Markov Chain Modelling of the Probabilistic Packet Marking Algorithm [*]

Tsz-Yeung Wong, John Chi-Shing Lui, and Man-Hon Wong
(Corresponding author: T. Y. Wong)

Department of Computer Science and Engineering, The Chinese University of Hong Kong
Room 1028, 10th floor, Ho Sin Hang Engineering Building, Shatin, New Territories, Hong Kong
(Email: {tywong, cslui, mhwong}@cse.cuhk.edu.hk)

(Received Oct. 25, 2005; revised and accepted Apr. 7, 2006)

Abstract

In this work, we show that the current termination condition of the Probabilistic Packet Marking (PPM) algorithm is not correct for general networks, and this implies the estimation of expected number of marked packets is not accurate. As a result, this may lead to an incomplete attack graph construction. To remedy this problem, we propose a Markov chain modelling of the PPM algorithm. By applying the fundamental matrix theory, one can result in an accurate estimation of the expected number of marked packets. Our simulation results show that the Markov chain modelling technique is effective in calculating the expected number of marked packets.

Keywords: Denial-of-service attack, IP traceback, Markov chain model, PPM algorithm

1 Introduction

The denial-of-service (DoS) attack is a pressing problem for recent years [2]. DoS defense research blossomed into one of the main stream in the computer security fields. Various techniques such as the pushback message [7, 9], ICMP traceback [15], and the packet filtering techniques [5, 10, 17] are the results from this active field of research.

The Probabilistic Packet Marking (PPM) algorithm by Savage [12] et al. had attracted the most attentions in contributing the idea of IP traceback [1, 4, 8, 11, 13, 14]. The most interesting point of this IP traceback approach is that it allows routers to mark certain information on attack packets based on a pre-determined probability. Upon receiving enough marked packets, the victim (or a data collection node) constructs an *attack graph*, which is the set of paths the attack packets traversed.

In [12], authors proposed a criterion to determine how many packets are enough for the PPM algorithm to construct a *meaningful* attack graph, which requires a victim to collect at least one marked packet from every router on the attack graph. Let d be the distance between the furthest router and the victim, and let p_m be the marking probability of every router. Also, let X be the random variable representing the number of marked packets received before the victim can construct to a meaningful graph. Then, the criterion suggested by Savage [12] is given by:

$$E[X] < \frac{\ln d}{p_m(1-p_m)^{d-1}}, \tag{1}$$

and we name it the *Savage's Equation.* The Savage's Equation is working in a single-attacker environment. Further, in a multi-attacker environment, the author claims that:

> *"the number of packets needed to reconstruct each path is independent, so the number of packets needed to reconstruct all paths is a linear function of the number of attackers."*

However, one can easily find that Equation (1) cannot be applied to *all* kinds of network topologies. More specifically, unless the network is a linear one, i.e., a connected acyclic graph with degree at most two, then Equation (1) always *under-estimates* the number of marked packets required and the result may lead to wrong attack graph construction.

We illustrate our finding through the following example. Let there be two network graphs G_1 and G_2 as shown in Figures 1 and 2 respectively. G_1 contains three routers while G_2 contains 14 routers. Nevertheless, the furthest distance d in both networks are the same, i.e., $d = 3$.

In determining the marking probability, a corollary can be derived from Equation (1): the upper bound of $E[X]$ reaches the minimum when $p_m = \frac{1}{d}$ (the proof can be done by differentiating the equation with respect to p_m). This result suggests that routers should choose p_m to be $\frac{1}{d}$ since a smaller upper bound of $E[X]$ implies an earlier attack graph reconstruction. Hence, we choose $p_m = 0.3333$ and

[*]This research is supported by the RGC Grant 4218/04E and by grant, CUHK 4208/04E.

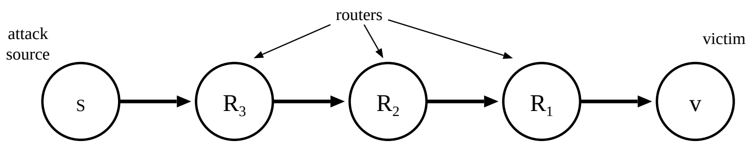


Figure 1: G_1: A linear network with three routers

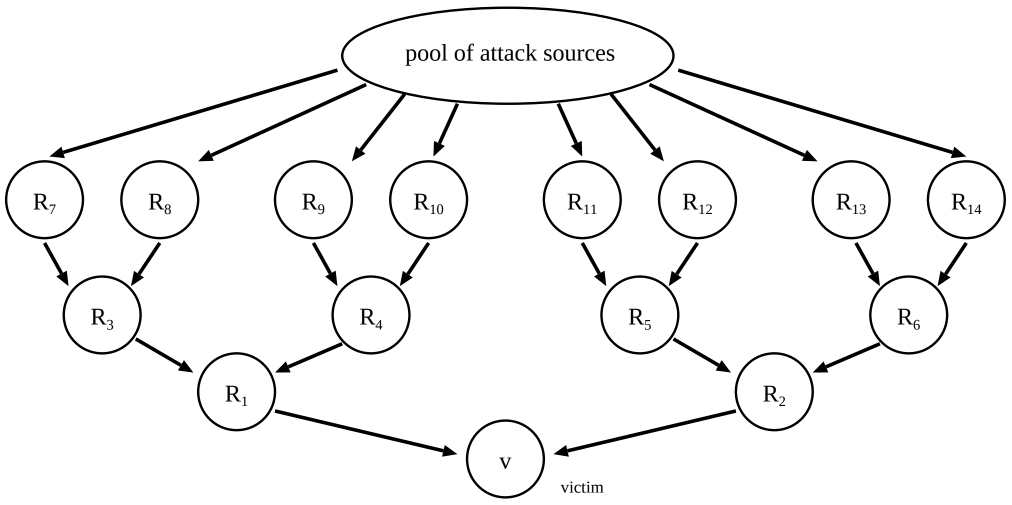


Figure 2: G_2: A complete binary tree-network with 14 routers

we have $E[X] < 7.42$ for G_1. For G_2, since there are eight identical paths, we have $E[X] < 7.42 \times 8 = 59.36$.

We have implemented a simulator to verify the correctness of such bound, and it is performed with the marking probability 0.3333 and with networks G_1 and G_2. The result shows that, with the graph G_1, the average number of marked packets is 6.50, thus, it is consistent with Equation (1). Nevertheless, when the simulation is performed with network G_2, the conclusion is totally different as the average number of marked packets is 103.27, which is quite far from the estimated upper bound based on the Savage's Equation.

The reason for the outstanding result of the later simulation is that: the assumption that "the number of packets needed to reconstruct each path is independent" is wrong. It is quite easy to observe that, in the example network, the paths have shared edges and the number of packets needed for each path is no longer independent.

Now, the problem we are facing is that there does not exist an accurate estimation of the expected number of marked packets $E[X]$. This problem implies a more serious outcome: the graph constructed is *incorrect* with a high probability, where an incorrect constructed graph means the constructed graph does not contains all the routers nor all the links that the attack packets traversed.

In this work, we aim to derive an accurate estimate for the expected number of marked packets $E[X]$. Our proposed solution is to model the PPM algorithm as a *discrete-time Markov chain with an absorbing state*. Secondly, we calculate the expected time till absorbing using the theory of the *fundamental matrix* [16].

2 Proposed Solution

In this section, we present in detail of how one can accurately calculate the expected number of received packets so as to construct an attack graph. First, we state the assumptions of our solution. Second, we revise the *packet marking procedure* of the PPM algorithm before introducing the Markov chain modelling technique. Then, we describe how one can correctly model the PPM algorithm as a Markov chain, and hence can formulate the corresponding transition probabilities. Next, we calculate $E[X]$, the expected number of marked packets, through the use of the fundamental matrix. Lastly, we provide an example to illustrate the computation process.

2.1 Assumptions

In our work, we assume that all routers in the network are equipped with the ability to mark incoming packets following the *packet marking procedure* of the PPM algorithm. We also assume that by the time the victim starts collecting packets, all routers have already invoked the PPM algorithm.

```
Packet Marking Procedure(Packet $w$)

1. let $x$ be a random number in $[0 \ldots 1)$
2. if $x < p_m$, then
3.     write router's address into $w.start$ and 0 into $w.distance$
4. else
5.     if $w.distance = 0$ then
6.         write router's address into $w.end$
7.     end if
8.     increment $w.distance$ by one
9. end if
```

Figure 3: The packet marking procedure of the PPM algorithm

For every router, we assume that they share the same marking probability (as a matter of fact, our solution can handle the cases that there are different marking probabilities among the routers). Also, the network should be a tree, so every router only has one outgoing edge. Lastly, we assume that all leaf routers are the sources of the attack packets.

2.2 A Brief Revision on the Packet Marking Procedure

Before we describe the details of the Markov chain modelling technique, we would like to revisit the details of the *packet marking procedure* of the PPM algorithm. The purpose of this subsection is to let the readers to get familiar with how the routers participate in the PPM algorithm, and hence generate different types of packets.

The purpose of this procedure is to let router to mark information on a packet such that a packet can represent one of edges in the network graph, and the information is contained in the *marking fields* of a marked packet. There are three mark fields of a packet: the start field, the end field, and the distance field[1]. These fields together represent an edge. In the following, we present how the routers can encode an edge in a packet, and the pseudocode of the packet marking procedure is given in Figure 3 for reference.

When a packet arrives at a router, the router determines how to process the packet based on a random number x (line number 1 in the pseudocode). If x is smaller than the pre-defined marking probability p_m, then the router chooses to start encoding an edge. The router sets the start field of the incoming packet to the router's address, and resets the distance field of that packet to zero. Then, the router forwards the packet to the next router. When the packet arrives at the next router, the router again chooses if it should start encoding another edge.

[1]Interested readers should refer to [12] for the design of a marked packet.

Table 1: The list of all possible marked packets arrived at the victim given that the real network graph is G_1 in Figure 1.

Type	Start field	End field	Distance field
Marked by R_1	R_1	v	1
Marked by R_2	R_2	R_1	2
Marked by R_3	R_3	R_2	3

Say, in this time, it chooses not to start encoding a new edge. Then, the router will find out that the previous router has started marking an edge because the distance field of the packet is zero. Therefore, the router sets the end field of the packet to the router's address. In addition, the router increments the distance field of the packet by one so as to indicates the end of the edge encoding process. Now, the start and the end field together encode an edge of the attack graph. For this encoded edge to be received by the victim, successive routers should choose not to start encoding an edge, i.e., the case $x > p_m$ in the pseudocode, because a packet can only encode one edge. Last but not least, every successive router should increment the distance field by one so that the victim will know the distance of the encoded edge.

2.2.1 Types of Packets

According to the above description, when a packet arrives at the victim, the packet should contain either the information of one of the edges that the packet has passed through or no information (because there is a chance that no router chooses to mark that packet). In the following, we name the different types of packets.

We call a packet a *marked packet* if the marking fields of that packet is initialized. Also, we call that a packet is *"marked by the router R_i"* if R_1 is the last router that marks the *start field* of the packet. On the other hand, if none of the routers mark the start field of an incoming packet, then we call that packet an *unmarked packet.*

To illustrate, let us consider the network G_1 in Figure 1 as an example. Table 1 shows a complete set of all possible marked packets arrived at the victim. Let us take the last type of marked packets, i.e., "marked packets marked by R_3", as an example to show how to obtain this kind of marked packets.

When an attack packet arrives at the router R_3, R_3 chooses to mark the start field and initializes the distance field to be zero based on the pre-defined marking probability p_m. Next, on the successive router R_2, R_2 does not chooses to update the start field. Thus, by the packet marking procedure, R_2 updates the end field and increments the distance field by one. On R_1, the start field is not updated again, and the distance field becomes two. At last, the victim increments the distance field to three.

Eventually, the victim has collected a packet marked by the router R_3. By similar steps, we have all kinds of marked packets as shown in Table 1.

Having denoted different types of packets, we are ready to define the Markov chain modelling of the PPM algorithm.

2.3 Markov Chain Modelling

The PPM algorithm collects marked packets from a set of routers, and this process terminates when there are at least one marked packets from every router. Let us now define the underlying Markov process $\mathcal{M}$.

In our model, each Markov state represents a combination of the collected marked packets. Let $\mathcal{R}$ be the set of routers. The state space $\mathcal{S}$ of the Markov process $\mathcal{M}$ is the power set of $\mathcal{R}$ and is stated as follows:

$$\mathcal{S} = \{\mathcal{R}_s \mid \mathcal{R}_s \subseteq \mathcal{R}\}.$$

For the example network G_1 in Figure 1, the state space is as follows:

$$\{\phi,\ (R_1),\ (R_2),\ (R_3),\ (R_1, R_2),\ (R_1, R_3),\ (R_2, R_3),\ (R_1, R_2, R_3)\ \},$$

where, without loss of generality, (R_i, R_j) represents the victim has collected two types of marked packets: packets marked by R_i and packets marked by R_j. On the other hand, ϕ represents the victim has not collected any marked packets.

The physical meaning of the Markov states in $\mathcal{M}$ is as follows. The state which represents the victim has collected none of the marked packets is the *start state*, while the state which represents the victim has collected the complete set of marked packets is the *absorbing state*. When the process reaches the absorbing state, it means the PPM algorithm can construct the attack graph. The remaining state represents the intermediate or transient states of the PPM algorithm.

According to the above descriptions, if the set of routers contains n routers, then there will be totally 2^n Markov states. The total number of Markov states is large and may be computationally expensive to model as the number of routers grows. Nevertheless, one can employ efficient techniques such as aggregation/dis-aggregation [3] and stochastic complementation [6] to reduce the state space of the model.

A discrete-time Markov chain also includes its one-step transition probability matrix, and, in our Markov model, a transition implies an arrival of a packet. A transition occurs if one of the following two situations happens: (1) an arrival of an unmarked packet, or an arrival of a marked packet but the victim has received this type of packet already. Under these two cases, the process stays in the same state; (2) an arrival of a marked packet which was never received by the victim. In this case, the process advances to another state. For example, considering the graph G_1 again, if the current state is (R_1) and the victim receives a packet marked by R_2, then the process transits from (R_1) to (R_1, R_2).

Let there be n routers in the router set $\mathcal{R}$, and denote a router in $\mathcal{R}$ as R_i where $i \in [1, n]$. The transition structure in Equation (2) formally defines the transitions when the Markov process advances from one state to another.

$$\begin{array}{rcl}
\phi & \longrightarrow & R_i; \\
(R_{i_1}) & \longrightarrow & (R_{i_1}, R_{i_2}),\ \text{where } R_{i_1} \neq R_{i_2}; \\
(R_{i_1}, R_{i_2}) & \longrightarrow & (R_{i_1}, R_{i_2}, R_{i_3}),\ \text{where } R_{i_3} \neq R_{i_1} \\
 & & \quad \& R_{i_3} \neq R_{i_2}; \\
\vdots & & \quad\quad \vdots \\
(R_{i_1}, \ldots, R_{i_{n-1}}) & \longrightarrow & (R_{i_1}, R_{i_2}, \ldots, R_{i_{n-1}}, R_{i_n}), \\
 & & \quad \text{where } i_1, i_2, \ldots i_n \in [1, n].
\end{array} \tag{2}$$

Lastly, every transition is associated with a transition probability. In turns, the transition probability involves the formulations of the probability that a packet is marked by one of the routers and the probability that a packet is not marked by any routers. In the next subsection, we specify the *packet-type probabilities* which represent the mentioned probabilities.

2.4 Packet-type Probability

Let R_i be a router of the network graph G. Denote $P(T(G){=}R_i)$ as the packet-type probability which represents the probability that a packet is marked by R_i. Also, denote $P(T(G){=}\phi)$ as the probability that a packet is unmarked. Let the distance between R_i and the victim v be $d(R_i, v)$. Lastly, let n be the total number of participating routers. Then, the packet-type probability for a packet marked by R_i is given by Equation (3) and the packet-type probability for an unmarked packet is given by Equation (4).

$$\begin{aligned}
P(T(G){=}R_i) &= \left(\frac{\text{Number of sources reachable to } R_i}{\text{Total number of sources}}\right) \\
&\quad \times [p_m(1-p_m)^{d(R_i,v)-1}], &(3)\\
P(T(G){=}\phi) &= 1 - \sum_{i=1}^{n} P(T(G){=}R_i). &(4)
\end{aligned}$$

The fractional term in Equation (3) represents the portion of attack packets that a router can receive from different sources. To illustrate the concept, let us consider the two example networks in Figures 1 and 2 again. For G_1, there is only one source, and, for any router R_i in G_1, all packets can reach R_i. Hence, that fractional part is always one. However, in G_2, there are totally eight sources and an intermediate router may not be reachable for every source. For example, half of the sources can reach R_2 while only a quarter can reach R_3. Thus, the fractional part for R_2 and R_3 are 0.5 and 0.25 respectively.

Note the packet-type probability takes unmarked packets into account, and this implies that successive steps in calculating $E[X]$ will include unmarked packets. Nevertheless, one can exclude the unmarked packets by normalizing the packet-type probabilities. Let the normalized packet-type probability be $P(T'(G))$. Equations (5) and

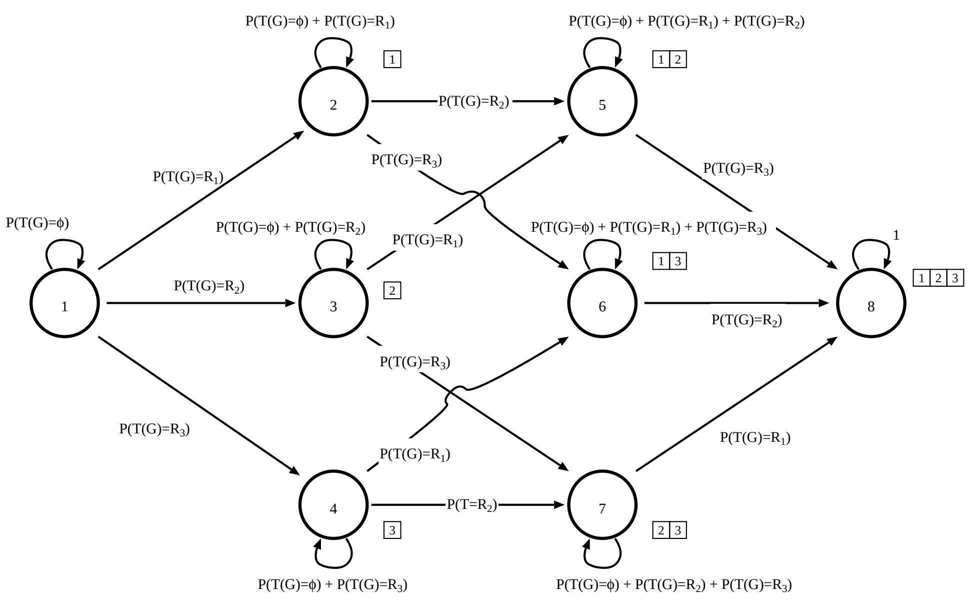


Figure 4: Markov chain model of network G_1 in Figure 1

(6) are the normalized packet-type probabilities for unmarked and marked packets respectively, and we will use the normalized packet-type probability throughout the remaining context.

$$
\begin{aligned}
P(T'(G)=R_i) &= \frac{P(T(G)=R_i)}{1-P(T(G)=\phi)}; \qquad (5)\\
P(T'(G)=\phi) &= 0. \qquad (6)
\end{aligned}
$$

We now show the formulation of the transition probability matrix of the PPM algorithm. Denote the transition probability matrix of the Markov chain as $\mathbf{P}$, and denote an entry at the ith row and the jth column as $\mathbf{P}[i,j]$. With the packet-type probability formulated, we can formally define the transition probability matrix $\mathbf{P}$ according to the Markov state transitions defined in Equation (2) as follows:

$$
\begin{aligned}
\mathbf{P}[\phi,\phi] &= P(T'(G)=\phi);\\
\mathbf{P}[\phi,(R_{i_1})] &= P(T'(G)=R_{i_1});\\
\mathbf{P}[(R_{i_1}),(R_{i_1})] &= P(T'(G)=R_{i_1});\\
\mathbf{P}[(R_{i_1}),(R_{i_1},R_{i_2})] &= P(T'(G)=R_{i_2});\\
\vdots \qquad &\quad \vdots\\
\mathbf{P}[(R_{i_1},\ldots,R_{i_r}),(R_{i_1},\ldots,R_{i_r})] &= \textstyle\sum_{k=1}^{r} P(T'(G)=R_{i_k});\\
\mathbf{P}[(R_{i_1},\ldots,R_{i_r}),(R_{i_1},\ldots,R_{i_{r+1}})] &= P(T'(G)=R_{i_{r+1}});\\
\vdots \qquad &\quad \vdots\\
\mathbf{P}[(R_{i_1},\ldots,R_{i_{n-1}}),(R_{i_1},\ldots,R_{i_{n-1}})] &= \textstyle\sum_{k=1}^{n-1} P(T'(G)=R_{i_k});\\
\mathbf{P}[(R_{i_1},\ldots,R_{i_{n-1}}),(R_{i_1},\ldots,R_{i_n})] &= P(T'(G)=R_{i_n});\\
\mathbf{P}[(R_{i_1},\ldots,R_{i_n}),(R_{i_1},\ldots,R_{i_n})] &= 1.
\end{aligned}
$$

Before we apply the fundamental matrix technique to calculate the expected number of marked packets $E[X]$, we illustrate the steps in modelling the PPM algorithm into a Markov chain in the following subsection.

2.5 Example on Discrete-time Markov Chain Modelling

We present an example of the discrete-time Markov chain model using the network G_1 in Figure 1. When one follows the modelling rules stated in the previous subsection, one can construct the Markov chain as shown in Figure 4. Every state except the start state, state 1, has a set of boxes along side with it. For example, state 2 has a '1' in the box. These boxes indicates which types of packets the victim has received. Then, for state 2, the box states that the marked packets from R_1 have been received. For state 8, all types of marked packets are collected with the three boxes appeared, and it is the absorbing state. Its stationary transition probability is equal to one, and this implies that further packet arrivals will not change the state of the process.

In the next step, we set the marking probability to be $p_m = \frac{1}{d} = \frac{1}{3}$ again, and, then, the packet-type probabilities are then given by Equation (7). Accordingly, one can construct the transition probability matrix according to both Figure 4 and Equation (7) as shown in Figure 5.

$$
P(T(G)=i) = \begin{cases} 0.4737 & i = R_1;\\ 0.3158 & i = R_2;\\ 0.2105 & i = R_3;\\ 0 & i = \phi. \end{cases} \qquad (7)
$$

The transition probability matrix is a vast source of information about the PPM algorithm. Let the transition probability matrix to be $\mathbf{P}$. If we raise the matrix to a certain power, say k, then $\mathbf{P}^k$ represents the system's states after k packets have been arrived, and the entry $\mathbf{P}^k[1,8]$ in $\mathbf{P}^k$ represents the probability that, the system transits from state 1 to state 8 after k packets have been arrived. More importantly, this entry represents the ac-

$$\mathbf{P} = \begin{pmatrix} 0 & 0.4737 & 0.3158 & 0.2105 & 0 & 0 & 0 & 0 \\ 0 & 0.4737 & 0 & 0 & 0.3158 & 0.2105 & 0 & 0 \\ 0 & 0 & 0.3158 & 0 & 0.4737 & 0 & 0.2105 & 0 \\ 0 & 0 & 0 & 0.2105 & 0 & 0.4737 & 0.3158 & 0 \\ 0 & 0 & 0 & 0 & 0.7895 & 0 & 0 & 0.2105 \\ 0 & 0 & 0 & 0 & 0 & 0.6842 & 0 & 0.3158 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.5263 & 0.4737 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0000 \end{pmatrix}$$

Figure 5: Transition probability matrix formulated by the Markov chain in Figure 4

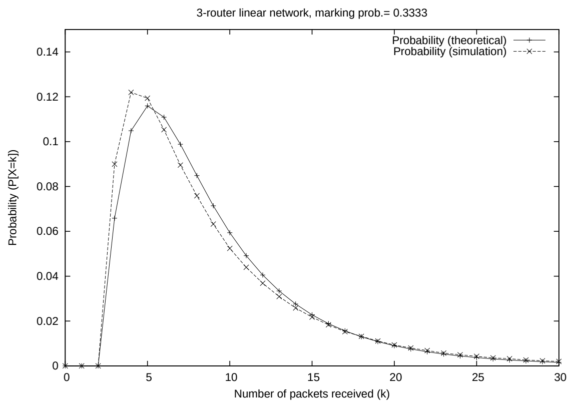


Figure 6: Comparison of the number of required marked packets between the simulation results and the theoretical results of example network G_1

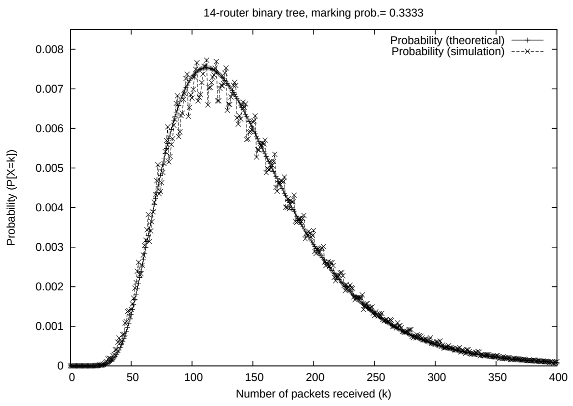


Figure 7: Comparison of the number of required marked packets between the simulation results and the theoretical results of example network G_2

cumulative probability that k marked packets are enough to construct an attack graph, i.e,

$$\mathbf{P}^k[1,8] = \sum_{i=0}^{k} P[X{=}i].$$

Further, one can obtain the probability $P[X=k]$ as follows:

$$P[X{=}k] \;=\; \mathbf{P}^k[1,8] - \mathbf{P}^{k-1}[1,8]. \tag{8}$$

By Equation (8), one can construct the probability density function of $P[X]$ as shown in Figure 6. This figure shows a close result between the simulation data (the distribution of the number of required marked packets generated by 10,000 individual simulation samples) and the density function.

Also, in Figure 7, it shows the probability density function and the plot of the simulation with 10,000 individual simulation samples of the example network G_2. The figure not only show a close relation between the two plots, but it also suggests that the PPM algorithm requires far more packets for network G_2 than that for network G_1. In this next subsection, we show how one can calculate the expectation $E[X]$.

2.6 Fundamental Matrix

The calculation of the fundamental matrix [16] is the last and essential step to obtain $E[X]$. In here, we describe the steps involved in the calculations of the fundamental matrix and $E[X]$, the expected number of marked packets received before the victim can construct a meaningful attack graph.

To calculate the fundamental matrix, one has to first partition the state space of the Markov chain into two mutually exhaustive and exclusive partitions: $\mathcal{S}_t$ is the partition for all transient states while $\mathcal{S}_a$ is the partition for the absorbing states (e.g., state wherein all marked packets are received by the victim). After the partition, the one-step transition probability matrix can be represented as:

$$\mathbf{P} = \left(\begin{array}{c|c} \mathbf{Q} & \mathbf{C} \\ \hline \mathbf{0} & 1 \end{array}\right).$$

In other words, in the case that there is only one absorbing state, and if $\mathbf{P}$ is of size $n \times n$, then $\mathbf{Q}$ is of size $n-1 \times n-1$ wherein $\mathbf{Q}$ is the transition sub-matrix for all transient states in $\mathcal{S}_t$ (e.g., these are states which have less than n received marked packets). The sub-matrix $\mathbf{Q}$ is used to calculate the fundamental matrix $\mathbf{M}$ as:

$$\mathbf{M} = (\mathbf{I} - \mathbf{Q})^{-1} = \sum_{k=0}^{\infty} \mathbf{Q}^k. \tag{9}$$

$$\mathbf{M} = \begin{pmatrix} 1.0000 & 0.9001 & 0.4616 & 0.2666 & 2.3890 & 0.9999 & 0.3829 \\ 0 & 1.9001 & 0 & 0 & 2.8505 & 1.2665 & 0 \\ 0 & 0 & 1.4616 & 0 & 3.2890 & 0 & 0.6495 \\ 0 & 0 & 0 & 1.2666 & 0 & 1.8999 & 0.8444 \\ 0 & 0 & 0 & 0 & 4.7506 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3.1666 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2.1110 \end{pmatrix}$$

Figure 8: Fundamental matrix calculated by Equation (9) with transition probability matrix $\mathbf{P}$ shown in Figure 5

Theorem 1. *Let* $\mathbf{M}$ *be the fundamental matrix of the underlying Markov chain describes by the one-step transition probability matrix* $\mathbf{P}$*. The* $(i,j)^{th}$ *entry of* $\mathbf{M}$*, denote as* $\mathbf{M}[i,j]$*, represents the expected number of visits from the transient state* i *to the transient state* j *before entering the absorbing state.*

Proof. Note that $\mathbf{Q}^n$ represents the probability of going from the transient state i to the transient state j in n steps without leaving the partition $\mathcal{S}_t$. Consider all possible sample paths of moving from state i to state j. One can move from state i to state j with probability $\mathbf{Q}[i,j]$ in one transition, or with probability $\sum_{k=1}^{n-1} \mathbf{Q}[i,k]\mathbf{Q}[k,j]$ in two transitions, etc. In general, let $\mathbf{M}[i,j]$ be the probability of moving from state i to state j without leaving the partition $\mathcal{S}_t$, we have

$$\mathbf{M}[i,j] = \begin{cases} \mathbf{Q}[i,j] + \mathbf{Q}^2[i,j] + \mathbf{Q}^3[i,j] + \cdots & \text{for } i \neq j, \\ 1 + \mathbf{Q}[i,i] + \mathbf{Q}^2[i,i] + \mathbf{Q}^3[i,i] + \cdots & \text{for } i = j. \end{cases}$$

Also, $\mathbf{M}[i,j]$ has the probabilistic interpretation as the expected number of visits to state i to state j before exist to $\mathcal{S}_a$ given that the process starts from state i. □

Based on the theorem above, one can calculate the expected number of visits from the starting state to every transient states before entering the absorbing state, which is $E[X]$ in our application. Thus, $E[X]$ can be expressed as:

$$E[X] = \sum_{i=1}^{n-1} \mathbf{M}[1,i] \qquad (10)$$

where state 1 is the initial state that the victim has not received any marked packet.

2.7 Example on Calculating $E[X]$

We continue the example in Section 2.5 to calculate $E[X]$. By following the method in Equation (9), one can calculate the fundamental matrix $\mathbf{M}$ as shown in Figure 8. Lastly, by following Equation (10), $E[X]$ is calculated as follows:

$$\begin{aligned} E[X] &= 1.0000 + 0.9001 + 0.4616 + 0.2666 + 2.3890 \\ &\quad +0.9999 + 0.3829 \\ &= 6.4001, \end{aligned}$$

where $E[X]$ is very close to the simulation result of 6.50.

For the example network G_2 shown in Figure 2, the number of state is too large to be presented in this work. Instead, we employ the stochastic complementation technique to reduce the state space and the result is $E[X] = 103.57$, which is very close to the simulation result of 103.27, as compare with the result of Savage's formula of 56.36.

2.8 Further Examples

We present more examples of the comparison between the Savage's equation, the simulation results, and the results via the Markov chain modelling. Figures 9 and 10 are slightly more complex networks with the furthest router distance being three hops from the victim, and there are totally two paths leading from the attackers to the victim. Our solution shows close results to the simulations, for both the cases of unmarked and marked packets. It is interesting to observe that the networks G_1, G_3, and G_4 are quite simple and similar, however, according to Table 2, the value of $E[X]$ for these networks are not the same, or even not close to each other.

3 Conclusion

In this work, we have shown that there is a termination problem in PPM algorithm: the calculation of the expected number of marked packets $E[X]$ is not accurate for general networks. Without a correct calculation of the expected number of marked packets, one cannot judge when is the right time to start the attack graph reconstruction procedure, or an incomplete attack graph is reconstructed. We propose to model the PPM algorithm as a discrete-time Markov chain, and, by using the theory of the fundamental matrix, $E[X]$ can be found precisely. Also, the Markov chain model can be simplified by the efficient techniques like the aggregation/dis-aggregation and the stochastic complementation. Lastly, the simulation results and the theoretical results from the Markov chain model are consistent. This shows that the proposed method is effective in calculating $E[X]$.

References

[1] M. Adler, "Trade-offs in probabilistic packet marking for IP traceback," *Journal of the ACM*, vol. 52,

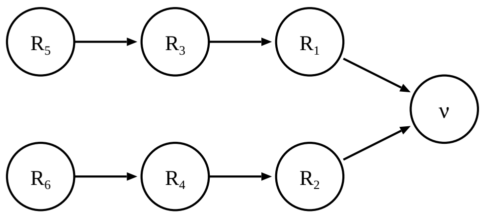


Figure 9: G_3: furthest router distance $d = 3$

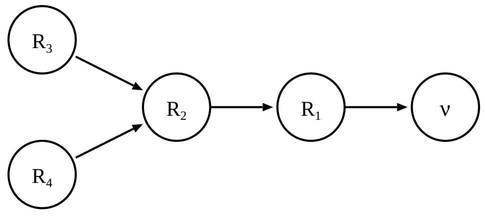


Figure 10: G_4: furthest router distance $d = 3$

		Number of Required Packets				
		marked packets only			**with unmarked packets**	
Network	**Dist.**	**Savage's**	**Simulation**	**Our Work**	**Simulation**	**Our Work**
G_1	3	7.42	6.50	6.40	9.03	9.09
G_2	3	59.36	103.27	103.57	148.31	147.18
G_3	3	14.84	17.46	17.43	24.80	24.77
G_4	3	14.84	14.61	14.65	20.75	20.82

Table 2: Comparison between Savage's equation, simulation results, and results of the Markov chain modelling

pp. 217–244, Mar. 2005.

[2] Computer Emergency Response Team, *Cert Advisory Ca-2000-01: Denial-of-service Developments*, http://www.cert.org/advisories/CA-2000-01.html.

[3] P. J. Courtois, *Decomposability: Queueing and computer system applications*, Academic Press, 1977.

[4] D. Dean, M. Franklin, and A. Stubblefield, "An algebraic approach to IP traceback," in *Proceedings of Network and Distributed System Security Symposium, NDSS'01*, Feb. 2001.

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

[6] L. Golubchik and J. C.S. Lui, "Bounding of performance measures for threshold-based queueing systems: Theory and application to dynamic resource management in video-on-demand servers," *IEEE Transactions of Computers*, vol. 51, pp. 353–372, Apr. 2002.

[7] J. Ioannidis and S. M. Bellovin, "Implementing Pushback: Router-based defense against ddos attacks," in *Proceedings of Network and Distributed System Security Symposium*, Feb. 2002.

[8] K. T. Law, J. C. S. Lui, and D. K. Y. Yau, "You can run, but You can't hide: An effective methodology to traceback DDoS attackers," *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, no. 9, pp. 799-813, 2005.

[9] R. Mahajan, S. M. Bellovin, S. Floyd, J. Ioannidis, V. Paxson, and S. Shenker, "Controlling high bandwidth aggregates in the network," *ACM SIGCOMM Computer Communication Review*, vol. 32, pp. 62–73, July 2002.

[10] K. Park and H. Lee, "On the effectiveness of route-based packet filtering for distributed dos attack prevention in power-law internets," in *Proceedings of ACM SIGCOMM'01*, pp. 15-26, 2001.

[11] K. Park and H. Lee., "On the effectiveness of probabilistic packet marking for IP traceback under denial of service attack," in *Proceedings of IEEE INFOCOM'01*, pp. 338-347, 2001.

[12] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, "Practical network support for IP traceback," in *Proceedings of the 2000 ACM SIGCOMM Conference*, pp. 295-306, 2000.

[13] 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 *Proceedings of the ACM SIGCOMM 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, Aug. 2001.

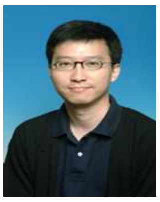
[14] D. X. Song and A. Perrig, "Advanced and authenticated marking schemes for IP traceback," in *Proceedings of IEEE INFOCOM'01*, pp. 1–9, Apr. 2001.

[15] E. Steven and M. Bellovin, *ICMP Traceback Messages, Internet Draft: draft-bellovin-itrace-00.txt*, submitted Mar. 2000, expiration date Sep. 2000.

[16] K. S. Trivedi, *Probability and Statistics with Reliability, Queuing and Computer Science Applications*, Wiley-Interscience, 2002.

[17] D. K. Y. Yau, J. C. S. Lui, F. Liang, and Y. Yeung, "Defending against distributed denial-of-service attacks with max-min fair server-centric router throttles," *IEEE/ACM Transactions on Networking*, vol. 13, no. 1, pp. 29-42, Feb. 2005.

Tsz-Yeung Wong is a Ph.D candidate in the Department of Computer Science and Engineering at the Chinese University of Hong Kong. He received his M.Phil and B.Sc degrees from the Department of Computer Science at the Chinese University of Hong Kong in 2002 and 2000 respectively. His research interests include distributed algorithms, networking, and computer and network security.



John Chi-Shing Lui was born in Hong Kong. He received his Ph.D. in Computer Science from UCLA. After his graduation, he joined the IBM Almaden Research Laboratory/San Jose Laboratory and participated in various R&D projects on file systems and parallel I/O architectures. He later joined the Department of Computer Science and Engineering at the Chinese University of Hong Kong. He has been a visiting professor in computer science departments at UCLA, Columbia University, University of Maryland at College Park, Purdue University, University of Massachusetts at Amherst and Universit degli Studi di Torino in Italy. His research interests span both in system and in theory/mathematics, with current research interests in theoretic/applied topics in data networks, distributed multimedia systems, network security, OS design issues and mathematical optimization and performance evaluation theory. John received various departmental teaching awards and the CUHK Vice-Chancellor's Exemplary Teaching Award, as well as the co-recipient of the Best Student Paper Award in the IFIP WG 7.3 Performance 2005 and the IEEE/IFIP Network Operations and Management (NOMS) Conference. He is an associate editor in the Performance Evaluation Journal, member of ACM, a senior member of IEEE and an elected member in the IFIP WG 7.3. John was the TPC co-chair of ACM Sigmetrics 2005. His personal interests include films and general reading.

Man-Hon Wong received his B.Sc. and M.Phil. degrees from The Chinese University of Hong Kong in 1987 and 1989 respectively. He then went to University of California at Santa Barbara where he got the Ph.D. degree in 1993. Dr. Wong joined The Chinese University of Hong Kong in August 1993 as an assistant professor. He was promoted to associate professor in 1998. His research interests include transaction management, mobile Databases, data replication, distributed systems, expert systems and applications of fuzzy logic.