Performance Evaluation of AQM Techniques with Heterogeneous Traffic

Andrea Marin
and Sabina Rossi
Università Ca' Foscari Venezia
Venezia, Italy
Email: {marin,srossi}@dais.unive.it

Armir Bujari
and Claudio Palazzi
Università degli Studi di Padova
Padova, Italy
Email: {abujari,cpalazzi}@math.unipd.it

Abstract—**Active Queue Management (AQM) techniques have been proposed to support scenarios with many connections sharing the same bottleneck. The basic idea is that a smart management of the bottleneck queue can avoid the saturation of the link and ensure a smoother use of the available bandwidth. This is generally achieved by exploiting the flux control mechanism of TCP and its behavior in case of packet losses or other explicit notifications. In this paper we consider classic and innovative AQM techniques and analyse their performance under different scenarios through the use of mean field models.**

I. Introduction

In modern computer networks the problem of defining protocols that allow several devices to share the same gateway gains in importance. Different end-user applications may have different requirements in terms of bandwidth and responsiveness. In some cases, such as online gaming, a failure to meet some Quality of Service (QoS) requirements makes the application usability not acceptable. We consider the scenario in which a set of devices compete for the access to the same gateway and we study the performances of some schemes which have been proposed to manage the resource allocation, i.e., the bandwidth assignment. In specifics, several mechanisms have been proposed to cope with the problem of bandwidth allocation in presence of TCP and UDP transport protocols. In general, it is widely accepted (see e.g., [1], [2]) that TCP is adopted for reliable data transfer such as file transfer or web browsing, while UDP is used for transferring multimedia resources and by applications with strict QoS requirements. A typical example of such applications is represented by online gaming whose responsiveness requirements are often even more important than data transfer reliability.

We consider two types of schemes for the gateway bandwidth assignment: those based on exploiting the TCP congestion control mechanism and those which make use of the TCP flux control mechanism. Two examples of protocols belonging to the former family are the well-known *Random Early Detection* (RED) [3] and the *Explicit Congestion Notification* (ECN) [4], while the Smart Access Point with Limited Advertised Window (SAP-LAW) [5] belongs to the latter family.

The SAP-LAW control scheme has been proposed in [5], [6] with the aim of solving the access point bandwidth allocation problem in the context of the integrated home server. In this paper, we propose some stochastic models with the aim of studying the behaviour of the above mentioned protocols (ECN/RED and SAP-LAW) for bandwidth sharing in presence of both TCP and UDP traffic. The coexistence of these two transport protocols poses some problems since UDP-based flows do not have any mechanism to control their resource demands and hence may result aggressive toward the active TCP connections. On the other hand, the presence of TCP flows may cause an important deterioration on the responsiveness of the UDP-based applications [5]. These problems can be only partially mitigated by augmenting the available bandwidth since the TCP congestion control mechanism tends to have a high utilisation of the resources thanks to its continuous built-in probing mechanism demanding for higher data rates. As a consequence, the packets are enqueued at the bottleneck causing an undesired delay for the UDP packets. Different versions of TCP have been proposed over the years tackling this problem ([7], [8]), while the simplicity of the UDP does not allow for effective solutions based on its modifications. The performance of TCP and ECN has been widely investigated in the recent literature (see, e.g., [7], [9]–[11]) especially by considering a single TCP connection under different working scenarios. Conversely, in [12] the authors propose a mean field model to study the global system performance for ECN/RED under TCP traffic. We consider a similar setting for the ECN and SAP-LAW and we consider the problems introduced by the coexistence of TCP and UPD flows. The theoretical foundation of our findings are based on the results presented in [13] where a general framework for studying mean field models has been introduced.

The novelties proposed in this paper can be summarised as follows:

- We extend the model considered in [12] by taking into account the effects of UDP based traffic.
- We introduce a new model for the SAP-LAW congestion control mechanism.
- Differently from [5], [12], [13], the proposed models do not require the assumption about the greedy behaviour of the TCP flows. In other words, we consider the possibility that a TCP connection terminates its transmission and a new one is started.

978-1-4673-9292-1/16/$31.00 ©2016 IEEE

- Finally, we compare the ECN/RED and the SAP-LAW schemes based on the numerical evaluation of the proposed models.

The paper is structured as follows. Section II explains the ECN/RED and the SAP-LAW congestion mechanisms and Section III gives some background about the mean field theory. In Section IV we introduce our models and give the iterative schemes for the computation of the performance indices. In Section V we compare the performance indices of the two congestion mechanisms under different scenarios. Finally, Section VI concludes the paper.

II. Congestion control

In this section we review the ECN/RED and the SAP-LAW congestion control mechanisms. ECN [3] is used in conjunction with RED and aims at avoiding packet losses due to the congestion experienced in the bottleneck. In fact, packets are marked (and not dropped) when the bottleneck's queue has a certain size. RED works with a minimum and a maximum threshold for the queue: if the enqueued packets are less then the minimum threshold then the packets are not marked, if it is between the minimum and the maximum threshold, the packets are marked with a probability that depends on the queue size. In the case of a number of enqueued packets larger than the maximum, all the new arrivals are either marked or dropped. In this paper we approximate this behaviour as proposed in [14].

SAP-LAW [5] aims at reducing the problems caused by the coexistence of TCP and UDP flows in the same gateway. SAP-LAW fonds a trade-off between throughput and waiting time at the gateway by dynamically modifying the TCP connection sending rates. In many practical cases, this avoids the over utilization of the gateway's buffer while maintaining a reasonably high throughput. The main idea consists in determining the upper bound for the TCP sending rate by taking into account the bandwidth consumed by the UDP traffic. The formula introduced in [5] is:

$$maxTCP_{\text{traffic}}(t) = \frac{(C - UDP_{\text{traffic}}(t))}{\#TCP_{\text{flows}}(t)} ,$$

where $UDP_{\text{traffic}}(t)$ is the bandwidth occupied by the real time applications at time t, C is the capacity of the gateway, and $\#TCP_{\text{flows}}(t)$ is the number of TCP connections at time t.

III. Theoretical background

Following the lines of [13], we consider a population of N objects whose states are (c, i), with $c \in \Gamma$ the object class, and $i \in \{1, ..., S_c\}$ its actual state. These objects interact with the environment and can perform a state transition. Let $X_n^N(t)$ be the state of object n at time t, and let $K^N_{(c,i),(c,j)}(t)$ be the transition probability for a class c object from state i to state j at t. The object state transitions depend on the current state of the objects and on the memory $\vec{R}^N(t) \in \mathbb{R}^P$ where P is the dimension of the vector. $\vec{M}^N(t)$ denotes the *occupancy measure*, i.e., the vector whose component $M^N_{c,i}(t)$ represents the proportion of objects of class c that are in state i at time t, i.e.,

$$M^N_{c,i}(t) = \frac{1}{N} \sum\nolimits_{n=1}^{N} 1_{\{X_n^N(t)=(c,i)\}} ,$$

where $1_{x=(c,i)} = 1$ when $x = (c, i)$ and 0 otherwise. If we want to consider the single class of objects, the portion of class c objects in state i at time t is $\frac{1}{p_c} M^N_{c,i}(t)$, where $p_c = \sum_{i=1}^{S_c} M_{c,i}(t)$ at some time slot t. Function g is a deterministic continuous function $g : \mathbb{R}^d \times \mathbb{P}(\varepsilon) \mapsto \mathbb{R}$ such that, for $t \geq 0$:

$$\vec{R}^N(t+1) = g(\vec{R}^N(t), \vec{M}^N(t+1)) . \quad (1)$$

The role of function g is to govern the evolution of the model's memory. Hence, $\forall n$ we have:

$$\mathbb{P}\{X_n^N(t+1) = (c, j) \mid \vec{R}^N(t) = \vec{r}, X_n^N(t) = (c, i)\} = K^N_{c,i,c,j}(\vec{r}) .$$

For each class c, the $S_c \times S_c$ matrix $K^N_c(\vec{r})$ is the transition matrix for an individual object of class c, and may depend on the population N and on the value $\vec{r}$ of the global memory. $K^N_{c,i,c,j}$ is the transition probability from state i to state j, therefore we assume $K^N_{c,ic,j}(\vec{r}) \geq 0$ and $\sum_{j=1}^{S_c} K^N_{c,ic,j}(\vec{r}) = 1$ for all $1 \leq i \leq S_c$. We assume that:

Assumption 1: $\forall c \in \Gamma, \forall i, j \in \varepsilon_c$, for $N \to \infty$, $K^N_{c,i,c,j}(\vec{r})$ converges uniformly in $\vec{r}$ to some $K_{c,i,c,j}(\vec{r})$, which is a continuous function of $\vec{r}$.

Theorem 1 (Mean field [13]): Assume that the initial occupancy measure $\vec{M}^N(0)$ and memory $\vec{R}^N(0)$ converge almost surely to deterministic limits $\vec{\mu}(0)$ and $\vec{\rho}(0)$ and for $t \geq 0$:

$$\begin{aligned} \vec{\mu}(t+1) &= \vec{\mu}(t) K(\vec{\rho}(t)) && (2) \\ \vec{\rho}(t+1) &= g(\vec{\rho}(t), \vec{\mu}(t+1)) && (3) \end{aligned}$$

Then for any fixed time t, almost surely:

$$\lim_{N\to\infty} = \vec{\mu}(t) \text{ and } \lim_{N\to\infty} \overline{R}^N(t) = \vec{\rho}(t) .$$

IV. Mean field models for the ECN/RED and SAP-LAW under TCP and UDP traffic

In this section we propose different models for the ECN/RED and SAP-LAW congestion avoidance mechanisms. Differently from other approaches previously addressed by the literature, we consider the presence of UDP traffic (either bursty or smooth) and the cases if which TCP are not greedy, i.e., they terminates their transmissions and a new one is open.

A. Modelling the ECN/RED

In this section we propose a mean field model for TCP and UDP connections sharing the same gateway. The time slot is a round trip time as in [12], [13]. We assume that once the connection is established it has to send an amount of packets which is geometrically distributed with parameter $w \in (0, 1)$, i.e, the probability of sending exactly ξ packets is $w(1-w)^{\xi-1}$, with an expected value of $1/w$. Let $L_f(t)$ for $1 \leq f \leq N_p$ and $t \geq 0$ be a set of independent and identically distributed geometric random variables with parameter $w \in (0, 1)$ that represent the number of packets that TCP connection i has still to send at time t. The event of closure of TCP connection f occurs at time t_0 if the number of packets that it sends at t_0 is equal to $L_f(t_0)$. Notice that due to the memoryless property of the geometric random variable, the distribution of the number of packets sent by a TCP connection from its opening to its closing epoch is still geometric.

According to the methodology described in Section III, we annotate each object with a class $c \in \{p, u\}$ where p denotes the TCP connections and u the UDP ones. The state (i, s) of each TCP object consists of an integer $i \in [1, I_p]$ that denotes the connection state and an integer $s \in [1, s_p(I_p) + 1]$ that denotes the number of packets that have still to be sent. At state (i, s), the maximum number of packets sent in a time slot is $s_p(i) \in \mathbb{N}$ and, without loss of generality we assume $i' > i'' \implies s_p(i') > s_p(i'')$.

Although UDP transmissions are stateless, we model them by using a probabilistic automaton whose transition probabilities are independent of the rest of the model (in particular of the queue length). The set of states associated with a process using a UDP transmission is denoted by $\mathcal{S}_u = \{1, \dots, I_u\}$ and the sending rate at state $i \in \mathcal{S}_u$ is $s_u(i)$. Notice that the flexibility of this modelling choice is very high: it allows us to represent synchronised or partially synchronised UDP transmissions (such as those occurring during online gaming), the autocorrelation and the periodicity in the transmission bursts, as well as a simple smooth transmission of packets. To avoid trivialities we assume $I_p > 1$ and $I_u \geq 1$.

Let $N \in \mathbb{N}$ be the number of objects and $C \in \mathbb{R}$ the bottleneck capacity for every connection. We define N_p, N_u as the number of TCP and UDP connections, respectively. Let $p_{\mathrm{p}}, p_{\mathrm{u}}$ be the proportion of TCP and UDP agents, respectively.

The memory of the system is a pair of non-negative reals $\vec{R}^N(t) = (R_c(t), R_p(t))$ that denotes the normalised queue length at the gateway at epoch t and $t-1$, respectively. The update rule for the memory is:

$$
\begin{aligned}
&R_c^N(t+1) = \max\left(R_c^N(t) + S_p^N(t+1) + S_u^N(t+1) - C, 0\right) \\
&R_p^N(t+1) = R_c^N(t)
\end{aligned}
$$

where:

$$
S_p^N(t+1) = \sum_{i=1}^{I_p} \sum_{s=1}^{s_p(I_p)+1} M_{p,i,s}(t+1) \min(s_p(i), s) \,,
$$

where $M_{p,i,s}$ is the occupancy vector defined in Section III. We can approximate the behaviour of the ECN/RED mechanisms as proposed in [12]: as soon as a packet arrives at the bottleneck it is marked for being discarded or sent. The marking of each packet is independent of the others arriving during the same time slot, and cannot be changed. The marking of packets at time slot t depends only on the queue length seen by the packets immediately before their arrival, i.e., on the memory at time t via function $q(\vec{R}^N(t))$. According to [3], [4], [14] we define function q as follows:

$$
q(R_p^N(t)) = 1 - \exp(-\gamma R_p^N(t)) \,, \tag{4}
$$

i.e., the probability of discarding a packet grows exponentially with the total queue length. Parameter γ is a positive real number whose value will be specified in the following sections. Finally, the transition probabilities for TCP and UDP objects are:

$$
\begin{aligned}
&\tau(s') = w(1-w)^{s'-1} 1_{s' \leq s_p(I_p)} \\
&\qquad + (1-w)^{s_p(I_p)} 1_{s' = S_p(I_p)+1} &(5)\\
&K_{(p,i,s),(p,i+1,s')}(\vec{r}) = (1 - q(r_p))^{s_p(i)} \\
&\qquad \cdot 1_{i<I_p, s>s_p(i)} \tau(s') \\
&K_{(p,i,s),(p,1,s')}(\vec{r}) = 1_{s \leq s_p(i)} \tau(s') &(6)\\
&K_{(p,I_p,s),(p,I_p,s')} = (1 - q(r_p))^{s_p(I_p)} 1_{s>s_p(i)} \tau(s') \\
&K_{(p,i,s),(p,d(i),s')} = \left(1 - (1 - q(r_p))^{s_p(i)}\right) 1_{s>s_p(i)} \tau(s') \\
&K_{(u,i),(u,j)}(\vec{r}) = \kappa_{(u,i),(u,j)}, &1 \leq i, j \leq I_u
\end{aligned}
$$

where $\kappa_{(u,i),(u,j)} \in [0,1]$ and for all $i \in [1, I_u]$ $\sum_{j=1}^{I_u} \kappa_{(u,i),(u,j)} = 1$. Note that Equation (6) is the key point to decide when a TCP connection finishes its transmission and a new one is created. Function τ is the density function of a truncated geometric random variable, where the probability mass of outcomes greater that $s_p(I_p)$ is concentrated in the last state, $s_p(I_p) + 1$. Function $d(i)$ is used to model the destination state in case of a marked packet. We may have $d(i) = 1$ for all i or as in [12] $d(i) = \lfloor i/2 \rfloor$ (and the number of packets sent $s_p(i)$ is proportional to i). Generally speaking, we require that $d(i) = i$ only for $i = 1$, while for $i > 1$ we have that $d(i) < i$.

Proposition 1: If $\vec{M}^N(0)$ and $R^N(0)$ converge almost surely to $\vec{\mu}(0)$ and $\vec{\rho}(0)$, respectively, and $\forall i = 1, \dots, I_p$ it holds that $M_{p,i,s}^N(0) / \sum_{s'=1}^{s_p(I_p)+1} M_{p,i,s'}$ converge almost surely to $\tau(s)$, as $N \to \infty$, then for any finite horizon t we have that:

$$
\begin{aligned}
&\lim_{N\to\infty} \sum_{s=1}^{s_p(I_p)+1} M_{p,i,s}^N(t) = \tilde{\mu}_{p,i}(t) \\
&\lim_{N\to\infty} M_{u,i}^N(t) = \mu_{u,i}(t) \qquad \lim_{N\to\infty} \vec{R}^N(t) = \vec{\rho}(t)
\end{aligned}
$$

almost surely, where $\tilde{\mu}_{p,i}(t)$, $\mu_{u,i}(t)$ and $\rho(t)$ are defined by the iterative scheme depicted in Table I.

B. Modelling the SAP-LAW

In SAP-LAW [5], the gateway counts the arrived UDP packets in a time interval and hence estimates the total instantaneous UDP arrival rate at time t, $\xi(t)$. Let NC be the total capacity of the gateway, and $N_p(t)$ the number of TCP connections at time t, then the maximum window size which is sent back to the TCP transmitters is the same for all the TCP connections and equal to:

$$
maxTCP(t) = \frac{NC - \xi(t)}{N_p(t)} \,. \tag{9}
$$

In our setting, we assume the time interval that we use to estimate the instantaneous arrival rate for the UDP packets to be equal to a multiple Y of the round trip time, i.e., $\xi(t) = y(S_u^N(t-Y), \dots S_u^N(t-1))$. The notation, unless differently specified, is that of Section IV-A. For the sake of simplicity, we assume that $s_p(i) = \alpha i$ for some $\alpha \in \mathbb{N}$, i.e., the maximum number of packets sent in a time slot by a TCP connection grows linearly with the state numbering.

$$\begin{aligned}
\tilde{\mu}_{p,1}(t+1) &= \sum_{j:d(j)=1} \left(1-(1-q(\rho_p(t)))^{s_p(j)}\right)\tilde{\mu}_j(t)(1-w)^{s_p(j)} + \sum_{j=1}^{I_p}\tilde{\mu}_{p,j}(t)(1-(1-w)^{s_p(j)}) && (7)\\
\tilde{\mu}_{p,i}(t+1) &= \sum_{j:d(j)=i} \left(1-(1-q(\rho_p(t)))^{s_p(j)}\right)\tilde{\mu}_j(t)(1-w)^{s_p(j)} + (1-q(\rho_p(t)))^{s_p(i-1)}\tilde{\mu}_{p,i-1}(t)(1-w)^{s_p(i-1)} \quad 1<i<I_p\\
\tilde{\mu}_{p,I_p}(t+1) &= (1-q(\rho_p(t)))^{s_p(I_p-1)}\tilde{\mu}_{p,I_p-1}(t)(1-w)^{s_p(I_p-1)} + (1-q(\rho_p(t)))^{s_p(I_p)}\tilde{\mu}_{p,I_p}(t)(1-w)^{s_p(I_p)}\\
\mu_{u,i}(t+1) &= \sum_{j=1}^{I_u}\kappa_{(u,j),(u,i)}\mu_{u,j}(t) \qquad \sigma_p(t+1)=\sum_{i=1}^{I_p}\tilde{\mu}_{p,i}(t+1)\frac{1-(1-w)^{s_p(i)}}{w} && (8)\\
\sigma_u(t+1) &= \sum_{i=1}^{I_u}\mu_{u,i}(t+1)s_u(i) \qquad \rho_c(t+1)=\max\left(\rho_c(t)+\sigma_p(t+1)+\sigma_u(t+1)-C,0\right)\\
\rho_p(t+1) &= \rho_c(t) \qquad \rho_{up}(t+1)=\rho_{uc}(t), i>0 \qquad \rho_{uc}(t+1)=\sigma_u(t+1)
\end{aligned}$$

TABLE I
EQUATIONS FOR THE MODEL OF ECN/RED.

The memory $\vec{R}^N(t) = (R_c^N(t), R_p^N(t), \vec{R}_u^N(t))$ is a pair of real numbers (r_c, r_p) followed by a vector of reals $\vec{r}_u$ where r_c denotes the normalised queue length at the bottleneck (counting both TCP and UDP packets) at a given time slot, r_p is the normalised queue length at the previous time slot and $\vec{r}_u = (r_{u0}, \ldots, r_{uY})$ denotes the normalised counting of the arrived UDP packets at the latest $(Y+1) > 1$ time slots.

The transitions for UDP objects are the same of those shown in Section IV-A.

Each TCP connection has a state represented by a pair of numbers $(i,j) \in \mathbb{N}^2$, where i denotes the state corresponding to the sender window size $s_p(i)$, while j takes into account the number of received acknowledgements. Indeed, the SAP-LAW mechanism does not allow the sender to use its full window size and consequently the growth of the sender window size is in general slower than what would be observed with a gateway with infinite capacity. Formally, the state of an object associated with a TCP connection has the form $(p, \vec{x}, s)$ where p marks the TCP object, $\vec{x}$ is

$$\vec{x} = \begin{cases} (i,j) : 1 \le i \le I_p \wedge 0 \le j < i & \text{if } l = p \\ i : 1 \le i \le I_u & \text{if } l = u\,, \end{cases}$$

and s denotes the number of packets that has to be sent. Similarly to what we proposed in Section IV-A, we assume that the number of packets sent by a TCP connection before being closed and restarted is geometrically distributed with mean $1/w$. The dynamic of the memory is specified by the following equation:

$$\begin{aligned}
R_c^N(t+1) &= \max\left(R_c^N(t) + S_p^N(t+1) + S_u^N(t+1) - C, 0\right),\\
R_p^N(t+1) &= R_c^N(t)\,,\\
R_{ui}^N(t+1) &= R_{u(i-1)}^N(t) \qquad\qquad 1 \le i \le Y\\
R_{u0}^N(t+1) &= S_u^N(t+1)\,.
\end{aligned}$$

Function $S_u^N(t)$ is specified as shown in Section IV-A, while S_p^N is defined as:

$$S_p^N(t) = \sum_{i=1}^{I_p}\sum_{j=0}^{i-1} M_{p,i,j}(t)\cdot\min\left(s_p(i), s_p(h(\vec{R}_u^N(t)))\right)\,,$$

where

$$h(\vec{r}_u) = \text{argmax}j\left(s_p(j), s_p(j) \le (C - \tilde{y}(\vec{r}_u))\frac{1}{p_p} \vee j = 1\right)\,, \quad (10)$$

and $\tilde{y}(\vec{r}_u) = y(r_{u1}, \ldots, r_{uY})$. Given a state (i,j) of a TCP connection, and the state z corresponding to the maximum sending window obtained by Equation (10), the following state of the connection is $f(i,j,z)$ defined as follows:

$$f(i,j,z) = \begin{cases} (i, j+\min(z,i)) & \text{if } j+\min(z,i) < i\\ (i+1, j+\min(z,i)-i) & \text{if } j+\min(z,i) \ge i \wedge i < I_p \\ (I_p, I_p - 1) & \text{otherwise}\,. \end{cases} \quad (11)$$

Concluding, we obtain the following transition matrix for TCP objects:

$$\begin{aligned}
&K_{(p,i,j,s),(p,i',j',s')}(\vec{r}) = 1_{f(i,j,h(\vec{r}_u))=(i',j')}\\
&\quad\cdot 1_{s > s_p(h(\vec{r}_u))}\tau(s')\\
&K_{(p,i,j,s),(p,1,1,s')}(\vec{r}) = 1_{s \le s_p(h(\vec{r}_u))}\tau(s')
\end{aligned}$$

where $\vec{r} = (r_p, r_c, r_{u0}, \vec{r}_u) = (r_p, r_c, r_{u0}, r_{u1}, \ldots, r_{uY})$ is a value of the memory at a certain time slot and $\tau(s')$ is defined in (5).

Proposition 2: If $\vec{M}^N(0)$ and $R^N(0)$ converge almost surely to $\vec{\mu}(0)$ and $\vec{\rho}(0)$, respectively, and $\forall i = 1, \ldots, I_p$ it holds that $M_{p,i,j,s}^N(0)/\sum_{s'=1}^{s_p(I_p)+1} M_{p,i,j,s'}$ converge almost surely to $\tau(s)$, as $N \to \infty$, then for any finite horizon t we have that:

$$\begin{aligned}
&\lim_{N\to\infty}\sum_{s=1}^{s_p(I_p)+1} M_{p,i,j,s}^N(t) = \tilde{\mu}_{p,i,j}(t)\\
&\lim_{N\to\infty} M_{u,i}^N(t) = \mu_{u,i}(t) \qquad \lim_{N\to\infty}\vec{R}^N(t) = \vec{\rho}(t)
\end{aligned}$$

almost surely, where $\tilde{\mu}_{p,i,j}(t)$, $\mu_{u,i}(t)$ and $\vec{\rho}(t)$ are defined by the iterative scheme of Table II and function h is defined by Equation (10).

V. ANALYSIS OF THE PERFORMANCE

In this section we study the expected queue length Q, the system's throughput T and the expected waiting time W for

$$
\begin{aligned}
\tilde{\mu}_{p,1,1}(t+1) &= \sum_{i'=1}^{I_p}\sum_{j'=1}^{i} \tilde{\mu}_{p,i',j'}(t)(1-(1-w)^{\min(s_p(i'),s_p(h(\vec{\rho}_u(t))))}) \\
\tilde{\mu}_{p,i,j}(t+1) &= \sum_{i'=1}^{I_p}\sum_{j'=1}^{i} \tilde{\mu}_{p,i',j'}(t)1_{f(i',j',h(\vec{\rho}_u(t)))=(i,j)}(1-w)^{\min(s_p(i'),s_p(h(\vec{\rho}_u(t))))} \\
\sigma_p(t+1) &= \sum_{i=1}^{I_p} \tilde{\mu}_{p,i,j}(t+1)\frac{1-(1-w)^{\min(s_p(i),s_p(h(\vec{\rho}_u(t))))}}{w} \qquad \sigma_u(t+1) = \sum_{i=1}^{I_u} \mu_{u,i}(t+1)s_u(i) \\
\rho_c(t+1) &= \max\Big(\rho_c(t)+\sigma_p(t+1)+\sigma_u(t+1)-C, 0\Big) \\
\rho_p(t+1) &= \rho_c(t) \qquad \rho_{up}(t+1) = \rho_{uc}(t), i>0 \qquad \rho_{uc}(t+1) = \sigma_u(t+1)
\end{aligned}
$$

TABLE II
EQUATIONS FOR THE MODEL OF SAP/LAW.

	SAP-LAW	ECN $\gamma = 5E-6$	ECN $\gamma = 5E-7$
T	591.27	585.69	594.23
Q	0	5.73	53.72
W	0	0.098	0.0901

TABLE III
ECN/RED VS. SAP-LAW: SCENARIO WITH GREEDY TCP AND SMOOTH UDP. PARAMETERS: $I_p = 100$, $I_u = 10$, $s_p(i) = 10 * i$, $C = 600$ $s_u = (0, \dots, 1000, 3000, 1000)$.

	SAP-LAW	ECN $\gamma = 5E-6$	ECN $5E-7$
T	n/a	276.00	428.85
Q	n/a	64.52	155.34
W	n/a	0.234	0.3622

TABLE IV
ECN/RED VS. SAP-LAW: SCENARIO WITH GREEDY TCP AND BURSTY UDP. PARAMETERS: $I_p = 100$, $I_u = 10$, $s_p(i) = 10 * i$, $C = 600$ $s_u = (0, \dots, 1000, 3000, 1000)$.

	SAP-LAW	ECN $\gamma = 5 \cdot^1 0{-}6$	ECN $\gamma = 5 \cdot 10^{-7}$
T	713.27	333.29	529.4077
Q	153.95	35.20	83.79
W	0.2000	0.1056	0.1583

TABLE V
ECN/RED VS. SAP-LAW: SCENARIO WITH GREEDY TCP AND BURSTY UDP. PARAMETERS: $I_p = 100$, $I_u = 10$, $s_p(i) = 10 * i$, $C = 800$ $s_u = (0, \dots, 1000, 3000, 1000)$.

the END/RED and SAP-LAW model. In all the models of our experiments we use for SAP-LAW $Y = 1$ and function $\tilde{y}$ is the identity, i.e., we use a time window of one round trip time to estimate the UDP traffic. Function $d(i)$ for ECN/RED simply halves the size of the current window. The percentage of TCP objects is 70% of the total. The finite horizon is fixed at 800 time slots.

Greedy TCP connections and smooth UDP traffic. We study the models for ECN/RED and SAP-LAW under the scenario of greedy TCP connections, i.e., they tend to use all the size of their sending windows ($w \to 0$), and smooth UDP traffic. The results of the experiment and its parameters are shown in Table III. In this scenario, the smoothness of the UDP traffic allows the SAP-LAW gateway to accurately predict the bandwidth to reserve for UDP. For this reason, in this ideal scenario, SAP-LAW gives the best performances by obtaining a high throughput with no waiting time.

Greedy TCP connections and bursty UDP traffic. We compare the ECN/RED and the SAP-LAW congestion control mechanisms under bursty UDP. In this case the SAP-LAW model shows an instable behaviour that proves that this mechanism alone is not sufficient to safely prevent the congestion in any scenario. This issue is presented when the UDP traffic is very bursty and the monitor window is small as in our case. The SAP-LAW policy does not take into account the actual queue length but only estimates the instantaneous UDP traffic. Therefore, this parameter must be chosen carefully when implementing this congestion control mechanism. Table IV shows the performance measures for ECN. It is noteworthy pointing out that the burstiness of the UDP activity strongly deteriorates the performance measures of TCP traffic. In Table V we show the behaviour of the two congestion mechanisms under bursty UDP traffic but in stability. We can see that although the throughput of SAP-LAW is much higher than that of ECN, also the queue length tends to be higher, so reducing the benefits that we observed in the scenario with smooth UDP.

Temporary TCP connections. In this section we assume that each TCP connection has to send a number of packets modelled by an independent geometric random variable with mean $w^{-1} < \infty$. After sending this amount of packets, the connection restarts from state 1. The results of the analysis are shown in Table VI. The table shows that SAP-LAW pays a reduction of the throughput because the mechanism is fair in dividing the residual bandwidth among the TCP connections not considering that some of them may be in their initial phase. This does not happen for ECN/RED since it tends to reduce the window size of fast connections and allows the new ones to grow. This is even more evident when we introduce UDP burstiness. Figure 1 and 2 compare the queue length and the bottleneck's throughput for different sizes of

	SAP-LAW	ECN $\gamma = 5E-6$	ECN $\gamma = 5E-7$
T	219.17	300.06	295.50
Q	0.2160	361.1346	39.335
W	$0.85E-4$	1.2035	0.133

TABLE VI
ECN/RED VS. SAP-LAW: SCENARIO WITH GREEDY TCP AND SMOOTH UDP. PARAMETERS: $I_p = 100$, $I_u = 10$, $s_p(i) = 10 * i$, $C = 800$ $s_u = (0, \dots, 1000, 3000, 1000)$, $w = 10^4$.

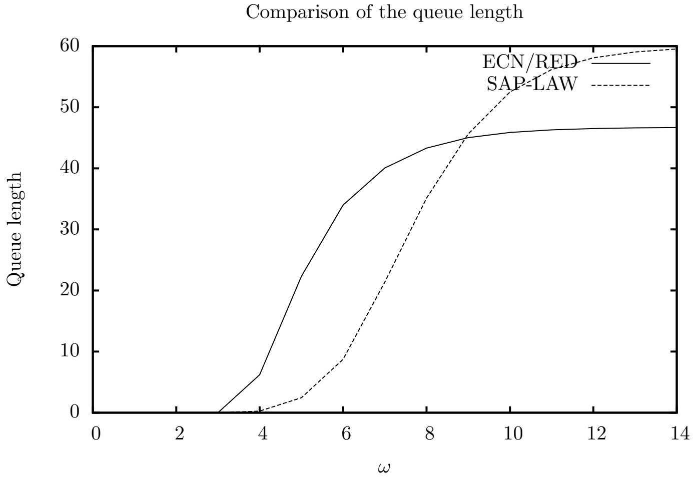

Fig. 1. ECN/RED vs. SAP-LAW: queue length in the scenario with UDP burstiness and temporary TCP connections.

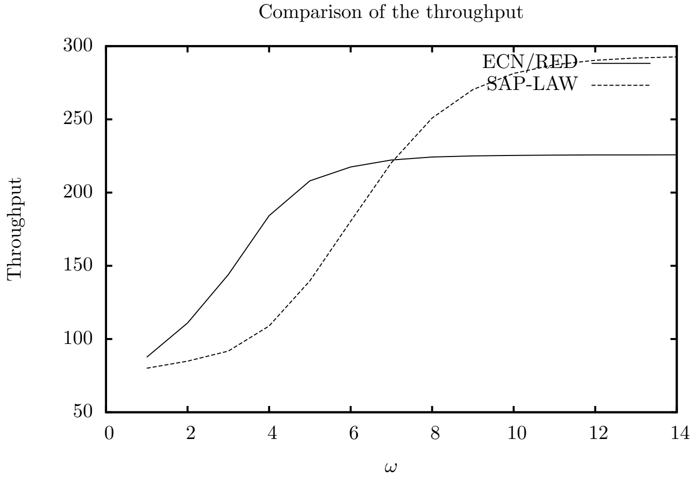

Fig. 2. ECN/RED vs. SAP-LAW: throughput in the scenario with UDP burstiness and temporary TCP connections.

w. We assume $w(\omega) = 10^3 2^\omega$, and $C = 300$, $I_p = 100$, $s_p(i) = 10i$, $I_u = 15$, $s_u = (0, 0, \dots, 500, 600, 500)$. Notice that for low values of w ECN/RED shows its benefits in the queue length control policy as discussed above, whereas when w grows, SAP-LAW takes advantage of being more aggressive in the bandwidth usage and this compensates the assignment of bandwidth to slow connections.

VI. CONCLUSION

AQM mechanisms try to exploit the congestion/flux control mechanism of TCP to avoid the saturation of the bottleneck link and provide a smoother use of the available bandwidth. In this paper we have discussed some classic and an innovative AQM technique and analysed their performance through the use of mean field models. We have considered a large number of transmissions and two different behaviours for both the TCP connections and for the UDP traffic so as to represent their general behavior depending on the applications relying on them. Through our analysis, we have shown that SAP-LAW is able to ensure a shorter expected queue length and a higher throughput than classic ECN/RED schemes in case of greedy TCP connections. On the other hand, with temporary TCP connections, ECN/RED performs better as SAP-LAW reserves more-than-needed bandwidth to TCP connections in their initial phases. Furthermore, we have also shown that erroneous estimations of the UDP traffic may lead to instability of SAP-LAW and that the burstiness of UDP traffic can strongly reduce the throughput achievable even when employing ECN/RED.

REFERENCES

[1] A. S. Tanenbaum, *Computer Networks*. Prentice-Hall, 2003.

[2] L. L. Peterson and B. S. Davie, *Computer Networks: A Systems Approach*. Elsevier, 2007.

[3] S. Floyd, “TCP and Explicit Congestion Notification,” *SIGCOMM Comput. Commun. Rev.*, vol. 24, no. 5, pp. 8–23, 1994.

[4] S. Floyd and V. Jacobson, “Random early detection gateways for congestion avoidance,” *IEEE/ACM Trans. on Networking*, vol. 1, no. 4, pp. 397–413, 1995.

[5] C. E. Palazzi, S. Ferretti, M. Roccetti, G. Pau, and M. Gerla, “What’s in that Magic Box? The Home Entertainment Center’s Special Protocol Potion, Revealed,” *IEEE Trans. on Consumer Electronics*, vol. 52, no. 4, pp. 1280–1288, 2006.

[6] I. Han, H.-S. Park, Y.-K. Jeong, and K.-R. Park, “An Integrated Home Server for Communication, Broadcast Reception, and Home Automation,” *IEEE Trans. on Consumer Electronics*, vol. 52, no. 1, pp. 104–109, 2006.

[7] M. Jeonghoon, R. La, V. Anantharam, and J. Walrand, “Analysis and Comparison of TCP Reno and Vegas,” in *Proc. of INFOCOM 1999*, vol. 3, 1999, pp. 1556–1563.

[8] S. Liu, T. Başar, and R. Srikant, “TCP-Illinois: A Loss- and Delay-based Congestion Control Algorithm for High-Speed Networks,” *Performance Evaluation*, vol. 65, no. 6–7, pp. 417 – 440, 2008.

[9] E. Altman, K. Avrachenkov, and C. Barakat, “Tcp in presence of bursty losses,” in *Proc. of SIGMETRICS*, 2000, pp. 124–133.

[10] T. Bonald, M. May, and J.-C. Bolot, “Analytic Evaluation of RED Performance,” in *Proc. of INFOCOM 2000*, vol. 3, 2000, pp. 1415–1424.

[11] W. Kang, F. Kelly, N. H. Lee, and R. R.J. Williams, “Fluid and Brownian Approximations for an Internet Congestion Control Model,” in *Proc. of CDC 2004*, vol. 4, 2004, pp. 3938–3943.

[12] P. Tinnakornsrisuphap and A. Makowski, “Limit Behavior of ECN/RED Gateways Under a Large Number of TCP flows,” in *Proc. of INFOCOM 2003*, vol. 2, 2003, pp. 873–883.

[13] J.-Y. Le Boudec, D. McDonald, and J. Mundinger, “A Generic Mean Field Convergence Result for Systems of Interacting Objects,” in *Proc. of QEST 2007*. IEEE, 2007, pp. 3–18.

[14] R. Srikant, *The Mathematics of Internet Congestion Control*. Springer, 2004.