DENIAL OF SERVICE ATTACKS:
PATH RECONSTRUCTION FOR IP TRACEBACK USING ADJUSTED PROBABILISTIC PACKET MARKING

A Thesis

by

RAGHAV DUBE

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

December 2004

Major Subject: Electrical Engineering

DENIAL OF SERVICE ATTACKS:

PATH RECONSTRUCTION FOR IP TRACEBACK USING ADJUSTED PROBABILISTIC PACKET MARKING

A Thesis

by

RAGHAV DUBE

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Approved as to style and content by:

Deepa Kundur (Chair of Committee)	
A. L. Narasimha Reddy (Member)	Scott L. Miller (Member)
Joobin Choobineh (Member)	Chanan Singh (Head of Department)

December 2004

Major Subject: Electrical Engineering

ABSTRACT

Denial of Service Attacks:

Path Reconstruction for IP Traceback Using Adjusted Probabilistic Packet Marking. (December 2004)

Raghav Dube, B.E., Motilal Nehru Regional Engineering College, Allahabad, India

Chair of Advisory Committee: Dr. Deepa Kundur

The use of Internet has revolutionized the way information is exchanged, changed business paradigms and put mission critical and sensitive systems online. Any disruption of this connectivity and the plethora of services provided results in significant damages to everyone involved. Denial of Service (DoS) attacks are becoming increasingly common and are the cause of lost time and revenue.

Flooding type DoS attacks use spoofed IP addresses to disguise the attackers. This makes identification of the attackers extremely difficult. This work proposes a new scheme that allows the victim of a DoS attack to identify the correct origin of the malicious traffic. The suggested mechanism requires routers to mark packets using adjusted probabilistic marking. This results in a lower number of packet-markings required to identify the traffic source. Unlike many related works, we use the existing IPv4 header structure to incorporate these markings. We simulate and test our algorithms using real Internet trace data to show that our technique is fast, and works successfully for a large number of distributed attackers.

To my Parents

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my advisor, Dr. Deepa Kundur, without her guidance this work would not have been accomplished. I am greatly indebted to her for providing me with advice and encouragement during the course of my research. The calm with which she handled the obstacles we faced during the research helped in keeping my enthusiasm high and the research focussed. She was always open to new ideas and I appreciate the freedom she gave me while working on my research.

I thank Dr. Reddy, Dr. Miller and Dr. Choobineh for taking interest in my research and providing invaluable suggestions.

My wholehearted gratitude to my parents for their immeasurable support throughout my life. Without their unconditional love, encouragement and support, I could never have come this far.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Denial of Service (DoS) attacks are increasingly becoming a security threat and nuisance for the Internet community, especially online businesses and mission-critical systems. These attacks result in system downtime, lost revenues, and the physical labor involved in identifying and recovering from such attacks. Denial of service (DoS) attacks consume the resources of a host or a network. The host or network is thereby unable to provide the expected service to legitimate users. Increased availability of Internet access and high-end computer systems has made DoS attacks more severe and easy to execute. With many "over-the-counter" tools being available for creating trojans, viruses and back-doors, it is getting increasingly simple, even for a person not having an in-depth knowledge of computer systems, to launch DoS attacks. According to the 2004 Computer Security Institute/FBI Computer Crime and Security survey [1], DoS attacks were responsible for more than $26 million in total losses among those surveyed. Recent denial of service attacks on web sites operated by Microsoft Corporation, Yahoo! Inc., and Google [2], [3], [4] are perfect examples of how these attacks affect businesses as well as customers accessing their services.

A. Denial of Service Attacks

A Denial of Service attack is a process of blocking access to data or systems wherein a user or organization is deprived of resources they would normally expect to have.

Denial of Service Attacks may be classified into two basic types:

1. *Logic or Software Attacks*: These attacks exploit software bugs at the targeted

---

The journal model is *IEEE/ACM Transactions on Networking*.

system. Only a few packets are required to disrupt the normal functioning of the victim of the attack.

2. *Flood Attacks*: The attacker directs a large volume of traffic to the victim. This high-volume traffic overwhelms the victim's resources, or may simply take up network bandwidth.

Henceforth, the system that initiates a DoS attack shall be referred to as the "attacker", and the system that is the target of the attack shall be referred to as the "victim". Let us now look at some logic- and flood-type DoS attacks.
Examples of logic-type DoS attacks are:

- *Ping of Death*: The attacker sends an oversized IP packet [5], [6] to the targeted system. The legal size limit for IP packets is 65,535 bytes. The attacker fragments an oversized IP packet and sends them to the victim. When the fragments are reassembled by the victim into a complete packet, it overflows the buffer on some systems, causing a reboot or hang.

- *Teardrop Attack*: The attacker sends IP fragments that cannot be reassembled properly. This causes the victim to reboot or hang.

Examples of flood-type DoS attacks are:

- *TCP SYN Flood Attack* [7]: This attack exploits the 3-way handshake used for a TCP [8] connection setup. The attacker sends a TCP SYN request to the victim using a spoofed IP address. The victim responds with a TCP SYN-ACK response and allocates memory for the potential connection. It waits for an acknowledgement (TCP ACK) from the attacker. However, since the IP address is spoofed, the victim does not receive a response from the attacker.

After a certain period of time, called the time-out interval, the victim closes the half-open connection and frees up the reserved memory. Since the memory resources at the victim are limited, if the attacker sends enough connection requests, and fast enough, it can tie-up the resources of the victim. Thus, connections from legitimate users cannot be processed.

- *UDP Flood Attack*: UDP [9] does not require any connection setup procedure to transfer data. An attacker sends UDP packets to random ports on the victim system. Since there have been no service requests from these ports, the victim has effectively wasted CPU cycles and memory resources to process these packets. Large number of such packets tie-up the victim's resources. And since UDP does not have congestion control, this attack can also be used to target the bandwidth resources of the victim's network.

- *Smurf IP Attack*: Forged ICMP [10] packets are sent by the attacker to the broadcast address of a network. All the systems on the network then send an ICMP reply back to the victim. This large volume of replies inundates the victim's bandwidth.

- *Mail Bomb*: A mail server can fail if a very large number of bogus emails are sent in a very short time.

The focus of this research is on flood-type DoS attacks.

B. IP Traceback

As seen in the previous section, an attacker may employ different techniques to flood a victim's resources and to avoid detection. An easy way to avoid detection is to use spoofed IP addresses in the packets that are sent.

Several mechanisms have been proposed to deal with the problem of Denial of Service Attacks. While some of these mechanisms focus on detecting a DoS attack and filtering the malicious traffic, others focus on identifying the source of the attack itself. These methods may be used in conjunction with each other to provide a comprehensive defense mechanism against DoS attacks.

Traffic in a DoS attack usually uses spoofed IP addresses. This makes it difficult to determine the source of the traffic. Hence, it becomes imperative that techniques to accurately determine the real source of the attack traffic, or the path traversed by the packets be developed and used to mitigate the DoS attack. This work uses IP traceback to identify the source of the DoS attack. IP traceback is used to determine the path followed by a packet, that is, trace a packet back to its source (or attacker).

Figure 1 shows a DoS attack scenario. An attacker sends high-volume traffic to its target. The packets comprising the attack are forwarded by the routers to their destination. An IP traceback mechanism enables the victim to know the about the routers that are forwarding the packets, and hence trace the packets' path back to its source, the attacker.

An IP header [5], shown in Figure 2, is used by routers to route a packet. Since all the routers along the path from the source to the destination of a packet process this header, it becomes an obvious choice for determining the path traversed by the packet. This header can be used to send information useful for traceback. Each router along a packet's path marks the packet with some unique information. The destination system (or victim) uses this information to reconstruct the path, and hence determine the real source of the DoS attack.

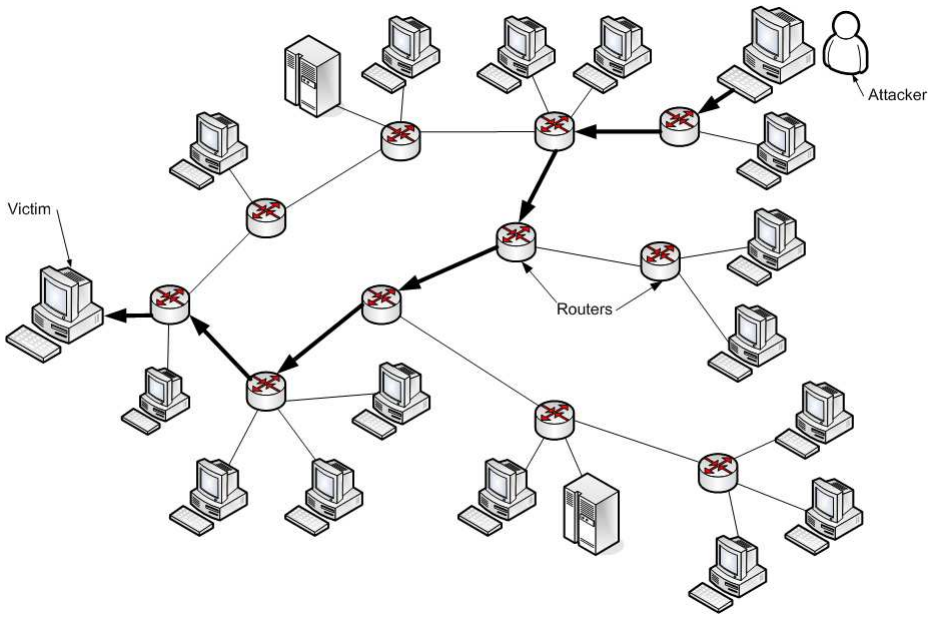

Fig. 1. A Simple Denial of Service(DoS) Attack Scenario.

C. Contribution of this Work

We propose a new traceback algorithm that employs adjusted router marking probability. No modification or addition to the IP header is required. The existing IP identification field is used to mark the packets. We show mathematically and through simulations that our packet marking technique requires a lower number of packets for reconstruction.

The victim uses existing tools to obtain a map of the Internet and reconstruct the attack path using the map and the packet markings. For a high-volume DoS, this means faster identification of the sources of the traffic.

D. Organization of Thesis

Chapter I provided a brief introduction to Denial of Service (DoS) attacks and their types. This chapter also described the contribution of this research.

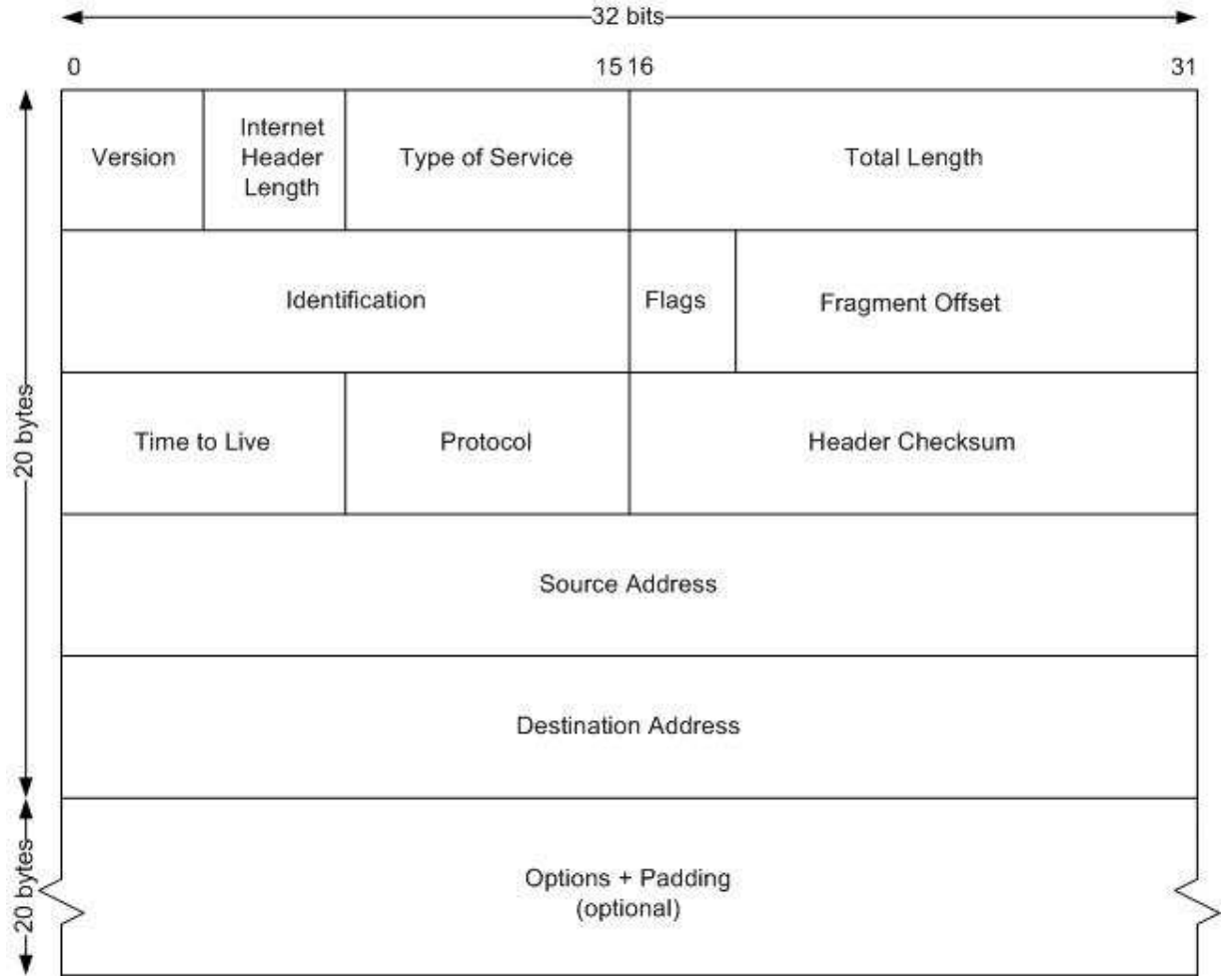

Fig. 2. IPv4 Header.

The rest of the thesis is organized as follows. Chapter II throws light on some of the related work done by researchers in the area of IP traceback and discusses the advantages and disadvantages of these methods. The proposed algorithm and a detailed mathematical analysis is provided in Chapter III. Chapter IV presents the simulation methodology, results and the interpretation of the findings. Chapter V concludes this thesis and discusses the future scope and applications of this work.

CHAPTER II

CURRENT RESEARCH AND RELATED WORK

Since the goal of this work is identifying the source of DoS attacks using IP traceback, and not filtering traffic, we shall discuss related work in this area in detail. However, for the sake of completeness, some DoS traffic filtering algorithms are also mentioned.

Network Ingress [11] and Egress [12] Filtering are preemptive measures to stop DoS attacks that use spoofed IP addresses. An Internet Service Provider's (ISP) aggregation point filters out traffic that does not belong to its network. This method is not effective against DoS attacks that use legitimate IP addresses, or zombie machines.

Gil and Poletto [13] have proposed MULTOPS, a data structure that can be used to detect flooding type DoS attacks. The algorithm uses the fact that during a DoS attack, there is a considerable difference between the traffic rates going to, and coming from the victim of the attack.

Yaar, Perrig and Song [14] have developed a Path Identifier algorithm that can be used to filter DoS attack traffic based on unique path identifying markings in the packet. Their filtering scheme is based on routers marking packets as they route them, creating a unique path-fingerprint that the victim can then use to filter traffic. They propose that routers mark the last n bits of their IP address into one of the $\lfloor 16/n \rfloor$ sections of the IP identification field. The packet's TTL value modulo $\lfloor 16/n \rfloor$ is used to determine which section to mark. They also propose an edge marking to increase the accuracy of their scheme. However, the algorithm proposed by Yaar et al., although uses router marking, is a DoS attack traffic detection and filtering mechanism, rather than a DoS attack source identification algorithm.

IP traceback has been discussed in Section B of Chapter I. The following schemes employ some form of an IP traceback mechanism to determine the source of the DoS

attack.

Doeppner, Klein and Koyfman [15] introduce a deterministic router stamping algorithm wherein each router that routes a packet appends its IP address to a new, variable length field in the IP header. Clearly, this is not a practical solution as this would mean that for a large path length, the IP header would expand by 120 bytes. To overcome this problem, the authors describe a probabilistic marking algorithm that requires routers to mark packets with a stamp that consists of the router's IP address and the interface on which the packet was received. Routers mark a packet with probability p, and place the marking into one of s slots in the IP header. The parameters p and s are constant. For each packet, the router calculates a random number, x between 0 and 1. If $x \leq s \cdot p$, it marks the packet in slot $\lfloor x/p \rfloor$, otherwise it just routes the packet without marking.

The probability that the victim receives a packet having a marking from a router k hops away is given by $s \cdot p \cdot (1-p)^{(k-1)}$. As the path length increases, the probability of receiving a packet marked by the farthest router decreases rapidly. The authors propose increasing the IP header size by 40 bytes for the probabilistic marking algorithm.

Although the authors describe a source identification algorithm, they do not present results showing its performance, simulated or otherwise.

Savage, Wetherall, Karlin and Anderson [16] propose a basic packet marking algorithm in which each router along a path decides to mark a packet with a probability p. The routers mark the packets with their IP addresses. The victim receives a packet marked by a router if that router marks the packet and no other router along the path overwrites the marking. The probability that the victim receives a packet marked by a router d hops away is thus equal to $p \cdot (1-p)^{(d-1)}$.

We observe that, as the distance of the router from the victim decreases, the

fraction of packets received by the victim that is marked by that router decreases very rapidly. For a marking probability, $p = 0.51$, and a router 15 hops away, this value is $2.3459\text{x}10^{-5}$. This means that 1 out of every $42,627$ packets received by the victim has a marking by the router 15 hops away.

To overcome this problem, the authors propose other algorithms that encode fragments of edge information in the packets rather than information about individual nodes. An edge is a bit pattern that contains information about both, the current router, and also the router from which the packet was received, thus describing an "edge". Each router writes, with marking probability p, onto either the start or the end field and also updates the distance information in the packet. The number of packets, X, required by the victim to reconstruct the path is bounded by

$$E[X] < \frac{k \cdot \ln(k \cdot d)}{p \cdot (1-p)^{d-1}}$$

where k is the number of fragments per edge and d is the distance of the attacker from the victim. If $k = 8$ fragments per edge, an attacker is $d = 10$ hops away, and $p = 1/25$, then approximately 1,200 packets on an average are required by the victim to reconstruct the attack path.

Song and Perrig [17] use the IP identification field to encode edge information and a distance value. They propose several algorithms for tracing the path of a packet back to its source. Their Advanced Marking Schemes require routers to encode edge values into the IP identification field of a packet. The identification field is sub-divided into distance and edge fields. Each router convert's its IP address into a set of hash values. The packets are marked with a fixed probability, q. If a router decides to mark a packet, it writes a hash of its IP address into the edge field and 0 into the distance field. If the packet has already been marked by a previous router, it XORs the existing packet marking with a different hash of its IP address and overwrites the

packet marking. If the router decides not to mark a packet, it always increments the distance field. Thus, edge information is encoded into the packet.

A second algorithm proposes using a many sets of hash values and sub-dividing the IP identification field into an edge field, a flag field (to denote which set of hash values was used to mark the packet), and a distance field. The difference in marking here is that the hash value to be stamped in the packet is also chosen at random.

The authors also propose an authenticated marking scheme in which routers use time-released key chains. Packets arriving in different intervals of time are marked using different keys.

All the algorithms proposed by Song and Perrig [17] require the victim to use a map of the routers to reconstruct the path of the packet.

The traceback algorithms discussed so far require routers to mark packets using a fixed, pre-determined probability. This results in a very small probability of receiving packets having a marking of a router farther away from the victim. Thus, the victim has to receive a large number of packets before it gets a marking by the farthest router. Peng, Leckie and Ramamohanarao [18] propose a marking scheme wherein routers mark the packets with a adjusted probability. The packet marking probability is inversely proportional to the number of hops of the router from the source of the packet. The marking probability is calculated as,

$$p = \frac{1}{d} \tag{2.1}$$

where d is the number of hops of the router from the source of the packet.

When a router receives a packet, it decides to mark it with a probability computed using (2.1). If it decides to mark the packet, it writes its IP address into the edge field and zero into the distance field. If the distance field is non-zero, it combines its IP address with the marking in the packet. A router always increments the value in

the distance field.

As will be shown in the following sections, using (2.1), we get a uniform probability of receiving packets marked by any router along the attack path. In addition, the expected number of packets required by the victim for reconstruction of the attack path is considerably smaller than that required by the previous scheme. The authors propose to add an extra field in the IP option field. The path reconstruction algorithm is not discussed.

Table I provides a qualitative comparison of the above algorithms.

Table I. Qualitative Comparison and Summary of Existing Algorithms.

Author(s)	Marking probability	Marking complexity	Where marking inserted	Reconstruction accuracy	Reconstruction complexity	Map required
Yaar et al. [14]	1	Moderate	IP identification	N/A	N/A	No
Doeppner et al. [15]	Fixed	Low-Moderate	Add new field in IP header	High	Low	No
Savage et al. [16]	Fixed	High	IP identification	Moderate-High	Moderate-High	No
Song et al. [17]	Fixed	High	IP identification	Moderate-High	High	Yes
Peng et al. [18]	Adjusted	Moderate-High	Add new field in IP option field	Not discussed	Not discussed	Not discussed

CHAPTER III

PACKET MARKING AND PATH RECONSTRUCTION

In this chapter, we discuss the specific objective of this research. We describe the proposed algorithm in detail and analyze the proposed algorithm mathematically.

A. Problem Formulation

This research work aims at reconstructing the attack path of a Denial of Service Attack. The path is reconstructed using IP traceback that employs adjusted probabilistic packet marking.

The IP identification field is used to mark packets as they are routed. The routers mark packets with a hash of their IP address. The marking probability depends on how far the packet has travelled from its source. In other words, the marking probability is not fixed, but is *adjusted*. At the victim, these markings are matched against a map of the routers to reconstruct the attack path.

We assume that the routers themselves are not participating in the DoS attack. Router markings are thus considered to be authentic.

In the sections that follow, we discuss in depth the above traceback technique and provide a mathematical analysis of the various aspects of the algorithm.

B. IP Header

To choose a suitable field to send information useful for traceback, let us first take a look at the IP header [5], [6] shown in Figure 2.
The various fields in an IPv4 header as follows:

1. *Version*: The version of IP being used (currently 4).

2. *IP Header Length*: Number of 32-bit words forming the header.

3. *Type of Service*: Indicates Quality of Service requirements. Usually set to 0.

4. *Total Length*: Total length of header and data in bytes.

5. *Identification*: Used for reassembly of fragmented packets in case of fragmentation.

6. *Flags*: Three bits (one of the 4 bits is unused) used to control whether routers are allowed to fragment a packet and to indicate the parts of a packet to the receiver.

7. *Fragment Offset*: Distance in bytes from the start of the original packet, set by a router that performs fragmentation.

8. *Time To Live*: Number of hops that a packet may be routed over. Decremented at each hop.

9. *Protocol*: Indicates the type of transport packet being carried, e.g. TCP, UDP, ICMP, IGMP.

10. *Header Checksum*: 1's complement checksum inserted by the sender and updated whenever the packet header is modified by a router.

11. *Source Address*: IP address of the original source of the packet.

12. *Destination Address*:the IP address of the final destination of the packet.

13. *Options*: Provide for control functions needed or useful in some situations, not normally used.

The Options field in the IP header may be used to record the route of a packet. However, most routers drop the packet if *any* option is specified. Hence, it is not feasible to use the Options field to send traceback information.

The Identification field in the IP header is the most suitable field that can be used for our purpose. This field is normally used for reassembly of fragmented packets in case of fragmentation. However, it has been shown [19] that a very small percentage (less than 0.25%) of packets in the Internet are fragmented. Hence, for a vast majority of packets, the IP identification field is irrelevant. The 16-bit IP identification field can thus be used by routers to mark packets to send traceback information.

C. Packet Marking

The structure of the IP header has been discussed in the previous section and we saw that the IP identification field is suited for marking packets and sending traceback information to the victim of a DoS attack.

The IP identification field is 16 bits in length. For the purpose of traceback, we require to fit a router marking and a distance value in this 16-bit field. Theilmann and Rothermel [20] show that most paths in the Internet are 30 hops or less. Thus, five bits (representing distances up to 32 hops) will suffice for our purpose. The remaining 11 bits (giving a total of $2^{11} = 2048$ possible values) are to be used for router markings. Figure 3 shows the modified IP identification field in the IP header. The IP identification can now be divided into a distance field and a marking field.

An IP address is 32 bits in length. However, we only have 11 bits into which we can write the router marking. A hash function, $h(.)$, is used to map a router's 32-bit IP address to a 11-bit marking value. The hash function maps the IP address space uniformly over the 2^{11} possible router markings. This function is a statistically

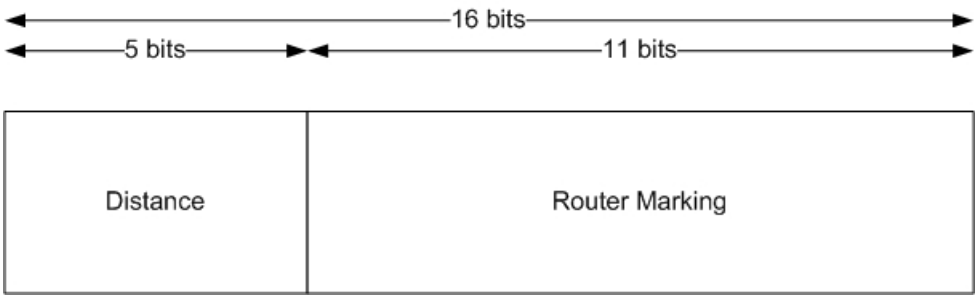

Fig. 3. The Modified IP Identification Field.

good random function. This means that for any input of an IP address, all of the $2^{11} = 2048$ markings are equally likely as an output. Also, each router has a marking which is independent of the markings of all other routers. It is also assumed that the routers are not compromised, and hence, their markings are trusted.

1. Packet Marking Probability

As mentioned in Chapter II, Peng, Leckie and Ramamohanarao [18] propose a marking probability, $p_d = 1/d$, where d is the distance (number of hops) of the router from the source of the packet.

Assume an attack path of length k. By this we mean that there are k routers participating in the marking scheme between the source and the destination. Hereafter, when we say that a path is of length k, or that the attacker is k hops away from the victim, we mean that there are k routers present in the path.

Consider the marking probability assignment,

$$p_d = \frac{1}{d - 1 + c} \tag{3.1}$$

where $d-1$ is the value in the distance field of the packet received by a router d hops away from the attack source, and $c \geq 1, c \in \mathbb{R}$. Later in this chapter we shall look

at the packet marking algorithm. We will see that for a router that receives a packet with distance field value equal to zero, we need to make the marking probability less than or equal to 1. Hence, $c \geq 1$. This parameter also controls the fraction of packets that are marked by routers.

Let α_d be the probability that the victim receives a packet marked by a router d hops away from the attacker. Then,

$$\alpha_d = p_d \cdot \prod_{i=d+1}^{k} (1 - p_i) \tag{3.2}$$

Using (3.1),

$$\begin{aligned}
\alpha_d &= \left(\frac{1}{d-1+c}\right) \cdot \prod_{i=d+1}^{k} \left(1 - \frac{1}{i-1+c}\right) \\
&= \left(\frac{1}{d-1+c}\right) \cdot \left(1 - \frac{1}{d+c}\right) \cdot \left(1 - \frac{1}{d+1+c}\right) \cdot \ldots \\
&\quad \ldots \cdot \left(1 - \frac{1}{k-3+c}\right) \cdot \left(1 - \frac{1}{k-2+c}\right) \cdot \left(1 - \frac{1}{k-1+c}\right) \\
&= \left(\frac{1}{d-1+c}\right) \cdot \left(\frac{d-1+c}{d+c}\right) \cdot \left(\frac{d+c}{d+1+c}\right) \cdot \ldots \\
&\quad \ldots \cdot \left(\frac{k-4+c}{k-3+c}\right) \cdot \left(\frac{k-3+c}{k-2+c}\right) \cdot \left(\frac{k-2+c}{k-1+c}\right)
\end{aligned}$$

$$\Rightarrow \qquad \alpha_d = \left(\frac{1}{k-1+c}\right) \tag{3.3}$$

Thus, we see that the probability of receiving a packet marked by any router along the attack path depends on the length of the path, not the position of the router along the path. Also, this probability is equal for all the routers along a path.

2. Packet Marking Algorithm

Each router does a one-time calculation of its IP address hash and h(IP address), while the distance values and their corresponding probabilities are calculated us-

ing (3.1).

A router along a packet's path reads the distance value in its IP identification field. The router then looks up a table containing distance values and the corresponding marking probabilities. Using this probability, it decides whether or not to mark the packet. This decision is made as follows - the router generates a random number. If this random number is less than or equal to the marking probability, the packet is marked, otherwise not. If the router decides to mark the packet, it writes its marking, h(IP address), in the IP identification field. The distance value in the IP identification field is then incremented and the packet is routed.

If, however, the router decides not to mark the packet, it always increments the distance value in the IP identification field and routes the packet.

The packet marking algorithm is shown in Table II.

Table II. Packet Marking Algorithm

$m = h$(IP address)

for each packet

read d =distance field value

generate a random number $x \in [0, 1)$

p = marking probability corresponding to d, looked up from table

if $x \leq p$ (if packet is to be marked)

write m into the marking field

distance field value= $d + 1$

The random number generated need not be cryptographically secure. It is a statistically good random number and may be evaluated using time seed values in simple random number generators.

We should keep in mind that a router at a distance d from the attacker receives packets with a distance field value equal to $d - 1$. Figure 4 illustrates this point. Each router increments the distance field value in the packet, irrespective of whether it marks the it or not. Consequently, for a given path, all the packets received by the victim have a distance field value that is equal to the path length.

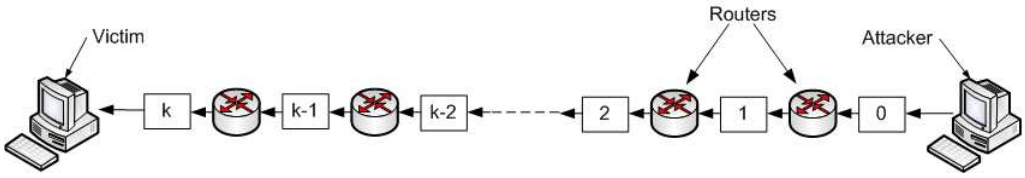

Fig. 4. Distance Field Values in a Packet for a Path of Length k.

D. Attack Path Reconstruction

1. Overview

To reconstruct the path of a packet and identify the source of the attack, the victim requires a map of the routers. The victim matches packet markings with the routers on the map and can thus reconstruct the attack path. Obtaining or constructing this map is not difficult. A number of tools [21], [22] and [23] are available that can be used to obtain a map of the the routers and the Internet. This map is in the form of a directed acyclic graph, $\mathbb{G}$. Readers may refer to Table III for the nomenclature used in the following sections. The root of $\mathbb{G}$ is the victim. All other nodes in $\mathbb{G}$ are routers. As shown in Figure 5, for each router, y, in $\mathbb{G}$, denote the set of children of that router by ρ_y.

During a DoS attack, the victim receives a large number of router markings. Before we can reconstruct the attack path using these markings, we need to group

Table III. Nomenclature

k:	Attack path length
$h(.)$:	Hash function that takes a 32-bit IP address as an input and produces a 11-bit random output
p_i:	Marking probability of router
α_d:	Probability that the victim receives a packet marked by a router d hops away from the attacker
n_k:	Number of attackers k hops away from the victim in case of a DDoS attack
μ:	Set of various distance field values of packets received by the victim
λ_k:	Set of markings received by the victim with a distance value $k \in \mu$
$\mathbb{G}$:	The directed acyclic graph that represents the map of the routers
ρ_y:	Set of children of a router y in the graph, $\mathbb{G}$
N:	Number of possible markings $= 2^{11} = 2048$
$\mathcal{U}_{N,k}$:	Expected number of unique markings when k drawings are made with replacement from a set of N different markings
m_{ρ_y,λ_k}:	Set of routers in ρ_y that should also be in the set λ_k, or, set of routers in ρ_y that are actually present in the attack paths
S_d:	Nodes at level d in the graph, $\mathbb{G}$, that are part of the attack graph

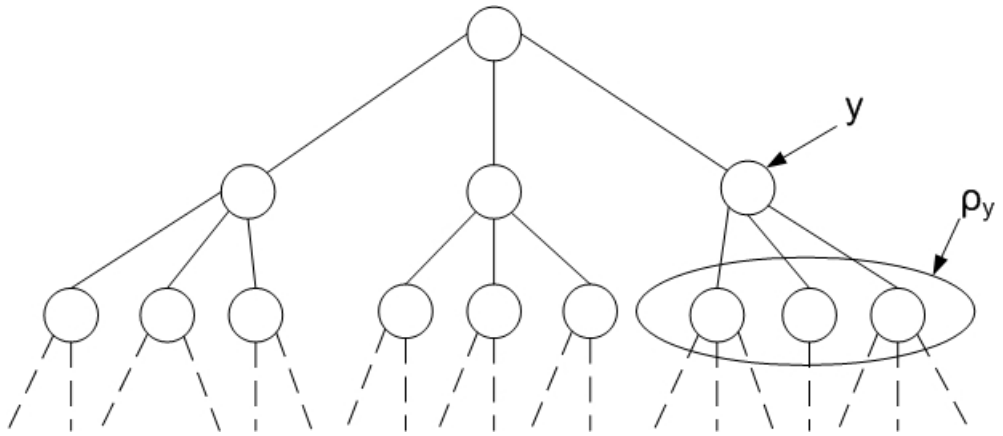

Fig. 5. y and ρ_y in the Graph, $\mathbb{G}$.

the markings based on attack path lengths.

As stated in the previous section, for a single attacker, all the packets that a victim receives have a distance field equal to the path length. Now, if there are multiple attackers at the same distance from the victim (may have different routes), the victim still receives all the packets with distance field containing the same attack path length. Thus, at the victim, there will be a set of markings, each with the same distance field value.

Consider the case when there are n multiple attackers. This now becomes a Distributed Denial of Service Attack(DDoS) scenario shown in Figure 6. Some attackers may be at the same, and some at different distances from the victim. In this case, there will be different sets of markings at the victim, each set containing markings from attackers at the same distance from the victim. Let these different distance values of packets in the sets of markings be elements of the set μ. It follows that the victim now has $\mid \mu \mid$ different sets of markings, each set corresponding to markings on packets sent by attackers that are at the same distance away from the victim. The distance field value ranges from $0 \leq k \leq 31$. Let the set of markings received by the

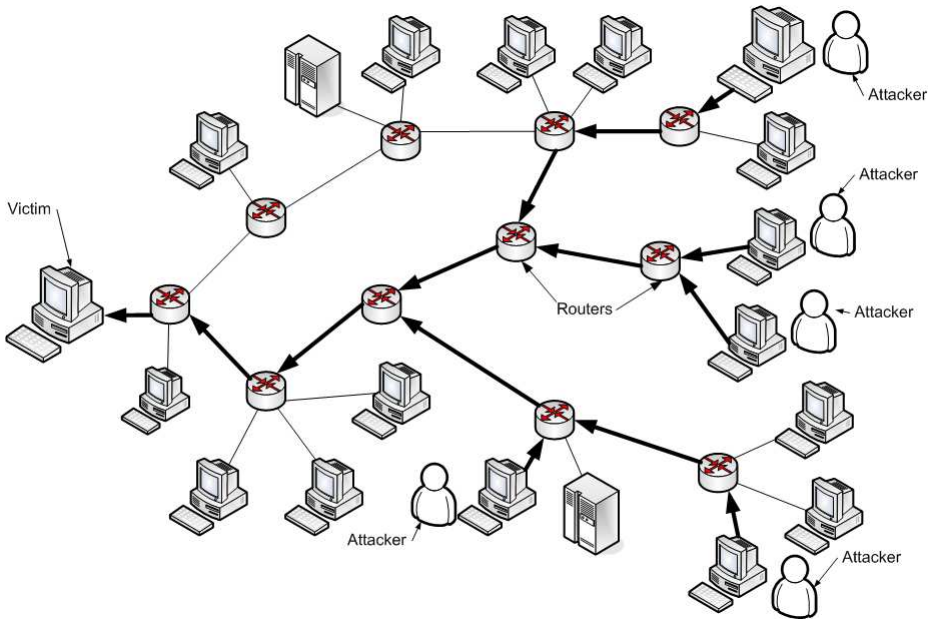

Fig. 6. A Distributed Denial of Service(DDoS) Attack Scenario.

victim with a distance value $k \in \mu$ be denoted by λ_k. Denote the number of attackers at a distance k hops away by n_k. Thus, it follows that

$$\lambda_k = n_k \cdot k \tag{3.4}$$

As an illustration, let us take a look at a scenario where there are 10 attackers. Of these, 3 attackers are at a distance 15 from the victim, 2 attackers are at a distance 19, 1 attacker is at a distance 23 and 4 attackers are at a distance of 30 hops from the victim. In this case, $\mu = \{15, 19, 23, 30\}$; $n = 10$; $n_{15} = 3$, $n_{19} = 2$, $n_{23} = 1$ and $n_{30} = 4$. This scenario is shown in Figure 7 (the number of packets shown in each set are for illustration purposes only and are not equal to the value in (3.4)).

2. Path Reconstruction Algorithm

The path reconstruction algorithm is shown in Table IV. The graph, $\mathbb{G}$, is traversed for each set of packets having the the same distance field value (for each set λ_k,

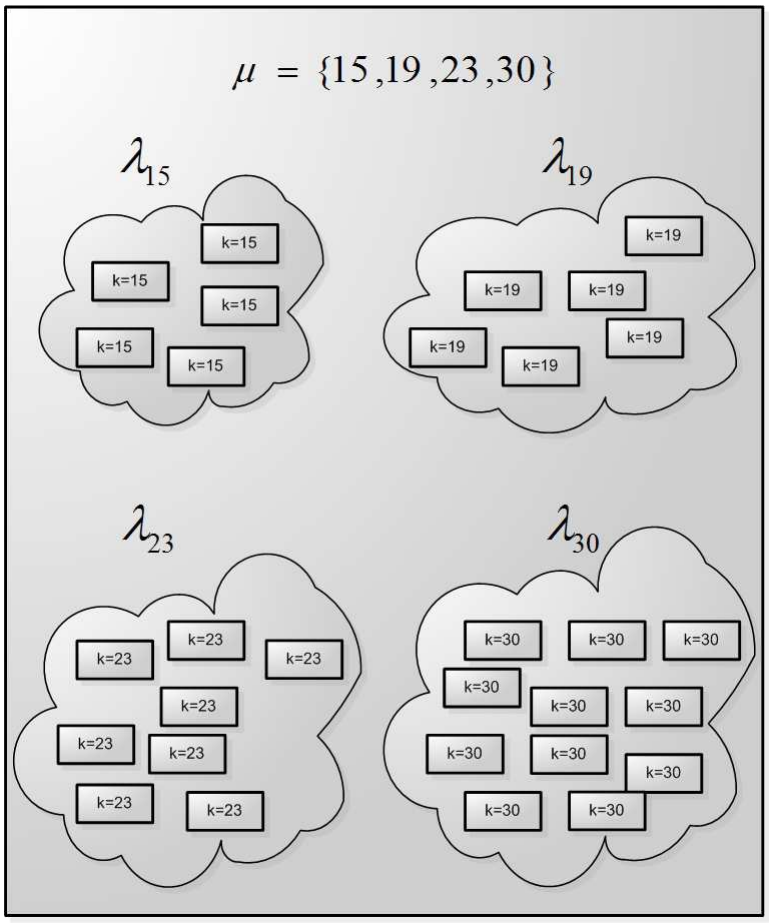

Fig. 7. Each Set of Packet Markings Has the Same Distance Field Value.

$\forall k \in \mu$). The starting root for the attack path is the victim. The markings of the immediate children of the victim are checked with each marking in the set λ_k. Routers whose markings match are added to the attack graph. In the next iteration, the children of these routers are checked in a similar fashion. This process is repeated till the depth in the graph equals the path length. The attack path is contained in S_d, where $0 \leq d \leq k$.

Table IV. Path Reconstruction Algorithm

$\forall\ k \in \mu$

$S_0 = victim$

for $d = 0$ to $(k-1)$

$\forall\ y$ in S_d

$\forall\ R \in \rho_y$

if $R \in \lambda_k$ then

insert $R \longrightarrow S_{d+1}$

output S_d

output S_k

E. Analysis

The packet marking and path reconstruction algorithms were presented in the previous sections. We now look at the computational complexities and the number false positives that are obtained while executing these algorithms.

1. Computational Complexity

Table II shows the packet marking algorithm. Before we discuss the complexity involved in the packet marking scheme, we first discuss the algorithm performed by the router to route a packet without any marking scheme involved [24].

When a router receives a packet on its interface, basic checks are performed. These include IP header validity verification, packet filtering policy and a TTL field check.

The router then makes a routing decision based on a search of its routing table

and determines the outbound interface and the IP address of the router that should be the next hop in the path. Before the packet is forwarded, its TTL vale is updated and so is the checksum.

Returning to our packet marking algorithm, the marking value, $m = h(\text{IP address})$ is computed one time by a router and not on a packet-by-packet basis. The same marking is put on all marked packets. The value in the distance field can be read at the same time the TTL value and other IP header validity checks are performed.

After the value in the distance field has been read, the corresponding probability is looked up from a pre-calculated table. This table may be a data structure best optimized for the search. We know that the distance value in the packet can be one of 32 values. Hence a simple binary search on the ordered table can provide a look-up complexity of only $O(\log_2 32)$. Or, since the number of entries to be searched is small, the router can perform a sequential search on the entries, the entries being ordered in decreasing order of occurrence of various path lengths in the Internet as given in [20]. As mentioned previously, the random number used for the marking decision is not a cryptographically secure random number. Generation of a statistically good random number may be performed using time seed values in a simple random number generator.

Writing the marking and incrementing the distance field value in the packet can be done when the TTL and header checksum fields of the IP header are updated. Hence, we see that the packet marking algorithm does not increase the routing overhead significantly and can be implemented easily on the routers.

Observing the path reconstruction algorithm shown in Table IV, the computational complexity of the path reconstruction algorithm can be expressed as being $O\Big(\sum_{k\in\mu}\sum_{0\le d\le k-1}\big[\ \mid S_d \mid \cdot \sum_{y\in S_d}[\mid \rho_y \mid \cdot \log(\mid \lambda_k \mid)]\big]\Big)$.

2. Number of Packets Required for Reconstruction

We now look at the number of packets required for reconstructing the attack path. The Coupon Collector's problem [25] is used to find the number of packets required for path reconstruction. This has been discussed in Appendix A. Path reconstruction for a particular path length can only be done when the victim has received packets marked by all the routers in the attack path.

Using the result from the Coupon Collector's problem, (A.1), for a path of length k, the expected number of packets required to receive markings by all the routers in that path is bounded as follows:

$$E[\text{number of packets}] \leq (k - 1 + c) \cdot [\ln(k - 1 + c)] \tag{3.5}$$

For a Distributed Denial of Service attack, the number of packets required is determined using (3.5)

$$E[\text{number of packets}] \leq \sum_{k \in \mu} \left[n_k \cdot (k - 1 + c) \cdot [\ln(n_k \cdot (k - 1 + c))] \right] \tag{3.6}$$

3. False Positives

Referring to the attack path construction algorithm, the markings of the routers in the graph are compared to the set of markings received by the victim for different distance values. A false positive occurs when a reconstructed attack path is actually not an attack path. This means that the reconstruction algorithm falsely identified a path in the graph as being part of the DoS attack.

For all $y \in S_d$, there are some routers in ρ_y that are also be in the set λ_k. This is the set of routers in ρ_y that are actually present in the attack paths. Denote these routers by m_{ρ_y, λ_k}, the length of the attack paths being reconstructed is k.

For each router, y, in the graph,

$$\mid \rho_y \cap \lambda_k \mid = \mid m_{\rho_y,\lambda_k} \mid$$

or

$$\mid \{\rho_y - m_{\rho_y,\lambda_k}\} \cap \lambda_k \mid = 0$$

However, for a false positive, there are markings in ρ_y other than m_{ρ_y,λ_k} that are also present in λ_k. This can be expressed as

$$\mid \{\rho_y - m_{\rho_y,\lambda_k}\} \cap \lambda_k \mid \geq 1$$

Consider $\{\rho_y - m_{\rho_y,\lambda_k}\}$ and λ_k to be sets of markings that have been selected from a possible set of N different markings. Let $(\rho_y - m_{\rho_y,\lambda_k})_1$ be the first element of the set $\{\rho_y - m_{\rho_y,\lambda_k}\}$. The probability that this marking does not match with any other marking in λ_k is equal to the probability that it belongs to the set of all possible markings other than those present in the set λ_k.

$$Pr[\mid \{(\rho_y - m_{\rho_y,\lambda_k})_1\} \cap \lambda_k \mid = 0] \;=\; \left(\frac{N - \mid \lambda_k \mid}{N}\right)$$

Probability that no marking in $\{\rho_y - m_{\rho_y,\lambda_k}\}$ matches any marking in λ_k is given by,

$$Pr[\mid \{\rho_y - m_{\rho_y,\lambda_k}\} \cap \lambda_k \mid = 0] \;=\; \left(\frac{N - \mid \lambda_k \mid}{N}\right)^{|\{\rho_y - m_{\rho_y,\lambda_k}\}|} \tag{3.7}$$

From the above discussion,

$$Pr[\mid \{\rho_y - m_{\rho_y,\lambda_k}\} \cap \lambda_k \mid \geq 1] \;=\; 1 - Pr[\mid \{\rho_y - m_{\rho_y,\lambda_k}\} \cap \lambda_k \mid = 0]$$

Using (3.7),

$$\begin{aligned} Pr[\mid \{\rho_y - m_{\rho_y,\lambda_k}\} \cap \lambda_k \mid \geq 1] &= 1 - \left(\frac{N - \mid \lambda_k \mid}{N}\right)^{|\{\rho_y - m_{\rho_y,\lambda_k}\}|} \\ Pr[\mid \{\rho_y - m_{\rho_y,\lambda_k}\} \cap \lambda_k \mid \geq 1] &= 1 - \left(\frac{N - \mid \lambda_k \mid}{N}\right)^{(|\rho_y| - |m_{\rho_y,\lambda_k}|)} \end{aligned} \tag{3.8}$$

A false positive occurs when, at each level up to the path length, for each y, there is at least one other element in $\{\rho_y - m_{y,\lambda_k}\}$ that is also in λ_k. Using (3.8),

$$Pr[\text{false positive}] = \prod_{\substack{y \in S_d \\ 0 \leq d \leq k-1}} \left[1 - \left(\frac{N - \mid \lambda_k \mid}{N}\right)^{(|\rho_y| - |m_{\rho_y,\lambda_k}|)}\right] \tag{3.9}$$

Probability of a false positive for the Distributed Denial of Service attack is the sum of probabilities of false positive for each path length.

$$Pr[\text{false positive for DDoS attack}] = \sum_{\forall k \in \mu} \prod_{\substack{y \in S_d \\ 0 \leq d \leq k-1}} \left[1 - \left(\frac{N - \mid \lambda_k \mid}{N}\right)^{(|\rho_y| - |m_{\rho_y,\lambda_k}|)}\right] \tag{3.10}$$

4. Number of Unique Objects

We use 11 bits of the IP identification field for the router markings. This means that there are a total of $N = 2^{11} = 2048$ possible unique markings.

The probability that the markings of two routers are the same is non-zero. This is because the entire 32-bit IP address space is mapped to 2048 markings by the mapping function, $h(.)$ which can result in collisions. Hence, if we pick a set of routers, not all corresponding markings will be unique. Intuitively, for a small selection of routers, we can expect to have almost all unique markings. However, as the number of routers in our selection increases, the number of duplicate markings i.e., collisions of the hash function also increases. We now need to quantify this behavior and find the expected number of unique markings given a particular number of routers.

This problem can be viewed as being similar to a coupon collection problem [26] and the results have been summarized in Appendix A.

Let $\mathcal{U}_{N,k}$ be the expected number of unique markings when k drawings are made with replacement from a set of N different markings.

Hence, using (A.3),

$$\mathcal{U}_{N,k} = \sum_{i=1}^{k} \left[i. \binom{N}{i} \frac{\sum_{j=0}^{i} (-1)^j \binom{i}{j} (i-j)^k}{N^k} \right] \tag{3.11}$$

Up until now (3.4), the definition of λ_k made it equal to $n_k \cdot k$. We can now more accurately define λ_k as being the set of *unique* markings received by the victim with distance field value k.

$$\mid \lambda_k \mid = \mathcal{U}_{N,n_k \cdot k} = \sum_{i=1}^{n_k \cdot k} \left[i \cdot \binom{N}{i} \frac{\sum_{j=0}^{i} (-1)^j \binom{i}{j} (i-j)^{n_k \cdot k}}{N^{n_k \cdot k}} \right] \tag{3.12}$$

In the next chapter, simulation results are presented.

CHAPTER IV

SIMULATION RESULTS

A. Overview

The proposed packet marking and attack path reconstruction algorithms, and their mathematical analysis were presented in Chapter III. The performance of these algorithms was tested using data collected from the Internet. In this chapter, we discuss the simulation methodology and present our findings. We test and observe the performance of our algorithms for the probability of receiving packets marked by routers at various distances from the victim, the number of packets required for attack path reconstruction and the time required for receiving packets and reconstructing the attack paths. The performance of the proposed scheme is compared with the Advanced Marking Scheme proposed by Song and Perrig [17].

B. Simulation Tools

To test and simulate the proposed algorithms, we use data collected by CAIDA's Skitter tool [21][1]. This data is a real traceroute data set generated when each of CAIDA's topology monitors continuously probe various destinations in the Internet. CAIDA's Arts++ package [27] is used to parse and manipulate the trace data. All simulations are performed using Network Simulator-2 (NS-2) [28].

The BRITE topology generator [29] is used to convert the trace data into a format that is compatible with NS-2. The general simulation methodology is shown in Figure 8. All simulations were carried out on the Linux operating system running

[1]The data used in this research was collected as part of CAIDA's skitter initiative, http://www.caida.org. Support for skitter is provided by DARPA, NSF, and CAIDA membership.

on a 2.40 GHz Pentium 4 system.

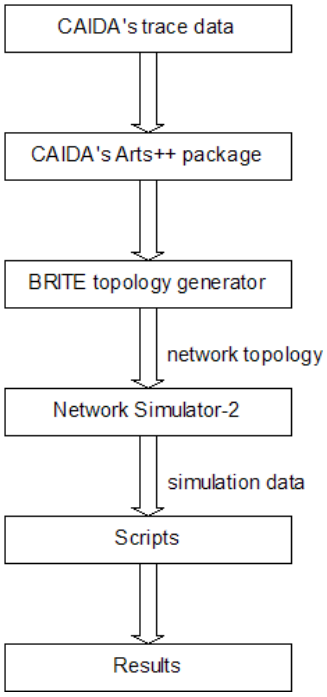

Fig. 8. Simulation Methodology.

C. Simulation Scenario

In Chapter III, we introduced a hash function, $h(.)$, that maps the IP address space over the 2^{11} possible router markings. We use the MD5 [30] cryptographic hash of a router's IP address to obtain the packet marking. The hash function, $h(.)$, returns the last 11 bits of the MD5 cryptographic hash of the router's IP address. This computation is not done on a per-packet basis. Each router computes this only once, and uses it to mark packets as necessary.

In the data set used, there are a total of 365605 different destinations at various hop counts from the single source that probes these destinations. The distribution of the number of hops from the source to the different destinations is shown in Figure 9.

We can see that all paths are less than 32 hops in length. This justifies our choice of the size of the distance field value in the IP identification field as 5 bits.

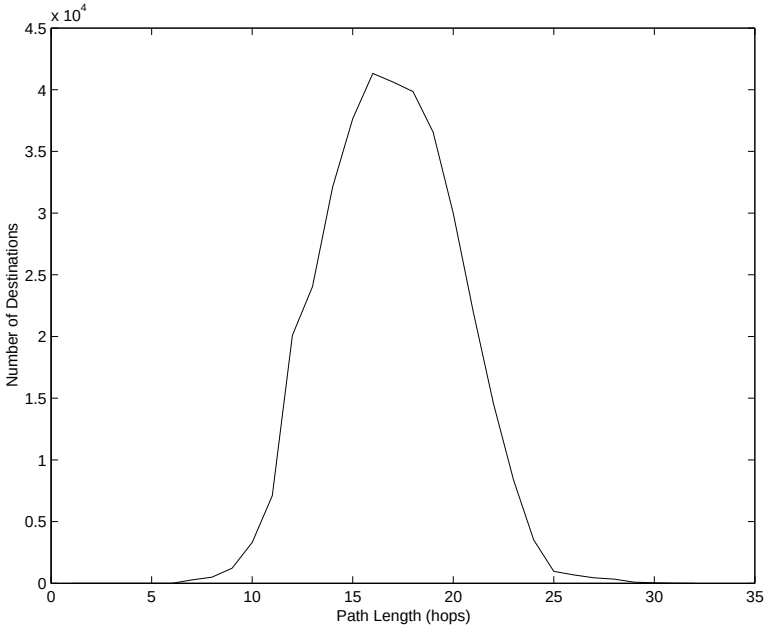

Fig. 9. Hop Count Distribution in the Trace Data.

For the purpose of comparison, we perform a similar simulation using the Advanced Marking Scheme I (AMS) proposed by Song and Perrig [17]. For each data point, the simulation was repeated 100 times and the average was taken as the final result.

D. Simulation Results and Discussion

1. Marking Probability

The probability of receiving a packet marked my a router at different distances from the victim for various attack path lengths is shown in Figure 10. We have seen earlier in Chapter II that routers in the AMS use a fixed probability, q, to mark packets.

The parameter c has been described in (3.1). In general, other parameters remaining the same, lower the value of q and c, lower is the probability that a router marks a packet it forwards. As can be seen, the proposed marking scheme provides a uniform probability of receiving packets marked by all routers along the attack path. For AMS, the probability of receiving packets marked by a router decreases as the distance of the router from the victim increases. This results in a large number of packets being required, and hence a longer delay before path reconstruction can begin.

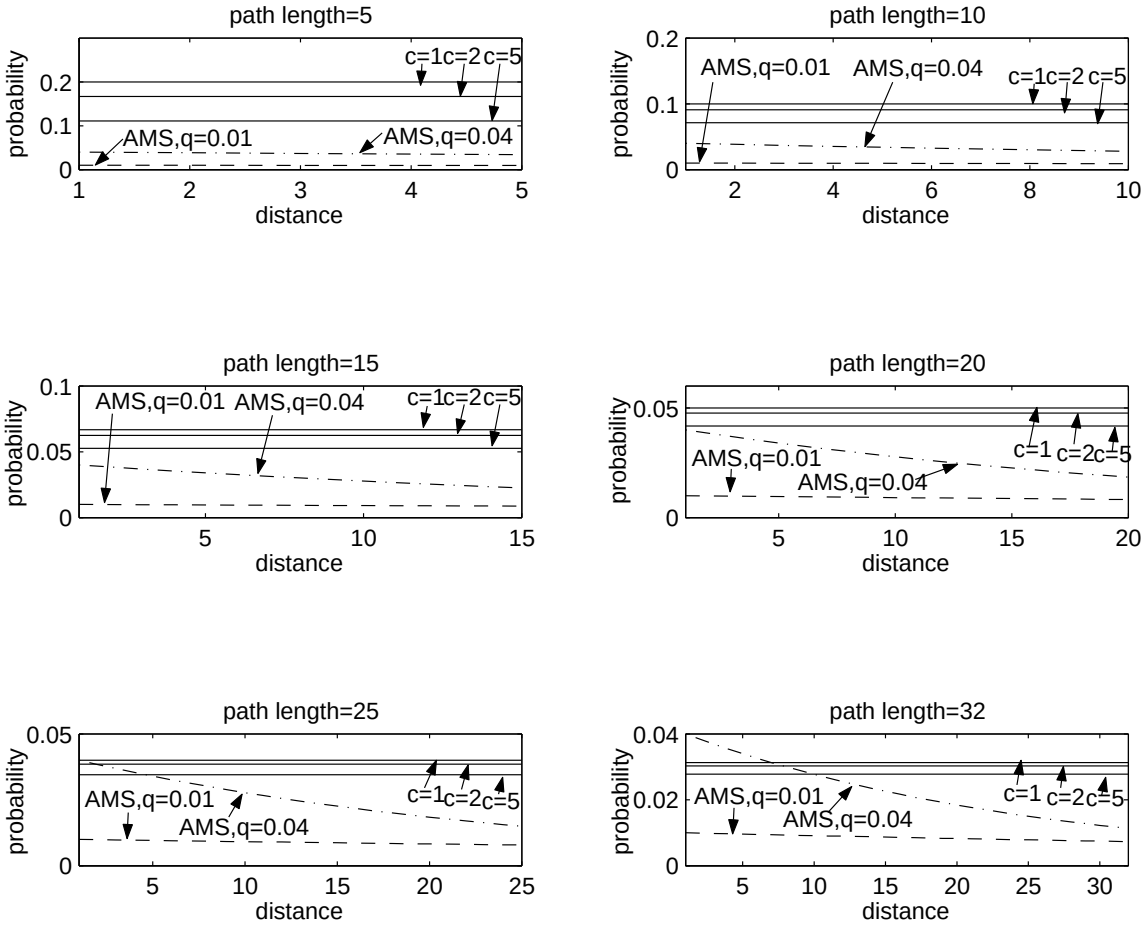

Fig. 10. Probability of Receiving a Packet Marked by Routers at Different Distances from the Victim for Various Path Lengths.

a. The parameter c

We have seen that routers mark packets with a probability that is computed using (3.1). Besides the distance field value in the IP identification field, the parameter, c, also determines the fraction of total forwarded packets marked by a router.

For the first router in a packet's path, the distance field value, $d-1$, is zero. This means that the router marks packets with a probability, $1/c$. Thus, c solely determines the fraction of total packets marked by the first router. This first router is usually the gateway, or an ISP's aggregation point. Table V shows the fraction of packets marked for different values of c. We see that if $c = 1$, all the packets passing through the router are parked. This may put an excessive load on the gateway router and may be unjustified in the case of legitimate traffic during a non-attack period. For $c = 5$, only 20% of the packets are marked. In the case of a high-volume DoS attack, this may let through enough packets to cause problems for the victim. Hence, a tradeoff ought to be made between the load on the routers and the number of packets that are allowed to pass through without any markings. From our simulations, we observed that $c = 2$ provides a good compromise. In a real world deployment, this value may be a commonly agreed upon parameter for the scheme.

Table V. Fraction of Total Packets Marked by the First Router for Different Values of c.

c	percentage
c=1	100%
c=2	50%
c=5	20%

We will also see in the subsequent sections that from the victim's perspective, the

choice of the value of c influences only the number of expected packets required for path reconstruction and hence the expected waiting time before path reconstruction can commence. It has no effect on the actual path reconstruction time, or the number of false positives.

2. Expected Number of Packets Required

Figure 11 shows the number of packets required for path reconstruction. This quantity is important because it determines the amount of time the victim has to wait before it can start reconstructing the attack path. In other words, it is the number of packets required to receive markings from all the routers along the attack path. Larger the value of this quantity, larger is the waiting time for the victim. From Figure 11, clearly,

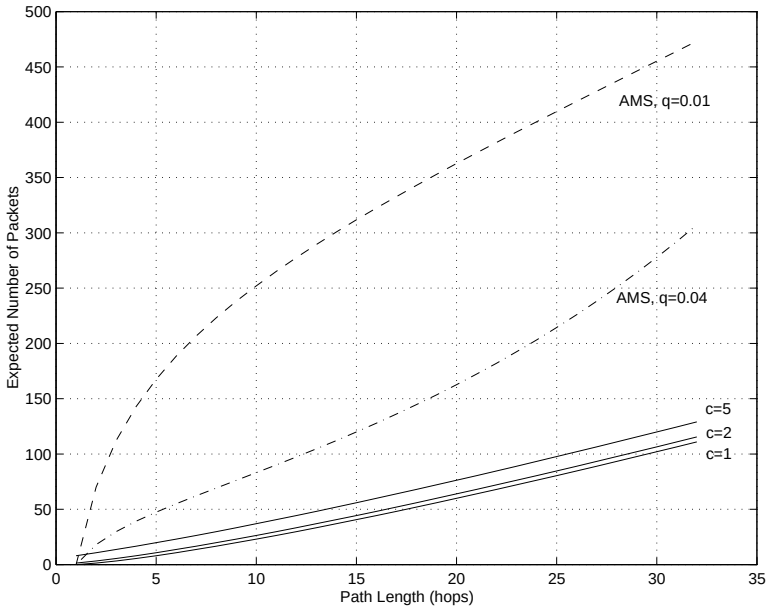

Fig. 11. Expected Number of Packets Required for Reconstruction.

the proposed marking algorithm requires several times lower number of packets than

the AMS scheme. For large attack path lengths, the AMS scheme using $q = 0.01$ requires about 300% more packets than our proposed scheme with $c = 2$.

The time required to receive the expected number of packets required for reconstruction for different path lengths and various traffic rates is shown in Figure 12. For clarity of presentation, only our proposed algorithm with $c = 2$ and the AMS with $q = 0.01$ are shown. Again, time required for the AMS is several times higher than that required by our algorithm.

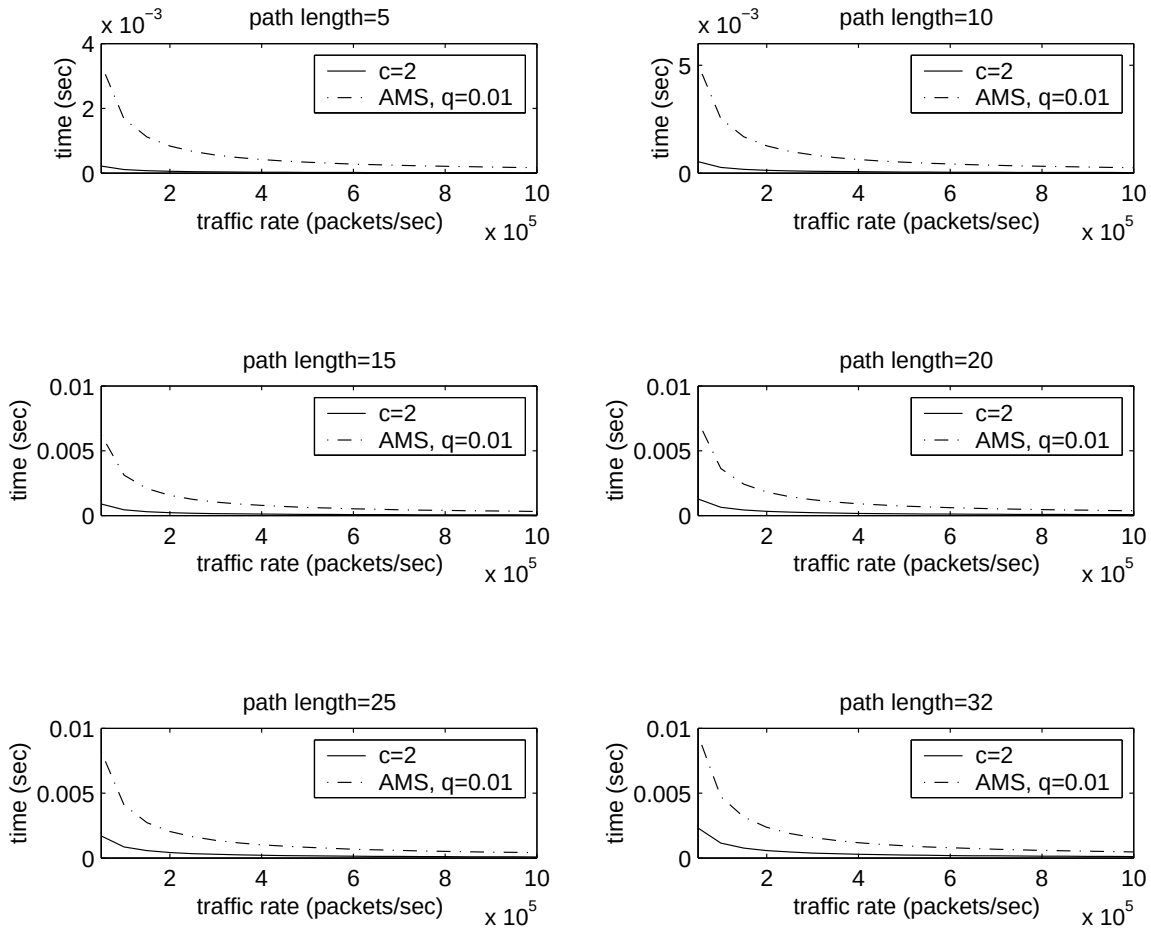

Fig. 12. Time Taken to Receive the Expected Number of Packets Required for Reconstruction for Different Path Lengths and Various Traffic Rates.

3. Path Reconstruction Time

Figure 13 shows the attack path reconstruction time for different number of attackers. To test the reconstruction time of our algorithm for a distributed denial of service attack, we chose random attackers at different distances from the victim. The number of attackers at various distances have a distribution similar to Figure 14. Our proposed mechanism performs better than the AMS for all test cases.

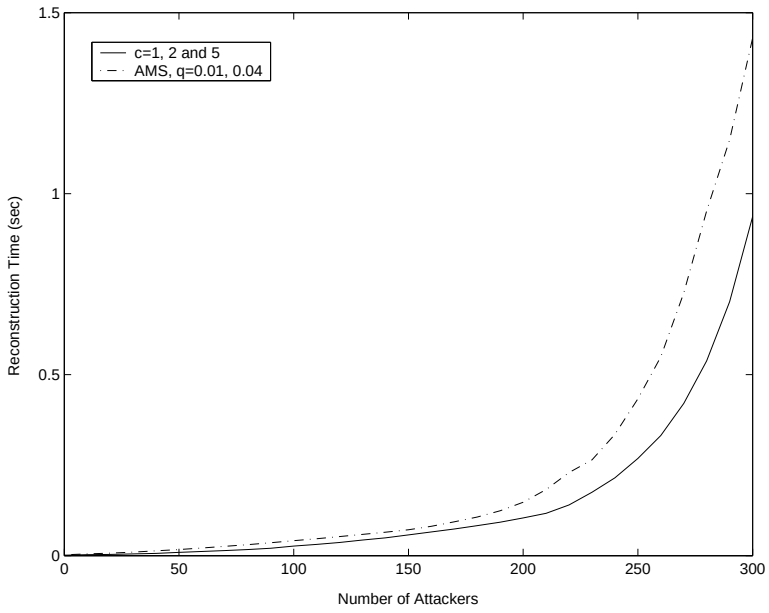

Fig. 13. Path Reconstruction Time.

4. False Positives

An important metric in path reconstruction is the number of systems incorrectly identified as attackers. These are the number of false positives obtained when reconstructing the attack paths from the received markings. We tested our scheme for the number of false positives obtained during reconstruction. For a single attacker, the proposed scheme never returned a false positive for any path length. That is,

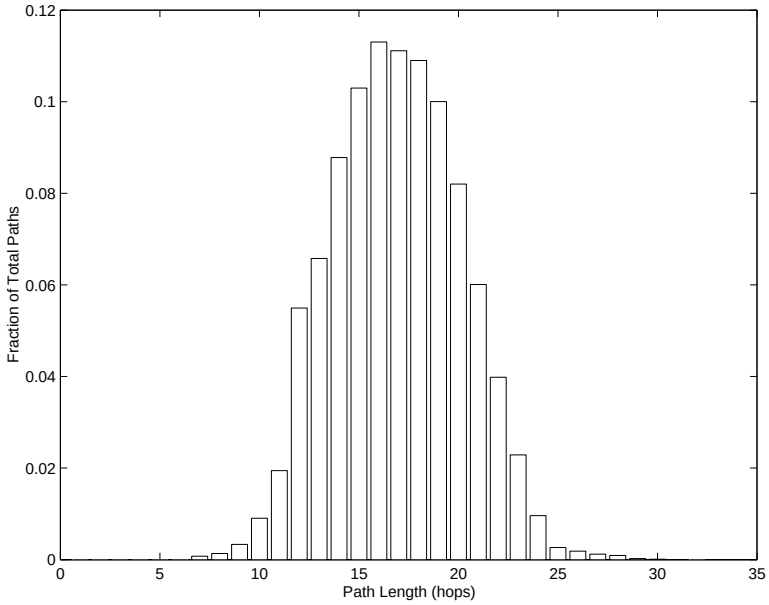

Fig. 14. Hop Count Distribution as a Fraction of Total Paths.

it correctly identified a single attack source in all the trials. The AMS produced a similar result.

To test the response of our algorithm to a distributed denial of service attack, we chose random attackers at different distances from the victim. The number of attackers at various distances have a distribution similar to Figure 14. In Figure 15, we see that the proposed path reconstruction algorithm performs well under an attack from multiple attackers and has a better performance than the AMS under similar conditions. For a small number of attackers, the algorithm performs well. However, as the number of attackers increases, the number of sources incorrectly identified as attackers also increases. The false positive rate is 30% when there are 300 multiple attack sources. This may be explained using the fact that an increase in the number of attackers decreases the number of unique markings received. With a large number of received router markings, the chances of having similar markings increases, hence the high false positive rate.

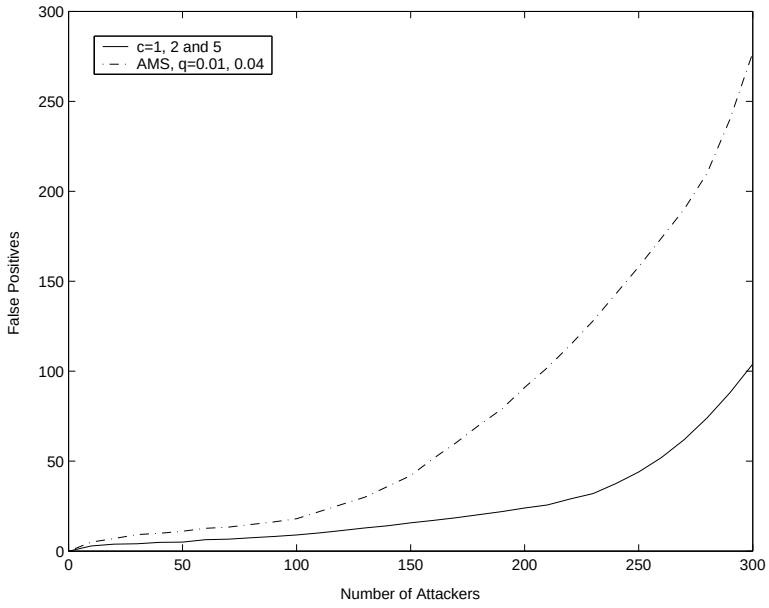

Fig. 15. Number of False Positives.

CHAPTER V

CONCLUSIONS AND FUTURE WORK

A. Conclusions

This research work proposes a path reconstruction mechanism for identifying the source of Denial of Service attacks. The proposed algorithm uses adjusted probabilistic packet marking for IP traceback.

The scheme does not require any new fields in the IP header and overloads the existing IP identification field with the router marking. A simple algorithm is used by routers to mark packets. The markings received by the victim are used to reconstruct the attack path using an easily available map of the Internet. In Chapter IV, we saw that the scheme performs well in terms of time required for path reconstruction and the number of false positives. It shows a good response against a distributed denial of service attack.

B. Future Work

While the algorithm works well when none of the routers participating in the scheme are compromised, the authenticity of the markings cannot be verified in case a router itself is participating in the DoS attack. An authentication mechanism needs to be introduced into the proposed algorithm to make it more robust against compromised routers. This would result in a scheme that provides a good performance along with a high degree of security.

Our scheme is not a complete solution against DoS attacks. It needs to be used in conjunction with other filtering mechanisms such as that proposed by Yaar, Perrig and Song [14]. Combined together, these schemes present a formidable challenge to

the attackers, who now have to deal with a target that can not only filter out the malicious traffic but can also pro-actively identify them.

In our scheme, routers are used to send traceback information to the victim of a DoS attack. With some simple modifications, our scheme may also be used to exchange covert information in steganographic applications [31], [32].

REFERENCES

[1] L. A. Gordon, M. P. Loeb, W. Lucyshyn, and R. Richardson, "2004 CSI/FBI Computer Crime and Security Survey," Computer Security Institute, June 2004, http://www.gocsi.com.

[2] "Microsoft's Web Site Brought Down By Attack," [Online]. Available: http://www.informationweek.com/story/showArticle.jhtml?articleID=12808118. Accessed: August 2003.

[3] "Denial-of-Service attack causes web blackout," [Online]. Available: http://networks.silicon.com/webwatch/0,39024667,39121399,00.htm. Accessed: June 2004.

[4] "Attacks on Windows PC's Grew in First Half of 2004," [Online]. Available: http://www.nytimes.com/2004/09/20/technology/20secure.html. Accessed: September 2004.

[5] J. Postel, "RFC 791 - Internet Protocol," RFC 791, September 1981, http://www.faqs.org/rfcs/rfc791.html.

[6] A. Leon-Garcia and I. Widjaja, *Communication Networks: Fundamental Concepts and Key Architectures*, Boston, MA: McGraw-Hill Higher Education, 1st edition, 2000.

[7] "Defining Strategies to Protect Against TCP SYN Denial of Service Attacks," [Online]. Available: http://www.cisco.com/warp/public/707/4.html. Accessed: January 2004.

[8] J. Postel, "RFC 793 - Transmission Control Protocol," RFC 793, September 1981, http://www.faqs.org/rfcs/rfc793.html.

[9] J. Postel, "RFC 768 - User Datagram Protocol," RFC 768, August 1980, http://www.faqs.org/rfcs/rfc768.html.

[10] J. Postel, "RFC 792 - Internet Control Message Protocol," RFC 792, September 1981, http://www.faqs.org/rfcs/rfc792.html.

[11] P. Ferguson and D. Seniel, "RFC 2267 - Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing," RFC 2267, January 1998, http://www.ietf.org/rfc/rfc2267.txt.

[12] SANS Institute, "Egress Filtering v0.2," Global Incident Analysis Center-Special Notice, February 2000, [Online]. Available: http://www.sans.org/y2k/egress.htm.

[13] T. M. Gil and M. Poletto, "MULTOPS: A Data-Structure for Bandwidth Attack Detection," in *Proc. Tenth USENIX Security Symposium*, USENIX, Ed., Washington, D.C., USA, August 13–17 2001, pp. 23–38, USENIX.

[14] A. Yaar, A. Perrig, and D. Song, "Pi: A path identification mechanism to defend against DDoS attacks," in *Proc. 2003 IEEE Symposium on Security and Privacy*, Berkeley, CA, USA, May 11–14 2003, pp. 93–109, IEEE Computer Society.

[15] T. W. Doeppner, P. N. Klein, and A. Koyfman, "Using router stamping to identify the source of IP packets," in *Proc. 7th ACM Conference on Computer and Communications Security*, Athens, Greece, November 2000, pp. 184–189, ACM Press.

[16] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, "Practical network support for IP traceback," in *Proc. 2000 ACM SIGCOMM Conference*, Stockholm, Sweden, August 2000.

[17] D. X. Song and A. Perrig, "Advanced and authenticated marking schemes for IP traceback," in *Proc. IEEE Infocom 2001*, Anchorage, AK, USA, April 2001, vol. 2, pp. 878 – 886.

[18] T. Peng, C. Leckie, and K. Ramamohanarao, "Adjusted probabilistic packet marking for IP traceback," in *Networking 2002*, Pisa, Italy, May 2002, pp. 697–708.

[19] I. Stoica and H. Zhang, "Providing guaranteed services without per flow management," in *Proc. SIGCOMM '99*, Cambridge, MA, USA, September 1999, pp. 81–94.

[20] W. Theilmann and K. Rothermel, "Dynamic distance maps of the internet," in *Proc. 2000 IEEE Computer and Communications Societies Conference on Computer Communications (INFOCOM-00)*, Tel Aviv, Israel, March 26–30 2000, pp. 275–284, IEEE.

[21] "Cooperative Association for Internet Data Analysis (CAIDA) Skitter Tool," [Online]. Available: http://www.caida.org/tools/measurement/skitter/. Accessed: November 2003.

[22] "The NLANR Measurement and Network Analysis Group," [Online]. Available: http://moat.nlanr.net. Accessed: November 2003.

[23] "The Internet Mapping Project," [Online]. Available: http://research.lumeta.com/ches/map/index.html. Accessed: November 2003.

[24] A. Zinin, *Cisco IP Routing : Packet Forwarding and Intra-domain Routing Protocols*, Boston, MA: Addison-Wesley, 1st edition, 2002.

[25] W. Feller, *An Introduction to Probability Theory and its Applications*, vol. 1, New York: John Wiley & Sons, Inc., January 1968.

[26] E. Parzen, *Modern Probability Theory and Its Applications*, chapter 2, New York: John Wiley & Sons, Inc., 1960.

[27] "Cooperative Association for Internet Data Analysis (CAIDA) arts++ package," [Online]. Available: http://www.caida.org/tools/utilities/arts/. Accessed: November 2003.

[28] "The Network Simulator - NS-2," [Online]. Available: http://www.isi.edu/nsnam/ns/. Accessed: November 2003.

[29] "BRITE - Boston University Representative Internet Topology Generator," [Online]. Available: http://www.cs.bu.edu/brite/. Accessed: November 2003.

[30] R. Rivest, "RFC 1321 - The MD5 Message-Digest Algorithm," RFC 1321, April 1992, http://www.faqs.org/rfcs/rfc1321.html.

[31] K. Ahsan and D. Kundur, "Practical Data Hiding in TCP/IP," in *Proc. ACM Multimedia 2002 Workshop W2 - Workshop on Multimedia and Security: Authentication, Secrecy, and Steganalysis*, Juan-Les-Pins, France, Dec 1–6 2002, pp. 7–14.

[32] D. Kundur and K. Ahsan, "Practical Internet Steganography: Data Hiding in IP," in *Proc. Texas Workshop on Security of Information Systems*, College Station, TX, USA, April 2003.

APPENDIX

COUPON COLLECTION PROBLEMS

Expected Number of Drawings Before We Get One Coupon of Each Type:

We now examine the classic Coupon Collector's problem. This problem has been discussed by Feller [25]. Assume we have a set of M distinct, equiprobable coupons. We pick one coupon from the set, examine it and replace it back into the set. Trials are mutually independent. We keep doing so till we get one coupon of each type. The expected number of such drawings before we get one coupon of each type is given by

$$\begin{aligned} E[\text{number of drawings}] &= M\left[1+\frac{1}{2}+\frac{1}{3}+\ldots+\frac{1}{M}\right] \\ &\approx M\left[\ln(M)+\gamma\right] \\ &\approx M\ln(M) \end{aligned} \tag{A.1}$$

where $\gamma = 0.5772\ldots$ is the Euler's constant.

Expected Number of Unique Coupons:

This problem has been discussed by Parzen [26]. Suppose we have a set of M coupons numbered from 1 to M. We select r coupons from the set one-by-one, each time noting the number on the coupon, and replacing it back. Selections are mutually independent. Then, the probability that exactly m of the M integers, 1 to M will be selected is given by

$$Pr[m \text{ integers selected}] = \binom{M}{m}\frac{\sum_{j=0}^{m}(-1)^j\binom{m}{j}(m-j)^r}{M^r} \tag{A.2}$$

Using (A.2), we can find the expected number of unique integers when we select r

coupons. This is given by

$$E[\text{number of unique integers}] = \sum_{i=1}^{r} \Big[i.Pr[i \text{ integers selected}] \Big]$$

$$E[\text{number of unique integers}] = \sum_{i=1}^{r} \left[i. \binom{M}{i} \frac{\sum_{j=0}^{i} (-1)^j \binom{i}{j} (i-j)^r}{M^r} \right] \quad \text{(A.3)}$$

VITA

Raghav Dube received the B.E. degree in Electronics Engineering from Motilal Nehru Regional Engineering College, Allahabad, India in June 2002. He received the M.S. degree in Electrical Engineering from Texas A&M University, College Station, Texas, U.S.A. in December 2004.

Address:

Raghav Dube
Department of Electrical Engineering
214 Zachry Engineering Center
Texas A&M University
College Station, TX 77843, U.S.A.

The typist for this thesis was Raghav Dube.