Efficient Deterministic Packet Marking

Lachlan L. H. Andrew*,†, Stephen V. Hanly*, Sammy Chan† and Tony Cui*,†

*ARC Special Research Centre for Ultra-Broadband Information Networks (CUBIN)
Department of Electrical Engineering, University of Melbourne, Australia
† Department of Electrical Engineering, City University of Hong Kong.

***Abstract*— An efficient method is presented for signalling link price information using single-bit marks. The algorithm exploits side information in the `IPid` field of the IP header to allow the maximum price on a flow's path to be estimated. The algorithm automatically adapts the resolution with which the price is quantised, depending on how quickly the price changes, and allows non-uniform quantisation to be used. The algorithm does not depend on the number of hops in a path. A marking scheme with improved compatibility with RFC 3168 is also proposed.**

I. Introduction

Many congestion control algorithms have been proposed which require explicit feedback of congestion ("price") information from routers [1]–[10]. RFC 3168 [11] provides two "ECN" bits in the IP header for this purpose. Pricing information can be transmitted by randomly marking packets with these bits [2], [12]. It has recently been proposed [13] that the process of setting these bits take into account "side information" contained in the IP header. This idea was extended by Thommes and Coates [14] to provide an efficient, deterministic marking algorithm, using the value of the `IPid` to assist in conveying the base-two representation of the price. (We use the terms "base-two" and "single-bit marking" to avoid confusion over the common use of "binary" for both concepts.)

The present paper applies the idea of using the `IPid` field to the task of transmit the unary representation of the price. This approach has many benefits, such as automatically adapting the quantisation resolution of the price to the rate at which the price changes, so that static values can be estimated precisely, while rapidly changing values can be tracked quickly. A notable way in which it differs from previous marking schemes is that it conveys the *maximum* link price, as used in [5], rather than the sum of the prices, as used in [1]–[3]. Indicating the maximum of the link prices on the path can yield (weighted) max-min fairness, rather than maximising the "utility" of the network. Until now, no algorithm has been proposed which can communicate the maximum price using single-bit marking.

Unlike previous approaches to deterministic marking, the proposed algorithm does not need to probe each router along the path separately. This means that the price may be estimated accurately with many fewer packets, allowing changing prices to be tracked more accurately.

After a description in Section II of how the `IPid` field was used in [14], the new DMTM marking scheme is described in Section III. Section IV investigates the estimation accuracy achieved by DMTM, when estimating either static or changing link prices, and this performance is compared qualitatively with that of other schemes in Section V. Section VI numerically demonstrates the effectiveness of DMTM and quantitatively compares it with alternatives. Finally, a wide range of implementation issues are covered in Section VII, such as the compatibility with existing use of the ECN bits, incremental deployment, and the impact of IP tunnels.

II. Using `IPid` for packet marking

Communicating pricing information by packet marking has several constraints. It must not assume that routers retain state-information about each flow, and it must be robust to the reordering or loss of individual packets. This precludes the use of the traditional approach to single-bit quantisation, sigma-delta coding [15].

Thommes and Coates [14] proposed a deterministic algorithm for communicating congestion prices, which uses side information in IP packets. The `IPid` field is set by the sender and used when reassembling fragmented packets to identify which IP fragments belong to the same original IP packet; it will differ for all IP packets in close proximity. The key proposal of [14] was to use this field to specify how the ECN mark in a packet should be determined.

In the algorithm of [14], a router quantises its link price to n levels, yielding a $\lceil \log_2 n \rceil = b$-bit base-two number. A hash function of the `IPid` field determines the *probe type* of a packet. When a packet of probe type i is transmitted, the router marks the packet if bit i of the quantised price is 1.

In order to communicate the sum of prices along a path of at most h hops, the algorithm introduces h probe types for each bit position. Denote the probe types by the pair (h_i, b_i), where h_i is a hop number and b_i the number of a bit position. Following [13], each router determines its "number" from the time to live (TTL) field of the IP header, in this case as (TTL mod h). For probes of type (h_i, b_i) only router h_i will mark the packet, if bit b_i in its price is set. From this, the receiver can determine the price of each hop on the path.

The actual algorithm of [14] makes sophisticated use of the fact that RFC 3168 [11] specifies two bits for explicit congestion notification (ECN), reserving the combination 00 to mean that ECN is not supported, but leaving three possible mark values. This allows the algorithm to obtain data from up to six routers along the path with a single ECN probe. This requires

$$T_{\text{sum}} = 2b\lceil h/6 \rceil \tag{1}$$

probe types for a b-bit quantiser and paths of at most h hops. At least T_{sum} packets corresponding to a given price must be received before the price can be estimated reliably.

We now apply Thommes and Coates' concept of probe types to a simpler form of marking.

III. Single bit marking for maximum prices

The original random marking schemes of [1]–[3],[12] essentially used unary encoding of signals; the price is estimated as the number of bits received, requiring at least $n-1$ packets to signal n different values. Adding prices was performed implicitly by the independent marking by the routers. Deterministic marking [14] allows more efficient base-two encoding to be used; however, this requires explicit adding of the link prices, and the number of packets required increases linearly with the maximum path length.

In the present paper, we focus attention on encoding the maximum price along a path. When unary encoding is used with deterministic marking, it is simple to calculate the *maximum* of the prices along the path. Deterministic Multi-Threshold Marking (DMTM) is a simple algorithm which implements this, as follows.

In general, link prices can have arbitrary positive values. The algorithm is most easily understood if the true link price, p, is first mapped to a price q in the interval $[0,1]$, by a possibly non-linear mapping. Define $\theta^{-1} : \mathbb{R} \mapsto [0,1]$ to be an increasing mapping from link prices into the interval $[0,1]$, with inverse θ.

Similarly, define a mapping $F : \{0,1\}^{16} \mapsto [0,1]$ from 16-bit `IPid` values approximately uniformly into the interval $[0,1]$. Suitable forms of the function F are discussed below in Sections III-A, and III-B. Section VII provides more details on the choices available, and also on the choice of θ. The discussions in this paper will be in terms of q, and apply to any choice of θ.

When a router transmits a packet, it will mark the packet if the link price, p, and `IPid` value, d, satisfy $\theta^{-1}(p) = q > F(d)$. Otherwise, it leaves the mark unchanged. The value $i = F(d)$ is analogous to the *probe type* of [14], but is an approximately continuous quantity.

At the receiver, the mark of a packet of probe type i will be set if any router on the path had a price, p, exceeding $\theta(i)$. Decoding is simple. The receiver maintains a current estimate of the price, $\hat{p}$. If it sees a marked packet of probe type i with $\theta(i) > \hat{p}$ or an unmarked packet of probe type i with $\theta(i) < \hat{p}$, then it sets $\hat{p}$ to $\theta(i)$. If p (and hence q) is constant, the smallest probe type for which an unmarked packet has been received is an upper bound on the price, q, and the largest probe type for which a marked packet has been received is a lower bound.

There are many possible forms for the function F which maps the `IPid` value, d to the threshold for q. This section describes two; implementation issues of the more sophisticated one are futher discussed in Section VII-B.

A. Random thresholds

One approach is to use a pseudo-random mapping, so that consecutive packets have independent thresholds uniformly distributed on $(0,1)$. This approach is robust to the order in which the source generates the `IPid` values. As discussed in [14], some sources generate approximately sequential values, and some generate pseudo-random values. Yet others count sequentially and then swap the order of the two bytes (corresponding to counting on a little-endian architecture). If F is a pseudo-random mapping, then all of these will yield independent identically distributed (i.i.d.) thresholds.

B. Bit-reversed counting

An alternative to random thresholds is to exploit the fact that the source has the freedom to generate `IPid` fields consecutively for each given destination. If the sequence of d's are known to form a sequence of consecutive integers, it may be possible to produce an optimal sequence of thresholds. Let us first consider a suitable sequence of thresholds, and then explore how to obtain that sequence based on the `IPid` field.

Let $R : \mathbb{Z}^+ \mapsto [0,1)$ be a function which reverses the bits in the base-two representation of its argument, and places a (binary) "decimal point" in front of them. That is, for a base-two integer $\ldots b_2 b_1 b_0$,

$$R\left(\sum_{i=0}^{\infty} b_i 2^i\right) = \sum_{i=0}^{\infty} b_i 2^{-1-i}. \tag{2}$$

For example, $R(1) = 0.1_2 = 1/2$, $R(2) = R(10_2) = 0.01_2 = 1/4$ and $R(3) = R(11_2) = 0.11_2 = 3/4$, where a subscript 2 denotes base 2.

The sequence $R(1)$, $R(2)$, $R(3)$, $R(4)$, ... is a very suitable sequence for the thresholds. It performs the equivalent of a binary search without feedback; that is, the thresholds divide the interval $[0,1]$ into regions, and the R values systematically bisect the largest region to form smaller regions. This sequence can be achieved by setting $F = R$, and using consecutive `IPid` values, d, starting from 1 for each connection. Call this "pure bit-reversed counting". If d does not start from 1 ("random bit-reverse counting"), then performance is reduced slightly, but numerical results in Section VI indicate that it still outperforms random ordering of thresholds.

The maximum resolution is limited by the number of distinct values d can take. After all 2^{16} possible thresholds have been probed, a fixed q is known to within 2^{-16}.

If the price, q, is distributed uniformly on $[0,1]$, this "bit-reversed counting" sequence of thresholds performs much better than random thresholds, as shown in Section IV. On connections with low bandwidth delay products, it may be possible to obtain a better sequence than bit-reversed counting by allowing the sender to adjust the sequence of `IPid` values sent in response to the current price estimate. This is the subject of ongoing research.

IV. Performance Analysis

The performance of a packet marking scheme can be measured by how precisely it can communicate the link

price information, and how rapidly it responds when a price changes. These two issues will be looked at in turn for DMTM. In the following, the probability-zero events that the price, q, is equal to $\theta(i)$ for some i will be ignored.

A. Error bounds for a fixed price

Consider first the encoding of a fixed price, $q = \theta^{-1}(p)$. Each packet that arrives provides a bound on q; packets of probe type i tell us whether or not $q \geq i$. After k packets have arrived, there is an interval in which q is known to lie, given by $(i^-, i^+]$, where i^- is the largest probe type which has been seen such that $i < q$ and i^+ is the smallest probe type which has been seen such that $i \geq q$.

Let us assume first the following two conditions:

C1: The price, q, is constant.

C2: There have been k packets received since the last change in price, carrying thresholds $F(d_1)$, $F(d_2)$, ..., $F(d_k)$.

Under these conditions, an adaptive estimator for q, consists of the best threshold seen so far. Thus, the estimate only changes, $\hat{q}_{i-1} \neq \hat{q}_i$, if there is a probe with threshold $F(d_i)$ in the interval between the true price q and the current estimate; that is, $\hat{q}_{i-1} < F(d_i) < q$ or $q \leq F(d_i) < \hat{q}_{i-1}$. After the update, $\hat{q}_i = F(d_i)$. The error, ϵ, in the estimated price, $\hat{q}$, can be bounded above by

$$\epsilon \equiv |\hat{q} - q| < i^+ - i^-. \tag{3}$$

Lemma 1: Given condition C1 and C2, the sequence $\hat{q}_i$ is monotonic.

Proof: Follows immediately by induction: $\hat{q}_0 < q$ implies $\hat{q}_i \geq \hat{q}_{i-1}$ for all i, and $\hat{q}_0 > q$ implies $\hat{q}_i \leq q_{i-1}$ for all i. ■

Theorem 1: Given conditions C1 and C2, if $F(d_1)$, ..., $F(d_k)$ are independent and uniformly distributed on $(0, 1)$, then for large k, the distribution of ϵ is asymptotically exponential with mean

$$\mathbb{E}[\epsilon] = \mathbb{E}[|\hat{q}_k - q|] = \frac{1}{k} + o(1/k). \tag{4}$$

Proof: Consider without loss of generality the case that $\hat{q}_0 > q$. The final estimate $\hat{q}_k$ will be either $\hat{q}_0$, if no thresholds fall in $(q, \hat{q}_0)$, or $\min\{F(d_i) : F(d_i) \geq q)\}$ otherwise. First, assume that the number of packets that have occurred is not k, but $K(\eta) = \text{Poisson}(k\eta)$, for any $\eta > 0$. Then the points $F(d_i)$ for $i = 1, \ldots, K(\eta)$, when ordered, form a Poisson process of rate $k\eta$ on the interval $[0, 1]$. Denote the points of the Poisson process by $(T_j)_{j=1}^{K}$, and let T_0 and T_{K+1} be respectively the largest point of a Poisson process of rate $k\eta$ on $(-\infty, 0)$ and the smallest point of a Poisson process of rate $k\eta$ on $(1, \infty)$. Let L be the random index such that $T_{L-1} < q \leq T_L$, and let $E = T_L - q$. By the memoryless nature of the Poisson process, E is an exponential($k\eta$) random variable. The error, $\hat{q}_K - q$, is given by

$$\hat{q}_K - q = \begin{cases} E & \text{if } E < \hat{q}_0 - q \\ \hat{q}_0 - q & \text{otherwise.} \end{cases}$$

Direct computation provides:

$$\mathbb{E}[\hat{q}_K - q] = \frac{1}{k\eta}(1 - \exp(-k\eta(\hat{q}_0 - q))) = \frac{1}{k\eta} + o(1/k\eta)$$

Now consider the case that there have been exactly k packets. For large k, the expected error will be greater than that obtained after $K(\eta)$ packets, for any $\eta > 1$. Thus,

$$\begin{aligned} \liminf_{k\uparrow\infty} k\mathbb{E}[\hat{q}_k - q] &\geq \liminf_{k\uparrow\infty} k\mathbb{E}[\hat{q}_{K(\eta)} - q] \\ &= \frac{1}{\eta} \end{aligned}$$

This is true for any $\eta > 1$, and hence

$$\liminf_{k\uparrow\infty} k\mathbb{E}[\hat{q}_k - q] \geq 1$$

Similarly, for large k, the expected error after k packets will be less than that obtained after $K(\eta)$ packets, for any $\eta < 1$. Thus,

$$\limsup_{k\uparrow\infty} k\mathbb{E}[\hat{q}_k - q] \leq \limsup_{k\uparrow\infty} k\mathbb{E}[\hat{q}_{K(\eta)} - q] = \frac{1}{\eta}$$

This is true for any $\eta < 1$, and hence

$$\lim_{k\uparrow\infty} k\mathbb{E}[\hat{q}_k - q] = 1$$

which is the claim of the Proposition. ■

Note that (4) is within a factor of four of the mean absolute error quantisation of a k-level quantiser. However, this resolution is adaptive to the number of samples that have been seen, and need not be set *a priori*. If the price changes rarely (or slowly), then a large number of samples are received, and a high resolution estimate is obtained. However, if the price changes rapidly, then a good estimate $\hat{q}$ can be formed after only a small number of packets. In contrast, random marking [1]–[3],[12],[13] would need to adapt the interval over which it averages marks, and base-two marking [14] would need an adaptive quantiser resolution.

Consider now the case that the probe type sequence is bit-reversed counting. Let

$$K = 2^{\lfloor \log_2(k+1) \rfloor} \tag{5}$$

be the largest power of 2 not greater than $k + 1$, and $\Delta = k + 1 - K$. In addition to C1 and C2, the following condition will also be used in the theorem below:

C3: The price, q, is drawn from a uniform $[0, 1]$ distribution.

Theorem 2: Given conditions C1 and C2, if the probe type sequence is pure bit-reversed counting, $F(d_i) = R(i)$, then the estimation error after k packets is bounded by

$$\epsilon = |\hat{q} - q| \leq \frac{1}{K}. \tag{6}$$

If, in addition, condition C3 holds, then the pdf of the error is

$$f(\epsilon) = \begin{cases} (K + \Delta)/2 & \text{if } 0 \leq \epsilon < 1/2K \\ (K - \Delta)/2 & \text{if } 1/2K \leq \epsilon < 1/K \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

whence

$$\mathbb{E}[\epsilon] = \frac{1}{2K}\left(1 - \frac{\Delta}{2K}\right) \tag{8}$$

$$\mathbb{E}[\epsilon^2] = \frac{1}{3K^2}\left(1 - \frac{3\Delta}{4K}\right). \tag{9}$$

Proof: The thresholds $F(d_i)$ partition the set $(0,1]$ into intervals. Let $n = \lfloor \log_2(k+1) \rfloor$. After $K-1 = 2^n - 1$ packets, the intervals will all be of the form $(i/K, (i+1)/K]$, of length $1/K$. Similarly, after $2K-1$ packets, the intervals will all be of length $1/2K$. Each threshold from $F(d_K)$ to $F(d_{2K-1})$ bisects an interval of length $1/K$ to form two intervals of length $1/2K$. After k packets, there are $K-\Delta$ intervals of length $1/K$ and 2Δ intervals of length $1/2K$.

The estimate $\hat{q}_k$ will be one of the boundaries of the interval containing q. Thus, the error is bounded by the size of the interval containing q, which is at most $1/K$, establishing (6).

If C3 holds, then the probability that q lies in a particular interval is equal to the size of that interval. Thus, with probability $1-\Delta/K$ it lies in an interval of length $1/K$, and with probability Δ/K it lies in an interval of length $1/2K$. Conditioned on q lying in an interval of length l, the error is uniformly distributed on $[0,l]$. Let $f_{[a,b]}$ be the pdf of a uniform $U[a,b]$ random variable. Then the pdf of the error is $(1-\Delta/K)f_{[0,1/K]} + (\Delta/K)f_{[0,1/2K]}$, which is (7). Integration yields (8) and (9). ∎

If packets are lost, then some intervals will be merged. If i packets are lost, then the bound on the error is increased by a factor of $i+1$. The actual increase in error will be 0 unless the packet with $F(d_i) = \hat{q}_k$ is lost.

Theorem 3: Let d_0 be a random integer. Given conditions C1 and C2, if the probe type sequence is bit-reversed counting starting from d_0, $F(d_i) = R(d_0 + i - 1)$, then the estimation error after k packets is bounded by

$$\epsilon = |\hat{q} - q| \leq \frac{2}{K}. \tag{10}$$

Proof: After K packets, each region of the form $[i/K, (i+1)/K)$ will have been probed once. In the worst case, the probes lie at the far ends of the regions, yielding an interval between thresholds of length at most $2/K$. ∎

Note that DMTM is analogous to sampling the most significant bits more often in base-two marking, as suggested in [14]. However, because the lower order bits of the threshold are different for the different probe types in DMTM, increased precision can be obtained from the multiple samples, assuming the price is constant.

B. Response to changes in price

Let us now consider what happens if the price changes.

First, let us consider the error if the price increases such that q increases by δ per packet, assuming independent, uniformly distributed probe thresholds. We wish to characterise the mean square error of the estimator $\hat{q}$ that we have previously defined (as opposed to an optimal estimator, designed for this specific scenario). Assume that $\delta < 1/2$.

We begin by defining the random process that models the error. Let $d(k)$ be the `IPid` of the kth packet, and let $\epsilon(k) = q(k) - \hat{q}(k) > 0$ be the error immediately before the kth probe is processed. We assume packets are processed at constant rate, and identify a packet transmission interval with a time-slot in the discrete time model of the error process. Without loss of generality, we assume that the initial condition is such that $\hat{q}(k) < q(k)$ for all k. Clearly, the process $\epsilon(k)$ will undergo a zig-zag evolution, with steady increase at rate δ, followed by a jump in the slot after an update is detected. Let $H(k)$ be the event that the threshold of packet k, $F(d(k))$, lies in the interval $(\hat{q}(k), q(k))$, termed a "hit". If $H(k)$ occurs, then $\hat{q}(k+1) = F(d(k))$; otherwise, $\hat{q}(k+1) = \hat{q}(k)$. The process increases at constant rate δ, until the random event of a "hit", and at the time-slot following a hit, it makes a random-sized jump back towards zero.

The process $\epsilon(k)$ forms a continuous state space Markov chain, and we will show below that it can be stationary until the time that $q(k)$ reaches unity. The mean square error we calculate in Theorem 4 applies to the process in equilibrium.

Lemma 2: The Markov chain $\epsilon(k)$ can be taken to be stationary up until the time that $q(k)$ reaches unity, under the assumption that $\delta < 1/2$.

Proof: See appendix. ∎

The stationarity of the chain allows us to consider the mean square error. It is shown below that this mean square error is 2δ, implying that the mean error is less than $\sqrt{2\delta}$. Thus, the error tends to zero if the price is constant ($\delta \to 0$), and increases gracefully as the rate of change increases.

Theorem 4: If the maximum link price in increasing such that q increases by $\delta < 1/2$ per packet, and probe thresholds are independent and uniformly distributed, then in equilibrium, $\mathbb{E}[\epsilon^2] = 2\delta$.

Proof: Let $\mathbb{P}(H)$ be the equilibrium probability of a hit, averaged over the equilibrium statistics of ϵ. Let $\mathbb{P}(H|x)$ denote the conditional probability of a hit, given $\epsilon = x$, which is given by $\mathbb{P}(H|x) = x$, since the thresholds are $U[0,1]$. Averaging over the statistics of ϵ, we obtain

$$\mathbb{E}[\epsilon] = \mathbb{P}(H) \tag{11}$$

Now consider two randomly chosen adjacent hit times, T_1 and T_2, and let $X = T_2 - T_1 > 0$ denote the time between these two hits. Clearly,

$$\mathbb{P}(H) = 1/\mathbb{E}[X] \tag{12}$$

If ϵ is in equilibrium, then so is the embedded chain obtained by sampling at the hit times. Thus,

$$\mathbb{E}[\epsilon(T_1)] = \mathbb{E}[\epsilon(T_2)] \tag{13}$$

and we denote the common value by $\mathbb{E}[\epsilon|H]$. However, consideration of the conditional drift of the embedded chain provides that

$$\mathbb{E}[\epsilon(T_2) - \epsilon(T_1)|\epsilon(T_1)] = \delta\mathbb{E}[X|\epsilon(T_1)] - \frac{\epsilon(T_1)}{2} \tag{14}$$

Taking expectations in (14) and applying (13), we obtain that

$$\mathbb{E}[X] = \frac{1}{2\delta}\mathbb{E}[\epsilon|H] \tag{15}$$

Putting (11), (12) and (15) together, we obtain

$$\mathbb{E}[\epsilon] = 2\delta/\mathbb{E}[\epsilon|H] \tag{16}$$

But by Bayes' Theorem,

$$\mathbb{E}[\epsilon|H] = \int \epsilon f(\epsilon|H)\,d\epsilon = \int \epsilon \frac{f(\epsilon)}{P(H)} P(H|\epsilon)\,d\epsilon$$
$$= \int \epsilon^2 \frac{f(\epsilon)}{P(H)}\,d\epsilon = \frac{\mathbb{E}[\epsilon^2]}{\mathbb{E}[\epsilon]}. \quad (17)$$

Combining (16) and (17) gives $\mathbb{E}[\epsilon^2] = 2\delta$. ■

Another consequence of the price changing is that it may not lie within the interval in which the receiver believes it to lie. If it lies far outside the interval, this condition will be short lived. Denote the interval in which the receiver believes q to lie by $(i^-, i^+]$, and consider without loss of generality the case that q has increased such that $q > i^+$. The error will be detected as soon as a packet arrives with a probe type $i \in (i^+, q)$. Consider the probability that a step-change in price which causes $q > i^+$ will remain undetected after k packets have been received since the change in price. If probe types are sent randomly,

$$P(\text{undetected after } k) = (1 - (q - i^+))^k. \quad (18)$$

If probe types are sent according to bit-reversed counting starting from a random value, intervals of length 2^{-j} are sampled once every 2^j packets, and the probability that the sample will lie in a given sub-interval of length a is $a/2^{-j}$. Thus

$$P(\text{undetected after } 2^j) = \begin{cases} \prod_{m=1}^{j}(1 - (q - i^+)2^m) & \text{if } (q - i^+)2^j < 1 \\ 0 & \text{otherwise.} \end{cases} \quad (19)$$

In particular, the condition will be detected within $\lceil \log_2(1/(q - i^+)) \rceil$ packets.

When it is detected that $q > i^+$, the receiver can set $i^+ \leftarrow 1$, establishing an interval in which q is known to lie.

Let $\hat{q}_0$ be the estimate before the step change. Then the error will be approximately $\min(|q - \hat{q}_0|, 1/k)$ after k packets have arrived since the step change.

V. Comparison with other schemes

A. Deterministic base-two marking

If the price must be estimated with very high precision, then Thommes and Coates' approach [14] of marking based on the base-two representation of the price is asymptotically the most efficient approach. It requires $O(\log 1/\epsilon)$ packets to achieve an error of ϵ. However, the constant multiplier can be very large, and when the estimates must be made after a limited number of packets are received, other approaches are preferable.

DMTM addresses a weakness inherent to marking based on the base-two representation of the price. If the price changes between samples, say from 3 (011) to 4 (100), then a scheme which transmits the base-two representation could estimate the price as anything from 000 to 111. This cannot occur when unary coding is used, because the interpretation of each mark is independent of the values of the other marks. Base-two schemes are also vulnerable to the loss of packets carrying the most significant bits. This can be addressed by transmitting the higher order bits more often [14]. This is implicitly done by DMTM.

If the number of probe types is limited to k, then DMTM approximates a k-level quantiser. For a 16-level quantiser, as proposed in [14], we only require 16 probe types, regardless of the length of the path, rather than 40 for paths of up to $h = 30$ yielded by (1). More importantly, DMTM provides good estimates even after a small fraction of the probe types have been received, as is shown in Section IV. This allows high resolution quantisation to be used, with the effective resolution of the quantiser adapting to the number of samples available.

B. Random early marking and additive marking

Under REM [12], packets at the receiver have been randomly marked with probability $q = \theta^{-1}(p) = 1 - \phi^{-p}$, where p is the sum of the prices of the links. After k packets have been received, the estimate $\hat{q}$ is the fraction of packets which have been marked. Its variance is $q(1-q)/k$, so the mean error is $O(1/\sqrt{k})$, compared with $O(1/k)$ for DMTM.

Compare REM with DMTM using random thresholds. In both cases, the routers mark a fraction q of the packets, and the standard deviation of the actual number of packets marked is $\sqrt{q(1-q)/k}$. The difference is that in DMTM, marked and unmarked packets carry information about the specific interval in which q (or p) lies.

The decoding procedure for RAM [13] is the same as that for REM, except that it avoids the non-linear mapping. Thus, the error it observes in the final price, p, is statistically identical to the error that REM observes in the normalised price, q.

If $\hat{q}$ is estimated over a fixed time interval, as in [16], or over a fixed number of packets, then both REM and RAM required a tradeoff to be made between speed of response and maximum resolution. This is performed automatically by DMTM.

VI. Numerical results

In this section, we evaluate the performance of DMTM and compare it with other marking schemes using simulations. First, we evaluate the estimation error in DMTM after k probe packets have been received by the receiver. Here, we consider three kinds of probe type sequence: pure bit-reversed counting ("pure BRC" starting from $R(1)$), random bit-reverse counting ("random BRC" starting from $R(d)$ with d uniformly distributed on $[1, 65535]$), and pure random. We also assume that there are 2^{16} thresholds. Figure 1 plots the mean estimation error against k on a log-log scale. For pure BRC, each point is obtained by averaging the errors of 1000 different prices; for random BRC and pure random probes, they are averaged over 256 different random probe sequences, each using 100 different prices. Also plotted in the figure are the curves of $1/2(k+1)$ and $1/(k+2)$ for reference. The figure shows that BRC outperforms random probing. This is because BRC systematically generates the probe sequence such that, for a given k, more different price ranges could be probed and hence a better estimation can be obtained. Random BRC, performs like random probing for the first few packets.

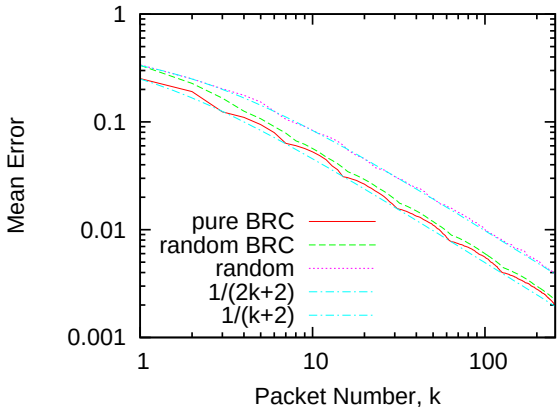

Fig. 1. Mean estimation error of DMTM after k packets have been received.

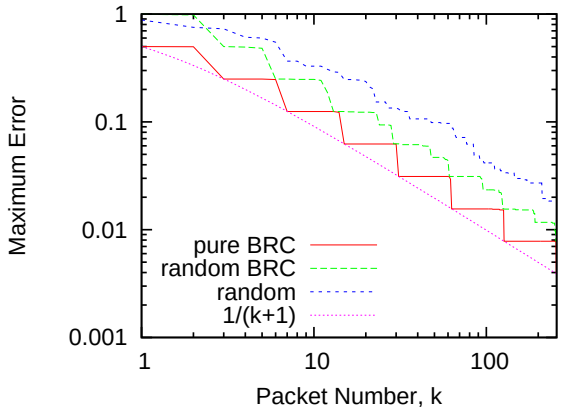

Fig. 2. Maximum estimation error of DMTM after k packets have been received.

However, as more packets are received, its performance gets closer to that of pure BRC because the feature of systematic probe type generation is preserved.

The simulation results also confirm the accuracy of our analytical prediction of estimation error. For the case of random probing, (4) states that the asymptotic mean estimation error is $1/k + o(1/k)$. This is supported by the simulation results which overlap with the curve $1/(k+2)$, which is approximately $1/k$ for large k. The small difference suggests that our analysis can possibly be refined by including the two implicit thresholds of 0 and 1 as points in the Poisson process, yielding a Poisson process with rate of $k+2$. For the case of pure BRC, the simulation results support (8), showing that for $k+1 = K$, the mean error is equal to $1/2(k+1)$, and for other k, the mean error is slightly larger than $1/2(k+1)$, but less than $1/2K$.

Figure 2 shows the maximum error taken over the same ensemble as Figure 1. For purely random probes, the error is approximately exponentially distributed (see Theorem 1), and so the maximum error is not well defined; instead the 99th percentile of error was plotted. The maximum observed error for pure BRC corresponds well to the bound of (6). After a small number of steps (small k), the maximum error observed for random BRC is approximately twice that for pure BRC, as predicted by the bound of Theorem 3. However, for large k, this bound becomes loose, and there are additional small "steps" in the graph. To obtain a heuristic understanding of these steps, notice that the bound in Theorem 3 is actually the sum of the lengths of adjacent intervals in which probes are known to occur. Consider these intervals after 2^n probes have been made. After 1.5×2^n probes, every alternate interval has been probed a second time. This causes the maximum sum of the lengths of adjacent intervals to be 1.5 times, rather than twice, the length of one of the intervals after 2^n probes.

The mean square error performance of DMTM with purely random probes is compared with alternative marking schemes in Figure 3. The results for REM [12] and RAM [13] are the closed form expression, $1/6k$, which is the average of $q(1-q)/k$ for q uniform in $[0,1]$. For RAM, this represents the

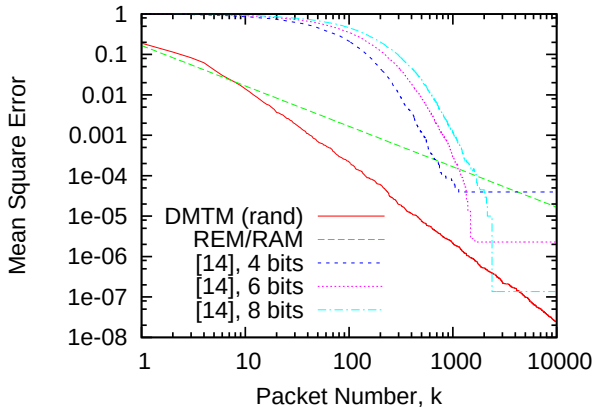

Fig. 3. Comparison of mean square estimation error after k packets have been received for different marking schemes.

error in the actual price, while for REM, it is the error after the exponential mapping (21). The results for [14] are for a version of that algorithm using single-bit marking (see Section VII-A). The curve for "[14], n-bit" used $30n$ probe types to allow for paths of up to 30 routers with n bit quantisation. The actual path measured had 10 routers, yielding aggregate prices in the range $[0, 10]$; to avoid bias against this scheme, the prices were scaled to the range $[0, 1]$ for this figure. The order of probe types was random.

The results for DMTM and REM/RAM show the expected power law behaviour, with DMTM yielding significantly lower error after a moderate number of packets. The results of [14] are more complex. When only a small fraction of the probe types have been received, there is a high probability of high-order bits not being received, yielding a large mean square error. The error then drops rapidly after about $30n$ packets, as most probe types have been observed. However, because a fixed quantiser is used, there is an square-error floor at $2^{-2n}/12$ per node ($2^{-2n}/120$ for the average of 10 nodes). This clearly shows the tradeoff inherent in the scheme of [14] between responsiveness and steady state accuracy.

Finally, the ability of DMTM and the scheme of [14] to

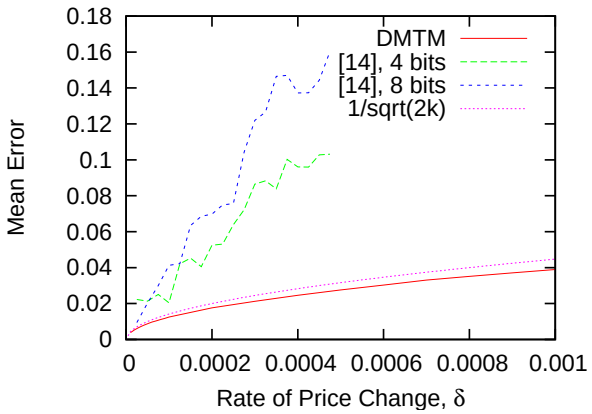

Fig. 4. Mean error when tracking an increasing price.

track a changing price is considered in Figure 4. Only the maximum link price on the path was changing. All other prices were taken to be 0. As a result, the error for each scheme is in the range $[0,1]$, and there is no need to scale the results for [14]. The ability of REM and RAM to track changing prices depends on the "forgetting" mechanism used in averaging the random marks, and is beyond the scope of this paper.

To select meaningful values for the rate of change of price, δ, recall that increasing the price from 0 to 1 corresponds to telling the sources to go from transmitting at the maximum rate permitted by the standard (say 160 Gbit/s) to the minimum possible rate (say 10 kbit/s). A change of 1/1000 of this magnitude might occur on a scale of between every packet and every 100 packets. The numerical results considered δ in the range 10^{-5} to 10^{-3} per packet.

Because DMTM can estimate the price closely from a small number of probes, it tracks the changing price much more closely than the scheme of [14], with a mean error slightly below $\sqrt{2\delta}$ as shown in Theorem 4.

VII. Implementation issues

This section highlights some issues which must be considered before implementing DMTM, most of which are shared by any explicit marking scheme. For some issues there is a clear resolution, while for others, a variety of tentative solutions are discussed.

A. Marking using the ECN bits in IP

Packet marking in IP is achieved using the two ECN bits proposed in RFC 3168 [11]. This states that a source should set these bits to "codepoint" 00 to indicate that it does not understand ECN, or 01 or 10 to indicate an unmarked packet. Routers set the bits to 11 to indicate congestion, which the source must treat as equivalent to a packet loss, that is, by halving its window. RFC 3168 also says that a router should only set the bits to 11 if it would otherwise drop the packet. Here we will call such mark "loss-equivalent" marks, which we distinguish from pricing-based marks. If a marking scheme sends codepoint 11 frequently [12]–[14], this will cause RFC 3168-compliant flows to slow to a crawl. This can be avoided by using 10 to indicate no mark, 01 to indicate a mark due to pricing, and 11 to indicate a loss-equivalent mark. Unfortunately, this interferes with the ingenious use of the three non-zero codepoints in [14]. To be compatible with RFC 3168, [14] can be modified to probe a single router with each probe type, requiring $T_{\text{sum}} = bh$ probe types.

RFC 3168 recommends that protocols not requiring two different codepoints to represent unmarked packets should use 10 in preference to 01, for backwards compatibility with RFC 2481 [17]. The above proposal to use 10 to indicate no mark and 01 to indicate a price-induced mark allows the system to work in systems where the sender is unaware of the marking scheme, and receiver-based flow control is used, such as CLAMP [4].

If loss-based congestion control becomes widely replaced by pricing congestion control, it may be possible to use all four codepoints to convey pricing information. In that case, each `IPid` should have lower, middle and upper thresholds, $F_l(d)$, $F_m(d)$ and $F_u(d)$, which specify four large intervals. The value of the mark generated internally by the router will then be 00, 10, 01 or 11, depending on which interval the price lies in. This value will be placed in the ECN field of the packet if it is greater than the current ECN value.

This approach allows a smooth upgrade path from the current RFC 3168. A sender transmits 10 to indicate that the packet is unmarked, and that it will interpret 11 as a packet loss. Such packets should be marked with 01 if $p > \theta(F_m(d))$, and unchanged otherwise. A sender which transmits 00 indicates that it will not interpret 11 as a packet loss. Such packets can be marked using all four codepoints, as described above.

Spacing $F_l(d)$, $F_m(d)$ and $F_u(d)$ widely increases the information content of the mark, and allows DMTM to track price changes faster. The proposed backward compatibility requires that F_m must cover the entire interval $[0,1)$, and so must sometimes be close to either $F_l(d)$ or $F_u(d)$. Suitable choices are $F_l(d)F_m(d)/2$ and $F_u(d) = (1 + F_m(d))/2$, which bisect the two intervals of q values produced by $F_m(d)$. Without the need for backward compatibility, a better choice would to set $F_l : \{0,1\}^{16} \mapsto [0,1/3)$, $F_m(d) = 1/3 + F_l(d)$ and $F_u(d) = 2/3 + F_l(d)$. This approach clearly generalises to m-ary marking for arbitrary m.

The analysis of Section IV is essentially unchanged using m-ary thresholds. The primary difference is that the probability that a given interval (a,b) will be probed is increased from $b-a$. With the backward compatible ternary scheme (F_m covering $[0,1)$), the probability becomes

$$3(b-a) - \max(b-2a,0) - \max(2b-a-1,0)$$

while with the m-ary symmetric case, it becomes

$$\min(m(b-a),1).$$

For $m = 3$, both approach $3(b-a)$ for small intervals, which $(\hat{q},q)$ will generally be. This shows that the loss in performance due to the backward compatible scheme will be small when the price is tracked accurately.

B. Issues with bit-reversed counting

One potential problem with bit-reversed counting is that the source does not have complete freedom over the values of the `IPid` field. The fragmentation/reassembly process requires values to be unique for a given source-destination pair for the duration of the "packet reassembly timeout", which is up to two minutes [18]. Thus, if a source has multiple connections to the same destination, then `IPid` values will be divided between the two connections. In the worst case, a given connection may observe only even `IPid` values, in which case it would observe only thresholds in the interval $(0, 0.5)$. This would make it unable to estimate any price larger than 0.5.

A solution would be to do price estimation at the network layer rather than the transport layer. If the source node uses a separate `IPid` counter for each destination, then the destination node receives the entire sequence (excluding random packet loss), and can thus estimate the true price of the path.

This solution works well unless there are multiple connections between the same source and destination with different paths, possibly resulting from different quality of service requirements. In such cases, it may be better to perform estimation separately for each connection. This relies on the random interleaving of the packets from the different connections to allow a sufficiently wide range of thresholds to be observed. The details would be implementation dependent.

If price-based flow control is implemented by the receiver, as in CLAMP [4], then DMTM can be implemented with no modification to the sender. In this case, it is important that the mapping F provide a suitable sequence of thresholds for all commonly implemented sequences of `IPid` values.

Common sequences reportedly include sequential values, pseudo-random values, and "byte-swapped" sequential values [19]. In the first two cases, setting $F = R$ is suitable, yielding bit-reversed counting or pseudo-random thresholds respectively. However, byte-swapped counting, in which $d = 256A + B$ while $256B + A$ is incremented between packets, yields a very poor sequence of thresholds; runs of 256 packets have thresholds equal in the first 8 bits making the marks highly correlated.

A more robust solution is to set

$$F(d) = F(256A + B) = R(256A + (B \oplus A)), \qquad (20)$$

where $\oplus$ denotes exclusive-OR. Again, for pseudo-random d, this yields pseudo-random thresholds. If d values are sequential or byte-swapped sequential, then it yields a sequence which has most of the desirable properties of bit-reversed counting. This is because the 8 high-order bits again form a bit-reversed counting sequence, and there are again 2^{16} distinct and equally spaced possible thresholds.

C. Nonlinear mapping of prices

Most single-bit marking schemes [1], [3], [13], [14] involve the step of mapping link prices into the interval $[0, 1]$. DMTM allows increased flexibility in how this is performed. In Kelly's original proposal [1], the mapping was a linear mapping such that the maximum price mapped to a value much less than one; that was necessary for the superposition of marks to approximate the addition of prices. This was refined by Low and Lapsley [3] to use the exponential mapping

$$q = \theta^{-1}(p) = 1 - \phi^{-p}. \qquad (21)$$

This allows the receiver to determine the true sum of the prices even if the marking probability is high, and also allows arbitrarily high prices to be represented. In [13], [14], a linear mapping into the interval $[0, 1]$ is used; the mapping must be linear for the algorithm to calculate the sum of the link prices.

Since DMTM does not perform arithmetic on the "mapped" price, q, it has total flexibility about the form that the mapping, θ, takes. The choice of mapping is essentially the same as optimal design of a scalar quantiser, which has been well studied. A classic result is that the mean square quantisation error (MSE) is minimised if the density of quantisation levels is proportional to the cube-root of the probability distribution function (PDF) of the prices [20]. That is,

$$q = \theta^{-1}(p) = K \int_{-\infty}^{p} f^{1/3}(p)\, dp, \qquad (22)$$

where $f(p)$ is the PDF of the prices and K is a normalising constant. It is not clear that the MSE is the most appropriate quality measure for the price estimate, but it is likely to yield a "reasonable" mapping.

Note that the form (21) used in [3] is MSE-optimal if the prices are exponentially distributed as

$$f(p) = e^{-3p \log \phi} / 3 \log \phi.$$

To investigate the benefit obtained from non-uniform quantisation of the prices, it is informative to look at the asymptotic behaviour of the mean square error of optimal uniform and non-uniform quantisers. The Laplacian distribution is the symmetric version of the exponential distribution, with probability density function (pdf) $p(x) = e^{-\sqrt{2}|x|}/\sqrt{2}$. When uniform quantisation is used, there is a tradeoff between the dynamic range and resolution of the quantiser. It has recently been shown that the mean square error of an optimal N-level uniform quantiser for a Laplacian distribution is approximately [21, Equation (34)]

$$\text{MSE(uniform)} \approx \frac{2}{3} \frac{(\log N)^2}{N^2} \qquad (23)$$

for large N, while the mean square error for a non-uniform quantiser is

$$\text{MSE(non-uniform)} \approx \frac{1}{12N^2} \left(\int p^{1/3}(x)\, dx \right)^3 = \frac{9}{2} \frac{1}{N^2}. \qquad (24)$$

These formulae are valid asymptotically for large N; for small N, explicit values are given in [22]. The factor by which the MSE is reduced by non-uniform quantisation, MSE(uniform)/MSE(non-uniform), is plotted in Figure 5, using exact values from [22] for N up to 32. The point here

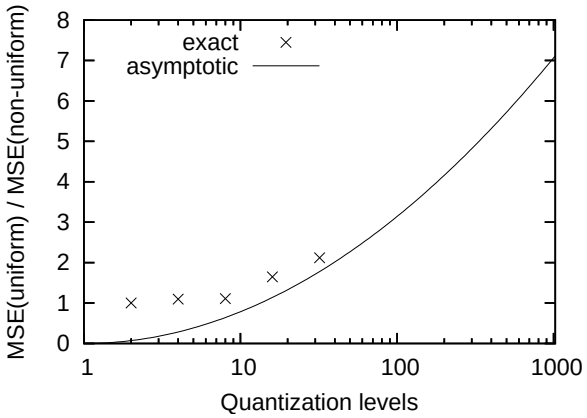

Fig. 5. Reduction in mean square error due to non-uniform quantisation.

is that there is considerable potential benefit in allowing non-uniform quantisation, and DMTM can exploit this via suitable choice of the mapping θ. The optimal mapping will depend on the specific flow control scheme employed, and studying this in detail is beyond the scope of the present paper.

Note that schemes such as [13], [14] can also be modified to use a non-linear mapping (non-uniform quantisation). If [14] uses single-bit marking (see Section VII-A) then it knows the price of each router; thus it can perform the non-linear mapping before adding the prices with no performance penalty. If [13] uses a non-linear mapping, it will no longer calculate the true sum of the link prices.

D. Protocols which do not have `IPid` fields

Many protocols, such as Frame Relay and HDLC, have single-bit congestion indication but do not have fields analogous to the `IPid` field. As defined in RFC 2460 [23], IPv6 also does not have an IPid field, as fragmentation is performed at the sender. It is possible to use a modified version of DMTM in such cases.

DMTM only uses `IPid` to be a source of pseudo-random (or sequential) data known to all routers and the destination. An alternative source of such pseudo-randomness is the packet payload.

A simplistic approach is to set d to two (or more) bytes of the payload. However, they must be two bytes that will typically differ between packets. In particular, the first few bytes of the payload will typically be a header from a higher-layer protocol and may be the same for all packets on a given connection, making them unsuitable as a source of pseudo-random data. Since the amount of header information will differ for different protocols, and with different numbers of IPv6 internal options [23], this approach is problematic.

A more robust solution is to set d to a checksum (or CRC) of the entire payload. This is a high-quality pseudo-random source unless the application data has a very high degree of redundancy, such as transmitting an uncompressed file containing all 0s.

If the protocol allows fragmentation, it would be necessary to take the checksum over only that portion of the payload which is guaranteed to be in the first fragment. In that case, the prices of all links are reflected in the mark of the first fragment, while the pricing-based marks on subsequent fragments can be discarded. Note that loss-equivalent marks on any fragment must be retained for compatibility with [11]. Taking the checksum over only a portion of the payload may also reduce the computational requirements at the routers.

If end-to-end encryption is used, then the packets will be marked based on the threshold specified by the encrypted payload. Thus, the receiver must also estimate the price based on the encrypted, not unencrypted, payload.

With the above caveats, DMTM can be applied to essentially any protocol which has single-bit congestion indication.

E. Conveying prices from destination to source

In all congestion marking schemes, it is the destination rather than the source which receives the marks, while it is generally the source which must respond to the congestion. The obvious solution of marking acknowledgements rather than data packets does not work in networks allowing asymmetric routing, and a more robust solution is required.

Since the feedback information does not need to be modified by the routers, it can be sent back by the transport layer.

One possibility would be for the receiver to use a similar single-bit marking scheme with the `IPid` value obtained as a checksum of the data packet, as described in Section VII-D. In that case, the mark bit could taken from the reserved bits in the TCP header (bits 4–7 of bytes 13 and 14 [11]). A more concise option would be to overload the CWR (congestion window reduced) bit already allocated in [11], and rely on the asymmetric nature of most TCP connections. This bit is used by the sender to indicate to the receiver that it has responded to the congestion notification; if data is only being transmitted on one direction, then the receiver never needs to signal this to the sender, but the bit is still present. Thus, it would be possible to use the CWR bit on "pure" acknowledgements (which are not piggybacked on data packets) as a mark bit.

This approach has significant drawbacks, such as the extra inaccuracy involved in using two stages of single-bit marking, the computational demands of computing checksums, the ambiguity of checksums in the presence of fragmentation, and (if the CWR bit is overloaded) problems with bidirectional TCP flows. A more effective solution may be to send the price as a TCP option. This raises the obvious question: If multi-bit feedback from the receiver to the sender is needed, what is the benefit of using single-bit marking on the forward path? The answer has two parts.

The primary reason that multi-bit feedback at the TCP layer is more acceptable than multi-bit signalling on the forward path is that it is purely end-to-end signalling, and need not be accessed by the routers. There are no available bits in the standard IP header, and so multi-bit marking on the forward path would either require IP options or a price stored in the IP payload, such as in another TCP option. IP options incur a very large performance penalty in current routers, because they must be processed in software rather than the

high speed hardware switching fabric. For this reason, many operators drop all packets carrying IP options, which makes them unsuitable for congestion marking. TCP options are not suitable for carrying congestion information in the forward path because they may be encrypted if the flow is using IPsec [24], or they may not be easily accessible if the packet is being tunnelled [25].

Another reason that it is sensible to combine multi-bit feedback in TCP options with single-bit marking is that the price feedback can be sent much less frequently. Ideally, if packet loss were negligible, the feedback would only be needed when the price changes. Using DMTM, that happens approximately every ϵ times per packet when the estimation error $|\hat{q} - q|$ is ϵ. Allowing 32 for the option (8 bits for option type, 8 for option length and 16 bits for the price [26]), this averages less than one bit per packet as long as $\mathbb{E}[\epsilon] < 1/32$. By Theorem 4, that happens when the price, q, changes by less than 0.001 per packet.

In order to be robust against packet loss, the price should be fed back occasionally even if the price is not changing. If this is done once every 32 packets, then the overhead is only 1 bit per packet. However, more sophisticated rules could be adopted. For example, the frequency of redundant transmissions could be increased if the most recent change in the estimate was large, or be larger if the change was an increase than if it was a decrease. At the cost of increased complexity, the spacing between redundant transmissions of the price signal could double after each retransmission, being reset after each actual price update.

F. Incremental deployment

There are three questions to ask with respect to incremental deployment: will benefit be obtained by partial deployment, can the system coexist with previous generation technology, and how much needs to be standardised before deployment can commence?

The answer to the first question is determined by the congestion control algorithms which use the marking, rather than the marking scheme, and is beyond the scope of this paper.

Using the marking scheme described in Section VII-A, it is possible for single-bit pricing-based marking to coexist with ECN based on RFC 3168. Even before end systems are able to interpret the marks, it is possible to start deploying routers which implement DMTM marking. One caveat is that this mechanism would interfere with the experimental nonce mechanism of RFC 3540 [27]. However, this is only a concern if the sender penalises a receiver which returns incorrect nonce-sums [27]; since many receivers currently do not support RFC 3540, senders cannot currently enforce that mechanism, which means that the proposed mechanism can be used.

For DMTM to be able to estimate the maximum q of the links on a path, the mapping F from `IPid` value to q must be standardised across the entire network. In order to estimate the maximum price, p, it is also necessary that the mapping θ be standardised, which will require further investigation. However, it is not required that all routers calculate the prices in the same way; some could signal the queueing delay, other could use the delay of a "virtual queue" [28], while others could encourage queueing by setting the price to be zero if the queueing delay is less than a threshold [4]. Similarly, different end systems can apply different congestion control algorithms, and can control the actual probe sequence by controlling the order in which `IPid` values are sent. None of this needs to be standardised for the deployment of routers which implement the marking.

G. Tunnels

For DMTM to work over tunnels, the d value of the encapsulating header must be equal to that of the tunnelled packet. In IPv4, this can be achieved by setting the `IPid` field of the encapsulating header to be that of the tunnelled header. Routers will then use the same threshold for marking both inside and outside the tunnel. The ECN field must be copied from the inner to outer header and then back again at the end of the tunnel.

This will not work with the approach described above for generating d from the packet payload, because the payload of the outer packet will be different from the payload of the encapsulated packet.

One solution, which requires routers to have per-tunnel state information, is for the egress point of the tunnel to estimate the maximum congestion on the tunnel, by means of the marks on the outer header, and then to mark the inner header based on this maximum congestion. This scales well if the router supports a small number of tunnels each supporting multiple packet flows, such as the tunnels between MBONE nodes, or IPsec tunnels between private networks. In such cases, this is the preferred method of implementing DMTM over tunnels.

As an aside, marking schemes relying on the TTL field to indicate the position of a router along the packets path, such as [13], [14], require a change in the way the TTL field is handled by tunnels. Both IPsec [24] and IP-over-IP [25] specify that the encapsulation/decapsulation process decrements the TTL of the inner datagram. This implies that the entire tunnel is treated as a single link. In order to for each router to know its position along the path, the TTL of the encapsulating header must instead be initialised to the TTL of the inner header at the ingress to the tunnel, and the TTL of the inner header be replaced by the TTL of the encapsulating header at the egress point. Subject to this modification, both of the above approaches also apply to the scheme of [14].

The rest of this section considers another solution to the problem of using DMTM over non-IPv4 tunnels. This solution does not require any per-tunnel state information, but relies on the 00 'non-ECT' codepoint [11], and loses many marks. Consider a packet leaving a tunnel. If the outer header has a loss-equivalent mark, then this is copied to the inner header. If the outer header has a pricing-based mark, and the threshold induced by the outer packet is greater than that induced by the inner packet, then the inner packet would also have been

marked. Conversely, if a outer header has not marked, and the threshold induced by the outer packet is less than that induced by the inner packet, then the inner packet would also not have been marked. In either of these cases, the mark can be copied from the outer header to the inner header. In the remaining cases, the ECN field is reset to to the 00 codepoint, to indicate that it is impossible to tell whether the routers on the path would have chosen to mark the inner packet or not. Unfortunately, this disables all ECN on the remainder of the path, including RFC 3168 marking.

Note that this does not work for [14], because in that algorithm a mark to one probe type cannot "imply" a mark to another probe type.

Consider now the performance of this modified algorithm. Let P be the threshold of the source packet, T be the threshold of the packet as it is seen by routers in the tunnel, and Q be the maximum congestion level along the tunnel. A negative mark (codepoint 10) will only get through the tunnel if $Q < T < P$, a positive mark (codepoint 01) will always get through the tunnel, and a positive mark will be placed by the tunnel if $Q > T > P$. Other cases result in a void ECN mark (codepoint 00).

The analysis can be broken into three cases depending on the location of the most expensive link and whether the current price estimate, $\hat{q}$, is an overestimate or an underestimate.

1) Underestimated bottleneck before the tunnel: If the bottleneck is before the tunnel and $\hat{q} < q$, then all packets with thresholds between $\hat{q}$ and q will be marked before they enter the tunnel. All positive marks get through, and so the analysis of Section IV applies unchanged.

2) Bottleneck after the tunnel, or overestimated bottleneck before tunnel: If either the bottleneck is after the tunnel, or $\hat{q} > q$, then negative marks need to get through the tunnel (either to reduce $\hat{q}$, or to be eventually turned into positive marks by the bottleneck). Thus, an informative mark only gets through if both $Q < T < P$ and either $\hat{q} < P < q$ or $q < P < \hat{q}$; that is, if $Q < T < P$ and $\min(\hat{q}, q) < P < \max(\hat{q}, q)$.

If the level of congestion in the tunnel is below $\min(\hat{q}, q)$ then the analysis is again unchanged. Thus, the error $\epsilon = |q - \hat{q}|$ is still $O(1/k)$ for large k, although the asymptotic regime will only be entered once $k > 1/(\min(\hat{q}, q) - Q)$.

3) Bottleneck inside the tunnel: If the bottleneck is inside the tunnel, then informative marks (those with $\min(\hat{q}, q) < P < \max(\hat{q}, q)$) will only get through if $Q > T > P > \hat{q}$ or $\hat{q} < P < T < Q$.

In this case, the performance degrades considerably. Consider again the case of tracking a price increasing at a rate of δ per packet. In order for the error interval to be reduced, this requires both the outer and inner packet thresholds lie within the interval, and that the outer threshold lie on the correct side of the inner threshold. Since these events are independent, the probability of probing an interval of length ϵ is $P(H|\epsilon) = \epsilon^2/2$ instead of $P(H|\epsilon) = \epsilon$. Moreover, the expected reduction in the interval becomes $\mathbb{E}[D] = \epsilon/3$ instead of $\mathbb{E}[D] = \epsilon/2$. Thus, (16) becomes

$$\mathbb{E}[\epsilon^2] = 2P(H) = 2 \times 3\delta/\mathbb{E}[\epsilon|H].$$

Unfortunately, (17) no longer holds, and it is not easy to obtain a simple expression for $\mathbb{E}[\epsilon^2]$.

In the analysis of static q, the probes no longer form a uniform Poisson process. They are still a Poisson process, but points distance x from q are thinned by a factor of x, since the threshold inside the tunnel, T, must satisfy $q < T < q + x$ for the mark to be received. Consider an initial over-estimate, $\hat{q} > q$. After k probes, the expected number of probes in the interval $(q, q + y]$ is

$$\int_0^y kx\, dx = ky^2/2$$

But for the error to be ϵ, there must have been a probe at $q + \epsilon$, and no probes in $(q, q + \epsilon)$; that is, there must have been exactly 1 probe in $(q, q + \epsilon]$. This suggests $\mathbb{E}[k\epsilon^2/2] \approx 1$, giving $\mathbb{E}[\epsilon^2] = 2/k$ and $\mathbb{E}[\epsilon] < \sqrt{2/k}$. This is $O(1/\sqrt{k})$ like REM, and with a worse constant multiplier. However, this is only the case when the most congested link is inside a tunnel. The benefits in the other, more common cases are enough to justify using DMTM instead of REM.

VIII. Conclusion

The proposed deterministic multi-threshold marking (DMTM) scheme has been demonstrated to provide a high-resolution estimate of the maximum price on a connection's path after a small number of packets are received. Moreover, this is achieved without needing to adjust parameters such as a fixed quantiser resolution, or an interval over which to average marks. The algorithm also tracks a changing price with smaller error than other schemes proposed in the literature, to which it is compared.

The algorithm is suitable for flow control algorithms attempting to achieve *max-min* fairness, rather than for flow control algorithms attempting to achieve maximum utility under more general optimization frameworks. Its robustness, and relative ease of implementation, increases the attractiveness of the *max-min* framework. The algorithm has been shown to be suitable for implementation in both the current Internet and future IPv6 networks.

IX. Acknowledgement

The authors thank Fred Baker of Cisco for helpful discussions. This work was supported by the Australian Research Council (discovery grant DP0557611), by and a grant from City University of Hong Kong (Project No. 7001584). CUBIN is an affiliated programme of the National ICT Australia.

Appendix

In this appendix, we prove Lemma 2. To this end, we first characterize the transition probability function of a related (modified) Markov chain.

The temporally homogeneous transition function for the Markov chain $\epsilon(k)$, $P(x, A) \equiv \mathbb{P}(\epsilon(2) \in A|\epsilon(1) = x)$ for $x < 1 - \delta$ and A a Borel-measurable set contained in $[0, 1]$, can be calculated using the fact that $\mathbb{P}(H|\epsilon(1) = x) = x$, and that conditional on $\epsilon(1) = x$ and H occurring, $\epsilon(2)$ is the sum

of a uniform random variable on $[0, x]$ plus $\delta - x$, whereas if H does not occur, $\epsilon(2) - \epsilon(1)$ increases by the constant δ. Thus,

$$P(x, A) = \| A \cap [\delta, x + \delta] \| + (1 - x) I[x + \delta \in A]$$

in this case.

Consider first an expansion of the state space to allow the (modified) error process to take values in $[0, 1+\delta]$, and expand the above definition of the transition function to also allow $1-\delta < x < 1$, and to allow A to be contained in the expanded state-space $[0, 1 + \delta]$. Apart from these changes, the above definition for $P(x, A)$ is retained. Further, for $1 < x < 1 + \delta$, we set the next value of the process to be $U + \delta$, where U is independently drawn uniform on $[0, 1]$. (This reflects the fact that a "hit" occurs with probability 1 when the error is $\epsilon = 1$.) Thus, for $1 < x < 1 + \delta$, the transition function is

$$P(x, A) = \| A \cap [\delta, 1 + \delta] \|$$

Since $\epsilon(k)$ must lie in $[0, 1]$, this Markov chain is not exactly the same as the error process, so we will denote the modified process by $\varepsilon(k)$.

Lemma 3: The Markov chain $\varepsilon(k)$ is ergodic, and hence has a stationary distribution.

Proof: Stability of the process ε can be verified from the conditions in Corollary 5.2 in [29]. The main issue is to demonstrate that for x sufficiently large, the drift function

$$\gamma_x \equiv \mathbb{E}[\varepsilon(2) - x | \varepsilon(1) = x]$$

is bounded above by a negative constant, and for smaller x, γ_x is bounded. These facts are easily verified, under the assumption that $\delta < 1/2$. The conditions stated in Corollary 5.2 in [29] seem to require in addition that $P(x, A)$ is strongly continuous [29] for any Borel measurable set A, to conclude that ε is ergodic, and this condition does not hold for our transition probability function. However, from the note added in proof in [29], it is in fact sufficient in our case to verify instead that the function $P(x, A)$ is weakly continuous for any Borel measurable set A, to conclude that ε is ergodic. This weaker condition holds because our state-space is a Banach space. Weak continuity is the requirement that $\int g(y) P(x, dy)$ is a continuous bounded function of x, for any continuous, bounded function $g(y)$. This is the case for our transition function $P(x, A)$, and hence $\varepsilon(k)$ is ergodic. ■

Proof of Lemma 2

Proof: We can define $\epsilon(k), \varepsilon(k), q(k)$ and $\hat{q}(k)$ on the same sample space, where $\epsilon(k)$ and $\varepsilon(k)$ share the same starting state, and both are at first driven by $q(k)$ and $\hat{q}(k)$. Both $\epsilon(k)$ and $\varepsilon(k)$ have identical sample paths until the point that $q(k)$ reaches unity. After this point, we allow $\varepsilon(k)$ to continue its evolution independently according to the above transition probabilities, whereas $\epsilon(k)$ must start decreasing toward zero. Since $\varepsilon(k)$ is ergodic, it can be started in the stationary distribution, and it follows that $\epsilon(k)$ is also stationary, until the point at which $q(k)$ reaches unity. ■

REFERENCES

[1] F. Kelly, "Charging and rate control for elastic traffic," *European Transactions on Telecommunications*, vol. 8, pp. 33–37, 1997.

[2] F. Kelly, A. Maulloo, and D. Tan, "Rate control in communication networks: Shadow prices, proportional fairness and stability," *J. Op. Res. Soc.*, vol. 49, pp. 237–378, 1998.

[3] S. H. Low and D. E. Lapsley, "Optimization flow control I: Basic algorithm and convergence," *IEEE/ACM Trans. Networking*, vol. 7, pp. 861–875, Dec. 1999.

[4] L. L. Andrew, S. V. Hanly, and R. Mukhtar, "CLAMP: A system to enhance the performance of wireless access networks," in *Proc. IEEE Globecom*, 2003.

[5] B. Wydrowski, L. L. H. Andrew, and M. Zukerman, "MaxNet: A congestion control architecture for scalable networks," *IEEE Commun. Lett.*, vol. 7, pp. 511–513, Oct. 2003.

[6] J. M. Jaffe, "Bottleneck flow control," *IEEE Trans. Commun.*, vol. COM-29, pp. 954–962, 1981.

[7] T.-J. Lee and G. de Veciana, "A decentralized framework to achieve max-min fair bandwidth allocation for ATM networks," in *Proceedings of IEEE Globecom*, vol. 3, pp. 1515–1520, Nov. 1998.

[8] A. Arulambalam, X. Chen, and N. Ansari, "An intelligent explicit rate control algorithm for ABR service in ATM networks," in *Proc. IEEE Int Conf. Commun. (ICC)*, pp. 200–204, June 1997.

[9] W. Tsai and Y. Kim, "Re-examining maxmin protocols: A fundamental study on convergence, complexity, variations, and performance," in *Proceedings of IEEE INFOCOM*, vol. 2, pp. 811–818, Mar. 1999.

[10] S. P. Abraham and A. Kumar, "A new approach for asynchronous distributed rate control of elastic sessions in integrated packet networks," *IEEE/ACM Trans. Networking*, vol. 9, Feb. 2001.

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

[12] S. Athuraliya, V. H. Li, S. H. Low, and Q. Yin, "REM: Active queue management," *IEEE Network*, vol. 15, pp. 48–53, May/June 2001.

[13] M. Adler, J.-Y. Cai, J. K. Shapiro, and D. Towsley, "Estimation of congestion price using probabilistic packet marking," in *Proc. IEEE INFOCOM*, pp. 2068–2078, 2003.

[14] R. W. Thommes and M. J. Coates, "Deterministic packet marking for congestion price estimation," in *Proc. IEEE INFOCOM*, 2004.

[15] P. M. Aziz, H. Sorensen, and J. Van Der Spiegel, "Overview of sigma-delta converters," *IEEE Signal Processing Magazine*, vol. 13, pp. 61–84, Jan. 1996.

[16] S. Athuraliya, D. E. Lapsley, and S. H. Low, "An enhanced random early marking algorithm for internet flow control," in *Proc. IEEE INFOCOM*, (Tel Aviv, Israel), pp. 1425–1434, 2000.

[17] K. Ramakrishnan and S. Floyd, "A proposal to add explicit congestion notification (ECN) to IP," RFC 2481, IETF, Jan. 1999.

[18] R. Braden, "Requirements for internet hosts – communication layers," RFC 1122, IETF, Oct. 1989.

[19] S. M. Bellovin, "A technique for counting NATted hosts," in *Proc. ACM Internet Measurement Workshop*, (Marseille, France), pp. 267–272, 2002.

[20] R. M. Gray and A. H. Gray, "Asymptotically optimal quantizers," *IEEE Trans. Inform. Theory*, vol. 23, pp. 143–144, Jan. 1977.

[21] D. Hui and D. L. Neuhoff, "Asymptotic analysis of optimal fixed-rate uniform scalar quantization," *IEEE Trans. Inform. Theory*, vol. 47, pp. 957–997, Mar. 2001.

[22] W. C. Adams and C. E. Giesler, "Quantizing characteristics for signals having Laplacian amplitude probability density function," *IEEE Trans. Commun.*, vol. 26, pp. 1295–1297, Aug. 1978.

[23] S. Deering and R. Hinden, "Internet protocol, version 6 (IPv6) specification," RFC 2460, IETF, Dec. 1998.

[24] R. Atkinson, "Security architecture for the internet protocol," RFC 2401, IETF, Nov. 1998.

[25] C. Perkins, "IP encapsulation within IP," RFC 2003, IETF, Oct. 1996.

[26] Information Sciences Institute University of Southern California, "RFC 793: Transmission control protocol," RFC 793, IETF, 1981.

[27] N. Spring, D. Wetherall, and D. Ely, "Robust explicit congestion notification (ECN) signaling with nonces," RFC 3540, IETF, June 2003.

[28] R. J. Gibbens and F. P. Kelly, "Resource pricing and the evolution of congestion control," *Automatica*, vol. 35, pp. 1969–1985, 1999.

[29] R. L. Tweedie, "Sufficient conditions for ergodicity and recurrence of Markov chains on a general state-space," *Stoch. Process. Appl.*, vol. 3, pp. 385–403, 1975.