Trading Precision for Stability in Congestion Control with Probabilistic Packet Marking

Jonathan Shapiro
Dept. of Computer Science and Eng.
Michigan State University
East Lansing, MI 48824-1226
Email: jshapiro@cse.msu.edu

C. V. Hollot
Dept. of Electrical and Computer Eng.
University of Massachusetts
Amherst, MA 01002
Email: hollot@ecs.umass.edu

Don Towsley
Dept. of Computer Science
University of Massachusetts
Amherst, MA 01002
Email: towsley@cs.umass.edu

Abstract

In pricing-based congestion control protocols it is common to assume that the rate of congestion feedback from the network is limited to a single bit per packet. To obtain a precise estimate of available bandwidth (as summarized by the congestion price) under the single-bit constraint, a session must consider feedback contained in a number of recently received packets. As more packets are considered, however, the estimate includes increasingly older information about the network state. We study this tradeoff between the quality and timeliness of feedback using control-theoretic approach, modelling the 'memory' incorporated into the price estimate as additional feedback delay. We show through analysis that obtaining arbitrary precision in the estimated price causes control instability, making it more difficult for a session to track its targeted optimal rate. Through continuous-time simulation of our model and packet-level simulations, we find that crude estimates of congestion price based on very few packets can yield good performance while allowing the session to operate far away from the boundary of instability. We also investigate the impact of estimation bias on protocol performance, showing that protocols use a form of integral control can compensate for biased price estimates.

1 Introduction

A recurring feature in optimization-based congestion control (OBCC) protocols is that congestion prices must be computed by individual links in the network and communicated to sessions [7, 2, 12, 13, 11]. In particular, each session must learn the sum of link prices along its path in order to make globally optimal rate control decisions. Network-layer protocols typically devote a very small number of bits in the packet header to carry congestion feedback. In IP, for example, the proposed explicit congestion notification (ECN) mechanism essentially makes a single bit available to carry congestion state along the forward path [18]. Recent research in OBCC has focused on the use of a single bit to carry price information. In particular, it has been shown that routers can encode prices by probabilistically setting a single ECN bit[1] in such a way that the end-to-end marking probability encodes the sum of prices along a path—a technique known as *probabilistic packet marking* (PPM). Using PPM, receivers can estimate the total price along a session path, by recording the fraction of marked packets.

Optimization-based congestion control protocols include a component running at each link that sets the link's price and marks packets, and a component executed at end-hosts that estimates the total price and sets the transmission rate accordingly. In an important class of protocols (e.g., [14, 2]) links combine an iterative algorithm for setting prices with probabilistic packet marking for encoding prices.

Prior research in PPM [2, 3, 1] has established the existence of exact and approximate PPM algorithms and has studied their properties in isolation. That is, the properties of PPM price estimators are well-understood for the case

[1] In the rest of this paper, we will adopt the term *explicit congestion notification* (ECN) to generically refer to any mechanism that reserves bits of the packet header to communicate congestion state.

of static link prices. In a real congestion control protocol, however, PPM is one of several components of an integrated mechanism. Other components are the link pricing algorithm, which dynamically adjusts prices in response to offered load, and the sender rate adaptation algorithm, which adjusts the load in response to the price. Taken together, these components form a dynamic system characterized by fluctuating prices.

It is reasonable to expect that the static properties of various PPM algorithms will be reflected in some aspects of congestion control including stability and the ability to track an optimal rate allocation. However, results from the static case are insufficient to fully understand the dynamic behavior of PPM. For example, in the case of static prices all PPM agorithms have the property that the price estimation error decreases as more samples of the ECN bit are collected. This fact alone, however, does not tell us how many samples *should* be collected for each rate control decision. Furthermore, although some PPM algorithms are known to return a biased price estimate, the impact of this bias in a dynamic setting is unclear.

This research makes several novel contributions.

- We develop a model of optimization-based congestion control that relates the static properties of a PPM algorithm to the performance of the congestion control protocol in which it is embedded.
- We characterize a fundamental tradeoff between the precision of the price estimate and the stability of the control loop.
- Through analysis and simulation, we show that coarse estimates of the session price based on a small number of packets—on the order of ten—can lead to better protocol performance than precise estimates.

The rest of this paper is organized as follows. In Section 2, we present an overview of the three known PPM algorithms and discuss their properties for static prices. We then present an analytical model of closed-loop congestion control in Section 3. Section 4 presents a linearized analysis of our model, illustrating the tradeoff between estimation precision and the stability of the loop. This section also explores this tradeoff—both through analysis and simulation—for unbiased and biased price estimates. We validate our model using packet-level simulations of a realistic protocol in Section 6. Sections 7 and 8, respectively, discuss related work and conclude the paper.

2 Probabilistic Packet Marking

Consider a set of links $1,\ldots,n$ forming an end-to-end path from a source to a receiver. Associated with each link i is a non-negative price p_i. Let $q_n = \sum_{i=1}^{n} p_i$ denote the sum of prices along the path. As data packets traversing the path arrive at a receiver, the receiver must determine q_n and provide this quantity as feedback to the sender. We assume that a single bit in the packet header is available for the purpose of communicating this sum. The problem of path price estimation is to design a *marking algorithm*—that is, some strategy for computing the price bit X_i at each link i—to allow the receiver to estimate the total price q_n.

A class of practically implementable protocols collectively known as *probabilistic packet marking* (PPM) has been identified in recent literature [2, 1]. In PPM, each link randomly and independently sets the price bit in each arriving packet in such a way that the total path price is encoded in the probability that the bit is set when it arrives at the receiver. Marking decisions at each link are based on locally available information and, since packets are marked independently, no per-flow state is required.

2.1 Random Early Marking

The Random Early Marking (REM) scheme proposed by Athuraliya et al. [2] was, as far as we are aware, the earliest proposed PPM algorithm for price estimation. In REM, the designer selects some base $\phi > 1$. The initial price bit X_0 is set to zero. The i^{th} link, where $i \geq 1$, sets the price bit to 1 with probability $1-\phi^{-p_i}$. The bit arriving at the receiver, X_n remains unmarked with probability

$$\prod_{i=1}^{n} \phi^{-p_i} = \phi^{-\sum_{i=1}^{n} p_i},$$

and is set to one otherwise. Hence marking probability is given by the expectation

$$\mathbf{E}[X_n] = 1-\phi^{-q_n}.$$

To estimate the total price q_n the receiver first collects N packets, obtaining N independent samples of the price bit $X_n^{(1)}, X_n^{(2)}, \ldots, X_n^{(N)}$. The receiver then computes the sample mean of the price bit

$$\overline{X} = (\sum_{j=1}^{N} X_n^{(j)})/N, \tag{1}$$

and estimates q_n to be

$$q_n \approx -\log_{\phi}(1-\overline{X}) \tag{2}$$

.

It can be shown [1] that REM is a biased estimator for any finite number of samples N. However, it is a consistent estimator since its bias converges to zero as N approaches infinity. Additional bias is introduced by practical implementations of REM, which must carefully handle the case where all of the samples are marked, because of the singularity at $\overline{X} = 1$ in the REM price function (2). Assuming that individual link prices are tuned so as to avoid end-to-end marking probabilities close to one, the probability that $\overline{X} = 1$ becomes vanishingly small as the number of samples N increases. In the interest of protocol performance, however, it may be desirable to limit the number of samples required to make a rate control decision. In this case the handling of this boundary condition has a non-negligible effect on the price estimate.

2.2 Random Additive Marking

An alternative PPM algorithm, proposed by Adler, et al [1], is known as Random Additive Marking (RAM). RAM is feasible if the range of each link price p_i is restricted such that $0 \leq p_i \leq 1$, and if the step index i is assumed to be available for local computation at the i^{th} link. Again, the price bit X_0 is initialized to zero. At each step $i \geq 1$, link i leaves the price bit unchanged ($X_i = X_{i-1}$) with probability $(i-1)/i$. With probability p_i/i the link sets the bit to 1 and sets it to 0 otherwise. The resulting X_n is a 0-1 random variable with

$$\mathbf{E}[X_n] = \sum_{i=1}^{n} p_i/n.$$

To estimate the total price, the receiver collects N samples of the price bit and computes the sample mean $\overline{X}$ as in (1). In practical implementations, the semantics of time-to-live field in the IP header can be used to infer the step index at each hop. The receiver can similarly determine n and thus obtain q_n. It is readily shown that RAM computes an unbiased estimate of the total price.

2.3 Approximate RAM

It has been shown [1] that only REM, RAM, and linear transformations of the two satisfy asymptotic convergence of the estimated price to the true price as the N approaches infinity. However, a scheme based on a widely used approximation has also been identified.

Let link prices p_i be bounded by $[0,1]$ and let each link independently set the price bit to 1 with probability p_i, yielding an end-to-end marking probability of

$$\mathbf{E}[X_n] = 1 - \prod_{i=1}^{n}(1 - p_i)$$

If the values of p_i are small, then we may approximate their sum as

$$q = \sum_{i=1}^{n} p_i \approx 1 - \prod_{i=1}^{n}(1 - p_i). \tag{3}$$

This observation suggests that given a set of initially normalized link prices $p_i \in [0,1]$, we may *re–normalize* the entire set with the transformation

$$p_i \rightarrow \alpha p_i, \tag{4}$$

where $0 < \alpha \leq 1$. This simple marking scheme yields an end-to-end marking probability of approximately αq. Dividing the estimated marking probability by α, we obtain the true path price. For a suitably chosen normalization constant α the probabilities will combine approximately additively; therefore we refer to this scheme as *approximate RAM* (A-RAM).

The A-RAM approximation introduces a non-zero bias

$$B_a = \sum_{i=1}^{n} \alpha p_i - \left(1 - \prod_{i=1}^{n}(1 - \alpha p_i)\right) \tag{5}$$

We observe that B_a is independent of N the number of samples collected and is thus a constant bias for a fixed path of links.

3 Closing the Control Loop

Existing research into PPM algorithms has assumed constant prices on each link drawn independently from a fixed distribution. We now relax these assumptions and investigate the behavior of closed-loop congestion control protocols that make use of PPM.

REM, RAM and A-RAM all compute an estimate of the total path price with error due to two possible contributions—variance and bias. The variance is an intrinsic byproduct of estimating the end-to-end marking probability with a finite number of samples, whereas the bias depends on the particular estimation algorithm used. The total estimation error can be reduced by collecting more samples of the price bit before making a price estimate, which reduces the variance. In the case of a biased but consistent estimator like REM, collecting more samples also reduces bias. Because rate control protocols require ongoing price estimation, a natural way to integrate PPM algorithms into protocols is to continuously update the price estimate based on a moving average of price bits in the N most recently received packets.

It is important to understand, however, that although the use of a moving average allows for rate adjustment at small time scales, it can still allow outdated feedback to influence the sending rate. The moving average estimate summarizes, in a statistical sense, the price history over the last N packets. Reducing price estimation error by increasing the estimation window size (i.e. increasing N) comes at the expense of incorporating increasingly delayed information into the estimate. Thus, the moving average effectively introduces additional feedback delay on top the existing round-trip delay, which can adversely affect protocol stability. A well-designed protocol based on a moving average must strike a balance between precision and stability.

To evaluate this tradeoff quantitatively we must first define a measure of protocol performance. Let x_{opt} denote the target optimal rate for the source—the equilibrium rate of the congestion control scheme given a price signal with no error or delay. The actual rate x achieved in operation will vary around an equilibrium that is influenced by the error in the price estimate. The magnitude of its deviation from the target rate is influenced both by error in the price estimate and by feedback delay. A natural performance measure, is the mean squared error (MSE) of the achieved rate,

$$M_x = E[(x_{opt} - x)^2].$$

We refer to M_x as the *source-rate MSE*. As we will see, this metric captures the effects of both estimation error and control instability. We will be interested in how M_x varies as a function of N for protocols using biased and unbiased price estimators.

3.1 System Model

We model the network as a single link used by a single source. We will find it helpful to introduce some notation based on the way a real PPM scheme would operate in this simple network. The link computes a congestion price q which it communicates to the source by encoding q as a marking probability

$$p = F(q).$$

The source computes an estimate $\hat{p}$ of the end-to-end marking probability based on a sliding window of the N most recent samples of the ECN bit, from which it computes an estimate of the price

$$\hat{q} = G(\hat{p}).$$

The functional forms of F and G depend on the particular PPM algorithm used.

Instead of modelling the exact behavior of the PPM protocol, we assume that the network provides the exact value of the instantaneous price, to which the estimation process adds both a bias and a variance as shown conceptually in Figure 1. We may select the functional dependence of the bias and variance on N and q to model various different PPM algorithms. Since the price estimation window represents a finite history of the link price, Paganini, et al [17] have argued that the price estimation technique adds an effective delay to the control loop over and above any propagation delay.[2] We model this effect as a pure delay that is increasing in N but decreasing in the transmission rate, since the time to acquire N packets decreases as the rate increases.

[2]We assume that the network is operated with negligible queueing delay, as OBCC protocols are expected to be configured so as to maintain small queues at routers.

We assume that link prices are bounded within $[0,1]$, because it is only in this regime that all three PPM algorithms—REM, RAM, and A-RAM—are possible. We must therefore either determine an appropriate constant with which to normalize the true prices or adopt a link pricing scheme where prices are explicitly bounded by $[0,1]$. Fortunately, such an explicitly bounded scheme exists in the *adaptive virtual queue* (AVQ) scheme of Kunniyur and Srikant [12]

$$q(x) = \frac{(x - \tilde{c})^+}{x}, \tag{6}$$

where $\tilde{c}$ is known as the *virtual capacity*. Virtual capacity is itself adapted over a slower time scale than the link price according to

$$\dot{\tilde{c}} = \alpha(\gamma c - x), \tag{7}$$

where c is the true link capacity, γ is the desired utilization of the link, and α is a damping parameter. The effect of this slow adaptation of $\tilde{c}$ is to stabilize source rates around a desired link rate γc. In the analysis that follows, we will assume without loss of generality that $\gamma = 1$.

To further simplify the analysis, we assume a weighted logarithmic utility function for the source

$$u(x) = w \log x,$$

which implies the following primal gradient descent algorithm for adjusting the source rate:

$$\dot{x} = \kappa\left(w - x(t-\tau)\hat{q}(x(t-\tau))\right), \tag{8}$$

where τ is the round-trip delay [15]. The round-trip delay is the sum of a fixed propagation delay τ_0 and the effective delay introduced by the price estimation window. As we have explained above, this effective delay is equivalent to the time required to collect $N/2$ samples of the price bit (see Paganini, et al [17]). Thus

$$\tau = \tau_0 + \frac{\beta N}{2x},$$

where β is the number of bytes per packet and x is the rate.

The estimated link price $\hat{q}$ is equal to the true link price q plus an error term η,

$$\hat{q}(x) = q(x) + \eta$$

We model the estimation error η as Gaussian white noise with mean $\overline{\eta}(N,q)$ given by the bias of the price estimation technique and a variance $\sigma^2_\eta(N,q)$ related to the PPM variance, as we will discuss below. Although the use of Gaussian white noise lends our model mathematical tractability,

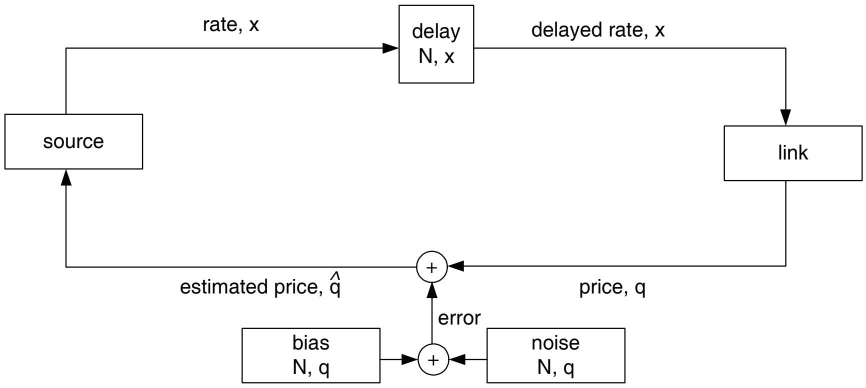

Figure 1. Simple model of a congestion control loop using PPM for price estimation. Functional dependencies on N, q, and x are indicated in italics.

it does introduce modelling error. We discuss this issue in more detail in Section 6.

To model an unbiased estimator like RAM we set the mean $\bar{\eta} = 0$; for biased estimators like A-RAM and REM, $\bar{\eta} \geq 0$. Note that for a biased but consistent estimator like REM, considering possibly small estimation window sizes N means that we may not be operating in the asymptotic region $N \to \infty$. We can not, therefore, neglect the effect of bias on the equilibrium rates *a priori*.

Setting the estimation variance is more complicated since we are modelling the inherently discrete process of repeated price estimation as a continuous signal—Gaussian white noise added to a continuous price signal. We want the variance of the noise to have the same magnitude and functional dependence on N as the that of the modelled PPM algorithm. For any PPM algorithm running over a single link, the price estimate is the result of a Bernoulli trial with variance inversely proportional to N. As we move from a discrete to a continuous noise signal, however, it is also necessary to *scale* the variance, or, thinking in the frequency domain, the power of the noise. Consider the noise process resulting from the discrete sequence of price estimates taken at the receiver as each packet arrives. This process is a sequence of binomially distributed random variables with variance $\sigma_D^2 = F(q)\,(1-F(q))/N$. For moderately large N, we may approximate the binomial distribution by a normal distribution with variance σ_D^2 and treat the noise as discrete Gaussian white noise. We then imagine converting this discrete signal to a continuous one by passing it through a zero-order hold (ZOH) device with a sample time t_s equal to the packet transmission rate. If data is transmitted in evenly spaced packets with an average bit rate of x_0,

$$t_s = \frac{\beta}{x_0}$$

It can be shown that the variance for the resulting continuous signal is given by

$$t_s^2\,\sigma_D^2$$

Taking this scaling into account, we set the variance of the estimation noise to be

$$\sigma_\eta^2 = \frac{\beta^2}{x_0^2}\frac{F(q)\,(1-F(q))}{N}, \tag{9}$$

where x_0 is taken to be the source rate at equilibrium.

4 The Variance-Stability Tradeoff

We will initially ignore the slow time scale adaptation of the virtual capacity (7) and assume that $\tilde{c}$ is constant. Let x_τ and q_τ denote the delayed source rate and link price, $x(t-\tau)$ and $q(t-\tau)$, respectively. The interaction between source behavior, PPM price estimation and the induced effective delay can be approximated by the dynamics:

$$\frac{dx}{dt} = \kappa(w - x_\tau(q_\tau + \eta)); \tag{10}$$

$$q = \frac{x-\tilde{c}}{x} \tag{11}$$

$$\tau = \tau_0 + \frac{\beta N}{2x_0} \tag{12}$$

where x is source rate, q is the link price, β is the bits per packet, N is the number of samples, and η is the price estimation error with bias $\bar{\eta}(N,q)$ and variance

$$\sigma_\eta^2 \leq \frac{\kappa}{w+c}\frac{1}{4N}. \tag{13}$$

Note that we can remove the dependence of σ_η^2 on q by taking as an upper-bound the maximal variance of a Bernoulli trial, which occurs when $F(q) = 0.5$.

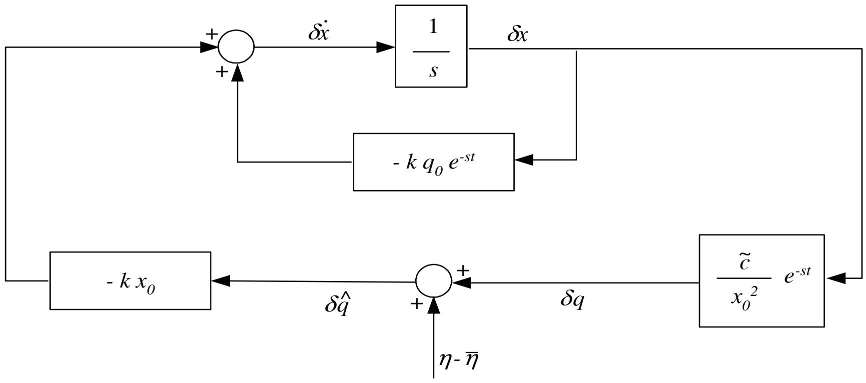

Figure 2. Linearized congestion control loop block diagram.

Denote by (x_0, q_0) the equilibrium point defined by $\dot{x} = 0$. Linearization of (10-12) about this equilibrium point gives

$$\begin{aligned} \dot{x} &= -\kappa\,(q_0 - \overline{\eta}(q_0))\,x_\tau - \kappa\left(x_0 + \frac{d\overline{\eta}(q_0)}{dq}\right) q_\tau; \\ q_\tau &= \frac{\tilde{c}}{(w+\tilde{c})^2} x_\tau \end{aligned}$$

where we have retained the (x,q) notation for the linearized variables. Observe that in the linearized system, the estimation noise has zero mean and variance σ_η^2. As we will see below, the effect of *bias* is captured in the value of the equilibrium point (x_0, q_0). Figure 2 shows a block diagram for the linearized system.

4.1 Results for an Unbiased Estimator

We first consider the case of an unbiased estimator, where $\overline{\eta}(q) = 0$. In this case, the equilibrium of (10-12) is $(x_0, q_0) = (w + \tilde{c}, \frac{w}{w+\tilde{c}})$. We linearize the dynamics given above around this equilibrium and compute the closed-loop transfer function from from η to x, which we denote $T_{x\eta}$. Ultimately, we will need to compute the 2-norm of $T_{x\eta}$. In order to obtain a closed form expression for the 2-norm, we use a first-order Pade approximation for the delay; i.e.,

$$e^{-s\tau} \approx \frac{-s\tau/2 + 1}{s\tau/2 + 1}.$$

Using this approximation we can compute $T_{x\eta}$ to be

$$\frac{x}{\eta} = \frac{\kappa(w+\tilde{c})(s + 2/\tau)}{s^2 + s(2/\tau - \kappa) + 2\kappa/\tau}.$$

This transfer function is stable for small τ with instability occurring when $2/\tau - \kappa < 0$. Because we have assumed that estimation error η takes the form of Gaussian white noise, the variance of rate x is given by a simple transformation of the noise

$$\sigma_x^2 = \|T_{x\eta}\|_2^2\, \sigma_\eta^2 \qquad (14)$$

where the 2-norm is given by

$$\|T_{x\eta}\|_2^2 = \frac{1}{2\pi} \int_{-\infty}^{\infty} |T_{x\eta}(j\omega)|^2 d\omega.$$

Computing the 2-norm using a state space method (see, for example, Section 2.6 of [4]), we obtain

$$\|T_{x\eta}\|_2^2 = \frac{(w+\tilde{c})^2\,\kappa\,(2+\kappa\tau)}{2(2-\kappa\tau)}.$$

Combining equation (14) with the upper bound for σ_η^2 in (13) gives an expression for the variance of the source rate

$$\sigma_x^2 = \frac{\kappa\,(w+\tilde{c})^2}{8N}\,\frac{2+\kappa(\tau_0 + \frac{\beta N}{2\,(w+\tilde{c})})}{2-\kappa(\tau_0 + \frac{\beta N}{2\,(w+\tilde{c})})}.$$

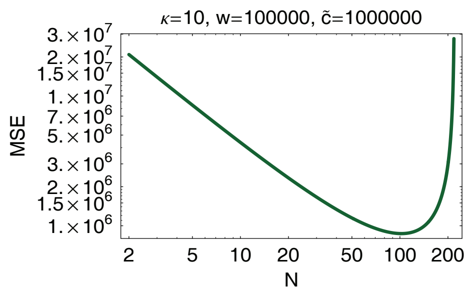

Figure 3. Source-rate MSE as function of the number of samples N used to estimate link price using an unbiased price estimate.

Under the assumption of an unbiased estimate, the source-rate MSE is equivalent to its variance (i.e. $M_x = \sigma_x^2$).

A plot of M_x versus N is given in Fig. 3 for an unbiased estimator (e.g. RAM).[3] MSE initially decreases as N increases due to the improvement in the price estimate but eventually begins to increase and ultimately becomes very large. This increase in MSE is due to the effective delay introduced by price estimation, which makes the loop unstable.

The curve plotted in Fig. 3 shows a broad region where MSE is quite close to its minimum, although it is obscured by the logarithmic scale of the plot.[4] This fact implies that we can achieve rather good protocol performance (low MSE) with just a few samples per estimate, well away from the boundary of instability.

Figure 4 shows continuous-time simulations[5] of the loop in Fig. 1 for the same parameters at three values of N. The onset of instability is clearly visible in the form of a periodic component in the trace that emerges and becomes more pronounced as N increases.

4.2 The Impact of Estimation Bias

We have seen that relatively few samples per estimate are needed for an unbiased estimator such as RAM to track the target equilibrium rate quite well. In the case of a biased estimator such as REM, however, using a small number of samples will contribute significant bias to the estimate. The linearized analysis presented above will no longer yield the relationship between estimation error and MSE because, as we now show, the presence of bias alters the equilibrium operating point of the network.

Consider again the case of a single session using a single link, now operating with a biased price estimator. As above,we take the value of N to be fixed and start by finding the equilibrium point (x_0, q_0). Note that due to our use of a symmetric error model (Gaussian white noise) the equilibrium is is unaffected by the variance of the estimate. We can therefore assume $\sigma_\eta^2 = 0$ for the purposes of computing (x_0, q_0). The dynamics in this case are given by

$$\begin{aligned} \dot{x} &= \kappa(w - x(q + \bar{\eta}(q))) \\ q &= \frac{(x-c)^+}{x}, \\ \bar{\eta}(q) &= \sum_{k=0}^{N} B(N,k,F(q))(q - G(k/N)), \end{aligned}$$

where $B(N,k,F(q))$ is the binomial PMF. Since price q is a function of rate x, we may express $\bar{\eta}$ as a function of x and write the equilibrium condition $\dot{x} = 0$ purely in terms of rate

$$x = \frac{w - \tilde{c}}{1 + \bar{\eta}(x)}. \qquad (15)$$

[3]Parameters are $\kappa = 10$, $w = 10^5$, $\tilde{c} = 1$Mbs, $\tau_0 = 100$ms and $\beta = 8000$.

[4]When plotted on a linear scale, the curve has a 'goal-post' shape.

[5]We used Simulink [16] to perform continuous-time simulations.

Equation (15) can be solved numerically via fixed point methods to find the equilibrium rate x_0. Using this solution, we then perform the linearized analysis about the shifted equilibrium.

Figure 5 illustrates the impact of bias on protocol performance assuming the bias of the REM estimator.[6] We observe that the plotted curves only differ at the low end of the horizontal scale, where the biased protocol has higher MSE. The difference between the biased and unbiased curves in the Figure is mainly due to the different equilibrium point in the biased case. This result confirms our intuition that—for a consistent but biased estimator such as REM—bias has an effect for very small N, but its impact quickly becomes negligible as more samples are used.

Figure 6 shows traces from the simulated model and illustrates the convergence to a different equilibrium point. For the parameters chosen, the biased price is an underestimate of the true price, which drives the equilibrium rate higher than the optimal rate. The lower trace in the figure is for the unbiased case and we see that the source rate fluctuates around the targeted optimum. The source rate under a biased estimator (shown in the upper trace) fluctuates around the shifted equilibrium. We will see below that the effect of bias on the target equilibrium is mitigated when we further incorporate the long time-scale dynamics of AVQ.

5 Long Time-Scale Dynamics

The full AVQ algorithm includes the dynamic (7) for adapting the virtual capacity $\tilde{c}$. This rule was introduced by Kunniyur and Srikant [13], who combined it with the primal rate adaptation rule (8) and the virtual queue pricing function (6). They showed that in the presence of this additional dynamic the equilibrium rates will converge to the exact solution of a global optimization problem proposed by Kelly [9] Without the AVQ rule 7, rates converge to the solution of a relaxed problem [10]. Because of this property, the AVQ algorithm is said to implement an *exact penalty function*.

In addition to implementing an exact penalty function, the AVQ rule is an example of *integral control*, in which the control action is proportional to the control error accumulated over time. Integral control is typically introduced in a system to eliminate steady state error. As we saw in the previous section, a biased price estimate offsets the equilibrium point of the control loop thus introducing precisely this type of error. Therefore, we expect that the AVQ algorithm can eliminate the effect of a biased price estimate—albeit, over a larger time scale than if we simply adopted an unbiased estimate.

[6]Parameters for Figures 5 and 6 are $\tilde{c} = 10^6$, $w = 10^5$, $\kappa = 10$, $\tau_0 = 0.1$, $\beta = 1000$. For the biased estimate, we add a bias equal to that of REM with parameter $\phi = 1.2$

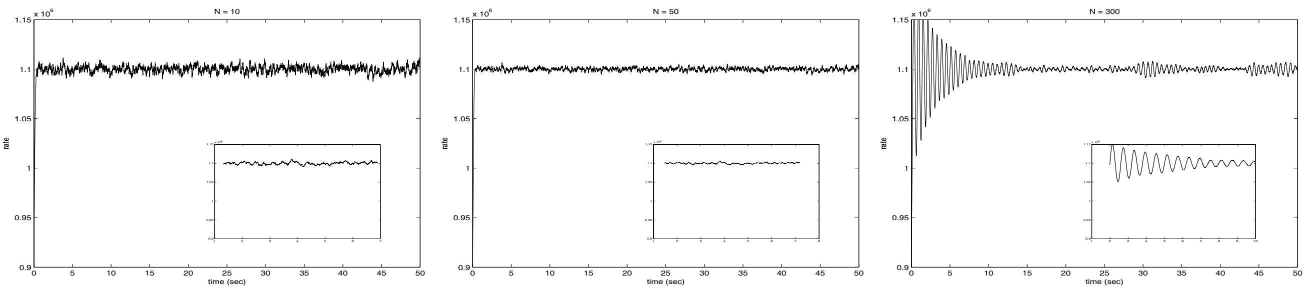

Figure 4. Simulation results for the parameters of Fig. 3. Each plot contains an inset showing an expanded view of a portion of the trace. As N increases from 10 to 50, the reduced noise in the price estimate lowers the variance of the rate somewhat. However, when $N = 300$ we observe the emergence of instability.

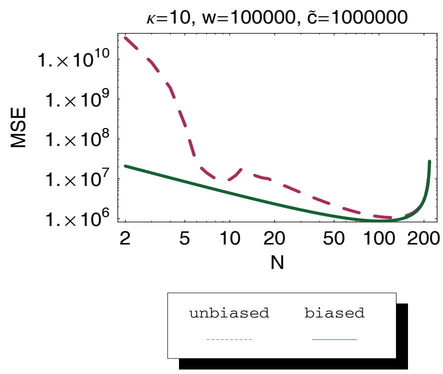

Figure 5. Source-rate MSE for as a function of N for both biased an unbiased estimators. We observe that bias contributes significantly to the error at low values of N.

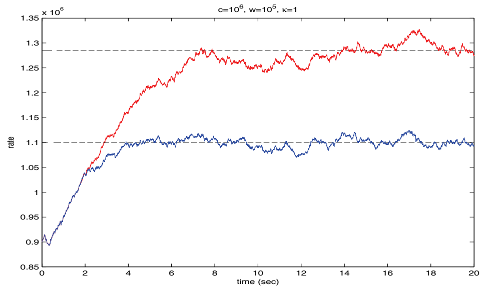

Figure 6. Simulation traces showing the effect of bias on the equilibrium rate when $N = 2$. The lower dashed horizontal line indicates the targeted optimal rate. The upper dashed line indicate the predicted equilibrium in the presence of estimation bias.

Simulation of AVQ confirms our intuition, as Figure 7 illustrates.[7] In this simulation, AVQ adapts the rate of both the unbiased and biased sources to match the rate to the desired link capacity c.[8] However, we observe that for the biased source, AVQ must set the virtual capacity $\tilde{c}$ to a lower level than for the unbiased source to achieve the same target operating point.

6 Packet-Level Simulations

The analytical and fluid simulation results presented in the preceding section suggest that the tradeoff between precision and stability favors low precision estimates of congestion price. However, this result was based on a simplified model of the actual estimation error as Gaussian white noise. The true variance of the price estimate is given by a binomial distribution and the normal approximation is only valid for reasonably high values of N. It is reasonable to suspect, therefore, that the results of the fluid simulation at small estimation window sizes will deviate from the behavior of a true system. Furthermore, the fluid model represents an upper bound on the error introduced by price estimation due our assuming maximal variance in (13). The modelling error thus introduced should tend to overestimate the variance of the actual system, particularly at small estimation window sizes. It is therefore important that we validate our findings with a packet-level simulation.

We implemented a rate-based congestion control protocol in the NS-2 simulator [5]. The links in these simulations implement the AVQ pricing rules (6-7) for setting price. We

[7] Parameters for this simulation are $\tilde{c} = 10^6$, $w = 10^5$, $\kappa = 2$, $N = 2$, $\phi = 1.2$, $\tau_0 = 0.1$, $\beta = 1000$.

[8] Note that without the AVQ rule, the equilibrium rate for an unbiased source would be $(w + c)$, assuming $\tilde{c} = c$.

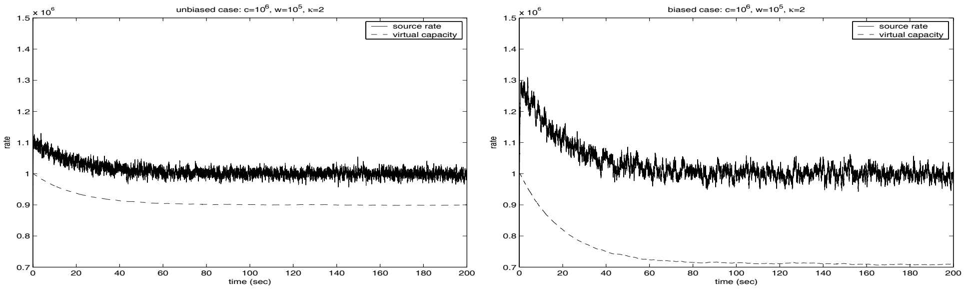

Figure 7. Simulation traces of full AVQ protocol showing source rate (solid line) and adaptation of virtual capacity $\tilde{c}$ (dashed line). Observe that the rate in both the biased and unbiased cases is stabilized around the targeted optimal rate.

Estimation Window N	Source Rate Variance ($\times 10^{10}$)
1	0.4731
10	0.5731
50	0.9097
100	1.8777

Table 1. Variance of the simulation traces shown in Fig. 8 (including a trace for $N = 50$ not shown in the figure. For this case we see that estimations based on a single bit are optimal.

also implemented a complete version of the RAM protocol, which we used to communicate the link prices to the receiver using one of the ECN bits of the IP header. The sender updates its rate according to (8) and adaptively sets the rate of a token bucket filter to enforce the new rate.

Figure 8 shows three representative traces from our packet-level simulation. Simulation parameters are identical in all three traces with the exception of the estimation window size N.[9] For each trace, the figure shows the sender's rate as a function of time along side a plot of the actual and estimated link price. We can easily observe the improvement of the price estimate with N. In the case of $N = 1$ the possible values of the estimated price are zero and one. As N increases, the price estimation scale becomes finer until we see in the third plot ($N = 100$) that the estimated price tracks the actual price very closely, although a time lag between the two is plainly visible. The effect of this time lag is evident in the oscillatory component of the source rate—a signature of control instability. Despite the extremely crude estimate of the price provided when $N = 1$ the variance in sender rate appears similar to that when $N = 10$ where the estimate is more precise.

We computed the variance of the source rate time series for each trace with the initial transient removed. The results, shown in Table 1, suggest that estimates based on a single packet are surprisingly effective, yielding the lowest variance in this example. This result differs from the continuous-time simulation of our system model, where the minimum variance occured at a higher values of N. This difference can be attributed to the reasons mentioned at the begining of this section, namely, that the Gaussian noise model we adopted overestimated the error at small values of N.

We also ran packet-level simulations for paths with multiple links and a single point of congestion. These simulations use a tandem topology, an example of which is shown in Fig. 9. We ran simulations for topologies with one, two, and three links, running each simulation ten times with different random seeds. We then looked at the source-rate variance for the session that uses all of the links, which we refer to as the *long session*. To isolate the effect of multiple links, we chose to control all other simulation parameters. Therefore, in all simulations the total end-to-end delay is fixed at 200ms, the gain parameter κ is fixed at 4.0, and the average total price is fixed due to the fact that only one link is a bottleneck.[10] We compare the results in Figure 10. Observe that the variance increases as a function of the number of links. This increase is expected for RAM because the RAM marking probability encodes average price per link. To get the total price, we must multiplying by the number of links n, increasing the variance of the estimated price by n^2. We see in all three cases that increasing N beyond a modest value (around 20) is of limited utility. It is worth pointing out that the gain for this simulation is set significantly higher than for the results presented in Fig. 8. It is for this reason that we see the predicted increased variance at $N = 1$.

[9]Parameters are $\kappa = 4$, $w = 10^5$, $\tilde{c} = 1$Mbs, $\tau_0 = 100$ms and $\beta = 1000$.

[10]To compensate for the high gain parameter, the packet size is reduced to 100 bytes ($\beta = 800$) in these simulations, which increases the rate of feedback.

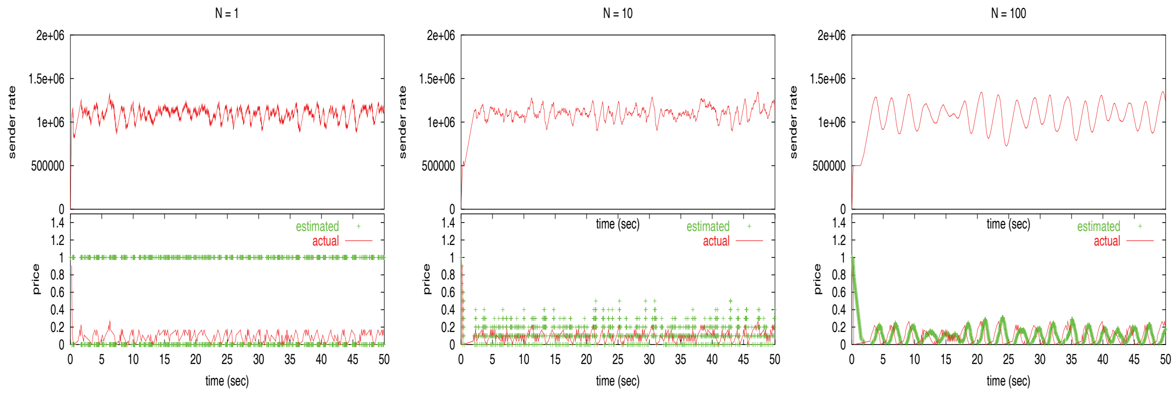

Figure 8. Emergence of instability in packet-level simulations

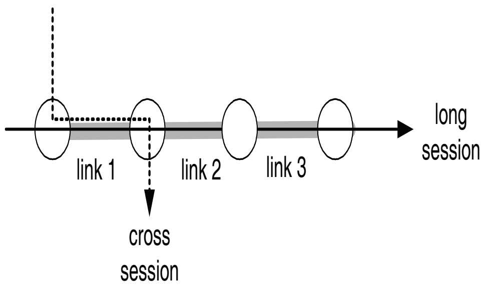

Figure 9. Example of a tandem topology with three links

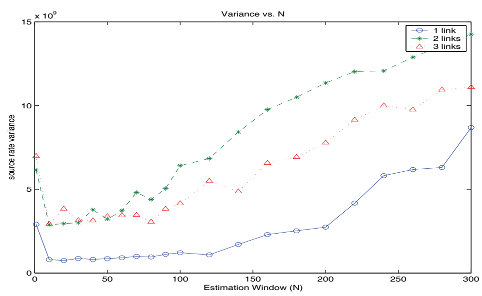

Figure 10. Plot of the ratio of variance to mean (averaged over 10 simulation runs) for railroad topologies with one, two, and three links.

7 Related Work

Ganesh, Key and Massoulié [6] introduce the source-rate MSE metric that we also adopt in our work and address the question of *how many* bits of feedback per packet are necessary in a pricing-based congestion control protocol. They found that in the presence of feedback delays shorter than the timescale of price fluctuations, only a small number of bits are helpful. These authors did not explicitly consider the additional delay associated with collecting multiple bits spread over multiple packets, nor did they investigate the impact of price estimation bias.

Paganini, et al [17] also identify the additional delay due to price estimation based on a moving average. They observe that the frequency response of a moving average filter has an effect similar to that of a pure delay equal to the time required to collect $N/2$ packets. On the basis of these observations, the authors make several recommendations, including keeping N below 100. However, the effect of bias in the estimate is not considered. In our work, we attempt to develop a more quantitative understanding of the effect of price estimation both with and without bias.

The impact of a moving average on protocol stability was observed by Hollot, et al. in the RED active queue management (AQM) algorithm's use of average queue length to compute a congestion signal [8]. They proposed an alternate AQM controller based on instantaneous queue length to improve the stability of TCP congestion control.

Thommes and Coates propose a deterministic packet marking strategy that takes advantage of two features of the IP header that are ignored by RAM and REM [19]. First, the IP header actually contains two ECN bits. Three of the four possible codepoints defined by these two bits can actually be used to carry price information.[11] Second, the semantics

[11] The codepoint defined by having both bits set to zero is used to indicate a lack of ECN processing capability.

of the IPid field in the IP header can be exploited to give additional power to a protocol. The fact that the IPid value changes for each packet but remains constant as a packet traverses a path is used to accumulate the exact path price by collecting a set of partial sums, each encoded as a ternary value. The expected number of packets required to collect the total path price is on the order of 60 to 100, depending on how IPid values are generated. A comparison of this deterministic strategy with probabilistic techniques is beyond the scope of this paper, but is an important topic for further study.

8 Conclusion

In this paper we have investigated a simple model of a congestion control loop with feedback delay and noisy (perhaps biased) price estimation. We first observed that the need to maintain stability of the loop places a limit on the number of samples we can collect for each price estimate, even when using a moving average to compute an updated estimate for every packet arrival. Through simulation, we found that price estimates can be made using surprisingly few samples—on the order of ten—while still maintaining close to minimal error in source rates.

If price estimates are based on small numbers of samples, however, it is necessary to consider the effects of bias, even for biased-but-consistent estimators like REM. We find that bias has an effect of shifting the equilibrium rate allocation away from what it would be in the unbiased case. The steady-state error thus introduced can be eliminated by introducing integral control into the loop, as the AVQ price-setting algorithm effectively does.

This work provides an important step toward understanding the behavior of congestion control protocols that use PPM to encode a congestion signal in a single bit per packet. In particular, we have identified and evaluated a fundamental tradeoff between stability and error in the equilibrium rates. To complete this picture, continuing work must address the three-way tradeoff relating stability, error, and responsiveness of the protocol.

Acknowledgments

We thank Yu Gu of Umass Amherst for significant help with the packet-level simulations presented in this paper.

References

[1] M. Adler, J.-Y. Cai, J. K. Shapiro, and D. Towsley. Estimation of congestion price using probabilistic packet marking. In *Proceedings of IEEE Infocomm*, San Francisco, CA, April 2003.

[2] S. Athuraliya, D. Laspsley, and S. Low. An enhanced random early marking algorithm for internet flow control. In *Proc. INFOCOM*, 2000.

[3] S. Athurliya. A note on parameter values of rem with reno-like algorithms. March 2002.

[4] J. C. Doyle, B. A. Francis, and A. R. Tannenbaum. *Feedback Control Theory*. Macmillan Publishing Company, 1992.

[5] K. Fall and K. Varadhan. ns notes and documentation. Technical report, The VINT Project at LBL, Xerox PARC, UCB, and USC/ISI, August 2001. http://www.isi.edu/nsnam/ns/.

[6] A. Ganesh, P. Key, and L. Massoulié. Feedback and bandwidth sharing in networks. In *Proceedings of 39th Annual Allerton Conference on Communication, Control, and Computing*, 2001.

[7] R. Gibbens and F. Kelly. Resource pricing and the evolution of congestion control. *Automatica*, 35:1969–1985, 1999.

[8] C. Hollot, V. Misra, D. Towlsey, and W. Gong. On designing improved controllers for aqm routers supporting tcp flows. In *Proc. IEEE Infocom 2001*, 2001.

[9] F. Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, volume 8:33–37, 1997.

[10] F. Kelly, A. Maulloo, and D. Tan. Rate control for communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49, 1998.

[11] P. Key, L. Massoulié, and J. K. Shapiro. Service differentiation for delay-sensitive applications: an optimisation-based approach. *Performance Evaluation*, 49:471–489, 2002.

[12] S. Kunniyur and R. Srikant. End-to-end congestion control: utility functions, random losses and ECN marks. In *Proc. INFOCOM*, 2000.

[13] S. Kunniyur and R. Srikant. A time-scale decomposition approach to adaptive explicit congestion notification (ECN) marking. *IEEE Trans. on Automatic Control*, 47(6):882–894, June 2002.

[14] S. H. Low and D. E. Lapsley. Optimization flow control, i: Basic algorithm and convergence. *IEEE/ACM Transactions on Networking*, December 1999.

[15] S. H. Low and R. Srikant. A mathematical framework for designing a low-loss, low-delay internet. *Networks and Spatial Economics*, Jan-Feb 2003. special issue on Crossovers between Transportation Planning and Telecommunications.

[16] Mathworks. Simulink. http://www.mathworks.com/products/simulink/.

[17] F. Paganini, S. H. Low, Z. Wang, S. Athuraliya, and J. C. Doyle. A new tcp congestion control with empty queues and scalable stability. submitted for publication, 2002.

[18] K. Ramakrishnan and S. Floyd. A proposal to add explicit congestion notification (ECN) to IP. Technical report, INTERNET DRAFT draft-kksjf-ecn-02.txt, 2000.

[19] R. Thommes and M. J. Coates. Deterministic packet marking for congestion price estimation. In *Proc. IEEE Infocom*, Hong Kong, March 2004.