PERFORMANCE EVALUATION OF A TTL-BASED DYNAMIC MARKING
SCHEME IN IP TRACEBACK

A Thesis
Presented to
The Graduate Faculty of The University of Akron

In Partial Fulfillment
Of the Requirements for the Degree
Master of Science

Shanmuga Sundaram Devasundaram
December 2006
PERFORMANCE EVALUATION OF A TTL-BASED DYNAMIC MARKING
SCHEME IN IP TRACEBACK

Shanmuga Sundaram Devasundaram

Thesis

Approved:  
Advisor  
Dr. Xuan-Hien Dang 

Faculty Reader  
Dr. Zhong-Hui Duan 

Faculty Reader  
Dr. Yingcai Xiao 

Department Chair  
Dr. Wolfgang Pelz 

Accepted:  
Dean of the College  
Dr. Ronald F. Levant 

Dean of the Graduate School  
Dr. George Newkome 

Date
ABSTRACT

Providing networks with countermeasures against Denial of Service (DoS) attacks has become a pressing security issue in the Internet today. Network services get disrupted or become totally unavailable as malicious attackers flood a victim network with large amount of useless traffic. For accountability purpose and to thwart those attacks, it is essential to identify the source of these attacks, which is usually concealed using faked or spoofed IP addresses, and is known as the IP Traceback problem.

Packet marking is a traceback approach that calls for routers to mark packets along the attack path with self-identifying information. In Probabilistic Packet Marking (PPM) routers probabilistically decide whether or not to mark packets. A victim node relies on the amount of marked packet samples received to reconstruct the attack path. However, a fixed marking probability set for all routers in PPM has proved to be ineffective as marked packets from distant routers are more likely to be remarked by downstream routers. This entails a loss of information and leads to increase in the volume of packets needed to reconstruct the attack path. Enabling each router to adjust its marking probability so as to obtain equal samples of marked packets, in particular from the furthest routers would help in minimizing the time taken to reconstruct the attack path.
Dynamic schemes have been proposed for adjusting the marking probability, which can be derived by accurately estimating a router’s position in the attack path. However, most schemes are highly dependent on the underlying protocols and require routers to have knowledge of distance information to the potential victim node. This adversely increases the router overhead and is time consuming for real-time packet marking scenarios.

In this work we propose an algorithm that dynamically set the value of the marking probability based on the 8-bit Time-To-Live (TTL) field in the IP header, which is a value that can be directly accessed by routers without external support. Our proposed scheme utilizes the variable TTL value as an estimate of the distance traveled by a packet and thereby its position in the attack path to derive the marking probability value. Our algorithm was simulated with a number of test cases using a user-friendly simulator that was developed to that effect. Results in terms of false positives, reconstruction time and number of packets needed for reconstruction have shown the efficacy of our dynamic scheme, which offers significantly higher precision with fewer overheads both at the router and at the victim in reconstructing the attack path. The main advantages of the proposed scheme reside both in its simplicity and low router overhead while offering comparable results with other dynamic schemes and outperforming static schemes at large attack distances.

Future work includes fine-tuning the derivation of the dynamic marking probability to further improve performance at larger attack distances and a study of its applicability and performance in IPv6 networks.
ACKNOWLEDGEMENTS

I am indebted to all of my professors, whose able guidance gave me the knowledge and patience necessary to complete this thesis and my degree. In particular, I would like to thank my advisor, Dr. Xuan-Hien Dang, for answering my many questions and her suggestions and criticisms to help me wend my way through the oft-daunting tasks my research presented, and she met my sometimes-unfocused queries and concerns with the utmost patience. I must also single out Dr. Wolfgang Pelz, in gratitude for the hours he frittered away advising me throughout my graduate studies. I am certain that I would have succumbed to numerous academic pitfalls had he not been willing to guide me around them. Last but not least, I would thank my readers, Dr. Zhong-Hui Duan and Dr. Yingcai Xiao for accepting my request to be in my thesis committee and for their time to read and suggest changes to my thesis. I could not have asked for a better committee for my thesis.
DEDICATION

As I leave the loving arms of academia I would like to dedicate this thesis to:

My parents, for their patience and love in completing my degree,
My sister for all her support,
My uncles and aunts for their love, motivations and inspirations,
My cousins, for being my friends and their motivations,
My friends for being there when I needed them, and
My teachers and professors for all the knowledge and wisdom they have instilled in me throughout my life so far.
Also, I would like to dedicate my thesis to very few special people I have met in my life so far, who has inspired me and has made a positive change in my life. They are:
Mr. Mike Manickam,
Mr. Jeff Wiedl,
Mr. A.P.N Paramasivan,
Mr. Raja,
Mr. Rashmi Yajnik,
Mr. D.V.V Prasad and
Mr. Venkateshvara Rao.
Ms. Janete Juliano, and
Ms. Mae Schreiber.
I would like to thank Mr.Manoranjan, 3SG Corporation, for giving me the opportunity to do my internship at his company and making my graduate degree curvaceous. Many thanks to all!

Thirukkural: 391

Lore worth learning, learn flawlessly
Live by that learning thoroughly.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>LIST OF FIGURES</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ix</td>
</tr>
</tbody>
</table>

## CHAPTER

I. INTRODUCTION ......................................................... 1

II. BACKGROUND AND RELATED WORK ................................. 4

2.1 General Background .............................................. 4

2.2 Definitions and terminologies .................................. 5

2.3 Link testing ......................................................... 6

2.4 Logging .............................................................. 7

2.5 Packet marking ...................................................... 8

2.5.1 ICMP based marking ........................................... 8

2.5.2 Packet marking in IP header ................................ 10

III. DESIGN AND IMPLEMENTATION ................................. 18

3.1 Design .............................................................. 18

3.2 Metrics ............................................................. 21

3.3 Simulation .......................................................... 21

3.4 Implementation .................................................... 22

IV. RESULTS AND ANALYSIS ..................................... 25

4.1 Number of packets needed to reconstruct the attack path ...... 26
4.2 Reconstruction time of the schemes compared................................28

4.3 Performance in DDOS attacks..................................................30

V. CONCLUSION AND FUTURE WORK........................................32

REFERENCES.............................................................................34

APPENDIX..................................................................................36
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Upstream router map</td>
<td>6</td>
</tr>
<tr>
<td>2. Link testing</td>
<td>6</td>
</tr>
<tr>
<td>3. Logging</td>
<td>7</td>
</tr>
<tr>
<td>4. ICMP Traceback</td>
<td>8</td>
</tr>
<tr>
<td>5. Standard IPv4 header</td>
<td>10</td>
</tr>
<tr>
<td>6. Probabilistic Packet Marking</td>
<td>11</td>
</tr>
<tr>
<td>7. Savage’s Overloaded IPv4 header</td>
<td>12</td>
</tr>
<tr>
<td>8. Song and Perrig’s Overloaded IPv4 header</td>
<td>13</td>
</tr>
<tr>
<td>9. Song &amp; Perrig’s packet marking in IP header</td>
<td>13</td>
</tr>
<tr>
<td>10. Song &amp; Perrig’s Marking and Reconstruction Algorithm</td>
<td>14</td>
</tr>
<tr>
<td>11. XOR Encoding</td>
<td>16</td>
</tr>
<tr>
<td>12. Marking Procedure of TTL-based scheme</td>
<td>20</td>
</tr>
<tr>
<td>13. Screen shot of the application</td>
<td>24</td>
</tr>
<tr>
<td>14. Comparison of average packets needed</td>
<td>26</td>
</tr>
<tr>
<td>15. Comparison of reconstruction time</td>
<td>28</td>
</tr>
<tr>
<td>16. Comparison of average false positives in DDoS attacks</td>
<td>30</td>
</tr>
<tr>
<td>17. Comparison of average reconstruction time in DDoS attacks</td>
<td>31</td>
</tr>
</tbody>
</table>
CHAPTER I

INTRODUCTION

Denial Of Service (DoS) attacks are one of the major problems on today’s highly networked environment. These focused attacks thwart the valuable services provided by the network and deny access to legitimate users. Identifying the attack origin in order to hold the attacker accountable has proved to be a difficult task as attackers often use spoofed IP addresses and is known as the IP Traceback problem. This difficulty comes from the open and stateless nature of the IP protocol, which by design does not include built-in mechanisms in routers to verify the authenticity of the source IP address inscribed in IP packets.

Many IP traceback techniques have been proposed for the IP Traceback problem, each with their own advantages and disadvantages. One of the most promising approaches is by marking packets with routing information, which is used to reconstruct the attack path [1-5]. In Probabilistic Packet Marking (PPM) [1], routers probabilistically decide whether or not to mark packets. A victim node then relies on the volume of marked packet samples received to reconstruct the attack path. However, in PPM, a fixed marking probability is set for all routers, which was demonstrated to be ineffective as marked packets from distant routers are more likely to be remarked by downstream
Dynamic packet marking schemes were introduced to overcome such drawbacks from static packet marking schemes [3] by allowing routers to individually adjust their marking probability. Research has shown that a fixed marking probability set at $l/d$, where $d$ is the distance from the current router to the victim, would provide optimal results. However, the distance $d$ is not known in advance. Dynamic approaches seek to provide a close estimate of this distance for each router, mainly relying on various external parameters available. Many techniques involve the use of the underlying protocols to retrieve and store distance information into the routing table to adjust the marking probability. This adversely increases the router overhead and is time consuming for real-time packet marking scenarios.

In this work, we propose a new technique for dynamically adjusting the marking probability by each individual router and which does not suffer from those shortcomings. The major challenge is to derive the marking probability in such a way that it does not exceed an upper threshold value which might cause remarking of the packets and should not go below a lower threshold value, causing an increase in the number of packets needed to reconstruct the path. We propose to utilize the 8-bit Time-To-Live (TTL) field in the IP header, which is a value that is directly accessible to routers without external support. TTL values are used to avoid the indefinite routing of packets in an internet system. In practice the TTL values are set by the sender of the IP datagram. Each router processing the packet decrements the TTL value by one. When the TTL value reaches
zero, the packet is discarded and this helps in preventing the unnecessary circulation of packets and reduces router overhead in the Internet.

Our proposed scheme utilizes the value of the TTL in the IP header, as an estimate of the distance the packet has traveled and thereby its position in the attack path to derive the new marking probability. We applied our TTL based marking scheme and simulated various attack scenarios using a user-friendly simulator that was developed to that effect. With results obtained we see that our proposed scheme proves to be more efficient and has higher precision than other marking schemes in terms of number of false positives, reconstruction time and number of packets needed to reconstruct the attack. The main advantages of the proposed scheme reside in both its simplicity and low router overhead while offering comparable results with other dynamic schemes and outperforming static schemes at larger attack distances.

This thesis is organized as follows. In Chapter 2, we present a brief background to the IP traceback problem and some of its proposed solutions. The proposed scheme has been detailed in chapter 3 and the analysis of results is presented in chapter 4. Finally, chapter 5 provides conclusion and future work to improve the proposed technique.
CHAPTER II

BACKGROUND AND RELATED WORK

2.1 General Background

Denial of service (DoS) is one of the most common attacks on the Internet. The DoS attacks consume immense network resources, as a multitude of packet floods the victim network. The attacks can be from one attacker or from many attackers. The latter is called as distributed DoS (DDoS). These attacks are simple to implement, hard to prevent, and difficult to trace. IP traceback is one of the many effective methods for restoring normal network functionality as quickly as possible, preventing reoccurrences, and, ultimately, holding the attackers accountable [6].

They can be roughly categorized into 3 different categories: link testing, logging and packet marking and are described in the following sections. Many IP traceback techniques have been proposed to solve the IP traceback problem, each with its own advantages and drawbacks [1-5].
2.2 Definitions and terminologies

In this section, we describe the basic concepts and terminologies that are referred to in this thesis. An attack source is a device that is used to generate the attack traffic. A victim is a system that provides a service over the internet, whose service availability is disrupted during the attack. The directed acyclic graph (DAG) rooted at victim V in figure 1 represents the network as seen from the victim V and a distributed denial-of-service attack from A_2 to A_3. V could be a single host under attack or a network border device such as firewall protecting a network. Nodes R_i represent the routers, which we refer to as upstream routers from V, and we call the graph the map of upstream routers. The routers below the A_i are called downstream routers and we call the graph the map of downstream routers. The leaves A_i represent the potential attack origins. The attack path from A_i is the ordered list of routers between A_i and V that the attack packet has traversed, for example, R_6-R_3-R_2-R_1. The distance of R_i from V on a path is the number of routers between R_i and V on the path. The attack graph is the graph composed of all the attack paths. Attack packets used in the DoS attacks are called attack packets. We call a router false positive if it is in the reconstructed attack graph but not in the real attack graph. Similarly we call a router false negative if it is in the true attack graph but not in the reconstructed attack graph. We call a solution to the IP traceback problem robust if it has very low rate of false positives and reconstruction time.
2.3 Link testing

Link testing methods also called hop-by-hop tracing [15] work by inspecting network links between routers to get back to the attack source. The link testing technique starts from the router closest to the victim and interactively tests its incoming (upstream) links to determine which one carries the attack traffic. This process repeats recursively on the upstream routers until it reaches the attack source as shown in Figure 1.

![Figure 1: Upstream router map](image)

Figure 1: Upstream router map

![Figure 2: Link testing](image)

Figure 2. Link testing.
Link testing is a reactive type of traceback and requires the attack to remain active until the traceback is done. Input debugging [7] is one implementation of the link testing approach.

2.4 Logging

Logging is another way to traceback to the origin of the attack. All traffic is logged by key routers in the Internet and data-mining techniques are used to traceback to the attacker’s source. When an attack has been detected the victim can poll the upstream routers to check if the router has logged the attack packets. By recursively polling the upstream routers, the victim reconstructs the attack path. Logging seems to be a straightforward solution and allows accurate analysis of attack traffic even after the attack is over. But the main drawbacks of the technique are the amount of processing power involved and the amount of data needed to be logged and to be shared to the partners involved in the attack traceback. Figure 3 shows the logging in action.

![Figure 3. Logging.](image)
Logging can be used for local and internal purposes. Present day logging-based traceback methods use sliding time window for storing logged data. This avoids excessive storage requirements and reduces analysis cost. This is a trade off, in exchange for reactive traceback.

2.5 Packet Marking

Packet marking techniques rely on routers along an attack path to mark packets with self-identifying information. Routers can do so by either generating additional ICMP based packets or by directly inscribing such information in the packet header.

2.5.1 ICMP based marking

iTrace is an ICMP based marking technique proposed by Internet Engineering Task Force (IETF) in July 2000 [7]. This approach uses ICMP traceback router-generated messages, which the victim receives in addition to information from regular network traffic. These messages contain partial path information that indicates where the packet came from, when it was sent, and its authentication. Figure 4 shows ICMP marking in action.

---

**Figure 4. ICMP Traceback.**
Network administrators, with the help of these messages can trace a packet’s path back to its origin. To reduce the overhead involved in sending those additional packets in the network a router would generate an ICMP traceback message for only one in 20,000 packets passing through it. This probability has proved to be sufficient to find out the attack traffic’s actual path. In a DoS attack, the victim’s receives thousands of packets in a matter of seconds. iTrace’s disadvantage arises in DDoS attacks in which each zombie process contributes only a small amount of the total attack traffic. In such cases, the probability of choosing an attack packet is much smaller than the sampling rate used. Based on the information provided in the routing table, the decision module would select the kind of packet to use next to generate an iTrace message. It then sets a special bit in the packet-forwarding table. When the special bit is set it indicates that the next packet corresponding to that particular forwarding entry will be chosen to generate an iTrace message. The iTrace generation module then processes this chosen packet and sends a new iTrace message. The intention-driven traceback also lets a recipient network signal whether it is interested in receiving iTrace packets, which helps in increasing the proportion of messages considered useful to the receiving network. This scenario would also be helpful if a given network suspects or detects that it is under attack. It could request iTrace packets from the upstream routers to identify the attack traffic’s origin.
2.5.2 Packet marking in IP header

Packet-marking methods [1-5] have attracted a lot of attention in IP traceback. IP packets are marked with important information by the routers along the path. The victim uses the markings in the IP packets and tries to reconstruct the attack path. In this method, reconstruction of the attack path relies on the volume of marked packets collected at the victim. PPM received a widespread attention because of its low cost in terms of router processing overhead and reconstruction time.

The marking in the IP packet poses an important question as how and where in the IP header the information need to be inscribed as shown in the standard IPV4 header in figure 4.

<table>
<thead>
<tr>
<th>Version</th>
<th>IHL</th>
<th>Type of Service</th>
<th>Total Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification</td>
<td>Flags</td>
<td>Fragment Offset</td>
<td></td>
</tr>
<tr>
<td>Time to Live</td>
<td>Protocol</td>
<td>Header Checksum</td>
<td></td>
</tr>
<tr>
<td>Source Address</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Destination Address</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 5. Standard IPv4 header.

The two possible fields that can be overloaded to inscribe such marking information are the options field and identification field. The options field in the IP packet is used for adding extra information for additional processing like testing, debugging and security. However, marking in the options field, poses a few problems. Marking in the options field increases the packet size, which might cause the IP packet to be fragmented along the way. So, options field is not an appropriate field to be used for
marking as it might pose a problem while reconstructing the attack path. Also, there is a good possibility that an attacker might modify the route data and falsify the information.

The identification field in the IP packet was designed to hold the fragmented packet id. It differentiates the fragments of IP packets and allows proper reassembly on the receiver side. However, it was observed that less than 0.25% of the packets on the Internet is fragmented, thereby relying on the identification field. It was asserted to be negligible and served as a justification for overloading the identification field to inscribe the marking information. Figure 6 shows packet marking in action.

Many algorithms for packet marking were proposed [1-8], ranging from simply appending the current router address to employing probabilistic traffic-sampling and compression methods. Savage et al. first described and implemented probabilistic sampling with a probability of 1/25 to avoid excessive overhead on the routers’ packet marking. Furthermore, each packet stores partial route information (an edge) rather than

Figure 6. Probabilistic packet marking.
the full path. Using this approach in conjunction with compression techniques and an additional field to prevent spoofing of routing information, Savage et al. proposed to use the IP header’s 16-bit identification field to store the router’s information. Figure 7 shows Savage’s overloaded IP header.

<table>
<thead>
<tr>
<th>Version</th>
<th>IHL</th>
<th>Type of Service</th>
<th>Total Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offset</td>
<td>Distance</td>
<td>Edge Fragment</td>
<td>Flags</td>
</tr>
<tr>
<td>Time to Live</td>
<td>Protocol</td>
<td>Header Checksum</td>
<td></td>
</tr>
<tr>
<td>Source Address</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Destination Address</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 7. Savage’s overloaded IPv4 header.

Dawn Song and Adrian Perrig [2] proposed modifications to Savage’s edge-identification-based PPM method to further reduce storage requirements by storing a hash of each IP address instead of the address itself. The approach assumes the victim possesses a complete network map of all upstream routers. After edge-fragment reassembly, their method compares the resulting IP address hashes to the router IP address hashes derived from the network map (to facilitate attack path reconstruction). This modified method was demonstrated to be more effective against DDoS attacks than previous methods. The authors also proposed an authentication-marking scheme that uses message authentication codes to prevent packet-content tampering by compromised routers along the attack path. Figure 8 shows the overloaded IP header of Song and Perrig.
Figure 8. Song and Perrig’s Overloaded IPv4 header

In this approach, the routers IP address is encoded and its hash value is stored in the header. In this scheme, the 16-bit IP identification field is divided into a 5-bit distance field and a 11-bit edge field as shown in figure 9.

Figure 9. Song & Perrig’s packet marking in IP header.
The algorithm for marking and reconstruction is shown in figure 10.

**Marking procedure at router $R_i$:**
for each packet $P$
  let $u$ be a random number from $[0, 1)$
  if $u \leq q$ then
    $P$.distance $\leftarrow 0$
    $P$.edge $\leftarrow h(R_i)$
  else
    if ($P$.distance $== 0$) then
      $P$.edge $\leftarrow P$.edge $\oplus h'(R_i)$
    $P$.distance $\leftarrow P$.distance $+ 1$

**Reconstruction procedure at victim $v$:**
let $S_d$ be empty for $0 \leq d \leq \text{max}_d$
for each child $R$ of $v$ in $G_m$
  if $h(R) \in \Psi_d$ then
    insert $R$ into $S_0$
  for $d = 0$ to $\text{max}_d - 1$
    for each $y$ in $S_d$
      for each $x$ in $\Psi_{d+1}$
        $z = x \oplus h'(y)$
        for each child $u$ of $y$ in $G_m$
          if $h(u) = z$ then
            insert $u$ into $S_{d+1}$
  output $S_d$ for $0 \leq d \leq \text{max}_d$

Figure 10. Song & Perrig’s Marking and Reconstruction Algorithm.

Every router marks a packet with a certain fixed probability when forwarding the packet. If a router decides to mark the packet, it writes hash of its IP address into the edge field and 0 into the distance field in packet. Otherwise, if the distance field is 0, which implies its previous router has marked the packet, it XORs with the edge field value and overwrites the edge field with the result of the XOR. The router always increments the distance field if it decides not to mark the packet. The XOR of two neighboring routers encode the edge between the two routers of the upstream router map. The edge field of the marking will contain the XOR result of two neighboring routers, except for samples.
from routers one hop away from the victim. Because we could start with markings from routers one hop away from the victim, and then hop-by-hop, decode the previous routers. The reason for using two independent hash functions is to distinguish the order of the two routers in the XOR result.

The reconstruction works as follows. The idea is to perform a breadth-first search of the search space defined by our upstream router map, which is already available. We construct an attack tree, beginning with our victim as the root node. For the first iteration, the edge field simply represents the hash value of an upstream neighbor, rather than the XOR of two addresses’ hash values. Thus, each address having a hash value matching an edge field for a distance-zero packet will have their address added to the attack tree. At each subsequent level, the hash values of the addresses located at the prior level are XOR’d with the edge field of the packets marked. This value, being a recuperated hash value of an address, is then used to search amongst the hash values of the addresses located at this level. When a match is found, we add this address to our attack tree. This process continues until either all valid distance levels are exhausted, or a iteration fails to yield any new attack nodes to be added to our attack tree. Figure 11 shows the reconstruction at work.
Figure 11. XOR Encoding.

The efficiency of probabilistic packet marking depends on one main factor – its marking probability. The techniques discussed above consist of a static probability marking. If $P$ is the marking probability at router $R$ then $A$, the probability of receiving the marked packet from router $R$ is given by $A_d = P(1-P)^{d-1}$, where $d$ is the distance between the source and destination. This technique causes the marked packets from distant routers to be remarked by downstream routers. This leads to loss of information from distant routers and increase in number of packets to reconstruct the path. Research has shown that when the marking probability is set to $1/d$, where $d$ is distance between the attacker and victim, the probabilistic packet marking provides optimal results [1]. However the distance ‘$d$’ is not known in advance and it is not practically applicable idea.

To overcome this problem, Tao Peng et al. [3] proposed a new adjusted marking scheme to increase the probability of receiving packets from distant routers. In this technique, the marking probability at every router is computed by using the distance from itself to the destination. The marking probability is computed using the formula $1/(31-d)$ where $d$ is the distance from the current router to the victim. The drawback with this scheme is that the router is dependent on the underlying protocol to compute the distance...
from itself to the victim. This is a router overhead, which considerably slows down the packet marking.
A key issue with PPM schemes is the use and reliance on a fixed marking probability. As each router indiscriminately decides to mark incoming packets, marked packets by distant routers are more likely to be remarked by downstream routers. This loss of information from distant routers leads to increase in number of packets needed to reconstruct the attack path. Enabling each router to adjust its marking probability so as to obtain equal samples of marked packets, in particular from the furthest routers, would help in minimizing the time to reconstruct the attack path. In the following sections, we discuss and describe our approach in designing our scheme to adjust the marking probability in a dynamic fashion.

3.1 Design

The proposed dynamic marking scheme has been developed based on different perspectives. Our main objective is to minimize the time required to reconstruct the attack path, which means obtaining a complete array of marked packets from each router along the attack path as early as possible. This in turn depends on the choice for the value set for the marking probability $p$. Too large a value for $p$ would cause excessive packet
remarking by downstream routers and too low would cause an increase in the number of packets needed to reconstruct the path. Another main limitation of PPM is that the number of packets needed for reconstruction grows exponentially with path length. Research has shown that a marking probability set to the value $1/d$ where $d$ is the path length gives the best performance. This suggests that a router should adjust $p$ based on its position in the attack path. But it is not possible in practice to determine the distance between attack source and victim beforehand.

Tao Peng et al. [3] suggested estimating the path length based on available information and proposed a technique to dynamically adjust and derive the marking probability from the distance from the current router to the destination node. However, this technique would involve the underlying Internet protocols to obtain the distance to the destination router, which means additional router overhead.

Our proposed scheme follows a similar approach without the mentioned shortcomings. One of the variable fields in the IP header that can be correlated with the dynamic probability is the 8-bit Time-To-Live (TTL) field. It is used to keep track of the number of hops traveled and to limit the number of hops a packet can travel in the Internet thereby preventing packets to be indefinitely forwarded by routers. The TTL value is decremented by one at every hop during the routing process. It has been observed that in today’s Internet, a maximum number of 32 hops is in practice sufficient for packets to reach their destination. So, in general the TTL value is initialized to the value 32 when the IP header is being constructed, even though it can be set to any number between 1 and 255. The TTL field provides a good estimate on how far the packet has
traveled and can be correlated to the marking probability. We derive the marking probability $p$ as:

$$p = 0.51 - \left( \frac{1}{TTL} \right)$$

The constant value is chosen as 0.51 because the TTL value is assumed to range from 1 through 32. In the special case where TTL is 1, $p$ is set to the value 0.01.

Our proposed scheme is described below. A random number is generated between 0 and 1. If the generated number is less than or equal to the calculated marking probability then the packet is set for marking by the router. Else, the packet’s distance value is incremented by 1, in the same way as in the Advanced Marking Scheme I. The marking procedure of our TTL based scheme is shown in figure 12.

```
Marking procedure at router R
for each packet P
{
    TTLV: = get TTL value in IP header
    if TTLV == 1
    {
        q:= 0.01
    }
    else
    {
        q:= 0.51 - ( 1/TTLV )
    }
    let u be a random number from [0,1)
    if ( u <= q )
    {
        P.distance: =0
        P.edge: = h(R)
    }
    else
    {
        if (P.distance==0)
        {
            P.edge: = P.edge XOR h(R)
        }
        P.distance: = P.distance + 1;
    }
}
```

Figure 12. Marking Procedure of TTL-based scheme.
3.2 Metrics

The efficiency of the traceback algorithm is measured by various factors. Three important factors that determine the efficiency of an IP traceback algorithm are the number of false positives, number of packets needed to reconstruct attack path and the reconstruction time. False positives refer to instances where routers that do not belong in the real attack graph but are present in the reconstructed attack graph at the victim side. The lesser the number of false positives generated, lesser the confusion in tracing back to the attacker and more accurate the result is. The number of packets needed for path reconstruction is the minimum number of packets needed to successfully reconstruct the attack path. Lesser the number of packets needed, the quicker we can trace back to the attack source. The reconstruction time is the time taken to reconstruct the attack path on the victim side. The lesser the number of packets needed, the less time it takes to reconstruct the attack path.

3.3 Simulation

The simulation was setup and run on a Windows XP on a 2.4 Ghz Pentium 4 with 640 MB of RAM. The topological map of routers from a single point was obtained from the Internet Mapping Project [10] and was used for simulation. The Internet Mapping Project was started at Bell Labs in the summer of 1998. Its goal is to acquire and save
Internet topological data over a long period of time. This data has been used in the study of routing problems, DDoS attacks, and graph theory. The trial packets are from a host with 65.198.68.56 as the IP address. The list consisted of 135,629 different routes from that host. A simulator was developed in C# 2005 and used for our simulation experiments. The simulator is equipped with a graphical user interface and is user-friendly. All the parameters for the simulation can be set easily. The simulator is completely automated and autonomous. Once the initial parameters are set, the simulation runs, computes and displays the number of false positives, reconstruction time and number of paclets needed to reconstruct the attack path.

3.4 Implementation

Our objective was to find a technique to dynamically set the value of marking probability and obtain results similar to the results when the marking probability is set to $1/d$, where $d$ is the distance between the attacker and victim. It is expected that would minimize the number of packets to reconstruct the path. To find the least number of packets needed to successfully reconstruct the attack path we design a system where first the attack path of the desired length is chosen. Then the attack packets are sent to the victim as follows. To find the least number of packets needed to successfully reconstruct the path, we send increasing number of packets from the source to the victim starting from one to 10000. First we send the attack packets and perform the packet marking simulation. Then we collect the marked packets and begin to reconstruct the true attack path. If the reconstruction is successful then the simulation terminates. Otherwise the
number of packets sent is incremented by one and the simulation is repeated again. This will ensure that we find out the least number of packets needed. For every simulation that is run, the results are logged and averaged and presented to the user. The results included are average number of packets, average reconstruction time, and average false positives. We can also set the number of times the simulation needs to be done. We can completely set the lower bound and the upper bound and the increment for the number of packets to be sent.

The second objective is to find out the average false positives and the reconstruction time needed in a DDoS attack scenario. For this simulation, since it is going to be more than one number of attackers of varying attack path length, we choose the given number of attackers from a pool of all the attackers of varying attack path length randomly. Then 10000 packets are sent from every source to the victim and the routers in the attack path do the marking. When the simulation has been completed, we collect the marked packets at the victim and try to reconstruct the attack path. During which we find the total number of false positives and the reconstruction time. Then if the simulation is to be run more than once, it is set before the simulation. After the simulation is run, the average false positives and average reconstruction time is displayed. The trace route data for the simulation is obtained from [10].

The user is presented with a graphical user interface as shown in Figure 17, where the user sets the attack simulation setup. User can select the type of simulation he/she needs to run, to find least number of packets or to find the average false positives for DDoS attacks. Depending on the simulation, the interface displays the corresponding results after the simulation is completed.
The user can set the initial parameters of the simulation via the front end user interface. Once the initial parameters such as number of attackers and attack path length etc. are set, the user can continue with the simulation. The simulation is completely object oriented. Every inner system inside the whole simulation has been designed as classes and can be individually tweaked and modified without a lot of changes to the entire system. Everything part of the simulation is completely configurable and scalable and can be automated.
Our objective was to devise a new algorithm to independently and dynamically mark the packets to improve the PPM traceback performance. One of the metrics used to evaluate the performance of different marking schemes is the number of packets needed to reconstruct the attack path. The various marking schemes were compared based on the above factor, as well as the total number of false positives and reconstruction times. Extensive simulation were carried out to obtain each data point. The simulations and reconstructions were run under Windows XP on a 2.4 Ghz Pentium 4 with 640 MB of RAM.

In this chapter we present the results from the simulation comparing four different marking schemes.

1. Scheme 1: Fixed marking probability $p= 0.03$.
2. Scheme 2: Fixed marking probability $p = 1/d$, where $d$ = length of the attack path, which is known beforehand.
3. Scheme 3: Dynamic marking probability $p = 1/31-m$, where $m$ is the distance between the marking router and the victim. This corresponds to the formula derived by Tao Peng et al [19].
4. Our proposed TTL based marking scheme: Dynamic marking probability
\[ p = 0.51 - \frac{1}{\text{TTL}} \], where TTL is the Time To Live value present in the IP header.

4.1 Number of packets needed to reconstruct the attack path.

The graph comparing the attack path length and average number of packets needed is shown in Figure 14. In this graph, we also plotted the theoretical limit for the number of packets needed for successful reconstruction, calculated using the formula:

\[
\frac{\ln(d)}{q(1-q)^{(d-1)}} \quad [1].
\]

Figure 14. Comparison of average packets needed.
We observe that the performance of compared schemes converge closely when the attack path is approximately 20. For easier comparison of the schemes, we divide the attack path length into two sets. Set one ranges from one through twenty and set two ranges from twenty-one to thirty-two. Also, we compare the results from static and dynamic marking scheme perspectives.

In lower ranges of set one, we see that all the marking schemes perform the same. As the range increases we see that scheme 2 which is static probability-marking scheme and our TTL based dynamic probability-marking scheme performs the same and scheme 1 and scheme 3’s performance is low. As the range increases to higher end, there is a considerable difference in the performing of each scheme. The static probability-marking scheme 2 requires the least number of packets to reconstruct the attack path. But, it is not possible to know the attack path length in real-time scenarios. Our TTL based dynamic probability-marking scheme performs the best when compared to other schemes in mid ranges of set one. Scheme 1, which is static probability marking and scheme 3 which is dynamic probability-marking requires significantly more number of packets to reconstruct the attack path. In the higher ranges of set one, an interesting phenomenon occurs. Except for scheme 1, all other schemes converge to a similar performance index. All the schemes requires almost the same number of packets to reconstruct the attack path. This phenomenon cannot be technically attributed to any issues and need to be investigated further.

In lower ranges of set two, the trend continues the same as that of the earlier set. All the schemes have similar performance index. But, as the range increases to higher end, the performance index varies. Scheme 2 performs the best of all compared schemes.
But as said before, it is technically and practically not applicable to obtain the length of the attack path beforehand and use it for marking purposes. We see that, our TTL based scheme requires fewer number of packets than static probability-marking scheme 1 and dynamic probability-marking scheme 3 and thus performs best in the higher end of set two. Over all, we see that our TTL based dynamic marking-probability scheme performs better in all the attack length ranges.

4.2 Reconstruction time of the schemes compared.

The reconstruction times of the schemes are compared and the results are shown in the figure 15.

![Average Reconstruction Time](image)

Figure 15. Comparison of reconstruction time.
In the lower ranges of set one, we see that static-probability-marking scheme 2 and our TTL based dynamic-probability marking scheme has almost the same performance. Whereas, static probability-marking scheme 1 and dynamic probability marking scheme 3 needs considerably more time for reconstruction. As the range increases the static probability-marking scheme 2 requires the least amount of time for reconstruction. Our TTL based dynamic probability-marking scheme requires more time when compared to static probability-marking scheme 2, but still performs better than scheme 1 and scheme 3 which is static probability-marking and dynamic probability-marking respectively. As the range increases, the static probability-marking scheme 1 and dynamic marking probability-marking scheme 3 requires more time when compared to static marking-probability scheme 2 and our TTL based dynamic probability-marking scheme. As the range increases, the trend changes and a similar phenomenon occurs as that of previous comparison. All the marking schemes have a similar reconstruction time except that of static probability-marking scheme 1.

In lower ranges of set two, the performance of the schemes starts to vary and continues till the higher end. As the range increases we see that, dynamic probability-marking scheme 3 outperforms all other schemes requiring the least amount of time for reconstruction. The static probability-marking schemes 1 and 2 continually requires increasing amount of time for reconstruction in the range. Our TTL based dynamic marking-probability scheme requires significantly less amount of time when compared to dynamic marking-probability scheme 3 and performs better than static marking-
probability schemes 1 and 2. Our proposed dynamic marking-probability scheme continually performs better over the entire length of the attack path.

4.3 Performance in DDOS attacks:

Next we compare the number of false positives between our proposed scheme and marking probability of 0.03 scheme in a DDOS attack scenario.

![Average False Positives](image)

**Figure 16.** Comparison of average false positives in DDoS attacks

We see that our proposed TTL based dynamic marking-probability scheme performs comparable in a DDOS attack scenario when compared to the static probability-marking scheme of 0.03.

Next we compare the reconstruction times between our proposed scheme and the static 0.03 probability-marking scheme. Figure 17 shows the results of the comparison of reconstruction times in a DDOS attack scenario.
In comparing the reconstruction time, we see the reconstruction time for our TTL based dynamic probability-marking scheme is much better when compared to the static probability-marking scheme 1.

In summary, the marking schemes with a fixed probability will result in a small number of packets marked by distant routers at the victim. In contrast, we have come up with a new dynamic marking probability that solves this problem by adjusting the marking probability at each router, which significantly reduces the number of packets needed to reconstruct the attack path. Furthermore our proposed scheme does not rely on the underlying protocols or the routing table to compute the marking probability. Instead we use the readily available TTL field in the IP header to compute the marking probability. This proves to be more effective and has fewer overheads on the router.
CHAPTER V

CONCLUSION AND FUTUREWORK

Packet marking techniques are one of the most important factors in a successful IP traceback approach. A good packet marking technique will reduce the number of packets needed to do a successful traceback and reduce the time taken to find out the true source of the attacker. Packet marking techniques fall into two categories, static probability-marking and dynamic probability-marking scheme. The problem with fixed probability-marking in PPM is that, we do not receive enough marked packets from distant routers, which lead to many serious issues in IP traceback and increase the reconstruction time at the victim. Our objective in developing a new dynamic probability-marking scheme was to get enough number of marked packets from distant routers in a short period of time that will help in quicker and more reliable traceback. In this work, we proposed a new dynamic marking scheme, as an enhancement to PPM scheme, based on the 8-bit TTL (Time To Live) value present in the IP header. Our objective was to come up with a new marking scheme based on dynamic probability and without the drawbacks of the fixed probability and which performs comparably to the existing marking techniques. We calculate the marking probability for every packet using the formula $0.51 - \frac{1}{\text{TTL}}$, where TTL is the value obtained from the IP header of the packet. The main advantage of our scheme is that it does not rely on underlying protocols or
routing table information from the router and the probability can be altered and fine-tuned by varying the constant in the formula for every particular need. By varying the 0.51 value in our proposed scheme, the efficiency of the scheme can be fine-tuned for more accuracy. Our approach is simple, straightforward and easy to implement in a large scale environment. To verify the efficiency of our proposed technique, we applied our technique to Advanced Marking Scheme I proposed by Song & Perrig [1] and ran simulations and compared with other marking schemes. In contrast to previous work, our marking scheme has higher precision and low computing overhead for the router and for the reconstruction process at the victim. Our scheme requires significantly less number of packets to completely reconstruct the attack path than other marking schemes and also has very low reconstruction time at the victim to compute the attack path. This helps network administrators to take quick actions in safeguarding the network from the attacks and also in find out the origin of the attacker. Also, our proposed scheme was tested with DoS and DDoS attack simulation and demonstrated to be more effective than other schemes.

The future works that can be pursued on this work include implementing the same marking scheme for Advanced Marking Scheme II proposed by Song & Perrig. Applying the scheme to AMS II would reduce the total number of false positives to a significant extent and would improve the IP Traceback. Also, the same technique can be applied for IP version 6 to analyze its performance.
REFERENCES


APPENDIX

Using the software:

The IP traceback simulator was developed to help us in simulating the various traceback approaches. The simulator was developed in C# 2005 with Visual Studio 2005. The figure below shows the screen shot of the application.

![IP Traceback Simulation](image.jpg)

Figure 18. Screen shot of the simulator.
The software developed is intended to be simple and user friendly. It primarily has two sections. The first is to simulate a DDOS attack scenario, and the second is to simulate a single attacker scenario.

To simulate a DDOS attack scenario, the user selects the section on the left with a radio button labeled “Simulation to Find Average False Positives and Average Reconstruction time”. After the radio button has been checked, the user sets the numbers of attackers to be simulated and the number of times the simulation needs to be run. More runs will result in more accuracy. The user then clicks the “Load Data” button to load the data needed to simulate the network from the disk and prepares for the simulation. Upon load completion, the user can click on the button labeled “Simulate” to start the simulation process. After the simulation has been completed the results are displayed. The results displayed include average total time for simulation, average time taken for reconstruction alone, and average false positives generated.

To simulate a single attacker scenario, the user checks the radio button labeled “Simulation to find average number of packets needed”. The user can then set the attack path distance. The user also has the choice to set the minimum and maximum number of packets to send to the victim. These options are set to 1 and 10000 as the default values. The user can also set the increment value. This offers a greater flexibility and control of the simulation by the user. Then the user clicks the “Load Data” button to load the data and prepare it for the simulation. Upon load completion,
the user can click on the button labeled “Simulate” to start the simulation process. After the simulation has been completed the results are displayed. The results displayed include, average total time for simulation, average time taken for reconstruction alone, and average false positives generated and average packets needed to reconstruct the attack path.