ABSTRACT

ON MITIGATING DISTRIBUTED DENIAL OF SERVICE ATTACKS

by
Zhiqiang Gao

Denial of service (DoS) attacks and distributed denial of service (DDoS) attacks are probably the most ferocious threats in the Internet, resulting in tremendous economic and social implications/impacts on our daily lives that are increasingly depending on the well-being of the Internet. How to mitigate these attacks effectively and efficiently has become an active research area. The critical issues here include 1) IP spoofing, i.e., forged source IP addresses are routinely employed to conceal the identities of the attack sources and deter the efforts of detection, defense, and tracing; 2) the distributed nature, that is, hundreds or thousands of compromised hosts are orchestrated to attack the victim synchronously. Other related issues are scalability, lack of incentives to deploy a new scheme, and the effectiveness under partial deployment.

This dissertation investigates and proposes effective schemes to mitigate DDoS attacks. It is comprised of three parts. The first part introduces the classification of DDoS attacks and the evaluation of previous schemes. The second part presents the proposed IP traceback scheme, namely, autonomous system-based edge marking (ASEM). ASEM enhances probabilistic packet marking (PPM) in several aspects: (1) ASEM is capable of addressing large-scale DDoS attacks efficiently; (2) ASEM is capable of handling spoofed marking from the attacker and spurious marking incurred by subverted routers, which is a unique and critical feature; (3) ASEM can significantly reduce the number of marked packets required for path reconstruction and suppress false

positives as well. The third part presents the proposed DDoS defense mechanisms, including the four-color-theorem based path marking, and a comprehensive framework for DDoS defense. The salient features of the framework include (1) it is designed to tackle a wide spectrum of DDoS attacks rather than a specified one, and (2) it can differentiate malicious traffic from normal ones. The receiver-center design avoids several related issues such as scalability, and lack of incentives to deploy a new scheme. Finally, conclusions are drawn and future works are discussed.

ON MITIGATING DISTRIBUTED DENIAL OF SERVICE ATTACKS

by
Zhiqiang Gao

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Computer Engineering

Department of Electrical and Computer Engineering

August 2006

APPROVAL PAGE

ON MITIGATING DISTRIBUTED DENIAL OF SERVICE ATTACKS

Zhiqiang Gao

Dr. Nirwan Ansari, Dissertation Advisor — Date
Professor of Electrical and Computer Engineering, NJIT

Dr. Swades K. De, Committee Member — Date
Assistant Professor of Electrical and Computer Engineering, NJIT

Dr. Sui-hoi E. Hou, Committee Member — Date
Associate Professor of Electrical and Computer Engineering, NJIT

Dr. Teunis J. Ott, Committee Member — Date
Professor of Computer Science, NJIT

Dr. Roberto Rojas-Cessa, Committee Member — Date
Assistant Professor of Electrical and Computer Engineering, NJIT

BIOGRAPHICAL SKETCH

Author: Zhiqiang Gao

Degree: Doctor of Philosophy

Date: August 2006

Undergraduate and Graduate Education:

- Doctor of Philosophy in Computer Engineering,
  New Jersey Institute of Technology, Newark, NJ, 2006

- Master of Science in Electrical Engineering,
  Chinese Academy of Sciences, Nanjing, P. R. China, 1997

- Bachelor of Engineering in Computer Science,
  Zhejiang University, Hangzhou, P. R. China, 1989

Major: Computer Engineering

Presentations and Publications:

Zhiqiang Gao and Nirwan Ansari,
"Tracing cyber attacks from the practical perspective,"
IEEE Communications Magazine, vol. 43, no. 5, pp. 123-131, May 2005.

Zhiqiang Gao and Nirwan Ansari,
"A practical and robust inter-domain marking scheme for IP traceback,"
accepted by Computer Networks.

Zhiqiang Gao and Nirwan Ansari,
"Enhanced probabilistic packet marking for IP traceback,"
IEEE GLOBECOM'2005, St Louis, MO, Nov. 28-Dec. 2, 2005.

Zhiqiang Gao and Nirwan Ansari,
"Directed geographical traceback,"
3rd international conference on information technology: research and education (IEEE ITRE'2005), Taiwan, June 2005, pp. 221-224.

Zhiqiang Gao, Nirwan Ansari, and K. Anantharam,
"A new marking scheme to defend against distributed denial of service attacks," IEEE GLOBECOM'2004, Dallas, TX, December 2004, vol. 4, pp. 2256-2260.

Zhiqiang Gao and Nirwan Ansari,
"On the marking probability of probabilistic packet marking for IP traceback," 38th conference on information scieneces and systems (CISS'2004), Princeton, NJ, March 2004, pp. 1290-1293.

Zhiqiang Gao and Nirwan Ansari,
"Differentiating malicious DDoS attack traffic from normal TCP flows by proactive tests," submitted to IEEE Communications Letters.

Zhiqiang Gao and Nirwan Ansari,
"A routing friendly marking scheme for DDoS defense," submitted to Milcom 2006.

Zhiqiang Gao and Nirwan Ansari,
"A comprehensive framework for DDoS defense," in preparation.

US Provisional Patents:

Nirwan Ansari and Zhiqiang Gao,
"Behavior-based Traffic Differentiation (BTD) to Defend against Distributed Denial of Service Attacks," filed in January 2006.

Nirwan Ansari and Zhiqiang Gao,
"Autonomous System-based Edge Marking (ASEM) for IP Traceback," filed in January 2006.

To those I cherish, from the bottom of my heart

ACKNOWLEDGMENT

I would like to express my deepest appreciation to Dr. Nirwan Ansari, who not only served as my research supervisor, providing countless resources and insights, but also constantly gave me invaluable support and encouragement. I have learned a lot from him. Special thanks are given to Dr. Swades De, Dr. Edwin Hou, Dr. Teunis Ott, and Dr. Roberto Rojas-Cessa for actively participating in my committee and their constructive comments.

Many of my peers in the ECE department have supported me a lot whenever I needed. They are Dongdong Fu, Yuanqiu Luo, Amey Shevtekar, Zhicheng Ni, Kai Xu, Gang Cheng, Chao Zhang, and Pitipatana Sakarindr. I also wish to thank them for their help over the years.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES
(Continued)

CHAPTER 1

INTRODUCTION

1.1 Background

E-business has drastically enhanced companies' revenue growth, lowered operational costs, and increased customer satisfaction during the last decade. E-business applications such as supply-chain management and remote access require mission-critical networks that can handle voice, video, and data traffic and are scalable enough to support more users and meet more stringent performance requirements. However, as more applications are available online to more users, networks become even more vulnerable to a wider spectrum of security threats. To defeat those threats and ensure secure e-business transactions, security technology must play a major role in today's networks. This is justified by the fact that the security incidents in the Internet increase at an incredible rate. As shown in Figure 1.1, a recent statistics conducted by CERT (Computer Emergency Response Team) illustrates that the number of reported Internet security incidents has skyrocketed from 6 in 1988 to 137,527 in 2003 [1]. The trend is expected to sustain for some years.

Broadly speaking, confidentiality, integrity, and availability are the three aspects of network security [2]. Among them, confidentiality is the protection of sensitive information from unauthorized access or disclosure, and encryption is a major means to achieve confidentiality. Integrity concerns how to ensure that the received messages are the same as sent, with no duplication, insertion, modification, reordering, or replays while availability studies how to provide services even under adverse conditions. A recent

study shows that most of the previous works on network security focus on confidentiality, some on integrity, and few on availability [3].

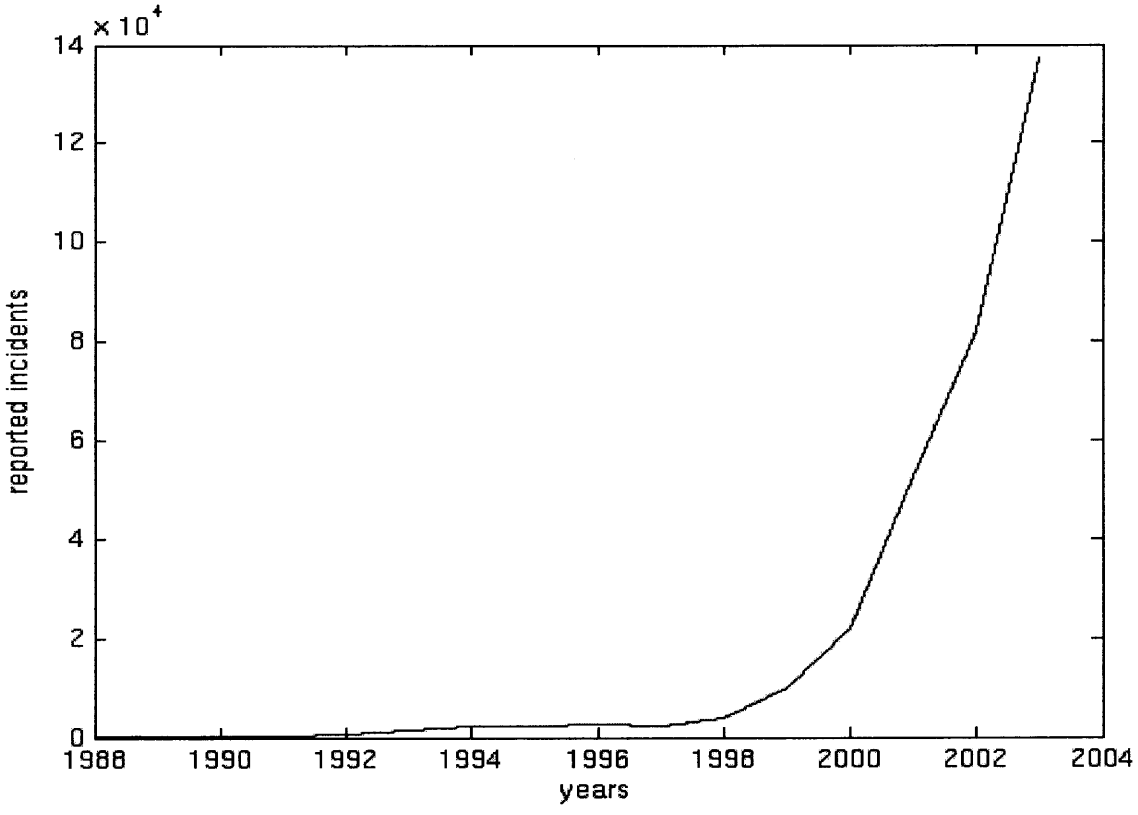


Figure 1.1 The number of Internet security incidents reported to CERT/CC [1].

The advent of the lethal denial of service (DoS) attack and its advanced variant, the distributed denial of service (DDoS) attack has quickly changed the landscape. Currently, DoS/DDoS attacks are probably the most ferocious threats in the Internet, resulting in tremendous economic and social implications/impacts on our daily lives that are increasingly depending on the well-being of the Internet. Imagine a life without the access of the Internet for a few days or weeks! You cannot run e-businesses, check e-mails, surf the web for breaking news, shop online, trade online, or play online games with peers over the Internet. What a painful experience! Some DDoS attacks did cause

outage of the networks and websites for some time. For instance, the network at University of Minnesota was shut down for more than 2 days in August 1999 [4]. More recently, Weaknees, a Los Angeles-based e-business site, was knocked out of the Internet for two weeks in 2004 [5]. Note that DoS/DDoS attacks are so detrimental that even high-provisioned sites can be easily overwhelmed. A tip-of-the-iceberg victim list includes Yahoo, CNN, Ebay, Amazon (in Feb. 2000), Domain Name Service (DNS) root servers (in Oct. 2002), and SCO group, Inc. in Dec. 2003 [6]. What makes things worse is that the frequency of DDoS attacks has been increasing rapidly. A recent study detected approximately 4,000 attacks per week (for a three-week period) against a variety of victims ranging from large companies such as Amazon and Hotmail to small Internet Service Providers (ISPs) and dial-up connections [7]. The recent survey performed by FBI/CSI showed that DoS/DDoS slaughters were the most expensive computer crimes in 2004 [8].

The prevalence and tremendous impacts of DDoS attacks have drawn serious concerns on network security and how to cope with these attacks has become an active research area. The critical issues here include 1) IP spoofing, i.e., forged source IP addresses are routinely employed to conceal the identities of the attack sources and deter the efforts of detection, defense, and tracing; 2) the distributed nature, that is, hundreds or thousands of compromised hosts are orchestrated to attack the victim synchronously. Other related issues are scalability, lack of incentives to deploy a new scheme, and the effectiveness under partial deployment.

Previous research on tackling DoS/DDoS may be categorized into four groups, i.e., intrusion prevention, intrusion detection, intrusion mitigation, and intrusion response

[9], [10], [11]. Intrusion prevention schemes attempt to prevent the occurrence of a DoS/DDoS attack. Proposals in the second group strive to detect an ongoing attack while it is raging on. Schemes in the third group concentrate on lessening the impact of an attack while those in the fourth group concern on how to respond to an attack. For instance, IP traceback endeavors to identify the attack sources so that the attacker may be located and prosecuted.

1.2 Objective

The objective of this research is to develop practical, scalable, effective and efficient mechanisms to mitigate DDoS attacks. The research focuses on IP traceback and DDoS defense techniques, which fall into the categories of intrusion mitigation and intrusion response, respectively.

Ultimately, the proposed IP traceback schemes should be

- Capable of handling large-scale DDoS attacks
- Capable of tackling spoofed marking inscribed by the attacker intentionally and spurious marking incurred by subverted routers
- Capable of minimizing the number of marked packets required for path reconstruction
- Capable of suppressing false positives
- Practical, effective and efficient

The proposed DDoS defense schemes should be

- Capable of handling a wide spectrum of DoS/DDoS attacks rather than a specified attack

- Capable of differentiating malicious attack traffic from normal ones and reacting accordingly
- Practical, effective and efficient

1.3 Organization

This dissertation is comprised of three parts. The first part includes Chapters 1 to 3. Chapter 1 introduces the problem of DoS/DDoS attacks. Chapter 2 classifies DoS/DDoS attacks while Chapter 3 evaluates existing schemes, whose deficiencies serve as the motivations.

The second part presents the proposed IP traceback schemes, namely, autonomous system based edge marking (ASEM) [12], [13], [14], which are discussed in Chapter 4. ASEM enhances probabilistic packet marking (PPM) in several aspects: (1) ASEM is capable of addressing large-scale DDoS attacks; (2) ASEM is capable of handling spoofed marking inscribed by the attacker and spurious marking incurred by subverted routers, which is a unique and critical feature; (3) ASEM can significantly reduce the number of marked packets required for path reconstruction and suppress false positives as well.

The third part introduces the proposed DDoS defense schemes, including four-color theorem based packet marking [15], and a comprehensive framework for DDoS defense [16], which are presented in Chapters 5 and 6, respectively. The salient features of the framework include (1) it is devised to tackle a wide spectrum of DDoS attacks rather than a specified one, and (2) it can differentiate malicious traffic from normal ones and respond accordingly. The receiver-center design avoids several related issues such as

scalability, and lack of incentives to deploy a new scheme. Finally, Chapter 7 draws conclusive remarks and discusses future works.

CHAPTER 2

CLASSIFICATION OF DOS/DDOS ATTACKS

In Chapter 2, diverse DoS/DDoS attack patterns are described and categorized. DoS attacks are introduced in the first section, followed by DDoS attacks in Section 2.2. Section 2.3 attempts to classify diverse assault patterns in terms of exploited protocols, attack rates, communication channels, and the use of reflectors.

2.1 DoS Attacks

The goal of a DoS attack is to disrupt or degrade the operation of the victim so that its regular clients cannot obtain required services. Different from traditional threats such as viruses, worms, and Trojan horses, whose purposes are to access, expose, or alter confidential information, the victim's system is not penetrated by any unauthorized third party, nor any sensitive information is disclosed to the public during a DoS attack. There are three types of DoS attacks. The first type of DoS attacks sends a single malformed packet (or several crafted packets) that crashes the victim system, such as the "land" assault and "ping of death" attack [17], [18], [19]. The second kind of DoS attacks is a form of amplified DoS attacks such as "smurf" [18]. A smurf attack bombards packets to the broadcast address of misconfigured networks so that all hosts inside these networks will respond to the forged requests, thus amplifying the attack effects and swamping the victim. The third sort of DoS attacks aims to consume the limited resources available on the target system. This kind of DoS assaults just hampers the normal operation of the

victim by mounting tremendous useless packets to use up any available resources of the victim so that the legitimate service requests cannot be satisfied.

So, how does this happen? In the case of a single packet assault, the intruder first needs to find some vulnerability in the target machine or inside the target applications and then sends a malformed packet that takes advantage of the given vulnerabilities to cripple the target. The vulnerabilities often exploited include buffer overflow, remote procedure call (RPC), and insecure protocols. For example, the "ping of death" attack sends an ICMP packet with an illegal payload (larger than 64K bytes) to lock up or reboot windows operating system due to buffer overflow. This kind of attacks can be defeated by updating the system in time with proper patches.

For the second kind of attacks, the attacker mounts a vast number of requests with spurious source IP addresses, using the IP address of the victim instead, to the broadcast address of a network. Unless properly configured, all hosts attached to the network will generate a response packet to the "supposed" source, the victim. The amplification factor can be up to 254 in a Class C network, and up to several thousands for a moderately populated Class B network [20]. This kind of attacks can be addressed by carefully configuring the network one owns and filtering such requests from other networks.

The third option is to clog the target. Typical resources that get drained in a DoS attack are network bandwidth, CPU cycles of the target host (usually a server), and specific TCP/IP protocol stack structures such as the fragmentation buffer and TCP SYN buffer. A perpetrator can launch a DoS attack by bombarding the specified target, termed victim, with a massive number of superficially normal but really useless packets, thus consuming the resources available to the clients of the victim. To that end, the cracker

first needs to recruit a host as the unwitting accomplice. By stealthy scanning and penetrating a vulnerable host with unprotected ports, insecure services, and other weakness over the Internet, the intruder will gain root access and can install daemons and plant malicious codes in the intermediate machine. Once installed, the daemons then quiescently listen to network traffic and wait for commands from the hacker's master machine to launch the DoS assaults. Note that in this case, the selected intermediate host must be more powerful than the victim with respect to bandwidth and CPU rate so that it can generate and send more packets than the victim can handle.

Known DoS attack tools include "land", "ping of death", "teardrop", "boink/bonk" [21], and "smurf". In a "land" attack, the cracker crafts packets with the same source and destination IP address and port, that confuse the host's operating system, and thus crashing the system. "Boink/bonk" attacks are new versions of "teardrop" attacks that use overlapped fragments to freeze a host.

2.2 DDoS Attacks

DDoS attacks are the advanced variants of DoS attacks. Rather than using a single intermediate host in a DoS attack, hundreds or even thousands of intermediate hosts are recruited and employed to assault the victim in a DDoS attack. In so doing, two benefits may be obtained by the attacker. First, the limitation that the accomplice is more powerful and high-profiled than the victim is eliminated due to the increasing number of useful hosts. Second, diverse attack paths may exist, thus effectively deterring the efforts of efficient detection and filtering. In contrast, there is only one attack path in a DoS attack.

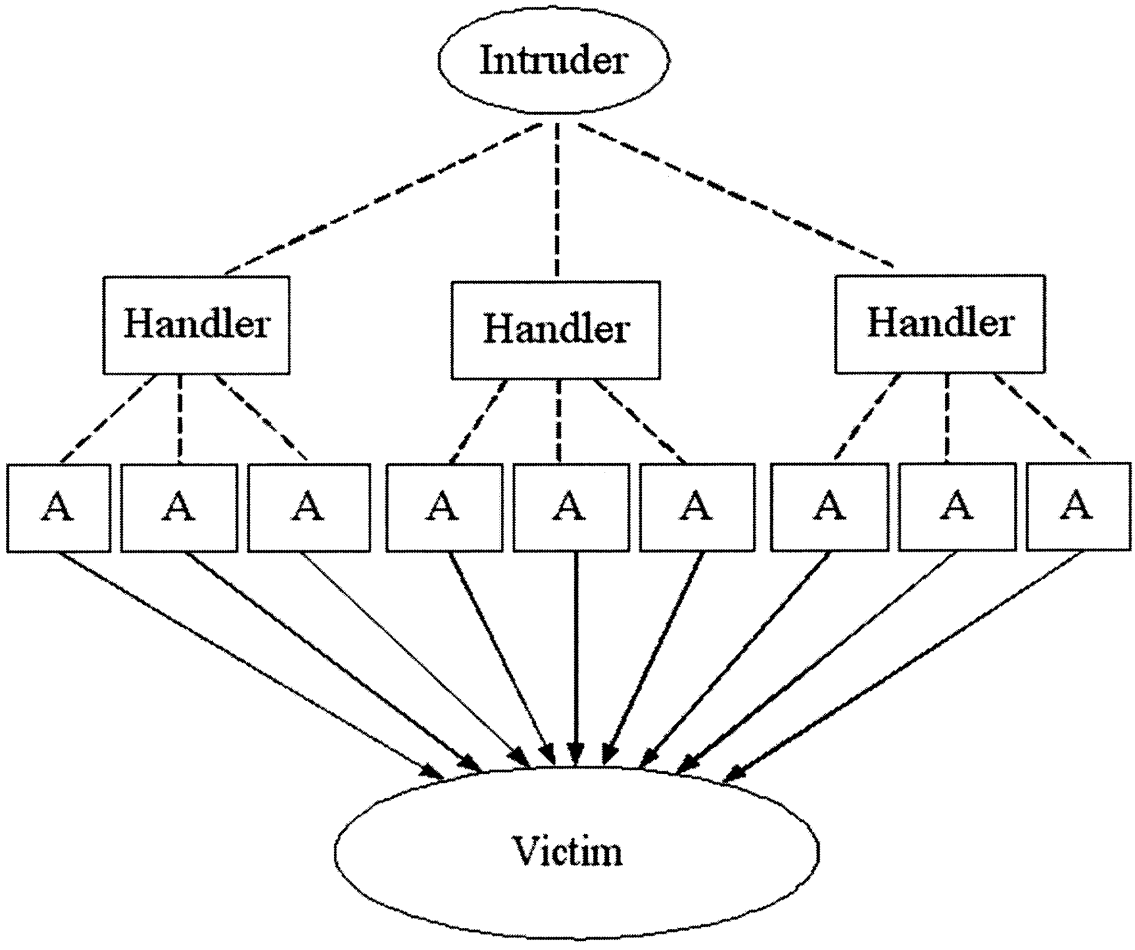


Figure 2.1 A DDoS attack network, where A stands for an agent.

Figure 2.1 sketches the DDoS attack network. For easy management, the intruder usually organizes the attack network by assigning different roles for different hosts. An intermediate host is called a master or a handler if it is the head of a group of hosts. The clients or agents lie on the lower level of the attack network, and are under the control of a handler. When the perpetrator decides to mount an attack, the command message goes from the attacker's main machine to the handlers, and then the handlers distribute the command to each agent. Two kinds of traffic exist inside the attack network. One is the

control traffic (dotted lines in Figure 2.1), flowing from the attacker to the handlers, and from the handlers to the agents. Another is the real attack stream (solid line in Figure 2.1), from the agents to the victim.

Known DDoS attack tools include Trinoo (also called Trin00), Tribe Flood Network (TFN), TFN2K, Stacheldraht, and trinity. Detailed analysis of these attack tools can be found in [9], [22], [23].

2.3 Classification of DoS/DDoS Attacks

To better understand DoS/DDoS attacks, different DoS/DDoS attacks are categorized in this section according to the exploited protocols, attack rates, communication channels, as well as the use of reflectors.

2.3.1 Classification Based on Exploited Protocols

The first two sections introduce different kinds of DoS/DDoS attacks. From the perspective of protocols used, some attacks exploit UDP, and some take advantage of ICMP while most utilize TCP. More sophisticated attacks employ TCP, UDP, and ICMP in one attack. Note that the combination of several protocols in a DDoS attack inflicts a severe challenge to DDoS defense because most proposed schemes are not generic enough to address a wide spectrum of attacks. Table 2.1 shows an incomplete list of those attacks.

Table 2.1 Diverse DDoS Attack Tools

Protocols used	DoS/DDoS attack tools/names
TCP only	SYN flood, RST flood, mstream
UDP only	trinoo
ICMP only	Ping of death, flood pinging, smurf
Combinations of TCP, UDP and ICMP	TFN, TFN2K, shaft, MIX, Stacheldraht, trinity v3

2.3.2 Classification Based on Attack Rates

While most DDoS attacks employ the strategy of flooding assault such as SYN flood, UDP flood, and ping flood, a novel DoS attack is low-rate based [24](see Figure 2.2). This kind of attacks exploits the inbuilt congestion control mechanism of TCP. Different from most other DoS/DDoS attacks, the attacker launches attack packets sporadically so that the average rate is low. The time period of the attack is about 1s, equivalent to the RTO (Retransmission Time Out) of a TCP connection. The burst rate is around the bottleneck capacity of a link so that the link is totally jammed. Since the bottleneck link is severely congested, packets traversing the link will most likely be dropped. Note that TCP has two levels of retransmission mechanisms. If the sender receives 3 duplicated ACKs in a short period, it infers that some outgoing packets are lost and will retransmit these packets. In a longer interval, if the sender does not receive any response from the receiver side, the timer in the sender side will expire, that forces the sender to enter the stage of slow-start. Frequent slow-start effectively diminishes the throughput of a connection.

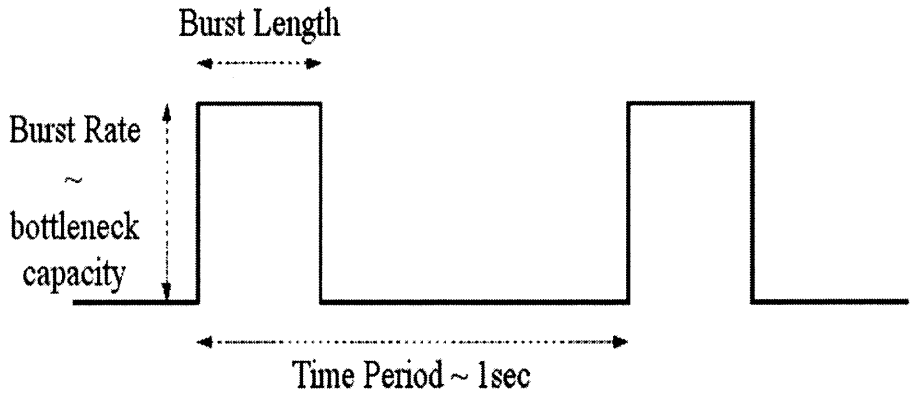


Figure 2.2 Traffic model of low rate DoS attacks.

2.3.3 Classification Based on Communication Channels

Two types of communication channels exist in the current DDoS attacks. One is shown in Figure 2.1, where the control flows traverse from the attacker's main machine to the handlers and then from the handlers to their agents. Another channel is IRC (Internet Relay Chat) networks [9], [18]. IRC networks are the closed networks that allow anonymous logins, an attractive feature to the intruder. In this kind of attacks, the functions of handlers are normally implemented in IRC servers while the agents are called bots. A bot, derived from robot, is a client program running in the background on a subverted machine and waiting for certain strings to show up in an IRC channel. These strings are encoded commands that the bots are going to execute. The bot first checks the specified default channel that is hard-coded and password protected to locate the current control channel. It will then jump to the control channel. Once in the current control channel, the bots are ready to execute the commands from the attacker.

The advantage of using IRC channels is manifold. First, anonymity supported in IRC is a good shelter for the intruder. Second, the attacker and the bots connected to the

IRC servers like any other IRC clients. As a result, the communication between the attacker and the bots is hidden among the chats between other users. Third, since a bot is often extended from its original design to contain the malicious attack code, DDoS communication does not generate anomalous event that can trigger the IDS (intrusion detection system). Figure 2.3 depicts an IRC-based DDoS network.

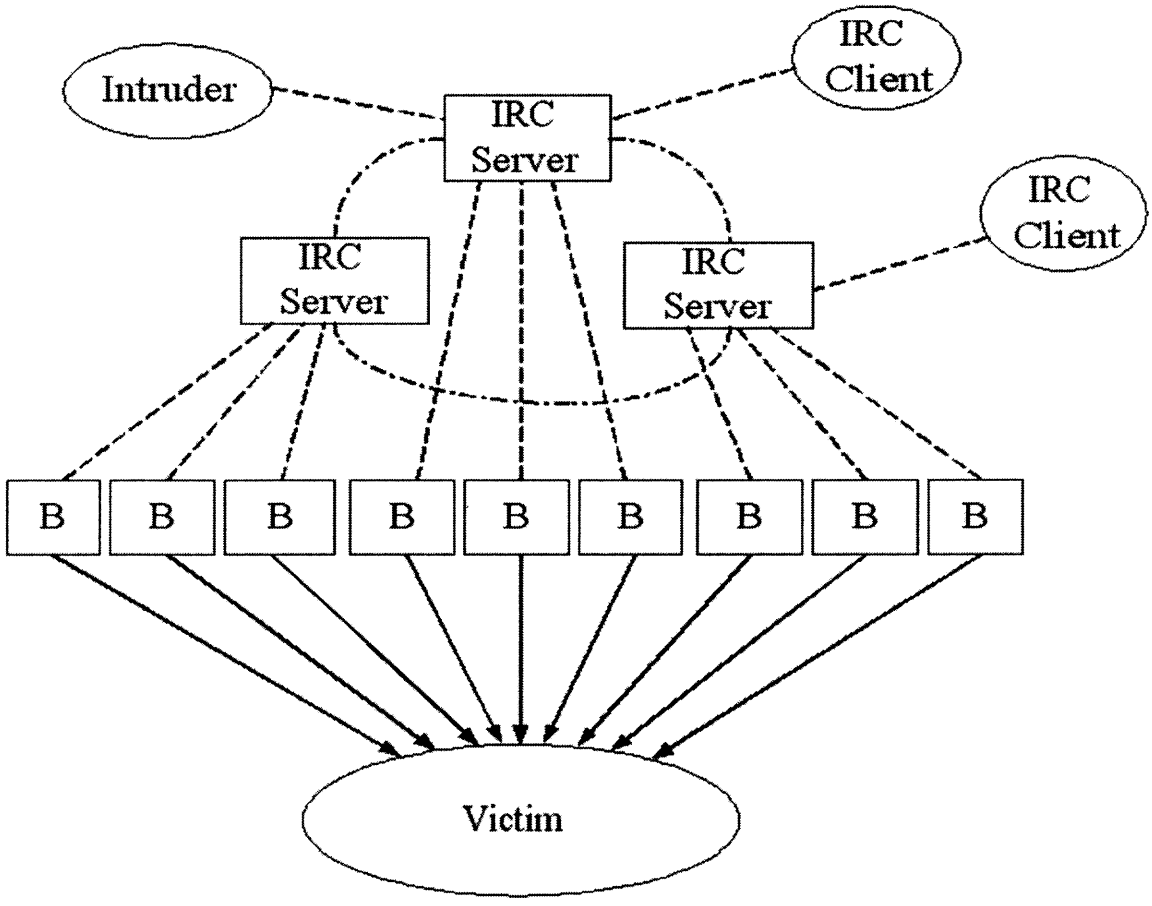


Figure 2.3 IRC-based DDoS attack network, botnet, where B stands for a bot.

2.3.4 Classification Based on Use of Reflectors

The perpetrator may further conceal the trait of an attack by using reflectors [25], [18]. A reflector is any host over the Internet that will generate a response upon receiving a request. These requests may be a ping packet, a DNS query, a Web request, and so on.

By crafting the source IP addresses of the requests, the attacker can lead tremendous responses to the victim at will. Figure 2.4 depicts a reflective DDoS attack and such an attack in the real world was reported by Gibson in 2002 [26].

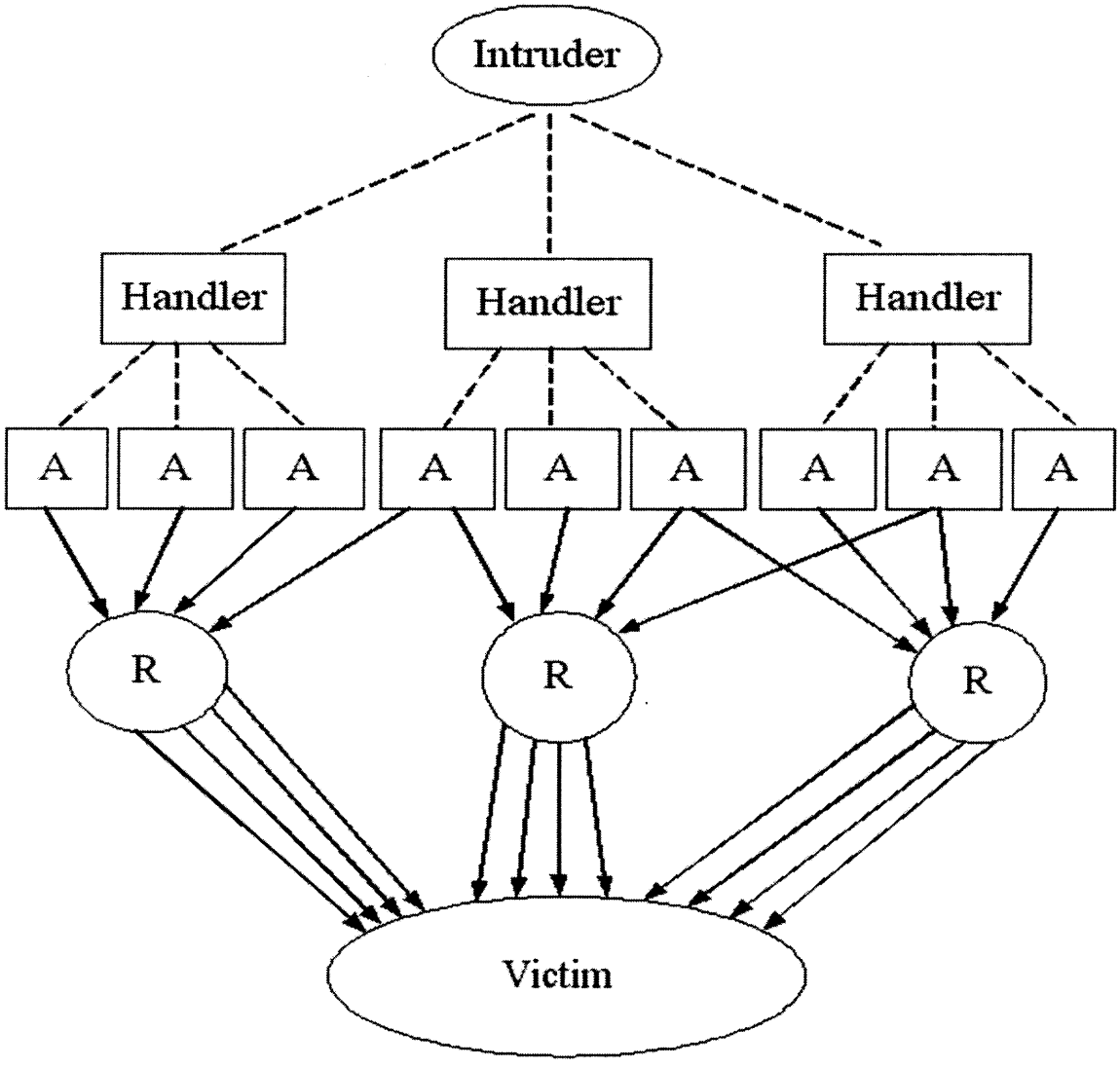


Figure 2.4 Reflective DDoS attack model. Here, R and A represent a reflector and an agent, respectively.

Reflective DDoS attacks introduce another level of indirectness, which represents a big challenge to IP traceback because coordination between different administrative domains is indispensable. However, no efficient and reliable trust relationships are

maintained in the Internet nowadays. How to effectively cope with reflective DDoS attacks is still an open issue.

CHAPTER 3

PREVIOUS WORKS

In this chapter, previous works on handling DoS/DDoS attacks are summarized. In general, previous works may be categorized into four groups, i.e., intrusion prevention, intrusion detection, intrusion mitigation, and intrusion response [9], [10], [11]. Since this dissertation focuses on DDoS defense and IP traceback, schemes fall in the categories of intrusion prevention and intrusion detection are only briefly reviewed. Schemes in intrusion prevention, intrusion detection, intrusion mitigation, and intrusion response are presented in Section 3.1, 3.2, 3.3, and 3.4, respectively.

3.1 Intrusion Prevention

Intrusion prevention schemes are those that attempt to prevent the occurrence of a DoS/DDoS attack.

Ingress and egress filtering are well-known attack preventive mechanisms. In ingress filtering [27], the ingress edge router of an ISP checks the source IP address of each incoming packet to determine its operation, forwarding as normal or dropping it. For a specified interface, only packets with the valid prefixes may pass the router and be forwarded to the rest of the Internet; all other packets are dropped. This mechanism is simple and may significantly diminish the chance of IP spoofing if globally deployed. However, the requirement for global deployment is infeasible currently given the extreme complexity of the Internet. Another issue here is that ingress filtering may block the

operation of mobile hosts. Egress filtering [28] is similar in principle, but the examining operation is conducted at the egress point of a network rather than at the ingress point of an ISP. Besides the cons of ingress filtering, another issue is that the ISP may lack sufficient incentives to do so because egress filtering is helpful to the security of other networks rather than the ISP's own network.

Mirkovic *et al.* proposed a source-end DDoS defense system, D-WARD [29]. Another promising attack preventive method is Secure Overlay Service (SOS) [30], which greatly lessens the risk of being attacked for the protected systems by limiting the number of access points. Only packets from these access points may traverse the overlay network while all others are discarded. History-based filtering proposed by Peng *et al.* filters packets based on the fact that most users frequently visit a limited number of sites [31]. Therefore, a packet may be admitted only if its source IP address can be found in the history-visit table maintained by the destination. Park and Lee [32] presented a route-based packet-filtering scheme that uses the routing information of Autonomous Systems (ASs) to identify lethal streams. Lakshminarayanan *et al.* [33] proposed to use the Internet Indirect Infrastructure so that the receiver may decide which packet it prefers to accept and at which rate. Other works include capability based network that requires a token-like capability before any session starts [34].

3.2 Intrusion Detection

Intrusion detection has been an active research field for a long time. Generally speaking, two types of mechanisms are employed, namely, signature-based detection and anomaly-based detection. Signature-based detection is efficient and fast to detect known attacks.

However, it cannot handle unknown attacks, which may be the variants of an existing attack or a brand new attack. On the contrary, anomaly-based attack is slow but can address novel attacks as well.

DDoS detection systems include MULTOPS [35], SYN flood detection [36], and spectral analysis [37]. MULTOPS identifies denial of service attacks by monitoring the packet rates in both directions of a link. The basic assumption is that the packet rates at both directions between two hosts are proportional for normal operations. A remarkable disproportion between the packet rate in the up and the down direction of a link may indicate an attack.

Wang *et al.* [36] used the CUSUM algorithm (non-parametric Cumulative Sum) to discover DoS attacks. Similar to MULTOPS, the packet ratio of SYN to the sum of RST and FIN is used to recognize an ongoing attack. The assumption they used is that a statistical change will be observed once an attack happens.

Cheng *et al.* [37] assumed that DoS attack traffic possesses different statistical properties in comparison with the normal ones. By observing whether a flow shows strong periodicity around its round-trip time (RTT) in terms of the number of packets in both directions, their scheme may detect a DoS attack.

3.3 Intrusion Mitigation

DDoS mitigation schemes include PacketScore [38], hop-count based packet filtering [39], IP-traceback based filtering [40], pushback [41], RED-PD [42], puzzle-auction based scheme [43], path identification (Pi) method [44], [15], and Honeypot [45].

Kim *et al.* [38] employed a statistical method to score incoming packets and determine the correct operation, discarding or forwarding. The filtering decision is based on two factors, the current score distribution of incoming packets and the load of the protected systems. No filtering occurs as long as the load of the protected system is less than a threshold.

Jin *et al.* [39] proposed to use the value of the TTL field in the IP header of each packet as a clue to filter packets. Their work is based on the observation that most current operating systems (Unix, Linux, Solaris, Mac, and Windows) specify only a few initial TTL values.

Sung and Xu [40] proposed to use PPM (Probabilistic Packet Marking) marking information for packet filtering. In their scheme, most marking information is used for path reconstruction while the rest for packet filtering.

Ioannidis and Bellovin [41] proposed a router-based defense scheme called pushback to punish the attack sources. By identifying malicious aggregates, the routers can limit the rate of such aggregates. Also, upstream routers are required to cooperate to throttle the bad traffic once it is identified.

Mahajan and Floyd [42] proposed RED preferential dropping (RED-PD) to regulate high rate flows. Their scheme uses the packet drop history at the routers to identify high-rate flows and preferentially drop packets of these flows upon congestion.

Wang *et al.* [43] proposed to limit the capability of a host to send attack packets by partially consuming the sender's resources in solving puzzles. When many hosts compete with each other to establish a connection with a server, the server can ask each

of them to solve a puzzle. The host that is willing to solve the most difficult puzzle is given the highest priority to set up a connection.

Yaar *et al.* [44] proposed a new scheme that uses path information as a hint for the victim to filter attack flows. The authors further extend their scheme from the perspective of practicality [15].

Besides these approaches, the honeypot technique is used to study the attack strategy of the attack tools, and locate the sources of a DoS/DDoS assault [45]. A survey of DDoS defense can be found in [46].

3.4 Intrusion Response

Once an attack is identified, the next step is to locate the attack sources and block the attack traffic accordingly. IP traceback is the technique to identify the attack sources. Up to date, a vast number of traceback schemes have been proposed. Among them, probabilistic packet marking (PPM) [47] and Hashed-based IP traceback [48] are well known in the community. To better comprehend and capture the properties of disparate traceback approaches, we first classify these schemes from multiple aspects, and then define several evaluation metrics to assess the pros and cons of each of them.

3.4.1 Classification

As shown in Figure 3.1, five aspects are selected to classify existing traceback schemes into different categories. They include the basic principle, the processing mode, the functionality supported, the location, and the requirement of an extra infrastructure.

The schemes illustrated in Figure 3.1 include Probabilistic Packet Marking (PPM) [47], Source Path Isolation Engine (SPIE, also called hash-based traceback) [48], ICMP

traceback (iTrace) [49], Algebraic-based Traceback Approach (ATA) [50], Deterministic Packet Marking (DPM) [51], and an overlay-based solution (Center-Track) [52].

3.4.1.1 Classification by the Basic Principle. According to the basic principle, most of the existing traceback schemes may be roughly categorized into two groups: marking and logging. In the logging schemes, routers record some information of traversing packets so that it may be verified whether suspected packets have been forwarded by a specific router or not. In the marking schemes, a portion of or all routers along an attack path from an attack source to the victim write some information of these routers into the packets so that the attack path(s) may be recovered by the victim, even though the source IP addresses of attack packets are spoofed. The marking information may be inscribed in the same attack packets (which is called inbound marking) or extra ICMP packets (called outbound marking) [49]. Inbound marking does not require extra bandwidth while the number of bits that may be used for marking is rather limited. Using the option field to store marking information is not a preferential choice because it triggers a significant delay in processing the marked packets at routers. On the contrary, there are far more bits available for outbound marking than inbound marking, that may mitigate false positives and greatly reduce the number of marked packets required for reconstruction. There are two main shortcomings in outbound marking. Outbound marking needs extra bandwidth that may further aggravate the performance of the network being attacked. A new ICMP message must be introduced into the Internet, and there shall not be any ICMP filtering. Otherwise, the ICMP message may be blocked due to ICMP filtering.

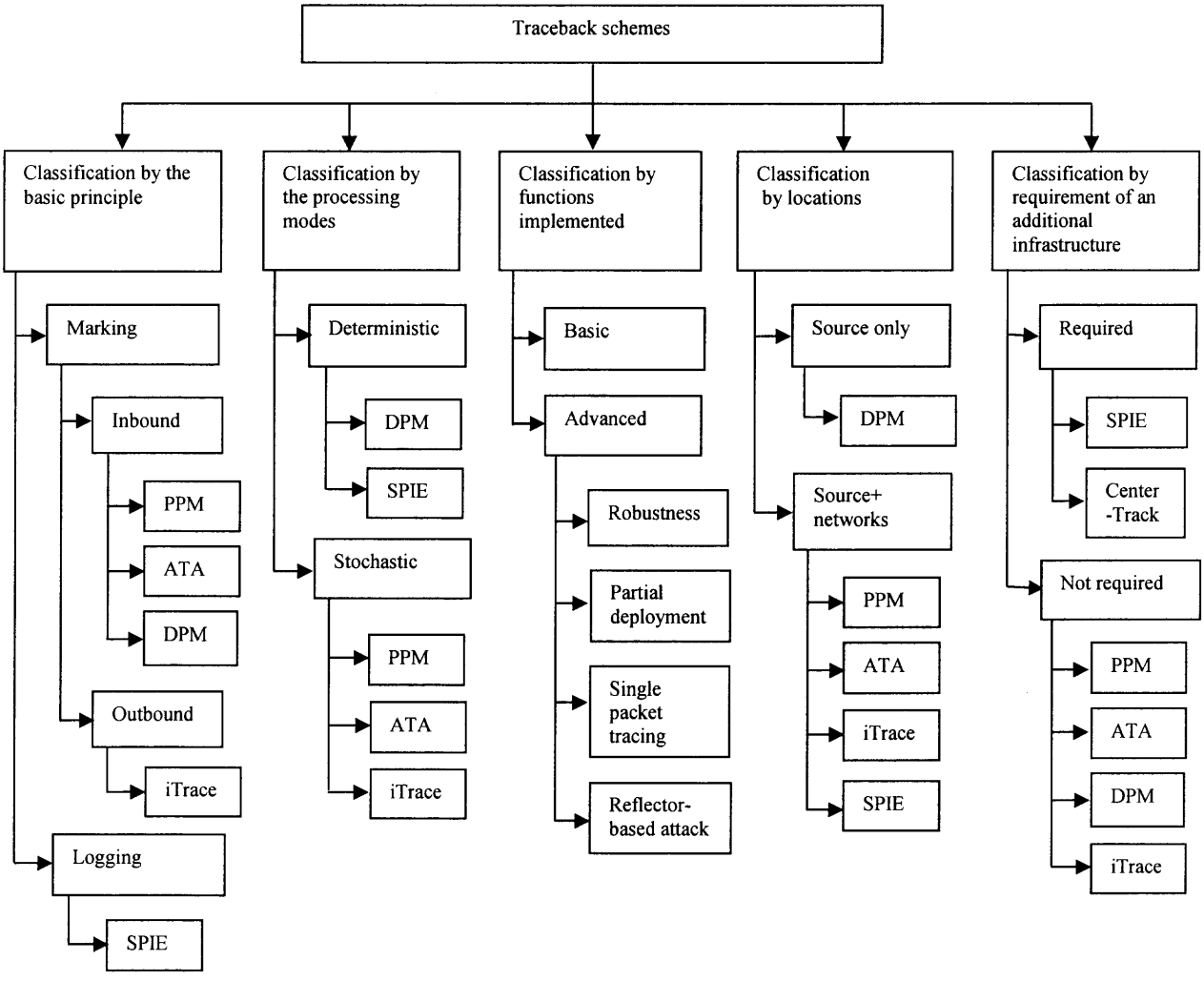


Figure 3.1 Categorizing IP traceback schemes.

Current traceback schemes that are based on marking include variants of PPM, ATA, DPM, and schemes that use ICMP messages, such as iTrace. Instances of logging schemes include SPIE and its variant [53].

3.4.1.2 Classification by Processing Mode. From the viewpoint of the processing mode, traceback schemes may be categorized into two groups, deterministic or stochastic mode.

The deterministic mode implies that every packet has to be processed, either marking or logging. DPM is an example of deterministic marking while SPIE deterministic logging. In comparison with deterministic mode, more stochastic schemes have been contrived. Three well-known examples are PPM (and many of its variants), ATA, and iTrace. Obviously, deterministic mode incurs more processing overhead. However, it may perform single packet tracing. Furthermore, deterministic processing mode may be indispensable for the more advanced security service, such as non-repudiation. Probabilistic mode is helpful to reduce bandwidth and processing overhead at the expense of increased complexity for path reconstruction at the victim.

3.4.1.3 Classification by Functions Implemented. There is no panacea in IP traceback. Different tracing schemes make different assumptions and strive to solve different problems. In general, each tracing scheme has to make some tradeoffs between the performance and the overhead. In the marking schemes, factors that shall be taken into account include marking every packet or marking at a certain probability, the number of bits used for marking, the place to store the marking information, and parts of the networks (the routers, the victim, or both) that bear the incurred overheads. In the logging schemes, the issues to be addressed include the content to be logged, the frequency of logging, the place to store the logging information, and an efficient approach to communicate between the victim and the routers where the logging information is stored.

The functionalities that a tracing scheme supports may be further divided into two groups, basic and advanced functions. Obviously, the basic function is the ability to trace to the attack source under a DoS attack or hundreds of sources under a DDoS attack. The

advanced proposals consider the following issues: the security of the scheme itself (e.g., support of authentication); the ability of tracing a single packet; the capability of tracing a reflector-based DDoS attack; the capability of being effective under partial deployment.

3.4.1.4 Classification by Locations. From the perspective of locations, existing traceback schemes may be divided into two types, i.e., those that inscribe information into the packets near the source, and in the network, respectively. DPM is an example that performs marking near the source (edge routers closest to the source). Most schemes work with the cooperation of the victim and the network. That is, the routers (some or all) in the network perform certain processing (marking or logging), either stochastically or deterministically, and inscribe required information into the packets. When these processed packets arrive at the victim, the victim may reconstruct the attack paths from the embedded information.

An associated issue with locations is whether the victim can reconstruct each path entirely or partially. Only recording the information of a single point is a special case of partial path information. Clearly, only single point information for each path may be provided for schemes performing marking at edge routers. In so doing, the most valuable information—the first edge router from which attack packets being mounted may be readily determined. Another benefit is that the victim is greatly relieved from the heavy computational and storage burden. However, the marking information may not be robust enough because of the lack of verifiability. If the first edge router along an attack path is compromised, no additional clue may be exploited. A good tradeoff between the computational burden and the reliability is to record partial path information, e.g., storing the path information of traversing Autonomous Systems (ASs).

3.4.1.5 Classification by Requirement of an Extra Infrastructure. The current tracing schemes may also be differentiated according to whether an extra infrastructure is required. Here, we focus on additional facilities that are required for the sake of tracing rather than normal packet forwarding. An extra infrastructure refers to some additional facilities such as the Tracking Router (TR) used in CenterTrack [52], and SPIE Collection and Reduction Agents (SCAR) used in SPIE [48]. In general, an extra infrastructure implies more financial investment and more management overhead; this is not attractive to the Internet Service Providers (ISPs). Note that although all traceback schemes expect certain modifications or function extensions to current facilities, especially routers, these modifications or extensions to existing devices are not considered as an extra infrastructure here. The instances that do not need an extra infrastructure include variants of PPM, iTrace, and DPM.

3.4.2 Evaluation Metrics

A number of metrics may be used to evaluate the performance of disparate traceback schemes, such as the minimum number of marked packets required for path reconstruction, processing burden, bandwidth overhead, memory overhead, robustness, scalability, ISP involvement, and so on [13]. This dissertation mainly assesses disparate tracing schemes from the practical perspective. The following criteria are thus considered: the minimum number of marked packets required for path reconstruction, the computational burden, effectiveness under partial deployment, and robustness.

The computational overhead—The computational overhead depends on several factors, such as processing packets stochastically or deterministically at the routers, inbound marking or outbound marking, and under a DoS or a DDoS attack. A good design will strive to minimize the computational burden on the victim. If an overwhelming computation is required, it may take the victim too long time to complete the path reconstruction process; this is definitely not a preferential choice.

Effectiveness under partial deployment—The distributed nature of the Internet renders deployment a big issue. First, the ISPs may lack incentives to deploy a scheme. Deploying a new scheme may imply more investment, more operational costs, and higher management complexity. Second, it may take a long time for a new scheme to be totally adopted in the Internet. Therefore, the effectiveness of a traceback approach under partial deployment is an important factor to be considered. When a scheme is devised, issues related to partial deployment shall be taken into account.

Robustness—In terms of robustness, we refer to the ability of an approach that can perform tracing reliably even under adverse conditions. When the stochastic mode is selected, it is critical to effectively process packets so that all information required for path reconstruction is reliably conveyed to the victim and that false positives are efficiently thwarted. Besides false positives incurred in processing (marking or logging) and reconstruction, it may be possible that some sophisticated attackers embed forged marking to amplify the false positives. Subverted routers are another issue to be addressed.

3.4.3 Evaluations of Schemes

According to the above criteria, we evaluate the following schemes: variants of PPM, iTrace, DPM, SPIE, and CenterTrack.

3.4.3.1 Variants of PPM. Among all previous works, PPM is a promising one which possesses several attractive features such as low router overhead, support of incremental deployment, and "post-mortem" tracing. Up to date, many variants of PPM have been developed [40], [50], [54], [55], [56], [57], [58], [59]. The work here is also based on PPM.

A. Basic PPM

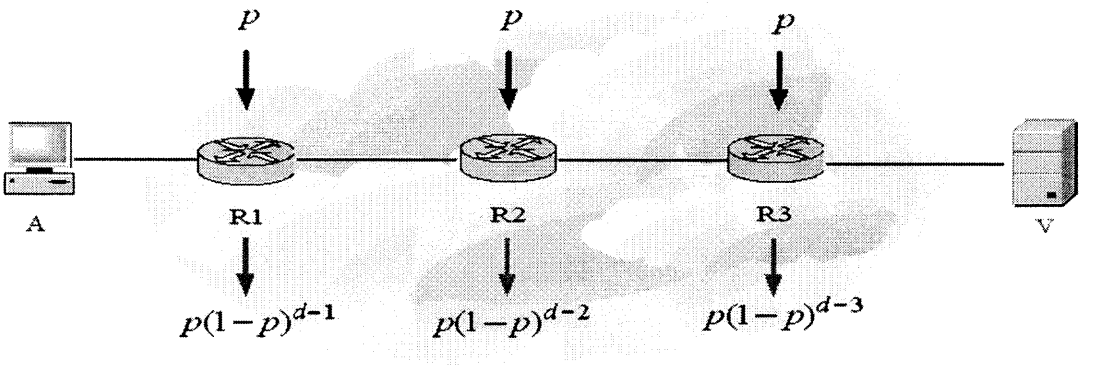


Figure 3.2 Marking probability with respect to the victim, where d=3.

PPM was first introduced by Burch and Cheswick [60], and cleverly re-developed by Savage *et al.* [47] later. The basic idea of PPM is simple. Suppose that one attack flow from an attack source to the victim traverses routers R_1, R_2, ..., R_d in order (see Figure 3.2 where d=3). Denote p as the marking probability of each router. For router R_i ($1 \le i \le d$), with respect to the victim, the probability of the current packet marked by R_i

is $p(1-p)^{d-i}$, which is different from p [57], [61]. The reason is that subsequent routers may "re-mark" packets already marked by previous ones, thus overriding marking information of previous routers. Generally speaking, the closer a router is to the victim, the more likely its marking survives. Therefore, the first router is the "weakest" part of the whole path [61].

To handle DDoS attacks, the edge-sampling method was proposed. The detailed marking procedure at each router is depicted in Figure 3.3, in which the attack packets traverses routers R_1, R_2, and R_3. Each router makes the decision whether to mark the current packet or not independently. At router R_1, the upper box shows the case that R_1 marks packets, and the unmarked case is presented in the bottom box. The probability of each case is also shown. At router R_2, four cases may arise. The upper two boxes show the scenario that packets have been marked by router R_1. Of these two boxes, the upper one stands for the scenario that router R_2 "re-marks" these packets while the bottom one does not. Similarly, the bottom two boxes represent those packets that have not been marked by router R_1. Of these two boxes, the upper box represents that packets have been marked by R_2, while the bottom one not marked by R_2. Using the similar procedure, the final result (what the victim receives) can be easily obtained. In Figure 3.3, "^x" represents that x is an exponent. Here, we do not attempt to calculate the final result of the probability for each case to clarify the marking and "re-marking" procedure. For instance, the probability of $p(1-p)p$ stands for the case that router R_1 marks the packets, and router R_2 does not while R_3 re-marks these packets. Note that though 2 cases may arise at router R_1, 4 cases at R_2, and 8 cases at R_3, the marking results may be

the same. For example, the boxes 1, 3, 5, and 7 at router R_3 own the same marking information that can be combined.

After having combined the results with the same marking information, the victim will see four different marks. The victim first locates the closest router, R_3, by looking at the packets whose dist field has a value of 0. Next, from the packets with dist=1, it can locate R_2. To save space, a new field called addr is used instead of the start field and end field shown in Figure 3.3, and its content is the result of executing the exclusive or (XOR) operation over the start and end fields. From the first step, we obtain the value of R_3; from the second step, we determine the value of $(R_2 \oplus R_3)$ [47]. Since $R_3 \oplus (R_2 \oplus R_3)=R_2$, R_2 may be located by using XOR. The procedure is repeated until the farthest router is reached.

B. Analysis of PPM

The above path reconstruction procedure works well if the victim is under a DoS attack (i.e., a single attack source). However, more common scenarios today are large-scale DDoS attacks where hundreds or thousands of attack sources are concerted to assault the victim synchronously. Under these cases, PPM has the following deficiencies.

1. Heavy computational load for path reconstruction. When there are 25 attack sources, path reconstruction will take days and thousands of false positives may be generated [54]. Currently, a DDoS attack may orchestrate thousands of zombies. As a result, the victim will never be able to complete the path reconstruction procedure. The daunting computational burden is caused by combinatorial explosion, which is originated from the insufficient number of bits for marking.

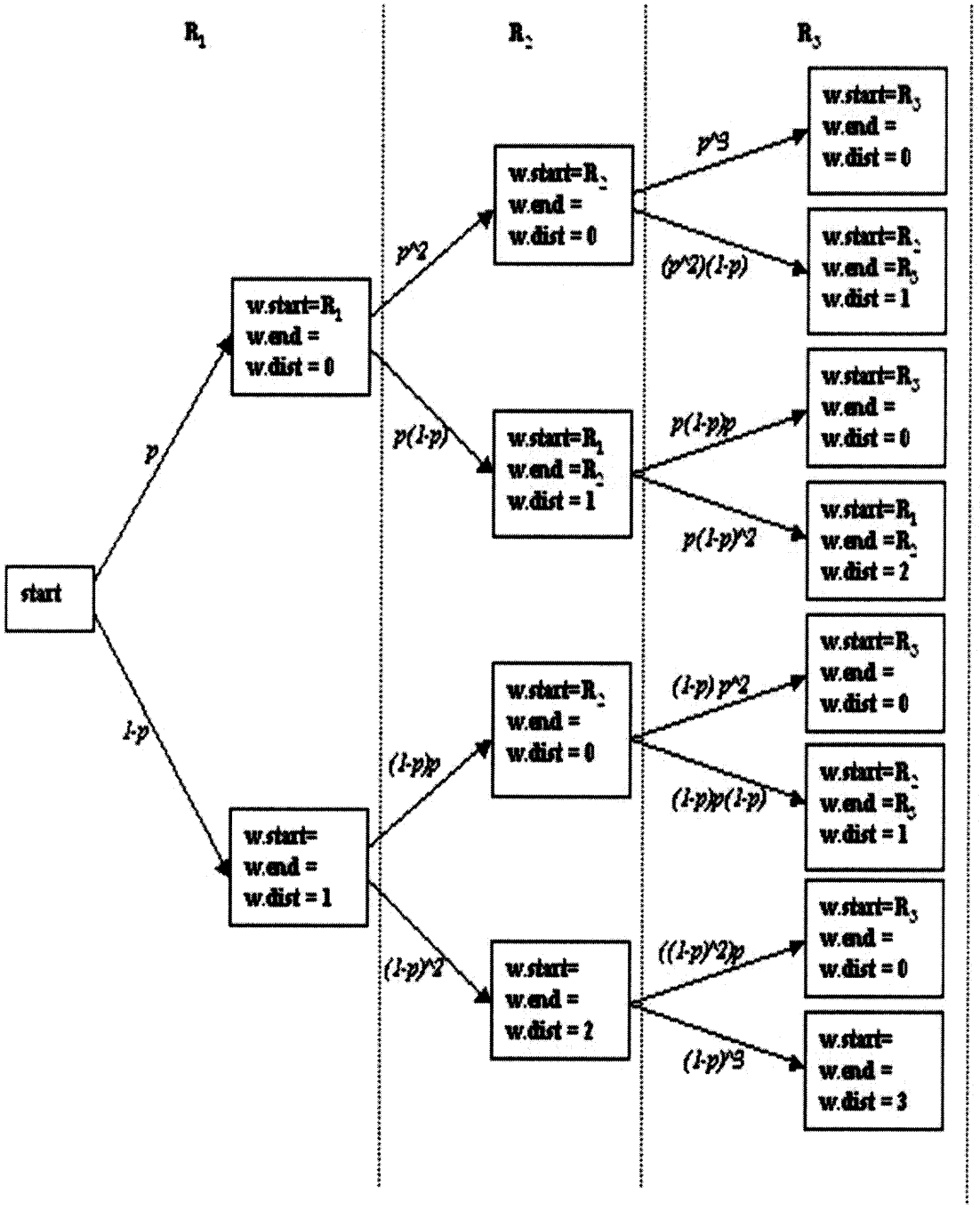


Figure 3.3 PPM marking procedure, where w.start records the information of a router R which marks packet w, w.end stores the information of the downstream neighbor router of R (the other endpoint of an edge), and w.dist stands for the distance between R and the victim.

2. High false positives. One source of false positives is limited marking bits. The IP address is composed of 32 bits while the length of the ID field where the marking is stored is only 16 bits. Another is rooted in the reconstruction algorithm. When there exist a large number of attack paths, the victim may be confused because many routers along different paths may be at the same distance to the victim.

3. Spoofed marking. The attacker may inscribe spurious marking in such a way that the victim receives more packets with forged marking information than those with the correct one [61], [62]. As a result, the victim will have little opportunity to discover the attack paths.

4. Unawareness of the path length[1] in advance. When a router decides to mark a packet, it has no idea of the path length, *d*. Therefore, it is incapable of setting *p* to the optimal value *1/d* [14], [47]. One possible choice is to use the recommended value, e.g., 0.04 [47]. If there are many attack paths with disparate lengths, simply using a predetermined marking probability for all paths may seriously degrade the performance.

5. Subverted routers. Few measures have been taken to defend against malfunctioned or subverted routers. Some subverted routers may be triggered by misconfigurations, and others may be resulted from internal vulnerabilities [63]. Note that subverted routers may also generate spoofed marking. Up to now, few schemes may contain this issue.

6. Ineffectiveness to address large-scale DDoS attacks [54], [61]. Two steps are required for path reconstruction in PPM. One is the recovery of the 32-bit IP address of each router from several packets. Another is the recovery of the whole path. In PPM, 8 packets marked by the same router need to be identified and combined to resume the IP address of that router. Since there exists no hint except distance field, it is difficult for the victim to identify which marked packets are from the same router when many routers are located at the same distance from the victim. Similarly, the victim cannot identify packets that are launched from the same attack source and traverse the same path because no clue is provided in PPM, thus seriously hampering the recovery of that path.

C. Possible Solutions

Song *et al.* [54] proposed an advanced and authenticated PPM based on the assumption that the victim knows the mapping of the upstream routers. Their scheme can mitigate Problems 1 and 2, and effectively address Problem 3 as well. Recently, Yaar *et al.* [55] further improved Song's work by eliminating the requirement of knowing mapping of the

---

[1] In this chapter, path length is defined as the number of routers eligible to conduct marking in between the attack sources and the victim. In PPM, all routers along an attack path can mark packets passing by, and therefore all routers along the path are eligible. In

upstream routers. Another method to partially thwart Problem 1 is to use varying marking probability at each router [56]. The exact value of marking probability at each router depends on the hop counts between the current router and the victim. A recent work done by Tseng *et al.* [57] used counters to complement the loss of marking information from upstream routers. Their scheme may address Problems 1 and 3, and decrease false positives. Goodrich [58] proposed to add linkage information, a hash function of the IP address of the current router, which can be used as a guide for the victim to recover an effective IP address. Aljifri *et al.* [59] proposed to use header compression to lessen the number of marked packets for path reconstruction. Note that Problems 1 and 2 are related. In general, an approach that may alleviate the computational overhead is helpful to moderate false positives.

Problem 4 may not be easily resolved at the IP layer. However, it is possible for a router to know the value of d at the Autonomous System (AS) level [14]. Schemes working at the AS level have the potential to address Problems 1 to 5 while it may only provide incomplete path information rather than hop-by-hop path information.

Problems 3, 5, and 6 are more difficult to resolve. A good scheme shall neglect the marking information from compromised routers while a better solution is to contrive a mechanism so that the correctness of marking information embedded by the upstream routers can be verified. Nowadays, a DDoS attacker normally has hundreds of zombies in hand, and thus large-scale DDoS attacks impose a new challenge to PPM.

ASEM, only ingress edge routers of each AS are allowed (eligible) to perform marking and the path length in the scheme is at the AS level rather than hop-by-hop as in PPM.

3.4.3.2 ICMP Traceback.

Basic Scheme—An ICMP traceback method called iTrace was proposed by Bellovin *et al.* [49]. In this scheme, each router selects one packet per 20,000 packets and then generates an ICMP message. The ICMP message has the same destination IP address as the traced packet. The ICMP message also contains the IP header of the traced packet, and the IP addresses of the incoming interface and the outgoing interface of the current router. As long as the victim receives sufficient ICMP messages, it may recover the whole attack path. In ICMP traceback, the TTL field in the IP header of the ICMP message is set to 255 so that the TTL value may be used as a clue to correctly reconstruct an attack path.

Analysis of iTrace—The marking procedure of iTrace is very similar to PPM. Therefore, it shares the similar pros and cons of PPM. Unlike PPM, ICMP traceback belongs to outbound marking, which constitutes two differences. First, ICMP traceback requires additional bandwidth to convey the marking information. Second, more marking bits can be used, and thus Problems 1 and 2 (as of PPM) can be effectively solved.

Suppose that the total number of attack packets from one source is N and the probability of generating an ICMP message at each router is p. For the first router closest to the specified source, the total number of generated ICMP packets is Np. For the second router, the total number of packets it receives (attack packets+ICMP packets) is $N(1+p)$, and thus $Np(1+p)$ ICMP packets are created. For a path with d routers between the attack source and the victim, the number of ICMP messages generated at the i-th router ($1 \le i \le d$) is $Np(1+p)^{i-1}$. Similar to PPM, the closer a router is to the victim, the more

ICMP packets are generated. Unlike PPM, the number of ICMP messages the victim obtains from a router is the same as that generated by the router because there is no "re-marking" (see Table 3.1). This desirable property implies a further improvement. That is, iTrace requires far less number of marked packets (ICMP packets here) than PPM for path reconstruction.

Variants—Mankin *et al.* [64] proposed an "intention-driven" ICMP traceback technology. The idea is to add some intelligence to the marking procedure, so that the information required for path reconstruction may be quickly gleaned by the victim. To implement "intention-driven" ICMP tracing, each router needs to modify its routing table to accommodate the intention information. This enhancement further thwarts Problems 1 and 2. Recently, Wang *et al.* [65], [66] proposed to develop a new ICMP message called iCaddie.

Problem 3 may be addressed by secure infrastructure such as Public Key Infrastructure (PKI) [67]. Although PKI can tackle the issue of false marking, it imposes too high overhead on each router. Further work is required to address Problems 4, 5 and 6 using ICMP tracing.

Table 3.1 Comparisons of PPM and iTrace

Schemes	The router along an attack path	# of packets passing by	# of packets marked by this router	# of marked packets from the current router received by the victim
iTrace	1st	N	Np	Np
	2nd	$N(1+p)$	$Np(1+p)$	$Np(1+p)$
	3rd	$N(1+p)^2$	$Np(1+p)^2$	$Np(1+p)^2$
	...	...	...	...
	d-th	$N(1+p)^{d-1}$	$Np(1+p)^{d-1}$	$Np(1+p)^{d-1}$
PPM	1st	N	Np	$Np(1-p)^{d-1}$
	2nd	N	Np	$Np(1-p)^{d-2}$
	3rd	N	Np	$Np(1-p)^{d-3}$
	...	...	...	...
	d-th	N	Np	Np

3.4.3.3 HASH-based IP Traceback.

Basic Scheme—Hash-based IP traceback (also called SPIE) was proposed by Snoeren *et. al* [48]. This scheme is composed of 3 components: Data Generation Agents (DGAs), SPIE Collection and Reduction Agents (SCARs), and SPIE Traceback Manager (STM). The function of DGA is implemented in routers using bloom filters in such a way that each router deterministically logs some information of each packet traversing the router. Each SCAR is in charge of one area of the network, and it is connected to all DGAs inside this area. STM is the central management unit that is responsible for handling the requests of the victim and assembling the path information from associated SCARs.

Whenever a server or a network is under attack, the Intrusion Detection System (IDS) at the victim will identify the features of attack packets and report these features to STM. STM then sends inquiry request to proper SCARs. Each SCAR collects the logging information (also called digest) of each router (or DGA) inside its area and analyzes whether the attack packets have passed through the current area or not. If this is

true, the SCAR determines the routers which forward these attack packets, and further reconstructs the attack path inside this area. All related SCARs submit their partial path reconstruction results to the STM so that the latter can reconstruct each path after gleaning these results.

The digest collected at each router is derived from the following information: the constant fields in the IP header and the first 8 bytes in the payload of the current packet. The digest table stored in a router is implemented by using the bloom filter, a specific space-efficient data structure. Whenever a bloom filter is about 70% full, this filter is archived for later querying and a new filter will be used. With the help of transform lookup table (TLT), SPIE is capable of tracing transformed packets.

Analysis of Hash-based Traceback—SPIE is a deterministic logging scheme. It requires an additional infrastructure such as STM and SCARs, and it supports advanced functions such as single packet tracing and transformed packets tracing that are especially useful in wireless networks.

Two main drawbacks exist in SPIE. It incurs very heavy computational, management, and storage overhead. Though the neat property of bloom filters mitigates the extent of storage requirement, the deterministic nature still creates a big problem. More importantly, SPIE is not scalable. The current Internet is decentralized. Therefore, it is very difficult to extend this scheme from one network to the whole Internet because no STM of one network can exceed its administrative border in reality. These shortcomings seriously impede the applicability of SPIE.

Variants and Possible Solutions—In terms of the problems exhibited in PPM, Problem 1 is also an issue here. Unlike PPM, however, the computational burden is distributed in the network (SCARs and STM) rather than the victim only. Furthermore, since the logging information is distributed in each router, a high communication/bandwidth burden is incurred for SCARs and STM to recover paths. False positives depend on the performance of the selected bloom-filter. An ideal bloom-filter can greatly lessen false positives. Problems 3 and 4 are not an issue any longer because of deterministic logging. Also, Problem 5 may be effectively thwarted with the help of the central management unit. As stated earlier, Problem 6 is an important issue here.

Li *et. al* [53] proposed a novel logging scheme which may further mitigate the storage requirement by sampling. By correlating the samples, the proposal may successfully construct an attack tree. Their simulations show that the scheme may scale well to more than 5000 attack sources, a significant improvement over SPIE.

3.4.3.4 Deterministic Packet Marking.

Basic DPM—Deterministic Packet Marking (DPM) was proposed by Belenky and Ansari [51]. In this scheme, *only* ingress edge routers perform the marking as indicated by the DPM enabled routers shown in Figure 3.4. All other routers are exempt from the marking task.

The basic DPM uses the 16-bit ID field of the IP header and one reserved bit to record the marking information. The IP address of every ingress edge router is split into two segments with 16 bits each. One segment will be randomly selected when a packet traverses this router. The idea is that the victim is capable of recovering the whole IP

address of an ingress edge router once it obtains both segments from the same router. For the victim to figure out which portion of the IP address the current packet carries, one bit is used as a flag. Therefore, the marking information is comprised of two parts, 16-bit partial IP address of the edge router and 1-bit flag.

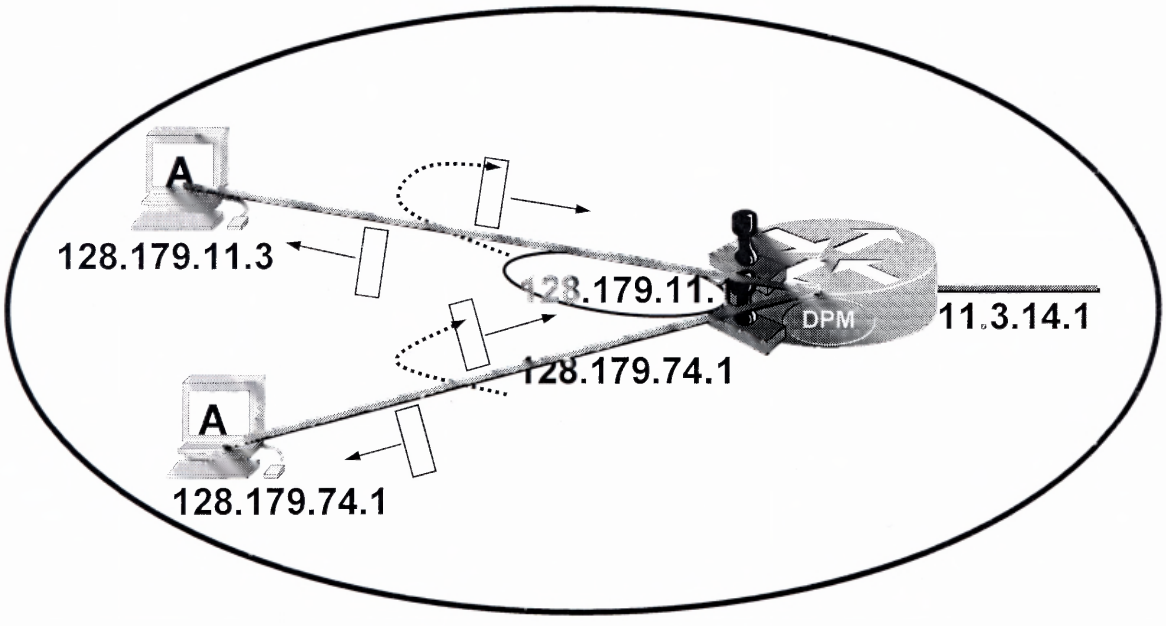


Figure 3.4 Deterministic Packet Marking (DPM).

The basic scheme can effectively handle a DoS attack. For a DDoS attack, the approach may introduce high false positives. Another shortcoming is that it cannot identify the ingress edge router if the attacker uses different source IP addresses for each packet. To address these issues, they further enhanced the basic DPM by using the "linkage" information [58], [68]. That is, a hash function is used to contain the identity of the ingress edge router so that all packets traverse the same router possess the same identity. The victim can use this identity to correctly combine the packets from the same

source so that the whole IP address may be recovered. Thus, the marking information is comprised of three parts, a segment of the IP address of the current router *s*, the index of the current segment *k*, and the fixed linkage information *d (digest)*. A good tradeoff is obtained when s=4, k=3, and d=10.

Analysis of DPM— Similar to PPM, DPM also uses the ID field to record the marking information. There are two main differences between DPM and PPM. First, PPM marks all routers along an attack path while DPM only marks the first ingress edge router (see Figure 3.5). Second, PPM marks probabilistically while DPM marks every packet at the ingress edge router.

These differences have the following implications. First, the task of ingress address reconstruction in DPM is much simpler in comparison with the task of path reconstruction in PPM. As a result, DPM may handle large-scale DDoS attacks better. Second, the false positives in DPM are far less than PPM. Third, DPM has the potential to tackle reflector-based DDoS attacks.

Possible Solutions—Problems 1 and 2 are effectively thwarted in DPM. For each attack path, only the IP address of each ingress edge router needs to be recovered. Thus, the computational burden is reduced significantly. Furthermore, the linkage information may be used as a guide to effectively prevent the "combinatorial explosion" problem in PPM. This further mitigates the computational overhead. A nice collateral effect is that the false positives are decreased as well.

Problems 3 and 4 are not an issue in DPM. However, Problem 5 needs to be further addressed. One possible solution is to record partial path information rather than the whole path information in PPM and one single point in DPM, e.g., path information at the AS level.

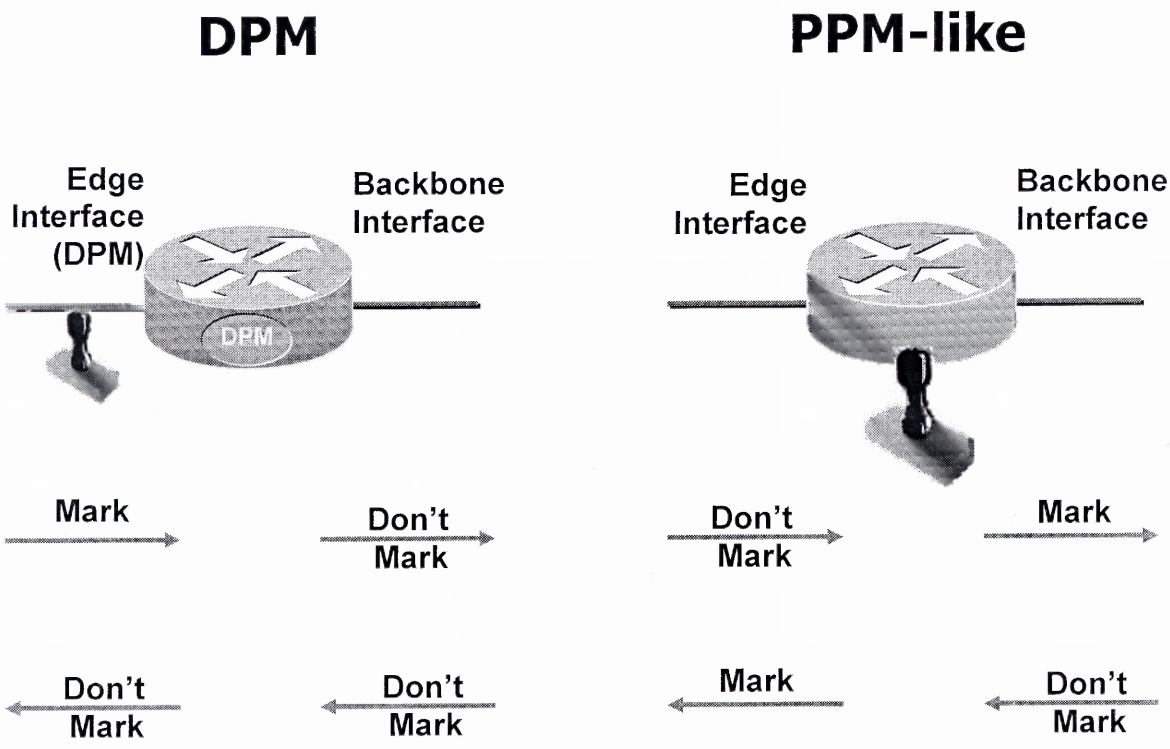


Figure 3.5 Marking in PPM and DPM.

3.4.3.5 Overlay Network.

Basic Scheme—Stone [52] presented CenterTrack, an overlay-based solution to IP traceback. In this approach, a specific router called Tracking Router (TR, or a group of TRs) is used. To trace one attack flow, dynamic routing is employed. All traffic to the victim is routed to the TR. The TR is logically directly connected to each ingress and egress edge router of the protected network through tunnels. Unlike other traceback

schemes that depend on the IDS of the victim to detect invasion, IDS in CenterTrack is implemented in the TR. When an intrusion is detected, TR is capable of locating the ingress edge router of the identified attack flow because the ingress edge router may be viewed as only one hop away from the TR.

Analysis of CenterTrack—Clearly, CenterTrack enforces a heavy management burden over the network. It also wears out tremendous system resources, such as bandwidth and processing capability due to establishment and maintenance of tunnels. Similar to SPIE, furthermore, scalability constitutes another major limitation to CenterTrack. Even though CenterTrack may determine the ingress edge router of one network with the help of TR, it cannot trace down the attack path once beyond the border of the current domain. Therefore, its applicability is rather limited.

Possible Solutions—Few updates to CenterTrack have been proposed at present. Recently, an associated defensive method, Secure Overlay Service (SOS) [30], was proposed. Unlike reactive tracing scheme, SOS is a proactive approach. By employing intensive filtering and anonymity into the forwarding structure (overlay network), SOS may effectively mitigate the impact of DDoS attacks.

In terms of the problems exhibited in PPM, Problem 1 is not a big issue. Since the ingress edge router is logically one hop away from the TR, path reconstruction in the specified network is straightforward because of the "simplified" topology. However, the usage of tunnels introduces some extra processing. Moreover, the computational burden is enforced on the TR and edge routers rather than the victim. False positives are well

thwarted in this scheme. Problems 3 and 4 do not need to be considered here. Another benefit of the "simplified" topology is that the chance of routers being compromised is rather low or at least much easier to be detected and diagnosed.

CHAPTER 4

AUTONOMOUS SYSTEM-BASED EDGE MARKING (ASEM)

4.1 Introduction

The proposed enhanced PPM improves the performance of PPM significantly. However, several issues still need to be addressed, including 1) to obtain the optimal marking probability, a router needs to know its distance to the victim, which is difficult to implement at the IP level; 2) the enhanced PPM does not handle spoofed marking well. For instance, it cannot handle subverted routers. When a compromised router embeds spurious marking information, the victim has no way to tell it from the correct marking information of normal routers. To cope with these issues, a novel marking scheme based on Autonomous Systems (ASs) is proposed.

Legacy IP traceback schemes use IP address information of each router to reconstruct the attack paths, hop-by-hop [47], [50], [56], [58], [59], [60]. Yaar *et al.* [44] first introduced the concept of path identification and they presented a new scheme, Pi. In their point of view, a path identifier does not have to be the IP address information. Using this idea, we here advocate a coarse-grained path identification at the AS level. Similar to the conventional Probabilistic Packet Marking (PPM) [47], routers along the attack paths mark packets according to a certain probability. The differences between ASEM and PPM are listed below. 1) Only the ingress edge routers of each AS conduct marking. 2) All routers are prohibited from re-marking packets already marked by any upstream router. 3) The marking information is the AS number (*ASN*) rather than the IP address of each traversed router.

The contributions are six-fold. First, ASEM greatly relieves the victim from the overwhelming computational burden. We define a metric—the number of marked packets required for path reconstruction—to evaluate disparate traceback schemes. Using this metric as the guideline, we explore two different approaches to mitigate the computational overhead. Second, these improvements not only reduce the number of packets needed for reconstruction, but also completely eradicate the threat of spoofed marking inscribed by the attacker. Third, ASEM can address spoofed marking incurred by subverted routers by allowing ingress edge routers in downstream ASs to examine the correctness of the marking information from their adjacent ingress edge routers in upstream ASs. Fourth, false positives are effectively suppressed. Fifth, ASEM outperforms PPM in that it can handle large-scale DDoS attacks. Finally, the power-law Internet renders ASEM effective even under partial deployment [32]. With the above merits, ASEM can be deployed in practice.

4.2 Background

Before proceeding to depict the whole picture of ASEM, some background is introduced.

Internet hierarchy is well known but rarely used in IP traceback. The Autonomous System is an important component of the Internet hierarchy. Normally, an AS is regulated by one entity, which can enforce a consistent routing policy inside the whole administrative domain. Among different ASs, the administrative policy may be distinct dramatically.

BGP is the de facto standard for inter-AS routing while the intra-AS routing frequently uses OSPF, IS-IS, RIP, and IGRP [69], [70], [71], [72], [73]. Multiple ASs

depend on BGP to exchange the route reachable information, and the task is conducted by a few routers called *BGP Speakers*. There are three nice characteristics of AS. The first characteristic is that an AS path is much shorter than the corresponding IP path [74], [75], [76], [77]. For instance, as shown in Figure 4.1, the attack path from A1 takes 8 hops, and the one from A2 takes 7 hops to V while the AS paths are 3 "hop"s away. The second nice property of ASs is that routing at the AS level is much more stable than at the IP level [78], [79]. Finally, one important attribute in the BGP routing message is called ASPATH, which provides the ordered list of the ASs needed to traverse before reaching a given destination. As shown in Figure 4.2, suppose that the BGP speaker inside AS 12654 receives two routing information for prefix 135.207.0.0/16, one is from AS 1129 with the ASPATH attribute "1129 1755 1239 7018 6341" and another is from AS 3549 with ASPATH attribute "3549 7018 6341" [72]. Since the latter is shorter, the BGP speaker in AS 12654 may keep it in its routing table. This implies that 1) the address prefix 135.207.0.0/16 is located inside AS 6341; 2) for packets with destination address in the range of (135.207.0.0, 135.207.255.255), they will traverse to AS 7018 via AS 3549, and to AS 6341 via AS 7018 (We here assume that there is not any other prefix inside this range. That is, no prefix such as 135.207.1.0/24 exists in the same BGP routing table).

Note that the above three features can be exploited by an IP traceback scheme. The first means less "hop" counts from the source to the destination, inferring less number of marked packets required for path reconstruction. To recover an attack path, the victim *only* needs to receive several marked packets in ASEM, which significantly outperforms other PPM schemes [40], [47], [56], [57], [58], [59], [60], [80]. The second

simplifies path reconstruction because fewer possible paths are needed to be

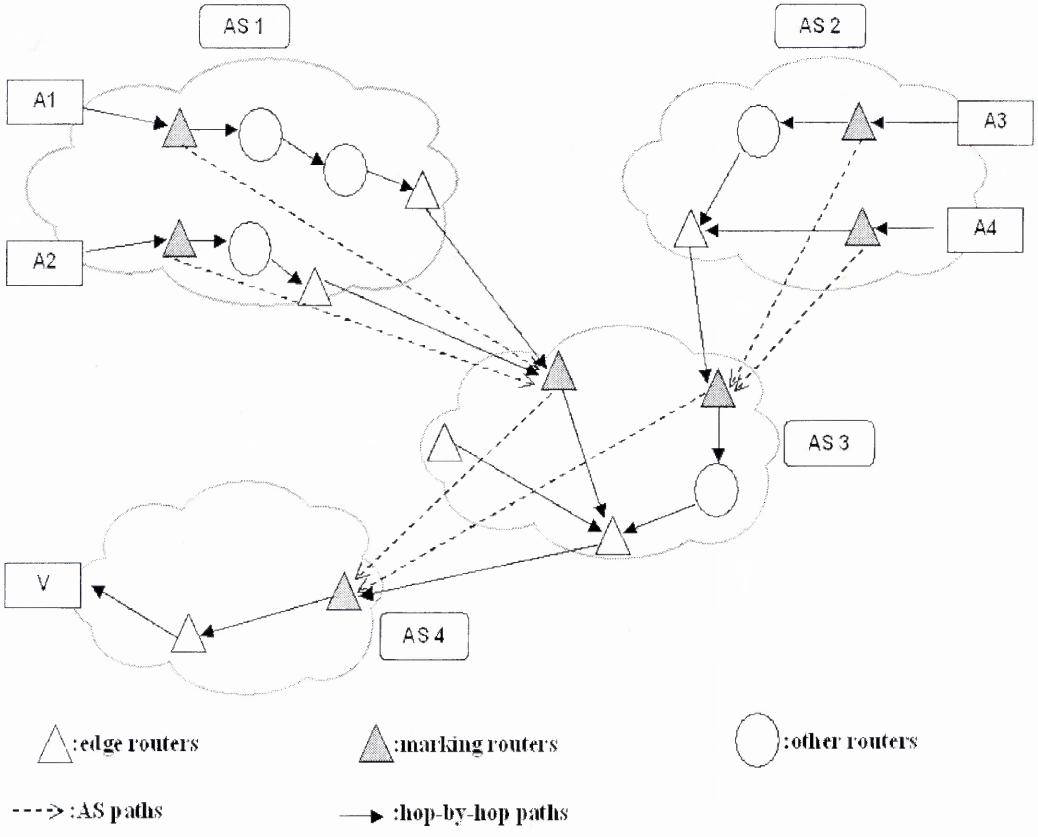


Figure 4.1 AS path vs. hop-by-hop IP path.

considered, and thus the victim is relieved from the problem of combinatorial explosion. The third can be used for marking verification if the ASPATH attribute is used for marking. Suppose a flow of packets are bombarding at a host 135.207.x.y, the marking at AS 12654 is then "3549 7018 6341", and the marking at AS 3549 is "7018 6341". It is easy for AS 3549 to determine whether the marking from its upstream neighbor AS 12654 is correct or not because the only difference of these two markings is the ASN of

the current AS. Since we only use 16 bits to record the ASPATH attribute, some transformation is required. Further details are provided in subsection 4.5.2.

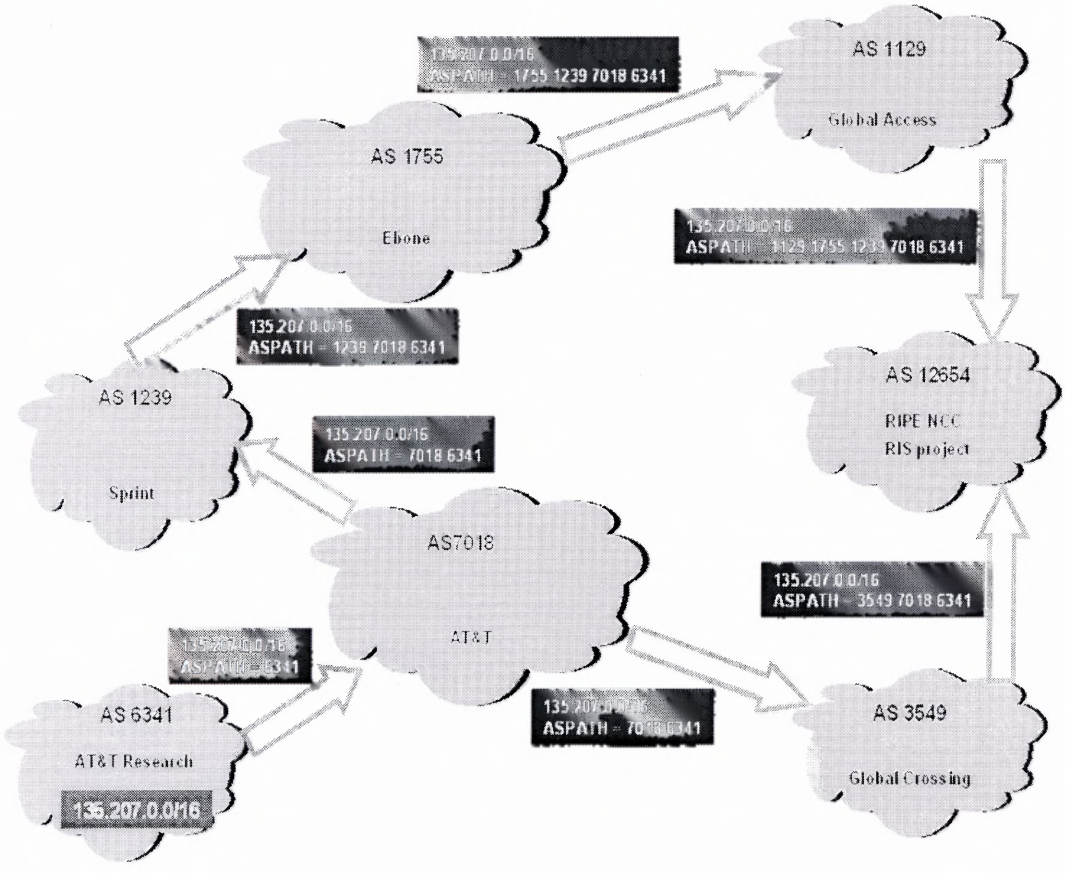


Figure 4.2 Prefix originated ASPATH attribute [72].

4.3 Assumptions

In order to outline the framework of the design, the following assumptions are made:

- The attacker may create any packet.
- The attacker may know the tracing scheme.
- The attack is at least composed of tens of packets.

- Only a few routers, if any, may be subverted. Compromised routers are not adjacent.
- Every ingress edge router of an AS shares the BGP routing information of its domain.
- The AS path is rather stable.
- The length of any AS path is limited.

The first two assumptions represent the fact that the attacker may have the root privilege over the zombies, and may generate any packet he/she wants, including spoofed marking intentionally. The third one indicates that ASEM is contrived for flood-based attacks, the dominant DoS/DDoS attack pattern [9], [10], [11], [81]. Different from previous works, we address the challenge of spoofed marking from both the attacker and compromised routers. We further assume that compromised routers are not adjacent. Considering the technical hurdle to subvert a router, the assumption is acceptable. The fifth one is critical to the design. We assume that all ingress edge routers in each AS share the BGP routing table of the BGP speaker in the same domain. This assumption requires some additional memory on each ingress edge router to store the BGP routing table. However, this requirement is not a big issue because the total number of ASs is only about 20,000 nowadays [82]. In ASEM, when an ingress edge router receives a packet, it uses the BGP routing table to conduct marking and marking examination. The last two assumptions are supported by the Internet measurement [75], [76]. The dominant AS path lengths are 3 to 5, with an average value of 4. The proposal assumes

that an AS path length is not greater than 8, which is satisfied by about 99.5% of all AS paths [75], [76].

4.4 Reducing The Computational Burden

The computational burden lies mainly on the procedure of path reconstruction. Reducing the total number of marked packets required for path reconstruction is therefore critical. Similar to enhanced PPM, we first attempt to find the optimal marking probability, then to enhance the marking mechanism, and finally to study the possibility of "reducing" the path length.

Denote k as the number of attack paths to the victim v. For path j ($1 \le j \le k$), the number of routers between the attack source and v is d_j. Let $p_j^i(m)$ be the marking probability of router i ($1 \le i \le d_j$) along path j, and $p_j^i(v)$ be the marking probability of router i along path j perceived by v. $p_j^i(v)$ may be different from $p_j^i(m)$, e.g., for PPM $p_j^i(m) = p$ and $p_j^i(v) = p(1-p)^{d_j - i}$ [57], [61]. Denote N_j as the number of packets traversing along path j, and M_j^i as the number of packets marked by the i-th router along path j and received by v. In other words, those packets initially marked by the i-th router but are re-marked by any subsequent router are not counted into M_j^i. Denote M_j as the number of packets marked by any router along path j and received by v. Clearly, the expectations of M_j^i and M_j are

$$E[M_j^i] = N_j p_j^i(v), \tag{4.1}$$

and

$$E[M_j] = E\left[\sum_{i=1}^{d_j} M_j^i\right] = \sum_{i=1}^{d_j} E[M_j^i] = N_j \sum_{i=1}^{d_j} p_j^i(v), \tag{4.2}$$

respectively.

Since PPM and ASEM mark packets probabilistically, M_j^i and M_j are random variables. Thus, it is difficult to directly compare the number of marked packets under PPM and ASEM. However, we can compare their performance given the same number of attack packets and the same attack path. The metric that we use is the expectation of the total number of marked packets, $E[M_j]$.

4.4.1 The Number of Marked Packets for Path Reconstruction

In PPM, $p_j^i(v) = p(1-p)^{d_j - i}$. From (4.2) we obtain

$$E[M_j] = N_j \sum_{i=1}^{d_j} p_j^i(v) = N_j \left(1 - (1-p)^{d_j}\right). \tag{4.3}$$

The design of ASEM ensures that all packets are marked somewhere along a path so that all spoofed markings from the attacker are overwritten. Therefore, spoofed marking from the attacker is not an issue for ASEM. Since

$$\sum_{i=1}^{d_j} p_j^i(v) = 1, \tag{4.4}$$

for ASEM,

$$E[M_j] = N_j \sum_{i=1}^{d_j} p_j^i(v) = N_j . \tag{4.5}$$

That is, given the same number of attack packets and the same path, on average, the victim can obtain more marked packets in ASEM than in PPM. Subsequently, the victim can more likely reconstruct the attack path in ASEM than in PPM.

4.4.2 Estimating the Number of Attack Packets Required for Path Reconstruction

In the last subsection, we study the number of marked packets and the probability for the victim to receive at least one marked packet from each router in ASEM and PPM, given

the number of attack packets. Here, we further study the number of attack packets required for successful path reconstruction.

We assume that the path reconstruction can be completed as long as the victim receives at least one marked packet from each router. In this subsection, to simplify the analysis, when we discuss the number of marked packets, we refer to their expected values. Similar simplification can be found in most previous traceback schemes, such as [47], [54], [57], [61].

Given

$$M_j^i = N_j p_j^i(v) \geq 1, \forall i (1 \leq i \leq d_j), \tag{4.6}$$

in PPM, since $p_j^i(v)$ is a monotonically increasing function of i (i.e., $p_j^1(v) < p_j^2(v) < \ldots < p_i^{d_j-1}(v)$), (4.6) can be simplified to

$$N_j \geq \frac{1}{p_j^1(v)}. \tag{4.7}$$

That is,

$$N_j \geq \frac{1}{p(1-p)^{d_j-1}}. \tag{4.8}$$

For PPM, the minimum value of N_j can be obtained by taking the derivative of (4.8) with respect to p, thus resulting in $p = \frac{1}{d_j}$.

In this case, N_j for PPM can be as low as

$$N_j \geq \frac{(d_j)^{d_j}}{(d_j-1)^{d_j-1}}. \tag{4.9}$$

Unlike PPM, the marking probability with respect to the victim is the same at each router in ASEM. That is,

$$p_j^i(v) = \frac{1}{d_j}. \tag{4.10}$$

Combining (4.4) with Inequality (4.10), it is easy to see that N_j can reach its minimum as long as (4.10) holds. In this case,

$$N_j \geq d_j. \tag{4.11}$$

In fact, (4.10) always holds in ASEM, and therefore, ASEM always uses the optimal marking probability.

Since Inequality

$$\frac{(d_j)^{d_j}}{(d_j - 1)^{d_j - 1}} > d_j \tag{4.12}$$

always holds, theoretically, the minimum number of attack packets required for path reconstruction in ASEM is less than that in PPM even both use the optimal marking probability.

4.4.3 Further Discussion on the Optimal Marking Probability

The last two subsections study the path reconstruction from the perspective of the victim v. Now, we consider the issue from the perspective of each router along the attack path. Two questions arise naturally. 1) What would the marking probability ($p_j^i(m)$) at each router be in order to obtain the optimal $p_j^i(v)$? 2) Can the derived optimal marking probability be practically implemented at each router?

For PPM, the marking probability ($p_j^i(m)$) at each router is the same: $p_j^i(m) = p$, $\forall i(1 \le i \le d_j)$. Furthermore, if each router can know in some way the path length (d_j) ahead of time, the router can set the marking probability to the optimal value. If this is the case, the number of packets required for path reconstruction can be reduced to the value shown in (4.9). However, since PPM works at the IP level, no feasible method exists in the current Internet to provide the path length for each router in advance. Therefore, the derived optimal marking probability is infeasible for PPM from the practical perspective.

For ASEM, the marking probability ($p_j^i(m)$) at each router is not the same. Each router determines its marking probability according to its distance to the victim. For path j, the i-th router sets its marking probability to be $p_j^i(m) = \frac{1}{(d_j - i + 1)}$, where $(d_j - i + 1)$ is the distance (path length) between the current router and v. This is feasible because the ASPATH attribute provides the exact length information (more details can be found in Section 6.2). For the first router, the marking probability is $1/d_j$; for the second router, the marking probability is $1/(d_j - 1)$; etc. However, since the policy of NO "re-marking" is imposed in ASEM, what the first router has marked cannot be re-marked by subsequent routers. Therefore, only $(1 - \frac{1}{d_j})N_j$ packets (average number) are available for the second router to mark. With respect to the victim,

$$p_j^2(v) = \frac{1}{(d_j - 2) + 1} \times (1 - \frac{1}{d_j}) = \frac{1}{d_j}. \tag{4.13}$$

Similarly,

$$p_j^i(v) = \frac{1}{(d_j - i) + 1} \times (1 - \sum_{s=1}^{i-1} p_j^s(v)) = \frac{1}{(d_j - i) + 1} \times (1 - \frac{i-1}{d_j}) = \frac{1}{d_j}. \tag{4.14}$$

That is, each router in ASEM always marks packets using the optimal marking probability. Thus, the computational burden is minimized.

In summary, with respect to the computational burden, ASEM distinguishes from PPM in two aspects. First, the derived optimal marking probability is feasible and practically used in ASEM while it is impractical for PPM to use the optimal marking probability because of its unawareness of the whole path length. Second, even assume that all routers in PPM always use the optimal marking probability, Inequality (4.12) shows that ASEM still requires less number of packets for path reconstruction.

4.4.4 Decreasing Path Length

Considering (4.11), N_j in ASEM may be further reduced by decreasing the value of d_j. Suppose that only $d_j^{'}$ of d_j ($d_j^{'} < d_j$) routers are used to recover the attack path. The smaller $d_j^{'}$, the smaller N_j.

$$N_j \geq d_j^{'},\ d_j^{'} < d_j . \tag{4.15}$$

We use the AS path, which is much shorter, instead of the hop-by-hop IP path. Since only marking routers along a path conduct marking, this is equivalent to a shorter path length with respect to path reconstruction. Note that the most important information for IP traceback is the information of the first router along a path. Though ASEM is based on the AS level, it also records the information of the first router along a path, and therefore ASEM can trace attack sources efficiently.

4.5 Robust Marking

A good marking scheme shall balance between efficiency and robustness. Section 4.4 investigates the issue of optimal marking. Here, we address the issue of bogus marking from the attacker and/or subverted routers.

4.5.1 Spoofed Marking Embedded by the Attacker

The attacker may effectively deter tracing by inscribing forged marking [61], [62]. In traditional PPM [47], with respect to v, the probability that packets marked by the farthest router is $p(1-p)^{d_j-1}$ along path j. Let q_j be the probability that a packet has never been marked by any router along path j,

$$q_j = (1-p)^{d_j}. \tag{4.16}$$

Clearly, if $p<0.5$, $q_j > p_j^1(v) = p(1-p)^{d_j-1}$. That is, the attacker may confuse v by filling bogus information on the unmarked packets so that v cannot locate the farthest router of each path. Even worse, the negative impact of spoofed marking is not limited to the farthest routers, i.e., the routers closest to the attack sources. For the average path length of 15, the optimal marking probability is p=0.0667. Thus, q_j=0.3553. Note that even for the closest router to v, $p_j^{15}(v) = 0.0667 < q_j$, letting alone any other farther routers (recall that $p_j^i(v)$ is a monotonically increasing function of i in PPM). This example shows how easy it is to disguise the victim v if the attacker embeds bogus marking information in PPM. However, with the NO "re-marking" strategy and the derived optimal marking probability $p=1/((d_j-i)+1)$, this is not an issue any longer because q_j becomes 0.

4.5.2 Spoofed Marking Caused by Subverted Routers

Another source of bogus marking is the subverted routers. Up to now, few works explored this problem. References [54] and [80] proposed to use authentication to ensure secure marking; here we attempt to tackle this problem by a simpler method.

The feature of BGP routing allows a downstream marking router R_b of AS_b to examine the correctness of the marking embedded by its adjacent upstream marking router R_a of AS_a because the ASPATH attribute of R_a shall be the concatenation of the ASN of R_b and the ASPATH attribute of R_b [70], [72]. Note that here AS_b is a neighbor of AS_a. If a mismatch is found, the downstream marking routers can filter or drop those packets with spoofed marking. For example, assume that a path from the source *src* to the destination *dst* traverses AS_a, AS_b, AS_c, AS_d, AS_e at the AS level. The ASPATH attributes for each AS mentioned above to *dst* are "$AS_b\ AS_c\ AS_d\ AS_e$", "$AS_c\ AS_d\ AS_e$", "$AS_d\ AS_e$", "AS_e", "•", respectively. We use "•" to denote the last AS because the destination *dst* is inside AS_e and then only IGP routing protocol rather than EGP routing protocol (such as BGP) is used. Note that $ASPATH(AS_a)$= ***Concatenate***$(AS_b, ASPATH(AS_b))$. Subsequently, if the ASPATH attribute is used as the marking information at each AS, the marking router at AS_b can then check the correctness of the marking information from the marking router of its upstream neighbor AS_a. Since only 16 bits are used to store the ASPATH attribute in the scheme, we use XOR operation to the ASN of the current AS and all of the ASN in the ASPATH attribute and record the final result in AS_PATH. At AS_a, the marking information for *dst* is $AS_a \oplus AS_b \oplus AS_c \oplus AS_d \oplus AS_e$, where $\oplus$ is the exclusive operator; at AS_b, the marking information for *dst* is $AS_b \oplus AS_c \oplus AS_d \oplus AS_e$. We

then have AS_PATH(AS_a)=AS_a⊕AS_PATH(AS_b). This relationship holds for all neighbors.

4.6 Effectiveness to Large-scale DDoS Attacks

PPM is ineffective to large-scale DDoS attacks [54], [61]. This is originated from the insufficient marking bits in the IP header. As mentioned earlier, two steps are required for path reconstruction in PPM. One is the recovery of the complete IP address of each router, and another is the recovery of each full path. The performance of the first step may be seriously degraded because many routers may have the same distances to the victim and there exists no hint for packets from the same router to combine into a complete IP address. Similarly, no clue for packets from the same sources is presented for the victim to reconstruct a path effectively.

Goodrich [58] presented the idea of using "linkage" information to identify packets from the same router. We employ this idea in ASEM. Note that *only* one step is required for path reconstruction in ASEM, and that *only* packets with the same linkage may be combined into a full path.

We propose to use the next 16 bits of the ID field (3-bit Fragment Flag field+13-bit Fragment Offset field) in the IP header to store the linkage information. The "No re-marking" flag occupies the 1st bit of the Fragment Flag field, which is the reserved bit with the default value of 0. The next 3 bits is used to record the length of the AS path. We propose to use a hash function to map the 32-bit IP address of the first router to a 12-bit hash value, called HASHIP. Using this field as the guide, ASEM is very effective in

determining the packets from the same sources. In so doing, ASEM may tackle large-scale DDoS attacks that are dominant today.

The following are the merits of using the HASHIP field:

1) Using HASHIP as the guide, the path reconstruction procedure is significantly simplified because blind combinations of nodes to form a path is effectively avoided.

2) The HASHIP field alone may be used as the identifier for the victim to block attack traffic, which is infeasible for PPM (and most other schemes) because the marking information of a router in PPM is segmented and transmitted in several packets.

3) With the help of HASHIP and AS_PATH, ASEM may be used to tackle large-scale DDoS attacks. AS_PATH may be used to differentiate attack flows traversing different ASs; HASHIP is used to distinguish attack flows launched from different sources at the same AS, thus facilitating ASEM to address large-scale DDoS attacks.

4) After determining the AS path that the attack packets have traversed, the system administrator of the first AS along the attack path can identify the ingress edge router from which attack packets are emitted as long as the number of the ingress edge routers in the AS is less than 4096 (2^12, we here suppose that an ideal hash function is used). For PPM, even if the victim can reconstruct the IP address of the ingress edge router along a path, it still requires the system administrator of the corresponding AS to take action because the victim is not entitled to manage that router. Therefore, telling the corresponding system administrator the full IP address of the ingress edge router or HASHIP is equivalent because the system administrator can keep a lookup table to determine the IP address from the HASHIP value.

4.7 Marking Algorithms

The proposed marking and path reconstruction algorithm is very similar to that of PPM. One difference is that the linkage information in ASEM avoids blind combination in the recovery of each attack path, thus rendering fast and efficient path reconstruction. Here, we present the marking algorithm only because our marking algorithm performs an additional job, marking verification.

The marking algorithms are further divided into the one for the first marking router (shown in Figure 4.3), and another for other marking routers (shown in Figure 4.4). If a marking router receives a packet from the same AS, it is the first marking router. On the contrary, if a marking router gets packets from other AS, it is not the first marking router. For the first marking router, it is important to check the value of the FLAG field because a sophisticated attacker may pre-set this field to 1 to block any further marking. For all other marking router, they need to check the AS_PATH field to address forged marking.

Marking procedure at the first ingress edge router R

```
For each packet w
   If w.FALG='1'                          //the attacker may spoof the flag intentionally
      w.FALG='0'
   write hash(R) into w.HASHIP
   Let dst be the destination IP address of w
   Lookup the BGP routing table of R to get the ASPATH attribute, ASPATH_R(dst)
   Delete repeated ASN at ASPATH_R(dst)
   p1=1/(len(ASPATH_R(dst))+1)            //the optimal marking prob. of R
   Let x be a random number from [0,1)
   If x<p1                                //mark the packet
      Write ASN(R) into w.AS_PATH         //initiate w.AS_PATH with the current ASN
      For each item u in ASPATH_R(dst)
         Write XOR(w.AS_PATH,u) into w.AS_PATH
      Write len(ASPATH_R(dst)) into w.LEN
      Write '1' into w.FLAG
   Forward w
```

Figure 4.3 Marking algorithm at the first edge router.

Marking and marking verification procedure at other ingress edge router S

```
For each packet w from neighbor AS T
   Let dst be the destination IP address of w
   Lookup the BGP routing table of S to get the ASPATH attribute, ASPATH_S(dst)
   Deleted repeated ASN at ASPATH_S(dst)
   current_mark=ASN(S)
   For each item u in ASPATH_S(dst)
        current_mark=XOR(current_mark, u)
   len2=len(ASPATH_S(dst))
   p2=1/(len2+1)                          //the optimal marking prob. of S
   If w.FALG='1'                          //w has been marked
       If w.LEN=len2+1 and w.AS_PATH!=XOR(ASN(T), current_mark)
                                          //spoofed marking from neighbor T
                Drop w
   Else
       Let x be a random number from [0,1)
       If x<p2                            //mark the packet
           Write current_mark into w.AS_PATH
           Write len(ASPATH_S(dst)) into w.LEN
           Write '1' into w.FLAG
       Forward w
```

Figure 4.4 Marking and verification algorithms at other routers.

4.8 Performance Analysis

4.8.1 Computational Burden

We compare the computational burden of ASEM with that of PPM from two aspects, with and without considering practical path length distribution.

4.8.1.1 Performance Comparison under Different Path Lengths without Considering Real Path Length Distribution. In PPM, routers are not cognizant of each path length ahead of time. To simplify the analysis, we assume that PPM will use the recommended marking probability, 0.04 [47]. We first present the effectiveness of each single improvement that we propose, and then show the synergic effect. Note that N_j shown in Figures 4.5-4.7 and Tables 4.1-4.2 is rounded up to the nearest larger integer, i.e., $\lceil N_j \rceil$.

A. Optimal Marking Probability.

The first improvement is achieved by using the proper marking probability (shown in (4.10)).

The value of N_j with PPM can be obtained by substituting $p=0.04$ into (4.8). For the improvement 1 (see subsection 4.4.1), the value of N_j is computed by using (4.11). The result is shown in Figure 4.5.

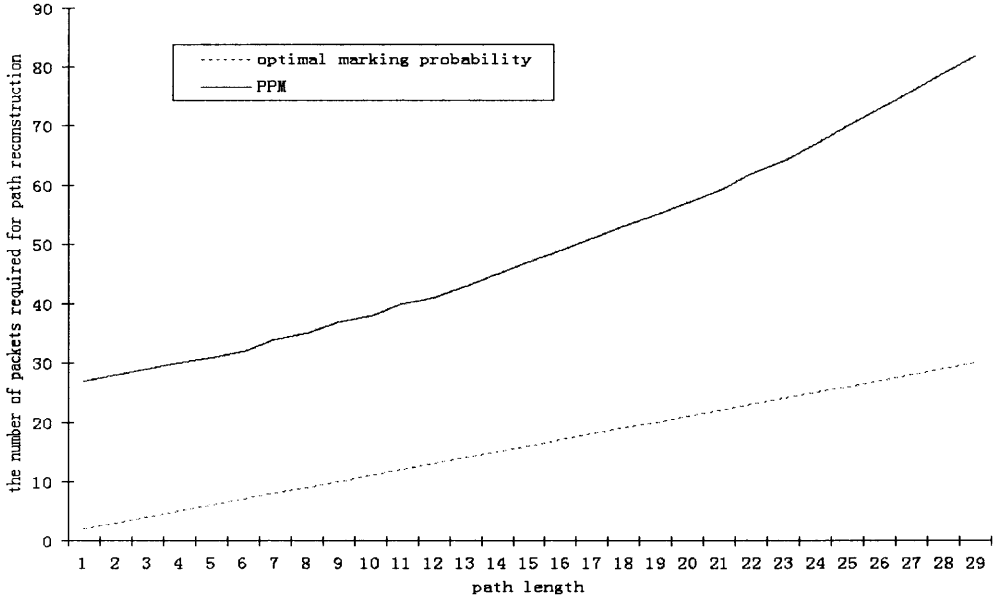


Figure 4.5 N_j for PPM *vs.* the improvement 1.

B. Shorter Path Length

Figure 4.6 demonstrates the advantage of the second improvement (see subsection 4.4.4) over PPM. Note that ASEM and PPM work at different granularity. Even for the same path, the value of path length is different for PPM and the approach because ASEM works at the AS level and only marking routers along each path are allowed to perform marking. Thus, ASEM has a "shorter" path length. According to the recent Internet

measurement [75], [76], on average the path length at the IP level is about 3 times the corresponding path length at the AS level. Hence, for simplicity, we only consider those IP paths with path lengths of 6, 9, 12, …, 30, corresponding to path lengths of 2,3,4,…,10 at the AS level. The simplification will be used whenever a comparison involves the improvement 2.

C. Putting Everything Together

Integrating both improvements into one scheme, the final result is shown in Figure 4.7. Obviously, ASEM outperforms PPM significantly.

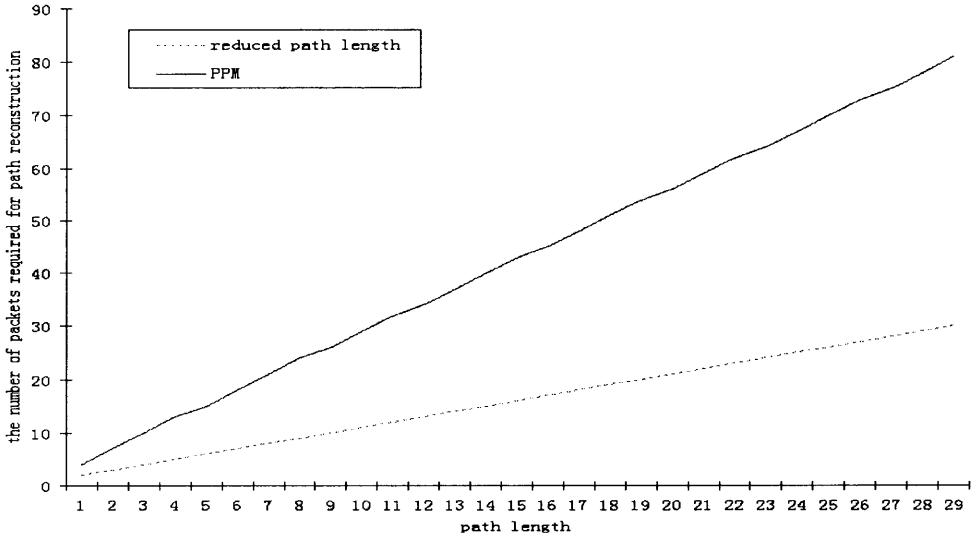


Figure 4.6 N_j for PPM *vs.* the improvement 2.

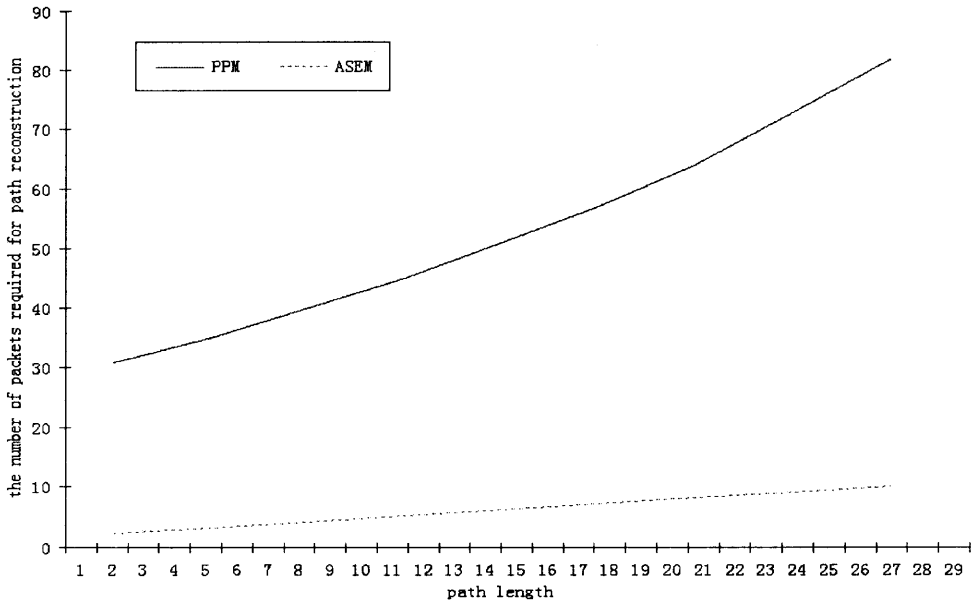


Figure 4.7 N_j for PPM *vs.* the scheme (integrating 2 improvements).

4.8.1.2 Performance Comparison Considering Real Path Length Distribution. In this subsection, the practical path length distribution is taken into account. In so doing, a more accurate picture of the performance of ASEM can be provided.

Two datasets are used. One is from the Skitter project of CAIDA [83], and another is the Internet Mapping data from Lumeta [84]. We simply average the number of paths from both datasets for each path length, and use the result as the dataset. Since a vast majority of IP path lengths fall in the range of (6,30) inclusively, we discard all paths whose lengths are out of this range. We choose a total of 9804 paths from the rest of the dataset. Among the 9804 paths, 3448 paths, which have IP path length of 6, 9, 12, ..., or 30, will be used for comparisons involving the improvement 2.

To reconstruct all 9804 paths (denoted as set S_1), we consider two related parameters: the total number of packets required to reconstruct all paths, N; and the average number of packets required to reconstruct a path, n. Similarly, for the selected

3448 paths (denoted as set S_2), N' and n' are used to represent the total number of packets required to reconstruct all paths and a path on average, respectively.

N, N', n, and n' are computed according to (4.17), (4.18), (4.19), and (4.20), respectively. The results are shown in Table 4.1 and Table 4.2.

$$N = \sum_{j \in S_1} N_j \,. \tag{4.17}$$

$$N' = \sum_{j \in S_2} N_j \,. \tag{4.18}$$

$$n = \frac{N}{9804}. \tag{4.19}$$

$$n' = \frac{N'}{3448}. \tag{4.20}$$

In Table 4.1, as explained before, we use only those IP paths whose lengths are multiples of 3 and in the range of (6,30) inclusive. Note that the approximation does not seem to affect the result much. Considering PPM, on average, the numbers of marked packets required for reconstructing a path from 9804 paths and 3448 paths are 68 and 65, respectively. These two values are very close (the difference is only 4.41%). With ASEM, a saving of **90.67%** on average of the total number of packets required for reconstructing a path may be achieved.

4.8.2 Robustness

ASEM can address spoofed marking from the attacker and subverted routers.

For PPM, the possibility that a packet reaches the victim untouched (i.e., unmarked) is $(1-p)^{d_j}$ along path j. To totally confuse the victim, the following inequality shall be satisfied,

$$q_j = (1-p)^{d_j} \geq \sum_1^{d_j} p_j^i(v). \tag{4.21}$$

In this case,

$$p \leq 1 - 2^{\left(-1/d_j\right)}. \tag{4.22}$$

Table 4.1 N and n under PPM and the Improvements 1

	PPM	Improvement 1
Total (N)	672,996	156,687
Average (n)	68	16

Table 4.2. N' and n' under PPM, the Improvement 2, and Both Improvements

	PPM	Improvement 2	Improvements 1-2
Total (N')	223,667	30,986	20,511
Average (n')	65	9	6

For the average path length of 15 [83], [84], (4.22) holds if $p \leq 0.04516$. Therefore, using the recommended value p=0.04 [47] will seriously impede reconstruction and invoke high false positives. In ASEM, on the contrary, q_j=0. In other words, even if all packets mounted by the attacker are inscribed with spurious marking, such bogus marking information will be totally overridden by correct marking information from routers as packets traverse along the attack path. Therefore, with this improvement, we eradicate spoofed marking from the attacker while optimizing N_j.

For subverted routers, ASEM thwarts their adverse impacts by examining the correctness of the marking information. In comparison with proposals using authentication [54], [80], ASEM introduces far less overhead.

4.8.3 False Positives

4.8.3.1 Less Marking Bits. One reason for high false positives is the insufficient marking bits. In PPM, the victim has to combine packets with 8 fragments to determine a 32-bit IP address while this step is not necessary in ASEM. Furthermore, the marking information for one router in ASEM is 16-bit, only half of that required in PPM. Therefore, false positives incurred by combinatorial explosion are mitigated significantly by both factors.

4.8.3.2 Linkage Information. The linkage information in ASEM can effectively avoid blind combinations in path reconstruction. This is very important especially in large-scale DDoS attacks, the dominant attack pattern today. The 12-bit linkage information can be used as a guide in path reconstruction.

4.8.3.3 Reduced Path Lengths. Note also the "avalanche" effect of false positives caused by routers closer to the victim. During path reconstruction, if a router R that is h hops away from the victim is added to the attack path by mistake, then this will affect locating routers $h+1$ hops away. The smaller h, the higher false positives. In general, the decrement in path length can reduce false positives exponentially, thus favoring the proposed scheme.

4.9 Conclusions

In this chapter, a robust and optimal marking scheme for IP traceback has been proposed. First, a metric for the optimization of path reconstruction is provided. Note that path reconstruction is the fundamental goal of packet marking. Using this metric as the guideline, two improvements have been presented. By integrating the two improvements,

ASEM possesses the following benefits: 1) Optimal marking probability. We have derived the optimal marking probability, and presented a *practical* implementation. In comparison with legacy PPM, as many as 90.67% of marked packets can be reduced on average. 2) Robust marking. ASEM can handle not only spoofed marking by the attacker, but also the phony marking incurred by subverted routers. 3) Effectiveness to handle large-scale DDoS attacks which are dominant in today's Internet. 4) Reduced false positives. High false positives are effectively suppressed due to the above improvements. 5) Partial Deployment. The power-law Internet facilitates effective partial deployment of ASEM.

CHAPTER 5

FOUR COLOR THEOREM-BASED PATH MARKING SCHEME

In the last chapter, an IP traceback scheme has been discussed. From this chapter on, various DDoS defense mechanisms will be presented and studied.

5.1 Introduction

Though many DDoS defense mechanisms have been proposed, different from IP traceback no systematic study has been conducted on DDoS defense. As mentioned earlier, previous works include D-WARD [29], PacketScore [38], Hop-count filtering [39], IP traceback-based filtering [40], Pushback [41], Puzzle Auctions [43], and Honeypot [45].

Similar to IP traceback, defending against DDoS attacks is a very difficult task because the source IP address can be easily forged. Recently, Yaar *et al.* [44] proposed an ingenious scheme, Pi (Path identification). Their idea is to distinguish one path from another with path identification rather than IP addresses. In comparison with other schemes, Pi [44] is promising. Two salient features make it attractive. Since it is deterministic marking rather than probabilistic marking, and all the marking information is imbedded in one packet, it is possible to defend against not only flood-based DDoS attacks, but also attacks induced by a few packets. Another benefit is that the victim can defend against DDoS attacks, without depending on the cooperation of the upstream system administrators. Using the complete binary tree model, the 16-bit ID field can be used to record 16 links, 1 bit for each link. However, a potential assumption of the binary

tree model is that each router has only two interfaces. In reality, many routers have more than two interfaces. Recent Internet measurement shows that the number of interfaces of 99% of the routers in the Internet is no greater than eight [77]. Therefore, it is reasonable to consider at most eight interfaces for each router. A naive extension to [44] can use 3 bits rather than 1 bit to distinguish one interface from another, and then the 16-bit ID field can record only five links. To reduce the possibility of false positives, we prefer to record as many links as possible in the 16 bits. Specifically, we want to use only 2 bits to distinguish interfaces (up to eight) of a router. This is the motivation to investigate the applicability of the Four Color Theorem [85] to tackle this problem.

The Four Color Theorem states that to color a map so that any adjacent region has a different color, at most four colors are required [85], [86] (see Figure 5.1). Though there may exist some areas that have many neighbors, the same color can be *reused* as long as their neighbors are not adjacent. Intuitively, this is very similar to the case here. That is, we have many routers, each with a different number of interfaces. Interfaces can use the same color as long as they are distinguishable. Based on the Internet hierarchy, we contrive a new color marking method.

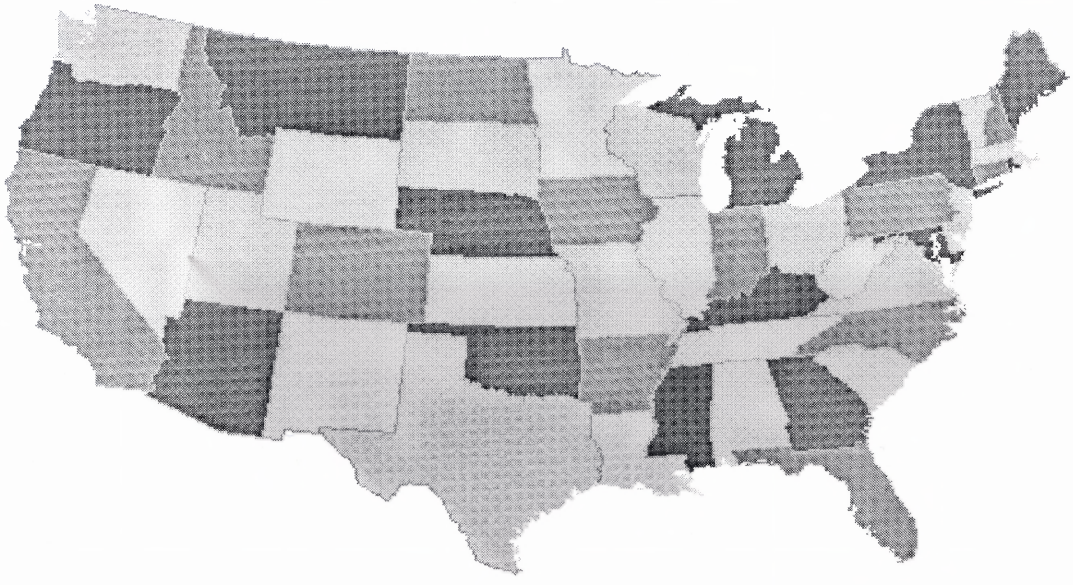


Figure 5.1 One example of color marking of the US mainland [86].

5.2 Background

Network hierarchy is well known but has rarely been applied in network security. Gao [73] first proposed a new scheme to infer Autonomous System (AS) relationships. Later, Subramanian *et al.* [87] advanced Gao's work by dividing the Internet into five layers. Another similar work can be found in [71]. The main contribution of their works is their observation that the path from one node to another in the Internet first goes "*uphill*" to the uphill top provider, then from the uphill top provider to the downhill top provider, and finally goes "*downhill*" from the downhill top provider to the destination. In Figure 5.2, node A and T in the uphill part stand for the attack node and the ingress edge router of the uphill top provider, respectively, and node T' and V in the downhill part stand for the egress edge router of the downhill top provider and the victim, respectively. Note that in this figure, an uphill vertical line stands for the link from a customer to the edge router of its provider, and a downhill vertical line stands for a link from an edge router of a

provider to its customer, while a horizontal line represents links within one domain (for example, one AS) which may contain several routers. That is, a vertical line represents only one link, but a horizontal line may represent several links as long as these links are in one domain.

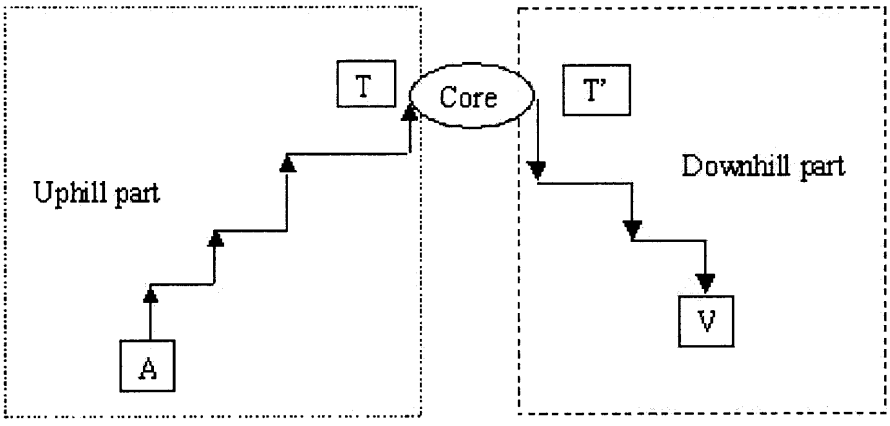


Figure 5.2 A schematic representation of one Internet path.

5.3 Extensions to Pi

As mentioned earlier, 99% of routers in the Internet own no more than eight interfaces [77] (see Figure 5.3). A naïve extension to Pi [44] may use the octary tree with the victim as the root. That is, each node in the tree has no more than 8 children. Since there are eight children, 3 bits are required to distinguish one from another. As shown in Figure 5.4, a path from A to V can be represented by a sequence of bits, e.g., 101110000. Note that this extension is not a "complete" octary tree since some nodes can have less than eight children.

Given the limitation of the 16-bit ID field, using 3 bits for each link can only record information of five links (or hops) out of the possible 32 hops (very few paths in

the Internet have more than 30 hops [47], [88]. A maximum path-length of 32 hops is considered as a common practice). Therefore, further improvements are required.

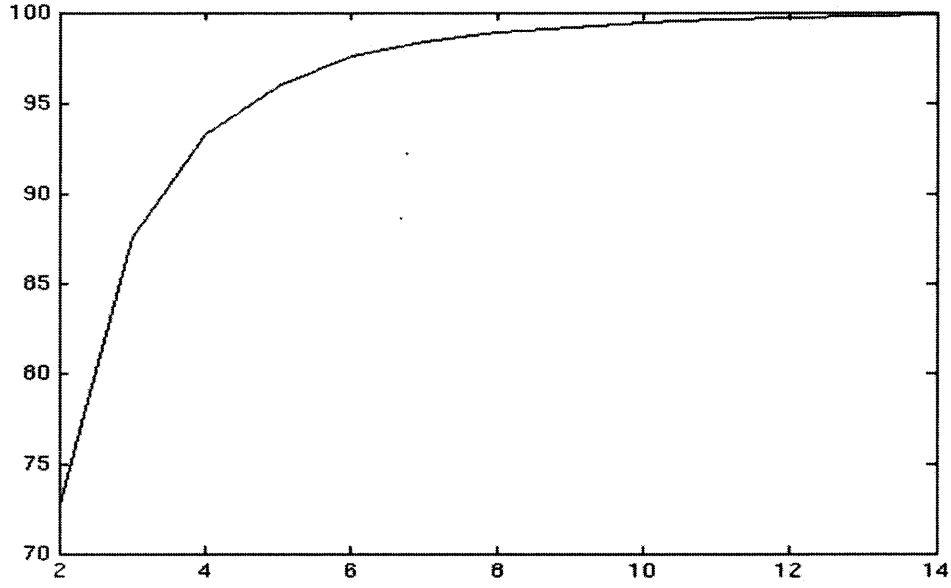


Figure 5.3 The cumulative probability distribution (CPD) of the number of interfaces among routers in the Internet.

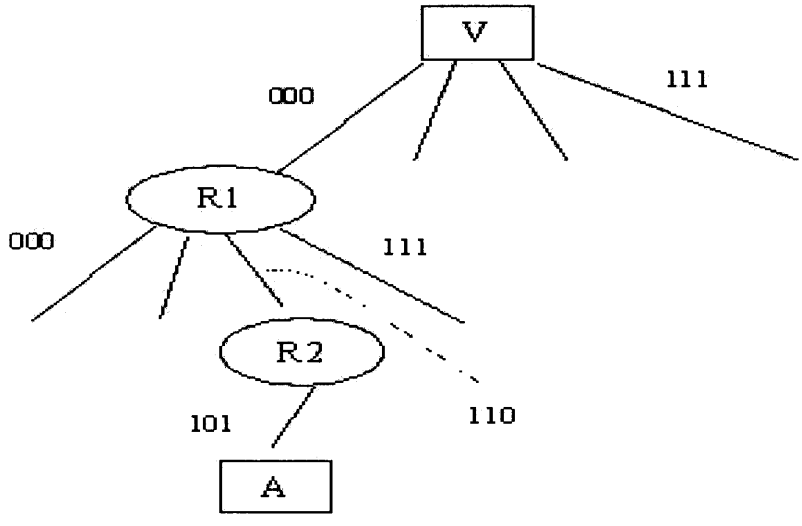


Figure 5.4 A schematic representation of an octary tree.

Note that in the naive extension model, up to eight children are considered for each node. If a node stands for a router, and a branch represents a link between the

interfaces of two different routers, the number of interfaces of a router can be as high as 9. In a *k*-ary tree, there can be up to *k* branches from a node to its children. However, the branch from the node to its parent is not taken into account. Since we only consider 8 interfaces (corresponding to seven children), a 7-ary tree should be used instead of the octary tree. We refer to this as the simple extension model. In fact, a router has one link to its "provider", maybe some to its "peers", and the rest to its "customers". Consider a generic case in which a router has a link to its provider via an interface, called *iprv*, and the rest of the interfaces of the router, called *icst*s, are used for its customers. Then, the color used to mark the link via *iprv* can be *reused* for another link via one of *icst*s, because the two links (reusing the same color) are at different distances (hops) with respect to the victim. A similar reuse example is shown in Figure 5.4, where two links of router R_1 using the same marking '000' are at different distance with respect to V, the victim. If '000' stands for one color (e.g., red), the naïve extension may be viewed as a color-marking scheme.

5.4 The Proposed Scheme

The fundamental problem is that the 16-bit ID field, where the marking is stored, is far less than required. As shown in [44], even if we record only 1 bit of each router's IP address information (not the IP address itself, but rather the mapping function of the IP address, such as using MD5 digest), 16 bits are insufficient. To overcome this problem, Yaar *et al.* [44] proposed to record the first edge router information of one AS rather than all routers along the attack path. Since the scheme is based on the Four Color Theorem, we need to use 2 bits to represent one color. As one link is represented by one color, we

can record eight links at most in the ID field. Obviously, this is not enough; we have to create an innovative scheme to cope with the case of having path lengths with more than eight hops.

5.4.1 Marking Algorithm

We propose to record information of the first and last four links. The goal is to distinguish one path from another, and therefore packets launched from a different source shall have a different marking; packets from the same source but traversing different paths shall have a different marking; only packets from the same source and taking the same path own the same marking. The marking method can satisfy the above requirements. Consider the directed acyclic graph (DAG) rooted at the victim as shown in Figure 5.5. The network as viewed from the victim is a DAG [47], [54]. Note that paths with the same first several links and the same last several links usually traverse the same route. In doing so, we can distinguish different paths with information of only 8 links.

Denote n as the hop count of the attack path. Before we proceed to the marking algorithm, let's consider the following two cases.

1) $n \leq 8$. In this case, the whole path information can be completely recorded.

2) $n>8$. Under this circumstance, we only want to record the first four links and the last four links. However, a router has no idea in advance how many hops there are between it and the victim. That is, routers do not know whether they lie in the last four links or not. To address this problem, we divide the ID field into two parts, the first four links are marked in the lower byte of the 16 bits, from bit 0 to bit 7. The last four links are recorded in bits 8 to 15. The marking algorithm can be summarized in Figure 5.6.

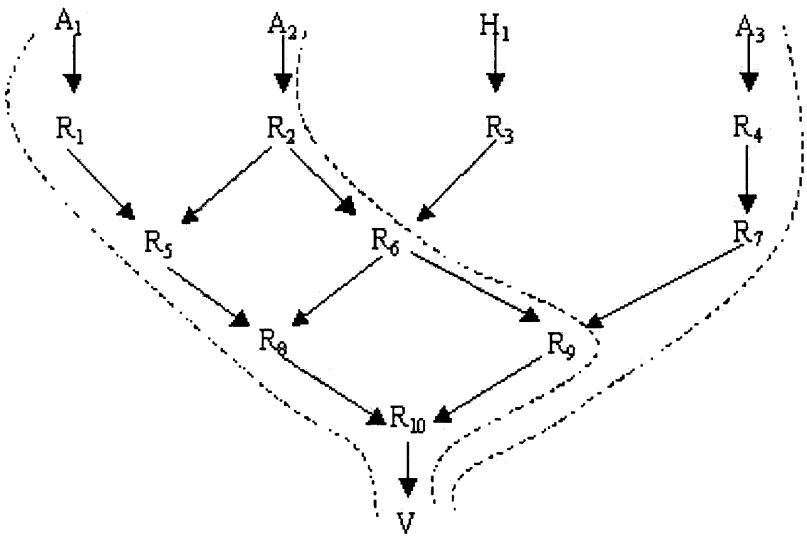


Figure 5.5 Network as seen from the victim, V, of an attack. The dotted lines stand for attack paths.

For the lower byte (bit 7 to 0)

0) index=0,
1) record the color of the current link in the position pointed by the index,
2) index=mod(index+1,4),
3) If index=0, then goto step 4; else, goto step 1.

For the upper byte (bit 15 to 8)

4) record the color of the current link in the position pointed by the index,
5) index=mod (index+1, 4),
6) goto step 4.

Figure 5.6 The proposed marking algorithm.

The index field is 2-bit long (two reserved bits in the IP packet header), which tells the current router the position of the marking. For example, assume the total hop count of a path is 11. Then, the value of the ID field changes in the following order (see Table 5.1).

Table 5.1 How the ID Field Is Marked along the Attack Path

Curr. link	index	Upper byte				Lower byte			
		Pos. 3	Pos. 2	Pos. 1	Pos. 0	Pos. 3	Pos. 2	Pos. 1	Pos. 0
1st	0								C1
2nd	1							C2	C1
3rd	2						C3	C2	C1
4th	3					C4	C3	C2	C1
5th	0				C5	C4	C3	C2	C1
6th	1			C6	C5	C4	C3	C2	C1
7th	2		C7	C6	C5	C4	C3	C2	C1
8th	3	C8	C7	C6	C5	C4	C3	C2	C1
9th	0	C8	C7	C6	C9	C4	C3	C2	C1
10th	1	C8	C7	C10	C9	C4	C3	C2	C1
11th	2	C8	C11	C10	C9	C4	C3	C2	C1

In Table 5.1, each position is composed of two bits, and *Ci* stands for the color of link *i*. When one packet arrives at the first router, the color of this link is recorded in bit 0 and bit 1 of the ID field; then, the color of the second link is saved in bits 2 and 3. Finally, when the packet reaches the victim, the ID field has the value {C8, C11, C10, C9, C4, C3, C2, C1}. This information can be used to actively defend against DDoS attacks. Note the victim can reorder the sequence to {C11, C10, C9, C8, C4, C3, C2, C1} from information provided by the index field (index=2, in this example), thus yielding the marking of the color of the first and last 4 links of the path in the reverse order.

5.4.2 A Related Issue

One problem remains to be addressed is how to assign color to each link. Assume that *N* stands for the total number of interfaces of a router. Let us consider the following cases.

Case 1: $N \leq 4$. In this case, one color is used for one interface. 4 colors are sufficient. Note that, as high as 93.30% of all routers in the Internet own no more than 4 interfaces (see Figure 5.3).

Case 2: $N=5$. In this case, one color has to be reused. Considering the Internet hierarchy, we can reuse the color for the uplink (the link to its providers). Note that 95.93% of routers own less than 6 interfaces.

Case 3: $N>5$. To cope with this case, we take advantage of the Internet hierarchy. According to [71], [73], [87], 3 types of links exist between two routers. That is "uphill" (from a router of a customer to a router of its provider); "downhill" (from a router of a provider to a router of its customer); and "peer" (from a router in a domain to a router of its peer). Assume that the number of links does not exceed 4 for the same type of links. This is reasonable because we only record the first and last 4 links of an Internet path, that are normally located in the access part of the Internet, and therefore the link number of the same type in a router is not expected to be too large. In this way, the same color can be reused in different types of links of the same router. When we need to distinguish the links with the same color (e.g., for IP traceback), say red, we may assign priority to each type of link according to the portions of the path. For instance, in the uphill part of Figure 5.2, for a router R, we first examine whether there exists one "uphill" link marked with red. If this is the case, we view this link as the one we are looking for. Otherwise, we check the "peer" links of router R. If no "peer" link is marked with red, we finally check the router R's "downhill" links. In the downhill part, the order to check whether there is one link marked with red shall be from the "downhill" link, to the "peer" link, and finally to the "uphill" link. In this way, we can reuse the same color.

5.4.3 Benefits

Several benefits can be achieved with this marking scheme. First, the marking scheme is more practical because the average number of interfaces of routers is 3 to 4. Second, we believe that with the first and last portion of the path information, we can reach the same defending effect owing to the DAG model. Third, all required path information is embedded in one packet, and all packets traversing the same path have the same marking. By counting the number of packets with the same marking, the victim can distinguish malicious attack packets from normal packets. Once identified, malicious packets may be dropped or filtered by the victim. This empowers the victim to actively defend against DDoS attacks and facilitate faster response time.

5.5 Simulations

To test the accuracy of the scheme, several simulations on different configurations have been conducted using ns-2. Here, we only present one example. Figure 5.7 shows the network topology used in the simulations. Two scenarios are considered. First, we simulate attack packets launched from different paths. Attackers 1, 2, and 3 traverse nodes 3-15-19-16-17-18 to the victim, node 14. Attackers 4 and 5 take the path 4-15-19-16-17-18, called path 1, to the victim. Attacker 7 goes through nodes 6-18 to node 14. Other nodes send normal traffic. At node 14, we verify and compute the percentage of packets that have been correctly marked by the scheme. The result shows that the marking information is 100% correct. Second, we consider the case that attack packets are launched from the same source but take different paths. This scenario happens for reasons such as load balancing and routing instability. We cut the link between node 16

and 17, and thus traffic from attackers 4 and 5 go through node 4-15-19-16-5-17-18, called path 2, to reach the victim. Note that these packets traversing path 2 have different markings from packets traversing path 1 even though they are from the same sources (attackers 4 and 5). Under this scenario, the marking information of all packets is still 100% correct. The results confirm the correctness of our scheme.

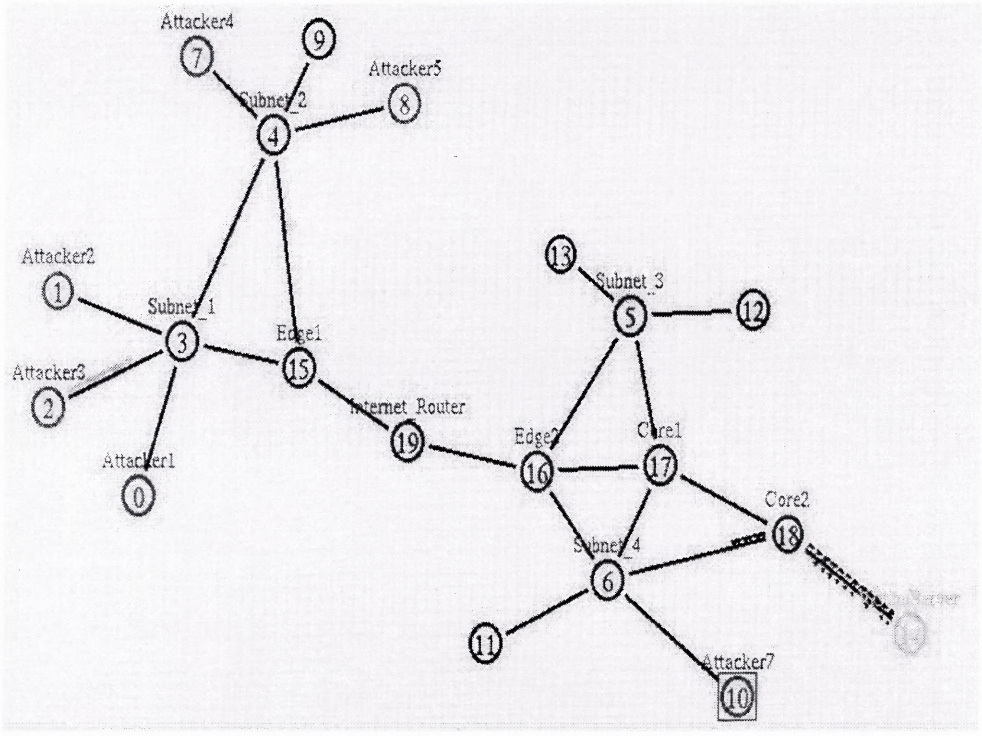


Figure 5.7 The network topology used for simulations.

5.6 Conclusions

In this chapter, we have proposed a new marking method based on the Four Color Theorem that takes advantage of the Internet hierarchy. With this scheme, one serious limitation on the number of interfaces of routers is relaxed, thus rendering the method

more practical. Furthermore, the victim can actively protect itself rather than passively depend on others. Both are very desirable features to defend against DDoS attacks.

CHAPTER 6

A COMPRENHENSIVE FRAMEWORK FOR DDOS DEFENSE

6.1 Introduction

It is well known that DDoS attacks exploit and consume the limited available resources of a system by sending superficially normal but really useless packets to degrade/corrupt the victim system, thus severely hampering the victim system from serving its normal clients. Typical resources that get drained in DoS/DDoS attacks are network bandwidth, CPU cycles of the target host, and specific TCP/IP protocol stack structures such as the fragmentation buffer and TCP SYN buffer. One critical issue in DDoS defense is how to isolate the attack traffic from the normal ones. Traffic differentiation is of great importance because with the knowledge of "good" and "bad" traffic in hand, the victim is ready to defeat a DDoS attack by taking different actions and reacting accordingly.

Given diverse DDoS attack models, another issue of importance is how to contain as many DDoS attack patterns as possible. From the perspective of protocol exploited, DDoS attacks may be TCP, UDP, ICMP, or other protocols based. Some attacks even use the combination of different protocols [9], [10]. From the viewpoint of the attack rate, most attacks are high-speed flood-based while a novel and more sophisticated attack is the low-rate DDoS attack [24], [46]. In contrast, unfortunately, defense schemes do not keep pace with the evolution of DDoS attacks. Most of the previous DDoS defense schemes aim to address one or two types of DDoS attacks and are thus inefficient and ineffective to such a wide spectrum of attacks.

We hereby propose a new framework that attempts to isolate attack traffic and contain as many DDoS attacks as possible. We view DDoS attacks as a resource management problem, and thus adopt QoS means to combat them [89], [90], [91]. Using QoS techniques is not a brand new idea to tackle DDoS attacks, our contributions here lie on an integrated scheme to efficiently and effectively contain a variety of DDoS attacks rather than one or two attacks.

Probably the works most related to ours are perimeter based DDoS defense [92] and using QoS to regulate the resources [89], [93]. Chen *et al.* [92] proposed to establish the line of defense at the perimeter of the protected system and presented two schemes based on multicast and traceback, respectively. A similar method is used in [93] to protect Web servers. Garg *et al.* [89] proposed an aggregated resource regulation mechanism to battle DDoS attacks. Furthermore, their scheme is not attack specific and may handle several attacks. Their scheme tries to maintain fair share among all competing flows by using the fair packet queueing technique. Simply using fair sharing without traffic discrimination does not work (similar limitation can be found in [91]). Consider the following two scenarios. 1) The high rate traffic is legitimate while the attack traffic is low-rate. Rate-limiting is improper in this case. 2) Most flows carry attack traffic during a flood-based DDoS attack while good traffic is low-rate. Under this scenario, even imposing fair sharing of bandwidth does not help the "good" traffic much because the majority of bandwidth is consumed by malicious flows. To ensure good performance and accommodate as many normal users as possible, it is critical to differentiate traffic. The differences between their schemes and ours include 1) their

scheme is passive in nature while ours are proactive, and 2) their method is incapable of isolating malicious flows while ours can.

Our framework is comprised of two components. The first component classifies different types of traffic based on the protocols used and limits their rates accordingly. This step serves to isolate UDP, ICMP and other traffic from TCP, and is helpful to mitigate some flood-attacks based on UDP and ICMP via limiting bandwidth allocation to non-TCP traffic. Since TCP is the dominant traffic in the Internet and most DDoS attacks are based on TCP [7], the next step is to differentiate disparate TCP traffic. All TCP traffic is categorized into two groups according to the status of the connection establishment. Among all connection-established TCP traffic, our scheme strives to identify the property of a flow, benign or malicious, according to its behavior. A flow is defined as "benign" or "normal" if it responds to the control signal of the other endpoint of the same connection appropriately. On the contrary, the malicious flows are those that do not follow the TCP congestion control principle and act aggressively. It is possible that a TCP unfriendly flow is classified as a lethal flow, and is punished by our scheme. This is the collateral side effect of our scheme because we currently do not distinguish between a TCP unfriendly flow and an attack flow. Based on the distinction, certain penalties may be imposed to the aggressive flows. Our design is “receiver-centered”, and does not require any modification of the source endpoints or intermediate routers, thus avoiding the issues of scalability, cooperation between different domains, and lack of incentives. Extensive simulations using ns-2 have been conducted to validate our design. Preliminary results show that traffic classification can improve the throughput of TCP traffic significantly (over 70%) while traffic differentiation can quickly block malicious

attack traffic. Other benefits of our mechanism include (1) minimal requirement of modification, thus practically deployable; (2) no issue of scalability because the deployment is at the receiver side only; (3) no issue of lack of incentives since the deployment of our scheme is served to solely protect one's own networks and hosts, not others.

6.2 The Proposal

6.2.1 Design Philosophy

As mentioned earlier, an ideal defense scheme shall be able to distinguish between attack flows and normal ones, and isolate deleterious traffic accordingly. However, it is by no means trivial to make such a distinction. One such effort was conducted by Xu *et al.* [93]. They proposed to isolate malicious traffic via HTTP redirect messages. For attack flows employing spoofed source IP addresses, their sources cannot receive the redirect messages and thus the subsequent packets from them will be blocked. This scheme is simple and may be readily implemented. However, their mechanism works only for web servers. Another deficiency is that their proposal cannot handle attack packets using genuine source IP addresses, e.g., reflective DDoS attacks where IP spoofing is not employed. Anderson *et al.* [34] proposed that the sender should get a token from the receiver and embed the token inside each packet it sends. This method needs to modify TCP implementations at both endpoints and slightly changes the TCP semantics as well. Ioannidis and Bellovin [41] proposed a general bandwidth flooding control method, called pushback. The attacked target generates the signature of the attack flows, and asks its upstream routers to filter them. The limitations of the pushback scheme lie in the

difficulty to precisely extract characteristics of attack flows, especially at line rates. One impediment is the diverse attack traffic, combining different protocols and different contents. Another is how to distinguish between good and bad traffic intellectually. Discrimination based on packet headers is vulnerable to IP spoofing; discrimination based on packet contents may be thwarted by the increasing use of end-to-end encryption.

We hereby propose to identify malicious traffic from their behaviors. We believe that aggressiveness is the salient feature of DDoS traffic, besides IP spoofing. One example of the aggressive behavior is that an attack source may not care about whether it may receive the response from the victim or not, and it can still conduct an attack by bombarding its target with a monstrous number of useless packets. Even the low-rate DoS attack that mounts attack packets sporadically using an on-off model demonstrates such aggressive feature during the "on" stage [24]. Note that "aggressiveness" is not equivalent to "high-rate". It is possible that a high-rate flow is a normal TCP stream. The receiver may identify the aggressive behavior by intentionally testing the response of a source upon certain control signals from the receiver. Any source which fails to pass such tests is regarded as a lethal one and can be punished accordingly. However, a source, which passes the test, may not be necessarily benign. We urge the receiver (usually a server) to conduct the test upon the receipt of a requirement for a connection. A sophisticated attacker may pass the test by behaving well initially, and perform deleterious operations later. To handle this case, the receiver may increase the frequency of such tests. A better solution is to introduce some dynamics into the test and randomly determine the frequency and interval of the test for each flow, especially the high-rate one. To accommodate high-rate legitimate traffic better, we set a threshold that defines

the maximal number of successful tests for a flow. No more tests are conducted on packets from a flow once the flow successfully passes the specified number of tests. By actively testing a source, the receiver can determine with high confidence the nature of a flow from that source and react accordingly. Filtering based on behavior brings an attack source into a dilemma: sends packet aggressively at the risk of being identified and punished, or reduce the attack rate to meet the requirements of the receiver so that the effect of an attack is diminished. In so doing, the receiver may throttle the scope and impact of potential attacks.

The above design is feasible for TCP solely because TCP has the built-in congestion control and reliable transmission mechanism. Note that TCP is the dominant traffic in the Internet, and as much as 90% of DDoS traffic use TCP [7]. Currently, TCP occupies 80% in terms of the number of flows, and 90% with respect to the number of packets [94], [95]. It is thus essential for DDoS defense schemes to accommodate TCP traffic effectively and efficiently. However, other traffic such as UDP and ICMP lack alternative mechanism that is exploitable by our framework. We propose to use traffic classification to handle non-TCP traffic. Traffic classification is a simple and efficient solution to handle flood attacks based on UDP and ICMP. By limiting the resource allocated to UDP and ICMP traffic, the receiver may significantly mitigate UDP and ICMP flood attacks. Another benefit of traffic classification is that UDP traffic is much more aggressive than TCP traffic. It is well established that when UDP traffic competes for bandwidth against TCP traffic, the former is capable of seizing most of the available bandwidth due to its lack of any congestion control mechanism and being more aggressive in nature. Therefore, sending UDP traffic may deprive the fair share of

bandwidth from the TCP traffic effectively, and degrade the performance of TCP streams. It thus turns out that UDP is a good vehicle for DDoS attacks. This demonstrates the difficulty to contain different kinds of DDoS attack patterns in one scheme, and justifies the necessity to perform traffic classification.

Other issues need to take into account include (1) how to address IP spoofing? (2) at which level our framework should be performing, packet level or flow level? We first try to identify attack traffic with forged sources IP addresses, and then discern the traffic with genuine IP addresses by observing their behaviors. Spoofing may be addressed by means of verification while aggressive behavior may be identified by manipulating the rate of returned ACK at the recipient side for TCP traffic. This method is also capable of identifying the more subtle and sophisticated low-rate DDoS attacks. For a heavily loaded receiver (e.g., a server) that serves a huge amount of packets from disparate sources, it shall make a fast and smart decision to admit or block an incoming packet. The ever-increasing link-speed motivates us to contrive countermeasures at the flow level. Figure 6.1 depicts the flowchart of our proposed framework.

6.2.2 Traffic Classification and Bandwidth Allocation

6.2.2.1 Traffic Classification. Traffic classification is a simple procedure in our scheme. Since UDP lacks built-in congestion control mechanism and is aggressive in nature, we first classify traffic based on protocols to isolate TCP and non-TCP traffic. Specifically, we propose to categorize traffic into 4 groups, that is, TCP, UDP, ICMP, and other. This may be readily implemented by checking the protocol field at the IP layer. For example, a system administrator may configure his/her network and assign the bandwidth in such

way, TCP: 85%, UDP: 13%, ICMP 1%, and others 1%. This specific configuration is determined by the site's profile, and may be changed dynamically. The goal of traffic classification according to normal traffic profile is to mitigate the impact of possible UDP and ICMP flood attacks. Unlike TCP traffic, whose property may be identified according to its behavior, UDP is aggressive in nature. As a less "reliable" protocol, it turns out that no easy way may be taken to distinguish between normal UDP traffic and attack UDP traffic. Imposing bandwidth limit is a simple but working strategy, and it does not affect the network performance in most cases. Limiting the amount of ICMP traffic allowed into a network is a good practice because it is used solely for diagnostic purpose and ICMP-based DDoS attack is common.

Traffic classification may be implemented at the entry point of the protected networks or systems, either at a firewall or an edge router (see Figure 6.2). The availability of network processors and other ASICs specifically designed for classification renders it possible for our scheme to be operated at the line rate.

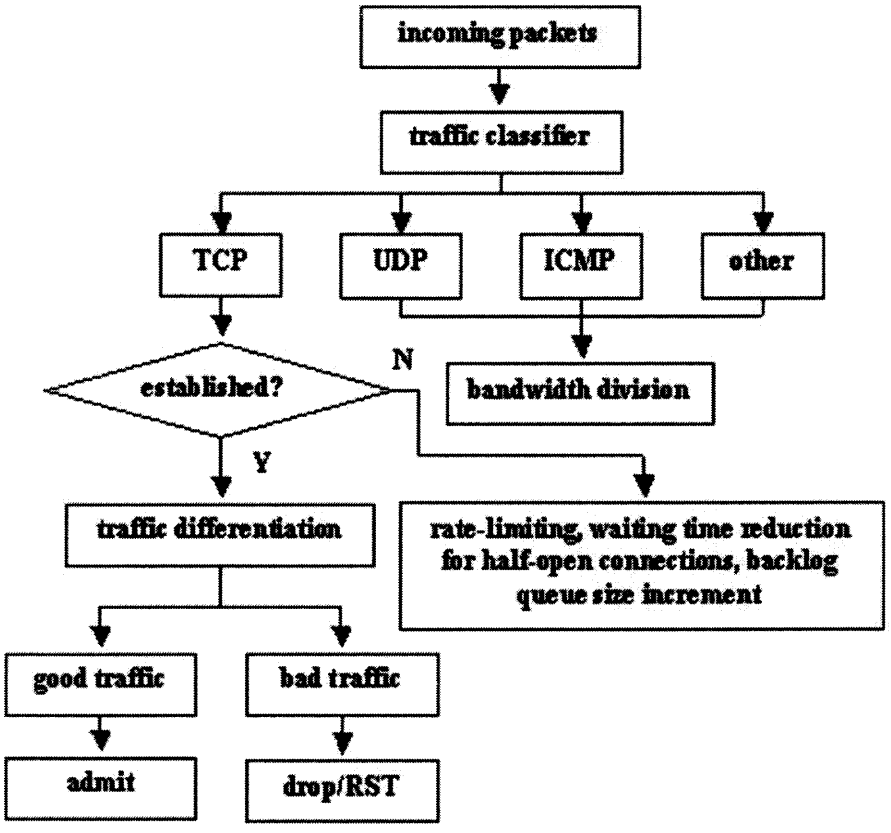


Figure 6.1 Flowchart of the proposed framework.

6.2.2.2 Bandwidth Division. The quota for each protocol is configurable and it is normally determined by the routine traffic profile at one site. Several recent measurements show that the dominant traffic in the current Internet is still TCP. It is thus reasonable to assign most of the bandwidth to TCP traffic for a normal site because most killer applications such as web, email, telnet, and FTP all use TCP. For example, a possible bandwidth allocation among different protocol is TCP: 85%, UDP: 13%, ICMP: 1%, and other: 1%. In case of a traffic pattern change in the future, what an end user needs to do is to adjust the bandwidth allocation percentage according to the new traffic model, and thus our scheme can be easily tailored for future traffic changes.

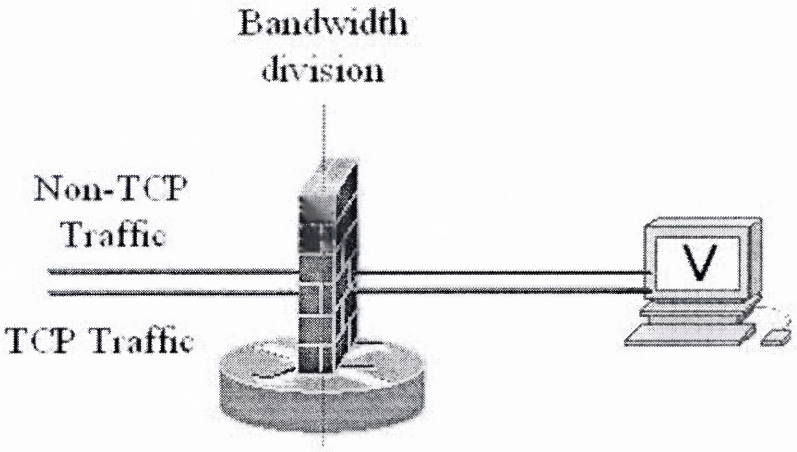


Figure 6.2 Deployment point for traffic classification.

6.2.3 TCP Flow Differentiation

6.2.3.1 Connection Establishment. Whether a connection has been established has a significant implication to the receiver. A successfully established connection indicates that both ends have completed the three-way handshaking procedure, which implies that IP spoofing is not used by the source. For an incomplete connection, on the other hand, the receiver shall be alert, and be conservative in its resource consumption. Possible measures to mitigate potential attacks include (1) tightening the total bandwidth allocated to all incomplete connections, and (2) significantly reducing the timeout value to avoid buffer occupied by half-open connections for a long time, or no buffer allocation at all for half-open connections. For example, many proposals have been presented to handle SYN flood attacks [36], [96].

6.2.3.2 Benign and Malicious Flows. After assigning the quota for each protocol, the next task is to isolate lethal TCP flows from normal TCP flows simply because TCP consumes most of the bandwidth of the protected network. As mentioned earlier,

straightforwardly adopting fair packet queuing is impractical in case of flood-based DDoS attacks when a large majority of traffic is attack streams. Therefore, traffic differentiation and isolation becomes indispensable. TCP is an end-to-end solution that requires close orchestration between the sender and the receiver. To characterize the nature of a TCP flow (after a successful connection), the receiver can actively test the response of the sender by delaying the ACK packets intentionally. If the sender is normal, it will take action accordingly and reduce its sending rate. On the contrary, for a DDoS attack, two cases may occur. One is that the sender uses forged source IP addresses, and thus cannot receive the rate-reduction message from the receiver. It has no idea of the proper sending rate. The other scenario is that the sender does receive the notification, but it neglects it and just keeps sending packets, thus violating the protocol, and it may be punished by the recipient to reduce its share or even block its traffic. This procedure is dynamic. The protected site can decide the frequency and extent of rate-reduction so that no perpetrator can easily fool the system to believe that the traffic from the perpetrator is normal.

Figure 6.3 depicts the flowchart of our traffic differentiation procedure. Upon the arrival of a new incoming packet, the receiver first determines the flow the current packet belonging to by checking the tuple of (source IP address, source port number, destination IP address, destination port number). If it is the first packet of a flow, the receiver examines whether the number of admitted flows reaches the maximal flow count, a threshold set by the receiver to ensure proper provisioning of quality of service. If this does occur, the packet is dropped. Otherwise, the new packet is admitted after updating the flow table maintained by the receiver, incrementing the flow count by 1, and

initializing several counters, such as the number of successful tests and the number of failure tests. For the packet of an existing flow, the receiver checks the behavior history of the flow. If the number of failure tests is no less than a threshold, f, the packet will be dropped. An integer larger than 1 is selected to prevent our scheme from falsely identifying the behavior of a flow. A low value of f may exacerbate packet dropping. In case of a false identification, subsequent packets from an innocent flow will be blocked. Selecting a too high value is unwise, either. A high f delays the packet dropping decision, and thus subsequent packets of a malicious stream may still consume system resources. Through extensive simulations, we found that f=3 provides a good balance between the proper identification rate and the acceptable performance impact. It is also worth to mention that a receiver has a couple of options to choose to punish the source at this point. One option is to send DUPACK on purpose forcing the source into the stage of slow start. Another is to send RST to halt the connection so that its resource is not wasted by the misbehaving sources. In Figure 6.3, we only show the operation of packet dropping without any further punishment.

For the flow whose behavior is not so bad in the past, our scheme further examines whether the flow has passed a certain number of tests, h. The receiver will admit directly any packet of flows having passed h tests successfully (Similarly, some tradeoff has to be made to determine a proper value of h. We set h to 6 by trials and errors). For other flows, we further check the current state of the flow. If the flow is under a test, its current rate shall not exceed one half of its previous one (the receiver enforces this constraint by manipulating the reverse ACK rate). If the flow conforms to that constraint, the flow passes the current test and its *pass_num* is incremented by 1.

Otherwise, the flow fails one test. In the case that the flow is not in the state of testing, its sending rate is compared with that of the fair share of each flow. The result of the comparison is used to determine the test probability for that flow. Obviously, a flow with less bandwidth consumption is subject to less number of tests. The test probability p for a high-rate flow (over the faire share) is $1/(pass\_num+1)$. At the very beginning, *pass_num* is 0 for all flows. Therefore, as long as a high-rate flow has not passed a test, its chance of being tested is 100%. As the number of successful tests of a flow increases, its test probability reduces. The test probability p for the less resource-consumption flow is $1/\max(m, 2*h)$, where m is the total number of flows. For the normal case, $m>>2h$; thus, $p=1/m$. We use the max(.) function to address the case that only a few flows exist in the system and ensure that the test probability for a low-rate flow is at most 1/2 of that of a high-rate one.

The rate of a flow is calculated according to the following formula,

$$(num\_pkt*sz\_pkt)/t, \tag{6.1}$$

where t is the time interval (window), *num_pkt* the number of packets received during this period, and *sz_pkt* the packet size. It is worth mentioning that the flow rate calculated here is not the average rate of a flow, as normally used by others, because we update the starting time of a flow once it passes a test. In so doing, we can effectively thwart a low rate DoS attack which sends a burst of attack packets to incite congestion and keeps silence for a much longer period to significantly lower its average rate in order to escape detection and filtering [24].

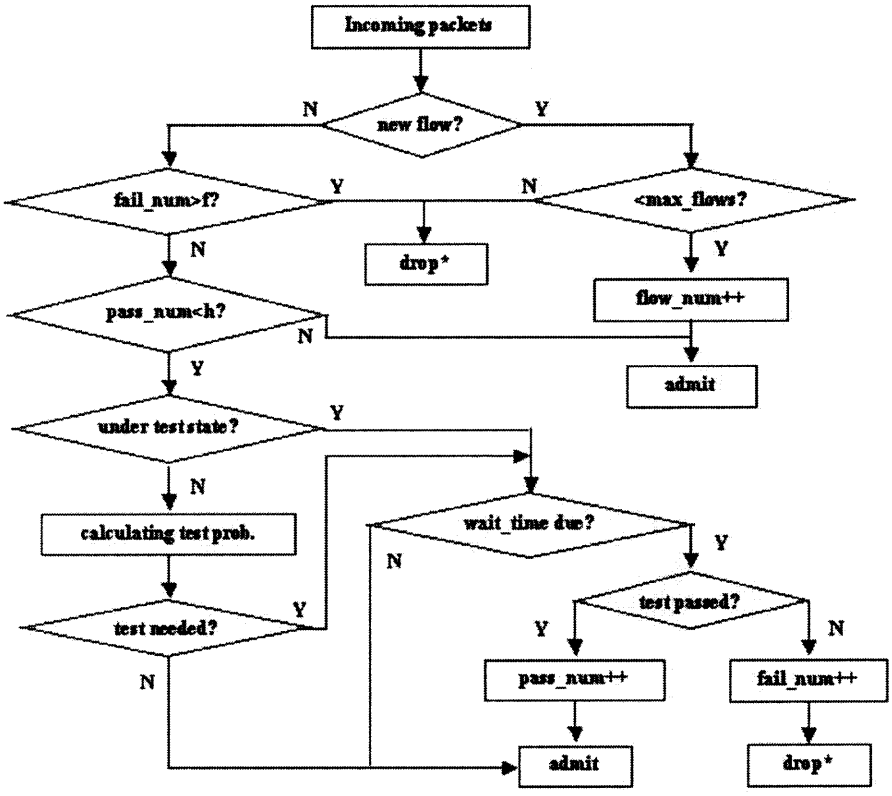


Figure 6.3 Flowchart of traffic differentiation procedure.

Four scenarios may happen. 1) An attack source always behaves well, and thus the effect of an attack is greatly diminished. 2) An attack source behaves well at first and misbehaves later. When tested, the constraint that the current rate is at most 1/2 of previous rate will not be satisfied, and the source fails the test. 3) An attack source always misbehaves; this may be easily thwarted by the fail count. 4) An attack source misbehaves at first and behaves well later. In this case, the attack source is exposed to more chances of being tested because its *pass_num* is offsetted by the *fail_num* once it fails a test. Note also that a low-rate flow is also subject to test, though at a lower probability in our design. As time passes by, the chance that a low-rate flow has never

been tested by the receiver is very low. We enforce this policy to contain the case that some low-rate streams are malicious.

6.2.3.3 Penalty to Bad TCP Flows. A receiver has a couple of options to choose to punish the source. One option is to send DUPACK on purpose, thus forcing the source into the stage of slow start. Another is to send RST to halt the connection so that its resource is not wasted by the misbehaving sources. Ebrahimi *et al.* [97] recently pointed out that a TCP flow in slow start stage may capture more bandwidth than long-living TCP flows. Thus, we believe sending RST is a better choice once the flow fails the tests three times. In Figure 6.3, we only show the operation of packet dropping without any further punishment.

In summary, we first try to reduce the volume of non-TCP traffic via traffic classification, thus mitigating the popular UDP and ICMP flood attacks. For TCP flows, we attempt to distinguish their nature by observing their behaviors. Table 6.1 lists the policies we adopt in our framework to accommodate the good traffic and to constrain the bad traffic.

Table 6.1 Measures to Address Disparate Traffic Models

traffic nature	Measures
high-rate attack traffic	1) high test probability 2) dynamic in terms of detection frequency and intervals 3) after certain times of failures, all packets of the flow are dropped
high-rate good traffic	1) after passing the test for certain times, all packets of the flow are accepted
low-rate attack traffic	1) still subject to tests, though at a low probability. As time passes by, its chance of being tested increases. 2) Once a flow passes a test, its rate is calculated from the time when it passes the last test. Thus, a flow adopting the strategy of low-rate first and high-rate later cannot escape being identified. 3) after certain number of failures, all packets of the flow are dropped
low-rate good traffic	1) low test probability 2) after passing the test for certain times, all packets of the flow are accepted

6.3 Simulations

We have conducted a large volume of simulations to validate our idea. The preliminary results are promising. Here, we present some representative results.

6.3.1 Traffic Classification

To test the effectiveness of traffic classification, we set up a simple simulation scenario including 1 FTP source and sink (using TCP), and 1 CBR source and sink (using UDP), as shown in Figure 6.4. These flows pass through the same bottleneck link. The link parameters between the FTP source and the checking point, the CBR source to the checking point, destination and CBR sink, and destination and FTP sink are all 10Mb in

bandwidth, and 2 ms in delay. The bottleneck link between the checking point and the destination is set to be 1 Mb and 10ms. The CBR rate is 10Mb. The traffic classification for TCP is 90%, and UDP10% at the checking point. In the comparison of simulations, everything is the same except whether the entry point enforces traffic classification or not. The throughput of the FTP traffic is presented in Figure 6.5.

Figure 6.5(a) depicts TCP goodput and TCP throughput for the TCP flow and UDP throughput without the traffic classification policy. In Figures 6.5(a) and 6.5 (b), TCP traffic starts at time 0s and UDP traffic begins 1s later. During 0-1s, TCP is the only traffic in the link and its goodput and throughput reaches 12 packets per 0.1s, the maximal value (the bandwidth of the bottleneck link is 1mb/s, each packet is 1000*8=8000 bits, 1mb*0.1/8000=12). From 1.2s on, UDP traffic starts to merge and it captures the available bandwidth so fast that TCP throughput reduces to 1 packet during 1.3-1.8s, starving since 1.9s. With traffic classification, in contrast, TCP throughput remains 3 packets per 0.1s even after 1.9s, while UDP traffic is no more than 9 packets per 0.1s. The total number of packets sent is 127 in Figure 6.5(a), and 234 in Figure 6.5(b) (we only depict the throughput till 1.9s in Figure 6b for comparison). Therefore, the throughput of TCP traffic is increased by 70.8% with traffic classification. Though UDP traffic still seizes more share than TCP, as explained earlier, this is due to the built-in congestion control mechanism of TCP.

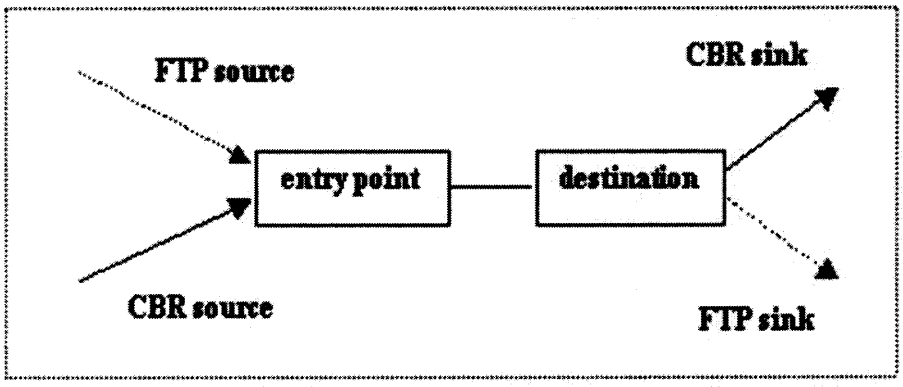
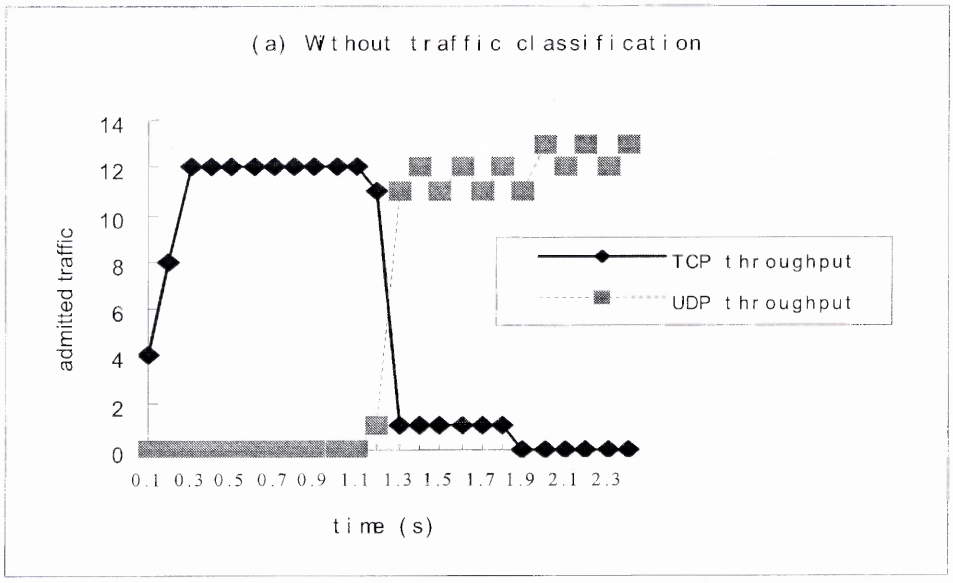


Figure 6.4 Simulation setup for comparative study of the effectiveness of traffic classification.

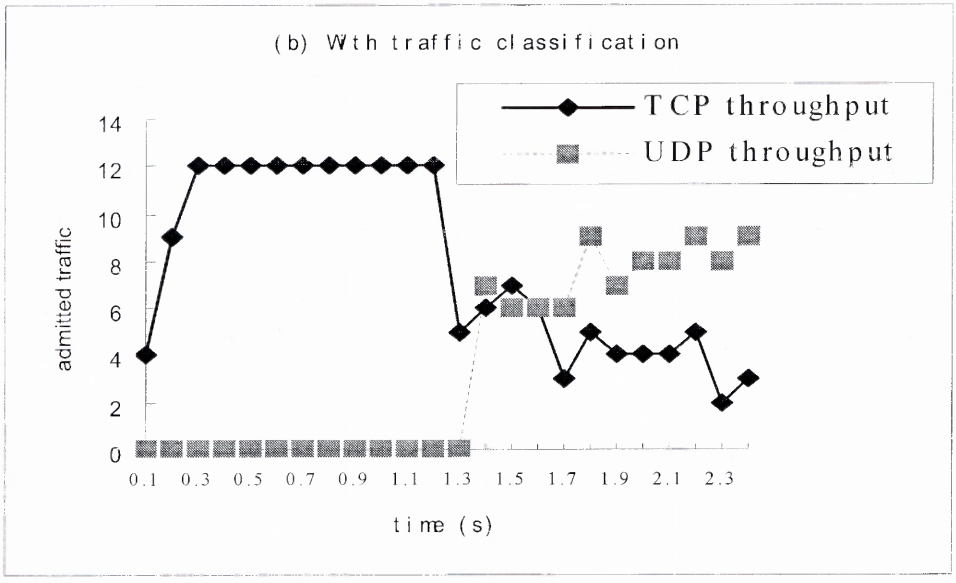
6.3.2 TCP Flow Differentiation

To test the effectiveness of our proposed traffic differentiation, we set up a similar simulation scenario including 1 FTP source and an attack source, as shown in Figure 6.6. These flows pass through the same bottleneck link. The difference is that one simulation uses a normal FTP sink to accept packets from both flows, and the other uses our developed TCP sink, called TCP smart sink. The simulation results are shown in Figure 6.7.

Figure 6.7(a) shows the throughput of the attack traffic using the FTP sink while 6.7(b) presents the throughput of the attack traffic using our proposed TCP smart sink, in which the throughput of attack traffic drops drastically after 3.2s. After 42.3s, the attack traffic is totally blocked. In contrast, using the FTP sink as the receiver, the attacker may keep the highest throughput during its lifetime. The result demonstrates the effectiveness of the traffic differentiation.



(a) Without traffic classification



(b) with traffic classification

Figure 6.5 Study of the effectiveness of traffic classification.

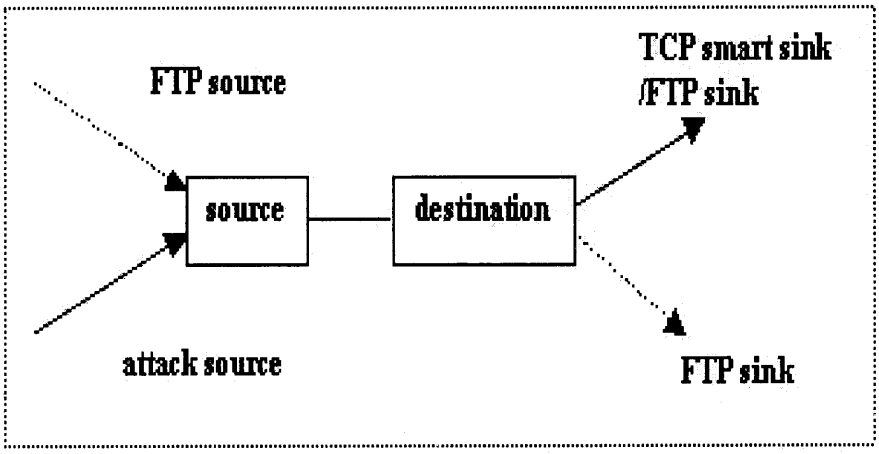
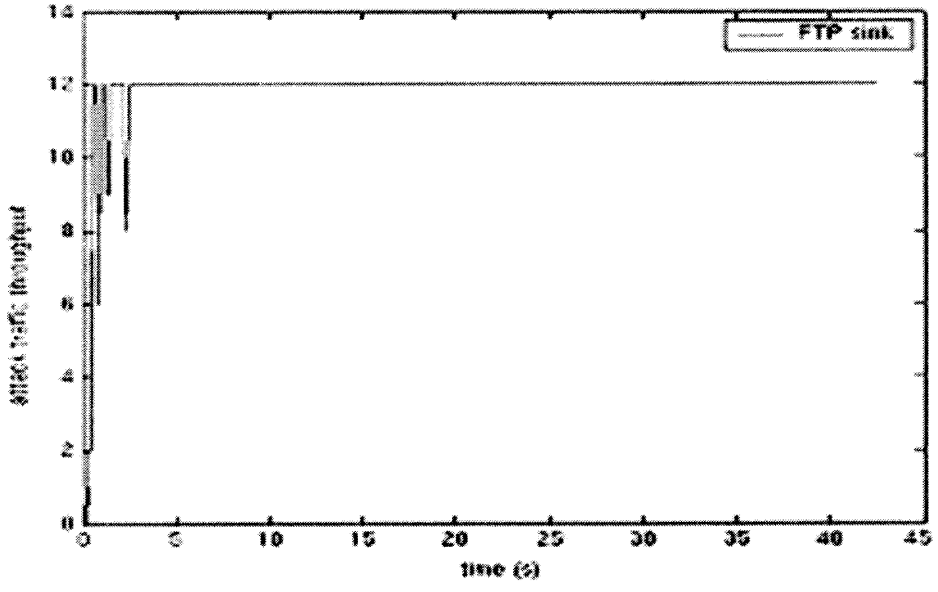
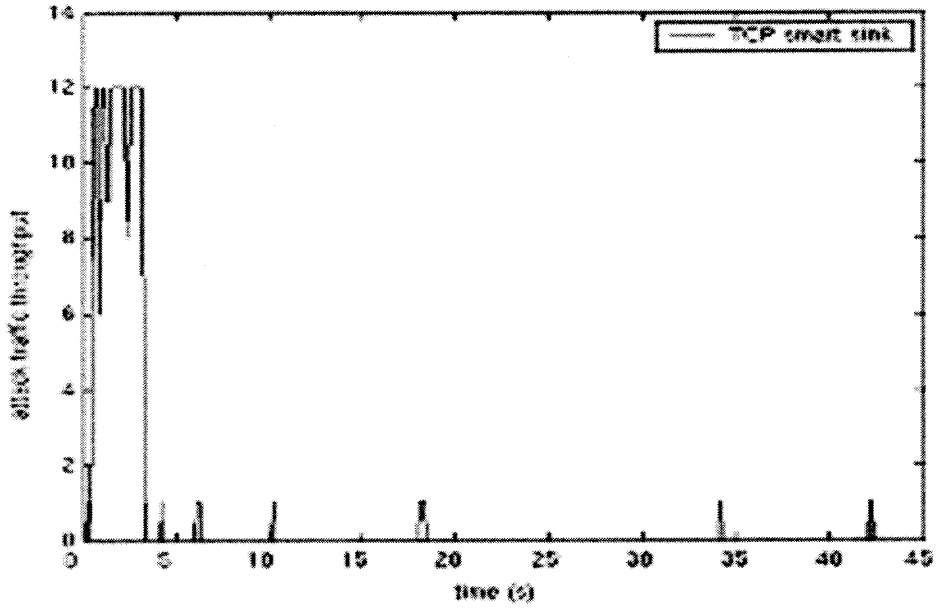


Figure 6.6 Simulation setup for comparative study of the effectiveness of traffic differentiation.



(a) Using FTP sink (without traffic differentiation)



(b) using TCP smart sink

Figure 6.7 Attack traffic throughput with different sinks.

Next, we consider the performance impact of TCP smart sink to the normal traffic. To that end, we set up a similar simulation scenario including n ($1 \le n \le 30$) FTP sources, as shown in Figure 6.8. These flows pass through the same bottleneck link. The difference is that one simulation uses the normal FTP sinks to accept packets from all flows, and the other uses TCP smart sinks. The simulation results are shown in Figure 6.9. Clearly, the performance of the TCP smart sink is very close to that of the TCP sink for normal TCP traffic. Therefore, TCP smart sink can differentiate malicious attack traffic while having little impact on the normal traffic. The result demonstrates the correctness of the traffic differentiation.

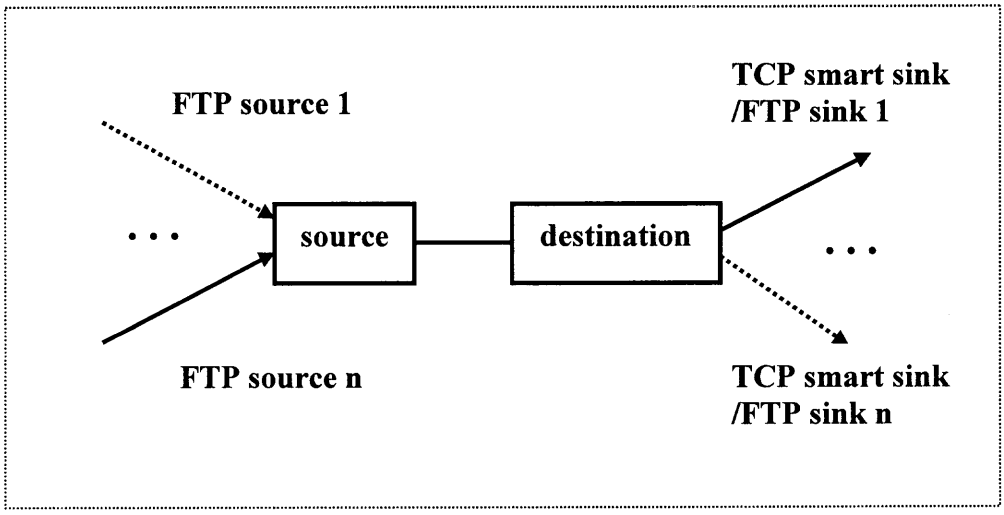


Figure 6.8 Simulation setup for comparative study of the impact on performance by traffic differentiation, where n is in the range of [1,30].

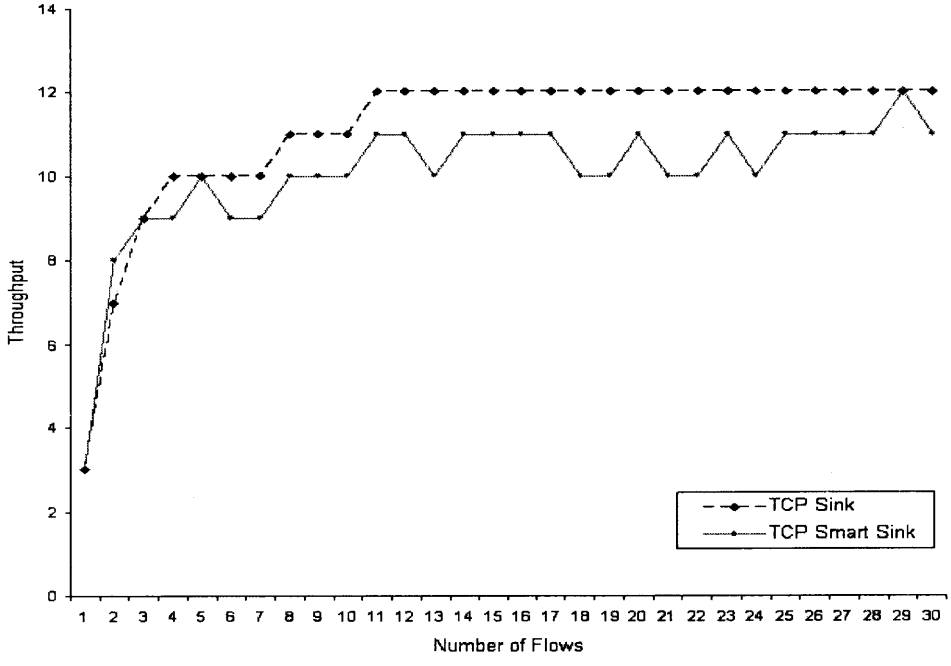


Figure 6.9 Normal traffic throughput with different sinks.

6.4 Conclusions

A comprehensive DDoS defense framework has been presented in this chapter. Two components of our framework are traffic classification and traffic differentiation. The former is used to reduce the volume of non-TCP traffic while the latter can identify malicious TCP flows by proactive tests. The salient benefits of our proposal are listed as follows:

- It can effectively and efficiently identify and block attack flows via proactive tests.
- It can contain many DDoS attack patterns.
- It requires minimal modification.
- No issue such as scalability and lack of incentives.

Preliminary simulation results have validated our design. We plan to further enhance our proposed framework in the following aspects: 1) identifying the network condition automatically, 2) selecting various parameters used for proactive testing adaptively, and 3) implementing the framework in Linux kernel.

CHAPTER 7

CONCLUSIONS AND FUTURE WORKS

This dissertation has presented several IP traceback and DDoS defense schemes. Among proposed IP traceback schemes, ASEM is promising. We first identify six drawbacks of Probabilistic Packet Marking (*PPM*), and then contrive a synergic scheme to address all of them. To relieve the victim from the daunting computational overhead, we derive the optimal marking probability with respect to the number of packets required for path reconstruction, and explore two different approaches to enhance PPM. In so doing, computational burden and spoofed marking inscribed by the attacker are thwarted. Next, we study the issue of bogus marking incurred by subverted routers. By coupling the marking and routing information, a downstream router can examine the correctness of the marking provided by upstream routers, thus eliminating the spurious marking embedded by subverted routers. Our coarse-grained marking tactic (marking at the AS level rather than hop-by-hop IP level) brings two additional benefits: our scheme can effectively suppress false positives, and partial deployment of our scheme may achieve the similar effect as global deployment in the power-law Internet. Finally, we evaluate and analyze the performance of our proposal on empirical Internet measurement data. Results show that as many as 90.67% of marked packets required for path reconstruction may be reduced on average while false positives are greatly suppressed and robustness is significantly enhanced.

As DDoS defense schemes are concerned, one critical issue in DDoS defense is how to isolate the attack traffic from the normal ones. Traffic differentiation is of great

importance because the goal of DDoS attack is to severely degrade the performance of target hosts and networks or even completely deprive the victim of the capability of serving its normal clients. With the knowledge of "good" and "bad" traffic in hand, the victim is ready to defeat a DDoS attack by taking different actions and reacting accordingly. Given the diverse DDoS attack patterns, another issue of importance is how to contain as many DDoS attack patterns as possible. We hereby propose a novel framework that can proactively identify and deter most of the malicious traffic, and tackle a variety of DDoS attack patterns as well. Two main components of our framework are traffic classification and traffic differentiation. The former aims to address non-TCP flood attacks while the latter is used to identify malicious TCP flows. Our framework is implemented using ns-2. Preliminary results show that traffic classification can improve the throughput of TCP traffic significantly (over 70%) while traffic differentiation can quickly block malicious attack traffic. Other benefits of our mechanism include (1) minimal requirement of modification, thus practically deployable; (2) no issue of scalability because the deployment is at the receiver side only; (3) no issue of lack of incentives since the deployment of our scheme is served to solely protect one's own networks and hosts, not others.

Mitigating DDoS attacks is a difficult and challenging task. The open issues include reflective DDoS attacks, and integrating DDoS detection, DDoS defense, and IP traceback. Reflective DDoS attacks introduce another level of indirectness, which represents a big challenge to IP traceback because coordination between different administrative domains is indispensable. However, no efficient and reliable trust relationships are maintained in the Internet nowadays. Another issue is how to integrate

the DDoS detection, defense, and IP traceback schemes effectively. An ideal solution can detect any intrusion precisely, and trigger the proper defense and traceback intelligently. To that end, more research efforts are required to characterize a flow and explore the better defense and traceback mechanisms, in addition to the requirement of trust relationship between different administrative domains. It may require considerable efforts of the security community.

REFERENCES

1. CERT/CC Statistics. Retrieved April 8, 2006 from the World Wide Web: http://www.cert.org/stats/cert_stats.html.

2. W. Stallings, Network Security Essentials: Applications and Standards, Prentice Hall, 2000.

3. R. Anderson and J. H. Lee, "Jikzi-a new framework for security policy, trusted publishing and electronic commerce," *Computer Communications*, vol. 23, no. 17, pp. 1621-1626, 2000.

4. L. Garber, "Denial-of-Service attacks rip the Internet," *IEEE Computer*, pp. 12-17, April 2004.

5. K. Poulsen, "FBI busts alleged DDoS Mafia." Retrieved April 8, 2006 from the World Wide Web: http://www.securityfocus.com/news/9411.

6. D. Moore and C. Shannon, "SCO offline from Denial-of-Service attack." Retrieved April 8, 2006 from the World Wide Web: http://www.caida.org/analysis/security/sco-dos/.

7. D Moore, G. Voelker, and S. Savage, "Inferring Internet Denial-of-Service Activity," *Proc. USENIX Security Symposium*, Washington, D.C., Aug. 2001, pp. 9-22.

8. Computer Security Institute and Federal Bureau of Investigation, "2004 CSI/FBI Computer Crime and Security Survey." Retrieved April 8, 2006 from the World Wide Web:http://i.cmpnet.com/gocsi/db_area/pdfs/fbi/FBI2004.pdf.

9. C. Douligeris and A. Mitrokotsa, "DDoS attacks and defense mechanisms: classification and state-of-the-art," *Computer Networks*, vol. 44, pp. 643-666, April 2004.

10. J. Mirkovic and P. Reiher, "A taxonomy of DDoS attack and DDoS defense mechanisms," *ACM Computer Communications Review*, vol. 34, no. 2, pp. 39-53, 2004.

11. C. Douligeris and A. Mitrokotsa, "DDoS attacks and defense mechanisms: A classification" *Signal Processing and Information Technology (ISSPIT)*, 2003, pp. 190-193.

12. Z. Gao, N. Ansari, "Enhanced probabilistic packet marking for IP traceback", *Proc. IEEE GLOBECOM'2005*, Nov. 28-Dec.2, 2005, St Louis, MO.

13. Z. Gao, N. Ansari, "Tracing cyber attacks from the practical perspective", *IEEE Communications Magazine*, vol. 43, no. 5, pp. 123-131, May 2005.

14. Z. Gao, N. Ansari, "A practical and robust inter-domain marking scheme for IP traceback", submitted to Computer Networks.

15. Z. Gao, N. Ansari, K. Anantharam, "A new marking scheme to defend against distributed denial of service attacks", *Proc. IEEE GLOBECOM'2004*, Dallas, TX, December 2004, vol. 4, pp. 2256-2260.

16. Z. Gao, N. Ansari, "Differentiating malicious DDoS attack traffic from normal TCP flows with proactive tests", submitted to IEEE Communications Letters.

17. W. G. Morein, A. Stavrou, D. L. Cook, A. D. Keromytis, V. Misra, D. Rubenstein,"Using graphic turing tests to counter automated DDoS attacks against web servers," *Proc. 10th ACM conference on Computer and communications security*, Washington, D.C., 2003, pp.8-19.

18. J. Mirkovic, S. Dietrich, D. Dittrich, and P. Reiher, "Internet Denial of Service attack and defense mechanisms," Prentice Hall, 2005.

19. CERT, "CRET advisory CA-1996-26 Denial-of-service attack via ping". Retrieved April 8, 2006 from the World Wide Web: http://www.cert.org/advisories/CA-1996-26.html.

20. D. J. Marchette, "Computer Intrusion Detection and Network Monitoring: A Statistical Viewpoint," Springer, 2001.

21. D. Dittrich, "DDoS: a look back from 2003". Retrieved April 8, 2006 from the World Wide Web: http://staff.washington.edu/dittrich/talks/I2-ddos.ppt.

22. D. Dittrich, "Distributed denial of service (DDoS) attacks/tools". Retrieved April 8, 2006 from the World Wide Web: http://staff.washington.edu/dittrich/misc/ddos/.

23. J. Barlow and W. Thrower, "TFN2k-An analysis". Retrieved April 8, 2006 from the World Wide Web: http://www.packetstormsecurity.org/distributed/TFN2k_Analysis.htm.

24. A. Kuzmanovic and E. knightly, "Low-Rate TCP-Targeted Denial of Service Attacks (The Shrew vs. the Mice and Elephants)," *Proc. ACM SIGCOMM 2003*, Aug. 2003, pp.75-86.

25. V. Paxson, "An analysis of using reflectors for distributed denial-of-service attacks," *ACM Computer Communications Review*, vol. 31, no. 3, pp. 38-47, 2001.

26. S. Gibson, "DRDoS: Distributed reflector denial of service," Technique Report of Gibson Research Corporation. Retrieved April 8, 2006 from the World Wide Web:www.grc.com/dos/drdos.htm.

27. P. Ferguson and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing," RFC 2267, Jan. 1998.

28. SANS institute, "Global incident analysis center:special notice". Retrieved April 8, 2006 from the World Wide Web:http://www.sans.org/y2k/egress.htm.

29. J. Mirkovic, G. Prier, and P. Reiher, "Attacking DDoS at the source," *Proc. IEEE ICNP'02*, Nov. 2002, pp. 312-321.

30. A. D. Keromytis, V. Misra and D. Rubenstein, "SOS: Secure Overlay Services," *Proc. ACM SIGCOMM 2002*, Pittsburgh, PA, Aug.2002, pp. 61-72.

31. T. Peng, C. Leckie, and K. Ramamohanarao, "protection from distributed denial of service attack using history-based filtering," *Proc. IEEE International Conf. On Communications (ICC 2003)*, Anchorage, AK, USA, May 2003, pp. 482-486.

32. K. Park and H. Lee, "On the effectiveness of route-based packet filtering for distributed DoS attack prevention in power-law Internets," *Proc. ACM SIGCOMM*, 2001, pp. 15-26.

33. K. Lakshminarayanan, D. Adkins, A. Perrig, and I. Stoica, "Taming IP Packet Flooding Attacks," *ACM Computer Communication Review*, Vol. 34, Jan. 2004, pp. 45-50.

34. T. Anderson, T. Roscoe, and D. Wetherall, "Preventing Internet Denial-of-Service with capabilities," *ACM Computer Communication Review*, Vol. 34, pp. 39-44, Jan. 2004.

35. T. Gil and M. Poletto, "Multops: a data-structure for bandwidth attack detection," *Proc. 10th USENIX Security Symposium*, Washington, DC, August 2001.

36. H. Wang, D. Zhang, and K. Shin, "Detecting SYN flooding attacks," *Proc. IEEE INFOCOM 2002*, Mar. 2002, pp. 1530-1539.

37. C. Cheng, H. Kung, and K. Tan, "Use of spectral analysis in defense against DoS attacks," *Proc. IEEE GLOBECOM 2002*, Dec. 2002, pp. 2143-2148.

38. Y. Kim, W. Lau, M. Chuah, and H. J. Chao, "PacketScore: Statistics-based Overload Control against Distributed Denial-of-Service Attacks", *Proc. IEEE INFOCOM 2004*, pp. 2594-2604.

39. C. Jin, H. Wang, and K. Shin, "Hop-count filtering: an effective defense against spoofed DDoS traffic," *Proc. ACM conf. on Computer and Communication Security (CCS'03)*, Washington, D.C., Oct. 2003, pp. 30-41.

40. M. Sung, J. Xu, "IP Traceback-based Intelligent Packet Filtering: A Novel Technique for Defending against Internet DDoS Attacks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 14, no. 9, pp. 861-872, September 2003.

41. J. Ioannidis and S. Bellovin, "Implementing Pushback: router-based defense against DDoS attacks," *Proc. Network and Distributed System Security Symposium*, San Diego, CA, USA, Feb. 2002.

42. R. Mahajan and S. Floyd, "Controlling high-bandwidth flows at the congested router," *Proc. IEEE ICNP'2001*, pp. 192-201.

43. X. Wang and M. Reiter, "Defending Against Denial-of-Service Attacks with Puzzle Auctions," *Proc. IEEE Symposium on Privacy and Security*, May 2003,Oakland, CA, pp. 78-92.

44. A. Yaar, A. Perrig and D. Song, "Pi: A Path Identification Mechanism to Defend against DDoS Attacks," *Proc. IEEE Symp. Privacy and Security*, Oakland, CA, May 2003, pp. 93-107.

45. S. Khattab, C. Sangpachatanaruk, and T. Znati, "Roaming honeypots for mitigating service-level denial-of-service attacks," *Proc. 24th International Conference on Distributed Computing Systems*, Tokyo, Japan, March 2004, pp. 328–337

46. R. Chang, "Defending against flooding-based, Distributed Denial of Service attacks: a tutorial," *IEEE Communications Magazine*, vol. 40, no. 10, pp. 42-51, 2002.

47. S. Savage, D. Wetherall, A. Karlin, and T. Anderson, "Network Support for IP Traceback," *IEEE/ACM Trans. Networking*, vol. 9, pp. 226-237, Jun. 2001.

48. A. C. Snoeren, C. Partridge, *et al.*, "Single Packet IP Traceback," *IEEE/ACM Trans. Networking*, vol. 10, pp. 721-734, Dec. 2002.

49. S. Bellovin, "ICMP Traceback Messages," *IETF Draft*, Mar. 2000. Retrieved April 8, 2005 from the World Wide Web: http://www.research.att.com/smb/papers/draft-bellovin-itrace-00.txt.

50. D. Dean, M. Franklin, and A. Stubblefield, "An Algebraic Approach to IP traceback," *ACM Tran. Information & System Security (TISSEC)*, vol.5, May 2002, pp. 119-137.

51. A. Belenky and N. Ansari, "IP Traceback with Deterministic Packet Marking," *IEEE Comm. Letters*, vol. 7, no. 4, pp. 162-164, Apr. 2003.

52. R. Stone, "CenterTrack: an IP overlay network for tracing DoS floods," *Proc. USENIX Security Symp.*, Jul. 2000, pp. 199-212.

53. J. Li, M. Sung, J. Xu, L. Li, and Q. Zhao, "Large-Scale IP Traceback in High-Speed Internet: Practical Techniques and Theoretical Foundation," *Proc. 2004 IEEE Symposium on Security and Privacy*, Oakland, California, May 2004, pp. 115-129.

54. D. Song and A. Perrig, "Advanced and Authenticated Marking Schemes for IP traceback," *Proc. IEEE INFOCOM 2001*, pp. 878-886.

55. A. Yaar, A. Perrig, D. Song, "FIT: Fast Internet Traceback," *Proc. IEEE INFOCOM 2005*, vol. 2, pp. 1395-1406.

56. T. Peng, C. Leckie, R. Kotagiri, "Adjusted Probabilistic Packet Marking for IP traceback," *Proceedings of Networking'2002*, pp. 697-708.

57. Y. Tseng, H. Chen, W. Hsieh, "Probabilistic Packet Marking with Non-Preemptive Compensation," *IEEE Communications Letters*, vol. 8, no. 6, pp. 359-361, 2004.

58. M. Goodrich, "Efficient packet marking for large-scale IP traceback," *Proc. 9th ACM conf. on computer and communications security*, 2002, pp. 117-126.

59. H. Aljifri, M. Smets, A. Pons, "IP traceback using header compression," *Computer and Security*, vol. 22, no. 2, pp.136-151, 2003.

60. H. Burch, B. Cheswick, "Tracing anonymous packets to their approximate source," *Proc. USENIX LISA Conference*, 2000, pp. 319–327.

61. K. Park, H. Lee, "On the effectiveness of Probabilistic Packet marking for IP traceback under Denial of service attack," *Proc. IEEE INFOCOM 2001*, 2001, pp. 338-347.

62. M. Waldvogel, "GOSSIB vs. IP traceback rumors," *Proc. Computer Security Applications Conference 2002*, 2002, pp. 5-13.

63. CERT, "Cisco IOS Interface Blocked by IPv4 Packet," Retrieved April 8, 2006 from the World Wide Web: http://www.cert.org/advisories/CA-2003-15.html.

64. A. Mankin, D. Massey, C. Wu, S. Wu, and L. Zhang, "On design and evaluation of 'intention-driven' ICMP traceback," *Proc. Comp. Comm. & Network*, Oct. 2001, pp. 159-165.

65. B. Wang, H. Schulzrinne, Multifunctioal ICMP messages for e-commerce, *Proc. 2004 IEEE International Conference on e-Technology, e-Commerce and e-Service*, 2004, pp. 325–332.

66. B. Wang, H. Schulzrinne, An IP traceback mechanism for reflective DoS attacks, *Proc. 2004 Canadian Conference on Electrical and Computer Engineering*, 2004, pp. 901–904.

67. R. Housley, W. Ford, W. Polk, D. Solo, "Internet X.509 Public Key Infrastructure and CRL Profile," RFC 2459, January 1999.

68. A. Belenky and N. Ansari, "Tracing Multiple Attackers with Deterministic Packet Marking (DPM)," *Proc. 2003 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM '03)*, Victoria, B.C., Canada, August 28-30, 2003, pp. 49-52.

69. F. Baker, "Requirements for IP version 4 routers," RFC1812, June 1995.

70. Y. Rekhter, T. Li, "A Border Gateway Protocol 4," RFC 1771, 1995.

71. L. Gao, J. Rexford, "Stable Internet Routing without Global Coordination," *IEEE/ACM Transactions on Networking*, vol. 9, no. 12, pp. 681-692, 2001.

72. T. Griffin, "The stable paths problem as a model of BGP routing". Retrieved April 8, 2006 from the World Wide Web: http://web.njit.edu/~ott/Griffin.ppt.

73. L.Gao, "On inferring autonomous system relationships in the Internet," *IEEE/ACM Trans. Networking*, vol. 9, pp. 733-745, Dec. 2001.

74. M. Fayed, P. Krapivsky, J. Byers, M. Crovella, D. Finkel, S. Redner, "On the size distribution of autonomous systems," technical report, Boston university, 2003.

75. National Laboratory for Applied Network Research, AS path length. Retrieved April 8, 2006 from the World Wide Web: http://moat.nlanr.net/ASPL.

76. B. Huffaker, M. Fomenkov, D. J. Plummer, D. Moore, K. Claffy, "Distance Metrics in the Internet", Retrieved April 8, 2006 from the World Wide Web: http://www.caida.org/outreach/papers/2002/Distance/distance.pdf.

77. CAIDA. Retrieved April 8, 2006 from the World Wide Web: http://www.caida.org/tools/measurement/iffinder.

78. C. Labovitz, G. Malan, and F. Jahanian, "Internet Routing Instability," *IEEE/ACM Trans. Networking*, vol. 6, no. 5, pp. 515-528, Oct. 1998.

79. V. Paxson, "End-to-End routing behavior in the Internet," *IEEE/ACM Trans. Networking*, vol. 5, no. 5, pp. 601-615, Oct. 1997.

80. V. Paruchuri, A. Durresi, R. Kannan, S. Lyengar, "Authenticated autonomous system traceback," *Proc. 18th International conference on advanced information networking and applications (AINA 2004)*, 2004, pp. 406-413.

81. A. Hussain, J. Heidemann, C. Papadopoulos, "A framework for classifying denial of service attacks," *Proc. ACM SIGCOMM'03*, 2003, pp. 99-110.

82. G. Huston, Growth of BGP routing table (94-present), http://bgp.potaroo.net/.

83. CAIDA, Skitter, Retrieved April 8, 2006 from the World Wide Web: http://www.caida.org/tools/measurement/skitter.

84. Internet Mapping Project. Retrieved April 8, 2006 from the World Wide Web: http://research.lumeta.com/ches/map.

85. T. Satty, "The four-color problem: assaults and conquest", New York, Dover Publications, 1986.

86. Galtech, Four Color Theorem. [Online]. Retrieved April 8, 2006 from the World Wide Web: http://www.math.gatech.edu/~thomas/FC/fourcolor.html.

87. L. Subramanian, S. Agarwal; J. Rexford, and R. Katz, "Characterizing the Internet hierarchy from multiple vantage points," *Proc. IEEE INFOCOM 2002*, vol. 2, Jun. 2002, pp. 618-627.

88. W. Theilmann and K. Rothermel, "Dynamic distance maps of the Internet," *Proc. IEEE INFOCOM*, vol. 1, Mar. 2000, pp. 275–284.

89. A. Garg and A. Reddy, " Mitigation of DoS attacks through QoS regulation," *Proc. IWQOS workshop*, May 2002, pp. 45-53.

90. S. Kim and A. Reddy, "Real-time detection and containment of network attacks using QoS regulation," *Proc. IEEE ICC 2005*, Seoul, Korea, vol. 1, May 2005, pp. 311-315.

91. D. Yau, J. Lui, F. Liang, and Y. Yan, "Defending against distributed Denial-of-Service attacks with max-min fair server-centric router throttles," *IEEE/ACM Transaction on Networking*, vol. 13, no. 1, pp.29-41, Feb. 2005.

92. S. Chen and Q. Song, "Perimeter-Based Defense against High Bandwidth DDoS Attacks, " *IEEE Transactions on Parallel and Distributed Systems*, vol. 16, no. 6, pp. 526-537, June 2005.

93. J. Xu and W. Lee, "Sustaining Availability of Web Services under Distributed Denial of Service Attacks," *IEEE Transaction on Computers*, special issue on Reliable Distributed Systems. vol. 52, no 2, pp. 195-208, Feb 2003.

94. M. Fomenkov, K. Keys, D. Moore, and K. Claffy, " Longitudinal study of Internet traffic from 1998-2003," *Proc. Winter International Symposium on Information and Communication Technologies*, Cancun, Mexico, January 2004, pp. 1-6.

95. Claffy, k, Miller, G. and Thompson K. "The nature of the beast: recent traffic measurements from an Internet backbone," *Proceedings of INET'98* (ISOC), Geneva, Switzerland, July 1998.

96. J. Lemon, "Resisting SYN flooding DoS attacks with a SYN cache," *Proc. USENIX BSDCon Conf.*, Feb. 2002, pp. 89-98.

97. S. Ebrahimi-Taghizadeh, A. Helmy, S. Gupta, "TCP vs. TCP: a Systematic Study of Adverse Impact of Short-lived TCP Flows on Long-lived TCP Flows," *Proc. IEEE INFOCOM 2005*, Miami, FL, March 2005, pp. 926-937.