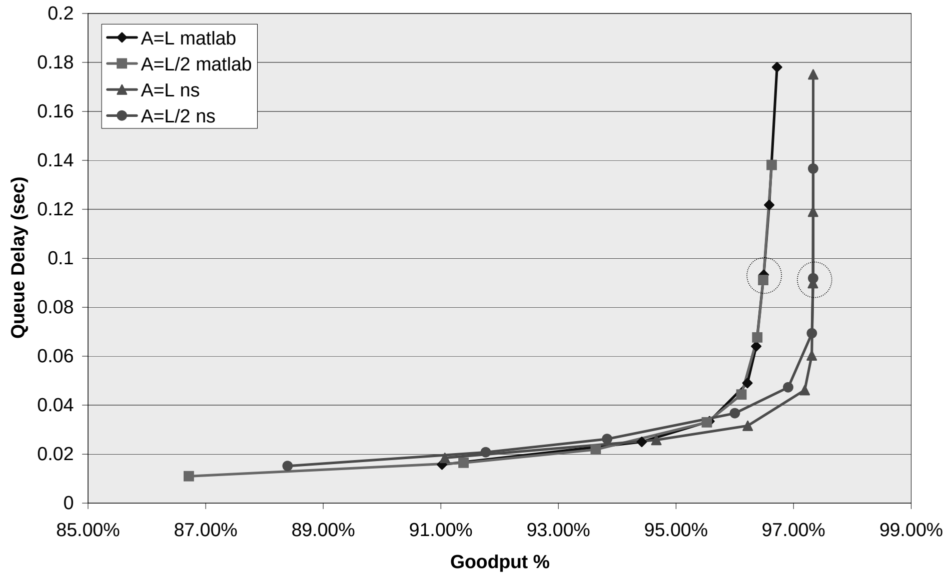

Fig. 9. A study of a variant of the PMA

is shifted towards lower utilization because of higher sensibility to congestion, but its shape is almost unchanged. In order to stress this point, two pairs of points are circled in the figure: they correspond to the $min_{out} = 12$ configuration for the original algorithm and $min_{out} = 16$ for the new variant. It appears that the two algorithms are able to achieve almost the same performance with different configurations.

At the same time simulation results show another effect that the model is not able to catch: the new variant exhibits also a higher queuing delay for average utilization. We think the reason is a higher traffic variability (the dropping probability of OUT packets decreases from about 80% to about 50%) which produces larger queues. This effect is not addressed by the model because the network sub-model takes into account mainly the average throughput assuming the same markovian arrival process independently from the specific marking strategy.

7 Conclusions and further research issues

In this paper we have presented an analytical model for our adaptive packet marking scheme proposed in previous works. From preliminary simulative results, model predictions about throughput and average queue occupancy appear to be quite accurate. We have also shown that the model can be employed to evaluate variants of the original marking algorithm. We are going to extend simulative evaluation and to employ such model to study possible variants of the marking algorithm and to establish optimal RIO settings.

Besides our network sub-model exhibits some novelty and seems to be more suited than traditional M/M/1/K proposals to capture the behavior of long lived TCP flows. We are going to study it deeply and to evaluate it in a simpler best effort scenario. We want to evaluate the effect of filtering, which is usually neglected in M/M/1/K models, but it appears to have a deep impact on the performance.

References

[1] D. D. Clark, W. Fang, Explicit Allocation of Best Effort packet delivery service, IEEE Transactions on Networking 6 (4) (1998) pp. 362–373.

[2] J. Ibanez, K. Nichols, Preliminary simulation evaluation of an assured service, IETF draft (1998) .

[3] J. Heinanen, R. Guerin, A Single Rate Three Color Marker, Request For Comments 2697 (1999) .

[4] J. Heinanen, R. Guerin, A Two Rate Three Color Marker, Request For Comments 2698 (1999) .

[5] W. Fang, N. Seddigh, B. Nandy, A time sliding window three colour marker (tswtcm), Request For Comments 2859 (2000) .

[6] N. Seddigh, B. Nandy, P. Piedu, Bandwith assurance issues for TCP flows in a Differentiated Services network, Proc. of IEEE Globecom (1999) pp. 1792–1798.

[7] S. Sahu, D. Towsley, J. Kurose, Quantitative study of Differentiated Services for the Internet, Proc. of IEEE Globecom (1999) pp. 1808–1817.

[8] J. Harju, Y. Koucheryavy, J. Laine, S. Saaristo, K. Kilkki, J. Ruutu, H. Waris, J. Forsten, J. Oinonen, Performance Measurements and Analysis of TCP Flows in a Differentiated Services WAN, Proc. of the 25th Annual IEEE Conference on Local Computer Networks (2000) .

[9] S. Sahu, P. Nain, D. Towsley, C. Diot, V. Firoiu, On achievable Service Differentiation with Token Bucket Marking for TCP, Proc. of ACM SIGMETRICS'00 (2000) .

[10] 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 7 (5) (1999) pp. 685–697.

[11] Y. Chait, C. Hollot, V. Misra, D. Towsley, H. Zhang, J. C. Lui, Providing Throughput Differentiation for TCP Flows Using Adaptive Two-Color Marking and Two-Level AQM, Proc. of IEEE Infocom (2002) .

[12] I. Yeom, A. L. N. Reddy, Adaptive Marking for Aggregated Flows, Proc. of IEEE Globecom (2001) .

[13] P. Giacomazzi, L. Musumeci, G. Verticale, Transport of TCP/IP Traffic over Assured Forwarding IP-Differentiated Services, IEEE Network (5) (2003) pp. 18–28.

[14] M. El-Gendy, K. Shin, Equation-Based Packet Marking for Assured Forwarding Services, Proc. of IEEE Infocom (2002) .

[15] J. H. Lee, C. K. Jeong, Improvement of fairness between assured service TCP users in a differentiated service network, Proc. of Joint 4th IEEE International Conference ATM (ICATM 2001) and High Speed Intelligent Internet Symposium (2001) .

[16] H. Wu, K. Long, S. Cheng, J. Ma, Y. Le, TCP Friendly Fairness in Differentiated Services IP Networks, Proc. of 9th IEEE International Conference on Networks (ICON) (2001) .

[17] M. Mellia, I. Stoica, H. Zhang, Packet Marking for web traffic in networks with RIO routers, Proc of Globecom (2001) .

[18] F. Azeem, A. Rao, S. Kalyanaraman, A TCP-Friendly traffic Marker for IP Differentiated Services, Proc. of IwQoS (2000) .

[19] G. L. Monaco, F. Azeem, S. Kalyanaraman, Y.Xia, TCP-Friendly Marking for scalable Best-Effort services on the Internet, Computer Communication Review (CCR) 31 (5) (2001) .

[20] G. Neglia, G. Bianchi, F. Saitta, D. Lombardo, Adaptive Low Priority Packet Marking for Better TCP Performance, Net-Con (2002) .

[21] J. Padhye, V. Firoiu, D. Towsley, J. Kurose, Modeling TCP throughput: A simple Model and its empirical validation, Proc. of ACM SIGCOMM (1998) .

[22] G. Neglia, G. Bianchi, M. Sottile, Performance Evaluation of a new Adaptive Packet Marking Scheme for TCP over DiffServ Networks, Proc. of Globecom (2003) .

[23] T. Bu, D. Towsley, Fixed Point Approximation for TCP behavior in an AQM Network, Proc. of ACM SIGMETRICS (2001) .

[24] M. Meo, M. Garetto, M. A. Marsan, R. L. Cigno, On the Use of Fixed Point Approximations to Study Reliable Protocols over Congested Links, Proc. of Globecom (2003) .

[25] F. P. Kelly, Blocking probabilities in large circuit-switched networks, Advances in Applied Probability 18 (1986) pp. 473–505.

[26] R. Gibbens, S. Sargood, C. V. Eijl, F. Kelly, H. Azmoodeh, R. Macfadyen, N. Macfadyen, Fixed-Point Models for the End-to-End Performance Analysis of IP Networks, Proc. of 13th ITC Specialis Seminar: IP Traffic Measurement, Modeling and Management (2000) .

[27] M. Roughan, A. Erramilli, D. Veitch, Network performance for tcp networks, part i: Persistent sources, Proc. of International Teletraffic Congress (2001) .

[28] U. Ayesta, K. Avrachenkov, E. Altman, C. Barakat, P. Dube, Simulation Analysis and Fixed Point Approach for Multiplexed TCP flows, INRIA Technical Report RR-4749 (2003) .

[29] C. Casetti, M. Meo, A New Approach to Model the Stationary Behavior of TCP Connections, Proc. of IEEE Infocom (2000) .

[30] N. M. Malouch, Z. Liu, Performance Analysis of TCP with RIO routers, Proc. of IEEE Globecom (2002) .

[31] A. Misra, T. Ott, The Window Distribution of Idealized TCP Congestion Avoidance with Variable Packet Loss, Proc. of IEEE Infocom (1999) .

[32] G. Neglia, Ingress Traffic Control in Differentiated Services IP Networks, Ph.D. thesis, Università degli studi di Palermo, Dipartimento di Ingegneria Elettrica (2004).

[33] V. Istratescu, Fixed point theory, Reidel, Dordrecht, Holland, 1981.

[34] G. Debreu, Theory of Value: An axiomatic analysis of economic equilibrium, Wiley, New York, NY, USA, 1959.

[35] V. Firoiu, M. Borden, A study of Active Queue Management for Congestion Control, Proc. of IEEE Infocom (2000) .

[36] F. Baccelli, D. Hong, Z. Liu, Fixed Point Methods for the Simulation of the Sharing of a Local Loop by a Large Number of Interacting TCP Connections, INRIA Technical Report RR-4154 (2001) .

[37] F. Baccelli, D. R. McDonald, J. Reynier, A Mean-Field Model for Multiple TCP Connections through a Buffer Implementing RED, INRIA Technical Report RR-4449 (2002) .

[38] D. Bertsekas, R. Gallagher, Data Networks, Prentice-Hall, Engelwood Cliffs, NY, USA, 1992.

[39] K. W. Ross, Multiservice loss networks for broadband telecommunication networks, Springer Verlag, Secaucus, NJ, USA, 1995.

[40] V. Firoiu, I. Yeom, X. Zhang, A Framework for Practical Performance Evaluation and Traffic Engineering in IP Networks, Proc. of IEEE International Conference on Telecommunications (2001) .

[41] E. Altman, K. Avrachenkov, C. Barakat, TCP Network Calculus: The case of large delay-bandwidth product, Proc. of IEEE Infocom (2002) .

[42] V. Misra, W. Gong, D. Towsley, A Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED, Proc. of ACM SIGCOMM.

[43] G. Neglia, G. Bianchi, V. Falletta, An Analytical Model of a new Packet Marking Algorithm for TCP flows: preliminary insights, Proc. of ISCCSP (2004) .

[44] J. Cao, W. S. Cleveland, D. Lin, , D. X. Sun, Nonlinear Estimation and Classification, Chapter: Internet traffic tends toward poisson and independent as the load increases, Springer, 2002.

[45] M. May, J.-C. Bolot, A. Jean-Marie, C. Diot, Simple performance models of tagging schemes for service differentiation in the Internet, Proc. of the IEEE Infocom (1999) .

[46] M. Garetto, R. L. Cigno, M. Meo, M. A. Marsan, Closed Queueing Network Models of Interacting Long-Lived TCP flows, IEEE/ACM Transactions on Networking 12 (2) (2004) pp. 300–311.

[47] R. Stankiewicz, A. Jajszczyk, Modeling of TCP behavior in a DiffServ Network supporting Assured Forwarding PHB, Proc. of ICC (2004) .