Efficient Probabilistic Packet Marking

Qunfeng Dong*, Micah Adler†, Suman Banerjee*, Kazu Hirata†
*Department of Computer Sciences, University of Wisconsin, Madison, Wisconsin 53706
†Department of Computer Science, University of Massachusetts, Amherst, Massachusetts 01003

***Abstract*— Probabilistic packet marking is a general technique which routers can use to reveal internal network information to end-hosts. Such information is probabilistically set by the routers in headers of regular IP packets on their way to destinations. A number of potential applications have been identified, such as IP traceback, congestion control, robust routing algorithms, dynamic network reconfiguration, and locating Internet bottlenecks, etc. In this paper, we define EPPM, an efficient general probabilistic packet marking scheme with a wide range of potential applications, of which locating Internet bottlenecks and IP traceback are investigated as two representative examples to demonstrate its effectiveness. Our proposed scheme imposes only a single-bit overhead in the IP packet headers. More importantly, it significantly reduces the number of IP packets required to convey the relevant information when compared to the prior best known scheme (almost by two orders of magnitude).**

I. Introduction

Probabilistic packet marking (PPM) was originally suggested by Burch and Cheswick [1] and was carefully designed and implemented by Savage *et. al.* [2] to solve the IP traceback problem which can be stated as follows: given a stream of packets arriving at a receiver, identify the source of these packets and the path they took through the network. However, it is apparent that PPM is a general technique (beyond IP traceback) to communicate internal network information to end-hosts. The basic idea of PPM can be explained using the illustration in Fig. 1. Consider traffic flowing on an Internet path from source S_2 to destination D along the path $S_2 \to 1 \to 3 \to 4 \to D$. A subset of the routers in the path has some local information that needs to be communicated to the destination R. (In the figure all routers in the path have some local information that need to be conveyed.) In order to communicate this information, a PPM scheme sets aside a few bits (PPM bits) in the header of IP packets. In the figure we assume that the number of such available bits is 4. Based on its local information, each router transforms the value of these bits as they pass through. The destination infers the local information at intermediate routers using the value of the PPM bits conveyed using a sequence of such IP packets.

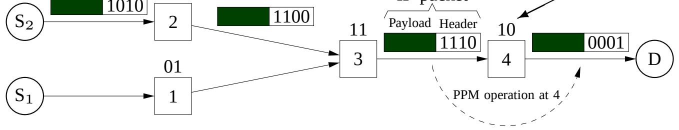


Fig. 1. **An example of probabilistic packet marking.**

PPM has natural applications in solving the IP traceback problem which is a potential countermeasure against distributed denial-of-service (DDoS) attacks. In this problem, the internal network information at each router is the IP address of its (incoming) interface and the goal of a PPM scheme for IP traceback is to convey the entire IP-level path from the source to the destination.

A number of potential applications of PPM have since been identified, which include congestion control [3], robust routing algorithms, dynamic network reconfiguration, and locating Internet bottlenecks. The internal network information in these applications are different from that of IP traceback. For example, in locating bottlenecks and congestion control, the internal network information corresponds to the IP address of the “narrow link” in the network and the level of congestion on this narrow link, respectively.

Different PPM schemes differ from each other in the number of PPM bits allocated, the transformation applied at each intermediate router, the number of IP packets required to convey the internal information to the destination, as well as the decoding algorithm performed at the destination. In particular, the two primary measures of the efficiency of a probabilistic packet marking scheme are the number of PPM bits required in the header of a packet and the number of packets required to convey the internal network information to the destination. *When applied to IP networks, reducing the number of PPM bits is a very significant requirement.* This is because IP is a mature and globally deployed protocol and has very limited number of available header bits that can be used by PPM applications. In fact, the fewer the number of PPM bits, the greater the number of PPM applications that can be deployed simultaneously in the network. Based on this observation, a number of recent papers have focussed on reducing the number of PPM bits required in IP packets, including [2], [4], [5], and [6]. Among these schemes, Adler’s scheme [6] distinguishes itself from others by demonstrating that it is feasible to use as few as a single PPM bit for certain applications. But unfortunately, the number of packets required by that scheme can be prohibitively large. In this paper we propose *Efficient Probabilistic Packet Marking or EPPM*, a generalized packet marking scheme that can be applied to a wide range of PPM applications. EPPM (like Adler’s scheme) can use as few as a single PPM bit, but decreases the number of packets required by almost two orders of magnitude. Unlike prior work in the context of IP traceback by Savage *et. al.* [2], Dean *et. al.* [4], and Song and Perrig [5], EPPM is a general scheme to convey arbitrary network information to end-hosts. Additionally, the number of PPM bits required by both EPPM and Adler's scheme can be flexibly reduced based on the amount of network information to be conveyed. The main advantage of EPPM over Adler's scheme is that using the same number of header bits, EPPM can reduce the number of packets required by almost two orders of magnitude. For example, we show later in the paper that to convey 32-bit information using 5 PPM bits, Adler's scheme requires about 28,000 packets while EPPM requires only 800 packets.

The number of packets required also depends on the nature of specific applications. In general, the number of packets required is higher for *adversarial* applications, e.g., IP traceback used to counter DoS attacks, and lower for *non-adversarial* applications, e.g., congestion control and locating Internet bottlenecks where the network does not attempt to obfuscate any information. In this paper, we demonstrate the effectiveness of EPPM using two representative applications, namely IP traceback to guard against DoS attacks representing the adversarial case and locating Internet bottlenecks representing the non-adversarial case.

Consider an Internet path, with source S, destination, D, and intermediate routers $R_1 \dots R_n$. Depending on the application, each router, R_i, on the network path uses PPM to convey some local information, L_i, to the destination, D. Let us consider our two specific applications in turn:

- IP traceback: The goal is to convey the entire IP network path to the destination, and L_i is the IP address of an interface of router R_i.
- Internet path bottleneck: If the bottleneck link on this path is $R_{j-1} \rightarrow R_j$, then $L_i = \phi, \forall i \neq j$, and L_j is the (incoming interface) IP address of R_j.

Depending on the application, the local information may sometimes be represented more succinctly and efficiently than its usual binary representation, e.g., successive IP addresses on a path can be represented using fewer bits by using a delta encoding (see Section III). In this paper we will make use of such opportunities in specific applications when available. In Fig. 1, the efficient representation of such local information at routers 1, 3, and 4 are the bit strings 10, 11, and 10 respectively. We use the term A_i to indicate the efficient representation of local information L_i and we refer to the concatenation of such efficient representation of local information across the sequence of routers, $A_1|A_2|\dots|A_n$, as the *Information Encoding String or IES.*

Using the above notation, we advocate a layered approach to the design of general probabilistic packet marking schemes, according to which EPPM is designed. This is shown in Fig. 2. The **representation layer** is the topmost layer in this structure. The goal of the representation layer in each hop is to represent the local information as succinctly and efficiently as possible. The concatenation of these bits define the IES. The lowest layer in this framework is the **transmission layer**. The goal of this layer is to efficiently encode the entire IES into the PPM bits of a sequence of IP packets such that the number of such packets required to convey this information is minimized. Finally, there is an optional **security layer** which can be used by sources and routers to prevent any tampering of the IES on the path to the destination.

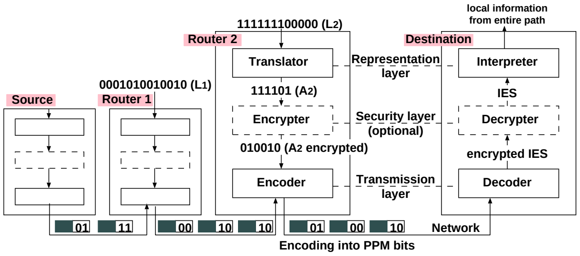


Fig. 2. Layered architecture of EPPM.

This proposed layered approach is quite general in that numerous prior PPM schemes can be easily modeled into this framework including [2], [4]–[6]. We discuss such a construction of prior schemes in Section II. Additionally we find this layered framework to be fairly intuitive. This is because the layers separate different sub-goals of a PPM scheme and each such sub-goal can be viewed independent of other sub-goals by a different layer.

For example, the goal of the representation layer is to minimize the length of IES. Similarly the goal of the transmission layer is to optimize the tradeoff between the number of PPM bits allocated and the number of packets required. Since information representation and information transmission is, thus, decoupled, EPPM can be used to transmit general network information efficiently from network routers to the destination. In fact, EPPM can be used to *efficiently* transmit any IES without requiring that all routers have equal number of contributed bits. In particular not every router needs to participate in EPPM. This is quite unlike most prior work, e.g., Savage *et. al.* [2], Dean *et. al.* [4], Song and Perrig [5], which were designed to transmit very specific IESs (e.g. concatenation of 32-bit IP addresses) for the IP traceback application only.

Key contributions

In this paper, we make the following key contributions.

- We propose a generalized probabilistic packet marking scheme called EPPM which can be used to convey any network information to destination end-hosts. Depending on the application, EPPM can be designed to use a single PPM bit in the IP packet headers. Compared to the the best known scheme for such generalized probabilistic packet marking (Adler [6]) EPPM reduces the number of packets required to convey this information by almost two orders of magnitude, while using the same number of header bits. Unlike other previously proposed schemes that are designed for some specific IES, EPPM can be used to transmit an arbitrarily formatted IES distributed at routers on the path of packets, using much fewer header bits. As is pointed out in [4], this may have a number of potential applications, such as congestion control, robust routing algorithms, dynamic network reconfiguration, as

TABLE I

A QUANTITATIVE COMPARISON OF RELEVANT PPM SCHEMES.

Scheme	Num. of different applications	Min. bits needed	IP traceback (16 hops)	
			Bits used	Packets needed
These schemes were specifically designed for the IP traceback application only.				
Savage *et. al.* [2]	1	16	16	$\sim$ 2,500
Song, Perrig [5]	1	16	16	$\sim$ 1,000
Dean *et. al.* [4]	1	15	15	$\sim$ 10,000
Adler [6]	All PPM apps.	1	7	$\sim$ 2,500
			6	$\sim$ 50,000
			5	$\sim 10^5$
EPPM	All PPM apps.	1	**7**	$\sim$ **400**
			6	$\sim$ 1,200
			5	$\sim$ 9,000
	(see Sec. I)			

well as IP traceback and locating Internet bottlenecks, which are investigated in this paper as two representative examples. We present a comparative summary of EPPM with some relevant PPM schemes in Table I.

- We advocate a layered approach to the design of general PPM schemes, and demonstrate the advantages of such an approach through the design of EPPM.

Roadmap

The rest of this paper is organized as follows. In Section II, we study two representative applications and review related work. EPPM is presented in details in Section III. In Section IV, simulation results are presented to demonstrate the effectiveness of EPPM. Finally, we conclude the paper in Section V.

II. APPLICATIONS AND PRIOR ART

We now discuss two applications of PPM, one representing the non-adverserial case namely locating Internet bottlenecks and the other representing an adverserial case namely IP traceback. For each application, we review the prior best-known mechanisms (both with and without PPM) to implement them.

A. Locating Internet bottlenecks

In locating Internet bottlenecks, the goal is to identify the bottleneck link in a path according to some given performance metric. A number of active probing tools (e.g. Pathchar [7], BFind [8], Pathneck [9]) have been proposed to obtain hop-by-hop performance measurements, which can be used to infer the location of the bottleneck. Unfortunately, it is hard to locate bottlenecks without a comprehensive knowledge of link load on all relevant links. Although these tools provide a viable solution without requiring additional network support, they display a number of shortcomings. Note that the problem of finding the location of the bottleneck is different from the problem of finding available bandwidth of an Internet path. Some examples of such available bandwidth estimation tools include Pathload [10], PathChrip [11], Spruce [12], and those by Hu and Steenkiste [13] and Melander *et. al.* [14].

Inferencing the location of Internet bottlenecks through end-to-end measurements is typically a difficult task and requires some probing overheads. Like any inferencing scheme, its accuracy varies and even the best-known tools exhibit non-negligible measurement errors [9]. However, a PPM-based scheme will suffer from no such shortcoming because it is no longer inferencing the internal network information. Instead in such a scheme, the routers will use the low-overhead PPM mechanism to explicitly convey this information to the destination through the PPM bits of IP packets. By utilizing network support, a PPM-based scheme incurs negligible overheads (some PPM bits), should never be inaccurate and can gather and convey such information in real time. Since the goal of our work is to demonstrate the efficiency of EPPM to convey network internal information to end-hosts, we will examine Internet bottleneck location as an application of EPPM.

B. IP Traceback

We will next focus on the utility of PPM to solve the IP traceback problem: given a stream of packets arriving at a receiver, identify the source of these packets and the path they took through the network.

One approach to solve the IP traceback problem is based on use of packet digests, e.g., work by Snoeren *et. al.* [15], and Li *et. al.* [16]. In this approach, the routers maintain packet digests and destinations can reconstruct paths by iteratively checking neighboring router paths with the same packet. Apart from these non-PPM approaches, researchers have examined some other non-PPM techniques for IP traceback including [1], [17]–[21].

Since PPM is the focus of this paper, we discuss some of the PPM-based IP traceback approaches in more detail. Although previously proposed schemes do not have a notion of layers in their design, it is easy to model such schemes into our proposed layered framework as follows.

1) Savage et. al. *[2]:* In this scheme the local information, L_i, at each router, R_i, is its IP address (or the XOR of consecutive IP addresses that correspond to an edge). The representation layer makes no change to this local information, i.e., $A_i = L_i$. Thus, the IES for this scheme is the concatenation of router IP addresses. There is no security layer. In the transmission layer, each router probabilistically inserts A_i into the packet headers. To reduce the number of PPM bits required, A_i is broken into smaller pieces, which are then sent separately, along with a piece ID used for reconstruction, using 11 PPM bits. In addition, a hop count is maintained using 5 PPM bits, which is reset to 0 when a piece is inserted. This scheme has been further analyzed in [5] and [22].

2) Song and Perrig [5]: This scheme can be viewed in our layered framework as follows. The representation layer transforms the IP address of each router to an 11-bit hash value. The security layer may encrypt the hash value using a Message Authentication Code or a time-release key. The transmission layer communicates the 11-bit string to the destination in the same way as the scheme of Savage *et. al.*, maintaining a hop count using 5 PPM bits as well.

3) Dean et. al. *[4]:* In this scheme, the representation layer is the same as the scheme of Savage *et. al.*. This scheme models the IP traceback problem as a polynomial reconstruction problem, where the local information at intermediate routers is mapped to the coordinates of the polynomial being

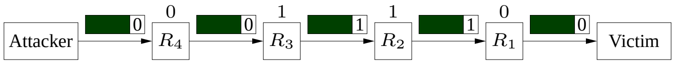


Fig. 3. **An example of the single-bit scheme.**

reconstructed. PPM operation at intermediate routers actually evaluates the polynomial at a number of different points. After collecting sufficiently many evaluations of the polynomial at different points, the destination will be able to reconstruct the polynomial with high probability. There is no security layer.

4) Adler [6]: This scheme does not present any specific representation or security layer. Instead it focussed primarily on the transmission layer. Since EPPM is partially based on this particular scheme, we will present some basic intuition of this work in this section. (Further details can be found in [6].) All of the above schemes and analysis only pay attention to *what* packet headers are received (as opposed to *how many* of each type of header). Adler introduces the more efficient types of transmission schemes that do pay attention to how many of each type of packet header is received.

We first assume that the size of the local information (efficiently represented) is 1, i.e., $|A_i| = 1$ at each router R_i and there are n routers on the path. The size of the IES is thus n. If a router actually has k bits, it can be modeled as k routers individually performing the scheme. Let b represent the number of PPM bits in IP headers. We start by describing the single PPM-bit scheme, i.e., $b = 1$, which is illustrated in Fig. 3. To start with, let us assume that the PPM bit is always initialized to 0. Each router, on receiving a packet, forwards the packet with the PPM bit unchanged and overwrites that bit with A_i with equal probability. When the packet arrives at the victim, the probability that its PPM bit is a 1 is $p = \sum_{i=1}^{n} A_i(\frac{1}{2})^i$, namely the real number obtained by interpreting the n-bit string $A_1A_2\cdots A_n$ as a binary fraction. Then, it is simple to perform the decoding: the receiver simply reads off the bits from the binary fraction representation of the estimate of p.

Adler demonstrates that the single-bit scheme requires to collect $O(2^{2n})$ packets. To decrease the number of packets required, Adler extends the single-bit scheme to the multi-bit scheme, which is illustrated in Fig. 4, where $b = 3$ and $n = 12$. We refer to the set of IES bits that are $k \cdot 2^{b-1} + (i + 1)$ hops away from the victim as *cell* i, where k can be any non-negative integral value. For example, in the figure, cell 2 consists of A_3, A_7, and A_{11}. In the multi-bit scheme, one of the PPM bits (e.g. the rightmost bit of each packet in Fig. 4) is referred to as the *marking bit*, and the other PPM bits are used as a $(b-1)$-bit *counter*. On receiving a packet, a router increments the counter in the packet. If the counter does not overflow, the marking bit is unchanged. Otherwise, the single-bit scheme is performed on the marking bit, as if the marking bit is the single PPM bit in the single-bit scheme and the path is composed of all routers in the cell. When decoding, for each individual cell, the victim independently performs the same decoding procedure as is used for the single-bit scheme.

In the current single-bit scheme, the victim is not able to differentiate between the case where $A_n = 1$ and the PPM bit is initialized to 0, and the case where $A_n = 0$ and the PPM bit is initialized to 1. To uniquely decode the IES, Adler modifies the single-bit scheme such that each router still forwards the bit that it holds with probability $\frac{1}{2}$, but now forwards the PPM bit that it receives with probability $r = \frac{1}{2} - \epsilon$, and 0 with probability ϵ, for some constant $0 < \epsilon < \frac{1}{2}$. With a sufficiently good estimate of p, the IES can be uniquely decoded. In the multi-bit scheme, the victim must receive enough packets for every cell. To ensure that, Adler modifies the multi-bit scheme such that each router with probability ρ performs a *reset* operation: it performs the single-bit scheme assuming that the incoming marking bit is a 0 and sets counter to 0.

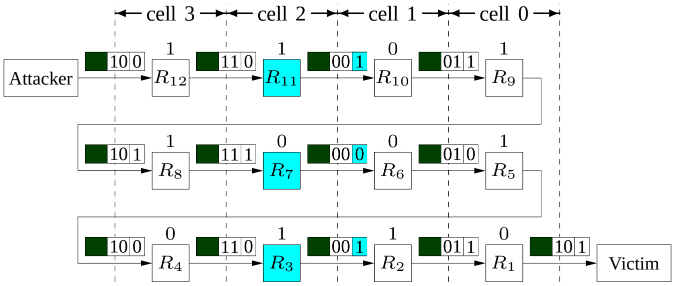


Fig. 4. **An example of the multi-bit scheme.**

TABLE II
KEY NOTATIONS

ρ	The probability of routers performing a reset operation
r	The probability of forwarding the received marking bit when performing the single-bit scheme
P_k	The set of packets that arrive at the victim with their counter set to k
$z(n,k)$	The number of bits in cell k of an n-bit IES
A_i^k	The ith bit in cell k
α_i^k	The probability that a packet in P_k is last reset by some router that is at least as far as the router holding A_i^k, given that it is actually reset
q_k^n	The fraction of packets in P_k that is not reset by any router between the victim and the attacker
p_k	The probability that a packet chosen uniformly at random from P_k has its marking bit set to 1 when it arrives at the victim

Using the notations in Table II, we next give a brief description of the decoding process of the complete multi-bit scheme. It can be shown that

$$p_k = \sum_{j=1}^{z(n,k)} A_i^k (q_k^n + (1 - q_k^n)\alpha_j^k)\frac{r^{j-1}}{2}$$

. If the victim actually had available to it the values q_k^n, then the decoding procedure would be a natural generalization of the single-bit case. To estimate q_k^n, the victim compares the number of packets it receives with counter k to the expected number of such packets it would receive if all such packets were reset; the additional number of packets are assumed to arrive without having been reset in between. In particular, let v_k^n be the probability that a packet is reset by some router on the path and is in P_k. Let T be the total number of packets.

TABLE III

SOME ADDITIONAL NOTATIONS OF EPPM

p_i^k	The minimum probability that a packet in P_k is received by the victim with its marking bit being a 1, given that $A_i^k = 1$ and that $\forall j$, $A_j^k = 0$
$\bar{p}_i^k$	The maximum probability that a packet in P_k is received by the victim with its marking bit being a 1, given that $A_i^k = 0$ and that $\forall j$, $A_j^k = 0$

TABLE IV

FAST DECODING SCHEME (FD) AT THE VICTIM

```
Input: PPM bits in collected packets
Output: decoded IES bits ($A_i^k$ denotes the $i$th IES bit in cell $k$)
for $k = 1$ to $2^{b-1}$
    Estimate $\bar{p}_k$
    Estimate $\bar{q}_k^n = \frac{|P_k| - v_k^n \cdot T}{|P_k|}$
    for $i = 1$ to $z(n,k)$
        $p_i^k = \frac{q_k^n + (1-q_k^n)\alpha_i^k}{2}$
        $\bar{p}_i^k = q_k^n r^{z(n,k)} + \sum_{j=i+1}^{z(n,k)} A_j^k (q_k^n + (1-q_k^n)\alpha_j^k) \frac{r^{j-1}}{2}$
        if $\bar{p}_k \geq \frac{p_i^k + \bar{p}_i^k}{2}$
            then $A_i^k = 1$
            else $A_i^k = 0$
        $\bar{p}_k = \bar{p}_k - A_i p_i^k$
    endfor
endfor
```

TABLE V

ADVANCED RESET SCHEME (AR) AT ROUTERS

```
On receiving a packet, the router:
/* counter corresponds to the $b-1$ PPM bits in the packet header */
With probability $\rho$
    counter $\leftarrow 0$
    marking bit $\leftarrow$ local IES bit
counter++
if (counter mod $2^{b-1}$) $== 0$
    With probability $\frac{1}{2}$: marking bit $\leftarrow$ local IES bit
    With probability $\frac{1}{2} - r$: marking bit $\leftarrow 0$
endif
```

The victim estimates q_k^n using the value $\bar{q}_k^n = \frac{|P_k| - v_k^n \cdot T}{|P_k|}$.

III. EPPM

In this section, we present our EPPM scheme for adversarial applications and non-adversarial applications, using IP traceback and locating Internet bottlenecks as two representative examples, respectively.

A. Adversarial case (e.g. IP Traceback)

Using IP traceback as an example, we here present our general framework of EPPM for adversarial applications, where the sender tries to obfuscate the PPM scheme by deliberately setting the initial values of PPM bits.

1) Transmission layer: We present the transmission technique of EPPM as a sequence of variations to Adler's technique. We start by proposing an intuitively simpler and more efficient decoding algorithm. Two other variations are proposed based on our analysis of this decoding algorithm. We then propose to use two marking bits instead of one marking bit, which can effectively decrease the number of packets required by enabling different cells to share packets. Finally, we propose an optional improvement that can significantly reduce the number of packets required, but would not allow the length of IES to be larger than 2^{b-1}.

Fast decoding scheme (FD) at the victim: We first propose the *fast decoding scheme (FD)*, an efficient and intuitively simpler decoding algorithm that converges much fast than the decoding algorithm in [6]. We start by describing the decoding procedure for the most significant bit of cell k, namely A_1^k.

If $A_1^k = 1$, the minimum probability that a packet in P_k is received by the victim with its marking bit being a 1 is given by

$$p_1^k = \frac{q_k^n + (1-q_k^n)\alpha_1^k}{2}.$$

If $A_1^k = 0$, the maximum probability that a packet in P_k is received by the victim with its marking bit being a 1 is given by

$$\bar{p}_1^k = q_k^n r^{z(n,k)} + \sum_{j=2}^{z(n,k)} A_j^k (q_k^n + (1-q_k^n)\alpha_j^k) \frac{r^{j-1}}{2}.$$

Let $\bar{p}_k$ denote the fraction of packets in P_k whose marking bit is received by the victim as a 1. We determine that $A_1^k = 1$ if $\bar{p}_k \geq \frac{p_1^k + \bar{p}_1^k}{2}$; Otherwise, we determine that $A_1^k = 0$. Then, we set $\bar{p}_k = \bar{p}_k - A_1 p_1^k$. We decode A_i^k for $i > 1$ by repeating the same decoding procedure. A pseudo code description of FD at the victim is given in Table IV. In our simulated scenarios, FD decreases the number of packets required by 41% to 47%.

From the description of FD, it is clear that larger gaps between p_i^k and $\bar{p}_i^k$ require less precision and hence fewer packets in P_k to be collected to estimate p_k. We next propose two effective schemes to broaden the gap between p_i^k and $\bar{p}_i^k$.

Advanced reset scheme (AR) at routers: The next and the most significant variation we introduce is the *advanced reset scheme (AR)*. In particular, when a packet is reset the router always forwards its local IES bit instead of forwarding it with a probability of $\frac{1}{2}$. A pseudo code description of the encoding algorithm with AR at routers is given in Table V.

Compared to Adler's scheme, this AR scheme increases p_i^k by an amount we denote by Δp_i^k, which can be calculated by

$$\Delta p_i^k = (1-q_k^n)(\frac{\rho(1-\rho)^{(i-1)d+k}}{v_k^n})\frac{r^{i-1}}{2}.$$

Similarly, the AR scheme increases $\bar{p}_j^k$ by an amount we denote by $\Delta \bar{p}_i^k$, which can be calculated by

$$\Delta \bar{p}_i^k = \sum_{j=i+1}^{z(n,k)} (1-q_k^n)(\frac{\rho(1-\rho)^{(j-1)d+k}}{v_k^n})\frac{r^{j-1}}{2}.$$

It can be shown through simple reasoning that $\Delta \bar{p}_i^k < \Delta p_i^k$. Consequently, the gap between p_j^k and $\bar{p}_j^k$ is enlarged. In our simulated scenarios, this AR scheme decreases the number of packets required by a factor of 5 to 6.

TABLE VI

ADVANCED SINGLE-BIT SCHEME (AS) AT ROUTERS

```
On receiving a packet, the router:
/* counter corresponds to the $b-1$ PPM bits in the packet header */
With probability $\rho$
    marking bit ← local IES bit
    counter ← 0
counter++
if (counter mod $2^{b-1}$) == 0
    With probability $1-r$: marking bit ← local IES bit
endif
```

Advanced single-bit scheme (AS) at routers: To further broaden the gap between p_j^k and $\bar{p}_j^k$, we introduce the *advanced single-bit scheme (AS)*: whenever the single-bit scheme is applied, the router forwards its local IES bit with probability $1-r$ instead of $\frac{1}{2}$, and forwards the incoming PPM bit with probability r. A pseudo code description of of the encoding algorithm with AS and AR at routers is given in Table V.

It is clear that both Δp_i^k and $\Delta \bar{p}_i^k$ are increased by a factor of $2(1-r) > 1$. Consequently, $\Delta p_i^k - \Delta \bar{p}_i^k$ is increased by the same factor, and the gap between p_i^k and $\bar{p}_i^k$ is proportionally enlarged. In our simulated scenarios, AS decreases the number of packets required by 16% to 18%.

Two marking bits (TMB) at routers: The last variation we introduce to improve the performance of EPPM is to use *two marking bits (TMB)*. Namely, we will use two of the given b PPM bits as marking bits and use the other $b-2$ bits as a counter. A router partitions its local IES bits by their indices into two sets, the *odd set* and the *even set*, and uses one marking bit on each set. In effect, the IES is partitioned into two strings: the *odd IES* and the *even IES*, each being encoded and decoded using one of the two marking bits. To use two marking bits, each router should have an even number of IES bits, or if not, it simply pads its own block of the IES. We can take this into consideration in the design of an efficient IES.

Here, both the odd IES and the even IES have 2^{b-2} cells, while the original IES has 2^{b-1} cells. On the other hand, every packet is utilized by both of them, which means that the number of packets required to decode both of them is equal to the number of packets required to decode an $\frac{n}{2}$-bit IES using one marking bit. Thus, by using two marking bits, we transform the case (b, n) into the case $(b-1, \frac{n}{2})$. Simulation results demonstrate that using two marking bits decreases the number of packets required by a factor of 3 to 4. The factor is greater than 2 because we have to wait for the slowest cell to converge. The slowest cell of the $(b-1, \frac{n}{2})$ case is intuitively faster than the slowest cell of the (b, n) case, which in combination with the halving of the number of cells result in a speedup factor that is great than 2.

Of course, this technique can be generalized to use $m > 2$ marking bits. Compared to two marking bits, the number of cells is decreased by a factor of 2^{m-2}, while the length of each cell is increased by a factor of $\frac{2^{m-1}}{m}$. Given that the number of packets required to correctly decode a cell is exponential in its length, large values of m are intuitively undesirable. Empirical evaluations demonstrate that increasing the length of a cell by 1 bit will increase the number of packets required by a factor of 3 to 10. Therefore, using more than two marking bits will not help for almost all reasonably large values of n and reasonably small values of b.

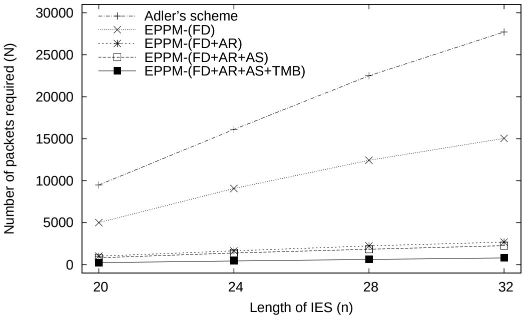


Fig. 5. Comparison of EPPM with Adler's scheme (b=5).

TABLE VII

OPTIONAL IMPROVEMENT AT ROUTERS

```
On receiving a packet, the router:
/* counter corresponds to the $b-1$ PPM bits in the packet header */
With probability $\rho$
    marking bit ← local IES bit
    counter ← 0
counter++
if (counter mod $2^{b-1}$) == 0
    marking bit ← local IES bit
endif
```

We evaluate the effectiveness of our proposed variations using simulation results presented in Fig. 5, where $b = 5$. Combining all the proposed variations, EPPM decreases the number of packets required by almost two orders of magnitude for a 32-bit IES. In our simulated scenarios, even larger performance improvements can be achieved for longer IESs.

Optional improvement at routers: It turns out that we can dramatically improve the performance of EPPM by going further than the advanced single-bit scheme: whenever the single-bit scheme is applied, we always forward the local bit instead of with a probability of $1-r$. A pseudo code description of this optional improvement at routers is given in Table VII.

In this case, all packets that are received with the same counter value k will have a marking bit that is equal to the kth IES bit. As soon as the victim has received at least one packet from each cell, it has enough information to reconstruct the entire IES. To put off this event, the attacker should choose to set the counter of all packets to some number between 1 and $2^{b-1} - n$ if $n < 2^{b-1}$, or to some fixed number if $n = 2^{b-1}$. With this technique, we observed the tradeoffs in Fig. 6.

There is, however, a cost to this variation. In the original scheme, there is no theoretical limit on n: any value of n can be decoded, given enough packets. With this variation, n is now limited to be no larger than 2^{b-1}. Because each cell can contain only one bit, since the bit closest to the victim will always overwrite the recorded value of preceding bits in the same cell. We point out that if $n \leq 2^{b-1}$, there is only

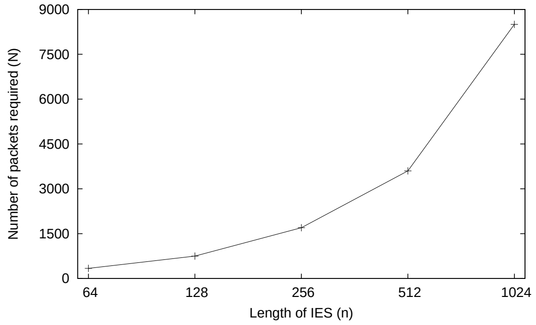


Fig. 6. Tradeoffs of EPPM with the optional improvement.

one IES bit per cell, and thus increasing b does not help. In fact, it has detrimental effect, since increasing b increases the number of cells, and thus we need more packets to make each cell have the same number of packets as before. Therefore, $b = \lceil \log n \rceil + 1$ is the optimal setting.

2) Representation layer: Recall that representation layer of EPPM only needs to minimize the length of IES. We demonstrate the advantage of this layered architecture of EPPM by introducing two novel and efficient IESs.

***Delta encoding**:* We first introduce *delta encoding* based on the observation that it is quite common for successive routers in a path to have very similar IP addresses. In delta encoding, each router encodes its identity as the XOR difference between the IP address of its own incoming interface and the IP address of its successor's incoming interface (referred to as δ). The IES is simply the concatenation of these individual encodings. This encoding is extremely lightweight: a router only needs to know its own IP address and the IP address of its successor. As a router's task is to forward packets to this successor, a router does have this information available.

In delta encoding, each router computes δ and transmits the concatenation of $h(len(\delta))$ and $chop(\delta)$, in this order, where

- $len(\delta)$ evaluates to the position of the highest 1 in the binary representation of δ. For router A in Fig. 7, $\delta = (192.5.89.9)$ XOR $(192.5.89.246) = (11111111)$ and hence $len(\delta) = 8$.
- $chop(\delta)$ evaluates to the binary representation of δ without the leading 1. For router A in Fig. 7, $chop(\delta) = chop(11111111) = 1111111$.
- $h(len(\delta))$ Huffman-encodes $len(\delta)$ based on a predetermined distribution of $len(\delta)$, because Huffman-encoding is known as the most compact encoding scheme.

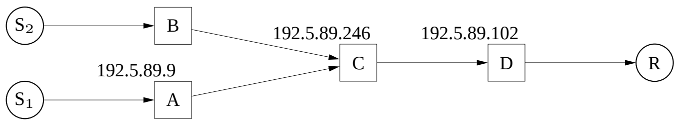


Fig. 7. **A network where IP addresses refer to incoming interfaces.**

The decoding on the victim side is straightforward. The victim first decodes $len(\delta)$, according to the Huffman code. Using that, $chop(\delta)$ can be retrieved. By repeating this procedure, the victim can discover the entire path leading up to the router immediately after the attacker.

We used a large collection of traceroute data [23] to evaluate the effectiveness of delta coding. With Huffman coding based on the observed distribution, it takes 4.30 bits on average to encode $len(\delta)$. Using the Huffman coding to encode all of the paths in [23], delta encoding yields an average value of $n = 314$ with a standard deviation of 72, a significant improvement over the simple concatenation of IP addresses. With $n = 409$, we cover 90% of the paths.

A problem is that if any of the routers along the path is not a traceback router, the victim can not reconstruct the path past that point, since at that point, the XOR information becomes meaningless. One solution to this problem is to have routers exchange some more information to determine if their successor is a traceback router. If it is not, the router can simply transmit the IP address of its own incoming interface instead of a δ. In this way, our delta encoding allows for incremental deployment. Although IP address is typically longer than δ, the overall IES length may be decreased since there are fewer router IP addresses to be transmitted. This is analyzed further later in this section.

***Topology-based encoding**:* To make delta encoding incrementally deployable, routers need to exchange some additional information. This idea leads to a still more efficient IES that uses topological information. In particular, a router assigns each of its neighbors a unique *neighbor ID*. The IES will be the concatenation of neighbor IDs of routers that a packet travels through. If router R_1 precedes router R_2 in the path from the attacker to the victim, the IES block held by router R_1 is the neighbor ID that is assigned by R_2. To make this topology-based encoding incrementally deployable, one special neighbor ID (e.g. 0) should be reserved for non-traceback routers. If router R_2 is not a traceback router, the IES block held by router R_1 is the special neighbor ID 0 followed by its own IP address. Thus, the IES is the concatenation of neighbor IDs and IP addresses. For example, assume that router C in Fig. 7 assigns neighbor ID 1, 2, 3 to router A, B, D, respectively. If router C is a traceback router, the IES block held by router A is its neighbor ID assigned by router C, namely 1. Otherwise, the IES block held by router A is 0 followed by its own IP address, 192.5.89.9.

We need $\lceil \log(\delta + 1) \rceil \leq \delta$ bits to encode a neighbor ID that is assigned by a router of degree δ. If we assume that a router in the path is a randomly chosen router of the Internet topology graph, then an upper bound on the expected number of bits per router is $\sum_i Pr[degree = i] \cdot i$. This is actually the average degree in the router-level Internet topology, which has been recently measured to be 2.81 [24].

To estimate the length of IES, we assume that each router is a traceback router independently with probability p. The expected number of IES bits required to encode a router is $p \cdot [2.81p + (32 + 2.81)(1 - p)]$, which is maximal if $p = 0.544$. With full deployment (i.e. $p = 1$), the average value of 16 hops [25] yields $n = 44.96$. A similar analysis applies to the case of delta encoding.

Compared to delta encoding, topology-based encoding requires the victim to have topological information to reconstruct paths. This topological information can be accessed by the victim using two different techniques: *local reconstruction* at the victim, using a map of upstream routers (as was utilized by [5]) or *interactive reconstruction* without such a map. Interaction reconstruction can be implemented in a way that is similar to the reconstruction process in digest-based schemes.

***Further improvements**:* The IESs we have introduced can be further improved by deploying IP traceback on a selective collection of routers. For example, we can have all stub routers perform packet marking, as well as any router that receives packets from a different AS. Thus, if an average Internet path of 16 hops [25] passes through 3 stub routers each at the start and end of the path, and goes through a series of 10 hops over 3 different ASes in between, the number of traceback routers would be 9, a significant improvement over the 16 required if every router were participating. Consequently, the IES becomes much shorter.

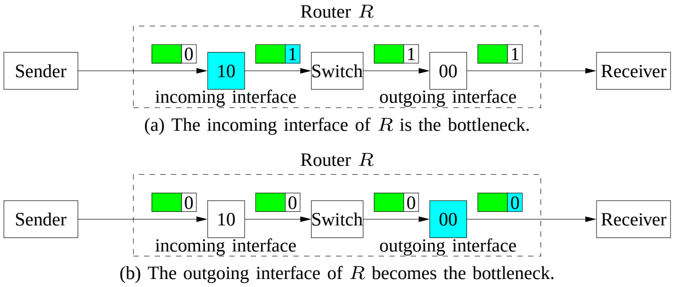
B. Non-adversarial case (e.g. locating Internet bottlenecks)

In this section, we propose a general framework of EPPM for non-adversarial applications, using locating Internet bottlenecks as an example. In non-adversarial applications, the sender helps improve the performance of the PPM scheme by carefully initializing PPM bits. We use a qualitative definition where the network interface with the "worst performance metric" is referred to as the *bottleneck* of the path. There are a variety of possible performance metrics (e.g. available bandwidth, delay, packet loss rate, etc), depending on the user's concern. The user is free to choose any metric of interest, as long as it can be effectively measured by routers.

1) Representation layer: In our scheme, allocated header bits are partitioned into two fields. One field is used to explicitly record the performance metric, and the number of bits in this field directly depends on the precision desired by the application. The other field is used to communicate the identity of the bottleneck interface to the receiver. On receiving a packet, each router checks the performance metric carried in the packet. If its own performance metric is worse, it marks its identity and performance metric into the header bits.

The identity of the bottleneck may be described in a number of different ways. The most straightforward IES is its IP address. However, if the ordered list of router interface IP addresses along the path can be obtained (e.g. by the `traceroute` utility), shorter IESs can be defined. For example, we may compute a 10-bit or 12-bit hash value of the IP address. Since the hop counts of Internet paths almost never exceed 32 [25], a 10-bit or 12-bit hashing scheme should suffice to avoid hashing conflicts with high probability.

2) Transmission layer: In the case of locating Internet bottlenecks, if a router needs to transmit its IP address, it simulates the process of $n = 32$ routers individually performing the multi-bit scheme. In such non-adversarial cases, the sender is presumed to be a cooperative host. The performance of EPPM is significantly better in such scenarios since some of the challenging decoding problems can be eliminated. In particular, the sender can always initialize the marking bit to 0, so that the destination host does not have to guess the initial value of the marking bit and there is no need to modify the single-bit scheme. Moreover, the sender can always initialize the counter to some random value between 0 and $2^{b-1} - 1$, so that packets are evenly distributed among cells. The insight is that, in order to correctly decode the whole IES, we have to wait for the slowest cell to converge, and an even distribution of packets among cells helps the slowest cell. Thus, routers do not have to perform the reset operation. This eliminates a number of complicated problems.



(a) The incoming interface of R is the bottleneck.

(b) The outgoing interface of R becomes the bottleneck.

Fig. 8. **The sender and the receiver are connected by a single intermediate router R. The identity of R's incoming interface hashes to "10" and the identity of R's outgoing interface hashes to "00".**

3) Discussion: A potential problem of our scheme is that its worst case performance is not predictable. This is because the bottleneck may change before the receiver has collected enough packets encoding information about that bottleneck. Consequently, the information encoded in the header bits will be distorted, and the receiver may not be able to correctly decode the identity of the bottleneck. For a simple example, consider the path in Fig. 8 where $b = 1$. Initially, the outgoing interface of R is the bottleneck, and 500 packets are marked. The probability that the header bit of these packet is received as a 1 is $(0.00)_2 = 0$. Then the performance metric at R's incoming interface degrades and it becomes the bottleneck. 500 packets are marked after this change. The probability that the header bit of these packet is received as a 1 is $(0.10)_2 = \frac{1}{2}$. Without loss of generality, assume that 250 of the 500 header bits are received as 1s and the other 250 header bits are received as 0s. Now the receiver has 250 header bits received as 1s and 750 header bits received as 0s. Consequently, the decoded IES turns out to be 01, which does not refer to either of the interfaces.

To handle such a mixture of packet headers carrying information of different bottlenecks, we look at a window of most recently received packet for each path and conduct decoding within this window. Depending on the number of PPM bits allocated, we can determine an appropriate window size, which is actually the number of packets required to correctly decode the bottleneck information with high probability. We demonstrate with simulation results in Section IV that our schemes provide an effective solution to locating Internet bottlenecks in reasonably dynamic network environments.

IV. Performance evaluation

In this section, we evaluate the performance of EPPM for locating Internet bottlenecks and IP traceback. In each of our experiments, we generate 5000 random paths (referred to as *trials*). For each trial, we simulate packets being encoded with that trial. We report the number of packets required to decode 90% of the trials, which is denoted by N. Our simulation proceeds in a round-by-round fashion. During each round, some number of packets are simulated and collected. After each round, we try to decode based on the packets we have collected so far. Once we have correctly decoded for reasonably many consecutive rounds, we determine that the decoding procedure has correctly converged.

A. IP traceback

We first discuss how EPPM parameters should be chosen, and then discuss the performance of EPPM for IP traceback using these parameters.

Parameter settings: In IP traceback, the defenders (i.e., routers and the victim) and the attacker are playing a *min-max game*. Given the values of the EPPM parameters, ρ and r (see Table II), we assume that the attacker can deliberately determine the initial settings of the PPM bits to maximize the number of packets required. This would measure the worst case performance of the scheme. Therefore, for the routers, our objective is to determine the optimal values of ρ and r such that the worst case number of packets required is minimized. For the decoder, the most difficult part is correctly decoding the IES bit that is the farthest from the victim, namely A_n, where n is the length of the IES. To maximize the number of packets that carry information about A_n, we should maximize $p_{last} = \rho(1-\rho)^{n-1}$, the probability that a packet is last reset by router R_n. Therefore, the optimal value of ρ is $\frac{1}{n}$.

It is not so obvious what is the optimal value of r and what are the optimal initial settings of the PPM bits. In our simulation, for each setting of (b, n, ρ), where b is the number of PPM bits allocated, we conduct exhaustive search for the optimum of the min-max game. In particular, for each setting of (b, n, ρ, r), the marking bit has three possible initial settings: always 0, always 1, or random, and the counter has $2^{b-1}+1$ possible initially settings: numbers between 0 and $2^{b-1}-1$, or random. All the $3 \cdot (2^{b-1}+1)$ combinations are simulated, and we report the resulting number of packets required of the (b, n, ρ, r) setting such that the number of packets required by the worst case of its $3 \cdot (2^{b-1}+1)$ combinations is minimum.

Performance evaluation: We next evaluate the performance of EPPM for IP traceback and compare with previous schemes [2], [5]. For that, we apply the transmission layer of EPPM to the IESs of these previous schemes as well as our proposed IESs. For IESs other than delta encoding, we assume full deployment on the average Internet path length of 16 hops [25] and the corresponding IES length. For delta encoding, we use the measured average IES length of $n = 314$. We present the tradeoffs between b and N for all IESs in Fig. 9.

For the IES of Savage *et. al.* [2], the transmission layer of EPPM only needs 10 PPM bits while the scheme of Savage *et. al.* uses 16 PPM bits, both using a few thousand packets. For the IES of Song and Perrig [5], the transmission layer of EPPM only needs 8 PPM bits while the scheme of Song and Perrig uses 16 PPM bits, both using a few thousand packets.

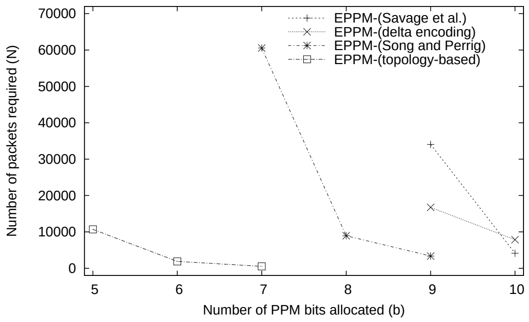


Fig. 9. Tradeoffs of EPPM for IP traceback.

Our proposed IESs are also more efficient than those of previous schemes [2], [5]. If we do not have an upstream map that is required by the scheme of Song and Perrig, both the IES of Savage *et. al.* and our proposed delta encoding can be used. Using the same transmission layer, namely that of EPPM, our proposed delta encoding has been shown to produce shorter IESs, and Fig. 9 demonstrates that fewer packets are required by delta encoding, using the same number of PPM bits. If we do have such an upstream map, then both the IES of Song and Perrig and our proposed topology-based encoding can be used. Both using a few thousand packets, topology-based encoding only needs 5 PPM bits while the IES of Song and Perrig requires 8 PPM bits. If we use just 7 PPM bits for topology-based encoding, EPPM only needs about 400 packets, while all previous schemes [2], [4], [5] uniformly require a few thousand packets.

Therefore, our conclusion is that EPPM is more efficient than previous schemes [2], [4], [5] in terms of both the number of PPM bits and the number of packets required. In addition, as we have pointed out in Section III, our transmission layer offers the unique advantage that the IES length is not limited by the number of PPM bits, while previous schemes not based on the technique of [6] uniformly rely on a hop count field of ℓ bits which limits the number of routers in paths to be no larger than 2^{ℓ}. This also limits how small a value of b can be realized with these previous schemes. As we have demonstrated in Fig. 6, the number of packets required can be significantly further decreased, if we limit n to be no larger than 2^{b-1}.

B. Locating Internet bottlenecks

The performance of EPPM is significantly better in non-adversarial cases such as locating Internet bottlenecks, since some of the complicated encoding and decoding problems are eliminated. We present the tradeoffs of EPPM in Fig. 10. We can see that less than 3000 packets suffice to transmit the IP address of the bottleneck link to the receiver using 4 header bits (plus the header bits used to record the queuing delay), and a few hundred packets suffice to transmit it using 5 header bits. Moreover, just a few hundred packets suffice to transmit

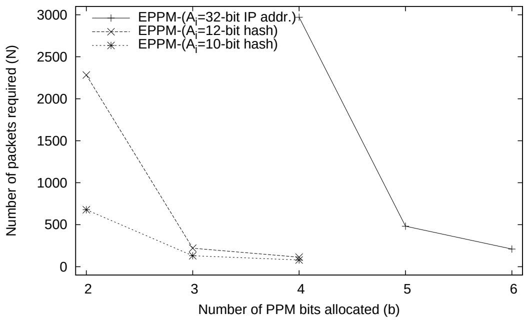


Fig. 10. Tradeoffs of EPPM for locating Internet bottlenecks. A_i denotes the local IES string that the router needs to communicate to the destination.

a 10-bit hash of the IP address using only 2 header bits. Dividing the values of N given above by the expected number of packets received per second, we expect the performance of EPPM to be good enough for reasonably dynamic network environments. For example, assume that 50 packets can be received per second. If we transmit the IP address using 4 header bits, a window size of 3000 suffices to correctly decode with high probability, as long as the bottleneck link does not change within a minute. If the 10-bit (12-bit) hash encoding is used, we can use only 2 header bits, as long as the bottleneck does not change within 15 seconds (45 seconds).

V. Conclusion

In this paper, we advocate a layered approach to the design of general probabilistic packet marking schemes and demonstrate its strengths with our proposed EPPM, a general scheme that has a wide range of potential applications such as IP traceback, congestion control, robust routing algorithms, dynamic network reconfiguration, and locating Internet bottlenecks. This should be contrasted with some of the prior PPM schemes that were designed specifically for IP traceback applications and require a fixed number (15 or 16) of PPM bits in IP headers. A main advantage of a a general probabilistic packet marking scheme like EPPM is that it can operate with even a single PPM bit and thus allow for more simultaneous PPM applications to be implemented. To be efficient in the number of packets when using EPPM with a single PPM bit (say, the number of packets is about a few thousand packets), the applicable IES length should be up to 6 bits long. In the representative applications we study in this paper the IESs are at least 10-bit long. In such cases using EPPM we need 2 or more PPM bits to limit the number of packets to the same bounds. Compared with another general probabilistic packet marking scheme that is proposed in [6], EPPM decreases the number of packets required by *almost two orders of magnitude*, using the same number of header bits.

In this paper we also present the first precise empirical study of the tradeoffs achieved by such techniques. We also use some analytical results as well as an extensive set of simulations to determine the optimal settings of a number of parameters.

References

[1] H. Burch and B. Cheswick, “Tracing anonymous packets to their approximate source,” in *Usenix LISA*, 2000.

[2] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, “Practical network support for ip traceback,” in *ACM SIGCOMM*, 2000.

[3] M. Adler, J.-Y. Cai, J. Shapiro, and D. Towsley, “Estimation of congestion price using probabilistic packet marking,” in *IEEE INFOCOM*, 2003.

[4] D. Dean, M. Franklin, and A. Stubblefield, “An algebraic approach to ip traceback,” in *Network and Distributed System Security Symposium*, 2001.

[5] D. X. Song and A. Perrig, “Advanced and authenticated marking schemes for ip traceback,” in *IEEE INFOCOM*, 2001.

[6] M. Adler, “Tradeoffs in probabilistic packet marking for ip traceback,” in *ACM STOC*, 2002.

[7] V. Jacobson, *Pathchar – a tool to infer charactertistics of internet paths*, 1997, presented as April 97 MSRI talk.

[8] A. Akella, S. Seshan, and A. Shaikh, “An empirical evaluation of wide-area internet bottlenecks,” in *ACM IMC*, 2003.

[9] N. Hu, L. E. Li, Z. M. Mao, P. Steenkiste, and J. Wang, “Locating internet bottlenecks: algorithms, measurements, and implications,” in *ACM SIGCOMM*, 2004.

[10] M. Jain and C. Dovrolis, “End-to-end available bandwidth: Measurement methodology, dynamics, and relation with tcp throughput,” in *ACM SIGCOMM*, 2002.

[11] V. J. Ribeiro, R. H. Riedi, R. G. Baraniuk, J. Navratil, and L. Cottrell, “pathchirp: Efficient available bandwidth estimation for network paths,” in *PAM*, April 2003.

[12] J. Strauss, D. Katabi, and F. Kaashoek, “A measurement study of available bandwidth estimation tools,” in *ACM IMC*, 2003.

[13] N. Hu and P. Steenkiste, “Evaluation and characterization of available bandwidth probing techniques,” *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 6, pp. 879–894, 2003.

[14] B. Melander, M. Bjorkman, and P. Gunningberg, “A new end-to-end probing and analysis methodfor estimating bandwidth bottlenecks,” in *IEEE Globecom*, 2000.

[15] A. C. Snoeren, C. Partridge, L. A. Sanchez, C. E. Jones, F. Tchakountio, S. T. Kent, and W. T. Strayer, “Hash-based ip traceback,” in *ACM SIGCOMM*, August 2001.

[16] J. Li, M. Sung, J. J. Xu, and L. E. Li, “Large-scale ip traceback in high-speed internet: Practical techniques and theoretical foundation,” in *IEEE SSP*, 2004.

[17] S. M. Bellovin, *ICMP Traceback Messages*, March 2000, internet Draft: draft-bellovin-itrace-00.txt.

[18] K. Park and H. Lee, “On the effectiveness of route-based packet filtering for distributed dos attack prevention in power-law internets,” in *ACM SIGCOMM*, 2001.

[19] T. Doeppner, P. Klein, and A. Koyfman, “Using router stamping to identify the source of ip packets,” in *ACM CCS*, 2000.

[20] P. Ferguson and D. Senie, *RFC 2267: Network Ingress Filetring: Defeating Denial of Service Attacks which employ IP Source Address Spoofing*, The Internet Society, 1998.

[21] S. Lee and C. Shields, “Tracing the source of network attack: A technical, legal and societal problem,” in *IEEE Workshop on Information Assurance and Security*, 2001.

[22] K. Park and H. Lee, “On the effectiveness of probabilistic packet marking for ip traceback under denial of service attack,” in *IEEE INFOCOM*, 2001.

[23] K. M. Hanna, http://signl.cs.umass.edu/logs/.

[24] S. J. S. S. Hongsuda Tangmunarunkit, Ramesh Govindan and W. Willinger, “Network topologies, power laws, and hierarchy,” Computer Science Department, University of Southern California, Tech. Rep. 01-746, ftp://ftp.usc.edu/pub/csinfo/tech-reports/papers/01-746.pdf.

[25] W. Theilmann and K. Rothermel, “Dynamic distance maps of the internet,” in *IEEE INFOCOM*, 2000.