On the Efficiency and Fairness of Transmission Control Loops: A Case for Exogenous Losses *

MINA GUIRGUIS
msg@cs.bu.edu

AZER BESTAVROS
best@cs.bu.edu

IBRAHIM MATTA
matta@cs.bu.edu

Computer Science Department
Boston University
Boston, MA 02215

BUCS-TR-2003-013

Abstract

We postulate that exogenous losses–which are typically regarded as introducing undesirable "noise" that needs to be filtered out or hidden from end points–can be surprisingly beneficial. In this paper we evaluate the effects of exogenous losses on transmission control loops, focusing primarily on efficiency and convergence to fairness properties. By analytically capturing the effects of exogenous losses, we are able to characterize the transient behavior of TCP. Our numerical results suggest that "noise" resulting from exogenous losses should not be filtered out blindly, and that a careful examination of the parameter space leads to better strategies regarding the treatment of exogenous losses inside the network. Specifically, we show that while high levels of exogenous losses lead to inefficient network utilization, lower levels of losses do help connections converge to their fair share. This observation suggests that the beneficial value of exogenous losses should be leveraged inside the network in favor of TCP connections. To that end, we propose an *eXogenous-loss aware* Queue Management (XQM) that actively accounts for exogenous losses to improve efficiency and fairness. Our proposed approach is based on classifying the effects of exogenous losses into long-term and short-term effects. Such classification informs the extent to which we control exogenous losses, so as to operate in an efficient and fair region. We validate our results through simulations.

Keywords: TCP; Error Control; Control Theory; Transient Analysis; Active Queue Management; Wireless Links.

1. Introduction

Independent exogenous packet losses pose a formidable challenge to an end-to-end transmission control protocol's ability to react effectively to changes in network resource availability. By *exogenous losses* we mean losses that are produced outside the transmission control system of a traffic source, *i.e.*, independently of that source's behavior and its long-term fair share of network resources. The emergence of independent exogenous losses could be attributed to two radically different causes: the first is simply a consequence of traversing lossy channels (*e.g.*, wireless first and last hops, satellite links), whereas the second is due to the bursty nature of cross-traffic.

Motivation: Exogenous losses are problematic because they are thought of as constituting "noise" with which a transmission control loop must reckon. Unchecked, exogenous losses have the potential of decreasing resource utilization, introducing unfairness in resource allocation, and even compromising the stability of an end-to-end control loop. Recent research efforts have started to address these issues—for example, by diagnosing the cause of packet losses and reacting differently to different types of losses [3], or by slowing down the reaction to packet losses through the use of "smoother" control rules [25].

While countering the effects of exogenous losses is a worthy goal, a more important goal is to assess the extent to which these losses actually impact the behavior of control loops. More to the point, to be able to assess the usefulness of the plethora of traffic control strategies dealing with effects of exogenous losses, we need a rigorous methodology for the analysis of the emergent behaviors that result from the composition of end-to-end protocols (*e.g.*, increase-decrease rules), network element behaviors (*e.g.*, RED/AQM [6]), and new application-level functionalities. To that end, a particularly promising approach is to marshal techniques from control theory and optimization theory to the modeling and evaluation of complex network transmission control strategies, as exemplified in a number of recent efforts [21, 15, 11]. While useful, these efforts were limited to the wired part of the network and did not explicitly model

*This work was supported in part by NSF grants ANI-0095988, ANI-9986397, EIA-0202067 and ITR ANI-0205294, and by grants from Sprint Labs and Motorola Labs.

exogenous losses.

Contributions: In this paper, we capture the effect of exogenous losses by extending a dynamic model of the widely deployed Transmission Control Protocol (TCP) [13]. As one would expect, we show that high levels of exogenous losses lead to inefficient network utilization. Surprisingly though, we also show that low levels of exogenous losses introduce convergence to fairness properties that are both beneficial and desirable.

Since TCP, by its nature, adaptively seeks available bandwidth, exogenous losses in effect impose an upper limit on achievable TCP throughput. The extent to which exogenous losses limit achievable throughput makes the crucial difference between desirable and undesirable exogenous losses. In particular, if this limit lies below a connection's long-term fair share, then exogenous losses cripple that TCP connection. Otherwise, we show that exogenous losses enable the fast and stable convergence of TCP connections to their long-term fair shares of network resources. This is because such exogenous losses serve as early error notifications to the sources, which, similar to RED (Random Early Detection) [6], randomize packet drops across all connections. This randomness prevents an individual TCP connection from monopolizing the bottleneck resource, in addition to preventing several connections from synchronizing their sending behavior which may result in high delay variance (jitter). Thus, low levels of exogenous losses, which do not force TCP throughput to dip below its long-term fair share, can be beneficial in reaching an *efficient*, *stable* and *fair* allocation of resources.

This observation suggests that the common wisdom of utterly hiding all exogenous losses may indeed be counterproductive. Even if such hiding is harmless, the overhead of implementing it—for example through local error recovery over wireless access links using Snoop [2]—may not be justified.

Our analysis has also identified two vulnerabilities of RED (as well as AQM techniques that employ queue-length dependent signaling of congestion conditions) in the presence of exogenous packet losses. First, since RED generates its congestion notification signals as a function of the *average*, rather than instantaneous, queue size, it may fail to react quickly to the behavior of TCP sources responding to fast-changing exogenous losses. This could lead to further inefficiencies until the average queue catches up with the new value imposed by exogenous losses. Second, with high levels of exogenous losses, the queue length at the bottleneck router may decrease below the minimum threshold of RED. In this case, RED does not drop (or mark) packets and the system degenerates into an open-loop control system.

Towards a constructive application of our findings, we propose an Active Queue Management (AQM) approach, which intelligently accounts for exogenous losses. The goal of such an approach is two-fold: (1) To hide from TCP senders the "harmful" level of losses so as they converge toward their long-term fair shares; and (2) To expose the "beneficial" level of losses that create enough randomness for improved convergence and stability, while removing the inherent bias of TCP against connections with long round-trip time (RTT). While implementation details of such AQM, which we term *eXogenous loss aware Queue Management* (XQM), is beyond the scope of this paper, XQM outlines a common framework that explicitly considers and exploits exogenous losses.

Paper Outline: The rest of the paper is organized as follows. In Section 2, we present and validate through ns-2 [5] simulations a dynamic model of TCP that incorporates exogenous losses. In Section 3, we analytically derive a lower bound on losses that need to be hidden from TCP sources to ensure efficient operation. In Section 4, we classify the effects of exogenous losses into short-term and long-term effects, and show that hiding short-term exogenous losses improves the transient behavior of TCP connections. We capitalize on this finding in Section 5, where we outline and evaluate the performance of XQM. We present relevant related work in Section 6 and also throughout the paper, when appropriate. We conclude in Section 7 with a summary of results and follow-up research.

2. Modeling TCP + Exogenous Losses

A transmission control loop is usually divided into two components: The *forward control path*, which governs how much data the sender can inject into the network, and the *feedback path*, via which the network (*e.g.*, AQM at the bottleneck) informs the sender of congestion or available bandwidth. Such feedback is always associated with a delay known as the *feedback delay*, which is the time it takes the feedback signal to reach the sender.[1]

In this section, we extend an analytical fluid model similar to that proposed in [11, 16, 21, 27] to capture the effect of exogenous losses on closed-loop TCP control loops. We validate our model using ns-2 [5] simulations.

2.1. Model Derivation

We consider a dynamic fluid model of m TCP connections traversing a single bottleneck of capacity C. The round trip time $r_i(t)$ at time t for connection i is equal to the round-trip

[1] Notice that not every transmission protocol has those two components; TCP [13] is an example of a closed-loop protocol where both components are present, whereas UDP [24] is an example of an open-loop protocol with no feedback component.

propagation delay D_i between the sender and the receiver for connection i, plus the queuing delay at the bottleneck router. Thus $r_i(t)$ can be expressed by

$$r_i(t) \quad = \quad D_i + \frac{b(t)}{C} \tag{1}$$

where $b(t)$ is the backlog buffer size at time t at the bottleneck router. We denote the propagation delay from sender i to the bottleneck by D_{s_ib}, which is a fraction α_i of the total propagation delay.

$$D_{s_ib} \quad = \quad \alpha_i D_i \tag{2}$$

The backlog buffer $b(t)$ evolves according to the equation

$$\dot{b}(t) \quad = \quad \sum_{i=1}^{m} x_i(t - D_{s_ib}) - C \tag{3}$$

which is equal to the input rate $x_i(.)$ from the m connections minus the output link rate. Notice that the input rates are delayed by the propagation delay from the senders to the bottleneck D_{s_ib}.

We assume that the links between the bottleneck and the receivers are subjected to exogenous packet losses, and that all connections see the same levels of exogenous losses. It follows that the total packet loss probability $q(t)$ observed by senders would comprise the congestion-induced loss probability $p_c(t)$ (due to buffer overflow at the bottleneck) as well as the exogenous loss probability $p_e(t)$. Thus, the total loss probability seen by senders is given by

$$\begin{aligned} q(t) &= 1 - (1 - p_c(t))(1 - p_e(t)) \\ &\approx min(p_c(t) + p_e(t), 1) \end{aligned} \tag{4}$$

where the congestion loss probability $p_c(t)$ depends on our choice of a queue management implementation at the bottleneck router.

For DropTail, $p_c(t)$ is simply given by

$$p_c(t) \quad = \quad \begin{cases} 0 & b(t) < B \\ 1 & b(t) = B \end{cases} \tag{5}$$

where B is the maximum buffer size.[2]

For RED [6], the congestion loss probability $p_c(t)$ is given by[3]

$$p_c(t) \quad = \quad \begin{cases} 0 & v(t) \leq B_{min} \\ \sigma(v(t) - \varsigma) & B_{min} < v(t) < B_{max} \\ 1 & v(t) \geq B_{max} \end{cases} \tag{6}$$

where σ and ς are the RED parameters given by $\frac{P_{max}}{B_{max} - B_{min}}$ and B_{min}, respectively, and $v(t)$ is the average queue size, which evolves according to the equation:

$$\dot{v}(t) \quad = \quad -\beta C(v(t) - b(t)), \qquad 0 < \beta < 1 \tag{7}$$

Notice that in the above relationship, we multiply β by C since RED updates the average queue length at every packet arrival, whereas our model is a fluid model [11, 21].

The throughput of TCP, $x_i(t)$ is given by

$$x_i(t) \quad = \quad \frac{w_i(t)}{r_i(t)} \tag{8}$$

where $w_i(t)$ is the size of the TCP congestion window for sender i.

According to the TCP Additive-Increase Multiplicative-Decrease (AIMD) algorithm, the dynamics of TCP throughput for each of the m connections can be described by the following differential equations

$$\begin{aligned} \dot{x}_i(t) &= \frac{x_i(t - r_i(t))}{r_i^2(t) x_i(t)}(1 - q(t - D_{bs_i}(t))) - \\ &\quad \frac{x_i(t) x_i(t - r_i(t))}{2}(q(t - D_{bs_i}(t))) \\ i &= 1, 2, .., m \end{aligned} \tag{9}$$

The first term represents the additive increase rule, whereas the second term represents the multiplicative decrease rule. Both sides are multiplied by the rate of the acknowledgments coming back due to the last window of packets $x_i(t - r_i(t))$. In the above equations, the time delay between the bottleneck and sender i, passing through the receiver i, is given by

$$D_{bs_i}(t) \quad = \quad r_i(t) - D_{s_ib} \tag{10}$$

2.2. Model Assumptions

The fluid model we presented above makes the following assumptions, some of which we have already mentioned.

(1) All connections with the same round-trip time (RTT) are synchronized in their feedback from the network.

(2) The level of exogenous losses experienced by all connections with the same RTT is identical.

(3) All losses (exogenous and congestion) are observed after the same feedback delay $D_{bs_i}(t)$.

(4) The effect of slow start and timeouts is ignored in the TCP equations, focusing only on the AIMD mechanism.

[2] We assume that when operating in a certain regime at time t, *e.g.*, when $b(t) < B$, the probability that the queue is full is small enough that the queue length is practically less than B over all sample paths. This assumption is validated by the ns-2 simulations presented later in this section.

[3] For simplicity, we follow the same assumptions of other studies by ignoring the uniformization of packet drops [6]. This assumption is relaxed in our ns-2 simulations.

As we will discuss later, some of the above assumptions do indeed hold in special settings (*e.g.*, when exogenous losses are due to wireless first/last-hop losses). However, in general, the above assumptions may not hold and will need to be relaxed—which we could do in simulation experiments, but not in analysis. For example, since exogenous loss levels and patterns depend on the traversed links, assumption (2) may not hold. Also, since exogenous losses may be produced on any point along a path (let alone at the congestion point), assumption (3) may not hold. Be that as it may, the analytical model captures the essential dynamics necessary to gain valuable insights, which could be confirmed using more empirical means.

2.3. Model Application

Low $et\ al.$ [21] studied the dynamics of TCP over RED queues through linearization around equilibrium points.[4] While useful, linearization fails to track the system trajectories across different regions dictated by the non-linear equations.

In Appendix 7, we show the linearization of the system (TCP + exogenous losses) modeled above. In particular, we show how such a system switches between an open loop control, when exogenous losses are high, and a closed loop control, otherwise. This switching between operating regions prevents us from using traditional transient control analysis. Thus, in the remainder of this section, we solve the above set of non-linear equations numerically for a careful and continuous tracking of the model's behavior through different operating regions.

It is important to note that the numerical solution of the non-linear fluid model (equations (1) through (10)) readily provides the *average* performance of the system, unlike simulation which requires the averaging of many independent runs.

2.4. Model Validation

To validate the predictive value of our analytical model, we compare results obtained from solving the set of equations (1) through (10) with those obtained from ns-2 simulations, with similar parameterization. Figure 1 depicts the topology under consideration. We set the total number of competing connections to 20; we set the capacity C to 2,000 $\frac{pkts}{sec}$; and we chose the propagation delay of all connections uniformly at random between 80 and 120 msec. Each connection's fair share of the link is around 100 $\frac{pkts}{sec}$. The total buffer size at the bottleneck is chosen to be 250 packets. RED's minimum and maximum buffer thresholds are set to

[4]Linearization assumes (and hence requires) that the system always stays within a certain operating regime.

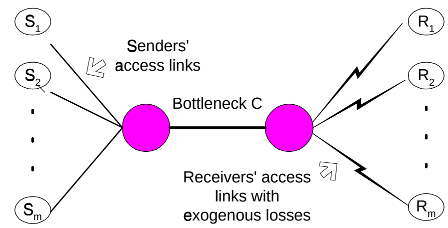


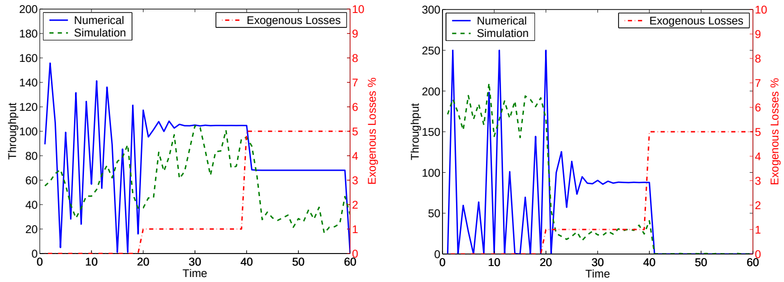
Figure 1: Dumbbell topology used in numerical evaluation and in ns-2 simulations.

50 and 120 packets, respectively. The weight parameter β was set to 0.0001 and P_{max} was set to 0.1. We also assume that the bottleneck is in the middle of the path between the senders and the receivers, *i.e.*, α in equation (2) is chosen to be 0.25.

In our ns-2 [5] simulations (parameterized as above), we assume that all sources start sending at the same time with unlimited data to send. The packet size is chosen to be 1,000 bytes. Loss modules, generating exogenous losses, are attached to every access link between the bottleneck and each receiver, hence relaxing assumption (3) of our analytical model. During the time period [0, 20) we introduce 0% exogenous losses, during [20, 40) the rate of exogenous losses is increased to 1% and finally during [40, 60], exogenous losses are increased further to 5%. The simulations results are averaged over ten independent runs.

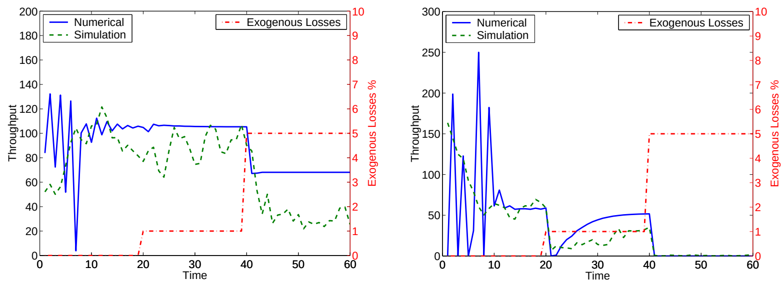
Figure 2 shows that the throughput and the queue size trend predicted by our analytical model closely matches those obtained via ns-2 simulations, under both DropTail (top row) and RED (bottom row).

In the first twenty seconds, *i.e.*, under zero exogenous losses, TCP throughput oscillates between low and high sending rates for DropTail. While RED sustains these oscillations only until the average queue size reaches its steady state value (around time 10). In the next twenty seconds, when the level of exogenous losses increases to 1%, TCP throughput converges to its fair share under both DropTail and RED. Notice how the queue size converges to a steady state (non-zero) value, hence the system is well utilized. In the last twenty seconds, exogenous losses (now increased to 5%) result in the convergence of each $x_i(t)$, albeit to a value lower than the fair share and the queue size drops to zero, hence the system is under utilized. This observation suggests that low levels of exogenous losses do not degrade the throughput of TCP. But clearly, when exogenous loss rates are increased to 5%, TCP's throughput suffers and the system becomes under utilized (below the fair share of 100 $\frac{pkts}{sec}$). Notice that while the results obtained via simulation does not match the numerical results of our model perfectly (due to the assumption in the fluid model, which is not preserved in simulations), the trends are in good agreement.



(a) Throughput (b) Queue Size

DropTail



(c) Throughput (d) Queue Size

RED

Figure 2: DropTail (top row with throughput on the right and queue size on the left) and RED (bottom row with throughput on the right and queue size on the left). The level of exogenous losses is increased from 0% to 1% at time 20 and further to 5% at time 40.

3. Impact on Efficiency

Figures 2(b) and 2(d) suggest that low levels of exogenous losses do not degrade the efficiency since the queue converges to a non zero steady state value. But clearly, when exogenous loss rates are increased to 5%, TCP's throughput suffers and the system becomes underutilized (queue size goes to zero). In this section we focus on that dimension of the impact of exogenous losses.

A transmission control loop is said to be *efficient* if the TCP throughput for that loop matches the bottleneck link capacity. Thus, at steady state, the following two equations should be satisfied for an efficient network utilization. These equations are obtained by setting the derivatives to zero in equations (3) and (9).

$$\sum_{i=1}^{m} \hat{x}_i = C \tag{11}$$

$$\hat{x}_i = \frac{1}{\hat{r}_i}\sqrt{2\left(\frac{1}{\hat{q}_i} - 1\right)} \tag{12}$$

Clearly, the steady-state TCP throughput $\hat{x}_i$ is inversely proportional to the square root of the total loss probability $\hat{q}_i$, which in turn is directly affected by the exogenous loss rate $\hat{p}_e$.[5] For a steady state behavior, $\hat{q}_i$ must be larger than zero. Having no drops remove the upper limit on the rate/window and this, in theory, will cause it to grow indefinitely.

As the steady-state value of $\hat{p}_e$ increases, the sending rate would start to decrease, approaching zero. This could prevent TCP throughput $\sum_{i=1}^{m} \hat{x}_i$ from reaching C, i.e. equation (11) cannot be satisfied. The value of $\sum_{i=1}^{m} \hat{x}_i$ being less than C means that the system is underutilized. Hence TCP is forced to operate with no buffering at the bottleneck, and no congestion signals going back to senders. When this happens, the TCP transmission control loop is actually broken—it operates as an *open-loop control system* with no feedback from routers.

When exogenous losses are not present, nothing hinders the increase of TCP throughput so as to match its bandwidth share.[6] Once the connection hits its bandwidth share, packets start to accumulate until $b(t)$ reaches B under DropTail, or the average queue size starts building up until it exceeds B_{min} under RED. At that time, congestion signals

[5] Observe that equation (12) resembles the so-called TCP-friendly equation [9], except that in the model of Section 2, $\hat{q}_i$ is not necessarily a Bernoulli probability, but depends on queue management parameters.

[6] The connection is still limited by its round-trip time, but eventually will hit its bandwidth share. We assume that connections are not limited by the advertised receiver's window.

are generated and the sender would back off and this cycle repeats (cf. equation (9)).

The presence of exogenous losses impose an upper limit on TCP's throughput and it is crucial *where* this upper limit lies. If this upper limit is close to the connection's long-term fair share, then these exogenous losses turn out to improve the connection's convergence to its fair share. This is exactly what happens in the time period [20, 40) in Figure 2. Without such exogenous losses [0, 20), the connection's throughput shows large oscillations.

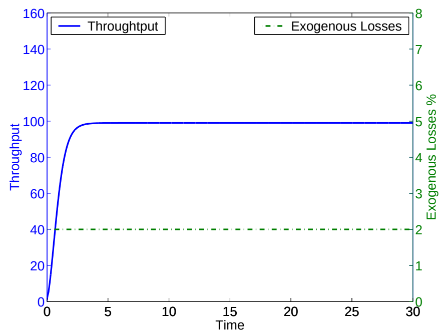


Figure 3: Efficient convergence with 2% exogenous losses

Consider the same setup described in the previous section, except that all connections have the exact same propagation delay of 100 msec. If the goal is to allocate an equal share of the bandwidth to each TCP connection, then using equation (11), we have each connection's (long-term) fair share, $\hat{x_i}$, equals 100 $\frac{pkts}{sec}$. Solving for $\hat{q}_i$ using equation (12), we get a value of almost 2%. Indeed, Figure 3 shows that when exogenous losses are at 2%, the throughput converges to the fair share value, with *no* oscillations.

Many protocols have been developed for hiding *all* exogenous losses from the sender [2, 1]. For example, in Snoop [2] the connection between the server and the client is intercepted by the proxy in the middle. The proxy buffers data packets to allow link-layer retransmission when duplicate acknowledgments, indicating packets lost over the wireless proxy-client link, arrive at the proxy. Snoop does not allow such duplicate acknowledgments to pass back to the sender to prevent it from doing fast retransmit and recovery (*i.e.*, halving its sending rate).

While such protocols attempt to improve efficiency by removing the upper limit imposed on throughput by exogenous losses, they could be hindering the convergence to fairness! Furthermore, hiding further packet losses from connections that are already getting their fair share would not be beneficial, but would only add the overhead of complete hiding (*e.g.*, the cost of buffering and local retransmission at the Snoop proxy).

Ideally, we would like to always report a value of $\hat{q}_i$ to sender i, since this would mean that the network is utilized efficiently while at the same time, connections have a fair chance to compete. In the next section, we address the challenges behind tuning (adjusting) exogenous losses inside the network to achieve this goal.

4. Active Tuning of Exogenous Losses

As discussed in the previous section, for given bottleneck link and RTT characteristics, there exists a desirable value for the loss rate that would promote both convergence and efficiency of a TCP control loop. We use the term *quiescent loss rate* to refer to this desirable value. For example, a quiescent loss rate of 2% yielded both efficiency and convergence for the experimental setting used in Figure 3. In this section we examine the pros and cons of various active approaches for relaying such a quiescent loss rate to senders.

Exogenous Loss Unaware Signaling: The traditional approach ignores the presence of exogenous losses and imposes a loss value that would improve some local metric (e.g; buffer size). An example of such an approach is RED (or some other variants thereof, e.g. [12, 10]), whereby dropping or marking of packets is conditioned on the local queue occupancy in order to stabilize the queue. Therefore, a natural question to ask is whether tuning the average drop rate of such queue management techniques over a long time scale (to match the quiescent loss rate) would yield the desirable efficient convergence to fair share. The results in Figure 2(d) suggest that this would be the case, but only over long time scales. As evident from the results in Figure 2(d), RED exhibits transient inefficiencies when faced with variability in exogenous loss rates over short time scales. Specifically, at time 20, RED's queue size drops to zero, before converging again to the new steady-state value around 50. This transient anomaly does not occur under DropTail.

The undesirable transient behavior exhibited under RED is due to the unawareness of RED of the presence of exogenous losses coupled with the time it takes for the average queue size to adapt to a new value. When the exogenous loss rate suddenly increases while the queue length is above B_{min}, TCP backs off in response to the (exogenous) losses and hence the queue starts to drain. The problem is that RED uses the *average* queue length as indication of congestion, and as long as the average is lagging behind the actual queue length, RED keeps generating congestion signals (by dropping or marking packets) for the stale higher value of its average queue length. Obviously, at that point, the congestion-oriented design of RED is challenged by the exogenous losses—it is no longer true that the sender reduces its rate only in response to congestion signals! As soon as the average queue size catches up with the new value below B_{min}, RED seizes to send feedback signals. At this point, TCP is in fact operating as an open loop control system and starts to increase its sending rate.

Exogenous Loss Aware Signaling: The above discussion suggests that it is crucial for any AQM to take into account the presence of exogenous losses. In particular, FRED [19] implicitly makes few steps forward toward this goal (although this was not in its design motivation to tune for exogenous losses) by protecting fragile flows, which could have been fragile (i.e. have small windows) by the presence of exogenous losses. In contrast, XQM as will be presented later in this work, makes the decision of when to introduce losses and when to hide losses and when not to interfere based on the level of exogenous losses present. In effect, XQM utilizes such external losses, toward its own feedback signal. Thus it provides the minimum interference and only when needed. If the value of exogenous losses occurring on the rest of a connection's path can be magically relayed to XQM, XQM can make sure, through adjusting its own control rules, that the sender will only see the quiescent end-to-end loss rate. However, this is hard to achieve in general, so we focus more on how can XQM relay the quiescent loss rate, based on each connection's performance.

In a real setting, the level of exogenous losses varies over time. If that level is below the quiescent rate, then it is possible to "introduce" losses to promote efficient convergence to a fair share. This could be done through randomly dropping or marking packets. If that level is higher than the quiescent rate then it would be necessary to "hide" such losses from the sender. This could be done in many ways, including link layer retransmission, forward error correction techniques, or replication over multiple paths.

More realistically, what if the level of exogenous losses fluctuates over time, exhibiting a long-term, average behavior, as well as a short-term, oscillatory (or cyclic) behavior?

With the same setup described in the previous section, Figure 4(left) shows the effect of short-term fluctuations (captured by a sinusoidal cyclic behavior) of ±2% during the interval [10,20], which is superimposed on an average, persistent (*i.e.*, long-term) level of exogenous losses of 3% during the interval [0,30]. Clearly, the higher-than-desirable long-term exogenous losses result in lower efficiency during the intervals [0,10] and [20,30], with throughput of $80\frac{pkts}{sec}$ (as opposed to the $100\frac{pkts}{sec}$ fair-share). Given such a mix of long-term and short-term effects, how could an agent (*e.g.*, a wireless proxy [2, 26] or an Internet Traffic Manager [23]) massage the exogenous losses observable by senders?

Long-Term Adjustment: As a first approximation, such an agent may attempt to bring the long-term average of observable exogenous losses to the quiescent rate. We refer to such an approach as *long-term adjustment* of exogenous losses, whereby an agent would hide losses in the amount of $p_h(t)$, allowing senders to observe loss rates of $p_r(t)$ that are equal to the difference between the real losses $p_e(t)$ and $p_h(t)$.

$$p_r(t) = max(0, p_e(t) - p_h(t)) \tag{13}$$

The equation above effectively shifts $p_e(t)$ down by a value that is equal to $p_h(t)$ in order to match a desirable quiescent loss rate.

A long-term reduction of exogenous loss rates will result in larger TCP congestion window sizes (*i.e.*, higher throughput). Thus, by appropriately setting the value of $p_h(t)$, we are able to bring connections operating in an inefficient region to an efficient one.

Figure 4(center) shows the results of applying such a policy, where the average level of exogenous losses over a long time scale (*e.g.*, calculated over the interval [0,30]) is brought down from 3% to 2%. As expected, the resulting throughput is decidedly better during the intervals [0,10] and [20,30], but the short-term variability in exogenous losses during the interval [10,20] results in wide oscillations that are clearly undesirable.

Short-Term Compensation: To tackle variations in exogenous loss rates over shorter time-scales, we may extend our policy to allow for a *dead-band controller* which filters out short-term changes in exogenous losses (by hiding and/or introducing losses) that are within a prescribed range (*e.g.*, ±1%) around the long-term average. We refer to this as *short-term compensation*.

Short-term compensation causes the sender to see smoother loss patterns (not affected by the real dynamics of exogenous losses). Let the losses $p_e(t)$ be evolving around a certain average denoted by $p_a(t)$. Under short-term compensation, the rate of exogenous losses $p_r(t)$ reported back to senders is given by:

$$p_r(t) = \begin{cases} p_e(t) - p_h(t) & \text{if } p_e(t) > p_a(t) + p_h(t) \\ p_e(t) + p_h(t) & \text{if } p_e(t) < p_a(t) - p_h(t) \\ p_a(t) & \text{otherwise} \end{cases} \tag{14}$$

The equation above reports to senders an average exogenous loss rate of $p_a(t)$, unless short-term variability in exogenous loss rates is beyond what can be hidden or introduced, in which case remnants of this variability are observable by senders.

Figure 4(right) shows the results from applying short-term compensation as well as long-term adjustment policies. Clearly this two-pronged strategy results in significant smoothing of achievable throughput during times of short-term variability in exogenous loss rates, while keeping the long-term average around the quiescent value for efficient convergence to a connection's fair share.

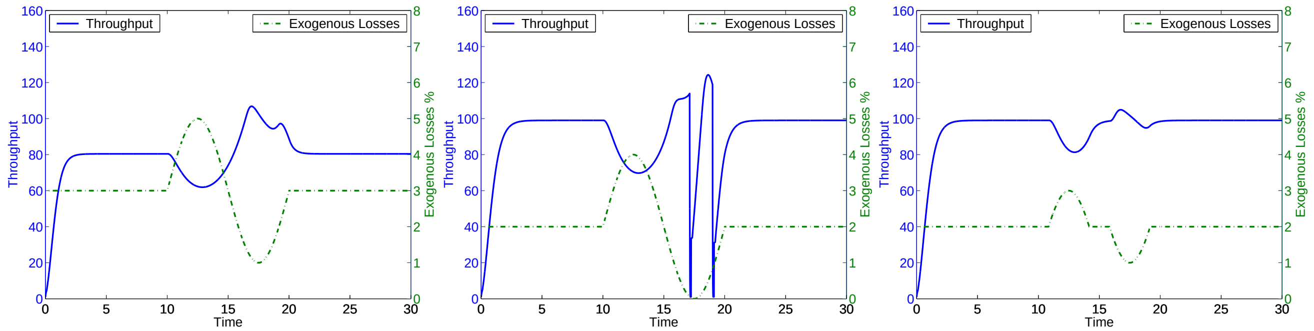


Figure 4: Effect of short-term cyclic (sinusoidal) exogenous losses on efficiency (left); effect of hiding exogenous losses over long time scales (middle); and effect of hiding exogenous losses over long and short time scales (right).

5. Leveraging Exogenous Losses

In this section, we outline and evaluate an instantiation of a network agent that leverages our findings regarding the promise of tuning exogenous losses seen by long-lived TCP connections so as to control their share of bandwidth. We term such active queue management *eXogenous-loss aware Queue Management* (XQM).

5.1. XQM

We consider a simple scenario where XQM is employed at a bottleneck router at the edge of a network cloud, over which exogenous losses can occur. We choose this particular setting because it ensures that many of the assumptions in our model are satisfied (*e.g.*, all flows experience the same levels of exogenous losses). Given this setup, and assuming both data and acknowledgments traverse the XQM-enabled access point, XQM can distinguish exogenous losses over the downstream cloud.[7] Similar to RED [6], XQM would additionally generate early error notifications through packet drops (or markings). However, unlike RED, these losses are not a function of the queue length, rather they depend on the profile of each individual TCP flow. This profile includes the round-trip time of the connection and the throughput that we would like to provide to this connection.

In order to set the quiescent loss rate $\hat{q}_i$ for a flow (or aggregate set of flows), an XQM agent needs to maintain estimates of the RTT of long-lived connections under its management, for example using "measurement-in-the-middle" techniques [14]. From equation (12), this quiescent loss rate is given by:

$$\hat{q}_i = \frac{2}{(\hat{x}_i \times \hat{r}_i)^2 + 2} \quad (15)$$

[7] Note that this setup subsumes a wireless access point (base station) at which XQM, rather than Snoop [2], is employed.

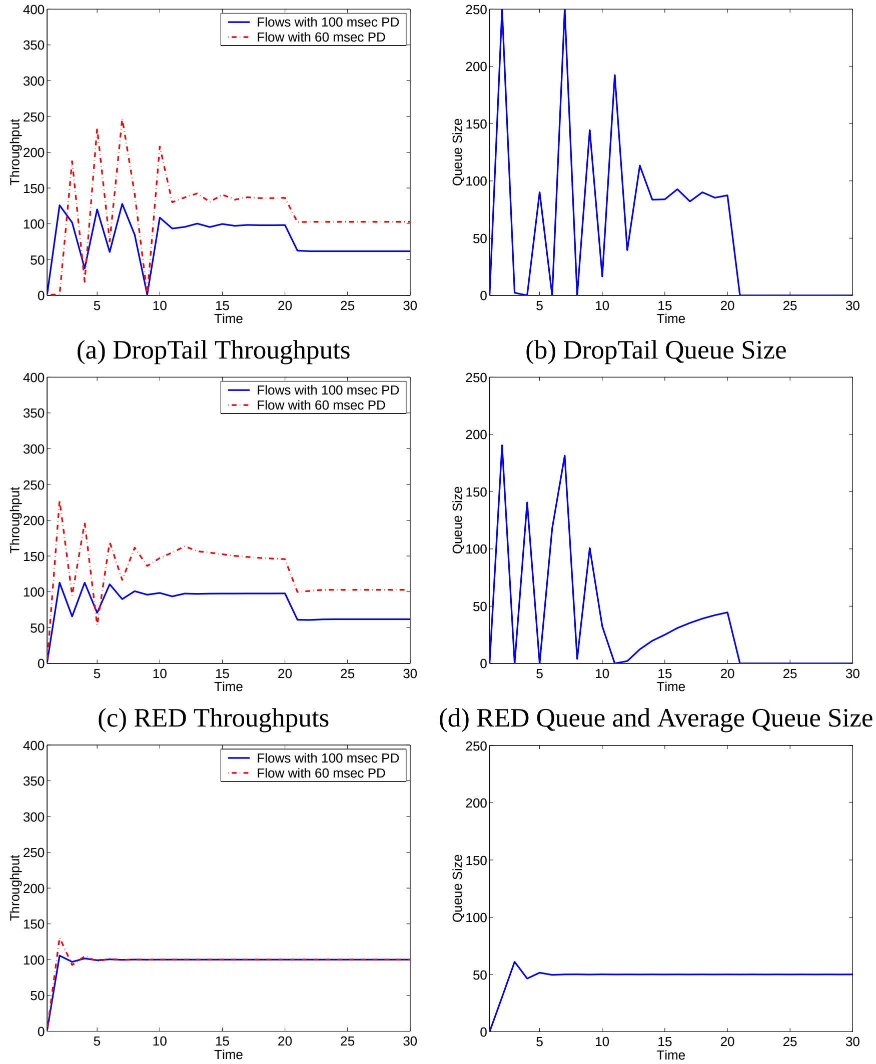
As discussed in Section 4, an XQM agent would employ long-term adjustment and short-term compensation of exogenous losses to match the value of $\hat{q}_i$ obtained from equation (15).

It is important to note that setting the quiescent loss rate using equation (15) could be modulated to allow for a differentiated/weighted allocation of the bottleneck bandwidth (*e.g.*, due to traffic engineering rules based on type of traffic). Also, the flow under consideration in equation (15) may represent an aggregate set of flows, that are identifiable by an XQM agent (*e.g.*, through source/destination IP prefixes), or an equivalent class of flows (*e.g.*, within a similar RTT range).

Implementation details of the XQM agent, including efficient techniques for flow classification and RTT measurement are beyond the scope of this paper. However, in the remainder of this section, we present XQM performance evaluation results using the model and analysis presented in this paper.

5.2. Performance Evaluation

We present numerical results to demonstrate the effect of XQM on competing long-lived TCP connections that have different round-trip times. In this setup, we have 20 connections that traverse a bottleneck of capacity 2,000 $\frac{pkts}{sec}$. 19 of these connections have the same propagation delay of 100 msec, while the remaining one has a propagation delay of 60 msec. The total buffer size at the bottleneck is chosen to be 250 packets. RED's minimum and maximum buffer thresholds are set to 50 and 120 packets, respectively. The weight parameter β was set to 0.00001 and P_{max} was set to 0.1. During the time period [0, 10), 0% exogenous losses are present. During the time period [10, 20), we introduce 1% exogenous loss rate, followed by an increase to 5% during [20, 30]. Figure 5 illustrates the system's behavior, namely the throughputs and queue length, under DropTail (first row), RED (second row), and XQM (last row).



(a) DropTail Throughputs (b) DropTail Queue Size

(c) RED Throughputs (d) RED Queue and Average Queue Size

(e) XQM Throughputs (f) XQM Queue Size

Figure 5: One connection with 60 msec delay competing with 19 other connections, each with 100 msec delay

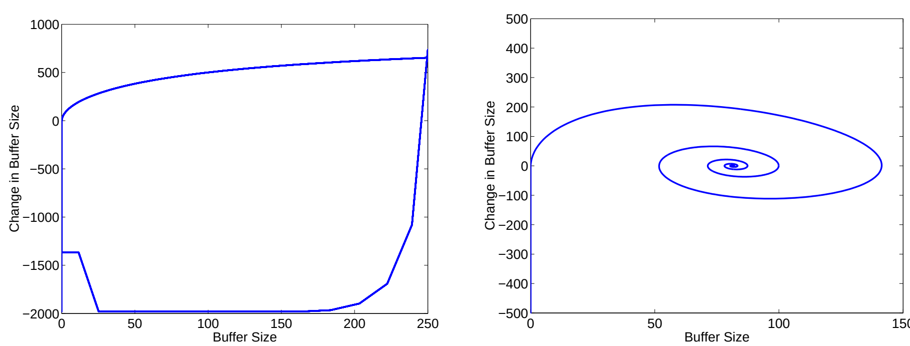


Figure 6: Buffer trajectories with no exogenous losses (left) and with 1% exogenous losses (right).

Figures 5(a) and (c) show that the presence of exogenous losses improves convergence and reduces oscillations under both DropTail and RED. However, the connections are still limited by the unfairness imposed by the different round-trip times—the shorter RTT connection (with 60 msec propagation delay) achieves better throughput over the whole run. On the other hand, by tuning exogenous loss rates to match the quiescent value, XQM (Figure 5(e)) achieves fairness among connections *regardless* of round-trip times.

Another metric of interest is jitter. Jitter is a measure of delay variability experienced by flow packets. It can be measured directly by the variance in buffer size. Figures 5(b), (d) and (f) show that the presence of exogenous losses reduces jitter, moreover XQM has much better transient response (almost zero jitter on average and full utilization).

Figure 6 shows the effect of exogenous losses on the DropTail buffer trajectory. Here we set the propagation delay of all 20 connections to 100 msec. Buffer trajectories are represented by the buffer size on the x-axis, while the y-axis represents the change in the buffer size. This kind of plots are used in control theory to visualize non-linear behaviors. Figure 6(left) shows how the buffer oscillates in a limit cycle when there are no exogenous losses. Introducing a quiescent value of 1% exogenous losses leads to convergence of the buffer trajectory to a non-zero value, where the system is efficient and stable.

6. Related Work

This paper is inspired by the need of merging two orthogonal directions that have been recently taken in the networking research community.

Control-theoretic Analysis: On one hand, marshaling techniques from control and optimization theory has been a fruitful direction [4, 11, 12, 22, 16, 17, 8, 20, 18]. In particular, studies in [21, 11] investigated the stability regions for TCP over RED. Katabi *et al.* [15] develops a controller for stable and efficient congestion control. Kelly *et al.* [16] model TCP/AQM as an optimization problem where they maximize the aggregate resource utility. However, all these techniques did not consider the effect of exogenous losses. Furthermore, the design of AQM schemes (e.g. [6, 12, 18, 7]) have been oriented toward congestion. Even those schemes which may react quickly to time-varying exogenous losses, e.g. by using more instantaneous measures of queue length or packet arrival rates, do not consider the profiles of flows in their design.

In contrast, our XQM approach is exogenous-loss aware and can thus be more effective in dealing with the heterogeneity in flow characteristics, e.g. exogenous losses they experience and their different RTTs. Furthermore, XQM goes beyond congestion avoidance by supporting fair (or weighted) allocation of resources to connections (or classes of them). To that end, XQM utilizes exogenous losses to provide efficiency and fairness to TCP flows. XQM framework leverages recent work on “measurement-in-the-middle” techniques [14] and differentiated flow services [10].

TCP over Wireless: On the other hand, TCP over wireless studies have always regarded wireless transmission errors as noise that should be completely hidden or filtered out from the senders [2, 1]. Other studies attempt to infer the reason of packet losses (e.g. congestion-induced or wireless) and investigate how the TCP sender should react to each type of loss [3]. None of this work investigated this topic from a control-theoretic approach and none advocated that “some” exogenous losses are beneficial to fairness and stability.

7. Conclusion

In this paper, we captured the effect of exogenous packet losses by extending a TCP dynamic fluid model. We showed that, while high levels of exogenous losses lead to inefficiencies, low levels may be desirable as they improve other properties such as jitter and fairness. Given that the congestion-oriented design of TCP fails to effectively deal with exogenous losses, a “hide-all-exogenous-losses” strategy would generally improve efficiency, but may unnecessarily incur overhead while forfeiting the performance gains that come from low levels of exogenous (or random) losses. In this paper, we showed that through the use of an eXogenous-loss aware Queue Management (XQM) framework, it is possible to tune exogenous losses at the connection or aggregate (class) level so as to ensure that the network operates in an efficient and fair regime. We are currently developing practical algorithms for the efficient implementation of XQM framework within the ITM [23] architecture and API. This involves the design of efficient data structures as well as accurate algorithms for round-trip estimation and flow classification. Preliminary results are promising.

Acknowledgments: We would like to thank all members of the Web and InterNetworking Group at Boston University for their feedback. In particular, we single out Liang Guo for his assistance and support with many aspects of this work.

Appendix

In this appendix, we show the effect of exogenous losses on TCP behavior, specifically, that TCP's behavior follows more than one transfer function, in accordance with the level of exogenous losses. For the sake of simplicity and ease of exposition, we derive our control-theoretic results for the case of one TCP connection ($m = 1$) traversing a RED bottleneck router.

A standard control-theoretic approach to studying a non-linear dynamical system is to linearize it around an operating point. This is done by expanding the equations using Taylor series and ignoring the high order terms. After linearization, the variables in the equations describe *perturbations* around the operating point that we have chosen. The question then becomes whether the system converges to the chosen operating point, i.e. whether perturbations in the system outputs eventually vanish after the system inputs were subjected to small perturbations. Clearly, based on our choice of the operating point, we would obtain a different set of (linear) equations.

In [21], Low $et\ al.$ assumed that the system always operates in the region of $B_{min} < v(t) < B_{max}$. In general, this is not always the case because of the oscillatory behavior of TCP; TCP would keep decreasing its window until there are no losses, which suggests that $v(t)$ could drop below B_{min}. Then, TCP would keep increasing its window until there are losses, which suggests that the system may switch between the two regions, $B_{min} < v(t) < B_{max}$ and $v(t) \leq B_{min}$. We are particularly interested in this oscillatory regime around B_{min} since the presence of exogenous losses tend to drive $v(t)$ below B_{min}. Equation (9) contains the time-varying delay lag $r(t)$, which makes linearization hard as pointed out in [21]. As an approximation, we replace $r(t)$ by its steady-state value. Substituting equation (1) in (9) and linearizing yields (recall that after linearization, variables represent perturbations):

$$\dot{w}(t) \quad = \quad \frac{\hat{q}\hat{w}}{\hat{r}} w(t) - \frac{1}{\hat{r}\hat{q}} q(t) \tag{16}$$

where $\hat{r}$, $\hat{q}$ and $\hat{w}$ are the steady-state round trip time, total marking/loss probability and window size, respectively.

Similarly, substituting equation (1) in (3) and linearizing yields:

$$\dot{b}(t) \quad = \quad \frac{C}{\hat{w}} w(t - D_{sb}) - \frac{C}{\hat{w}} b(t - D_{sb}) \tag{17}$$

If we choose the operating point for RED's average buffer size $v(t)$ to lie between B_{min} and B_{max}, from (6) and (7), we can write the following RED's linear equations:

$$\dot{v}(t) \quad = \quad -\beta C(v(t) - b(t)) \tag{18}$$

$$p_c(t) \quad = \quad \sigma v(t) \tag{19}$$

On the other hand, by choosing the operating point to be $v(t) \leq B_{min}$, from (6) we get:

$$p_c(t) \quad = \quad 0 \tag{20}$$

Equation (20) says that when $v(t) \leq B_{min}$, there is no feedback given from the network back to the sender.

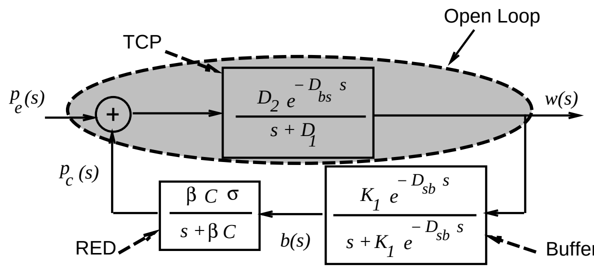


Figure 7: Linearized block diagram

Closed Loop System: Assuming the system operates in the region $B_{min} < v(t) < B_{max}$ and taking the Laplace Transforms for the above linear model (16)-(19), we get:

$$w(s) \quad = \quad \frac{D_2 e^{-D_{bs} s}}{s + D_1} (p_c(s) + p_e(s)) \tag{21}$$

$$b(s) \quad = \quad \frac{K_1 e^{-D_{sb} s}}{s + K_1 e^{-D_{sb} s}} w(s) \tag{22}$$

$$p_c(s) \quad = \quad \frac{\beta C \sigma}{s + \beta C} b(s) \tag{23}$$

where $D_1 = \frac{\hat{q}\hat{w}}{\hat{r}}$, $D_2 = \frac{-1}{\hat{q}\hat{r}}$, and $K_1 = \frac{C}{\hat{w}}$.

Hence the closed loop transfer function in the Laplace domain is given by:

$$\frac{w(s)}{p_e(s)} = \frac{D_2 (s + \beta)(s + K_1 e^{-D_{sb} s}) e^{-D_{bs} s}}{(s + D_1)(s + \beta)(s + K_1 e^{-D_{sb} s}) - D_2 K_1 \beta \sigma e^{-ds}} \tag{24}$$

Open Loop System: Observe that if the system operates instead in the region $v(t) \leq B_{min}$, then equation (20) replaces (19) and we have an open loop system whose transfer function is given by:

$$\frac{w(s)}{p_e(s)} \quad = \quad \frac{D_2 e^{-D_{bs} s}}{s + D_1} \tag{25}$$

Figure 7 illustrates the open loop and closed loop transfer functions of TCP over RED in the presence of exogenous losses p_e. One can easily see that if p_e is high enough to cause the TCP sender to shrink its window w to the point that the average RED queue size is below B_{min}, then p_c equals zero and the closed loop is broken, i.e. TCP operates as an open loop control system. Even with no exogenous losses, the system may oscillate between two different regimes, *e.g.*, the average queue length may oscillate around B_{min}. This switching between totally different behaviors, especially under time-varying exogenous losses, prevents us from applying traditional control-theoretic transient analysis techniques, thus we resort to the numerical solution of the non-linear equations.

References

[1] Ajay Bakre and B. R. Badrinath. I-TCP: Indirect TCP for Mobile Hosts. In *Proceedings of the 15th International Conference on Distributed Computing Systems*, 1995.

[2] Hari Balakrishnan, S. Seshan, and R. Kartz. Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks. *ACM Wireless Networks, 1(4)*, December 1995.

[3] Dhiman Barman and Ibrahim Matta. Effectiveness of Loss Labeling in Improving TCP Performance in Wired/Wireless Networks. In *Proceedings of ICNP'2002: The 10th IEEE International Conference on Network Protocols*, Paris, France, November 2002.

[4] T. Bu and D. Towsley. Fixed Point Approximations for TCP Behavior in an AQM Network. In *Proceedings of ACM SIGMETRICS 2001*, Boston, MA, June 2001.

[5] E. Amir et al. UCB/LBNL/VINT Network Simulator - ns (version 2). Available at http://www.isi.edu/nsnam/ns/.

[6] S. Floyd and V. Jacobson. Random Early Detection Gateways for Congestion Avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413, August 1993.

[7] Richard J. Gibbens and Frank P. Kelly. Distributed Connection Acceptance Control for a Connectionless Network. In *Proceedings of 16th International Teletraffic Congress*, pages 941–952, June 1999.

[8] R.J. Gibbens and F.P. Kelly. Resource Pricing and the Evolution of Congestion Control. *Automatica*, 35:1969–1985, 1999.

[9] The PSC Networking group. The TCP-Friendly Website. http://www.psc.edu/networking/tcp_friendly.html.

[10] Liang Guo and Ibrahim Matta. The War between Mice and Elephants. In *Proceedings of ICNP'2001: The 9th IEEE International Conference on Network Protocols*, Riverside, CA, November 2001.

[11] C.V. Hollot, V. Misra, D. Towsley, and W. Gong. A Control Theoretic Analysis of RED. In *Proceedings of IEEE INFOCOM 2001*, Anchorage, AL, April 2001.

[12] C.V. Hollot, V. Misra, D. Towsley, and W. Gong. On Designing Improved Controllers for AQM Routers Supporting TCP Flows. In *Proceedings of IEEE INFOCOM 2001*, Anchorage, AL, April 2001.

[13] V. Jacobson. Congestion Avoidance and Control. In *Proceedings of ACM SIGCOMM'88*, Stanford, CA, August 1988.

[14] Sharad Jaiswal, Gianluca Iannaccone, Christophe Diot, Jim Kurose, and Don Towsley. Measurement and Classification of Out-of-Sequence Packets in a Tier-1 IP Backbone. In *Proceedings of IEEE INFOCOM 2003*, San Francisco, CA, April 2003.

[15] Dina Katabi, Mark Handley, and Charles Rohrs. Internet Congestion Control for High Bandwidth-Delay Product Networks. In *Proceedings of ACM SIGCOMM 2002*, San Diego, CA, August 2002.

[16] F.P. Kelly. Mathematical Modelling of the Internet. *Mathematics Unlimited - 2001 and Beyond*, pages 685–702, 2001.

[17] F.P. Kelly, A.K. Maulloo, and D.K.H. Tan. Rate Control for Communication Networks: Shadow Prices, Proportional Fairness and Stability. *Journal of Operations Research Society*, 1998.

[18] S. Kunniyur and R. Srikant. Analysis and Design of an Adaptive Virtual Queue (AVQ) Algorithm for Active Queue Management. In *Proceedings of ACM SIGCOMM 2001*, pages 123–134, August 2001.

[19] D. Lin and R. Morris. Dynamics of Random Early Detection. In *Proceedings of ACM SIGCOMM'97*, Cannes, France, September 1997.

[20] S. Low and D. Lapsley. Optimization Flow Control, I:Basic Algorithm and Convergence. *IEEE/ACM Transactions on Networking*, 1999.

[21] S. Low, F. Paganini, J. Wang, S. Adlakha, and J. Doyle. Dynamics of TCP/RED and a Scalable Control. In *Proceedings of IEEE INFOCOM 2002*, New York, NY, June 2002.

[22] S. Low, L. Peterson, and L. Wang. Understanding TCP Vegas: A Duality Model. In *Proceedings of ACM SIGMETRICS 2001*, Boston, MA, June 2001.

[23] Ibrahim Matta and Azer Bestavros. QoS Controllers for the Internet. In *Proceedings of the NSF Workshop on Information Technology*, March 2000.

[24] J. Postel. RFC-768 User Datagram Protocol. *Request For Comments, Internet Network Information Center*, 1980.

[25] R. Puri, Kang-Won Lee, K. Ramchandran, and V. Bharghavan. An Integrated Source Transcoding and Congestion Control Paradigm for Video Streaming in the Internet. *IEEE Transactions on Multimedia*, 3(1), 2001.

[26] Karu Ratnam and Ibrahim Matta. WTCP: An Efficient Mechanism for Improving Wireless Access to TCP Services. *International Journal of Communication Systems — Special Issue on Wireless Access to the Global Internet: Mobile Radio Networks and Satellite Systems*, February 2003.

[27] S. Shenker. A Theoretical Analysis of Feedback Flow Control. In *Proceedings ACM SIGCOMM'90*, Philadelphia, PA, September 1990.