October 2003

End-to-End Congestion Control Schemes: Utility Functions, Random Losses and ECN Marks

Srisankar S. Kunniyur
University of Pennsylvania, kunniyur@seas.upenn.edu

R. Srikant
University of Illinois

End-to-End Congestion Control Schemes: Utility Functions, Random Losses and ECN Marks

Abstract

We present a framework for designing end-to-end congestion control schemes in a network where each user may have a different utility function and may experience noncongestion-related losses. We first show that there exists an additive-increase-multiplicative-decrease scheme using only end-to-end measurable losses such that a socially optimal solution can be reached. We incorporate round-trip delay in this model, and show that one can generalize observations regarding TCP-type congestion avoidance to more general window flow control schemes. We then consider explicit congestion notification (ECN) as an alternate mechanism (instead of losses) for signaling congestion and show that ECN marking levels can be designed to nearly eliminate losses in the network by choosing the marking level independently for each node in the network. While the ECN marking level at each node may depend on the number of flows through the node, the appropriate marking level can be estimated using only aggregate flow measurements, i.e., per-flow measurements are not required.

Keywords

Explicit congestion notification (ECN) marking, Internet congestion control, TCP, TCP over wireless

End-to-End Congestion Control Schemes: Utility Functions, Random Losses and ECN Marks

Srisankar Kunniyur and R. Srikant, *Senior Member, IEEE*

***Abstract*—We present a framework for designing end-to-end congestion control schemes in a network where each user may have a different utility function and may experience non-congestion-related losses. We first show that there exists an additive-increase-multiplicative-decrease scheme using only end-to-end measurable losses such that a socially optimal solution can be reached. We incorporate round-trip delay in this model, and show that one can generalize observations regarding TCP-type congestion avoidance to more general window flow control schemes. We then consider explicit congestion notification (ECN) as an alternate mechanism (instead of losses) for signaling congestion and show that ECN marking levels can be designed to nearly eliminate losses in the network by choosing the marking level independently for each node in the network. While the ECN marking level at each node may depend on the number of flows through the node, the appropriate marking level can be estimated using only aggregate flow measurements, i.e., per-flow measurements are not required.**

***Index Terms*—Explicit congestion notification (ECN) marking, Internet congestion control, TCP, TCP over wireless.**

I. Introduction

RECENTLY, there has been surge of interest in designing best-effort service networks that can deliver low-loss low-delay service by encouraging users to adapt to the network congestion using minimal information from the network. The potential advantages of such networks would be the ability to offer even real-time services with little or no interaction from the core network, i.e., without the need for a centralized admission control, resource reservation, or complicated scheduling mechanisms. This work is partly motivated by the recent works of Gibbens and Kelly [6], [7], who have demonstrated the possibility of designing such networks using simple models. Some of the issues that have to be addressed when designing these networks include the following:

- defining appropriate notions of fairness;
- designing a pricing scheme to induce noncooperative users to work toward an equilibrium that is fair;
- designing a control mechanism for achieving this fairness using end-to-end measurable quantities such as lost packets;
- replacing losses with alternate indicators of congestion to evolve toward networks where congestion-related losses are negligible;
- providing ways to combat spurious congestion indicators such as errors on wireless links.

In this paper, our goal is to provide a simple framework based on deterministic fluid models that would provide insight into the effect of utility functions, random losses, and explicit congestion notification on the design of end-to-end congestion controllers.

We start with the nonlinear programming formulation of a flow allocation problem suggested in [12] from which a penalty function formulation is derived in [11]. In [11], it has been shown that a congestion controller can be designed such that the equilibrium point of the congestion controller is stable and converges to the unique solution of the penalty function form of the nonlinear program. We incorporate random losses[1] in the model and first show that by appropriately designing the penalty function, the resulting congestion controller for each user is a function of only its own loss rate and does not depend on any other information from the network. We also show that the penalty function can also be thought as a pricing scheme which steers a set of noncooperative users to a socially optimal solution as in the "smart-market" proposal [22]. Our formulation is also closely related to the approach presented in [21]. While [21] uses duality theory to solve the nonlinear program formulation of the resource allocation problem, we use a penalty function approach as in [11]. Also, [21] does not deal with random losses in the network. As we will see later, our primary motivation for doing this is that we want TCP to be a special case of our formulation and thus, any controller in the class that we study can be checked for TCP-friendliness [4]. Further, our explicit congestion notification (ECN) marking scheme, described later, becomes a straightforward modification to TCP where losses are simply replaced by marks in the congestion avoidance algorithm. For yet another related but different approach, see [8].

Our approach allows for different congestion controllers for different users which are directly derived from their utility functions. This allows one to model the heterogeneity in the needs of different applications. It is now well established that various notions of fairness can be defined in terms of appropriate utility

Manuscript received February 14, 2001; revised August 13, 2002; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor T. V. Lakshman. This work was supported by the National Science Foundation under Grant ANI-9813710 and Grant ANI-9714685. An earlier version of this paper was presented at the IEEE INFOCOM 2000, Tel Aviv, Israel, March 2000.

S. Kunniyur is with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104 USA (e-mail: kunniyur@ee.upenn.edu).

R. Srikant is with the Department of Electrical and Computer Engineering and the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA (e-mail: rsrikant@uiuc.edu).

Digital Object Identifier 10.1109/TNET.2003.818183

[1]We use the term *random losses* to indicate noncongestion-related losses such as errors on wireless links. However, in our fluid models used throughout the paper, we do not explicitly model the stochastic behavior of the loss process. We simply reduce the number of packets by a certain fraction to account for noncongestion-related losses.

functions [12], [23], [26]. While the well-known *max-min* fairness [1] cannot be defined in terms of a single utility function, it can still be defined in terms of a sequence of utility functions [12]. Thus, another motivation for allowing different utility functions for different users is to develop a model that would potentially allow one to study interactions between different types of congestion controllers derived by starting with different notions of fairness.

We then study the effects of round-trip delay on different window flow control schemes that could be derived by starting with different utility functions. The effect of random losses and round-trip delay on TCP performance have been quantified in [5], [14], and [19]. Our results generalize these earlier works to the case of multiple nodes and to window flow controllers derived from other utility functions.

Finally, we study the use of early congestion notification, prior to losses occurring, using ECN marks. We show that, in the fluid model framework, there is a notion of a marking level at each node and that marking levels for each node can be chosen in a decentralized manner to achieve loss-free service globally. The framework developed in this paper allows us to show that, with ECN marking, the deleterious effects of noncongestion-related losses can be nearly eliminated.

The main contributions of this paper are as follows.

1) We present an analytical framework to study various congestion control schemes in the presence of random losses in terms of users optimizing some utility function.
2) We obtain window-flow control algorithms that approximate the optimal solution of the fluid model. In this framework, we consider a TCP-like source and derive its utility function.
3) We study the notion of ECN marking and marking levels that would lead to a low-loss operation throughout the network and the impact of the number of users on the possibility of marking leading to low-loss operation.
4) We present a decentralized adaptive marking algorithms at each link that leads to socially optimal operation of the network.

II. End-to-End Rate-Based Congestion Control in the Presence of Random Losses

Consider a network with a set of links $\mathcal{L}$ such that link $l \in \mathcal{L}$ has capacity C_l. The network is used by a set of users $\mathcal{R}$. Associated with each user $r \in \mathcal{R}$ is a route which is also denoted by r, and which consists of a subset of $\mathcal{L}$. Now, consider a loss-sensitive user r which generates traffic at rate x_r. Let x_r^c be a vector of rates of all the other users in the network. We think of a user as having a transmitter and a receiver. The transmitted rate is x_r and let the received rate in the absence of noncongestion related losses be $y_r \leq x_r$. The received rate could be less than the transmitted rate due to congestion in the network. However, the rate at which packets are received at the receiver for user r is, in general, not only a function of congestion, but is also a function of noncongestion-related losses such as hardware failures in a wireline network, or more frequently, due to errors on wireless links on the route. Typically, these are modeled as a random phenomena that are independent across users. In our deterministic fluid model, we simply let the received rate for user r be $z_r = \alpha_r y_r$, where $(1 - \alpha_r)$ is the fraction of packets lost due to noncongestion related reasons. These are typically referred to as *random losses*. The received rate z_r is some function of x_r and x_r^c and we denote it by $f_r(x_r, x_r^c)$. The objective of user r is to maximize

$$J_r(x_r) = w_r U_r(x_r) - \beta_r P(x_r, x_r^c) \tag{1}$$

where U_r is a utility function and P is some function of the transmitted rates of all the users. For example, since our goal is build a low-loss low-delay network, P could be thought of as a penalty on the loss rate $(x_r - z_r)$. The parameters w_r and β_r attempt to trade off between maximizing utility and minimizing loss rate. The above problem is a game among the heterogeneous set of users $\mathcal{R}$ where each user $r \in \mathcal{R}$ attempts to maximize its own objective given by (1). Throughout this paper, we will refer to this as the *congestion control game*. Ideally we would like this game to have a unique equilibrium point and for the set of users to converge to this equilibrium point from any arbitrary initial condition. In this section, under the following assumptions implicitly used in [11], we show that there is an end-to-end rate-based congestion control scheme which achieves these goals with no feedback from the core network.

1) The loss rate for user r on a link $l \in r$ is given by

$$x_r p_l \left(C_l, \sum_{j: l \in j} x_j \right)$$

where $p_l(C_l, z)$ is the total rate loss or the loss probability at link l, when the total arrival rate into the link is z. If there is any loss at a link, then we simply assume that the total loss is distributed among the users in proportion to their flow rates. For example, this would be a good approximation of FIFO queueing with small buffers and packets that are small compared to the capacity of the link, i.e., a fluid model for the traffic. Thus, we do not require any complicated per-flow scheduling mechanism at each node. We assume that $p_l(C_l, z)$ is an increasing function of z.

2) The total loss rate $(x_r - y_r)$ due to congestion for user r is given by

$$(x_r - y_r) = \sum_{l: l \in r} x_r p_l \left(C_l, \sum_{j: l \in j} x_j \right). \tag{2}$$

We will refer to this to as the *link independence* assumption for the loss. Thus, we assume that the same flow is presented by a user to all links on its route. This assumption is reasonable if the p_l's are small. If p_l's are not small and marking is assumed at the links, then it is possible to modify the utility function to account for the exact expression for the end-to-end marking probability (see [13]). Alternately, if the p_l's are interpreted as prices generated by the links, then the total price on a path is the sum of the prices on the links in the path and then again, this assumption is reasonable. However, our simulations indicate that this assumption

is reasonable even when congestion indication is provided in the form of lost packets.

3) The *penalty* function is of the form

$$P(x_r, z_r) = \beta \int_0^{x_r} \frac{x - f_r(x, x_r^c)}{x} dx.$$

Recall that $f_r(x, x_r^c)$ is simply the received rate z_r of user r when the transmitted rate of user r is x_r and x_r^c is the vector of transmitted rates of all the other users in the network. Later, we will argue that the penalty-per-unit flow has the interpretation of price.

4) $U_r(\cdot)$ is a continuously differentiable, strictly concave, increasing function in the interval $(0, \infty)$ and we assume that $U_r(x_r)$ is unbounded as $x_r \to 0$. Assuming $U_r(x_r)$ is unbounded as $x_r \to 0$ ensures that $x_r > 0$ for all $r \in \mathcal{R}$, i.e., all users have nonzero rates in the optimal solution. Examples of such a function include $\log x_r$ and $-1/x_r$. An open issue is to incorporate nonconcave utility functions such as those studied in [2].

Proposition 1: The game admits a unique Nash equilibrium which is also the unique maximum of the following team problem, i.e., one where all users jointly optimize a single performance objective, as follows:

$$\max_{\{x_r\}} \sum_r \left(\frac{w_r}{\alpha_r \beta_r}\right) U_r(x_r) - \beta \sum_{l \in \mathcal{L}} \int_0^{\sum_{j: l \in j} x_j} p_l(C_l, x) dx - \beta \frac{1-\alpha_r}{\alpha_r} x_r \tag{3}$$

subject to $x_r \geq 0$, $\forall r$.

Proof: The objective function in (3) is a strictly concave function, and thus, has a unique maximum. The first-order necessary conditions for the maximum (which is also sufficient because of the strict concavity) are given by

$$\left(\frac{w_r}{\alpha_r \beta_r}\right) U_r'(x_r) - \beta \sum_{l: l \in r} p_l \left(C_l, \sum_{j: l \in j} x_j\right) - \beta \frac{1-\alpha_r}{\alpha_r} = 0, \quad \forall r.$$

Due to the link independence assumption for the loss [Assumption 2)], this is the same as

$$\left(\frac{w_r}{\alpha_r \beta_r}\right) U_r'(x_r) - \beta \frac{x_r - y_r}{x_r} - \beta \frac{1-\alpha_r}{\alpha_r} = 0, \quad \forall r$$

$$\Rightarrow \quad \left(\frac{w_r}{\beta_r}\right) U_r'(x_r) - \beta \frac{x_r - z_r}{x_r} = 0, \quad \forall r \tag{4}$$

which are nothing but the necessary and sufficient conditions for the Nash equilibrium of the congestion-control game. Since (4) has a unique solution due to the concavity of the objective function of the team problem, the Nash equilibrium of the congestion-control game is unique and is the same as the optimal solution of the team problem. □

Let the loss probability at link l be given by the loss probability for a $M/M/1/B$ queue, where B is the buffer size at the link. We now know that

$$p_l(C_l, u) = \frac{(1-\rho)\rho^B}{1-\rho^{B+1}}$$

where $\rho := (u/C_l)$. Let the arrival rate, capacity, and buffer size be scaled by a factor K as in a many-sources large-deviation scaling. By letting $K \to \infty$, we get

$$\lim_{K \to \infty} \frac{(1-\rho)\rho^{KB}}{1-\rho^{KB+1}} = \frac{(u - C_l)^+}{u}.$$

Note that even though the buffer size goes to ∞, the delay remains constant as the capacity also goes to ∞. Throughout this paper, we will use

$$p_l(C_l, u) = \frac{(u - C_l)^+}{u}.$$

In a deterministic fluid model, this has the simple interpretation of fraction of fluid lost when the arrival rate exceeds capacity.

Corollary 1: In the absence of random losses ($\alpha_r = 1$, $\forall r$), the game admits a unique Nash equilibrium which is also the unique maximum of the following team problem, i.e., one where all users jointly optimize a single performance objective

$$\max_{\{x_r\}} \sum_r \left(\frac{w_r}{\beta_r}\right) U_r(x_r) - \beta \sum_{l \in \mathcal{L}} \int_0^{\sum_{l \in j} x_j} \frac{(x - C_l)^+}{x} dx \tag{5}$$

subject to $x_r \geq 0$, $\forall r$. Moreover, as $\beta \to \infty$, the Nash equilibrium of Proposition 1 converges to the unique optimal solution of

$$\max_{\{x_r\}} \sum_r \left(\frac{w_r}{\beta_r}\right) U_r(x_r) \tag{6}$$

subject to

$$\sum_{j: l \in j} x_j \leq C_l, \quad \forall l \in \mathcal{L} \tag{7}$$

$$x_r \geq 0, \quad \forall r \in \mathcal{R}. \tag{8}$$

□

From now on, for the purpose of simplicity, we consider only two classes of utility functions, although the following results can be easily extended to all functions satisfying Assumption 4) stated earlier. Let $\mathcal{R}_1$ be the set of users whose utility function is $\log x_r$, and let the utility function of user $r \in \mathcal{R}_2 = \mathcal{R} \setminus \mathcal{R}_1$ be $-1/x_r^{\nu_r}$. If $\mathcal{R}_1 = \mathcal{R}$, then (6) defines the proportionally fair solution [12] and if $\mathcal{R}_2 = \mathcal{R}$ and $\nu_r = 1$ $\forall r$, then (6) defines the minimum potential delay fairness [23].

Proposition 2: Suppose that each user $r \in \mathcal{R}_1$ employs the congestion-control algorithm

$$\dot{x}_r = \kappa_r \left(\frac{w_r}{\beta_r} - \beta(x_r - z_r)\right) \tag{9}$$

and each user $j \in \mathcal{R}_2$ employs the algorithm

$$\dot{x}_j = \kappa_j \left(\frac{\nu_j w_j}{\beta_j} - \beta x_j^{\nu_j}(x_j - z_j)\right) \tag{10}$$

where κ_r, $\kappa_j > 0$ are some constants. The above congestion-control scheme converges to the unique solution of (3).

Proof: It is easy to see that (9) and (10) can be rewritten as

$$\dot{x}_r = \kappa_r \left(\frac{w_r}{\beta_r} - \beta\alpha_r(x_r - y_r) - \beta(1-\alpha_r)x_r \right) = \kappa_r\alpha_r \left(\frac{w_r}{\alpha_r\beta_r} - \beta(x_r - y_r) - \beta\frac{(1-\alpha_r)x_r}{\alpha_r} \right)$$

for each user $r \in \mathcal{R}_1$, and each user $j \in \mathcal{R}_2$ uses the algorithm

$$\dot{x}_j = \kappa_j\alpha_j \left(\frac{\nu_j w_j}{\beta_j\alpha_j} - \beta x_j^{\nu_j}(x_j - y_j) - \beta\frac{(1-\alpha_j)}{\alpha_j}x_j^{\nu_j+1} \right).$$

Now, the convergence of the congestion-control scheme (9), (10) follows along the lines of the proof of [11]. □

By letting $\alpha_r = 1$, $\forall r$ we recover the result in [11] when there are no random losses in the system. The above proposition shows that, for each of the utility functions, in the class defined by Assumption 4), there exists a congestion-control scheme which achieves the Nash equilibrium (or team-optimality) using only information available at the transmitter (x_r) and the receiver (y_r). Mo and Walrand [26] have derived an alternate end-to-end control scheme in the case where there are no random losses in the system and the round-trip time measurements are explicitly accounted in their model. As we will see later, the window flow control approximation of our scheme is more along the lines of TCP which uses packet loss as the congestion indicator.

III. Window Flow Control

Window flow control where the window size is modified upon receipt of acks or nacks is a more convenient implementation than a rate-based control scheme because it is inherently self-clocking, i.e., there is no need to decide parameters like measurement intervals, discretization time-steps, etc. To obtain a window flow control mechanism to reach our stable Nash equilibrium point, we start by discretizing (9), (10) to obtain

$$\frac{x_r(t+\delta) - x_r(t)}{\delta} = \kappa_r \left(\frac{w_r}{\beta_r} - \beta(x_r - z_r) \right), \quad r \in \mathcal{R}_1 \tag{11}$$

and

$$\frac{x_j(t+\delta) - x_j(t)}{\delta} = \kappa_j \left(\frac{\nu_j w_j}{\beta_j} - \beta x_j^{\nu_j}(x_j - z_j) \right), \quad j \in \mathcal{R}_2.$$

Now, let the round-trip delay for user r be d_r and let $W_r(t)$ be its window size at time t. We make the following approximation relating data transmission rate and window size [19]:

$$x_r(t) \approx \frac{W_r(t)}{d_r}.$$

Let $A_r(t, t+\delta)$ denote the numbers of acks received by user r in the time interval $[t, t+\delta)$ and let $N_r(t, t+\delta)$ be the number of nacks received by user r in the same time interval. By acks, we refer to both *positive* and *negative* acknowledgments here. Thus, $A_r(t, t+\delta) \geq N_r(t, t+\delta)$. Although we use the term nacks, loss of packets could be conveyed through other mechanisms such as duplicate acks or timeouts as in TCP. We further note that

$$\frac{A_r(t, t+\delta)}{\delta} \approx x_r(t) \approx \frac{W_r(t)}{d_r}.$$

Thus, we have

$$\frac{x_r(t+\delta) - x_r(t)}{\delta} = \frac{W_r(t+\delta) - W_r(t)}{A(t, t+\delta)} \frac{A(t, t+\delta)}{d_r\delta} = \frac{W_r(t+\delta) - W_r(t)}{A(t, t+\delta)} \frac{W_r(t)}{d_r^2}.$$

Also, note that

$$(x_r - z_r)\delta \approx N_r(t, t+\delta).$$

Using these approximations, the congestion-control algorithms become

$$W_r(t+\delta) - W_r(t) = \frac{\kappa_r w_r d_r^2}{\beta_r W_r} A_r(t, t+\delta) - \kappa_r\beta d_r N_r(t, t+\delta), \quad r \in \mathcal{R}_1 \tag{12}$$

$$W_j(t+\delta) - W_j(t) = \frac{\kappa_j \nu_j w_j d_j^2}{\beta_j W_j} A_j(t, t+\delta) - \kappa_j\beta \frac{W_j^{\nu_j}}{d_j^{\nu_j - 1}} N_j(t, t+\delta), \quad j \in \mathcal{R}_2. \tag{13}$$

Remark 1: The discrete-time representation for the window flow control mechanism is simply used for convenience. It has to be interpreted as follows.

- $r \in \mathcal{R}_1$: For each received ack, the window size is increased in proportion to d_r^2/W_r; for each lost packet, the window size is decreased by a fixed amount.
- $j \in \mathcal{R}_2$: The window size is increased again in proportion to d_j^2/W_j for each received ack; however, it is decreased in proportion to a function of the current window size $W_j^{\nu_j}$ upon receipt of each nack.

□

A. Relationship to TCP

We discuss the similarities between the above congestion control algorithms and most current versions of TCP. In fact, our results allow us to generalize many earlier observations regarding the performance of TCP-like congestion-control algorithms for the cases of a single link shared by multiple users with different round-trip delays and a single link utilized by a single user who suffers from random losses. To this end, we first note the striking similarity between current versions of TCP and the algorithm for users in the set $\mathcal{R}_2$ when $\nu_r = 1$. The significant difference is that the *increase* term is dependent on d_r.

Ignoring the rapid slow-start phase, most current versions of TCP employ the following algorithm:

$$W_j(t+\delta) - W_j(t) = \frac{1}{W_j} A_j(t, t+\delta) - 0.5 W_j N_j(t, t+\delta). \tag{14}$$

Thus, a TCP source would correspond to a user in our framework whose parameters satisfy $\nu_j = 1$, $\kappa_j = 1$ $w_j/\beta_j = 1/d_j^2$,

and $\beta = 0.5$. A little care has to be used in interpreting the discretization that resulted in (13). The discretization was done assuming that we are considering very small intervals of δ units. However, nacks in TCP-type window flow control may not be frequent enough to assume that there would be several nacks in an interval of size δ. Thus, it is more reasonable to suppose that β is not exactly equal to 0.5. In fact, we will later show that the value of β can be approximated by ln(2). Due to this reason, we will simply use β, instead of using 0.5 in approximating the dynamics of TCP-type congestion avoidance as a continuous-time rate control. We also note that (14) is mainly intended to capture the steady-state behavior of TCP. A more precise model would account for feedback delay; we refer the interested reader to [16] and [25].

Remark 2: A network of users using TCP-type congestion avoidance can be thought of as a team of users whose goal is to optimize the following objective:

$$\max_{\{x_j\}} \sum_j \left(-\frac{1}{\alpha_j d_j^2 x_j} - \beta \frac{(1-\alpha_j)}{\alpha_j} x_j \right) - \beta \sum_{l \in \mathcal{L}} \int_0^{\sum_{j: l \in j} x_j} \frac{(x - C_l)^+}{x} dx. \tag{15}$$

□

In [9], a different utility function has been suggested for TCP-type congestion avoidance. The starting point for their analysis is a stochastic model from which an ordinary differential equation is derived. Events in TCP occur at packet level (i.e., most of the events are triggered by the arrival of an ack or a nack), while [9] also uses a discrete-time model. Ref. [9] also assumes a rare-negative feedback regime, ignores slow-start, and assumes sources have identical round-trip times. Their results suggest that the utility function is of the form $\log(x/(1+x))$. For large x

$$\log \frac{x}{1+x} = \log \frac{1}{1+\frac{1}{x}} \approx \log\left(1 - \frac{1}{x}\right) \approx -\frac{1}{x}$$

thus, recovering our result. We note that these methods are only approximations to TCP-type congestion controllers and do not attempt to model TCP precisely. For example, slow start, timeout, etc. are ignored. However, as we mention later, $-1/x$ captures the dependence of the throughput on the inverse of the product of round-trip delay and probability of loss. An alternate utility function for TCP has been proposed in [13]. However, when the loss rate is small, the two utility functions approximately yield the same steady-state throughput.

In the above model, we made the approximation that window size is reduced by a factor βW_j for each received nack. Alternately, we could assume that the congestion controller halves the window size for each received ack and this leads to

$$W(k+1) = \frac{A(k)}{W(k)} + \left(\frac{1}{2}\right)^{N(k)} W(k) \tag{16}$$

where the length of each time slot is some multiple of the round-trip delay, $A(k)$ is the number of acks received, and $N(k)$ is the number of packets lost or the number of nacks in the kth time slot. Using the approximation

$$\left(\frac{1}{2}\right)^{N(k)} = e^{-\ln(2)N(k)} \approx 1 - \ln(2)N(k)$$

we get the earlier model. Further, the decrease factor is given by $\beta = \ln(2) = 0.6931$. Note that this value of β is close to $\beta = 2/3$ proposed in [27].

The impact of random losses on the performance of TCP and other transport protocols has been widely studied due to the emergence of mobile computing applications [14], [15], [19], [20], [24], [28]. Typically, analytical results relating random losses and the delay-bandwidth product are available only for the case of a single link accessed by a single user. Remark 2 generalizes this to the case of *multiple users in a network*. To see this, let us specialize the result to the case of a single link and a single user. In this case, the objective of the single user is

$$\max_{\{x_j\}} -\frac{1}{d_j^2 x_j} - \beta \frac{(1-\alpha_j)}{\alpha_j} x_j - \beta \int_0^{x_j} \frac{(x - C_l)^+}{x} dx. \tag{17}$$

For large values of βd_j^2, it is easily seen that the optimal solution x^* is less than C_l. Thus, we get

$$x_j^*(\alpha_j, d_j) = \frac{1}{\sqrt{\beta} d_j \sqrt{1-\alpha_j}} \tag{18}$$

a fact observed even in the original TCP congestion avoidance paper [10] and rediscovered later by many others. While Remark 2 extends this to a network with multiple users, Proposition 2 presents the general result when a window flow control scheme (12), (13) is used in a network of heterogeneous users.

It is also instructive to compare the solution of (17) with $\alpha_j = 0$ to the solution (18) obtained with $\alpha_j \neq 0$. The solution with $\alpha_j = 0$ is given by

$$x_j^*(\alpha_j = 0, d_j) = \frac{1}{2}\left(C_l + \sqrt{C_l^2 + \frac{4}{\beta d_j^2}}\right) \approx C_l \tag{19}$$

for large βd_j^2. Comparing (18) and (19), it is easy to see that $x_j^*(\alpha_j = 0, d_j) > x_j^*(\alpha_j, d_j)$ implies that

$$1 - \alpha_j > \frac{1}{\beta d_j^2 C_l^2}.$$

Thus, the performance of the window flow control scheme (14) deteriorates when the random loss probability is much larger than the inverse of the square of the delay-bandwidth product, a fact observed in [19] and [24]. Thus, Remark 2 is a generalization of this fact observed earlier for a single-link single-user case.

B. Round-Trip Delay

As in TCP, supposing one ignores round-trip delay in the congestion control mechanism (12), (13), we obtain the following window adaptation scheme:

$$W_r(t+\delta) - W_r(t) = \frac{\kappa_r w_r}{\beta_r W_r} A_r(t, t+\delta) - \beta \kappa_r N_r(t, t+\delta), \quad r \in \mathcal{R}_1, \tag{20}$$

$$W_j(t+\delta) - W_j(t) = \frac{\kappa_j \nu_j w_j}{\beta_j W_j} A_j(t, t+\delta) - \kappa_j \beta W_j^{\nu_j} N_j(t, t+\delta), \quad j \in \mathcal{R}_2. \tag{21}$$

Letting $\delta \to 0$, this can written as

$$\frac{dW_r}{dt} = \kappa_r \left(\frac{w_r x_r}{\beta_r W_r} - \beta(x_r - z_r) \right), \quad r \in \mathcal{R}_1 \tag{22}$$

and

$$\frac{dW_j}{dt} = \kappa_j \left(\frac{\nu_j w_j x_j}{\beta_j W_j} - (x_j - z_j)\beta W_j^{\nu_j} \right), \quad j \in \mathcal{R}_2. \tag{23}$$

Using the relation $x_r = W_r/d_r$, we can rewrite this in terms of rates as

$$\dot{x}_r = \frac{\kappa_r}{d_r} \left(\frac{w_r}{\beta_r d_r} - \beta(x_r - z_r) \right), \quad r \in \mathcal{R}_1 \tag{24}$$

and

$$\dot{x}_j = \kappa_j d_j^{\nu_j - 1} \left(\frac{\nu_j w_j}{\beta_j d_j^{\nu_j+1}} - (x_j - z_j)\beta x_j^{\nu_j} \right), \quad j \in \mathcal{R}_2. \tag{25}$$

Remark 3: The window flow control scheme (20), (21) obtained by ignoring round-trip times can be thought as a team of users attempting to converge to the unique solution of the following problem:

$$\max_{\{x_r\}} \sum_{r \in R_1} \left(\frac{w_r}{\alpha_r \beta_r d_r} \right) U_r(x_r) + \sum_{j \in R_2} \left(\frac{w_j}{\alpha_j \beta_j d_j^{\nu_j+1}} \right) U_j(x_j) - \sum_{r \in R_1 \bigcup R_2} \beta \frac{(1-\alpha_r)}{\alpha_r} x_r - \beta \sum_{l \in \mathcal{L}} \int_0^{\sum_{l \in j} x_j} \frac{(x - C_l)^+}{x} dx. \tag{26}$$

□

IV. Pricing and TCP-Friendliness

Suppose β is the price per mark charged by the network. Then the cost incurred by user i at node/link l is $\beta f_l x_i$, where f_l is the fraction of packets marked by node l. Therefore, the total cost incurred by user i (by the link independence assumption) is $\sum_{l:l \in i} \beta f_l x_i$. However, the utility of rate x_i to user i is $(w_i/\beta_i)U_i(x_i)$. Therefore, since user i cannot estimate the impact of its own flow on the marking rate, it solves the following optimization problem:

$$\max g_i(x_i) = \frac{w_i}{\beta_i} U_i(x_i) - \sum_{l:l \in i} \beta f_l x_i.$$

Differentiating $g_i(x_i)$ with respect to x_i, we get

$$\frac{dg_i}{dx_i} = \frac{w_i}{\beta_i} U_i'(x_i) - \sum_{l:l \in i} \beta f_l = \frac{w_i}{\beta_i} U_i'(x_i) - \beta \sum_{l:l \in i} \frac{\left(\sum_{s:l \in s} x_s - \tilde{C}_l \right)^+}{\sum_{s:l \in s} x_s}.$$

Thus, the first-order necessary condition is the same as that of the original game that we considered. A gradient ascent procedure to find the maximum in the above optimization problem again leads to the same additive-increase-multiplicative-decrease algorithm as before.

Our results also provide a justification of the definition of TCP-friendliness in [4] in the context of the nonlinear program formulation of the congestion control problem. A flow is TCP-friendly if its arrival rate does not exceed the arrival of a conformant TCP connection in the same circumstances [4]. In particular, if the arrival rate of a flow exceeds $K/d\sqrt{1-\alpha}$, where K is some constant and $1-\alpha$ is the packet drop probability, then the flow is *not* TCP-friendly.

From (18), we see that the steady-state throughput of a TCP flow is equal to $1/(\sqrt{\beta} d \sqrt{1-\alpha})$, where d is the round-trip delay of the flow and $1-\alpha$ is the packet loss probability experienced by the flow. From (26) using the same type of argument used in obtaining (18) for TCP-type congestion avoidance schemes, it is easy to see that any congestion controller j whose throughput is proportional to $1/d_j(1-\alpha_j)^{1/(\nu_j+1)}$ can be thought as a user j with the following utility function:

$$U_j(x_j) = \begin{cases} -\frac{1}{x_j^{\nu_j}}, & \nu_j > 0 \\ \log x_j, & \nu_j = 0. \end{cases}$$

Thus, we can associate utilities with any congestion control algorithm and solve the optimization problem given in (3) to find whether the flow is TCP-friendly or not. In general, we can generalize the notion of TCP friendliness to compare the throughputs of any two arbitrary congestion controllers that fit the utility function model. As a result, one can design congestion controllers that are "friendly" to each other.

Now, consider a link of capacity $C = 300$ units that is shared by 100 flows, 50 of which are TCP users and the rest employ an additive-increase-multiplicative-decrease congestion controller whose throughput is proportional to $1/d_j(1-\alpha_j)$. The throughput seen by each user can be obtained by solving the following optimization problem:

$$\max \sum_{i=1}^{50} \log x_i - \sum_{j=51}^{100} \frac{1}{x_j}$$

subject to

$$\sum_{i=1}^{50} x_i + \sum_{i=51}^{100} x_j \le 300$$

and $x_i \ge 0, \forall i$. Solving this problem yields $x_i = 4.0$, $i \le 50$ and $x_j = 2.0$, $j > 50$. Thus, in this example, we see that the users with throughput proportional to $1/d_j(1-\alpha_j)$ get a larger share of the bandwidth compared to TCP flows. On the other hand if these non-TCP-friendly users were replaced by users whose throughput is proportional to $1/d_j(1-\alpha_j)^{1/3}$, then the corresponding optimization problem would be one where $\log x_i$ is replaced by $-(1/x_i^2)$. In this case, the solution yields $x_i = 2.76$, $i \le 50$ and $x_j = 3.24$, $j > 50$. In this example, users whose throughput is proportional to $1/d_j(1-\alpha_j)^{1/3}$ are not TCP-competitive, i.e., their share of the bandwidth is smaller than that obtained by TCP. In either case, the users using a larger share of the bandwidth will receive more marks. If the price charged is proportional to the marks received by a user, then users using a larger fraction of the resources will pay more than the rest of the users.

V. ECN Marks

Explicit congestion notification (ECN) has been recently proposed to provide early indication to sources about imminent congestion [3]. Current versions of TCP and the window flow control algorithms that we have discussed so far rely on loss as the congestion indicator. Clearly, this is not desirable if one wishes to operate the network at very low levels of loss. On the other hand, loss is a good indicator of congestion and one needs other signals from the network if we have to make congestion control decisions with very little or no loss. ECN marking is a mechanism to provide such information about the network to the users. We use the term ECN not to necessarily signify the implementations discussed in [3] or related works, but rather a simple marking scheme to serve as an early indicator of congestion before loss actually occurs at a node.

A. Decentralized Design of Marking Levels

In [6] and [7], marking mechanisms have been suggested for stochastic models of a single node accessed by many sources. To recast our fluid model to incorporate ECN marking, we simply have to interpret "lost" packets as "marked" packets. Since we have a bufferless model, we assume that, at each link l, a fraction of the packets are marked when the arrival rate exceeds some $\tilde{C}_l$, where $\tilde{C}_l \leq C_l$. The fraction of packets marked is given $(x-\tilde{C}_l)/x$ where x is arrival rate on link l. First, we consider the case with no random losses, and instead of interpreting z_r as the rate at which packets are received at the receiver r, we will now interpret z_r as the rate at which "unmarked" packets are received at the receiver. Thus, Proposition 1–2 can now be interpreted in these terms, with C_l replaced by $\tilde{C}_l$. Similarly, in the window flow control implementation, the window size should be reduced upon receipt of either a nack or a mark. In this framework, it is possible to offer a loss-free service if the marking level $\tilde{C}_l$ is chosen appropriately for each link. In what follows, we characterize the level $\tilde{C}_l$ at which marking should take place so that the total arrival rate on each link is less than the link capacity.

It is instructive to consider the case of a single link l of capacity C_l accessed by N_l sources, where the utility function of each user is $U_r(x_r)=\log x_r$ and $w_r=\beta_r=1$. The necessary and sufficient condition for the solution of (3) is given by

$$\frac{1}{x_r}-\beta\frac{\sum_r x_r-\tilde{C}_l}{\sum_r x_r}=0$$

for each r. By symmetry it is clear $x_r=x_s$ for any r, s. Therefore

$$\frac{1}{x_r}-\beta\frac{Nx_r-\tilde{C}_l}{Nx_r}=0 \quad \Rightarrow \quad Nx_r=\tilde{C}_l+\frac{N}{\beta}.$$

Thus, if $\tilde{C}_l+(N_l/\beta)\leq C_l$, then the solution to the optimization problem (3) results in zero loss. Note that $\tilde{C}_l$ depends upon the number of users in the system. Clearly, for a fixed β, if the number of users is very large, then there may not exist a marking level that ensures loss-free operation. Thus, increasing the available capacity through provisioning or increasing β are the only options to ensure loss-free service. Therefore, as N increases, we need to increase β to ensure loss-free service. It is interesting that, even in a network with multiple nodes, such a decentralized marking scheme, where the marking level on each link is obtained by considering it in isolation as above, results in zero loss. This is stated in the following proposition.

Proposition 3: For each l, suppose that the marking level $\tilde{C}_l$ is chosen to satisfy the following inequality:

$$\sum_{r\in\mathbf{R}_l\cap\mathcal{R}_1}\frac{w_r}{\beta\beta_r}\frac{C_l}{C_l-\tilde{C}_l}+\sum_{r\in\mathbf{R}_l\cap\mathcal{R}_2}\left[\frac{w_r\nu_r}{\beta\beta_r}\frac{C_l}{C_l-\tilde{C}_l}\right]^{\frac{1}{(\nu_r+1)}}\leq C_l \tag{27}$$

where $\mathbf{R}_l=\{r\in\mathcal{R}|l\in r\}$. (Recall that $\mathcal{R}_1$ is the set of users whose utility function is $\log x_r$, and $\mathcal{R}_2$ is the set of users whose utilities functions are of the form $-1/x_r^{\nu_r}$.) Then, the Nash equilibrium of the congestion control game satisfies

$$\sum_{r:l\in r}x_r\leq C_l,\quad \forall l.$$

Proof: We will prove the proposition by contradiction. Suppose there exists an l such that, at the Nash equilibrium point

$$\sum_{s\in\mathbf{R}_l}x_s>C_l. \tag{28}$$

Consider a route $r\in\mathcal{R}_1$ such that $l\in r$. (If such an r does not exist, it is trivial to modify the proof.) From the necessary and sufficient condition for the Nash equilibrium of the congestion control game, we have

$$\begin{aligned}\frac{w_r}{\beta_r}\frac{1}{x_r}&=\sum_{m:m\in r}\frac{\beta\left(\sum_{s\in\mathbf{R}_m}x_s-\tilde{C}_m\right)^+}{\sum_{s\in\mathbf{R}_m}x_s}\\&\geq\frac{\beta\left(\sum_{s\in\mathbf{R}_l}x_s-\tilde{C}_l\right)}{\sum_{s\in\mathbf{R}_l}x_s}\\&\geq\frac{\beta(C_l-\tilde{C}_l)}{C_l}.\end{aligned}$$

We have removed the superscript $+$ from the second equation above since we have assumed in (28) that $\sum_{s\in\mathbf{R}_l}x_s>C_l$. The last inequality follows from the fact that $(x-\tilde{C}_l)/x$ is an increasing function of x and we have assumed that on link l, $\sum_{s\in\mathbf{R}_l}x_s>C_l$. Thus

$$x_s\leq\frac{w_r}{\beta\beta_r}\frac{C_l}{C_l-\tilde{C}_l}. \tag{29}$$

Similarly, for $r\in\mathcal{R}_2$ such that $l\in r$,

$$\frac{w_r}{\beta_r}\frac{\nu_r}{x_r^{\nu_r+1}}\geq\frac{\beta(C_l-\tilde{C}_l)}{C_l}$$

and thus

$$x_s\leq\left[\frac{w_r\nu_r}{\beta\beta_r}\frac{C_l}{C_l-\tilde{C}_l}\right]^{\frac{1}{(\nu_r+1)}}. \tag{30}$$

From (29) and (30), the total flow on link l satisfies

$$\sum_{s\in\mathbf{R}_l\cap\mathcal{R}_1} x_s + \sum_{s\in\mathbf{R}_l\cap\mathcal{R}_2} x_s \le \sum_{r\in\mathbf{R}_l\cap\mathcal{R}_1} \frac{w_r}{\beta\beta_r}\frac{C_l}{C_l-\tilde{C}_l} + \sum_{r\in\mathbf{R}_l\cap\mathcal{R}_2}\left[\frac{w_r\nu_r}{\beta\beta_r}\frac{C_l}{C_l-\tilde{C}_l}\right]^{\frac{1}{(\nu_r+1)}}.$$

From (27), we have assumed that the marking level has been chosen such that this total flow on link l is less than or equal to C_l. This contradicts (28). □

A consequence of Proposition 3 is that the loss-based rate and window flow control algorithms described earlier can be used, along with appropriate marking, to provide loss-free service by simply substituting marks for negative acknowledgments.

B. Perturbations Due to Short Flows

In the previous section, we considered flows long enough to react to marks/losses. However, there might be some unresponsive flows or very short flows which might not react to marks. In this case, we can model such short flows/unresponsive flows as a bounded perturbation and study the existence of a decentralized marking algorithm that will lead to a near loss-free operation throughout the network. There are two ways to view this system.

1) Consider a deterministic fluid model where $p_l(.,.)$ is the marking rate for a specific marking level and total arrival rate at the link. In this case, the marking level has to chosen to account for the time-varying disturbance caused by the short flows.
2) We can also start with a stochastic discrete-time packet model of the system and use the stochastic-approximation approach [9] to infer that $p_l(.,.)$ is the expected loss rate with respect to the disturbance at link l, as a function of the marking level and total arrival rate. However, in this case, the marking level is designed such that the expected total arrival rate is less than the capacity of the link. This will lead to a near loss-free operation as the variability in the loss rate might lead to some losses in the system, but this variability is relatively small.

We discuss both models in this section.

1) Fluid Model Approach: In this section, we will assume a fluid model of the system and $p_l(.,.)$ to be the loss rate for a specific marking level and total arrival rate at the link. We assume that the total arrival rate of the short flows are bounded, i.e., if $\xi_l(t)$ represents the total arrival rate due to short flows at a link l, we assume that $\xi_l(t) \le \hat{\xi}_l$ for all $t \ge 0$.

We will show that in the case when all users have a log utility function, we can find a marking level in a decentralized way at each link such that there is no loss in the network. The case when the users have a general utility function is a topic of future research. We will first prove a lemma which upper bounds the total rate of the perturbed system into a link by the total rate of a perturbance-free system. We will then use this lemma along with Proposition 3 to show the existence of a decentralized marking algorithm.

The congestion control scheme for user i can now be written as

$$\dot{x}_i = \frac{w_i}{\beta_i} - \beta x_i \sum_{k:k\in i} p_k(\tilde{C}_k, \lambda_k + \xi_k) \tag{31}$$

where ξ_k is the bounded perturbation at link k and $\lambda_k = \sum_{j:k\in j} x_j$ is the total flow of the controlled sources into the link. Assume that perturbation at each link is bounded by $\hat{\xi}_l$.

Lemma 1: For a given $l \in \mathcal{L}$ and for each user i such that $i \in l$, consider the differential equation

$$\dot{y}_i = \frac{w_i}{\beta_i} - \beta y_i p_l\left(\tilde{C}_l, \sum_{i:l\in i} y_i\right) \quad y_i(0) = x_i(0). \tag{32}$$

Denote

$$y(t) = \sum_{i:l\in i} y_i(t) \quad x(t) = \sum_{i:l\in i} x_i(t).$$

Then, for all $t > 0$, we have

$$x(t) \le y(t).$$

Proof: We will prove the lemma by contradiction. Assume that there exists a finite time T such that

$$T := \sup\{t > 0 | x(s) \le y(s), \quad \forall s \le t\}. \tag{33}$$

By continuity, this implies

$$x(T) = y(T) \text{ and } \dot{x}(T) > \dot{y}(T). \tag{34}$$

Then

$$\begin{aligned}
\dot{x}(T) &= \left(\sum_{i:l\in i}\frac{w_i}{\beta_i}\right) \\
&\quad - \beta\sum_{i:l\in i} x_i(T)\sum_{k:k\in i} p_k\left(\tilde{C}_k, \sum_{j:k\in j} x_j(T) + \xi_k(T)\right) \\
&= \left(\sum_{i:l\in i}\frac{w_i}{\beta_i}\right) - \beta x(T) p_l\left(\tilde{C}_l, x(T) + \xi_l(T)\right) \\
&\quad - \beta\sum_{i:l\in i} x_i(T)\sum_{k\neq l:k\in i} p_k\left(\tilde{C}_k, \sum_{j:k\in j} x_j(T) + \xi_k(T)\right) \\
&\le \left(\sum_{i:l\in i}\frac{w_i}{\beta_i}\right) - \beta x(T) p_l\left(\tilde{C}_l, x(T) + \xi_l(T)\right) \\
&\le \left(\sum_{i:l\in i}\frac{w_i}{\beta_i}\right) - \beta x(T) p_l\left(\tilde{C}_l, x(T) + \xi_l(T)\right) \\
&= \left(\sum_{i:l\in i}\frac{w_i}{\beta_i}\right) - \beta y(T) p_l\left(\tilde{C}_l, y(T) + \xi_l(T)\right) \\
&= \dot{y}(T)
\end{aligned}$$

where the first inequality is due to the fact that $p_k(.,.)$ is nonnegative for all k, and the second inequality is due to the fact

that $\xi_l(t) > 0$ and $p_k(.,.)$ is an increasing function in its second argument, but this is a contradiction. Hence

$$x(t) \leq y(t) \quad \forall t > 0.$$

□

From Proposition 3, we know that in the case of no disturbances, we can obtain the marking level on each link by considering it in isolation and such a marking level would also lead to a loss-free operation for the entire network. We will show that we can choose the marking level in a decentralized fashion even in the presence of disturbances that will lead to a loss-free operation.

Proposition 4: For each l, suppose that the marking level $\tilde{C}_l$ is chosen to satisfy the following inequality:

$$\sum_{i:l\in i} \frac{w_i}{\beta\beta_i} \frac{1}{p_l(\tilde{C}_l, C_l)} \leq C_l - \hat{\xi}_l. \tag{35}$$

Then, the solution of the congestion control game (31) satisfies

$$\limsup_{t\to\infty} \sum_{i:l\in i} x_i(t) + \xi_l(t) \leq C_l.$$

Proof: We will prove the proposition by contradiction. Suppose there exists a link l such that, at the solution

$$\sum_{i:l\in i} x_i(t) + \xi_l(t) > C_l. \tag{36}$$

For each user i such that $l \in i$, consider the differential equation

$$\dot{y}_i = \frac{w_i}{\beta_i} - \beta y_i p_l \left(\tilde{C}_l, \sum_{i:l\in i} y_i\right) \quad y_i(0) = x_i(0). \tag{37}$$

From Proposition 3, we know that

$$\limsup_{t\to\infty} \sum_{i:l\in i} y_i(t) \leq C_l - \hat{\xi}_l.$$

We also know that

$$\begin{aligned} &\sum_{i:l\in i} x_i(t) \leq \sum_{i:l\in i} y_i(t) \\ \Rightarrow \quad &\limsup_{t\to\infty} \sum_{i:l\in i} x_i(t) \leq \limsup_{t\to\infty} \sum_{i:l\in i} y_i(t) \\ \Rightarrow \quad &\limsup_{t\to\infty} \sum_{i:l\in i} x_i(t) \leq C_l - \hat{\xi}_l \\ \Rightarrow \quad &\limsup_{t\to\infty} \sum_{i:l\in i} x_i(t) + \xi_l(t) \leq C_l \end{aligned}$$

which is a contradiction. □

Therefore, Proposition 4 shows that in the case of a proportionally fair network (i.e., one in which all the users have a $\log(x)$ utility function), a loss-free service can be guaranteed by choosing the marking level at each link according to (35). As mentioned earlier, the case where all users have a general utility function is a topic of future research.

2) Stochastic Approximation Model: In this case, we can interpret the congestion control scheme as a stochastic approximation of a stochastic discrete-time model of the system. With this model, we can now apply the method of ordinary differential equation (ODE) [9] to obtain the congestion control scheme

$$\dot{x}_i = \frac{w_i}{\beta_i} - \beta \left(U_i'(x_i)\right)^{-1} \sum_{k:k\in i} p_k \left(\tilde{C}_k, \sum_{j:k\in j} x_j + \bar{\xi}_k\right) \tag{38}$$

where $p_k(\tilde{C}_k, \sum_{j:k\in j} x_j + \bar{\xi}_k)$ is defined to be

$$E_\xi \left[\begin{array}{c}\text{Loss rate at link } k \text{ when the marking level is } \tilde{C}_k \\ \text{and the total arrival rate is } \sum_{j:k\in j} x_j + \xi_k\end{array}\right].$$

E_ξ denotes that the expectation is taken with respect to the perturbation term ξ. In this case, we can rewrite Proposition 3 to show that there exists a decentralized marking algorithm that ensures low-loss operation.

Proposition 5: For each l, suppose that the marking level $\tilde{C}_l$ is chosen to satisfy the following inequality:

$$\sum_{r:l\in r} U_r'^{-1} \left[\frac{\beta_r\beta}{w_r} p_l \left(\tilde{C}_l, \sum_{j:l\in j} x_j\right)\right] \leq C_l - \bar{\xi}_l \tag{39}$$

then the solution of the congestion-control game given by (38) satisfies

$$\sum_{j:l\in j} x_j \leq C_l - \bar{\xi}_l.$$

Proof: Similar to the proof in Proposition 3. □

Proposition 5 states that the expected value of the total flow (including the perturbations ξ_l) into a link is less than some desired level if the marking level is chosen according to (39). However, the arrival rate will have a small variance that is typically proportional to the increase and decrease parameters of the system [18]. In a real network, to account for this and for the unmodeled dynamics of window flow control, one has to operate the network at slightly less than full utilization if we desire very low-loss operation.

C. ECN Marks and Random Losses

In addition to nearly eliminating congestion-related losses, ECN marks can also be used to distinguish between congestion-related losses and random losses. Thus, it could help eliminate the deleterious effect of random losses on end-to-end rate and congestion control schemes. If there are no losses due to congestion, or at least if the congestion-related losses are a small fraction of random losses, then with negligible error all losses can be attributed to random losses. In other words, use only marks to reduce the rate of transmission and assume that all lost packets are due to noncongestion-related phenomena. This would be reasonable if the marking level is chosen to nearly eliminate congestion-related losses. Of course, some marked packets could also be lost to random losses. Thus, interpreting z_r as the rate at which unmarked packets are received by user r, the congestion control scheme can be written as

$$\dot{x}_r = \kappa_r \left(\frac{w_r}{\beta_r} - \beta\alpha_r(x_r - z_r)\right), \quad r \in \mathcal{R}_1 \tag{40}$$

and

$$\dot{x}_j = \kappa_j \left(\frac{\nu_j w_j}{\beta_j} - \beta \alpha_j x_j^{\nu_j} (x_j - z_j) \right), \quad j \in \mathcal{R}_2. \tag{41}$$

This leads to the following proposition.

Proposition 6: Consider the congestion control scheme given in (40), (41) and suppose that, for each link l, $\tilde{C}_l$ has been chosen such that there are no losses. Then the users' rates $\{x_r\}$ converge to the unique solution of

$$\max_{\{x_r\}} \sum_r \left(\frac{w_r}{\beta_r \alpha_r} \right) U_r(x_r) - \beta \sum_{l \in \mathcal{L}} \int_0^{\sum_{l \in j} x_j} \frac{(x - C_l)^+}{x} dx \tag{42}$$

subject to $x_r \geq 0, \forall r$. □

When $\{\alpha_r\}$ are small, we see from (42) that the effect of random losses are negligible despite the fact we may be losing some congestion indication signals by ignoring lost marks. In contrast, from Proposition 2, when losses are the indicators of congestion, even small values of $\{\alpha_r\}$ have a significant impact on the performance of the congestion controllers when β is large.

Remark 4: A window flow control implementation of (40), (41) obtained by ignoring the round-trip delays can be interpreted as a discrete-time version of the rate control algorithm that converges to the solution of

$$\max_{\{x_r\}} \sum_{r \in \mathcal{R}_1} \left(\frac{w_r}{\beta_r \alpha_r d_r} \right) U_r(x_r) + \sum_{j \in \mathcal{R}_2} \left(\frac{w_j}{\beta_j \alpha_j d_j^{\nu_j + 1}} \right) U_j(x_j) - \beta \sum_{l \in \mathcal{L}} \int_0^{\sum_{l \in j} x_j} \frac{(x - C_l)^+}{x} dx. \tag{43}$$

□

D. Adaptive Algorithm for Setting the Marking Level

According to Proposition 3, computing $\tilde{C}$ at each node requires the node to know the number of flows passing through it and the utility function of each user, or alternately the congestion control scheme used by each user. This is not practically feasible. From Proposition 3, it is clear that one can maintain the same $\tilde{C}$ independent of the number of users, provided β is scaled appropriately with N_l. Since β may be interpreted as price-per-mark, the price has to be modified according to the number of users in the network. This essentially amounts to time-of-day pricing. During peak hours, a larger price is charged than during off-peak hours. This requires a rough estimate of the number of users and their utility characteristics as a function of the time of day. Any uncertainty in this can be handled using an adaptive algorithm to estimate the appropriate marking level.

We propose the following adaptive algorithm for setting the marking level at link l:

$$\frac{d\tilde{C}}{dt} = \begin{cases} \alpha(\gamma C_l - x), & 0 < \tilde{C}_l < \gamma C_l \\ \max\left(0, \alpha(\gamma C_l - x)\right), & \tilde{C}_l = 0 \\ \min\left(0, \alpha(\gamma C_l - x)\right), & \tilde{C}_l = \gamma C_l \end{cases}$$

where x is the total flow through link l and α is a step-size parameter which can be adjusted to regulate how fast $\tilde{C}_l$ is changed. The basic idea behind the above algorithm is to attempt to regulate the total flow to γC_l: thus, $\tilde{C}_l$ is increased when x is less than γC_l and it is decreased when x is larger than γC_l. We note that the above algorithm can be used with or without time-of-day pricing or even without interpreting β as a price parameter, but simply treating it as a congestion control parameter. Simulations indicate that a discretized version of this update equation converges for sufficiently small values of α under very general conditions. The only assumption required is that a positive $\tilde{C}_l$ given by Proposition 3 exists. Clearly, for a fixed β, if the number of users is very large, then there may not exist a marking level that ensures loss-free operation. Thus, increasing the available capacity through provisioning or increasing β are the only options to ensure loss-free service. A variation of this algorithm is shown to be semi-globally exponentially stable in [17].

VI. Simulations and Numerical Results

In this section, we perform four different experiments using the software package *ns-2*. In the first experiment, we simulate various window flow control schemes. This is a detailed simulation taking into account finite packet sizes, round-trip delay, and window-flow control, and is designed to study the accuracy of the fluid model predictions for different utility functions with packet-level implementations of the congestion controller. In the second experiment, we consider ECN marks and random losses in the model. We then study the adaptive algorithm for setting the marking level such that the resulting steady-state throughput is less than the node's capacity, thus ensuring loss-free operation. Finally, we study the performance of the algorithms in the presence of short flows.

A. Experiment 1: Packet Model Simulations With Different Congestion Controllers

We use a packet model with round-trip delays to simulate the window flow control. The simulations were done using *ns-2*. Due to space limitations, we present only one among a set of simulations that we have conducted to validate our results.

Consider the network shown in Fig. 1. The network consists of nine nodes. Nodes $n0$ and $n1$ are connected by a 2-Mb/s link with a one-way propagation delay of 10 ms. (This roughly corresponds to a distance of 2000 km.) The reverse path, however, has a bandwidth of 1000 Mb/s, also with a delay of 10 ms. The reverse path has a higher bandwidth to prevent acks from getting lost. Nodes $n1$ and $n2$ are connected by a 1-Mb/s link which has a delay of 10 ms. In this case also, the reverse link has a bandwidth of 1000 Mb/s and a delay of 10 ms. Nodes $n0$, $n1$, and $n2$ can be thought of as the core network with the rest being access nodes. All other nodes are connected by links of 1000 Mb/s and have delays of 0.005 ms in both the directions. Thus, end nodes that are two hops away in the core network are separated by roughly 4000 km. The idea is to make the links between $n0$ and $n1$ and between $n1$ and $n2$ the bottleneck links. All other links are access links and, hence, have a much higher bandwidth and much lower delay than the bottleneck links.

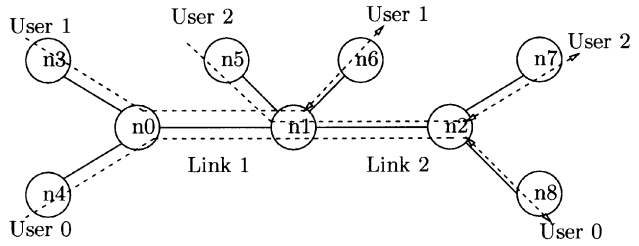

Fig. 1. Network used for packet simulations in *ns-2*.

User 0 traverses the links connecting nodes $n4$ and $n8$ and passes through both the bottleneck nodes. User 1 is between nodes $n3$ and $n6$ and user 2 is between nodes $n5$ and $n7$. Therefore, users 1 and 2 pass through only one bottleneck node. The queue size at each node is limited to 100 packets. The packet sizes are taken to be 32 bytes long while the acks are 16 bytes long, *though the packet sizes and acks can be taken to be arbitrary as long as we scale the bandwidth appropriately*. All the flows are assumed to experience no random losses. We also let $w_1 = w_2 = w_3 = 1$ and $\beta = 0.5$. We use the utility function $-1/\sqrt{x}$ for user 0, $\log x$ for user 1, and $-1/x$ for user 2.

It is known that using a simple FIFO queue with *drop-tail* mechanism results in synchronization-related problems that result in poor performance of window flow control mechanisms. Therefore, random scheduling mechanisms like RED [5] have been developed to combat this problem. Since, in our simulations we are trying to approximate the fluid model in which losses are proportionally distributed among all users, any mechanism which randomizes the drop (like RED) at the queue will work. However, for our simulation purposes *drop-front* FIFO queuing works well and in all our simulations we assume that all queues employ a *drop-front* scheduling mechanism.

We now implement the window flow control scheme given by (12) and (13) with increments and decrements measured in units of packets. From our network model, user 0 has a round-trip delay of approximately 0.04 s, while users 1 and 2 have a round-trip delay of 0.02 s, ignoring the buffering at each node. For round-trip delays in the window flow control scheme, we use the values 2 for user 0 and 1 for users 1 and 2. Thus, we are normalizing time such that 1 unit is 0.02 s. The throughputs of the users should thus be measured in packets per 0.02 seconds. Also, since the rates are measured in packets per (0.02) second, the bandwidth of the link between nodes $n0$ and $n1$ becomes 156.25 packets per (0.02) second and the bandwidth of the link between nodes $n1$ and $n2$ becomes 78.125 packets per (0.02) second.

The steady-state rates of this system should be equal to the optimum rates that solve the optimization problem given in Proposition 5. The rates obtained by solving the nonlinear program are

$$x_0 = 16.56, \quad x_1 = 139.69, \quad x_3 = 61.57$$

whereas the average rates obtained in the simulation are

$$x_0 = 16.81, \quad x_1 = 140.83, \quad x_2 = 60.34.$$

It can be seen that the rates obtained by solving the optimization problem and the rates obtained in the simulation match each other closely. The window sizes of the three flows are shown in Fig. 2. While the window size fluctuates, the average behavior of the windows is close to the predicted values after a very short initial transient period.

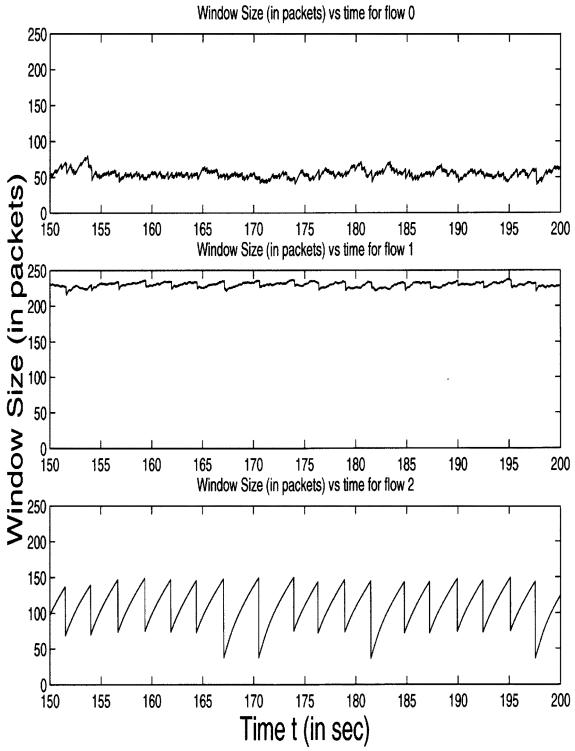

Fig. 2. Window size evolution for all users.

B. Experiment 2: ECN Marks and Random Losses

In this experiment, we will use a packet-level implementation to simulate the effects of random loss on the performance of the users using the $-1/x$ utility function. We will then provide results which show that with ECN marking and the users reacting only to marks, the performance improves dramatically as compared to the case of using losses for congestion control.

Consider a single node with three users having the same utility function, $-1/x$. The bandwidth at the node is 1 Mb/s and the queue size at the node is assumed to be 40 packets. The round trip delay of each user is assumed to be 40 ms. This would roughly correspond to the source and destination being 4000 km apart. We assume a random loss probability of 0.05 for each of the users.

In the first scenario, packet losses are indicators of congestion and the users react to packet loss. In the second scenario, the users use ECN marks as indicators of congestion in the network and attribute all packet losses to random losses. Therefore, the system decreases its window on receiving marks, but does not do so with packet losses. The marking level $\tilde{C}$ is chosen to be 0.99 C. A marking level of $\tilde{C}$ corresponds to using a virtual queue whose capacity is $\tilde{C}$ and marking packets in the real queue when the virtual queue exceeds its buffer capacity. While the idea of a virtual queue is used in [6], our implementation does not continue to mark till the virtual queue is empty. Fig. 3 shows the throughput of each user for a duration of 200 s for each of the above two scenarios.

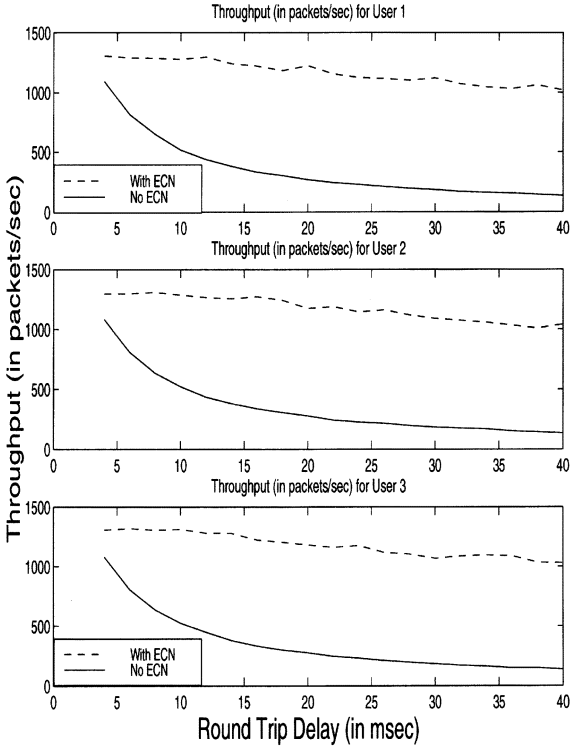

Fig. 3. Throughput for all users with and without ECN marking.

From Fig. 3, we can see that the throughput of an user using ECN marks is much better (about five times) than a user without ECN marks. This improvement in performance is due to the user attributing all losses to random losses in the network. Since the marking level makes sure that there are very few congestion related losses, most of the packet losses seen by the user are indeed due to random losses.

C. Experiment 3: Adaptive Estimation of Marking Level With 300 Sources

In the previous experiment, we saw that with a suitable $\tilde{C}_l$, we can have improved performance even when there are random losses in the system. However, the expression for $\tilde{C}_l$ depends upon N_l, the number of users using link l, which is not available to the node. In Section V-D, we gave an update equation for determining the value of $\tilde{C}_l$ at the node. In this section, we will provide some simulation results which indicate that it is possible to estimate $\tilde{C}_l$, without the knowledge of the number of flows through the node. We perform a packet-level simulation using ns for this purpose. From the update equation, we see that $\tilde{C}$ is updated as a function of the difference between γC and the total arrival rate λ. In a packet-level simulation, we calculate the total arrival rate at the node every K packets that come into the node. Note that unlike the discretized version of the update equation, this does not depend on any measurement interval. Therefore, $\tilde{C}$ is updated every K packets received at the node.

We consider the network shown in Fig. 1, but with 300 users, in three different classes. Class 1 consists of users that traverse both Links 1 and 2, while Class 2 users use only Link 1 and Class 3 users use only Link 2. Each class has 100 users. Within each class, 50 users have a $\log x$ utility function, and the remaining 50 users have a $-1/x$ utility function. Link 1 has a capacity of 2 Mb/s and a delay of 10 ms. Link 2 similarly has a capacity of 1 Mb/s and a delay of 10 ms. Thus, users in Class 1 have a round-trip delay of 40 ms, while users in Class 2 and Class 3 have a round-trip delay of 20 ms ignoring the queueing delays and we let $K = 1000$ and $\gamma = 1.0$. Figs. 4 and 5 show the evolution of $\tilde{C}_1$ and $\tilde{C}_2$ with time.

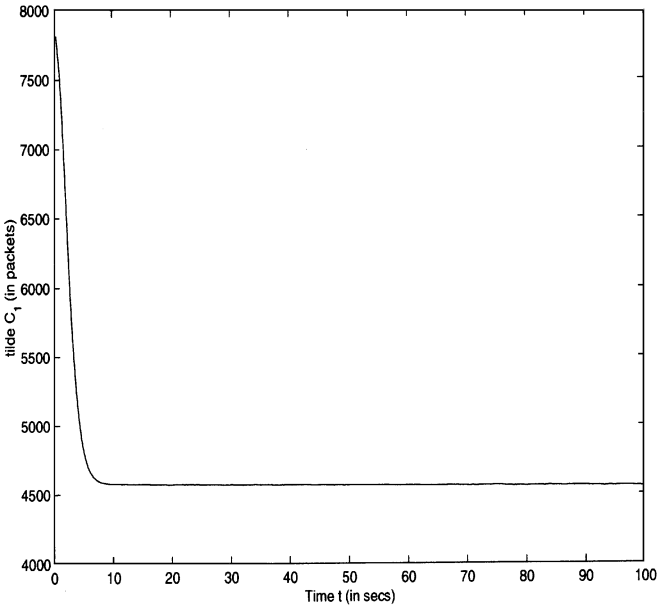

Fig. 4. Adaptive marking level $\tilde{C}_1$ (in packets per second) for link 1 in experiment 3.

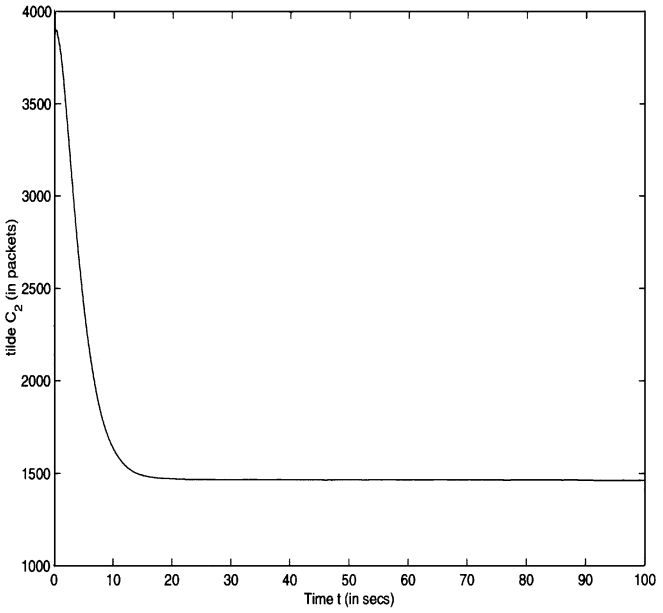

Fig. 5. Adaptive marking level $\tilde{C}_2$ (in packets per second) for link 2 in experiment 3.

From Figs. 4 and 5, we can see that $\tilde{C}_1$ and $\tilde{C}_2$ converge to their steady-state values quickly. More importantly, we also observed in the simulation that none of the users experience any packet drops after a short initial transient period. This can further be controlled by varying the utilization factor γ. Fig. 6 shows the window size of a typical user from a user class with time.

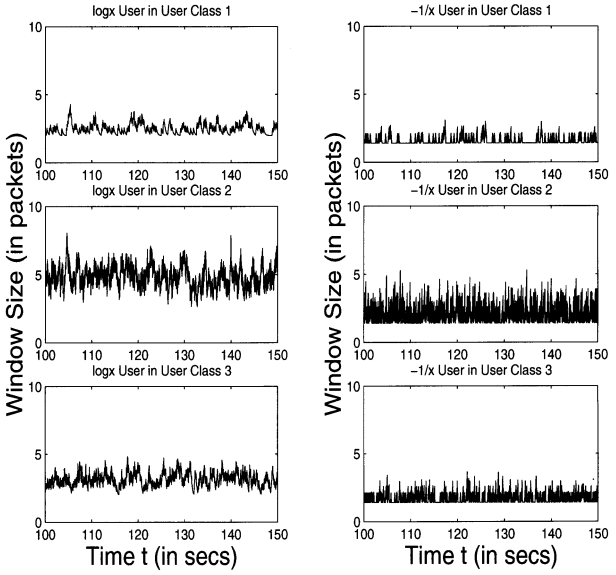

Fig. 6. Window sizes as a function of time for a typical user from each user class in experiment 3.

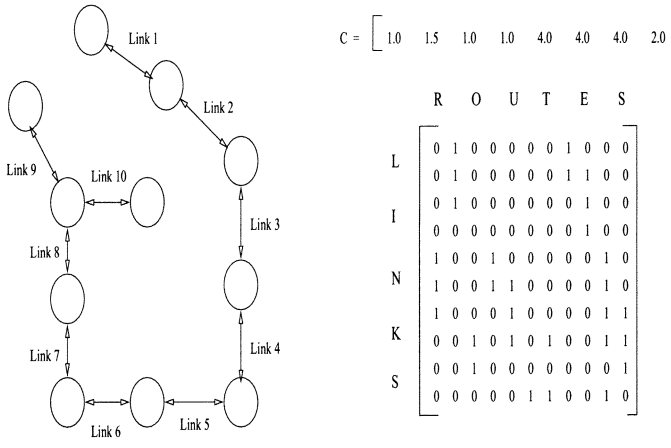

Fig. 7. Network along with the capacity vector and the routing matrix used in experiment 4.

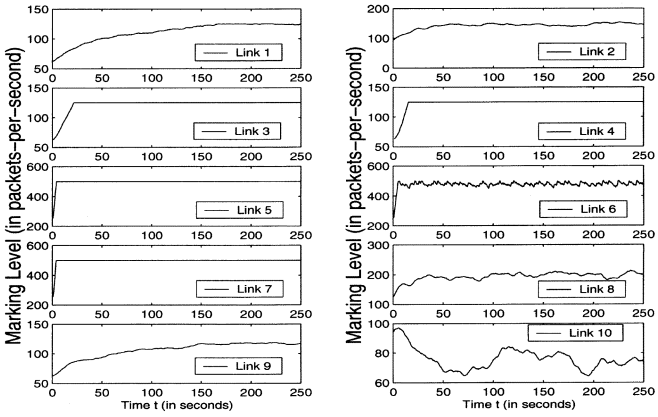

Fig. 8. Adaptive marking level (in packets-per-second) for each link of the network in experiment 4.

D. Experiment 4: Adaptive Estimation of Marking Level With Short Flows

In the previous experiment, we considered a scenario in which the flows are assumed to be present for the entire duration of the simulation. In this experiment, we will introduce some short flows along each route in addition to the long flows that will be present for the entire duration of the simulation. Packet sizes here are assumed to 1000 bytes.

Consider the network shown in Fig. 7. The routing or the incidence matrix and the capacity vector is also shown in the figure. Each link is also assumed to have a one-way propagation delay of 20 ms. Short flows are generated in a Poisson manner with an arrival rate (of the flows, not the packet arrival) of one flow per second per route. The flow lengths are chosen to be Pareto distributed with a mean of ten packets and truncated to 20 packets. Fig. 8 shows the evolution of the marking level at each link in the network for a duration of 250 s.

REFERENCES

[1] D. Bertsekas and R. Gallager, *Data Networks*. Englewood Cliffs, NJ: Prentice-Hall, 1987.

[2] L. Breslau and S. Shenker, "Best-effort versus reservations: A simple comparative analysis," *Comput. Commun. Rev.*, vol. 28, pp. 3–16, Sept. 1998.

[3] S. Floyd, "TCP and explicit congestion notification," *Comput. Commun. Rev.*, vol. 24, pp. 10–23, Oct. 1994.

[4] S. Floyd and K. Fall, "Promoting the use of end-to-end congestion control in the Internet," *IEEE/ACM Trans. Networking*, vol. 4, pp. 458–472, Aug. 1999.

[5] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. Networking*, vol. 1, pp. 397–413, Aug. 1993.

[6] R. J. Gibbens and F. P. Kelly, "Distributed connection acceptance control for a connectionless network," presented at the 16th Int. Teletraffic Congr., Edinburgh, U.K., June 1999.

[7] ——, "Resource pricing and the evolution of congestion control," *Automatica*, vol. 35, pp. 1969–1985, 1999.

[8] S. J. Golestani and S. Bhattacharyya, "A class of end-to-end congestion control algorithms for the Internet," presented at the Int. Conf. Network Protocols, Oct. 1998.

[9] P. Hurley, J.-Y. Le Boudec, and P. Thiran, "A note on the fairness of additive increase and multiplicative decrease," presented at the 16th Int. Teletraffic Congr., Edinburgh, U.K., June 1999.

[10] V. Jacobson, "Congestion avoidance and control," *Comput. Commun. Rev.*, vol. 18, pp. 314–329, Aug. 1988.

[11] F. P. Kelly, A. Maulloo, and D. Tan, "Rate control in communication networks: shadow prices, proportional fairness and stability," *J. Oper. Res. Soc.*, vol. 49, pp. 237–252, 1998.

[12] F. P. Kelly, "Charging and rate control for elastic traffic," *Eur. Trans. Telecommun.*, vol. 8, pp. 33–37, 1997.

[13] ——, "Mathematical modeling of the Internet," in *Mathematics Unlimited—2001 and Beyond*, B. Engquist and W. Schmid, Eds. Berlin, Germany: Springer-Verlag, 2001, pp. 685–702.

[14] A. Kumar, "Comparative performance analysis of versions of TCP in a local network with a lossy link," *IEEE/ACM Trans. Networking*, vol. 6, pp. 485–498, Aug. 1998.

[15] S. Kunniyur and R. Srikant, "Fairness of congestion avoidance schemes in heterogeneous networks," presented at the 16th Int. Teletraffic Congr., Edinburgh, U.K., June 1999.

[16] ——, "Analysis and design of an adaptive virtual queue algorithm for active queue management," in *Proc. ACM SIGCOMM*, San Diego, CA, Aug. 2001, pp. 123–134.

[17] ——, "A time-scale decomposition approach to adaptive explicit congestion notification (ECN) marking," *IEEE Trans. Automat. Contr.*, vol. 47, pp. 882–894, June 2002.

[18] H. J. Kushner and D. S. Clark, *Stochastic Approximations for Constrained and Unconstrained Systems*. New York and Berlin, Germany: Springer-Verlag, 1978.

[19] T. V. Lakshman and U. Madhow, "The performance of TCP/IP for networks with high bandwidth-delay products and random loss," *IEEE/ACM Trans. Networking*, vol. 5, pp. 336–350, June 1997.
[20] J. R. Li, D. Dwyer, and V. Bharghavan, "A transport protocol for heterogeneous packet flows," presented at the *IEEE INFOCOM*, New York, Mar. 1999.
[21] S. H. Low and D. E. Lapsley, "Optimization flow control, I: Basic algorithm and convergence," *IEEE/ACM Trans. Networking*, vol. 7, pp. 861–875, Dec. 1999.
[22] J. K. MacKie-Mason and H. R. Varian, "Pricing congestible network resources," *IEEE J. Select. Areas Commun.*, vol. 13, pp. 1141–1148, Sept. 1995.
[23] L. Massoulie and J. Roberts, "Bandwidth sharing: objectives and algorithms," in *Proc. IEEE INFOCOM*, New York, Mar. 1999, pp. 1395–1403.
[24] M. Mathis, J. Semke, J. Mahdavi, and T. Ott, "The macroscopic behavior of the TCP congestion avoidance algorithm," *Comput. Commun. Rev.*, vol. 27, 1997.
[25] V. Misra, W. Gong, and D. Towsley, "A fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED," in *Proc. ACM SIGCOMM*, Stockholm, Sweden, Sept. 2000.
[26] J. Mo and J. Walrand, "Fair end-to-end window-based congestion control," *IEEE/ACM Trans. Networking*, vol. 8, pp. 556–567, Oct. 1998.
[27] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: a simple model and its empirical validation," in *Proc. ACM SIGCOMM*, 1998.
[28] J. Waldby, U. Madhow, and T. V. Lakshman, "Total acknowledgment: a robust feedback mechanism for end-to-end congestion control," *Proc. ACM Sigmetrics*, 1998.

Srisankar Kunniyur received the B.E. degree in electrical and electronics engineering from B.I.T.S., Pilani, India, in 1996 and the M.S. and Ph.D. degrees in electrical engineering from the University of Illinois at Urbana-Champaign in 1998 and 2001, respectively.

He is currently an Assistant Professor of electrical engineering at the University of Pennsylvania, Philadelphia. His research interests include design and performance analysis of high-speed communication networks, wireless networks and ad-hoc networks, and congestion control and pricing in heterogeneous networks.

R. Srikant (M'91–SM'01) received the B.Tech. degree from the Indian Institute of Technology, Madras, in 1985 and the M.S. and Ph.D. degrees from the University of Illinois at Urbana-Champaign in 1988 and 1991, respectively, all in electrical engineering.

He was a Member of Technical Staff with AT&T Bell Laboratories from 1991 to 1995. He is currently with the University of Illinois, where he is an Associate Professor in the Department of Electrical and Computer Engineering and a Research Associate Professor in the Coordinated Science Laboratory. He was an Associate Editor of *Automatica*. His research interests include communication networks, stochastic processes, queueing theory, information theory, and game theory.

Dr. Srikant is currently on the editorial boards of the IEEE/ACM TRANSACTIONS ON NETWORKING and the IEEE TRANSACTIONS ON AUTOMATIC CONTROL. He was the Chair of the 2002 IEEE Computer Communications Workshop, Santa Fe, NM.