Deterministic Packet Marking for Max-Min Flow Control

Hyung-Keun Ryu, *Student Member, IEEE*, and Song Chong, *Member, IEEE*

***Abstract*— This letter proposes a deterministic packet marking scheme that estimates the maximum link price on a communication path. The proposed scheme is simple and IP-compatible because it uses two-bit Explicit Congestion Notification (ECN) field and IP identification (IPid) field in the standard IP header for the estimation. Through simulations using real IP packet traces, we show that our scheme indeed works as designed with small estimation errors, and thus enables existing max-min flow control algorithms to serve their purpose without the need of separate out-of-band control packets to carry link prices.**

***Index Terms*— Explicit Congestion Notification (ECN), packet marking, max-min fairness, distributed flow control.**

I. Introduction

MANY of congestion-price based flow control protocols [1]–[3] require each link to maintain congestion price as a congestion signal and convey this price information back to the source. Considering the practical implementation of such protocols, the two-bit Explicit Congestion Notification (ECN) field [4] in the standard IP header has emerged as a tool to carry link price information. Based on the ECN field, two probabilistic packet marking schemes have been proposed [2], [5]. These schemes require each source to estimate the path price (the sum of link prices) by evaluating the ratio of number of ECN-marked packets to total number of packets transmitted, and support proportional fairness.

In contrast to the previous works, we concentrate on a packet marking scheme to support max-min fairness. To our knowledge, there is no packet marking scheme available for max-min flow control. Max-min flow control requires each source to know the maximum link price (or the minimum fair rate) on its path instead of the sum of link prices [6], [7]. Obviously, this needs comparison of all the link prices along a path. However, existing probabilistic marking schemes such as [2], [5] can hardly be extended to carry out such max (or min) operation.

We take a different approach. The idea is to encode link prices onto the ECN fields of multiple data packets such that comparison of link prices becomes feasible along a path. A similar idea has been used in supporting proportional fairness in [8]. However, it differs from our scheme in its objective (proportional fairness) as well as the way to encode and decode link prices, which cannot be directly extended to our scheme. The applicability of our scheme is not just limited to flow control problems. It is generally applicable to other types of networking problems where in-band signaling is beneficial in probing maximum (or minimum) of associated link metrics.

Manuscript received February 16, 2005. The associate editor coordinating the review of this letter and approving it for publication was Prof. David Petr. This work was supported in part by the Ministry of Information and Communication, Korea, under the grant for the BrOMA-ITRC program supervised by IITA, and in part by the Korea Science and Engineering Foundation under Grant R01-2001-000-00317-0.

The authors are with the Department of Electrical Engineering and Computer Science, Korea Advanced Institute of Science and Technology (KAIST), Daejon, Korea (e-mail: hkryu@netsys.kaist.ac.kr; song@ee.kaist.ac.kr).

Digital Object Identifier 10.1109/LCOMM.2005.09017.

II. Deterministic Packet Marking

A. Overview of the Scheme

Consider a set of links, L_p, forming an end-to-end path p from a source to a receiver. Associated with each link ℓ is a non-negative price s_ℓ. Let s_p denote the maximum link price along path p, i.e., $s_p = \max_{\ell \in L_p} s_\ell$. Assume that every link price s_ℓ is upper-bounded by some value $\bar{s}$, i.e., $0 \le s_\ell < \bar{s}$, $\forall \ell$. Each link applies a N-level uniform quantizer Q to its price s_ℓ. The output of Q is then the quantized link price z_ℓ, given by $z_\ell = Q(s_\ell) = \lfloor s_\ell / \Delta \rfloor$ where $\Delta = \bar{s}/N$ is the quantizer stepsize. Let z_p denote the maximum quantized link price on path p, i.e., $z_p = \max_{\ell \in L_p} z_\ell$. The set to which z_ℓ or z_p belongs is then $\Phi = \{0, 1, \cdots, N-1\}$.

We define the notion of *probe types*. Suppose that we need M probe types. We then associate each data packet with a probe type, say k, by a mapping function $k = \text{IPid mod } M$ where IPid is the IPid field value of the packet. Setting $M = \lceil N/3 \rceil$, we partition the set of quantized link prices, Φ, into M disjoint subsets (called ranges), Φ_k, $k = 0, 1, \cdots, M-1$, such that $\Phi_k = \{i | i = 3k, 3k+1, 3k+2 \text{ and } 0 \le i \le N-1\}$. Obviously, $\cup_{k=0}^{M-1} \Phi_k = \Phi$. Provided this partition of Φ, data packets belonging to different probe types will have different scopes in probing; probe type k packets are eligible to carry the quantized link prices whose value belongs to the range Φ_k and are responsible to probe the maximum within this range along a path. Let z_p^k denote this maximum. Since the cardinality of Φ_k is at most 3, the value of z_p^k can be encoded onto the two-bit ECN of a probe type k packet. Then, the source will be eventually informed of all the z_p^k's of path p by receiving the ECN feedbacks from all probe types, being piggybacked on ACK packets. However, the source cannot indefinitely wait for ECN feedbacks until all z_p^k's are available. Thus, we define the notion of *block of length* K such that the source determines the estimate of z_p, denoted by $\hat{z}_p$, once every reception of K successive ACK packets, by choosing the maximum among z_p^k's available in this block of ACK packets. Consequently, the estimate of s_p, denoted by $\hat{s}_p$, is obtained by applying the inverse quantizer Q^{-1} to $\hat{z}_p$, i.e., $\hat{s}_p = Q^{-1}(\hat{z}_p)$. In this letter, we choose Q^{-1} to output the midpoint of each range such that $Q^{-1}(\hat{z}_p) = \hat{z}_p \Delta + \Delta/2$. Then, as long as the estimate of z_p is correct, i.e., $\hat{z}_p = z_p$, the estimation error

ECN state	Quantized link price
00	no mark
01	$3k$
10	$3k+1$
11	$3k+2$

(a)

Forward direction of packet ↓

Link	z_ℓ	ECN state (upon departure)
1	15	00
2	21	01
3	8	01
4	23	11
5	22	11

(b)

Fig. 1. The operation of deterministic packet marking scheme: (a) mapping between ECN states and quantized prices for probe type k packets, (b) ECN marking example for a probe type 7 packet traversing a 5-hop path.

$|\hat{s}_p - s_p|$ will never exceed the half of quantization resolution, $\frac{\Delta}{2}(=\frac{\bar{s}}{2N})$.

An interesting performance question arises here as follows: For a given quantization resolution (given N), how many successive ACK packets (a block of K ACK packets) should be seen by the source to obtain an correct estimate of z_p such that $\hat{z}_p = z_p$? The reality is that each probe type will not appear as regularly as once every M packets (see Section II-B) so that K must be sufficiently larger than M in order to ensure that no missing probe types occur in each block. The situation becomes even worse if packet losses are present in the round trip because packet losses will incur extra missing probe types. The question is then how much larger K should be in practice for a given M (or N).

Fig. 1 illustrates the ECN marking operation of the proposed scheme. Fig. 1 (a) shows the mapping between ECN states and quantized prices for probe type k packets where the elements of Φ_k, $\{3k, 3k+1, 3k+2\}$, are one-to-one mapped to the ECN states, $\{01, 10, 11\}$, in the ascending order. Fig. 1 (b) shows an ECN marking example where a probe type 7 packet is traversing a 5-hop path. Initially, ECN of the packet is set to 00 by the source. Since $k = 7$, this packet is eligible to carry link prices whose quantized value belongs to $\Phi_7 = \{21, 22, 23\}$, and is not eligible to carry other prices. Therefore, this packet will bypass links 1 and 3 without invoking marking. At link 2, marking is invoked since $z_2(= 21) \in \Phi_7$, and ECN is updated from 00 to 01. Recall that for probe type 7 ($k = 7$), ECN state 01 represents quantized link price 21 as given in Fig. 1 (a). At link 4, marking is invoked to update ECN from 01 to 11 since $z_4(= 23) \in \Phi_7$ and the price being carried from upstream, 21 (ECN = 01), is smaller than the price of link 4, 23 (ECN = 11). Finally at link 5, no marking is invoked even though $z_5(= 22) \in \Phi_7$, because the price being carried from upstream, 23 (ECN = 11), is greater than the price of link 5, 22 (ECN = 10). So, ECN remains at 11. The ECN mark is then echoed back to the source by the receiver, being piggybacked on a probe type 7 ACK packet (see details in Section II-D).

B. Mapping between IPid fields and Probe Types

The purpose of the IPid field is to provide a mechanism for fragmentation and reassembly of long Internet datagrams [9]. Many hosts in the current Internet implement the IPid field using a simple counter, as noted in [10]. That is, successive data packets emitted by a host carry sequential IPid fields. There are exceptions; some hosts use byte-swapped counters, and others use pseudo-random number generators [10]. The IPid field is 16 bit long, whereas the number of probe types we require is only $M = \lceil N/3 \rceil$ (e.g., $M = 334$ if $N = 1,000$). Thus, we need a many-to-one function which maps 16-bit IPid fields to M probe types. Moreover, the function should be able to generate each probe type as regularly as possible. A natural choice for this function is $k =$ IPid mod M in that many IP hosts generate IPid fields sequentially and the function can output each probe type periodically once every M packets when sequential IPid fields are applied as the input.

However, the real situation is more complicated. Not only there are exceptional hosts which generate IPid fields non-sequentially but also there are cases where even if a host generates sequential IPid numbers, the numbers are shared by many flows so that each flow carries non-sequential IPid fields. Consider a server A who serves two clients B and C. Assume that A generates IPid numbers sequentially. When B and C download files from A simultaneously, two flows, A→B and A→C, will share the sequential IPid numbers generated by A so that each flow will carry non-sequential IPid fields. In order to study the impact of this IPid number sharing as well as the non-sequential IPid number generation on the estimation performance of our scheme, we use 100 real IP traces collected by downloading files from 100 different servers scattered across the world in Section III.

C. The Link Marking Algorithm

The ECN marking procedure for link ℓ can be described by the following pseudo-code. Upon arrival of a data packet:

```
Indentify probe type k
    k = IPid mod M
Compare price and mark ECN
    if (z_ℓ == 3k + 2)
        if (ECN ≠ 11)
            ECN = 11
    elseif (z_ℓ == 3k + 1)
        if (ECN ≠ 11 & ECN ≠ 10)
            ECN = 10
    elseif (z_ℓ == 3k)
        if (ECN == 00)
            ECN = 01
Forward the packet to next link
```

D. Receiver Feedback and Maximum Price Estimation

For each flow, the receiver maintains a table of size M, of which entry k contains the latest ECN state of probe type k, and runs two processes as follows: 1) upon receipt of a probe type i data packet, the receiver updates entry i by the ECN value of the packet; and 2) upon transmission of a probe type j ACK packet to the source, the receiver writes the ECN value of entry j onto the ECN field of the ACK packet. This mechanism provides a simple but effective way to echo the ECN states of the forward path back to the source, even if the receiver sends an ACK packet once every reception of L data packets ($L > 1$) as in many TCP implementations.

The source waits for ECN feedbacks and updates the estimate $\hat{z}_p$ once every reception of K successive ACK packets by computing $\hat{z}_p = \max_k z_p^k$ with available z_p^k's. The estimate $\hat{s}_p$ is then obtained by $\hat{s}_p = Q^{-1}(\hat{z}_p)$.

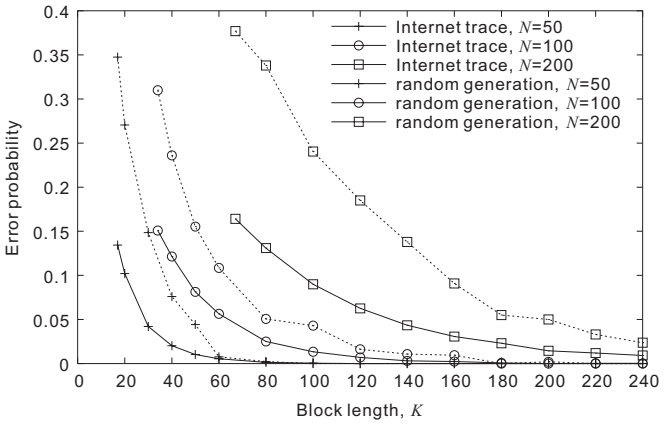

Fig. 2. The impact of missing probe types on estimation performance.

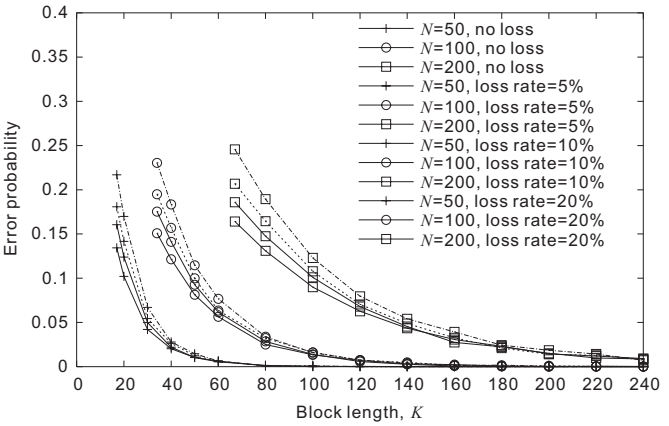

Fig. 3. The impact of packet losses on estimation performance.

III. Performance

The performance question we have is: for a given quantization resolution we desire (given N), how many successive ACK packets (a block of K ACK packets) should be seen by the source to obtain a correct estimate of z_p when packets carry non-sequential IPid fields and can be lost in the path? Recall that as long as $\hat{z}_p = z_p$, $|\hat{s}_p - s_p| \leq \frac{\Delta}{2} (= \frac{\bar{s}}{2N})$ holds. Thus, we define the error probability for a given N

$$err(N) = \Pr\left[\left|\frac{\hat{s}_p - s_p}{\bar{s}}\right| > \frac{1}{2N}\right]$$

as the measure of how missing probe types in each block affect the estimation performance. In fact, $err(N) \leq \Pr[\hat{z}_p \neq z_p]$.

In order to simulate the real IPid field patterns as seen by a typical flow in the Internet, particularly when the IPid numbers are generated by one IP host but shared by many flows, we collected 100 traces of 2,000 successive IP packets by downloading files from 100 different servers scattered across the world. We also use a synthetic model where IPid numbers are randomly generated based on uniform distribution in $[0, 2^{16}]$ for comparison purpose. We consider a 20-hop path, assume link prices are independent and uniformly distributed over [0, $\bar{s}$] (letting $\bar{s} = 1$ for simplicity), use 100 realizations of link prices, and assume the receiver sends an ACK packet for every data packet. Fig. 2 shows the error probability as a function of K for $N = 50$, 100, and 200 (correspondingly, $M = 17$, 34 and 67) in the absence of packet losses. When $K = M$, the error probability is about 0.15 in the IP trace case and about 0.35 in the random generation case for all N's. This implies that the real IPid fields incur much smaller number of missing probe types than the randomly generated IPid fields; nevertheless, $K = M$ may not be sufficient for reliable estimation. As K increases, the error probability decreases rapidly for all N's. For instance, by doubling the block length ($K = 2M$), the error probability in the IP trace case becomes roughly 0.05 for all N's. Packet losses in the path will incur extra missing probe types. Fig. 3 shows how packet losses affect the error probability for three different packet loss rates. We simulate the same scenario as in Fig. 2, except that we consider the IP trace only. Packet losses worsen the estimation performance. However, the performance degradation due to packet losses rapidly mitigates as K increases.

The results would be the same even if the receiver sends an ACK packet once every reception of L data packets ($L > 1$) as in many TCP implementations, except that the time required for the source to see K successive ACK packets would be increased roughly by the factor of L.

IV. Future Work

A question we have not tackled is how non-uniform quantization of link prices can improve the estimation performance while not increasing N. More specifically, $\lim_{K\to\infty} err(N) = 0$ does not mean that $\lim_{K\to\infty} |\hat{s}_p - s_p| = 0$ but means that $\lim_{K\to\infty} |\hat{s}_p - s_p| \leq \frac{\bar{s}}{2N}$, i.e., the error due to quantization would remain even in the limit. So, large N is preferred for a given $\bar{s}$ and N should scale with $\bar{s}$, which might raise a scalability problem as link capacity increases. Non-uniform quantization of link prices would play a role in mitigating this problem but it requires a priori knowledge on the distribution of link prices, which can hardly be known in advance.

References

[1] S. H. Low and D. E. Lapsley, "Optimization flow control I: basic algorithm and convergence," *IEEE/ACM Trans. Networking*, vol. 7, pp. 861875, Dec. 1999.

[2] S. Athuraliya, S. H. Low, V. H. Li, and Q. Yin, "REM: active queue management," *IEEE Network*, vol. 15, pp. 4853, May 2001.

[3] R. J. Gibbens and F. P. Kelly, "Resource pricing and the evolution of congestion control," *Automatica*, vol. 35, pp. 19691985, Dec. 1999.

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

[5] M. Adler, J.-Y. Cai, J. K. Shapiro, and D. Towsley, "Estimation of congestion price using probabilistic packet marking," in *Proc. IEEE INFOCOM*, vol. 3, Apr. 2003, pp. 20682078.

[6] B. Wydrowski and M. Zukerman, "MaxNet: a congestion control architecture," *IEEE Commun. Lett.*, vol. 6, pp. 512514, Nov. 2002.

[7] A. Charny, D. Clark, and R. Jain, "Congestion control with explicit rate indication," in *Proc. IEEE ICC95*, vol. 3, June 1995, pp. 19541963.

[8] R. W. Thommes and M. J. Coates, "Deterministic packet marking for congestion price estimation," in *Proc. IEEE INFOCOM*, vol. 1, Apr. 2004, pp. 7685.

[9] J. Postel, "Internet protocol," IETF RFC 791, Sept. 1981.

[10] S. Bellovin, "A technique for counting NATted hosts," in *Proc. Internet Measurement Workshop*, Nov. 2002, pp. 267272.