Analytic Evaluation of RED Performance

Thomas Bonald
CNET - France Telecom
Thomas.Bonald@cnet.francetelecom.fr

Martin May
INRIA Sophia-Antipolis
mmay@sophia.inria.fr

Jean-Chrysostome Bolot
ENSIM Corporation
bolot@ensim.com

Abstract— **End-to-end congestion control mechanisms such as those in TCP are not enough to prevent congestion collapse in the Internet (for starters, not all applications might be willing to use them), and they must be supplemented by control mechanisms inside the network. The IRTF has singled out Random Early Detection (RED) as one queue management scheme recommended for rapid deployment throughout the Internet. However, RED is not a thoroughly understood scheme – witness for example how the recommended parameter settings, or even the various benefits RED is claimed to provide, have changed over the past few years.**

In this paper, we describe simple analytic models for RED, and use these models to quantify the benefits (or lack thereof) brought about by RED. In particular, we examine the impact of RED on the loss and delay suffered by bursty and less bursty traffic (such as TCP and UDP traffic, respectively). We find that (i) RED does eliminate the higher loss bias against bursty traffic observed with Tail Drop, but not by decreasing the loss rate of bursty traffic, rather by increasing that of non bursty traffic; (ii) the number of consecutive packet drops is higher with RED than Tail Drop, suggesting RED might not help as anticipated with the global synchronization of TCP flows; (iii) RED can be used to control the average queueing delay in routers and hence the end to end delay, but increases the jitter of non bursty streams. Thus, applications that generate smooth traffic, such as interactive audio applications, will suffer higher loss rates and require large playout buffers, thereby negating at least in part the lower mean delay brought about by RED.

I. Introduction

Buffers are a key component of a packet-switched network, as they absorb burst arrivals of packets and hence reduce losses. Larger buffers can absorb larger bursts, but they tend to build up at high load and increase queueing delays. The traditional technique for managing delay is to set a maximum length for each buffer queue, accept packets in the queue until the maximum length is reached, then drop subsequent incoming packets until the queue decreases below its maximum value. This buffer management scheme is referred to as Tail Drop.

End-to-end control mechanisms are used in the Internet to regulate the amount of traffic in the network and match it to available capacity, thereby making sure that queue lengths and loss rates remain reasonable. The most widely used control mechanism is TCP's window based mechanism [12]. TCP has prevented an Internet-wide collapse, however some thorny problems remain. For example, the mechanism in TCP tends to keep queue occupancy high, and thus tends to discriminate against bursty traffic (since bursts of packets arriving at a router won't find much free buffer space to squeeze into). Furthermore, TCP traffic itself is bursty[1] [16], [1], which means that a loss event at a router tends to involve many packets at a time, leading to reduced throughput and synchronization between TCP connections sharing the ressources of that router. Finally, not all applications are willing to use control mechanisms; in particular, many interactive audio (IP telephony) applications send data at a rate independent of the state of congestion in the network, and thus grab all the bandwidth of the network when competing with rate adaptive applications such as those that rely on TCP. Clearly, the uncontrolled use of such applications again raises the possibility of Internet-wide congestion collapse.

The difficulties above bring out the necessity to complement end-to-end control mechanisms with router-based control mechanisms that extend beyond the current Tail Drop scheme. The Internet Research Task Force (IRTF) produced a document, now an information RFC [3], urging the deployment of router-based control schemes. Specifically, the document, often referred to as the "RED manifesto", singles out the Random Early Detection (RED) scheme, as the recommended scheme for use in the Internet.

The RED scheme was initially described and analyzed in [8]. Basically, RED starts dropping packets randomly before the buffer gets full. Thus, it forces connections to back off before the buffer fills up and multiple packets are dropped; if connections ignore packet drops and keep sending at too-high rates, they keep suffering from high loss rates. RED is claimed to provide several benefits, in particular 1) decrease the end-to-end delay for both responsive (TCP) and non necessarily responsive real-time traffic (UDP), 2) prevent large number of consecutive packet losses by ensuring available buffer space even with bursty traffic, and 3) remove the higher loss bias against bursty traffic observed with Tail Drop. Some of these claims have been validated with simulation studies. However, despite the IRTF recommendation that RED be widely deployed, RED is not thoroughly understood: there is little operational experience of RED in large scale networks – one of the few published measurement study is limited in scope because it only considers the router

[1] The burstiness of TCP traffic can be explained by user behavior, and by characteristics of the TCP closed-loop feedback control mechanism, coupled with ACK compression [5]. Thus, it appears to be a salient feature of the Internet.

performance (as opposed to the end-to-end performance) and it does not clearly describe the measurement settings and the exact information being measured [4] –, it is not quite clear how to choose RED parameters (and indeed the recommended values have changed over time), and there is, to our knowledge, no published analytical model of RED[2] that would for example allow us to quantify the impact of parameters settings on performance, or the impact of different parameters values taken by different ISPs in a large network.

In this paper, we develop simple analytic models for the RED and Tail Drop buffer management schemes, and use these models to quantify the benefits (or lack thereof) brought about by RED. In particular, we examine the impact of RED on the loss rates, the number of consecutively lost packets, the mean delay, and the delay jitter, suffered by bursty and less bursty traffic (such as TCP and UDP traffic, respectively). We find that (i) RED does indeed eliminate the bias against bursty traffic observed with Tail Drop (claim 3 in the previous paragraph); however it does so not by decreasing the loss probability of bursty (TCP) traffic, but rather by increasing that of smooth (UDP) traffic; (ii) the number of consecutively lost packets is larger with RED than with Tail Drop, suggesting that RED might not help as much as anticipated with the global synchronization of TCP flows (claim 2 above), (iii) RED is crucial to control the average queueing delay in routers and hence the end-to-end delay (claim 1 above), but increases the jitter of non bursty (UDP) streams, and hence their playout buffer requirements, thereby negating at least in part the gains on the lower mean delay. In addition to these three main findings, we also show that the often used claim that the loss rate suffered by a flow in a RED router is proportional to the flow intensity (claim first made in [8]) is true only if the flow arrival process is Poisson (specifically, it requires the PASTA property).

The rest of the paper is organized as follows. In Section II, we describe our basic model, and use it to examine the bias of Tail Drop against bursty traffic and whether RED eliminates this bias. In Section III, we examine the number of consecutively lost packets in both Tail Drop and RED routers. In Section IV, we compare the average delay and delay jitter in Tail Drop and RED routers. In Section V, we use simulations to validate the analytic results obtained with the model, and to further examine the issue of delay jitter for UDP flow with RED routers. Section VI concludes the paper.

II. BIAS AGAINST BURSTY TRAFFIC

In this section, we describe our basic model, and use it to examine the bias of Tail Drop against bursty traffic and whether RED eliminates this bias.

We consider a router with a buffer size of K packets. With the RED buffer management scheme, incoming packets are dropped with a probability that is an increasing function d of the average queue size $\hat{k}$. The average queue size is estimated using an exponential weighted moving average :

$$\hat{k} \leftarrow (1-w)\hat{k} + w\, k,$$

[2]There are specific models such as in [11] that abstract a RED router as a router in which the loss rate is proportional to input flow intensity; we get back to such models in Section II.

where w is a fixed (small) parameter and k is the instantaneous queue size. A typical drop function d is defined by three parameters $\min_{\mathrm{th}}$, $\max_{\mathrm{th}}$ and $\max_{\mathrm{p}}$ as follows :

$$d(\hat{k}) = 0 \quad \text{if } \hat{k} < \min_{\mathrm{th}}, \quad d(\hat{k}) = 1 \quad \text{if } \hat{k} \geq \max_{\mathrm{th}},$$

$$d(\hat{k}) = \frac{\hat{k} - \min_{\mathrm{th}}}{\max_{\mathrm{th}} - \min_{\mathrm{th}}} \times \max_{\mathrm{p}} \quad \text{otherwise.}$$

Refer to Figure 1.

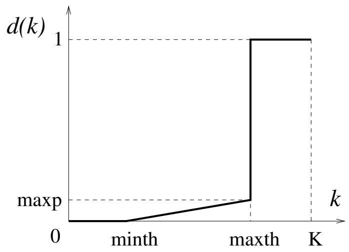

Fig. 1. Drop function of RED

A. A RED router with bursty input traffic

Let us first derive a model of a RED router with a single input stream of bursty traffic. We assume that packets arrive according to a batch Poisson process; specifically, bursts (or batches) of B packets arrive according to a Poisson process of rate λ. Note that this model does not really match empirically derived models of TCP and other bursty traffic patterns [13], [16], [19]. However, it is analytically tractable; furthermore, our purpose here is to compare the relative impact of RED on bursty and less bursty traffic. We can imagine (and this will be confirmed with simulation in Section V) that the difference between a smooth input traffic and a batch Poisson process (as examined here) would be a lower bound to that observed between a smooth input and an input process with long range dependence. The processing times of the packets in the router are assumed to be exponentially distributed with mean μ^{-1}. We define the *offered load* by $\rho = B\lambda/\mu$.

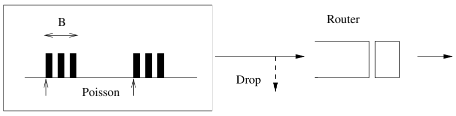

Fig. 2. Model of RED router with bursty input traffic

The number of packets buffered in the queue defines a Markov chain, the stationary distribution of which can be easily computed. We denote by π this stationary distribution. Using the PASTA property, we obtain the drop probability of a packet in a Tail Drop router:

$$P_{\mathrm{TD}} = \pi(K) + \pi(K-1)\frac{B-1}{B} + \ldots + \pi(K-B+1)\frac{1}{B}.$$

We now consider a RED router, and we make the assumption for now that the drop rate $d(k)$ depends on the instantaneous queue size k rather than on the average queue size $\hat{k}$ (i.e. we assume

$w = 1$). Note that there is no reason for choosing $\max_{\mathrm{th}} < K$ in this case, hence we let $\max_{\mathrm{th}} = K$. We use the following approximation:

Approximation 1: The RED router uses the same drop probability $d(k)$ on all packets in the same burst, where k is the instantaneous queue size at the time the first packet in the burst arrives at the router.

Note that in reality the difference between the drop probability of the first packet of the burst $d(k)$ and the drop probability of any other packet of the burst cannot exceed:

$$\Delta d(k) = d(k + B - 1) - d(k).$$

Thus, the approximation above provides a lower bound on the drop rate. Furthermore, it is accurate ($\Delta d(k)$ is small) when the drop function is sufficiently smooth (namely for small values of $\min_{\mathrm{th}}$ and high values of $\max_{\mathrm{p}}$) and the burst size B is not too large compared to the buffer size K. Now, using the PASTA property again, we approximate the drop probability of a packet in a RED router by:

$$P_{\mathrm{RED}} = \pi(K) + \pi(K-1)d(K-1) + \ldots + \pi(1)d(1).$$

Note that the stationary distribution π in this case is different from that obtained with Tail Drop.

Example 1: Consider a buffer size of $K = 40$ packets, with RED parameters $\min_{\mathrm{th}} = 20$, $\max_{\mathrm{th}} = 40$ and $\max_{\mathrm{p}} = 1$. Figure 3 shows the drop probability of an incoming packet as a function of offered load for different burst sizes, obtained by previous analysis (with Approximation 1) and by simulation (without Approximation 1). The figure clearly shows that the approximation is very accurate, even for large values of the burst size.

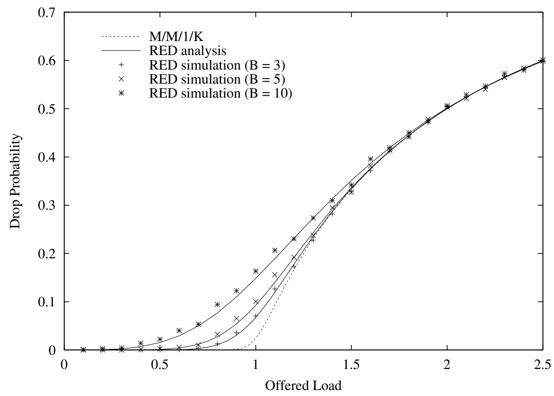

Fig. 3. Drop probability vs. offered load for different values of the burst size.

Note that the drop probability is always higher with RED than with Tail Drop (this is a sample-path property). For large offered load (which may represent transient congestion periods), the drop probability is very close to that suffered by a Poisson traffic in a Tail Drop router, which is given by the loss probability for the $M/M/1/K$ queue:

$$P_{M/M/1/K} = 1 - \frac{1-\rho^K}{1-\rho^{K+1}}.$$

We conclude that whatever the burst size,

$$P_{\mathrm{RED}} \sim P_{\mathrm{TD}} = 1 - \frac{1}{\rho} + o\left(\frac{1}{\rho}\right) \quad \text{when } \rho >> 1. \qquad (1)$$

B. A RED router with bursty and smooth input traffic

We consider now a router with two input flows, one bursty with batch Poisson arrivals as above and batch size B (we take $B = 3$ in the numerical examples below), the other a smoother (non batch) Poisson stream. We denote by $\rho(b)$ and $\rho(s)$ the load of the bursty and the smooth traffic, and by $\rho = \rho(b) + \rho(s)$ the total offered load.

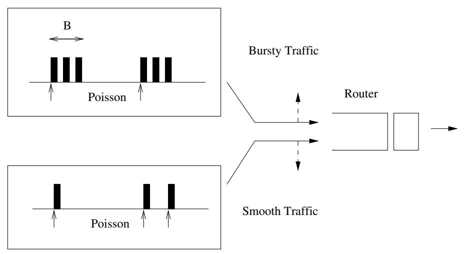

Fig. 4. Model of RED router with a mix of bursty and smooth traffic

Let π be the stationary distribution of the total number of packets in the queue. Using the PASTA property, we obtain the drop probability of a packet for the bursty flow and the smooth flow in a Tail Drop router:

$$P_{\mathrm{TD}}(b) = \pi(K) + \pi(K-1)\frac{B-1}{B} + \ldots + \pi(K-B+1)\frac{1}{B}$$

and

$$P_{\mathrm{TD}}(s) = \pi(K).$$

Clearly $P_{\mathrm{TD}}(b) > P_{\mathrm{TD}}(s)$, meaning that there is a bias against bursty traffic with Tail Drop. On the other hand, we obtain for the RED router (using the same approximation as earlier)

$$P_{\mathrm{RED}}(b) = \sum_{k=1}^{K} \pi(k)d(k) = P_{\mathrm{RED}}(s),$$

meaning that there is no bias against bursty traffic with RED. In fact, RED distributes the drops among both types of traffic. Noting that

$$P_{\mathrm{TD}} = \frac{\rho(b)}{\rho}P_{\mathrm{TD}}(b) + \frac{\rho(s)}{\rho}P_{\mathrm{TD}}(s),$$

we obtain in view of (1) for high values of the offered load,

$$P_{\mathrm{RED}}(b) = P_{\mathrm{RED}}(s) \sim \frac{\rho(b)}{\rho}P_{\mathrm{TD}}(b) + \frac{\rho(s)}{\rho}P_{\mathrm{TD}}(s). \qquad (2)$$

C. Including queue size averaging in the model

We have so far assumed that the drop probability in the RED router only depends on the instantaneous queue size. Adding queue size averaging increases the complexity of the model (as it increases the memory needed to keep track of past queue sizes). A key observation, however, is that when the weight w of the moving average scheme is small (which is the case in practice), the estimated average queue size $\hat{k}$ varies slowly, so that consecutive packets belonging to the same burst are likely to experiment the same drop probability $d(\hat{k})$. As a result, the Approximation 1 used in previous analysis is still valid in this case. Even more, it is acurate whatever the drop function, and in particular for the RED parameters recommended in [8].

Example 2: Consider a buffer of size $K = 40$ and RED parameters $\min_{\mathrm{th}} = 10$, $\max_{\mathrm{th}} = 30$, $\max_{\mathrm{p}} = 0.1$ and $w = 0.002$. Figures 5 and 6 show the drop probability as a function of the fraction of bursty traffic in the input traffic, obtained using the analytic expressions above (continuous line for RED, dashed for Tail Drop), and using simulations (done with queue size averaging, and without Approximation 1). Figure 5 shows that, with an offered load of $\rho = 2$, the drop probability is the same for both types of traffic with RED, namely $P_{\mathrm{RED}}(s) \sim P_{\mathrm{RED}}(b) \sim 0.5$, and it is equal to the average drop probability with Tail Drop, as predicted by equations (1) and (2).

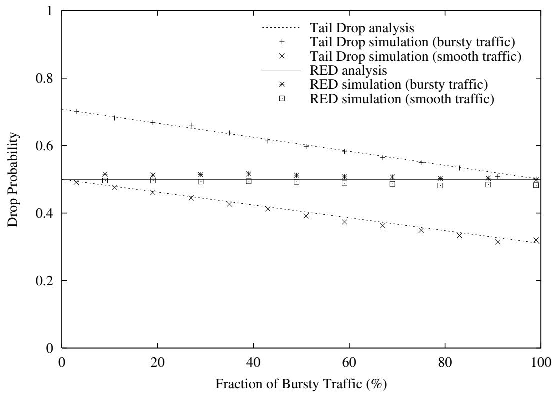

Fig. 5. Drop probability vs. fraction of bursty traffic for an offered load of $\rho = 2$

We conclude that RED avoids the bias against bursty traffic, and that this results in a significant decrease of the drop rate suffered by bursty traffic *only* when the fraction of bursty traffic is small (see Figure 5). Otherwise, the main effect of RED is to increase the drop probability of smooth traffic, without improving the drop probability of bursty traffic. In practice, if we replace "bursty" with "TCP" and "smooth" with "interactive UDP audio" for example, and if we note that TCP makes up the vast majority of Internet traffic, the result above means that the overall loss rate suffered by TCP connections when going from Tail Drop to RED will not change much, but that the loss rate suffered by UDP/IP telephony applications (whether they are rate adaptive or not) will increase significantly. In all cases, the *drop rate* (namely the number of packets dropped per unit of time) of a flow going through a RED router does not depend of the *burstiness* of this flow, but only on the *load* it generates (refer to Equation (2) above).

D. An important observation about PASTA

It is important to note that the analysis above heavily relies on the PASTA property of Poisson processes. In general, it is not true that the stationary distribution of the number of packets k buffered in the queue immediately *before* the arrival of a

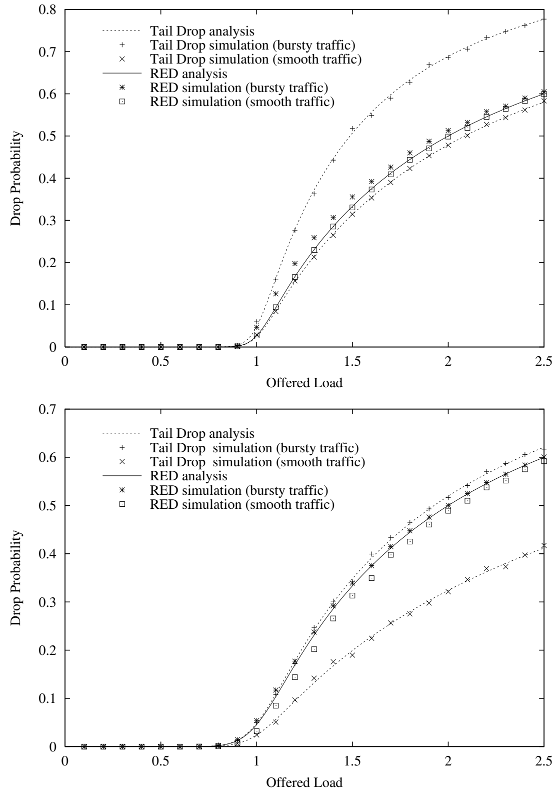

Fig. 6. Drop probability vs. offered load for small (10% – top) and large (90% – bottom) fraction of bursty traffic

burst of packets (that is under the Palm probability [2]) coincides with π, the continuous-time stationary distribution of k. This means that the claim made in [8], and used in recent models of additive increase and multiplicative decrease congestion control schemes in a RED environment (e.g. [11]), namely the loss rate of a flow in a RED router is proportional to the intensity of the flow, is valid only for Poisson flows. However, it is *not valid* for other types of flows found in practice such as periodic flows, or flows with heavy-tailed characteristics. For example, Figure 7 shows the drop probabilities obtained in a RED router with both a bursty input traffic with Pareto inter-arrival times between bursts and a Poisson input traffic. The Pareto coefficient in the figure is 1.4 and the RED parameters are those of Example 2. Unlike what we saw earlier in the case of the batch Poisson arrival process, the drop probability for the Pareto traffic is different from the drop probability for smooth traffic even for the RED router. Thus, it is important to be aware of, and careful about, strong traffic assumptions one might making when modeling RED routers as in [11].

III. SYNCHRONIZATION OF TCP FLOWS

The combination of a TCP mechanism which keeps queue occupancy high, of bursty TCP traffic, and of the Tail Drop bias against bursty traffic, means that loss events at a router tend to involve many packets. If these packets belong to different TCP connections, these connections then experience losses at about the same time, decrease their rates/windows in synchrony, and then tend to stay synchronized. This phenomenom, referred to

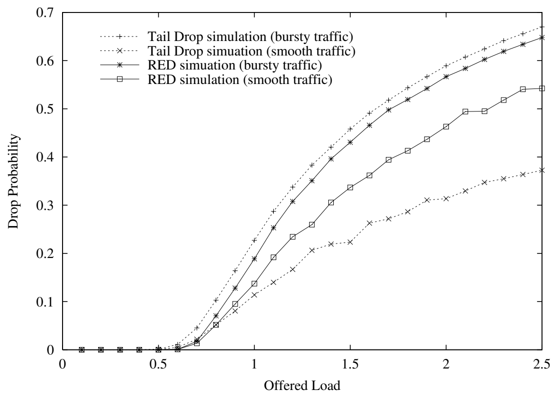

Fig. 7. Drop probability for RED and Tail Drop vs. offered load for bursty (batch arrivals and Pareto distributed interarrivals) and smooth (Poisson) traffic, and a high fraction of bursty traffic (90%).

as the synchronization of multiple TCP connections, has been observed in simulation [20], however it is hard to observe in the operational Internet.

In any case, one claim made by the RED designers is that, since RED spreads out packet drops, it will help break the synchronization pattern which (is thought to) occurs with Tail Drop. To investigate this claim, we examine in this section the impact of RED on the distribution of the number of consecutive packet losses in a loss event at a router. We consider the same model as before, except that the traffic is now simply a Poisson process of intensity λ, so that the offered load is equal to $\rho = \lambda/\mu$.

A. Tail Drop

Assume that a drop occurs at time $t = 0$ in a Tail Drop router. Since the exponential distribution is memoryless, the next incoming packet is dropped if and only if its arrival time is smaller than the service time of a packet. Thus when a packet is dropped, the next packet is dropped with probability p, where

$$p = \int_0^\infty \mu(1 - e^{-\lambda x})e^{-\mu x}\, dx = \frac{\lambda}{\lambda + \mu}.$$

As a result, the number of consecutive drops in a Tail Drop router N_{TD} satisfies

$$\forall n \geq 0, \quad P(N_{\mathrm{TD}} > n) = p^n.$$

Using the expression

$$p = \frac{\rho}{\rho + 1},$$

we conclude that the mean and the variance of the number of consecutive drops in a Tail Drop router are respectively given by

$$E(N_{\mathrm{TD}}) = \rho + 1 \quad \text{and} \quad \mathrm{var}(N_{\mathrm{TD}}) = \rho(\rho + 1). \qquad (3)$$

B. RED with instantaneous queue size

As in Section II, we first consider the case where the drop rate $d(k)$ depends on the instantaneous queue size k, and we let $\max_{\mathrm{th}} = K$ in this case. We use the following approximation:

Approximation 2: Consecutively dropped packets are dropped with the same probability.

Note that in reality when a packet is dropped with probability $d(k)$, the next packet is dropped with probability $d(l)$, where $l \leq k$ depends on the number of packets served between both arrivals. Thus the approximation above provides an upper bound on the number of consecutive drops. Furthermore, it is accurate (i.e. the difference $d(k) - d(l)$ is small) when the drop function is sufficiently smooth (namely for small values of $\min_{\mathrm{th}}$ and high values of $\max_{\mathrm{p}}$) and the offered load is high. Denoting by $\pi(\cdot|\mathrm{drop})$ the stationary distribution of the number of packets in the queue, conditionally to the fact that a drop occured, and assuming that $\pi(K|\mathrm{drop})$ is negligeable, the number of consecutive drops in a RED router N_{RED} satisfies

$$\forall n \geq 0, \quad P(N_{\mathrm{RED}} > n) = \sum_{k=0}^{K-1} \pi(k|\mathrm{drop})\, d(k)^n.$$

By Bayes' formula,

$$\pi(k|\mathrm{drop}) = \frac{\pi(k)}{P(\mathrm{drop})}\, d(k).$$

We conclude that

$$\forall n \geq 0, \quad P(N_{\mathrm{RED}} > n) = \frac{\displaystyle\sum_{k=0}^{K-1} \pi(k) d(k)^{n+1}}{\displaystyle\sum_{k=0}^{K-1} \pi(k) d(k)}. \qquad (4)$$

Figure 8 compares the analytic result above with simulation for an offered load of $\rho = 2$ and RED parameters as in Example 1. We observe a very good fit.

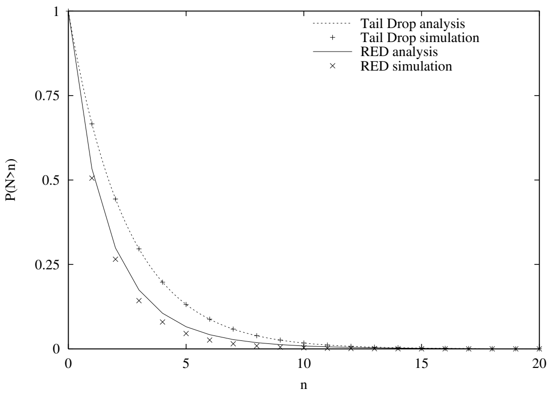

Fig. 8. Distribution of the number of consecutive drops for an offered load of $\rho = 2$

Equation (4) allows us in particular to evaluate the mean and the variance of the number of consecutive drops in a RED router. We obtain

$$E(N_{\mathrm{RED}}) = 1 + \frac{\displaystyle\sum_{k=0}^{K-1} \pi(k) \frac{d(k)^2}{1 - d(k)}}{\displaystyle\sum_{k=0}^{K-1} \pi(k) d(k)},$$

and

$$\mathrm{var}(N_{\mathrm{RED}}) = \frac{\sum_{k=0}^{K-1} \pi(k) \left(\frac{d(k)}{1-d(k)} \right)^2}{\sum_{k=0}^{K-1} \pi(k) d(k)}.$$

Table I shows the results obtained for a Tail Drop router and a RED router, when the offered load is $\rho = 2$. We conclude that RED effectively spreads out packet losses, and thus may avoid the synchronization of TCP flows.

	mean	variance
Tail Drop	3	6
RED	2.3	4.1

TABLE I
MEAN AND VARIANCE OF THE NUMBER OF CONSECUTIVE DROPS FOR AN OFFERED LOAD OF $\rho = 2$

C. RED with average queue size

As mentionned earlier, the model becomes much more complex when RED uses the average queue size instead of the instantaneous queue size to compute the drop probability of a packet. But, here again, the key observation is that when the parameter w is small, the estimated average queue size $\hat{k}$ varies slowly, so that consecutive packets are likely to experiment the same drop probability $d(\hat{k})$. As a result, Approximation 2 is still valid in this case. In fact, it is acurate whatever the drop function, and in particular when $\max_{\mathrm{th}} < K$, provided that the offered load is high. It follows then from (4) that the distribution of the number of consecutive drops satisfies

$$\forall n \geq 0, \quad P(N_{\mathrm{RED}} > n) \geq \frac{\sum_{k=\max_{\mathrm{th}}}^{K-1} \pi(k)}{\sum_{k=0}^{K-1} \pi(k) d(k)} > 0.$$

Hence, when the parameter w tends to 0, the number of consecutive drops becomes infinite with a positive probability ! The interpretation of this result is that, under high load, the average queue size slowly oscillates around the value $\max_{\mathrm{th}}$, resulting in long (infinite when w tends to 0) periods of consecutive drops (when $\hat{k} > \max_{\mathrm{th}}$), and long (infinite when w tends to 0) periods of random drops (when $\hat{k} < \max_{\mathrm{th}}$). This is illustrated by the simulation results of Figure 9 and Table II, obtained for an offered load of $\rho = 2$ and the same RED parameters as those of Example 2 (except that w takes the values 0.1, 0.01 and 0.001). The results show that RED dramatically increases the mean number of consecutive drops as well as the variance of the number of consecutive drops, in particular when w is close to its recommended value 0.002 [8]. This means that deploying RED might in fact contribute to the synchronization of TCP flows, which is exactly the opposite of one of its initial objectives.

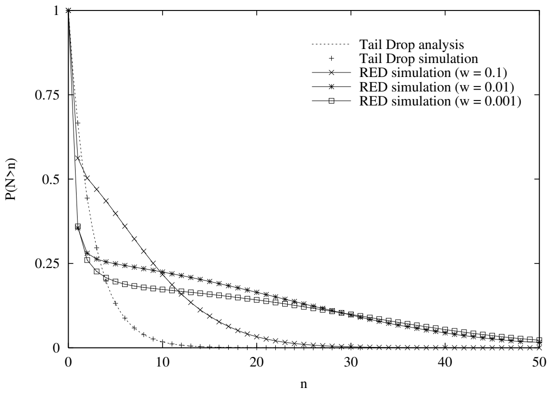

Fig. 9. Distribution of the number of consecutive drops for an offered load of $\rho = 2$

	mean	variance
Tail Drop	3.0	6.0
RED $w = 0.1$	5.9	40
RED $w = 0.01$	7.7	170
RED $w = 0.001$	7.2	190

TABLE II
MEAN AND VARIANCE OF THE NUMBER OF CONSECUTIVE DROPS FOR AN OFFERED LOAD OF $\rho = 2$

IV. QUEUEING DELAY

We next compare the delay through a router with both the RED and Tail Drop management schemes. We use the same model as in previous section, where the input traffic is a Poisson process of intensity λ, to evaluate the queueing delay (equivalently the queue size) in the router.

A. Tail Drop

The stationary distribution of the queue size in a Tail Drop router is simply given by

$$\forall k = 0, \ldots, K, \quad \pi_{\mathrm{TD}}(k) = \frac{\rho^k (1-\rho)}{1-\rho^{K+1}}.$$

B. RED with instantaneous queue size

As we did earlier, we assume here that the drop rate $d(k)$ depends on the instantaneous queue size k, and we let $\max_{\mathrm{th}} = K$. Then the number of packets in the queue is a birth-death process, the stationary distribution of which is simply given by

$$\forall k = 0, \ldots, K, \; \pi_{\mathrm{RED}}(k) = \frac{\rho^k \prod_{l=0}^{k-1} (1-d(l))}{\sum_{k=0}^{K} \rho^k \prod_{l=0}^{k-1} (1-d(l))}. \quad (5)$$

As illustrated by Figure 10 and Table III (for an offered load of $\rho = 2$ and the RED parameters of Example 1), RED reduces the mean delay, but increases the delay variance significantly.

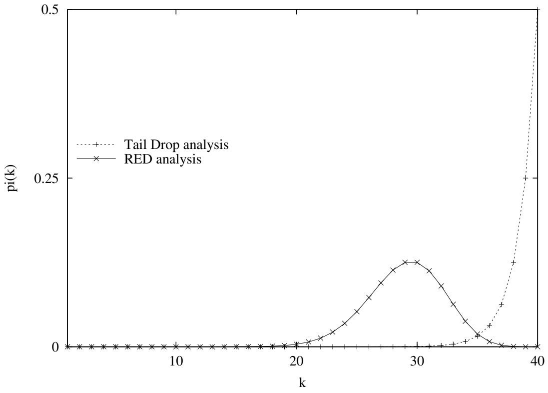

Fig. 10. Distribution of the queue size for an offered load of $\rho = 2$

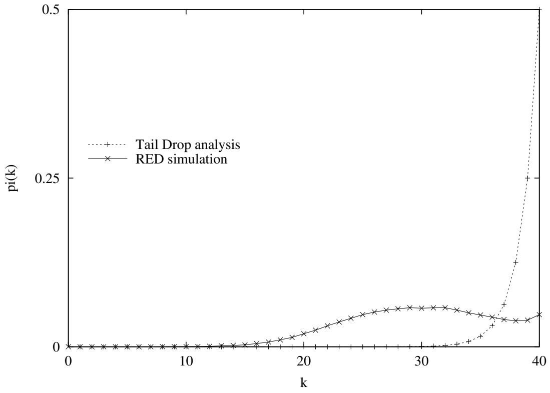

Fig. 11. Distribution of the queue size for an offered load of $\rho = 2$

	mean	variance
Tail Drop	39.0	2.0
RED	29.0	10.0

TABLE III

MEAN AND VARIANCE OF THE QUEUE SIZE FOR AN OFFERED LOAD OF $\rho = 2$

Consider then the case of a UDP-based IP telephony application, which sends smooth traffic (typically on/off periodic traffic when silence detection is used). We saw in Section II that that application would loose many more packets with RED than with Tail Drop. We see here that the average delay suffered by the UDP packets would be much lower than with Tail Drop (depending on the choice of max_{th}), which is a key benefit in telephony applications. However, the delay variance (computed from Equation 5) is such that the end to end delay, including the playout delay at the destination, does not reflect the gain RED brought to the mean delay. We can then expect the audio quality perceived at the destination to be mediocre at best.

C. RED with average queue size

Consider now the case when the drop rate computed by RED is a function of the average queue size. As mentionned earlier, provided that the parameter w is small and the offered load is high, the estimated average queue size (and hence the stationary instantaneous queue size) will slowly oscillate around the value $\mathrm{max}_{\mathrm{th}}$. Thus, although RED reduces the mean delay, RED also adds jitter in the delay, and so as much as the parameter w is small. This is illustrated in Figure 11 and Table IV, for an offered load of $\rho = 2$ and the RED parameters of Example 2.

	mean	variance
Tail Drop	39.0	2.0
RED	29.9	38.7

TABLE IV

MEAN AND VARIANCE OF THE QUEUE SIZE FOR AN OFFERED LOAD OF $\rho = 2$

V. SIMULATION

In Sections II and III, we derived analytic expressions of various measures of interest to evaluate RED. While the analytic approach is important to quantify relationships between parameters and performance measures, it must be complemented with simulation or experiments to validate the hypotheses made during the analysis, and to explore phenomena not amenable to tractable analysis.

In this section, we focus on simulation results. We obtained these with *ns* [9].

A. Validating the analytic results

In a first set of simulations, we verify that the main conclusions of our analysis are valid. We use a simple network setup with many sources s_i send TCP and UDP traffic to destinations d_i via a RED or Tail Drop router, as shown in Figure 12.

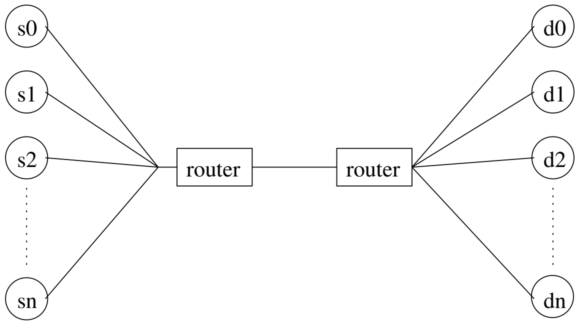

Fig. 12. Network topology for the simulation studies

The TCP sources use the NewReno algorithm, the UDP sources send CBR traffic. We use different propagation delays for the links between the sources and the router, so as to have a range of round trip delays. In practice, the round trip delays vary between 120ms and 220ms. In the first router we choose the buffer management scheme to be RED or Tail Drop. In our simulations, we have over 100 TCP connections sending packets from the sources to the destinations. We also have UDP connections sending at a constant rate which, summed over all UDP sources, equals 10% of the bottleneck link speed. The bottleneck in our setup is the link between the two routers, with a bandwidth of 10Mb/s. We measure in the first router the drop rates of both UDP and TCP traffic, and the delay of UDP packets; we also compute the total goodput of the TCP connections.

We have already compared earlier in the paper our analytic model with simulations that did not make assumptions (such as Assumption 1 and Assumption 2) used in the models, and we did observe good correlation. We now investigate how well our model, which models TCP connections as a bursty open-loop traffic source, ties in with reality, or at least with our simulation setup, in which TCP connections are closed-loop rate controlled connections. To do so, we compare the total TCP goodput and the loss rates for TCP and UDP traffic obtained with simulation. Refer to Table V.

	UDP loss rate	TCP loss rate	Goodput
Tail Drop	0.051	0.102	5.55 (Mb/s)
RED	0.083	0.102	5.56 (Mb/s)

TABLE V
LOSS RATES AND GOODPUT FOR RED AND TAIL DROP

We observe that the loss probability for TCP (bursty) traffic does not change between RED and Tail Drop. Furthermore, the loss rate for UDP (smooth) traffic increases significantly when going from Tail Drop to RED. Both these results match those obtained with the analysis in Section II. The first result also further suggests that TCP synchronization might not happen at all in practice. We also note that, unlike what is sometimes claimed, the total TCP goodput does not increase with RED.

We now compare the delay properties derived with the model with those obtained with the simulations. Figure 13 shows the evolutions with time, as the simulation progresses, of the delay in the router with RED and Tail Drop, when the buffer size is equal to 40 packets and 150 packets, respectively. With Tail Drop, and given the high load in the router, the buffer occupancy quickly increases then remains close to its maximum value. Note that with 100 TCP flows and different round trip delays, we do not observe system-wide synchronization patterns that would indicate large scale TCP synchronization. The situation is quite different with RED. The queue builds up quickly; RED starts dropping packets when the average queue size reaches $\mathrm{min_{th}}$, then drops all packets when the average queue size reaches $\mathrm{max_{th}}$. The drop rate decreases when the average queue drops below $\mathrm{max_{th}}$, traffic picks up, the average queue tends toward $\mathrm{max_{th}}$ and eventually exceeds it, and the cycle resumes (refer to our earlier discussion in Section III). Thus, as expected, the average queue stays close to $\mathrm{max_{th}}$, and the RED router behaves essentially like a Tail Drop router with buffer size $\mathrm{max_{th}}$ [6], [15]. However the instantaneous queue size varies heavily with time, more so than a Tail Drop queue does in the same situation. Again, this shows good correlation with our analytic results.

B. Impact of the number of flows

Finally, we use simulation to examine an issue we did not consider in our analysis, namely the impact of the number of TCP flows on performance.

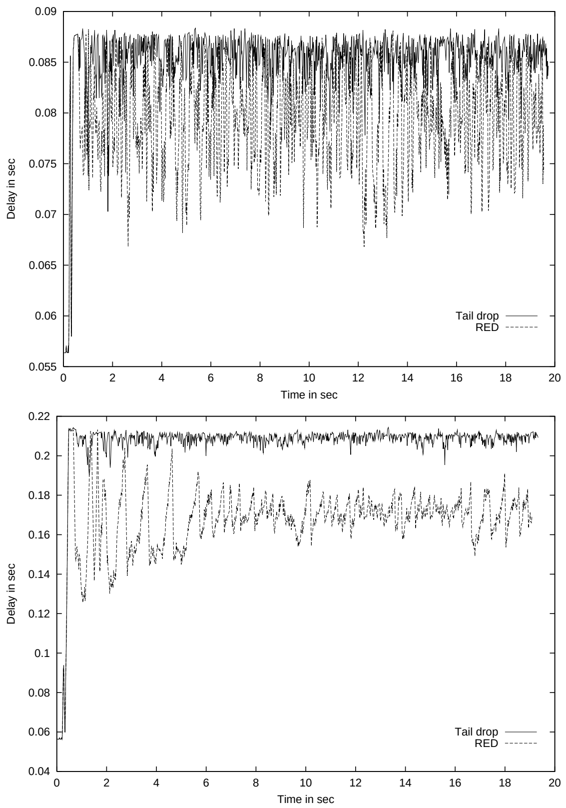

Fig. 13. Queueing delay for RED and Tail Drop with buffer size of 40 (top) and 150 packets (bottom)

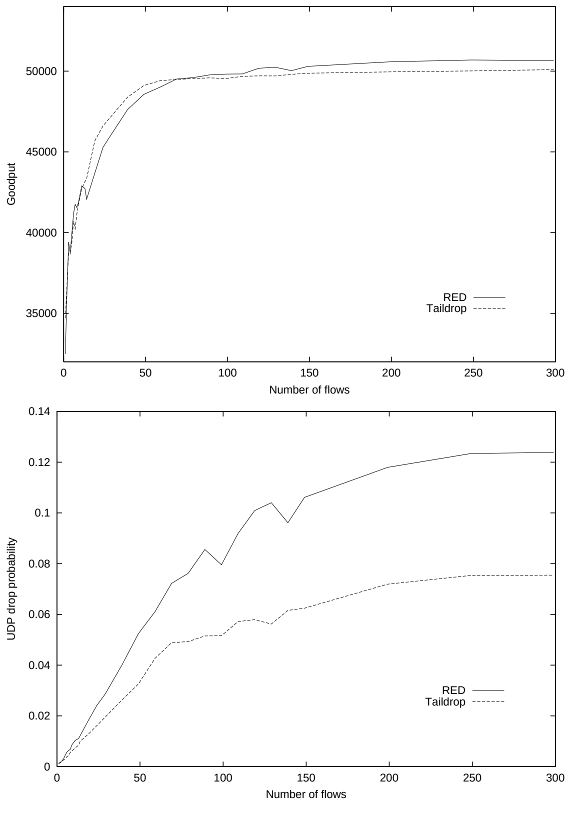

Fig. 14. Goodput of the TCP connections and UDP loss rate as a function of the number of active flows in the router for a buffer size of 40 packets

Figures 14 shows the evolutions of the total TCP goodput (top graph), and the UDP loss rate (bottom graph) as a function of the number of TCP flows. We first observe that using RED or Tail Drop does not change much the total TCP goodput, independent of the number of flows. When the number of flows is large (and therefore the load in the router is sustained and high), RED performs slightly better, but only at the cost of dropping many more UDP packets than Tail Drop would. Again, this ties in well with our analysis. We also observe that RED drops more UDP packets than Tail Drop independent of the number of flows, and that the drop rate keeps still increasing even as goodput remains steady when the number of flows increases. Furthermore, the goodput does not increase significantly when more than 75 flows are active in the network. In addition, Tail Drop performs better when only few flows are active.

We now examine how the number of flows impacts the router performance (as opposed to the end to end performance discussed above). To do this, we plot in Figures 15 the evolutions of the actual queue size during an experiment with different numbers of active flows in the network. We set up the network as described before but used a large buffer of 200 packets. For the RED router, we set $\min_{\rm th} = 50$, $\max_{\rm th} = 150$, $\max_{\rm p} = 10\%$ and the averaging parameter $w = 0.002$. When the number of flows is small the Tail Drop queue router is rarely empty, while the RED actual queue size oscillates heavily and is more often idle. When the number of flows is higher, the router is never idle for RED nor for Tail Drop. This means that we should not expect much difference in throughput for the two dropping schemes; this in turn confirms our earlier observation when we saw a larger throughput with Tail Drop than with RED. Note that, in any case, we observe a much more pronounced oscillation of the actual queue size with RED then with Tail Drop. This reflects our observations earlier in this section on large delay variance with RED.

VI. CONCLUSION

We have shown in the paper that (i) RED does eliminate the higher loss bias against bursty traffic observed with Tail Drop, but not by decreasing the loss rate of bursty traffic, rather by increasing that of non bursty traffic; (ii) the number of consecutive packet drops is higher with RED than Tail Drop, indicating that RED might contribute to, rather than solve, the global synchronization of TCP flows; (iii) the lower mean delay brought about by RED is compensated by a large delay variance for smooth traffic, which would be detrimental to interactive applications such as IP telephony.

Our results indicate that the benefits of RED are not as clear cut as claimed in [8]. Rather, they do point at a definite need to obtain a thorough analytic (quantitative) understanding of RED, together with clear operational supporting evidence, to weight the benefits that a large scale deployment of RED would bring.

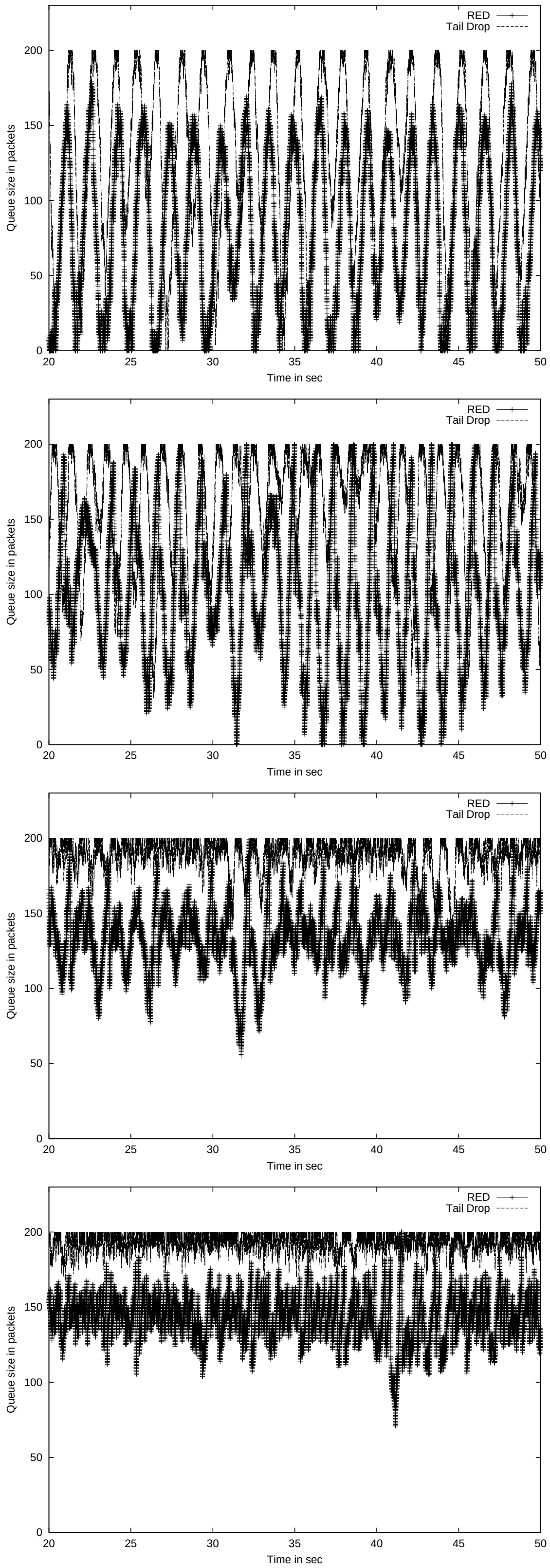

Fig. 15. Evolutions of the instantaneous queue size for RED and Tail Drop with 25, 50, 75, and 100 active TCP flows

REFERENCES

[1] P. Abry, P. Flandrin, M. Taqqu, D. Veitch, "Wavelets for the analysis, estimation, and synthesis of scaling data", in *Self-Similar Network Traffic Analysis and Performance Evaluation*, K. Park and W. Willinger (eds), 1999.

[2] F. Baccelli and P. Bremaud, *Elements of Queueing Theory*, Springer-Verlag, 1994.

[3] B. Braden et al, "Recommendations on Queue Management and Congestion Avoidance in the Internet", RFC2309, April 1998.

[4] S. Doran, Interface Graphs, http://adm.ebone.net/s̃md/red-1.html.

[5] A. Feldman, A. Gilbert, P. Huang, W. Willinger, "Dynamics of IP traffic: A study of the role of variability and the impact of control", to appear in *ACM Sigcomm'99*, Cambridge, MA, Aug. 1999.

[6] W. Feng, "BLUE: A New Class of Active Queue Management Algorithms", Department of EECS, Network Systems Department University of Michigan, 1999.

[7] W. Feng et al., "A self-configuring RED gateway", *IEEE Infocom '99*, San Francisco, CA, April 1999.

[8] S. Floyd and V. Jacobson, "Random Early Detection gateways for congestion avoidance", *IEEE/ACM Trans. on Networking*, vol. 1, pp. 397–413, 1993.

[9] S. Floyd, NS network simulator, www-mash.cs.berkeley.edu/ns/

[10] S. Floyd, K. Fall, "Promoting the use of end-to-end congestion control in the Internet", to appear in *IEEE/ACM Trans. Networking*, Aug. 1999.

[11] P. Hurley, J.Y. Le Boudec, P. Thiran, "A Note on the Fairness of Additive Increase and Multiplicative Decrease", *Proc. ITC 16*, Edinburgh, UK, June 1999.

[12] V. Jacobson, "Congestion avoidance and control", *Proc. ACM Sigcomm'88*, Stanford, CA, Aug. 1988.

[13] W. Leland, M. Taqqu, W. Willinger, and D. Wilson, "On the Self-Similar Nature of Ethernet Traffic", *IEEE/ACM Transactions on Networking*, vol. 2, no. 1, pp. 1-15, February 1994.

[14] D. Lin, R. Morris, "Dynamics of Random Early Detection", *Proc. ACM Sigcomm'97*, Cannes, France, Sept. 1997.

[15] T. J. Ott, T.V. Lakshman, L. Wong, "SRED: Stabilized RED", *Proc. IEEE Infocom'99*, San Francisco, CA, March 1999.

[16] V. Paxson, S. Floyd, "Wide area traffic: the failure of Poisson modeling", *IEEE/ACM Trans. Networking*, vol. 3, June 1995.

[17] RED page, www.aciri.org/floyd/red.html

[18] Stevens, W., "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms", RFC 2001, January 1997.

[19] D. Veitch, J. Andren, M. Hilding, "Understanding end to end Internet traffic dynamics", *IEEE Globecom'98*, Melbourne, AU, 1998.

[20] L. Zhang, D. D. Clark, "Oscillating behavior of network traffic: A case study simulation", *Internetworking: Research and Experience*, vol. 1, no. 2, pp. 101-112, Dec. 1990.