Estimation of Congestion Price Using Probabilistic Packet Marking

Micah Adler*, Jin-Yi Cai†, Jonathan K. Shapiro* and Don Towsley*

* Department of Computer Science
University of Massachusetts at Amherst
{micah, jshapiro, towsley}@ cs.umass.edu

† Computer Sciences Department
University of Wisconsin, Madison
jyc@cs.wisc.edu

Abstract— **One key component of recent pricing-based congestion control schemes is an algorithm for probabilistically setting the Explicit Congestion Notification bit at routers so that a receiver can estimate the sum of link congestion prices along a path. We consider two such algorithms—a well-known algorithm called Random Exponential Marking (REM) and a novel algorithm called Random Additive Marking (RAM). We show that if link prices are unbounded, a class of REM-like algorithms are the only ones possible. Unfortunately, REM computes a biased estimate of total price and requires setting a parameter for which no uniformly good choice exists in a network setting. However, we show that if prices can be bounded and therefore normalized, then there is an alternate class of feasible algorithms, of which RAM is representative and furthermore, only the REM-like and RAM-like classes are possible. For properly normalized link prices, RAM returns an optimal price estimate (in terms of mean squared error), outperforming REM even if the REM parameter is chosen optimally. RAM does not require setting a parameter like REM, but does require a router to know its position along the path taken by a packet. We present an implementation of RAM for the Internet that exploits the existing semantics of the time-to-live field in IP to provide the necessary path position information.**

I. Introduction

Recent theoretical advances in optimization-based congestion control have led to the development of protocols in which congestion signals—or *prices* in the common terminology—are computed by links in the network and communicated to sessions. The prices represent Lagrange multipliers in a global optimization problem of maximizing the aggregate user utility in the network subject to a capacity constraint on each link. By knowing only the total price along its own path, each session can independently adapt its rate in a greedy fashion, optimizing its individual utility minus cost. When prices are set correctly by the network, the joint actions of all the users track the globally optimal rate allocation.

In considering the issues surrounding the deployment of such protocols in IP networks, the explicit congestion notification (ECN) bit in the IP header [1] has emerged as a key tool for practical implementations. The importance of ECN is three-fold. First, ECN decouples congestion signals from packet loss—a necessary condition for operating networks with low loss and low delay. Second, an ECN bit already exists in the standard IP header. As we will see, a single bit is sufficient to communicate prices. Thus the debate can focus on how to use the existing bit rather than on how many bits (if any) should be reserved.[1] Third (and most relevant to this paper), it has been demonstrated that routers can encode prices by probabilistically setting the ECN bit in such a way that the end-to-end marking probability encodes the sum of prices along a path. Thus receivers can estimate the total price along a session path, by recording the fraction of marked packets.

Optimization-based congestion control protocols consist of 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. Two classes of protocols have been proposed to date. The first, originally described by Gibbens and Kelly [2], employs an open-loop marking policy at links and adjusts rates iteratively at the end hosts. In the second class [3], [4] end-hosts set rates deterministically, and links combine an iterative algorithm for setting prices with probabilistic packet marking for encoding prices. We concern ourselves with this latter class of protocols where the link price computation and marking scheme are easily separable.

In this work, we assume link prices have converged to steady-state values and focus on the the problem of communicating the sum of fixed link prices along a path by means of packet marking, which we now formalize. 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 s_i. Let $z_n = \sum_{i=1}^{n} s_i$ denote the sum of prices along the

This work was supported in part by the National Science Foundation (NSF) under grant numbers ANI-9980552, EIA-0080119, CCR-0133664 and CCR-9634665. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

[1] In actuality, two bits in the IP header have been reserved for the purposes of ECN. However, only one of these bits is used to carry congestion signals along the forward path.

path. As data packets traversing the path arrive at a receiver, the receiver must determine z_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, as is the case in the current IP standard. 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 z_n. To be practically implementable, a marking algorithm must obey the following design constraints: First, the algorithm must be fully distributed with each link making use of locally available information, namely, the price s_i and, if $i > 1$ the bit X_{i-1} computed at the previous step. In some cases, the step index i may also be considered available information. Second, the algorithm should not be required to maintain per-flow state, since this might impose prohibitive storage overheads on routers serving many simultaneous flows. This constraint is clearly satisfied if a link may not retain any memory of how previous packets were marked. Although there is no deterministic marking algorithm under these conditions, it is possible to *probabilistically* set the bit so that the end-to-end marking probability encodes z_n.

In this work, we consider two probabilistic packet marking algorithms—one by Athuraliya and Low [4] called *REM*, and a novel algorithm we have developed called *RAM*[2]—and characterize a large class of such algorithms with a generalized model. We show that REM is essentially the only method in this class possible when there are no further restrictions on s_i, except $s_i \geq 0$. However, this estimator is biased and, more serious, requires setting a parameter for which no uniformly good choice exists. When the additional information of the step index i is known at the i^{th} step and when we assume that each s_i is bounded by some fixed upper bound, say $0 \leq s_i \leq 1$, our RAM method becomes feasible. Moreover, when link prices are restricted to a finite interval, variations of RAM and REM are the only possible methods in the modelled class. We compare REM and RAM in terms of two common metrics. RAM is shown to be optimal in terms of mean squared error (M. S. E.) for the uniform a priori distribution of the average price z_n/n. Finally, we present an Internet implementation of RAM, exploiting the existing semantics of the IP time-to-live field to provide the step index i (or an estimate thereof) to each link along a path.

The rest of this paper is organized as follows: In Section II, we present the REM and RAM algorithms along with a generalized model of possible marking algorithms. In Section III we identify key properties of all feasible protocols and establish the uniqueness of REM for unbounded prices and of REM and RAM when prices are bounded. Sections IV and IV-B compare REM and RAM in terms of the tail probability of their price estimates and consider the problem of setting a key parameter in REM. In Section IV-C we compare REM and RAM in terms of mean squared error and establish the optimality of RAM under this criterion. We present an implementation for RAM for the Internet in Section V. Due to space limitations, we have omitted proofs for several theorems presented in this paper. The interested reader can find all proofs in our technical report [5].

II. Probabilistic Packet Marking

A. Random Exponential Marking

The Random Exponential Marking (REM) scheme proposed by Athuraliya and Low [4] is, as far as we are aware, the only existing marking algorithm for price estimation. In REM, the designer selects some base $\varphi > 1$. The initial price bit X_0 is set to 0. The i^{th} link, where $i \geq 1$, sets the price bit to 1 with probability $1 - \varphi^{-s_i}$. It is convenient to think in terms of conditioning on the incoming price bit X_{i-1}. If $X_{i-1} = 1$ then $X_i = 1$ as well, whereas if $X_{i-1} = 0$ then $X_i = 0$ with probability φ^{-s_i} and $X_i = 1$ with probability $1 - \varphi^{-s_i}$.

The bit arriving at the receiver is X_n. It is clear that $X_n = 0$ with probability $\prod_{i=1}^{n} \varphi^{-s_i} = \varphi^{-\sum_{i=1}^{n} s_i}$, and $X_n = 1$ otherwise. Hence the expectation $\mathbf{E}[X_n] = 1 - \varphi^{-z_n}$. To estimate the total price z_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 takes $\overline{X} = (\sum_{j=1}^{N} X_n^{(j)})/N$, and estimates z_n to be $-\log_\varphi(1 - \overline{X})$.

Note that since $\log_\varphi(x)$ is a non-linear function, the expectation $\mathbf{E}[-\log_\varphi(1 - \overline{X})]$ is not equal to $-\log_\varphi(1 - \mathbf{E}[\overline{X}]) = z_n$. By Jensen's inequality, since log is a strictly convex function, we have

$$\mathbf{E}[-\log(1 - \overline{X})] > z_n.$$

However, even though REM is a biased estimator, as $N \longrightarrow \infty$ we do have almost everywhere convergence $-\log(1 - \overline{X}) \rightarrow z_n$, a.s.. Note also that in REM, the local computation at each step depends only on the local price s_i and the previous bit X_{i-1}, but does not depend on the step index i. Finally, observe that the base φ is a parameter that must be chosen by the designer. Athuraliya and Low give no prescription for setting φ, but do observe that it should be chosen so as to keep the end-to-end marking probability away from the extreme values of 0 and 1.

B. Random Additive Marking

Suppose we restrict the range of each link price s_i to be $0 \leq s_i \leq 1$, and suppose the step index i is known for local computation at the i^{th} step. Under these conditions, an alternative scheme is feasible. Again, we set $X_0 = 0$. 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 s_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 $\mathbf{E}[X_n] = \sum_{i=1}^{n} s_i/n$. We thus have an unbiased estimator for z_n/n; we simply collect N i.i.d. samples and compute the average $\overline{X}$. Since the step index is known at each step, the receiver can determine n and thus obtain z_n. We call this scheme *Random Additive Marking* (RAM).

[2] We will define the acronyms REM and RAM below.

C. Generalized Protocol Model

The most general one-bit on-line assignment protocol can be described as follows. Consider the i^{th} step, where $i \geq 1$. The incoming price bit X_{i-1} is either 0 or 1. We then assign the outgoing price bit X_i according to a 0-1 random variable whose distribution is conditional on the value of the incoming bit. Any possible assignment of the price bit at step i can be defined in terms of the conditional probabilities of setting the output bit to 1, which depend on i and s_i. Thus, we have

$$p_i = p_{i-1} f(i, s_i) + (1 - p_{i-1}) g(i, s_i), \tag{1}$$

where

$$\begin{aligned} p_j(s) &= \Pr[X_j = 1], \\ f(i, s_i) &= \Pr[X_i = 1 | X_{i-1} = 0], \\ g(i, s_i) &= \Pr[X_i = 1 | X_{i-1} = 1]. \end{aligned}$$

In this work, we restrict our focus to protocols where f and g are continuous in the local price s_i. Note that this class of protocols is quite large, and includes protocols defined by any *computable real functions* f and g since, in the strict sense of computability, all computable real functions are continuous (See [6] Theorem 4.3.1, page 108). We discuss some implications of this restriction further in Section III-D.

III. Characterization of Protocols

In this section we provide a characterization of feasible protocols, such that, for all $(s_1, s_2, \ldots, s_n)$ the estimator converges to z_n, when sample size $N \to \infty$. In Subsection III-A we prove that the probability $p_n = \Pr[X_n = 1]$, as a function of $(s_1, s_2, \ldots, s_n)$ must be a function of $\sum_{i=1}^n s_i$, and must be continuous and strictly monotonic in this single argument. In Subsections III-B and III-C we give a complete analytic characterizations of all feasible protocols for the cases of unbounded and bounded link prices.

A. Strict monotonicity as a function of $\sum_{i=1}^n s_i$

No matter what it does at each step i, a marking algorithm ultimately produces a 0-1 random variable X_n. Thus looking at the problem externally any algorithm can be characterized by the probability that $X_n = 0$. This probability must be a function of $s_1, s_2, \ldots, s_n$; we will call it $p_n(s_1, s_2, \ldots, s_n)$.

Theorem 1: If for all $(s_1, s_2, \ldots, s_n)$ the estimator converges to $z_n = \sum_{i=1}^n s_i$ asymptotically, as the number of sample points $N \to \infty$, p_n must be a function of the sum z_n, and be continuous and strictly monotonic in its single argument z_n.

Some intuitive ideas behind the proof of Theorem 1 are as follows. The key claim is that p_n is a function of the sum alone. If $p_n(s_1, s_2, \ldots, s_n) = p_n(s'_1, s'_2, \ldots, s'_n)$, then they produce identical distributions, for all sample size N. In order to be able to converge to the sum, using the strong Law of Large Numbers, we can show that $p_n(s_1, s_2, \ldots, s_n) = p_n(s'_1, s'_2, \ldots, s'_n)$ implies $\sum_{i=1}^n s_i = \sum_{i=1}^n s'_i$. Then using a geometric argument, and the Intermediate Value Theorem, we can show that the converse also holds, namely $\sum_{i=1}^n s_i = \sum_{i=1}^n s'_i$ implies $p_n(s_1, s_2, \ldots, s_n) = p_n(s'_1, s'_2, \ldots, s'_n)$. The complete proof can be found in our technical report [5].

B. Solutions of functional equations over $[0, \infty)$

Now we fix $i \geq 1$. To simplify expressions, define $s = \sum_{j=1}^i s_j$ and $t = s_{i+1}$. Consider the functional equation transferring the probability from step i to $i+1$:

$$h(s+t) = p(s) f(t) + (1 - p(s)) g(t). \tag{2}$$

Note that implicitly, all of these functions can depend on i, which is fixed.

Theorem 2: Suppose h, p, f and g are continuous real valued functions defined on $[0, \infty)$, and satisfy the functional equation (2) for all $s, t \geq 0$. Assume furthermore that p is strictly monotonic and bounded, and h is non-constant. Then there exists a constant $0 < \psi < 1$, such that each function h, p, f and g is of the form $c + c'\psi^x$ for some constants c and c'. More precisely, there exist constants $0 < \psi < 1$, and a, b, c and d, such that

$$\begin{aligned} p(x) &= a + b\psi^x \\ f(x) &= c + (1-a) d\psi^x \\ g(x) &= c - ad\psi^x \\ h(x) &= c + bd\psi^x \end{aligned}$$

The proof of this theorem can be found in [5].

In the following we will write $\varphi = \psi^{-1}$, thus $\varphi > 1$.

In order to be a probability and strictly monotonic, the constants a and b in the function p must also satisfy

$$0 \leq a, a+b \leq 1, \qquad \text{and} \qquad b \neq 0. \tag{3}$$

We note that given this complete characterization, it is easy to see that REM corresponds to the choice of constants $a = 1$ and $b = -1$ for $p(s)$. There is a dual choice of $a = 0$ and $b = 1$, which we will call REM*.

For all parameters (technically for all *computable* parameters) a and b satisfying (3), the function $p(\cdot)$ is realizable as the probability function of some one-bit on-line protocol as defined. In fact, if $n = 1$, we can just take X_1 such that $\Pr[X_1 = 1] = a + b\varphi^{-s_1}$. This is legitimate since $a + b\varphi^{-s_1}$ is always between a and $a+b$, and thus $0 \leq a + b\varphi^{-s_1} \leq 1$, for all $s_1 \geq 0$. For $n > 1$, inductively we can assume $\Pr[X_{n-1} = 1] = a + b\varphi^{-(z_{n-1})}$ as p_{n-1}, then we let $f(s_n) = a + (1-a)\varphi^{-s_n}$ and $g(s_n) = a - a\varphi^{-s_n}$. Again it is easy to see that both $0 \leq f(s_n), g(s_n) \leq 1$, for all $s_n \geq 0$. It follows that $p_n(z_n) = h(z_n) = a + b\varphi^{-z_n}$. We will call all these feasible protocols REM-like.

For any fixed φ, the question of what choices of a and b are the best remains unanswered. It can be shown [5] that, in terms of M.S.E., REM and REM* are the best choices of all these REM-like protocols with the same φ.

C. Solutions of functional equations over $[0,1]$

In the previous subsection, we gave a complete characterization of probability functions of all admissible one-bit online protocols as defined before, provided that $s_i \in [0,\infty)$ $i = 1,\ldots,n$. When we have the further restriction that $s_i \in [0,1]$ $i = 1,\ldots,n$, there are other solutions to the functional equations, which we turn to in this subsection.

Fix $i \geq 1$ and consider again the functional equation (2), except now f and g are only defined for $x \in [0,1]$, and p is defined for $x \in [0,i]$ and h is defined for $x \in [0,i+1]$. Note that implicitly, all these functions can depend on i, which is fixed.

Theorem 3: Suppose h,p,f and g are continuous, real valued functions defined on $[0,i+1]$, $[0,i]$, $[0,1]$ and $[0,1]$, respectively, and satisfy the functional equation (2), for all $s \in [0,i]$ and $t \in [0,1]$. Assume furthermore that p is strictly monotonic and bounded, and h is non-constant. Then there are just two classes of solutions:

1) There exists a constant $\psi > 0$, $\psi \neq 1$, such that each function h,p,f and g is of the form $a + b\psi^x$ for some constants a and b. Or
2) Each function h,p,f and g is an affine linear function of x of the form $a + bx$ for some constants a and b.

Here note that all these constants may depend on i.

The proof of this theorem can be found in [5].

The first class of solutions is essentially the exponential family discussed above.[3] As we showed before, if we want the functional equation to hold over functions defined over $[0,\infty)$, then there is only this first class of solutions with $0 < \psi < 1$; the second class of solutions is not possible. What makes it possible here is the restriction of the functional equation to a finite interval.

For a path of length n, denote by $\theta = \sum_{i=1}^{n} s_i/n$. Any admissible protocol of the second class must have $\Pr[X_i = 0] = a + b\theta$ for some constants a and b (which may depend on i.) Since $a + b\theta$ is a probability, $0 \leq a, a+b \leq 1$. RAM simply takes $a = 0$ and $b = 1$ and is thus an unbiased estimator of θ. There is a dual choice that corresponds to $a = 1$ and $b = -1$. In Section IV-C, we show that RAM and its dual are uniquely optimal with respect to the criterion of Mean Square Error, among all solutions of the second class.

It is easy to verify that all feasible choices of (a,b) can be realized in a one-bit on-line protocol when each $s_j \in [0,1]$, and if at step i we know the index i. Assume we have the probability function $(\sum_{j=1}^{i-1} s_j)/(i-1)$ (as in RAM) for the $i-1$ step. Then let $f(s_i) = a + \frac{i-1}{i}b + \frac{b}{i}s_i$, and $g(s_i) = a + \frac{b}{i}s_i$. These choices are both legitimate since both $a + \frac{i-1}{i}b = \frac{1}{i}a + \frac{i-1}{i}(a+b)$, and $a + \frac{1}{i}b = \frac{i-1}{i}a + \frac{1}{i}(a+b)$, are convex combinations of a and $a+b$, and therefore, since both $a, a+b \in [0,1]$, it follows that all four numbers $f(0) = a + \frac{i-1}{i}b, f(1) = a+b, g(0) = a, g(1) = a + \frac{1}{i}b \in [0,1]$.

D. Discussion

Although we have shown that REM and RAM are the only feasible protocols among a large class of candidates, a fully general claim eludes us at this time due to the continuity assumption mentioned in Section II-C. Indeed, we know of at least one example of a theoretically valid, albeit impractical protocol defined by non-continuous functions f and g. Each node i of the path of n nodes takes the binary representation of s_i (the local price) and inserts n 0s between every pair of bits in this binary representation. The resulting sequence of bits is then shifted i bits to the right. Let this new value be called s_i'. The marking part of the protocol proceeds just like RAM, except that the s_i' are used instead of the s_i. The receiver can then estimate the sum z_n (and, in fact, the individual s_i values) by estimating the marking probability to σn bits of precision, where σ is the number of bits used to represent s_i.[4]

IV. Evaluation

A. Comparison of Tail Probabilities

We next consider the receiver's problem of estimating the price of a path using either RAM or REM for marking packets. Suppose the receiver collects N packets, giving it N samples of the price bit. Let $B = \sum_{j=1}^{N} X_n^{(j)}$ be the number of samples for which the price bit is set. The receiver can then estimate the path price by estimating the end-to-end marking probability. Let p denote the true end-to-end marking probability. The estimated marking probability is $\hat{p} = B/N$. For now, we assume the path length n is known to the receiver.

To simplify expressions, we will drop the superscript for the path price notation, thus

$$z = \sum_{i=1}^{n} s_i.$$

Let $\hat{z}$ be the price estimate provided by either algorithm. For RAM, we have

$$\hat{z} = \hat{p}n, \tag{4}$$

whereas for REM,

$$\hat{z} = -\log(1-\hat{p}). \tag{5}$$

The true path price z can also be expressed using equations (4) and (5) by substituting the true marking probability p for the estimated probability $\hat{p}$ on the right-hand side. Informally, we can think of the efficiency of a marking algorithm as the number of samples required to estimate the true price with high confidence. This notion is captured in the metric of *error probability*, denoted $err(\varepsilon)$ and defined as the probability that the price estimate falls outside of some range about the true price, where the range is determined by a parameter ε. Formally,

$$err(\varepsilon) = 1 - \Pr[(1-\varepsilon)z \leq \hat{z} \leq (1+\varepsilon)z] \tag{6}$$

[3] For finite interval $[0,i]$, $\psi > 1$ is possible; but it is easily transformed to the case with $\psi < 1$, by reversing the map $x \mapsto i - x$. For the infinite interval $[0,\infty)$, $\psi > 1$ is impossible, and we get $0 < \psi < 1$.

[4] Note that f and g are clearly computable, and the natural generalization of these functions to real functions is non-continuous. However, this fact does not contradict the continuity of computable real functions, since f and g as defined above are not real functions, but functions from strings to strings (due to the non-uniqueness of the binary representation of s_i).

It is natural to compare REM and RAM on the basis of efficiency, and the error probability provides one tractable metric for doing so.

Since both algorithms use the estimated marking probability $\hat{p}$ to estimate the price, it is also useful to relate the acceptable variation in $\hat{z}$ (as defined by the parameter ε) to an equivalent variation in $\hat{p}$,

$$err(\varepsilon) = 1 - \Pr[(1-\delta^-)\,p \le \hat{p} \le (1+\delta^+)\,p], \qquad (7)$$

where δ^- and δ^+ depend on the value of ε, the marking probability p, and the choice of marking algorithm. As will become clear below, we must distinguish between the values of δ for the upper and lower tail since these may not be equal.

Noting that the price estimates (4) and (5) are increasing in $\hat{p}$, we may conclude that

$$\hat{z} = (1-\varepsilon)\,z \quad \Leftrightarrow \quad \hat{p} = (1-\delta^-)\,p \qquad (8)$$

$$\hat{z} = (1+\varepsilon)\,z \quad \Leftrightarrow \quad \hat{p} = (1+\delta^+)\,p \qquad (9)$$

Taking RAM as an example, let us fix a value of ε and require that

$$\Pr[\hat{z} \le (1-\varepsilon)\,z] \;=\; \Pr[\hat{p} \le (1-\delta^-)\,p] \qquad (10)$$

$$\Pr[\hat{z} \ge (1+\varepsilon)\,z] \;=\; \Pr[\hat{p} \ge (1+\delta^+)\,p]. \qquad (11)$$

We can now solve for the values of δ^- and δ^+ that make this requirement true. Using equation (4) and observation (8), we have

$$\hat{z} = (1-\varepsilon)\,z = (1-\delta^-)\,pN. \qquad (12)$$

Using the fact that $z = pN$ we can rewrite (12)

$$(1-\delta^-)\,pN = (1-\varepsilon)\,pN$$

Thus, for a fixed $\varepsilon \in (0,1)$ we may set $\delta^- = \varepsilon$ for the case of RAM. Essentially identical reasoning establishes that $\delta^+ = \varepsilon$.

In the case of REM, a more complex relationship holds. Using equation (5) and observation (8), we can write

$$\hat{z} = (1-\varepsilon)\,z = -\log_\varphi(1-(1-\delta^-)(1-\varphi^{-z})),$$

Solving for δ^-, we have

$$\delta^- = \frac{\varphi^{-(1-\varepsilon)z} - \varphi^{-z}}{1-\varphi^{-z}} \qquad (13)$$

Applying the same reasoning for the upper tail gives

$$\delta^+ = \frac{\varphi^{-z} - \varphi^{-(1+\varepsilon)z}}{1-\varphi^{-z}} \qquad (14)$$

To facilitate the comparison of REM and RAM, we adopt a network model in which link prices are independent random variables uniformly distributed on the interval $[0,1]$. This model is perhaps not representative of the true distribution of congestion prices in a real network, where a relatively small fraction of links are highly congested and the majority of links are uncongested. The benefit of using this model is its simplicity; the expected path price $E[s]$ is proportional to path length.

To gain some understanding of how the error probability behaves as path length increases under our simple network model, we generated a set of n_{max} links with prices uniformly distributed on $[0,1]$. We then compute the end-to-end marking probability for a path consisting n links where $n = 1, 2, \cdots, n_{max}$ for both REM and RAM. Since we expect the performance of REM to depend on the choice of parameter φ, we consider two different values, $\varphi = 1.01$ and $\varphi = e$, as a baseline for comparison. Finally, for fixed values for the error tolerance ε and the number of samples N, we can compute the resulting δ^+ and δ^-.

Since we know the true marking probability (given a set of link prices), we can compute the error probability (6) exactly. Treating each packet sent as a Bernoulli trial with probability of heads p, we have

$$\Pr[\hat{z} > (1+\varepsilon)\,z] \;=\; \sum_{B=\lceil n(1+\delta^+)\,p\rceil}^{n} r(n,B,p) \qquad (15)$$

$$\Pr[\hat{z} < (1-\varepsilon)\,z] \;=\; \sum_{B=0}^{\lfloor n(1-\delta^-)\,p\rfloor} r(n,B,p), \qquad (16)$$

where $r(n,B,p) = \binom{n}{b}\,p^B\,(1-p)^{(n-B)}$ is the probability mass function for a Bernoulli random variable. The error probability defined in (6) can also be written

$$err(\varepsilon) = \Pr\{\hat{z} \notin [(1-\varepsilon)s, (1+\varepsilon)s]\},$$

from which it is easily seen that $err(\varepsilon)$ is the sum of equations (15) and (16).

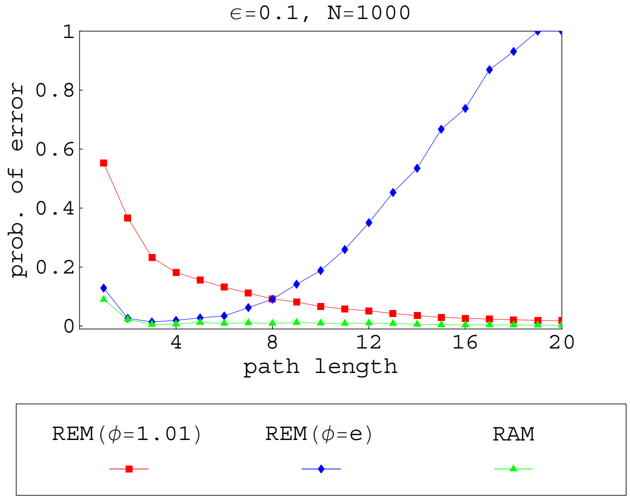

Fig. 1. Error probability as a function of path length for RAM and for two parameterizations of REM. We observe that RAM yields and error probability that is largely independent of path length and that this error probability is matched by REM only at specific path lengths, which depend on the value of the parameter φ.

Figure 1 shows the dependence of error probability on path length for two parameterizations of REM ($\varphi = 1.01$ and $\varphi = e$) and for RAM. For this plot, we have fixed the number of samples at 1000 and the error tolerance parameter ε at 0.1. The data plotted are averaged over 10 independently generated sets of link prices. We observe several interesting features in these results. First, the error probability of the RAM price estimate is unaffected by path length. Second, the REM error probability does depend on path length, with the two different parameterizations yielding error probabilities comparable to RAM at different path lengths. This result implies that the appropriate choice of φ may be path dependent. We note also that the error probability for $\varphi = e$ can approach 1 at long path lengths. This situation corresponds to an extremely high marking probability for which no unmarked packets are likely to be seen within 1000 samples.

These results suggest that RAM is well-suited for marking in a network environment where sessions see varying path lengths and path prices. We have seen that REM can also perform comparably to RAM but that its performance depends on the choice of parameter φ. To compare the two algorithms fairly, we must investigate the issue of parameter setting in REM more thoroughly.

B. *The Effect of Parameter φ in REM*

Figure 1 shows that the REM algorithm with $\varphi = e$ performs quite well at short path lengths but performs poorly for longer paths, whereas $\varphi = 1.1$ performs well on longer paths but poorly on short paths. This result suggests that a version of REM in which φ is selected according to the path length[5] might have performance comparable to RAM.

In the case of either REM or RAM, one must collect a significant number of packets in order make a close estimate of path price. The number of marked packets B is a Bernoulli random variable. However, since the number of samples is large, we may approximate B as a normally distributed random variable with mean $\mu = Np$, variance $\sigma^2 = Np(1-p)$ and CDF $F(x;\mu,\sigma)$.[6] Under this approximation, the error probability can be written

$$err(\varepsilon) \approx 1 - \int_{(1-\delta^-)pn}^{(1+\delta^+)pn} dF(x;\mu(p),\sigma(p)), \qquad (17)$$

where we have made explicit the functional dependence on p, the end-to-end marking probability.

Recall that the REM marking probability depends on the total path price s and the parameter φ. The limits of integration in (17) depend on p as does the pdf dN. Thus, the error probability is a continuous function of both s and φ.

Figure 2 shows the error probability for REM as a function of φ for values of the total path price s ranging from 0.1 to 10. Path prices ranging over three orders of magnitude is well within the realm of possibility for REM due to the varying number of hops and levels of congestion seen along different paths, and due to the fact that link price is not actually constrained to $[0,1]$ in REM. The plots shown in the figure suggest that it is impossible to fix a single value for φ that will yield a low error probability for all paths. Rather, it appears that the appropriate choice of φ is indeed path dependent. In particular, for a given path with path price s, there is an optimal parameterization $\varphi = \varphi^*(z)$ for which error probability is minimized.

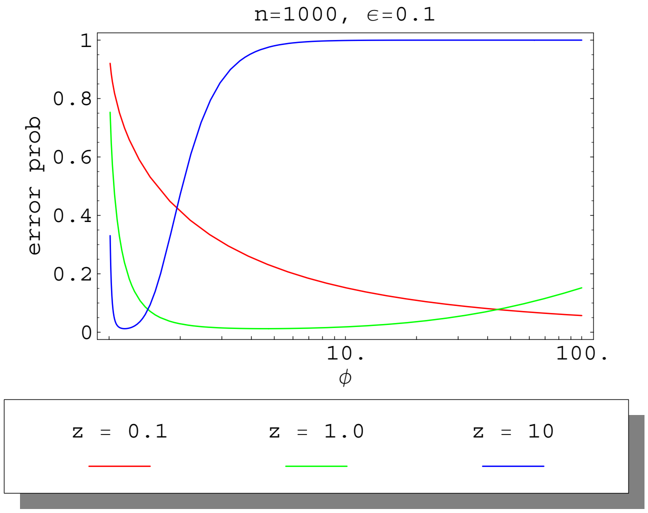

Fig. 2. Error probability as a function of REM parameter φ for three values of total path cost $S = s^n$. Note that the x-axis is on a logarithmic scale.

Although φ^* is path dependent, it is still necessary for each router along a path to use the same value when marking an individual flow. Incorporating such "path optimization" into REM would certainly add complexity to the implementation. For example, setting φ^* on a per-flow basis would require either per-flow storage at routers or including the value of φ^* in each packet header. More fundamentally, the value of φ^* depends on the end-to-end price, which is precisely the quantity to be estimated. Thus it would be necessary to jointly refine estimates of the price and φ^* as the protocol proceeds (and demonstrate the convergence of such an approach).

We next consider how well REM can perform in the best possible circumstances—if the path price (and, hence, φ^*) is known in advance. To address this question, we generated a sequence of links with prices uniformly distributed on $[0,1]$. For each experimental run, we computed φ^* for all paths starting at the first link and traversing a fixed number of hops in sequence. We obtained φ^* numerically by applying the normal approximation discussed above and then running a gradient descent algorithm on the resulting error probability function. For each path length, we collected 1000 samples of the marking bit with both RAM and a version of REM configured with φ^* for that path. We then compared the reduction in error probability as samples are accumulated for RAM and the optimally parameterized REM.

Figure 3 shows results for path lengths of 3 and 20 averaged over 10 experiment iterations. We see that RAM still performs as well or better than REM even when φ^* is known in advance.

Figure 4 shows the dependence of error probability on the error tolerance parameter ε for both RAM and optimally

[5] Recall that in the network model underlying Fig. 1, path price is proportional to path length.

[6] One rule of thumb for evaluating the validity of the normal approximation to the binomial is, for a binomial distribution with parameters N and p, that $Np \geq 5$ and $Np(1-p) \geq 5$ [7]. These conditions are satisfied in our model for $N > 200$ in the case of RAM and REM with $\varphi = e$. For REM with $\varphi = 1.1$, the conditions are satisfied for $N > 200$ for all path lengths greater than a single hop.

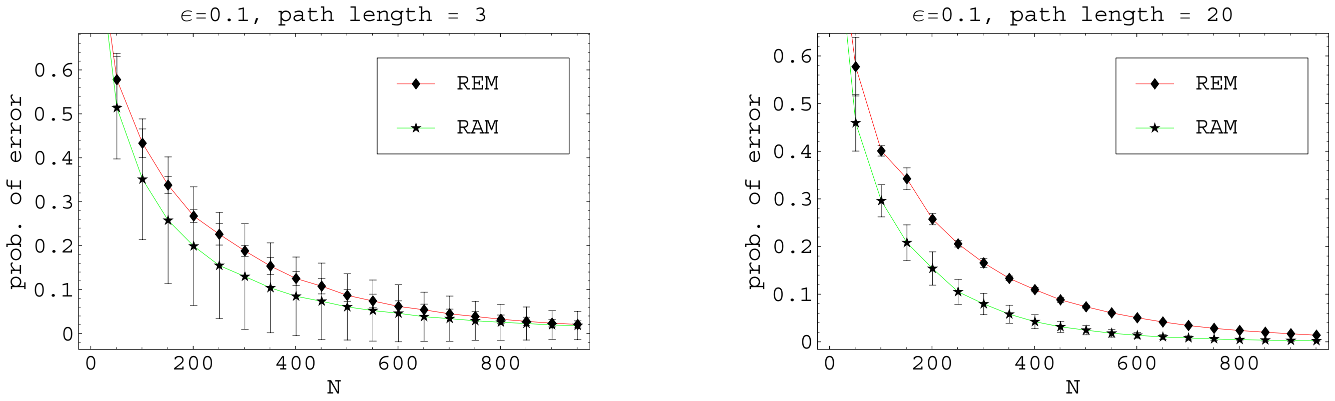

Fig. 3. Error probability as a function of number of samples for and optimally parameterized REM and for RAM.

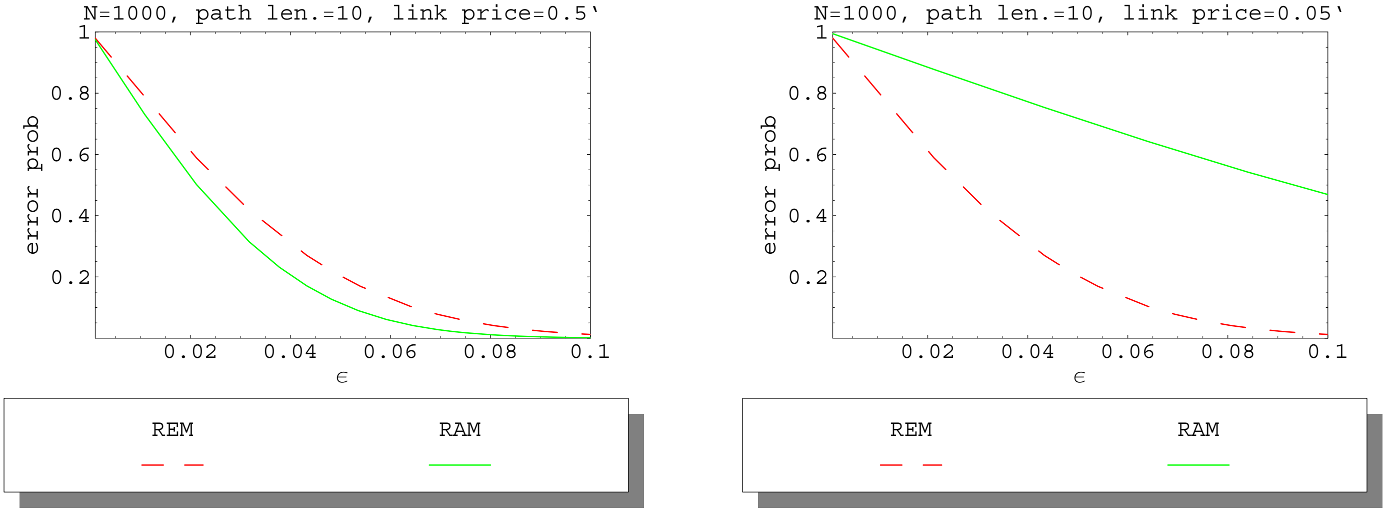

Fig. 4. Sensitivity of error probability to the parameter ε for a path of 10 links with constant price of 0.5 on the left and 0.05 on the right. These figures show the importance of correctly normalizing link price for RAM.

parameterized REM. For this analysis, we fix the price on each link to the same value and evaluate (6) for a range of ε. The figures presented use a path of 10 links and 1000 samples. In the left-hand figure the link price is set to 0.5. It turns out that RAM slightly outperforms optimized REM here. In the right-hand plot the link price is set to 0.05. Here optimized REM clearly outperforms RAM because optimized REM is able to maintain an end-to-end marking probability close to 0.5, which RAM cannot do. These results indicate that the performance of RAM relies on link prices being normalized "correctly"; at the very least, we require a mean link price close to 0.5.[7] We consider this issue in more detail in Section IV-C In related results, omitted here for reasons of space, we find *unoptimized* REM can perform as badly as RAM (or worse) if φ is set far from its optimal value.

C. Optimality in terms of Mean Square Error

The comparison among different *statistics* $\hat{\theta}$ for the same quantity θ is, in general, a multi-faceted endeavor. Several factors enter into consideration. One can compare with respect to mean, variance or higher moments, or tail distributions (as we have done above). But in terms of tail distributions, there is the choice of the parameter ε, and the comparison based on the quantity $\Pr[|\hat{\theta}-\theta| > \varepsilon]$ can vary: one *statistic* could be better than another for one setting of ε, but worse for another. In terms of convergence when the sample number $N \to \infty$, one can also discuss the rate of convergence. Finally, there is the issue of prior distribution of the parameter θ itself. If we attempt to compare REM with RAM there is the additional difficulty that in REM the estimated parameter ranges over all $[0,\infty)$ while RAM makes some a priori assumption on the range. Taking into account of all these disparate considerations, one classical choice of a measure in such cases is called *Mean Square Error* (M. S. E.) with respect to an *a priori* distribution on θ.

[7] Whether link prices can be normalized correctly remains an open question. For now, however, we will assume that such a normalization is possible for practical implementations.

For this subsection, let us define our parameter to be $\theta = \sum_{i=1}^{n} s_i/n$. To improve the prospects of REM in this comparison, we can allow the parameter ϕ to depend on n as

$$\phi(n) = \phi^{1/n}.$$

In the remainder of this section, we assume that ϕ implicitly contains this dependence.

The formulation of M. S. E. in our problem is as follows: Suppose θ has distribution $d\mu(\theta)$. For each parameter θ, the protocol constructs a 0-1 random variable Y with $\Pr[Y = 1] = F(\theta)$. Then N i.i.d. samples are taken, and the mean $\overline{Y}$ is computed. Then we estimate θ by $G(\overline{Y})$, a function of the mean. The *M. S. E.* is

$$\int_{\Theta} \mathbf{E}_{F(\theta)}[(G(\overline{Y}) - \theta)^2] d\mu(\theta).$$

As we show in Section III, REM corresponds to $F(\theta) = 1 - \phi^{-\theta}$, and $G(Y) = -\log_{\phi}(1 - Y)$ Unfortunately, REM has infinite expectation and mean square error, and consequently performs poorly in terms of M. S. E.,

$$\mathbf{E}_{F(\theta)}[G(\overline{Y})] = \infty$$

and

$$\mathbf{E}_{F(\theta)}[G(\overline{Y}) - \theta]^2 = \infty.$$

This is because there is a non-zero probability that $\overline{Y} = 1$, and then $G(\overline{Y}) = \infty$. Note that this $G(Y)$ is the inverse function of $F(\theta) = 1 - \phi^{-\theta}$ as defined over the *infinite interval* $[0, \infty)$. When we compare REM with RAM over $[0, 1]$, a natural modification to REM is to "infer" $\theta = 1$ whenever the statistic $\overline{Y} > \max F(\theta) = F(1) = 1 - \phi^{-1}$. For a more fair comparison between the two algorithms, it is reasonable to modify REM by taking its inference function G defined on $[0, 1]$ to be

$$G(Y) = \begin{cases} F^{-1}(Y) & \text{if } 0 \le Y \le 1 - \phi^{-1} \\ 1 & \text{if } 1 - \phi^{-1} < Y \le 1 \end{cases} \qquad (18)$$

Note that $[0, 1 - \phi^{-1}] = [F(0), F(1)]$, so that G is still the inverse function of F on the range of F, and thus Theorem 4 (presented below) applies. With this modification, REM no longer has infinite expectation and square error.

Theorem 4: Let F be a continuously differentiable and strictly monotonic function on $[0, 1]$ and let $G = F^{-1}$ be its inverse function defined on the image interval of $F([0, 1])$.[8] Let N be an integer ≥ 1. Let $\overline{Y} = \sum_{k=1}^{N} Y_k/N$ where $Y_1, \ldots, Y_N$ are i.i.d. 0-1 random variables with $\Pr[Y_i = 1] = F(\theta)$. Then the M.S.E. of $G(\overline{Y}) - \theta$ has an approximate order of $\frac{1}{N} \cdot \int_0^1 \frac{F(\theta)(1-F(\theta))}{[F'(\theta)]^2} d\theta$. i.e.,

$$N \cdot \int_0^1 \mathbf{E}_{N,F(\theta)}[(G(\overline{Y}) - \theta)^2] d\theta \to \int_0^1 \frac{F(\theta)(1 - F(\theta))}{[F'(\theta)]^2} d\theta,$$

when $N \to \infty$. Here $\mathbf{E}_{N,F(\theta)}$ denotes the expectation over the Binomial Distribution $B(N, F(\theta))$.

The proof of Theorem 4 (see [5]) uses Lebesgue's Dominated Convergence Theorem [8], [9], and the Chernoff bound [10]. Intuitively, when we map θ to $F(\theta)$ in order to obtain a Bernoulli random variable, the variance is $F(\theta)(1 - F(\theta))$. This variance is "amplified" or "shrunken" by a factor of $1/[F'(\theta)]^2$ when translated back to the θ-domain. The proof establishes that the integration of this quantity is indeed the controlling factor for the M.S.E. comparison. Although Theorem 4 is stated for the uniform distribution on $[0, 1]$ as the a priori distribution of θ, a similar statement can be proved for an arbitrary a priori distribution.

For REM, we obtain a lower bound for the M. S. E. using the following theorem, whose proof can be found in [5]:

Theorem 5: For every $\phi > 1$, with $F(\theta) = 1 - \phi^{-\theta}$ and G defined in (18), the M.S.E. of REM is asymptotically greater than $\frac{.54685578}{N}$. More precisely,

1)

$$N \cdot \int_0^1 \mathbf{E}_{N,F(\theta)}[(G(\overline{Y}) - \theta)^2] d\theta \to \frac{\phi - 1 - \log_e \phi}{(\log_e \phi)^3} \qquad (19)$$

2)

$$I(\phi) = \frac{\phi - 1 - \log_e \phi}{(\log_e \phi)^3}, \qquad (20)$$

is strictly monotonic decreasing in $[1, \phi_0)$, and strictly monotonic increasing in (ϕ_0, ∞), and achieves a unique minimum at ϕ_0, with value $I(\phi_0) = \frac{(\phi_0+2)2}{27(\phi_0-1)}$. Here ϕ_0 is the unique solution to the equation $\frac{1}{\log \phi} - \frac{1}{\phi - 1} = \frac{1}{3}$, and $\phi_0 \approx 8.577356793$, and $I(\phi_0) \approx .54685578$.

We now concentrate on the family of RAM-like estimators, where $F(\theta) = a + b\theta$, and G is the inverse function of F. Here, from Theorem 3, $0 \le a, a + b \le 1$ and $b \ne 0$, since F is strictly monotonic and represents a probability.

We first show that within the family of RAM-like protocols identified in Theorem 3, the RAM protocol presented in Section II is optimal in terms of M. S. E.[9] To see this, we compute the difference in variance between an arbitrary feasible RAM-like protocol and RAM itself. Note that since the probability functions for RAM-like protocols are affine linear, $\mathbf{E}[G(\overline{Y}) - \theta]^2 = \mathbf{Var}[G(\overline{Y})]$. For any RAM-like protocol,

$$\begin{aligned} \mathbf{Var}[G(\overline{Y})] &= \mathbf{Var}[\frac{\overline{Y}}{b}] \\ &= \frac{\mathbf{Var}[Y]}{b^2 N} \\ &= \frac{F(\theta)(1 - F(\theta))}{b^2 N} \\ &= \frac{1}{N}\left(\theta + \frac{a}{b}\right)\left(\frac{1-a}{b} - \theta\right) \end{aligned} \qquad (21)$$

RAM corresponds to $F(\theta) = \theta$, thus

$$\mathbf{Var}[\overline{Y}] = \frac{\theta(1 - \theta)}{N}. \qquad (22)$$

[8] We note that if F is monotonic increasing then this interval is $[F(0), F(1)]$, and if it is decreasing then it is $[F(1), F(0)]$. Moreover, where G is defined, G is also continuously differentiable and $G'(F(\theta)) = 1/F'(\theta)$.

[9] In [5] we establish the optimality of REM and REM* within the family of REM-like protocols. This result is omitted here due to space limitations.

Hence the difference

$$N \cdot [\mathbf{Var}[G(\overline{Y})] - \mathbf{Var}[\overline{Y}]] = \frac{a(1-a)}{b^2} + \left(\frac{1-b-2a}{b}\right)\theta.$$

Now we assume θ has a uniform distribution on $[0,1]$, then

$$\begin{aligned} & N \cdot \int_0^1 [\mathbf{Var}[G(\overline{Y})] - \mathbf{Var}[\overline{Y}]] d\theta \\ = & \frac{1}{2b^2}[(a+b)(1-a-b) + a(1-a)] \\ \geq & 0, \end{aligned} \qquad (23)$$

by elementary calculation, and since $0 \leq a, a+b \leq 1$.

We note that this result holds for *any* distribution on θ that has expectation 1/2. In particular, this would apply if each s_i is independently distributed and symmetric about 1/2.

The inequality (23) is strict, unless $(a+b)(1-a-b) = a(1-a) = 0$, which can happen in one of two ways (since $b=0$ is not allowed). If $a=0$ then $b=1$ and we have RAM as presented in Section II. If $a=1$ then $b=-1$ and we have the dual of RAM with $F(\theta) = 1-\theta$. We conclude that in terms of M.S.E. with respect to uniform distribution (or any other distribution with expectation 1/2) on θ, RAM (or its dual) is optimal.

The M. S. E. for RAM is readily computed as follows. Noting that $G(Y) = Y$, $E[\overline{Y}] = \theta$, and making use of (22) we obtain

$$E[(\overline{Y} - \theta)^2] = \frac{\theta(1-\theta)}{N}$$

If we assume θ has a uniform distribution on $[0,1]$, then the M. S. E. is $1/6N$ Comparing this result with the asymptotic M. S. E. for REM in Theorem 5, we conclude that, for a uniform a priori distribution of θ, no matter what choice we make for the base ϕ in REM, in terms of M.S.E. it is worse than RAM over a finite interval.

V. Implementing RAM in the Internet

Implementation of RAM on the current Internet is complicated by the fact that routers are typically not aware of their position along the path taken by a particular packet. Without this information, a router clearly cannot determine the correct marking probability for an incoming packet. One possible way to address this difficulty is to introduce a field in the IP header to be incremented at each hop which would contain the path length. However, requiring a change to a standard header would be a serious barrier to deployment. In addition, introducing a new field would effectively make additional bits available for packet marking and these bits might be used more profitably by some alternative marking scheme. Since we are interested in easily deployed single-bit schemes, we are motivated to explore other solutions.

The time-to-live (TTL) field in the IP header is an 8-bit field used to limit the maximum lifetime of a packet in the network. In addition to serving this intended purpose, the TTL field provides some information about path lengths and thus could plausibly be used by a marking algorithm. Unlike a path length field that is initialized to zero and incremented, TTL is initialized to some positive value and decremented. One problem with using TTL to perform marking within the network is that the routers along the path are unaware of the initial value placed in the TTL. Another problem is that the IP protocol allows routers to decrement the TTL value by more than one. Thus, even if a router knew the initial TTL value, it could not be sure of the number of intervening routers between it and the source on the basis of the observed value.

In the remainder of this section, we show how RAM can be implemented in the Internet using only the existing IP TTL field and a single ECN bit for marking. We do not require that routers know the initial TTL value. Instead, we will initially assume that the TTL field is always initialized to the maximum value of 255. We will show that in the case when the TTL is actually initialized to a lower value, the protocol still computes a correct estimate. Conceptually, assuming too high a value is equivalent to appending a chain of links with zero price to the beginning of the path, which collectively decrement the TTL to its actual initial value. However, overestimating the initial TTL value leads to slower convergence. We therefore adopt a heuristic, described below, to provide a much better estimate for initial TTL than the maximal field value. We also assume that each router knows the amount by which it will decrement the TTL, but require no knowledge about the behavior of other routers.

Consider the i^{th} link along a path. Assume that the link aware of its own price s_i, and the amount by which it will decrement the TTL of any packet passing through it, denoted k_i. Let T denote the actual initial TTL value and assume that the link has obtained a guess Ω of this initial value, where $\Omega \geq T$. We will find it convenient to work with the quantity $\tilde{T} = \Omega - T$, the amount by which the links overestimate T. Each arriving packet provides the router with an ECN bit having expected value θ_{i-1}, and a TTL field with value τ_i. Note that we may write

$$\tau_i = T - K_{i-1},$$

where

$$K_{i-1} = \sum_1^{i-1} k_i$$

For each packet received, the router computes

$$\begin{aligned} t_i &= \Omega - \tau_i \\ &= \tilde{T} + K_{i-1}. \end{aligned}$$

The value t_i is the path position inferred by router i and has the property $t_i \geq i$ with equality holding in the case that the TTL field is actually initialized to Ω and each preceding router only decrements the TTL by one. Also note that necessarily $t_i > t_{i-1}$ for all i.

Theorem 6: The expected value of the marking bit emerging from a chain of n routers running Algorithm 1 is

$$a_n = \frac{z_n}{(\tilde{T} + K_n)}.$$

Proof: The proof is by induction on n, the length of the router chain. The base case for $n=1$ follows trivially from the

Algorithm 1 TTL-RAM algorithm

```
Given: $s_i$, $k_i$, $\Omega$
Input: $(\tau_i, \theta_{i-1})$
With probability $\frac{t_i}{t_i+k_i}$, set $\theta_i = \theta_{i-1}$
With probability $\frac{s_i}{t_i+k_i}$, set $\theta_i = 1$
Otherwise, set $\theta_i = 0$
$\tau_{i+1} = \tau_i - k_i$
Output: $(\tau_i, \theta_i)$;
```

Algorithm definition. We provide the inductive step. Consider the expected value of the bit emerging from router i

$$\theta_i = \frac{t_i}{t_i + k_i}\theta_{i-1} + \frac{1}{t_i + k_i}s_i$$

Using the substitution $t_i = \tilde{T} + K_{i-1}$ and the fact that $K_i = K_{i-1} + k_i$, we have

$$\theta_i = \frac{\tilde{T} + K_{i-1}}{\tilde{T} + K_{i-1} + k_i}\theta_{i-1} + \frac{1}{\tilde{T} + K_i}s_i$$

By hypothesis,

$$\theta_{i-1} = \frac{z_{i-1}}{\tilde{T} + K_{i-1}}.$$

The theorem follows. □

The receiver can recover the sum of path prices using an estimate of the marking probability $\hat{\theta}_n$ and the TTL value of arriving packets $\tau_{n+1} = T - K_n$. The path price estimate is simply

$$\hat{\theta}_n \cdot (\Omega - \tau_{n+1})$$

In practice, it is extremely rare for routers to decrement the TTL by more than one. We will therefore assume henceforward that $k_i = 1$ for all i. It is also rare for sources to initialize the TTL field to its maximum value. The IP standard simply states that the TTL should be at least as large as the (unknown) diameter of the Internet [11] with 64 being a recommended value [12]. The default values chosen by popular operating systems vary between 30 and 255 [13]. There is a motivation to choose as low a value as possible to limit the lifetime of misrouted packets. Unfortunately, we expect the effect on RAM of a source setting TTL to less than the guessed value to be slower convergence since the probability of any router overwriting the marking bit would be reduced.

Figure 5 shows the convergence of RAM for three different combinations of Ω and T on a 10 link path with a price of 0.5 on each link. For each setting of the parameters we executed 10 simulation runs, collecting 10^4 packets in each run. The plots in Fig. 5 show the evolution of the minimum, mean and maximum price estimates. We see that in all three cases, the mean price estimate quickly converges to the correct value, but that the mismatch between Ω and T introduces substantial variability in the estimate. If we can ensure a small difference between the initial TTL and the guessed value, RAM can achieve extremely good performance. Fortunately, it is possible for a router to make a much better estimate for Ω than the maximum TTL value by observing the TTL of each packet as it is forwarded.

Measurements of TTL values taken within the Sprint backbone by Jaiswal, et al. [14], suggest that a reasonable guess for Ω is the smallest power of two greater than τ but at least 32. This result can be explained by observing default values chosen in practice by operating systems, which tend to equal or be slightly less (between 1 and 4) than some power of two and are never lower than 30 [13]. Despite the fact that initial TTL values are user-configurable parameters in most modern operating systems, users typically do not modify the default setting unless extremely long paths are encountered. Indeed it is likely that in many cases users do not know how to change these parameters or lack the authorization to do so. Furthermore, measurements by Begtasevic and Van Mieghen put the average path length in the Internet somewhere around 16 hops with paths of more than 30 hops being exceedingly rare [15].

Thus, we define

$$\begin{aligned} \Omega(\tau) &= [2^\lambda]_{32}^{255} \\ \lambda &= \lfloor \log_2 \tau \rfloor, \end{aligned} \tag{24}$$

where $[x]_a^b = \min(\max(a,x),b)$. Using this rule, the guessed initial TTL will likely be very close (within 4) to the actual value. In extremely rare cases, a path may be so long that the guessed TLL will change at some point along the path. Consider, for example, a packet with initial TTL of T traversing a long path. For simplicity of explanation, assume T is a power of 2. The first $k = T/2$ routers along the path will correctly set $\Omega = T$. At router, $k+1$, however, $\Omega = T/2$. The expected value of the marking bit arriving at this router is $\theta_k = z_k/k$. The TTL-RAM algorithm at link $k+1$ will assume it is the first link along the path since $t_{k+1} = 0$ and therefore overwrite the arriving bit with probability one, destroying all information about the path prior to itself. Unfortunately, link $k+1$ cannot distinguish between being the first link in the path and guessing a value of Ω lower than preceding routers.

It can be shown that the true path price cannot be recovered by means of local corrections at the links when Ω changes mid-path. However, this situation can be detected at the receiver if the initial TTL value T is sent end-to-end by the source. Specifically, if the receiver sees that $T - \tau_{n+1} \geq T/2$ then it knows that the value of Ω changed along the path and the RAM price estimate must be regarded as biased. We emphasize that such biased estimates are very rare events. A packet with an initial TTL of 32 (the value used in older Microsoft operating systems) would be discarded by the network before generating such an event. A packet with an initial TTL of 60 (a value used in several real-world operating systems) would have to traverse 28 hops before reaching a TTL of 32. A packet with an initial TTL of 128 (the default value for newer Microsoft operating systems) would have to traverse 64 hops.

VI. Conclusion

In this paper we have considered the problem of estimating the sum of congestion prices along a path using a one-bit probabilistic packet marking algorithms. We showed that

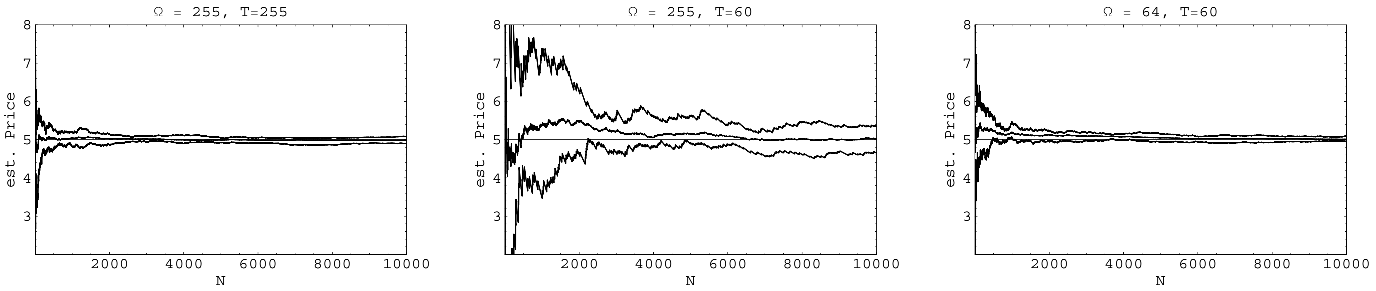

Fig. 5. Convergence of RAM using the TTL field for different combinations of Ω and T.

REM, the only previously proposed algorithm we are aware of, is, in fact, essentially unique if link prices are unbounded. By introducing a finite bound on link prices and allowing links to know their position along a packet's path, we found that an alternate class of algorithms becomes possible. We introduced RAM, a novel marking algorithm and showed that RAM together with the existing REM algorithm represent the only two possible classes of marking algorithms when link prices have finite bounds. By examining the tail probabilities of the two price estimates, we demonstrated the difficulty in setting the parameter φ in REM, which makes REM difficult to deploy in heterogeneous network environments. Furthermore, we showed that in terms of mean squared error, RAM out-performs even an optimally parameterized REM when prices are uniformly distributed. Finally, we showed that path position information required by RAM is already available in the form of the TTL field in IP.

The feasiblity of RAM depends on whether link prices are, in fact, bounded. This is a strong assumption, given the nature of congestion prices, which represent gradients and thus can, in principle, take on infinite values. However, this assumption is not as unrealistic as it might first appear. Prices may be explicitly bounded when they are are defined in terms of a link cost function with bounded slope, as they are, for example, in the work of Gibbens and Kelly [2], who adopt a loss-based cost model for which prices are explicitly bounded by $[0,1]$. Even in cases where prices are not naturally bounded, it may be desirable to work with a truncated price range. Paganini and collaborators have recently shown that enforcing an upper bound on price can simplify problem of setting the REM parameter φ and argue that natural price ranges do in fact exist for realistic networks [16]. Of course, truncating prices enables RAM as a feasible alternative. A comparison of RAM and REM under truncated prices would thus be a natural direction for further study.

More generally, comparing REM and RAM under realistic conditions remains a challenging problem. Future work to be done in this area includes accurately characterizing the prior distribution of link prices and path prices in large networks. RAM performs best when link prices can be effectively normalized to a finite range, symmetrically distributed about a mean value. Such a distribution is unrealistic, however, if only a few links on any given path are likely to be congested. Another open question is how to relate the performance of price estimation algorithms to the performance of the congestion control schemes in which they are embedded.

Acknowledgments

The authors would like to thank Frank Kelly and the anonymous reviewers for their insightful comments on early drafts of this paper.

References

[1] K. Ramakrishnan, S. Floyd, and D. Black, "The addition of explicit congestion notification (ECN) to IP," IETF, Tech. Rep., 2001, RFC 3168.

[2] R. J. Gibbens and F. P. Kelly, "Resource pricing and the evolution of congestion control," *Automatica*, vol. 35, 1999.

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

[4] S. Athuraliya and S. H. Low, "Optimization flow control II: Implementation," Caltech, Tech. Rep., 2000.

[5] M. Adler, J.-Y. Cai, J. K. Shapiro, and D. Towsley, "Estimation of congestion price using probabilistic packet marking," University of Massachusetts at Amherst, Tech. Rep., 2002, uM-TR-2002-23, http://www-net.cs.umass.edu/~jshapiro/um-tr-2002-23.pdf.

[6] K. Weihrauch, *Computable Analysis. An Introduction*. Springer-Verlag, 2000.

[7] W. A. Rosenkrantz, *Introduction to Probability and Statistics for Scientists and Engineers*. McGraw Hill, 1997.

[8] W. Rudin, *Real and Complex Analysis, Third Edition*. McGraw-Hill, 1987.

[9] P. R. Halmos, *Measure Theory*. Van Nostrand, 1950.

[10] N. Alon and J. H. Spencer, *The Probabilistic Method, Second Edition*. John Wiley & Sons, 2000.

[11] R. Braden, "Requirements for internet hosts – communication layers," IETF, Tech. Rep., 1989, RFC 1122.

[12] J. Reynolds and J. Postel, "Assigned numbers," IETF, Tech. Rep., 1994, RFC 1700.

[13] Swiss Academic and Research Network, "Default TTL values in TCP/IP," 1999, "http://www.switch.ch/docs/ttl_default.html".

[14] S. Jaiswal, G. Iannaccone, C. Diot, J. Kurose, and D. Towsley, "Measurement and classification of out-of-sequence packets in a tier-1 IP backbone," in *Proc. Infocom*, San Francisco, CA, 2003.

[15] F. Begtasevic and P. V. Mieghen, "Measurements of the hopcount in Internet," in *Proc. Passive and Active Measurement*, 2001.

[16] 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, http://netlab.caltech.edu/pub/papers/scalable2.ps.