Defeating Distributed Denial of Service Attacks with Source Router Preferential Dropping

by

Yinghong Fan

A thesis submitted to the

School of Computing

in conformity with the requirements

for the degree of the Master of Science

Queen's University

Kingston, Ontario, Canada

June 2003

National Library of Canada

Acquisitions and Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale du Canada

Acquisitions et services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

0-612-81063-1

Canada

Abstract

A Distributed Denial of Service (DDoS) attack is an explicit attempt to interrupt an online service by generating a high volume of malicious traffic. These attacks consume all available network resources, thus rendering legitimate users unable to access the services. Most existing solutions propose to detect and drop attack packets at or near the destination network where the attack packets have already traversed the network and consumed considerable bandwidth. The aggregate traffic at the destination router may consist of hundreds of thousands of flows. It is hard for the router to distinguish between legitimate and malicious packets. So, collateral damage is unavoidable.

In this thesis, we present a Source Router Preferential Dropping (SRPD) scheme to detect possible DDoS attacks and defeat them at their sources. SRPD monitors only high-rate outgoing flows at source networks and preferentially drops the packets belonging to these flows when it senses an existence of an attack.

A simulation model is constructed and a number of simulation experiments have been conducted to evaluate the performance of the proposed scheme. The results show that SRPD effectively controls DDoS attacks at their sources and reduces collateral damage to a minimum level.

Acknowledgements

I would like to express my gratitude to my supervisors Professor Patrick Martin and Professor Hossam Hassanein, for their excellent guidance, great advice and support.

I would like to thank all other members in the Telecommunications Research Laboratory and the Database System Laboratory at Queen's University for their suggestions, assistance, comments and friendship.

Special thanks for my husband Haicheng, for his selfless support and great advice. Without his love and support, I cannot make it.

Finally, I would like to thank the School of Computing at Queen's University for offering me such an invaluable opportunity to pursue my Master's Degree. The financial support provided by Communications and Information Technology Ontario (CITO) and Kingston Software Factory is appreciated.

Table of Contents

Chapter 1 Introduction 1

Chapter 2 Related Work 6

2.1 Preventive Mechanisms 10
2.2.1 Security Configuration Mechanisms 10
2.2.2 Firewall Approaches 12
2.2 Traceback Mechanisms 14
2.3 Reactive Mechanisms 16
2.4 Summary 18

Chapter 3 Source Router Preferential Dropping Scheme **20**

3.1 Overview of SRPD 21
3.2 Identifying High-Rate Flows 25
3.3 Preferential Dropping at Source Routers 27
3.3.1 Computing the Drop Probability 30
3.3.2 Queue Length Enquiry and Response Messages 33
3.3.3 Flow Information Table 35
3.3.4 Flow States 36
3.4 Summary 39

Chapter 4 Performance Evaluation **40**

4.1 Simulation Model 41
4.1.1 The SRPD Model 41
4.1.2 The Pushback Model 43
4.2 Simulation Scenarios 45
4.2.1 Victim Router is Congested 46
4.2.2 Victim Server is Overloaded 46
4.3 Simulation Parameter Setting 47
4.4 Evaluation Criteria 48
4.4.1 Attack Packet Drop Rate 48
4.4.2 Legitimate Packet Drop Rate 49
4.4.3 Average Response Time 49
4.4.4 Queue Length at Victim Router 49
4.4.5 Victim Server Workload 50

4.5 Simulation Software Implementation 50
4.6 Simulation Results 52
4.6.1 Simulations of Congesting the Victim Router 52
4.6.1.1 Comparing SRPD with Pushback 52
4.6.1.2 Effect of the High-rate Identifying Threshold 58
4.6.1.3 Effect of the Minimum Response Time Threshold 61
4.6.1.4 Effect of the Maximum Response Time Threshold 63
4.6.2 Simulations of Flooding the Victim Server 70
4.7 Summary 72

Chapter 5 Conclusion 75

5.1 Concluding Remarks 75
5.2 Discussion and Future Work 77

Bibliography 80

Appendix A 85

Appendix B 90

Vita 92

List of Figures

Figure 1.1 Distributed Denial of Service Attack Architecture 2

Figure 2.1 Smurf Attacks 8

Figure 2.2 Attack Scenario of Synkill Firewall 13

Figure 2.3 Normal Access Scenario of Synkill Firewall 13

Figure 3.1 Denial of Service Attack System 22

Figure 3.2 The flowchart of drop decision at a SRPD router 27

Figure 3.3 Average Response Time Calculation 30

Figure 3.4 Packet Drop Probability vs. Average Response Time 32

Figure 3.5 ICMP Message Format 33

Figure 3.6 Queue Length Enquiry and Response Message 34

Figure 3.7 Flow Information Table 35

Figure 3.8 Flow State Diagram 38

Figure 4.1 SRPD Simulation Architecture 41

Figure 4.2 A DDoS attack in Pushback Topology 44

Figure 4.3 Legitimate Packet Drop Rate (Compare with Pushback) 57

Figure 4.4 Attack Packet Drop Rate (Compare with Pushback) 57

Figure 4.5 Percentage of Queue Length Occupancy at the Victim Router (Compare with Pushback) 58

Figure 4.6 Legitimate Packet Drop Rate with Different Detection Rate 60

Figure 4.7 Attack Packet Drop Rate with Different Detection Rate 60

Figure 4.8 Good Packet Drop Rate at $Max_{response}$=500ms 63

Figure 4.9 Attack Packet Drop Rate at $Max_{response}$=500ms 63

Figure 4.10 Legitimate Packet Drop Rate with Lower $Max_{response}$ 65

Figure 4.11 Legitimate Packet Drop Rate with Higher $Max_{response}$ 65

Figure 4.12 Attack Packet Drop Rate with Various $Max_{response}$ 66

Figure 4.13 Average Response Time with Lower $Max_{response}$ 67

Figure 4.14 Average Response Time with Higher $Max_{response}$.....67
Figure 4.15 Legitimate Packet Drop Rate at $Min_{response}$=600ms.....68
Figure 4.16 Attack Packet Drop Rate at $Min_{response}$=600ms.....68
Figure 4.17 Legitimate Packet Drop Rate.....72
Figure 4.18 Attack Packet Drop Rate.....72
Figure 4.19 Victim Server Workload.....72
Figure 4.20 Average Response Time.....72

List of Tables

Table 3.1 SRPD Parameters.....24
Table 3.2 Variables used by SRPD.....25
Table 4.1 Parameters For Links.....47
Table 4.2 Parameters For Source Routers.....47
Table 4.3 Parameters For Victim Routers.....48
Table 4.4 Parameters For Victim Servers.....48

List of Acronyms

DoS	Denial of Service
DDoS	Distributed Denial of Service
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
ICMP	Internet Control Message Protocol
HTTP	Hypertext Transfer Protocol
DNS	Domain Name Service
SMTP	Simple Mail Transfer Protocol
FTP	File Transfer Protocol
RTT	Round Trip Time
RPC	Remote Procedure Call
IP	Internet Protocol
MULTOPS	Multilevel Tree for Online Packet Statistics
FDS	Flooding Detection System
SRPD	Source Router Preferential Dropping
ACC	Aggregate Congestion Control
FIFO	First In First Out
VLAN	Virtual Local Area Network

Chapter 1

Introduction

With the development of the Internet, more and more machines try to connect with the global Internet. Meanwhile, hacker communities advance their intrusion techniques and tools in order to make their invasions more difficult to detect and prevent. Unfortunately, a large amount of machines are still vulnerable. Many systems are not constantly updated to increase their security. Moreover, the current Internet infrastructure does not have powerful built-in protection mechanisms to prevent deliberate attacks. Therefore, network security problem has received more consideration in recent years.

Distributed Denial of Service (DDoS) attacks [1] have posed an immense threat to the Internet in recent years. A distributed denial of service attack is an explicit attempt to interrupt an online service by generating a high volume of malicious traffic [1] [2]. These attacks consume all available network or machines' resources, thus rendering legitimate users unable to access the services. The Internet community started to pay more attention to DDoS attacks after some famous web sites, such as Yahoo, Amazon.com and CNN.com, went under serious attack in February 2001. In today's Internet, DDoS attacks are widespread [3]. Although denial of service attacks have

existed for years, their evolving features make it a serious challenge to come up with a totally effective solution.

Currently, there are four popular known DDoS attack tools: Trinoo [4], Tribe Flood networks (TFN) [5], TFN2K and Stacheldraht [6]. They share a similar architecture and pursue a similar procedure. Figure 1.1 illustrates a distributed denial of service attack structure that is used by these tools.

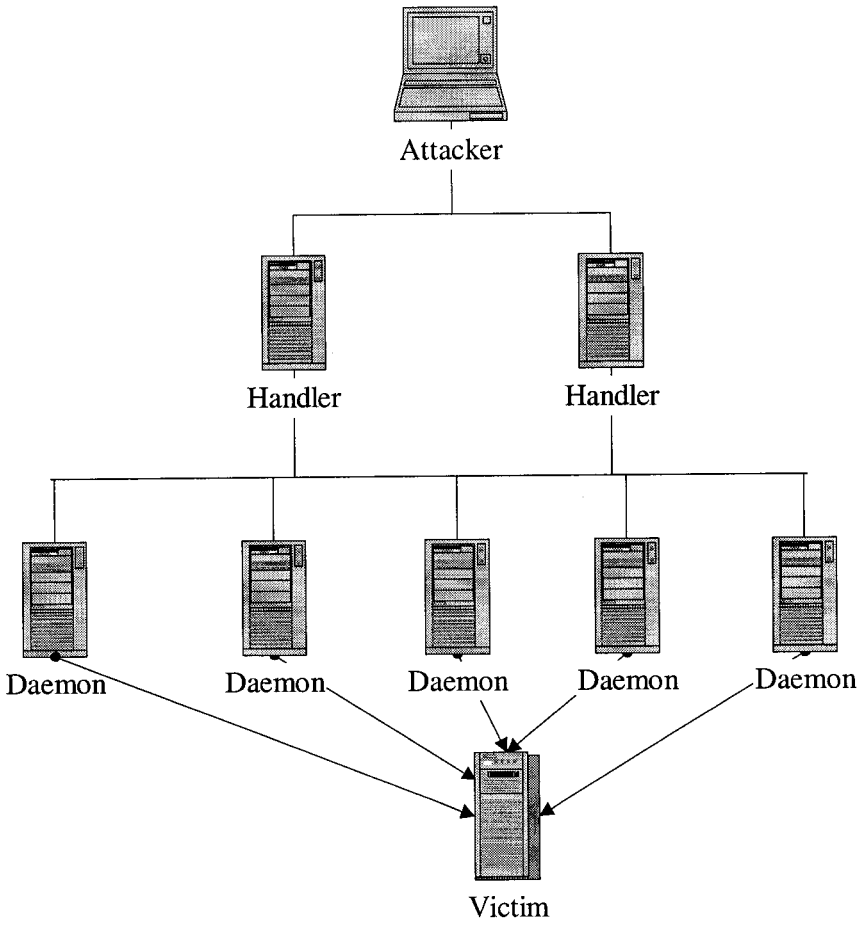

Figure 1.1 Distributed Denial of Service Attack Architecture

To conduct a DDoS attack, an attacker uses stolen accounts or takes advantage of some well-known UNIX system security bugs, such as buffer overrun bugs in the RPC services, to get root authorization so that it can install the so-called daemons on many hosts. These daemon programs are controlled by a small number of handlers, each of which hides somewhere else and knows a list of compromised hosts. The daemons wait for the command from their handlers while the real attacker directs handlers. As a result, the attacker can set up a DDoS attack network in a short time without being noticed by the owners of the compromised hosts.

DDoS tools can carry out different kind of attacks, such as UDP flood, TCP SYN flood, or ICMP flood. They may generate attack packets with randomized source IP addresses, which makes it very hard to find the sources of the real attack traffic. The target of these attacks is to consume link bandwidth or server resources, disabling legitimate users to access the services. Some other attack tools have been found as well. Some of them are just modified version from the above four tools. For example, "Shaft" [31] is modeled after Trinoo.

One of the crucial reasons why distributed denial of service attacks have opportunities to be successful is that Internet security is highly interdependent [7]. DDoS attacks are commonly launched from systems that are subverted through security-related compromises. Regardless of how well secured a victim system may be, its susceptibility to DDoS attacks depends on the state of security in the rest of the global Internet. A securely configured system can certainly decrease the frequency and

strength of DDoS attacks however good configuration cannot be one hundred percent effective. A well-installed machine with firewall systems, like those famous websites mentioned above, can still be a victim of DDoS attacks. A safe environment needs the cooperation of the global Internet.

Most existing solutions propose to detect and drop attack packets at or near the destination network where the attack packets have already traversed the network and consumed considerable bandwidth. The aggregate traffic at the destination router may consist of hundreds of thousands of flows. It is impossible to distinguish between legitimate and malicious packets from this massive amount of traffic. Consequently, collateral damage is unavoidable. Till now, no scheme has been designed to detect and control DDoS attacks at their source networks.

In this thesis, we present a Source Router Preferential Dropping (SRPD) scheme to detect possible attacks and defeat them at their sources so that most of the attack packets can be dropped at the client networks instead of the victim network. This scheme makes the identification between good and bad packets easier, thus minimizing the amount of collateral damage. It also helps to reduce bandwidth usage by the attack traffic. SRPD monitors only high-rate outgoing flows at source networks and preferentially drops the packets belonging to these flows when it senses the existence of an attack. SRPD is a practical scheme. There is no need to modify any existing Internet protocols. It also has the advantage that no core routers need to be involved.

The rest of the thesis is organized as follows. In Chapter 2 we review existing approaches for defeating DDoS. We categorize the approaches according to their design objectives and goals. Basically, we have three classes: preventive mechanisms, traceback mechanisms and reactive mechanisms. Chapter 3 provides a detailed description of our proposed SRPD scheme. Chapter 4 describes our simulation model and presents simulation results. The effectiveness of SRPD is compared with that of another well-known scheme and then SRPD's performance for different parameter settings is studied. Finally, Chapter 5 draws our conclusions and discusses future work.

Chapter 2

Related Work

DDoS attacks can be classified based on the protocol they use. Basically, attackers take advantage of TCP [36], UDP [37] and ICMP [32] protocols to manipulate different types of DDoS attacks. Applications based on the TCP protocol require the two end systems to establish a connection before the real data packets are transmitted. The well-known TCP based protocols are HyperText Transfer Protocol (HTTP), Simple Mail Transfer Protocol (SMTP) and File Transfer Protocol (FTP), etc. TCP protocols have built-in mechanisms to deal with congestion. For instance, a sender may reduce its sending rate by minimizing the congestion window when it senses the existence of congestion and then exponentially increases the congestion window if no further congestion is detected. The congestion window continues to grow up until the receiver's granted window size (credit window size) is reached. This procedure is called Slow Start, which is widely used by TCP protocol.

UDP and ICMP are connectionless protocols, which means that there is no need to set up a connection before sending data packets. Both UDP and ICMP protocols have no congestion detection and control mechanisms. Domain Name Service (DNS) and

many other applications are based on UDP protocols to communicate. An example of an ICMP-based protocol is the widely used "PING".

To conduct a TCP attack, an attacker can deploy either TCP SYN flood or HTTP request flood. Any system that is connected to the Internet and provides TCP-based network services, such as a Web server or email server, is potentially a victim to this kind of attack. UDP and ICMP floods are called "Brute force" attacks in that they require a vast amount of traffic to exhaust the victim system's network or server resources.

In a TCP SYN attack [8], the attacker exploits the vulnerability of the TCP protocol's three-way handshake to establish a TCP connection. The potential abuse arises at the point where the server system has sent an acknowledgment (SYN-ACK) back to a client but has not yet received the ACK message. This is the so-called *half-open connection*. When the attackers continue issuing multiple requests using bogus source IP addresses in the IP header of the attack packets, the server's buffer and CPU will be consumed and eventually filled up by these half-open connections that are never completed. An HTTP request flood is used to inflict damage at a Web server. HTTP flood packets must belong to an established TCP connection. Therefore, they cannot spoof their source IP addresses. In this case, they are more like a "Brute force" attack and need to generate a much higher volume of HTTP request traffic than TCP SYN floods in order to overload the victim.

To perform an ICMP or UDP flood attack, the attacker needs a huge amount of packets to overwhelm the victim network or server. An example of an ICMP attack is the "Smurf" attack [9]. In a smurf attack the attackers do not have to generate so much attack traffic by themselves. Instead, they send forged ICMP echo request packets directed to IP broadcast addresses and set a victim's IP address as the source IP address. For example, the IP broadcast address for the network 64.1.0.0 is 64.1.255.255. Therefore, large amounts of ICMP echo reply packets are sent from the intermediary sites to the victim, causing network congestion or server overload at the victim network. The following figure 2.1 shows how Smurf attack works.

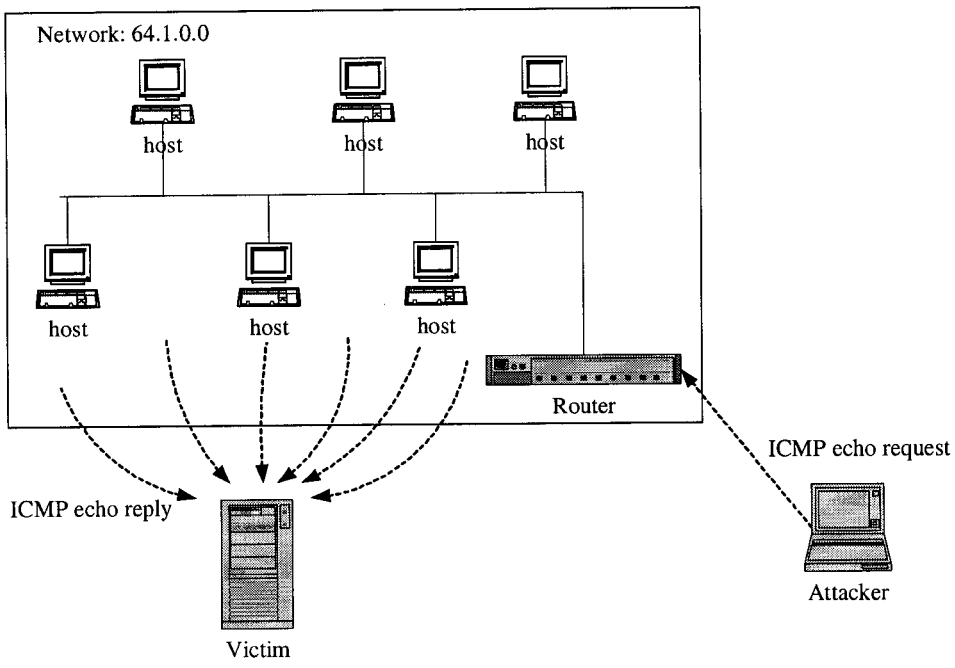

Figure 2.1 Smurf Attacks

There are three parties in these attacks: the attacker, the intermediary, and the victim. The spoofed IP address represents the victim of the attack. Of course, the

intermediary can also be a victim. In the above example, the network 64.1.0.0 is the intermediary network.

Another similar UDP attack uses a DNS request flood [10]. In this kind of attack, name server responses can be significantly larger than DNS requests so there is potential for bandwidth amplification. In both the Smurf and the DNS request attack, the owners of the machines in the intermediate network do not even notice that they send attack packets to the victim site. The true source of the attack is difficult for an intermediate or a victim site to determine due to the use of bogus source addresses. Another reason why UDP and ICMP flood are hard to control is that many firewalls are not able to block ICMP or UDP packets. They can go right through without any obstacles.

Defending against distributed denial of service attacks is usually treated as a congestion control problem. However, traditional end-to-end congestion control schemes are not directly applicable to DDoS attacks. In a normal situation, a sender senses an existence of congestion by detecting packet dropping. It then reduces its sending rate and retransmits the lost packets at some time later. In the case of DDoS attacks, the sender maintains its sending rate for the whole period of congestion. It may even increase the sending rate when it needs more traffic to flood the victim network. Therefore, malicious traffic consumes more bandwidth, causing legitimate traffic to back off and eventually starve. Traditional congestion control mechanisms

are not able to relieve the situation if the congestion results from an attack. Specially designed mechanisms to handle different kind of attacks are therefore required.

In this chapter, we provide a technical review of various approaches to DDoS attack detection and defense. According to the goal of DDoS defense mechanisms, we differentiate them into three classes: preventive mechanisms, traceback mechanisms and reactive mechanisms. Section 2.1 presents some preventive methods. Section 2.2 describes attack sources traceback mechanisms. Section 2.3 discusses some recently proposed reactive schemes. A brief summary is given in Section 2.4.

2.1 Preventive Mechanisms

Unfortunately, there is no simple solution that offers perfect protection against the different types of attacks. There are, however, a number of preventive steps that can help to minimize the impact of attacks.

2.2.1 Security Configuration Mechanisms

A securely configured system can certainly decrease the power of DDoS attacks. Egress and ingress filtering are techniques performed by routers to effectively eliminate IP spoofing. With ingress filtering [11] and egress filtering [38], edge routers match the IP source address of each inbound and outbound packet against a fixed set of known IP address prefixes. If no match is found, which means that the source IP addresses are unauthorized, the packets are discarded. In this way, attack

packets with bogus source IP addresses cannot reach their destination and are stopped at intermediate routers.

Another approach to block spoofed attack traffic is Unicast Reverse Path Forwarding [12]. When unicast reverse path forwarding feature is enabled on an interface, the router examines all packets received as input on that interface to make sure that the source address and source interface appear in the routing table and match the interface on which the packet was received. If there is no reverse path route on the same interface from which the packet was received, it means that the source address was modified or forged. The packet is then discarded.

These filtering and forwarding mechanisms can be used to decrease the impact of denial of service attacks that use spoofed IP source addresses. However, in some DDoS attacks, attack packets from daemons may not spoof their address. In that case, using this type of techniques is not able to efficiently reduce the damage.

Some other configurations can be deployed at routers or other network devices to improve the system security and decrease the possibility of being attacked or compromised by hackers. For example, IP-directed broadcast functionality is generally not needed. Disabling IP-directed broadcasts at routers can deny IP broadcast traffic into their networks from other networks [9].

There are several ways to increase the security of UNIX systems. Examples include monitoring access to the machine and checking the system for malicious DDoS daemons [19], closing unused TCP/UDP ports, turning off unnecessary services such as RPC services "statd", "cmsd" and "ttdbserverd", and installing security patches frequently.

2.2.2 Firewall Approaches

Statistics and analysis show that more than 50% of attack traffic comes from TCP attacks, especially from TCP SYN flood [3]. Therefore, mechanisms have been designed to specially cope with TCP SYN flooding attacks.

A number of firewalls are suggested to counter TCP SYN flooding attacks, such as Synkill [14], Syn cache [15], Syn cookies [16], SynDefender [17], and Syn proxying [18]. Firewall-based protection approaches first examine every packet destined to a host inside the firewall, and thus decisions can be made on its authenticity and actions can be taken to protect the internal hosts.

A Synkill firewall [14], for example, is treated as a semi-transparent gateway by internal hosts. The firewall monitors the traffic and lets SYN and SYN ACK packets pass. It completes a half-open connection by generating an ACK message to the internal host, thus closing the three-way handshake. In case of a SYN flood, even though the host does not receive any ACKs from the client, upon receiving the ACK injected by the firewall, it moves the connection out of the backlog queue. The

resources that were allocated for the half-open connections are then freed at the victim host. After the timeout period, the connection is closed. In the case of a legitimate connection, the firewall passes the ACK packet from the client to the hosts. The host receives two ACK packets: one is generated by the firewall; the other is from the client. The duplicate ACK packet is discarded since TCP is designed to handle duplicate packets. Figure 2.2 and 2.3 illustrate how a Synkill firewall deals with both attack packets and legitimate packets.

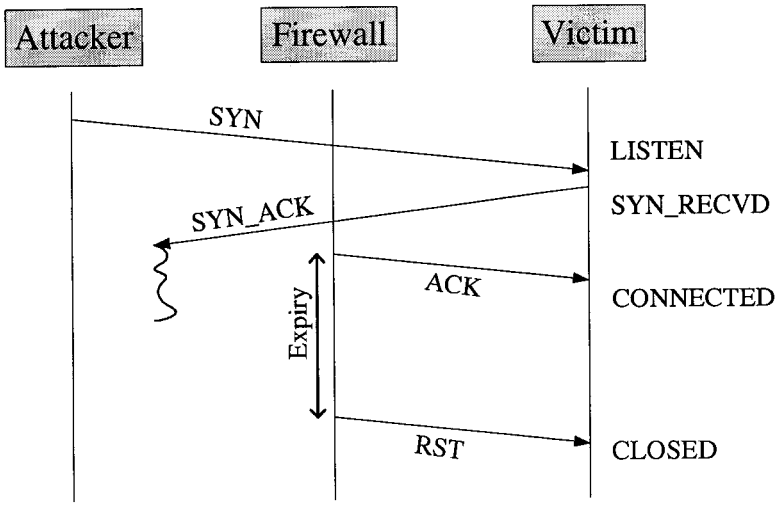

Figure 2.2 Attack Scenario of Synkill Firewall

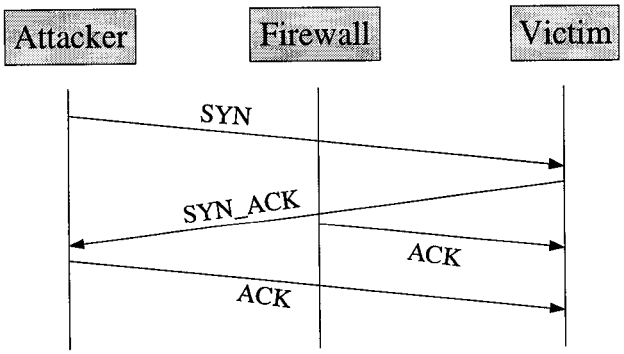

Figure 2.3 Normal Access Scenario of Synkill Firewall

These firewall approaches can help to protect the inside hosts from SYN floods. However, they cannot stop the flood and can themselves be targets of an attack. Some firewalls introduce extra delays for legitimate connections due to additional processing. Another drawback of the firewall approach is that the firewall could be a bottleneck and lead to service degradation in times of a severe attack. Moreover, most firewalls usually lack the ability to block ICMP and UDP floods. These protocols might use high-numbered ports to communicate and so can go right through the firewalls without any obstacles.

2.2 Traceback Mechanisms

Since the source IP addresses of the attack packets are not trusted, some approaches concentrate on tracking down attack sources. There are numerous traceback mechanisms [20] [21] [22] [23] [24] [25] that provide the victim with information about the identity of the daemon machines. A new ICMP Traceback Message is proposed by Bellovin [21]. This kind of message is emitted randomly by the routers along the path with low probability (e.g. 1/20,000) and forwarded to the destination. The traffic path can be determined as long as enough ICMP traceback messages are generated from intermediate routers and received by the destination. However, traceback messages consume bandwidth, especially in time of congestion when bandwidth is more important for other legitimate traffic. Another drawback of this scheme is that a victim may not have enough traceback messages in that those messages might be dropped at the congested router during the period of attack.

A better traceback technique is proposed by Savage, et al [20]. The solution is to probabilistically mark packets by intermediate routers along the path and sent to the destination. This approach exploits the observation that attacks generally comprise a large number of packets. The destination reconstructs the entire path by combining a modest number of such packets since each packet represents a sample of the path that it has traversed. In this scheme, a router calculates a hash of its IP address using a well-known hash function. Its adjacent router uses XOR to encode the two addresses. The resulting value represents the edge between these two adjacent routers. The victim extracts the hash portion and the address portion and then recalculates the hash over the address portion using the same hash function used by the routers. In this way, the victim can construct the path by combining all the edges at different distance. This approach has an advantage in that it does not generate extra traffic.

Some other approaches provide different methods to trace attack packets to their sources [22-25]. In such schemes, a network infrastructure is proposed to track down attack packets. These traceback mechanisms assist in identifying the true sources of the attack traffic and can be combined with other reactive approaches to alleviate the influence of the attacks. However, all these traceback mechanisms share a common weakness that they are not able to stop the attacks. Moreover, all the traceback approaches have serious deployment and operational challenges. A large number of routers need to be involved to support traceback otherwise the victim of the attack might not obtain enough information to reconstruct the path. In addition, if an attack is widely distributed, gathering enough traceback information is a big challenge.

Attackers may also generate traceback messages so some form of authentication of traceback messages is necessary.

2.3 Reactive Mechanisms

Since a large number of network devices in the Internet are not securely configured and not well protected, reactive approaches are more desirable. Reactive mechanisms strive not only to detect the attack but also to respond to it. A prompt automatic anomaly traffic detection strategy is necessary to pinpoint the presence of a flood. Some forms of rate limiting or traffic shaping mechanisms are also required to alleviate the impact on the victim.

Unfortunately, few of these automatic detection and response mechanisms have been proposed. Researchers are still looking for efficient ways to detect and defend against DDoS attacks. The Flooding Detection System (FDS) is a DDoS attack detection mechanism [13]. Its key feature is that it utilizes the inherent TCP SYN-FIN pairs' behavior for SYN flooding detection. The beginning of a SYN and the end of a FIN determines a TCP connection. FDS's algorithm is based on the theory that under normal conditions, one appearance of a TCP SYN packet results in the eventual return of a FIN packet. Therefore, the difference between the number of SYNs and FINs should be very small. A significant difference implies that there is a SYN flood. FDS, however, only helps to detect DDoS attacks and has no response scheme to control them.

Multilevel Tree for Online Packet Statistics (MULTOPS) [28] provides a data structure for DDoS attack detection. This scheme assumes that, during normal operations on the Internet, the packet rate of traffic going in one direction is proportional to the rate going in the opposite direction. MULTOPS monitors traffic characteristics and detects ongoing flood attacks by the significant, disproportional difference between packet rates going to (or coming from) the victim (or the attacker). This scheme maintains a tree of nodes that contains packet rate statistics for subnet prefixes at different aggregation levels. The disproportional traffic that comes from certain aggregation levels is treated as suspicious and is probabilistically dropped when necessary. However, the proportional assumption does not hold in every situation. For example, applications based on TCP protocols display proportional behavior while UDP and ICMP protocols do not require acknowledgements. Therefore, UDP and ICMP traffic is inherently disproportionate. In the latter case, MULTOPS cannot decide what is attack traffic and what is non-attack traffic. Furthermore, MULTOPS is invalid if the attack uses bogus source IP addresses.

A pushback mechanism [29] consists of a local Aggregate Congestion Control (ACC) mechanism for detecting and controlling aggregate traffic at a single congested router, and a cooperative pushback mechanism in which a router can ask its upstream adjacent routers to limit the rate of aggregate traffic to the victim. This scheme uses a local aggregate congestion-detection mechanism to identify the aggregate traffic that is responsible for causing serious congestion and to impose a rate limit on the traffic.

When the local mechanism is not powerful enough to alleviate the congestion and control the attack, it depends on the cooperative mechanism to send pushback messages to the upstream routers asking them to limit the rate of the malicious aggregate traffic. However, the local aggregate congestion control mechanism is not able to distinguish between the good and the bad traffic contained in the aggregate traffic heading for the victim and inflicts collateral damage by punishing both good and bad traffic equally.

Another rate throttle mechanism is proposed by Yau, et al [30]. In this scheme, the attack traffic destined for the victim server is throttled to a more moderate level at an intermediate router that is K hops away from the victim. These intermediate routers proactively regulate the aggressive packets to prevent them from overwhelming a server, thus forestalling an impending attack.

Although these reactive approaches can be effective in that they can somewhat relieve the congestion brought by the attacks, they have a disadvantage that their collateral damage is unavoidable, which may still make a DDoS attack successful while too many legitimate packets are dropped. As well, they require active involvement of core routers in the network.

2.4 Summary

We classified different techniques according to the goal of their strategy. Preventive mechanisms reduce or eliminate the possibility of DDoS attacks. Traceback

techniques identify the true sources of the attacks. Reactive mechanisms detect ongoing attacks and then control them. Reactive mechanisms are usually composed of an anomaly detection mechanism and a rate limiting mechanism. Anomaly detection schemes detect anomalies based on normal system behaviors. This kind of mechanism is usually parameter sensitive so setting appropriate thresholds is a key issue that needs to be addressed. In the rate-limiting schemes, how to minimize the collateral damage is the crucial issue. The influence on legitimate traffic is the main metric used to judge whether a rate-limiting mechanism is effective or not.

Chapter 3

Source Router Preferential Dropping Scheme

Defending distributed denial of service attacks is widely considered to be a congestion control problem. However, traditional end-to-end congestion control mechanisms cannot be directly extended to handle DDoS attacks. This is because they are not able to distinguish between good and bad packets, thus punishing both equally. In a normal situation, a sender senses the existence of congestion by detecting a packet drop. It then reduces its sending rate. In case of DDoS attacks, malicious traffic consists of many irresponsible flows that do not obey the rules of traditional end-to-end congestion control. They do not reduce their sending rate or may even increase the rate in times of congestion. Therefore, attack traffic consumes more bandwidth, causing legitimate traffic to back off and eventually starve. Furthermore, congestion control schemes, and most other existing solutions, try to detect and drop packets at or near the destination network where the attack packets have already traversed the network and consumed considerable bandwidth. The aggregate traffic at the destination router may consist of hundreds of thousands of flows. It is more difficult to distinguish between legitimate and attacking flows there. Therefore, some forms of congestion control mechanisms that can identify and handle DDoS attacks at their sources are required.

In this chapter, we present the Source Router Preferential Dropping Scheme (SRPD), which detects and defeats DDoS attacks at source networks. SRPD has two basic procedures: identifying the flows with high sending rate and controlling those flows' sending rate in the case of DDoS attacks. The goal of our work is to detect and drop most of the malicious packets at the edge routers close to the attacking sources instead of at the victim network. This ensures that the victim router is not seriously congested at the time of an attack and allows a minimal level of collateral damage to legitimate traffic. Normal requests are therefore able to reach their destination server even though it is under attack. Section 3.1 gives an overview of the SRPD scheme. Section 3.2 provides a detailed description of the SRPD scheme. Section 3.3 summaries this chapter.

3.1 Overview of SRPD

SRPD includes a pre-processing component at source routers: Egress Filtering. Before source routers process and forward each arrival packet, they check the source IP address of the packet for authenticity. If its source IP address comes from an unauthorized IP address, the packet is discarded. Source routers in SRPD trust the source IP addresses of all packets that have survived from this pre-processing procedure. This is very useful in SRPD since its algorithm is at the flow level, which represents both the source and the destination IP addresses.

In SRPD, the edge router at the source network computes the sending rate of all outgoing traffic. When the router detects the existence of a DDoS attack with highly

probability, it controls the rate of the flows with high sending rates by preferentially dropping the packets from these flows. A *flow* is defined as a set of packets with the same source IP, destination IP and protocol. In the case of a distributed denial of service attack either the edge router at the victim network or the victim machine itself is overloaded. This results in packet dropping at the victim network as well as much longer response times than in normal traffic conditions. By measuring packets' response times, the edge router at the source network knows whether there is congestion at the destination network. We assume that, for congestion along the path instead of the destination network, dynamic routing protocols usually load balance incoming traffic into different routes in order to relieve the congestion. Therefore, long response times that result from normal congestion can go back to a normal level in a relatively short period of time. Long response times that result from DDoS attacks, however, are kept abnormally long for the whole duration of an attack if there is no protection scheme. In our scheme, the edge router at the source network detects the ongoing attack behavior based on comparisons to normal traffic patterns. High packet sending rates associated with persistent long response times imply a possible DDoS attack.

Figure 3.1 shows the denial of service attack structure. We call the edge router at the source network the *Source Router* and the edge router at the destination network the *Victim Router*. We also name the destination host in the destination network the *Victim Host*. An attacking host generates malicious (bad) traffic, which consists of a high volume of attacking packets heading for the Victim Host. Legitimate clients

generate good traffic, which is generally at a relatively lower rate. Legitimate users may send non-attack packets to the Victim Host.

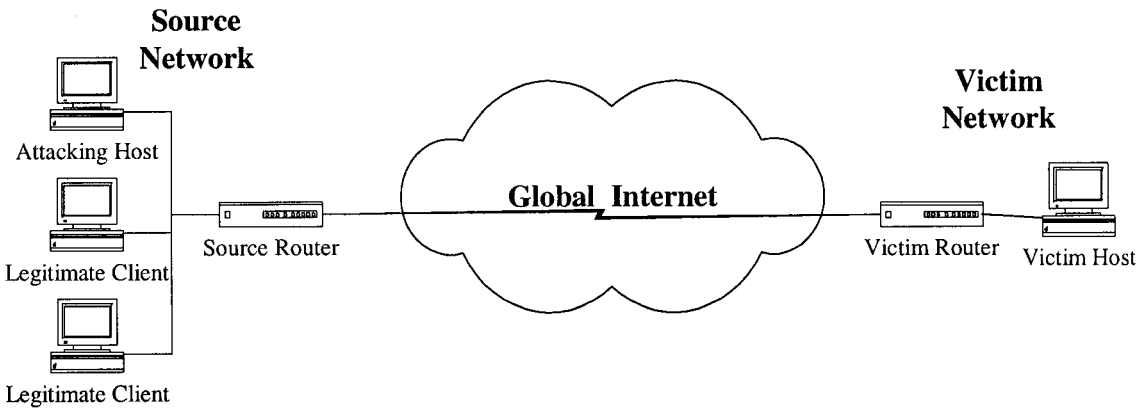

Figure 3.1 Denial of Service Attack System

In our protection scheme, a control message is sent by a Source Router to the Victim Router to collect buffer usage information on the output queue connected to the Victim Host. A long queue length at the Victim Router is an indication of congestion, or a DDoS attack. Source Routers drop packets based on a flow's arrival rate, flow status, response time and output queue length at the victim router. Under our scheme, most of the malicious packets are dropped at the source routers instead of at the victim network. This then ensures that the victim network is not seriously congested during an attack. Therefore, legitimate requests will still be able to have access to their destination.

Tables 3.1 and 3.2 summarize all the parameters and variables used in the discussion in this chapter.

$Threshold_{identify}$	The threshold for identifying those high packet arrival rate flows at source routers
$Min_{response}$	Minimum threshold for average response time $T_{response}$
$Max_{response}$	Maximum threshold for average response time $T_{response}$
$T_{refresh}$	Time period for reviewing the average response time of attack flows before relieve them from monitored flows
w	Weight used to compute average arrival rate, response time and output queue length at the victim router
β	Increasing factor for packet dropping probability
Max_{prob}	Maximum packet drop probability (maximum value for Prob)
Max_{queue}	Maximum threshold for output queue length occupancy rate $R_{occupancy}$
Min_{queue}	Minimum threshold for output queue length occupancy rate $R_{occupancy}$
B_{queue}	The output queue size at a victim router
$T_{icmp\_timeout}$	An ICMP message timeout period
$Latency_{sr\_vr}$	Half of the round trip time between a source router and a victim router
$Trans_{vr}$	Transmission time at a victim router. It is calculated by a packet size divided by the bandwidth of the output link.
$Trans_{vs}$	Transmission time at a victim server. It is calculated by dividing the packet size by the bandwidth of the output link
PT_{vr}	Average packet processing time at the victim router
ST_{vs}	Average packet service time at the victim server
QT_{vr}	Queuing delay at the victim router. It is determined by how many packets are in the queue awaiting processing

Table 3.1 SRPD Parameters

$R_{Arrival}$	Average packet arrival rate from a single flow estimated by a source router
$newR_{Arrival}$	Current average packet arrival rate in the latest one second from a single flow estimated by a source router
S_{flow}	The state of a flow
$T_{response}$	Average response time of each traffic flow
$newT_{response}$	Current response time of each traffic flow
Prob	Packet drop probability
$R_{occupancy}$	Relative occupancy of output queue at victim router
L_{queue}	Average output queue length at victim router
$newL_{queue}$	Current output queue length at victim router

Table 3.2 Variables used by SRPD

3.2 Identifying High-Rate Flows

A source router in the SRPD scheme keeps measuring each incoming flow's arrival rate. The average arrival rate at time t is computed as follows:

$$R_{Arrival}(t) := (1-w) * R_{Arrival}(t-1) + w * newR_{Arrival} \quad (0<w<1)$$

Packet arrival rate is computed with a moving time window of 30 seconds, which means we are only concerned with the average arrival rate in the last 30 seconds. Since most DDoS attacks last less than 10 minutes [3], 30 seconds is long enough for a referenced history of each flow. Although source routers update each flow's arrival rate, they only monitor those flows whose arrival rate exceeds a threshold. SRPD maintains different states for each high rate flow and updates them according to the flow's average response times and the output queue occupancy rate at the victim

router. Whenever the source router detects a likely DDoS attack, packets are probabilistically dropped from the high rate flows.

Choosing a proper value for the identification threshold, $Threshold_{identify}$, is a crucial issue in SRPD. Small values of $Threshold_{identify}$ result in monitoring more flows than necessary at source routers. In the worst case, more legitimate packets could be identified as bad packets and dropped unexpectedly. However, as long as there is no DDoS attack or no congestion at the destination network, monitoring more flows does not have adverse effects because there will not be any further action on those monitored flows. Large values of $Threshold_{identify}$ may result in failure to identify attacking flows.

The choice of $Threshold_{identify}$ varies for different network conditions and traffic patterns. It also depends on the composition of traffic and the desired level of the ability to control a DDoS attack. For instance, for a network that consists mostly of normal TCP traffic, source hosts reduce their packet rate whenever they detect congestion. Their average arrival rate in a large time window measured at source routers is usually low. In this case, we would choose a relatively small value for the identification threshold. On the other hand, if a network usually contains a large amount of UDP traffic, the packet rate in a same size of time window may be somewhat greater than the previous network. Therefore, we would consider setting a larger value for the threshold.

3.3 Preferential Dropping at Source Routers

After identifying the high rate flows we start to monitor and control them, which includes several steps:

(1) Measure each flow's arrival rate;

(2) Compute each flow's average response time;

(3) If the average response time is within a predefined range, send an ICMP control message to ask the output queue occupancy information at the destination router;

(4) Update each flow's state;

(5) Compute the drop probability of each flow and drop packets accordingly.

Figure 3.2 shows the basic preferential dropping procedure at a SRPD router. Whenever there is a new packet coming from a client, the SRPD router checks whether this packet belongs to a high rate flow. Packets from flows with normally low arrival rates are directly forwarded to the output queue and sent to the next hop. SRPD routers maintain a weighted average response time for all high-rate flows. The average response time at time t is calculated by:

$$T_{response}(t) := (1-w) * T_{response}(t-1) + w * newT_{response} \quad (3.1)$$

A weighted average avoids the situation where we unnecessarily drop packets due to bursty traffic or intermittent congestion along the path to the destination network. The value of w cannot be set too low otherwise the weighted average responds to traffic changes too slowly. It turns out that 0.002 is a proper value for w [33]. We have tried

the value of 0.2, 0.02 and 0.002 and the results have proved that 0.002 is a reasonable choice in our simulations.

If the average response time for a high rate flow is less than a minimum threshold for a certain period of time, the flow is no longer monitored. If the average response time for a flow is greater than a maximum threshold then all incoming packets belonging to the flow are dropped. If a SRPD router detects a long average response time then it believes the victim network is under attack. It assumes that the high rate flows more are responsible for congestion and only preferentially drop packets from these flows.

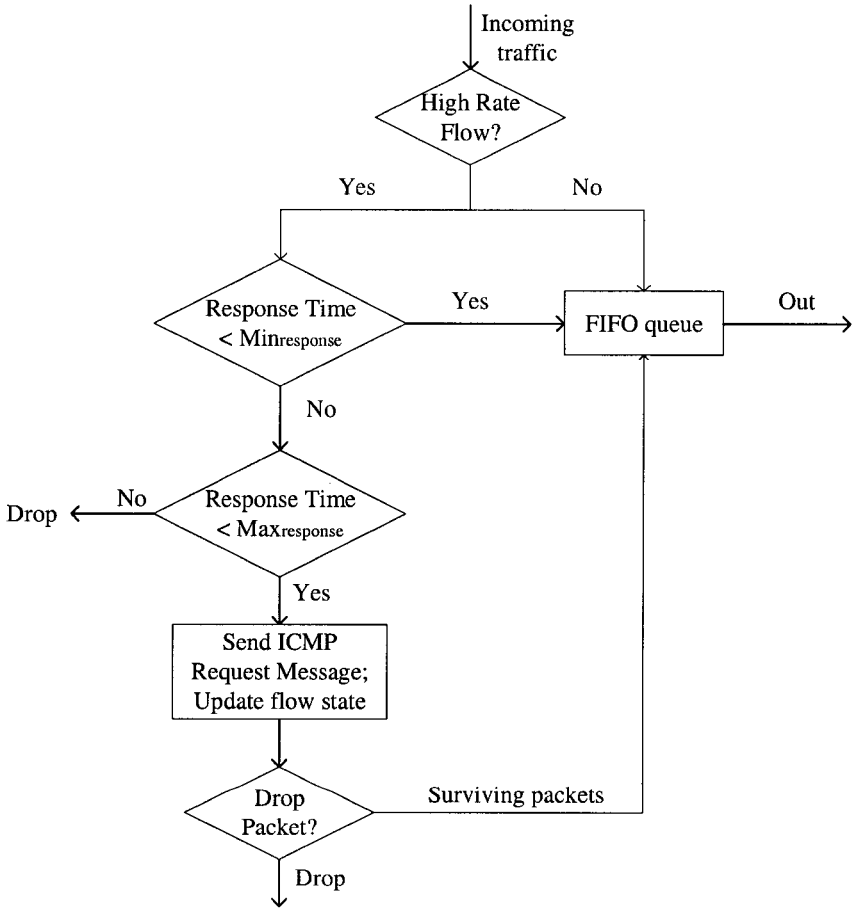

Figure 3.2 The flowchart of drop decision at a SRPD router

If the average response time is less than the maximum threshold, but greater than the minimum threshold, the SRPD router cannot be sure whether or not a client is part of a DDoS attack. The router sends an ICMP request message to the edge router at the destination network asking it about its buffer usage information. It is expected that, during DDoS attacks, the victim router has a much longer queue length at its output queue than normal. We also set a minimum threshold and a maximum threshold for the output queue occupancy. Upon receiving an ICMP request message from a source router, the victim router calculates its relative output queue occupancy rate and sends the value back to the source router. The average occupancy rate $R_{occupancy}$ at time t is computed as follows:

$$L_{queue}(t) := (1-w) * L_{queue}(t-1) + w * newL_{queue} \quad (3.2)$$

$$R_{occupancy} = L_{queue}(t) / B_{queue} \quad (3.3)$$

The CODE field of the ICMP response message is also set to indicate if the queue length increasing or decreasing. Queue length increases as congestion persists. Otherwise, it decreases or keeps constant. This information provides the source router with better knowledge of the congestion scenario so it can increase or decrease the packet dropping probability accordingly. When the source router receives the ICMP response message from the victim router, it computes the drop probability of each high rate flow based on the knowledge it has accumulated. Meanwhile, it updates a flow's state throughout the whole procedure of monitoring and controlling.

3.3.1 Computing the Drop Probability

A flow's drop probability is computed based on its average response time. At each source router, the response time, as shown in Figure 3.3, records the time duration between the time the request is transmitted from the source router and the time the response message is received at the source router (the time between point A and the point B in Figure 3.3). It consists of a round-trip time between a source router and a victim router, the transmission times at the victim router and the victim server, the processing time and the queuing delay at the victim router, and the service time at the victim server. Other sources of delay are usually negligible and have no effect on the accuracy of our scheme. For example, a source router does not care about the time duration from the time a client generates a request to the time that it arrives at the source router. We also ignore the propagation delay between the victim router and the victim server. These time periods are not affected by a DDoS attack. The response time is computed as:

$$T_{response} = 2*Latency_{sr\_vr} + 2*Trans_{vr} + Trans_{vs} + 2*PT_{vr} + ST_{vs} + QT_{vr} \quad (3.4)$$

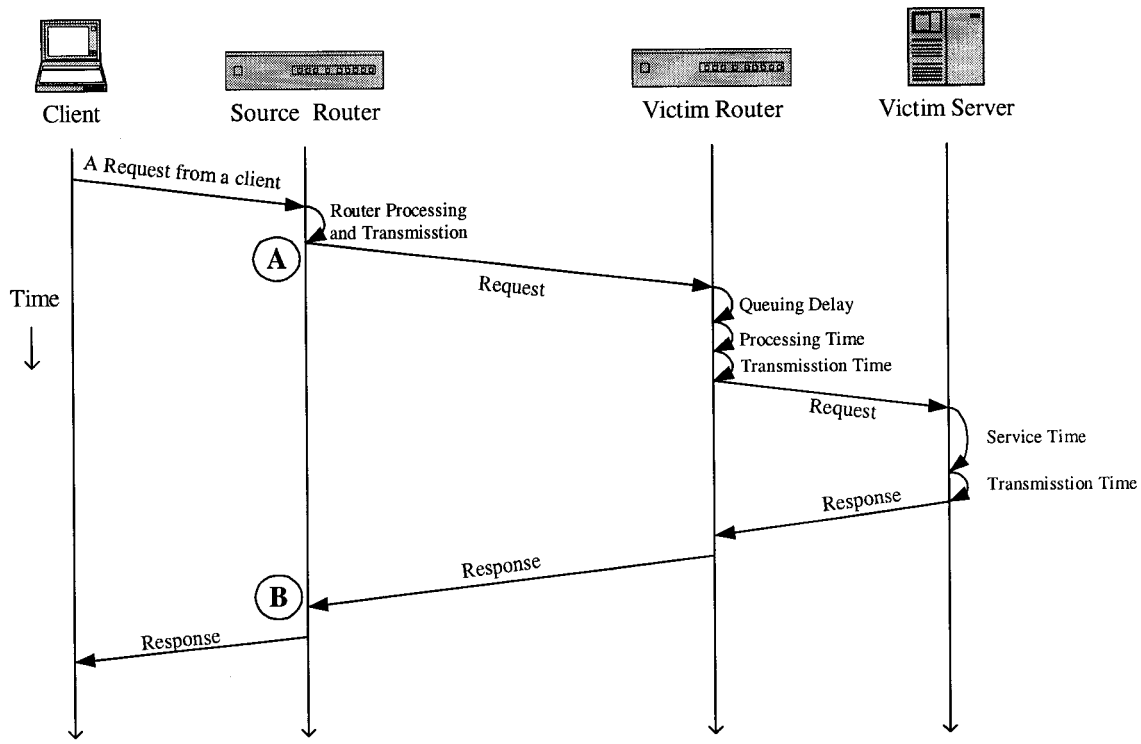

Figure 3.3: Average Response Time Calculation

Figure 3.4 shows the relationship between the packet drop probability and the average response time. When the response time is greater than the maximum threshold $Max_{response}$, all incoming high rate packets are dropped. While the average response time is between the minimum and maximum threshold, the packet drop probability is computed as follows:

$$Prob = Max_{prob} * \left(\frac{T_{response} - Min_{response}}{Max_{response} - Min_{response}}\right) \quad (3.5)$$

If the source router receives an ICMP response message with the CODE set to 1, which means the average output queue length at the victim router is increasing, the

source router increases the packet drop probability from high rate flows by β. The increased drop probability then becomes:

$$Prob' = (1 + \beta) * Prob \qquad (0<\beta<1) \qquad (3.6)$$

If the average response time is less than the minimum threshold or the output queue occupancy rate at the victim router is less than its minimum value, no packets are dropped.

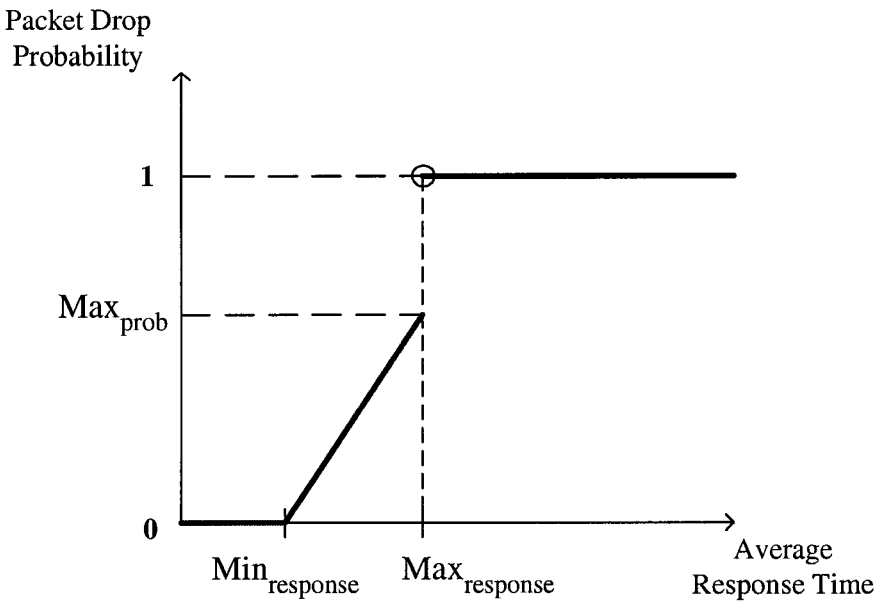

Figure 3.4 Packet Drop Probability vs. Average Response Time

Choosing proper minimum and maximum thresholds for average response time is another important issue of SRPD. These two thresholds determine when a source router starts to probabilistically drop packets from high rate flows and when it drops all incoming high rate packets. Large thresholds mean that the source router may take longer time to control a DDoS attack. In this case, if the congestion is already serious, the drop probability at the source router may not be high enough. On the other hand,

small thresholds may result in packets being dropped unnecessarily. In chapter 4, we study the effect of these two threshold values.

3.3.2 Queue Length Enquiry and Response Messages

If a high rate flow's average response time is between the minimum and maximum threshold, a control message is sent by the source router to the victim router to collect buffer usage information on the output queue connected to the victim machine. This message is sent at an interval of round trip time. Long queue length at the victim router is an indication of congestion. In SRPD, we propose a new ICMP (Internet Control Message Protocol) message to exchange information between the source router and the victim router.

An ICMP message [32] includes a *TYPE* field that identifies the message, a *CODE* field that provides further information about the message type, and a *CHECKSUM* field. Figure 3.5 shows the ICMP message format. An ICMP message payload depends on its type and code.

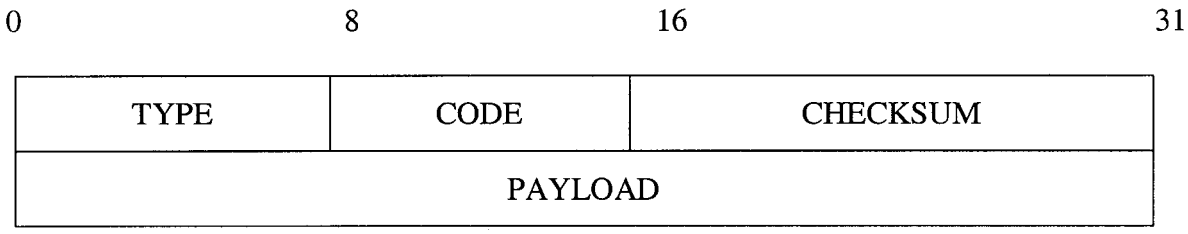

Figure 3.5 ICMP Message Format

In our scheme, when a high rate flow's average response time is between the minimum and maximum threshold, it sends a Queue Length Enquiry Message (an

ICMP Request message) to the victim router to ask about its output queue occupancy rate. In case of DDoS attacks, the victim router itself is very likely to have a high output queue length. Therefore, we use the output queue occupancy rate as an indication of an attack. A Queue Length Response Message (an ICMP response message) is used as feedback to the source router. Figure 3.6 shows the Queue Length Enquiry and its Response Message format.

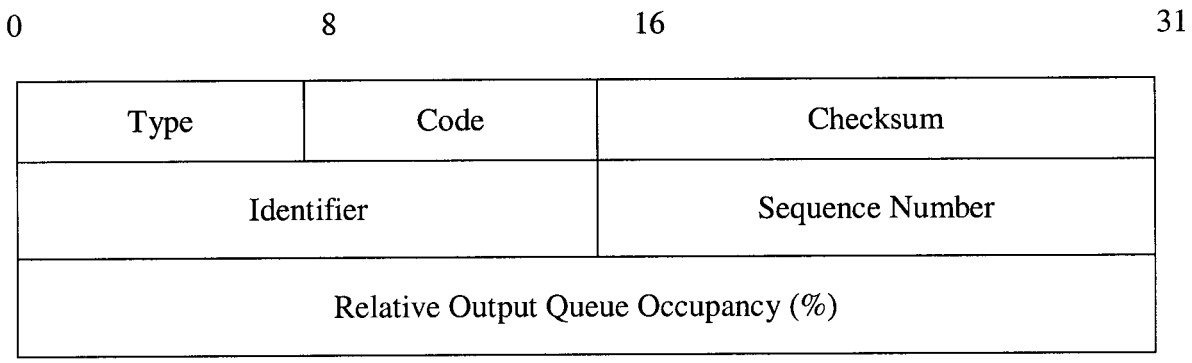

Figure 3.6 Queue Length Enquiry and Response Message

The detailed description of each ICMP field is given as follows:

- **Type:** 41 for Queue Length Enquiry Message
  42 for Queue Length Response Message
- **Code:** 0 Queue length is deceasing or is unchanged
  1 Queue length is increasing
- **Checksum:** The checksum is a 16-bit field used for error detection.
- **Identifier & Sequence Number:** These two fields are used by the enquiry source routers to match the replies with their requests. The "Identifier" is used to identify

a flow. The sequence number increments when each request is sent. The destination returns these same values in the response message.

- **Average Output Queue Occupancy Rate:** The payload of the ICMP response message is a 32-bit field used by the victim router to describe the average occupancy rate at its output queue. For example, the value of 0.36 in the payload means that the average queue length occupancy is 36% of the queue size. Upon receiving the response message, the source router may decide whether or not to drop packets based on this value.

3.3.3 Flow Information Table

We maintain a table at each source router to keep the necessary information for each flow. This flow table assists the router to make decisions whether or not to drop packets. The format of the flow table is illustrated in Figure 3.7:

Src_Add	Desti_Add	Rate_Arrival	Ave_ResponseTime	State_Flow	Res_Refresh_TS

Figure 3.7 Flow Information Table

The fields of the table are:

- Src_Add: Source client's IP address
- Desti_Add: Destination IP address
- Rate_Arrival: Average arrival rate. It is updated periodically, say every second.
- Ave_ResponseTime: The average response time of the flow.

- State_Flow: The state of the flow. The state may keep changing all the time. It is another key point to determine whether or not a packet needs to be dropped.
- Res_Refresh_TS: Response time refresh timestamp. A monitored high rate flow will change to be unmonitored if its average response time changes to be less than the minimum threshold for a period of time. This timestamp records the time at which the response time fell below the minimum threshold.

3.3.4 Flow States

We use the notion of the state of a flow to maintain information necessary to the decision to drop packets. We have the following 7 possible states for a flow:

- **Unmonitored:** A flow is not currently being monitored and is not considered to be a possible attack flow.

- **Monitored:** The flow is identified as a high rate flow and is being monitored. No further action is conducted in this state. It may switch to other states based on the flow's average response time and the output queue occupancy rate in the Queue Length Response message from the destination router.

- **Waiting:** A flow enters the Waiting state when the source router sends a Queue Length Enquiry message to the destination router for the flow. No packet is dropped in this state. It can then switch to the state of "Non-increasing", "Increasing" or "Dropping" upon receiving the victim router's queue length information from the Queue Length Response message.

- **Non-increasing:** A source router sets a flow's state to "Non-increasing" when it receives a Queue Length Response message from the victim router informing it that the output queue occupancy rate is between the minimum and maximum threshold from the response message payload and the CODE field is set to 0. An arriving packet at a source router with its flow's state in "Non-increasing" is probabilistically dropped. The exact dropping probability is calculated from the equation (3.5).

- **Increasing:** A source router sets a flow's state to "Increasing" when it receives a Queue Length Response message telling it that the output queue occupancy rate is between the minimum and maximum threshold and the response message's CODE field is set to 0, which means that the average queue length is increasing. An incoming packet with this state is probabilistically dropped with a probability calculated from equations (3.5) and (3.6).

- **Timeout:** If a source router sends a Queue Length Enquiry message to the destination router but cannot receive the response in a time period of $T_{icmp\_timeout}$, it sets the flow's state to "Timeout". The ICMP request message may have been dropped on its way due to congestion or the destination router may not support SRPD and so ignores the message. Incoming packets are probabilistically dropped based on equation (3.5).

- **Dropping:** A flow enters the Dropping state when the average response time is greater than the maximum threshold or when the output queue length occupancy is greater than its maximum value. Congestion is considered to be serious and

all incoming high rate packets have to be dropped until the congestion is relieved.

Figure 3.8 shows the state diagram for a flow illustrating the state transitions described above.

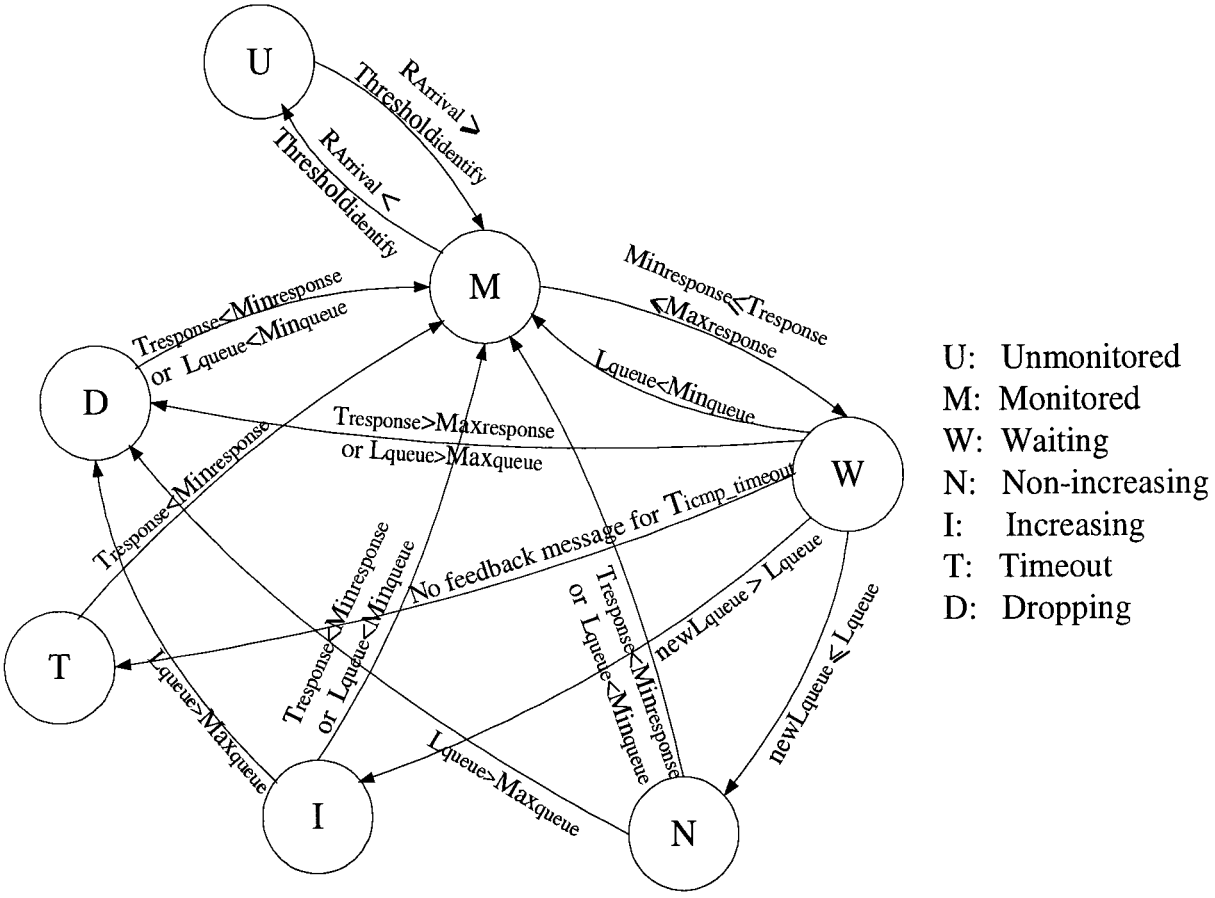

Figure 3.8 Flow State Diagram

3.4 Summary

In this chapter, we give a detailed description of our proposed Source Router Preferential Dropping (SRPD) scheme to detect likely distributed denial of service attacks and control them at their first place. The SRPD scheme is parameter sensitive. The arrival rate threshold for identifying high rate flows, the minimum and maximum average response time thresholds, the minimum and maximum average queue length thresholds need to be carefully configured based on the normal traffic composition and behavior. Several other properties of our scheme make it a promising approach to defeating DDoS attacks:

- SRPD can significantly reduce the amount of collateral damage to legitimate packets. Monitoring flows' behavior at their source makes it easier to identify attacking flows.

- SRPD only needs to be implemented at the edge routers close to clients' networks. Core routers in the backbone network do not have to be involved, which makes this scheme practical.

- Our proposed ICMP message is compatible with the existing ICMP protocol. Even if a destination router cannot recognize a Queue Length Enquiry message and thus cannot cooperate with the enquiring source router, SRPD still can detect attacks with high probability by measuring flows' average response time and control attacks by preferentially dropping high rate packets.

Chapter 4

Performance Evaluation

In this chapter, we examine and evaluate the performance of the Source Router Preferential Dropping (SRPD) scheme. The results are compared with the Pushback scheme [35]. Pushback includes two main components: a local Aggregate Congestion Control (ACC) mechanism and a cooperative pushback mechanism. Pushback is intended to drop enough attack packets to save bandwidth for legitimate packets, thus alleviating the influence on good traffic. Pushback is currently considered one of the most promising mechanisms to defend against DDoS attacks. Therefore, we choose it to compare with SRPD. Section 4.1 describes the simulation model, which consists of SRPD model and Pushback model. Section 4.2 outlines the different scenarios we use in our simulations. Section 4.3 gives the parameter settings in the simulations. Section 4.4 explains the criteria that we use to examine and judge the performance. Section 4.5 describes the main components of our simulation software. The effects of high-rate flow identifying threshold, attacking packet rate and the average response time thresholds on the preferential dropping rate are represented in Section 4.6. Section 4.7 provides a summary of this chapter.

4.1 Simulation Model

In this section, we describe the simulator we have developed to examine and compare the performance of SRPD and the Pushback. This simulation model includes the SRPD model and the Pushback model.

4.1.1 The SRPD Model

To evaluate SRPD's capability of defeating attacks, we simulate distributed denial of service attacks using the network architecture shown in Figure 4.1.

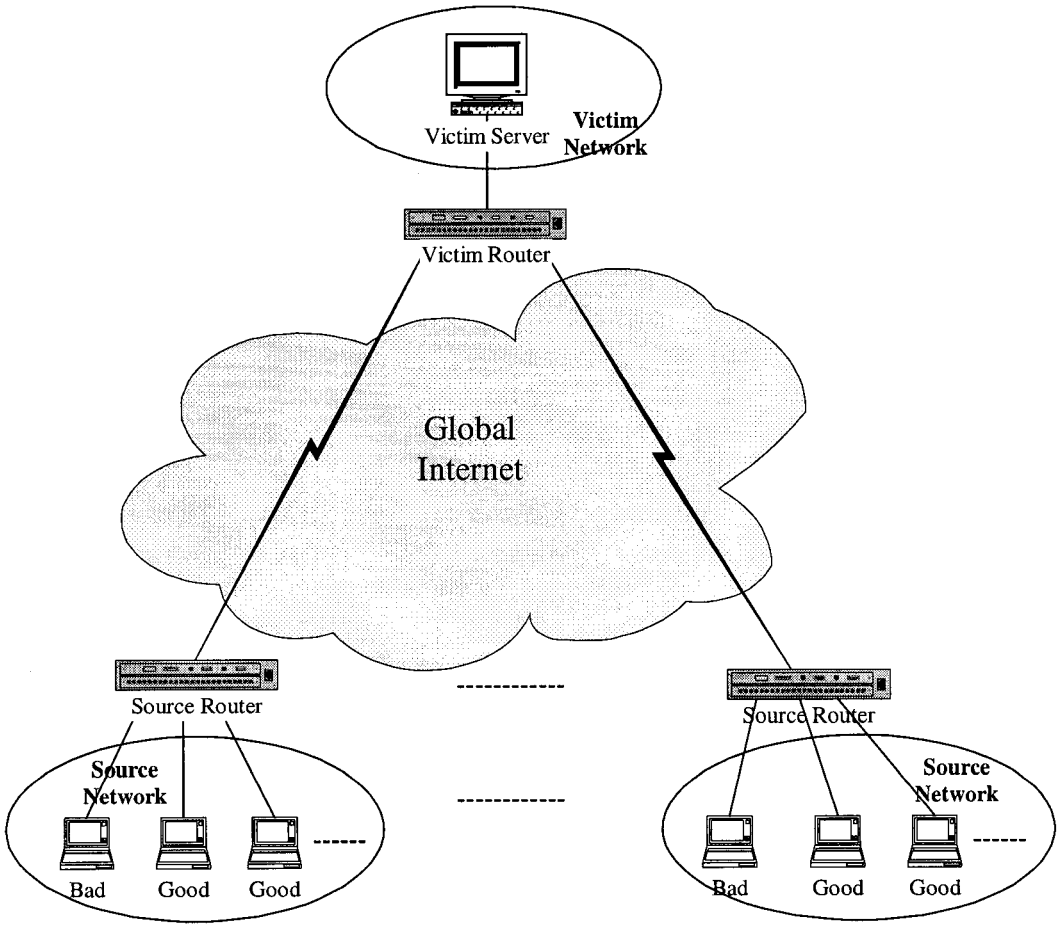

Figure 4.1 SRPD Simulation Architecture

There are multiple source networks, each of which includes both legitimate (good) clients and attacking (bad) clients. Attacking packets originate from bad clients targeting the destination network. However, those bad clients themselves may not realize that they are generating attack packets. The attack flows are called "irresponsible" flows in that they do not conform to normal end-to-end congestion control mechanisms, which means that they do not reduce their rate even in times of heavy congestion.

The victim router is connected with the source routers through the global Internet. They are located in different networks and areas. In a DDoS attack, attacking packets, which come from heterogeneous source networks, all head for the victim network. The final goal is to flood the victim router or the victim server, rendering them unable to provide normal service to legitimate users. In time of an attack, good clients may also send legitimate request packets to the victim server. However, they may not pass through and suffer drops due to the serious congestion in the victim network.

In our simulations, we use a single victim server and a victim router connected by a 10Mbps or 100Mbps link. We do experiments with different destination link bandwidth to get different simulation scenarios. All other links are assumed to have 10Mbps bandwidth. We construct 10 source networks, each of which consists 1 bad client and 3 good clients. All clients are defined as UDP sources. Legitimate clients generate outgoing traffic at a rate of 0.1Mbps while malicious clients' sending rate varies from 0.8 Mbps to 10Mbps.

4.1.2 The Pushback Model

The Pushback mechanism includes two main components: a local Aggregate Congestion Control (ACC) mechanism and a cooperative Pushback mechanism. The local ACC is implemented at the congested router near the destination. It is in charge of detecting the occurrence of DDoS attacks and controlling the aggregate traffic. Pushback defines an aggregate as a collection of packets that share a same destination IP address prefix. For example, all packets going to the subnet 64.3.11.0 with network mask 255.255.255.128 (a subnet of a class C address) belong to a same aggregate. The reason why pushback does not use source IP address to determine an aggregate is that source IP addresses are not trusted in the case of a DDoS attack. ACC measures the packet drop rate at the congested router's output queue. If the drop rate exceeds a predefined threshold, ACC starts to identify the aggregate traffic that consumes most of the bandwidth and thus is mainly responsible for the congestion.

Figure 4.2 shows a DDoS attack in the Pushback mechanism architecture. R1~R8 represent routers at different levels from the destination network 64.3.11.0 (a class C network). R8 is the edge router attached to the victim network. The paths used by attack traffic are shown in the bold lines. The arrows indicate the attack direction. Attack traffic is characterized by a congestion signature, which is the set of packets going along the bold lines targeting the network 64.3.11.0. In times of a DDoS attack, router R8 is congested. The local ACC at this router identifies that the aggregate 64.3.11.0 is the attack signature. Router R8 therefore starts to probabilistically drop

packets belonging to this aggregate until the drop rate at its output queue goes below the threshold.

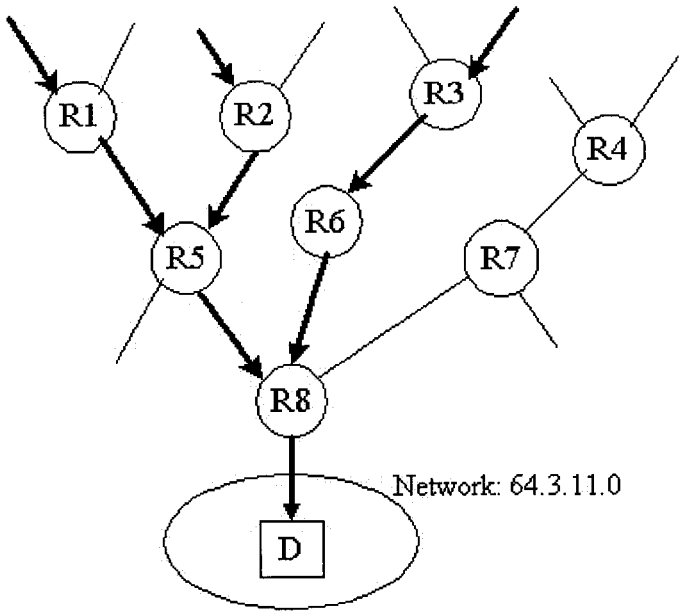

Figure 4.2 A DDoS attack in Pushback Topology

The local aggregate congestion control, however, is not able to distinguish between the good and the bad traffic within the aggregate, thus punishing traffic coming from all directions equally. For example, some legitimate packets from routers R1, R2 and R3 to the destination network may also be discarded at R8 in time of DDoS attacks. Moreover, in a severe attack, ACC may not have enough ability to control the attack and relieve the congestion by itself. The Pushback mechanism allows a router to ask its adjacent upstream routers to limit the rate of the aggregate traffic. With pushback, R8 sends pushback messages to its upstream routers R5 and R6 to limit the packet rate of the aggregate 64.3.11.0. However, R8 does not require R7 to rate-limit the aggregate traffic targeting the destination network in that the arrival rate from R7 to R8 within the aggregate is not a "contributing" traffic, which means that the traffic

from this upstream router does not occupy a significant fraction of the total aggregate. It is considered to be legitimate traffic that just happens to match the attack signature so is not punished.

We built a Pushback model to evaluate the performance of the current proposed defeating mechanism and compare with SRPD. We find out that SRPD outperforms Pushback in that it brings much less collateral damage to legitimate traffic while defending against DDoS attacks. In the Pushback model, we use 10Mbps for each link. In case of DDoS attacks, the edge router in the destination network is congested and then sends pushback messages to its upstream routers. SRPD outperforms Pushback in another respect, namely SRPD can also handle the scenario that the victim host is congested. If congestion only happens at the victim host, Pushback has no way to detect and control the attack at all.

4.2 Simulation Scenarios

In our simulations, we consider different scenarios in time of DDoS attacks and examine SRPD's performance under different situations. The first one is that the victim router is congested. The second one is that the victim server is overloaded. The second scenario usually happens in the situation of a large link bandwidth and a server with limited processing capability.

4.2.1 Victim Router is Congested

A victim router becomes congested relatively fast while under a DDoS attack. The reason is that most routers in the current Internet still use first in first out (FIFO) queue as their queuing scheme. In time of a DDoS attack, the victim router is congested as long as the aggregate arrival rate is greater than the destination bandwidth, which is the bandwidth of the link between the victim router and the victim server. When the destination link does not have enough bandwidth to transmit the incoming packets, the router has to buffer them at the output queue. Since a router's buffer size is limited, it will eventually overflow. Newly arrived packets have to be discarded unless congestion is relieved.

4.2.2 Victim Server is Overloaded

The victim server can also become overloaded while under attack. A server's capacity of serving concurrent incoming requests is not infinite due to limited buffer size and limited CPU processing ability. If the packet arrival rate exceeds the server's processing rate, newly arrived packets have to be dropped. Fortunately, this scenario is less likely to happen than the previous scenario because many websites usually load balance the workload by using several or even many hosts to provide a single type of Internet service to prevent the servers from being overloaded. In this case, the edge router is more easily to be flooded by attacking traffic.

4.3 Simulation Parameter Setting

The parameters used in the simulations are summarized in Tables 4.1, 4.2, 4.3 and 4.4. In our simulations, we use a fixed packet size and set it to 500 bytes.

Parameter	Meaning	Value
$Latency_{sr\_vr}$	The propagation delay between a source router and a victim router	100~300ms
B_{link}	The bandwidth of links	10 or 100Mbps

Table 4.1 Parameters For Links

Parameter	Meaning	Value
R_{good}	Legitimate packet sending rate	0.1Mbps
R_{bad}	Attacking packet sending rate	0.8~10Mbps
$Threshold_{identify}$	The threshold for identifying those high packet arrival rate flows at source routers	0.5~2.5Mbps
$Min_{response}$	The minimum threshold for average response time	100~600ms
$Max_{response}$	The maximum threshold for average response time	400~1000ms
$T_{request\_timeout}$	The period that a request packet times out at a source router	10 sec
$T_{icmp\_timeout}$	The period that an ICMP Queue Length Enquiry packet times out at a source router	8 sec
$T_{refresh}$	The time period for a source router to review the average response time of the attack flows before relieve them from unmonitored flows	2 sec

Table 4.2 Parameters For Source Routers

Parameter	Meaning	Value
$Trans_{vr}$	The time it takes a victim router to transmit a packet	0.04 or 0.4ms
B_{queue_vr}	The output queue size of a victim router	500 packets
Min_{queue}	The minimum threshold for average queue length	150
Max_{queue}	The maximum threshold for average queue length	300

Table 4.3 Parameters For Victim Routers

Parameter	Meaning	Value
ST_{vs}	Average packet service time at the victim server	40~240ms
B_{buffer_vs}	The buffer size of a victim server	500

Table 4.4 Parameters For Victim Servers

4.4 Evaluation Criteria

In our simulations, we use the following criteria to evaluate the performance of SRPD: attack packet drop rate, legitimate packet drop rate, average response time, queue length at the victim router and victim server.

4.4.1 Attack Packet Drop Rate

Our main goal is to identify and drop the attacking packets. The more attacking packets dropped, the less they consume network resources. Ideally, all the attacking packets can be dropped and all legitimate packets can go through. However, in practice, it is impossible to be exactly sure which packet is good and which is bad. We therefore drop as many bad packets as congestion brought by the DDoS attack is

relieved. Since SRPD detects and controls DDoS attacks at their sources, packet dropping mostly happens at source routers.

4.4.2 Legitimate Packet Drop Rate

Reducing collateral damage is an important issue in defeating distributed denial of service attacks. A defense scheme that also drops a large percentage of good packets still makes the DDoS attack successful. Minimizing the collateral damage is therefore another key aspect in SRPD.

4.4.3 Average Response Time

The response time is the time from when a source router sends a request out until it receives the response message. In times of a DDoS attack, the queuing delay at the victim router and the service time at the victim server can be much longer than in normal situations and keep high throughout the whole period of the attack. However, with an attack control scheme, the response time should decrease somewhat to an acceptable level, or even to normal. The sooner the response time returns to normal, the better the scheme controls the attack.

4.4.4 Queue Length at Victim Router

The queue length at a victim router represents whether or not there is congestion and how serious the congestion is. In the scenario with a narrow-bandwidth destination link, the output queue length of the victim router can be abnormally large and still keep increasing when the router is under attack. A persistent large queue length is an

indication of an attack. With an attack control scheme, the queue length should decrease in length and go back to the normal level sooner or later. Therefore, we choose the queue length at the victim router as one of the metrics to measure SRPD.

4.4.5 Victim Server Workload

If the destination link has enough bandwidth to accommodate all arrival packets and passes them to the victim server, congestion then happens at the server. A server does not necessarily use a first in first out queue. It can process many requests simultaneously. In case of a small-capacity server with a high-bandwidth destination link, the server is not able to process too many concurrent requests while under attack. Extra packets have to be discarded if the maximum concurrent request number has been reached. The workload of a victim server provides general information about the congestion situation. A server's average workload represents the ratio of the concurrent number of the requests. It is computed as follows:

$$\text{AverageWorkload} = \frac{\text{Average Concurrent Requests Number}}{\text{Maximum Number of Concurrent Requests}}$$

When the workload decreases to the normal level this indicated that congestion is relieved.

4.5 Simulation Software Implementation

In this thesis, we use the Java programming language to construct a simulator and use it to conduct discrete event driven simulation. This simulator consists of six major components (see Appendix A for details):

- **Client Cluster:** A client cluster is responsible for generating request packets that include both legitimate and malicious packets.
- **Source Router:** A source router forwards the requests coming from its connected clients to the outgoing links. It determines whether or not to drop or forward a packet based on its SRPD algorithm. When necessary, it sends a Queue Length Enquiry ICMP message to a destination router asking the queue length occupancy rate. We assume that source routers have enough buffer size so that they never overflow which means congestion does not take place at source networks.
- **Victim Router:** A victim router forwards incoming requests to its connected victim server. It is also responsible for replying Queue Length Enquiry Messages from source routers. A victim router drops packets if its output queue length is full.
- **Victim Server:** A victim server processes incoming requests and replies to them with corresponding response messages. In case of an attack, when a server does not have enough resources to process newly arrival requests, they have to be dropped.
- **Network Link:** A network link passes messages from one end to the other with a specified link delay.
- **Event Manager:** This event manager is in charge of dispatching and queuing simulation events. All simulation events are stored in an event queue and processed by the event manager.

4.6 Simulation Results

In this section we describe our simulation experiments and analyze the results. We compare the results of SRPD with the results of Pushback to show the capability of defending attacks of these schemes. Variable values are applied to the key parameters to imitate different situations and discuss the effect they bring to the performance of SRPD.

4.6.1 Simulations of Congesting the Victim Router

We first simulate the scenario that the victim router is congested. In the section 4.6.1.1, we represent and discuss the simulation results of both SRPD and Pushback. We can see how SRPD outperforms Pushback. Therefore, in the latter subsections 4.6.1.2- 4.6.1.4, we analyze the influences of different parameter settings on the performance of SRPD only.

4.6.1.1 Comparing SRPD with Pushback

We use three types of criteria to measure the performance of SRPD and Pushback, namely, legitimate packet drop rate, attack packet drop rate and victim router's queue length. A better scheme should drop less good packets, more bad packets and have a shorter queue length during a DDoS attack. In all the simulation experiments, DDoS attacks start at the 80th second, last about 8 minutes and then end at the 560th second. Each attack source starts to send malicious packets at an interval of one second, which means that the second attack source starts to generate attack traffic one second later than the first one. In the scenario of congesting a victim router, the bandwidth of

the destination link is 10Mbps. So as long as the aggregate traffic exceeds 10Mbps, the victim router is congested. The parameter settings in the experiments of comparing with Pushback are listed below:

Bandwidth between victim router and victim server:	10Mbps
Propagation delay between source router and victim router:	100ms
High-rate identification threshold:	0.5Mbps
Average response time minimum threshold:	200ms
Average response time maximum threshold:	500ms
Victim router to transmission time for each packet:	0.4ms

Pushback includes both a local aggregate congestion control mechanism at the victim router and a pushback mechanism that sends the pushback messages upstream until they reach the source routers. Local ACC is triggered when the packet drop rate at the output queue exceeds the configured value of 10%. ACC attempts to identify the aggregate traffic responsible for this serious congestion. It then limits the rate of the traffic by probabilistically dropping from this aggregate until the drop rate reduces below 10%. The amount of packet dropping depends on how much traffic exceeds the target bandwidth. ACC drops the excess traffic from its identified aggregates until the drop rate at the output queue is below the predefined level. ACC is not able to distinguish between the good and the bad traffic within an aggregate. It drops legitimate packets that belong to the aggregate as well. After the victim router identifies the aggregate, it enables Pushback. It sends pushback messages to its upstream adjacent routers asking them to limit the rate of its identified aggregates. When the pushback messages arrive at the source routers, the source routers identify which input links are the contributing links that are primarily responsible for the

aggregate traffic. The source routers then probabilistically drop the packets coming from these contributing links until they do not receive more pushback messages from downstream routers.

Figures 4.3 and 4.4 illustrate the legitimate and attack packet drop rate in SRPD and Pushback under different attack rates. In SRPD, source routers attempt to detect a possible attack as soon as a flow's rate is higher than a threshold and start to probabilistically drop when the victim router's queue length exceeds its minimum threshold. However, in Pushback, ACC is not triggered until the ambient drop rate is greater than the configured value of 10%. Therefore, SRPD responds to and control a DDoS attack faster than Pushback.

In SRPD, good packets are dropped only during the initial few seconds of the attack due to serious congestion at the victim router. After the congestion at the victim router is alleviated, the rest of the good packets can go through and reach their destinations. Pushback brings more or less collateral damage to good traffic throughout the whole procedure of attacks due to its local ACC's inability to differentiate good packets from bad packets. Although pushback messages are helpful in protecting the legitimate traffic within the aggregate, the damage from ACC is unavoidable. Figure 4.3 shows that legitimate traffic in Pushback always suffers more drops than in SRPD. For instance, at the 92nd second, the legitimate packet drop rate of SRPD is 63% lower than that of Pushback for an attack rate of 0.8Mbps, 7.7% at the rate of 1.2Mbps, 45.1% at the rate of 1.6Mbps, and 40.9% at the rate of 2.0Mbps.

In Pushback, good packets still suffer dropping, which ranges from 2% to 12% depending on the attack rate, even after the attack is under control.

Figure 4.4 shows the attack packet drop rates in SRPD and Pushback. Each point in this figure represents the average attack packet drop rate from the moment the attack starts until the attack ends. SRPD always drops more attack packets than Pushback does. The average attack packet drop rate of SRPD is more than that of Pushback by 43.7% at the attack rate of 0.8Mbps, 18.6% at the rate of 1.2Mbps, 15.4% at the rate of 1.6Mbps, and 20.6% at the rate of 2.0Mbps. This is because SRPD can more precisely pinpoint the attack traffic at the flow level. Pushback identifies attack traffic at aggregate level that includes a large number of flows, which consist of both the attack and legitimate flows. As well, Pushback allows a relatively high ambient drop rate at the output queue (10 percent), which means it can endure more packet drops before the detection mechanism is invoked. SRPD, however, does not permit the output queue length to exceed its maximum threshold. Therefore, more attack packets can pass by and reach their destination in Pushback than in SRPD.

Figure 4.5 represents the percentage rate of the output queue length occupancy at the congested victim router. In SRPD, the queue length goes back to the normal level soon after the attack is under control. By reducing the output queue length at the victim router, packets tolerate less queuing delay at the congested router. Output queue length in Pushback remains long until the pushback mechanism is invoked. With the invocation of pushback, upstream routers limit the rate of aggregate traffic

and thus the queue length at the victim router decreases significantly. Figure 4.5 shows that SRPD can reduce the queue length of the congested router 5~16 seconds faster than Pushback under various attack intensities. Long queuing delay results in long response time and may lead to a service degradation, which is another demonstration of DDoS attacks and not expected by legitimate users. The sooner the queue length goes back to normal, the sooner the congestion is alleviated. This percentage rate of the output queue length at the congested victim router can show that SRPD is more powerful in controlling and relieving the congestion than Pushback.

Furthermore, SRPD outperforms pushback in that it can also deal with the scenario that the victim server is overloaded (as will shown in section 4.6.2). If the link between the victim router and the victim server has enough bandwidth and congestion happens at the server, Pushback has no way to detect and control the attack because it only depends on the ambient drop rate at the victim router to detect ongoing DDoS attacks. In this case, the victim router is not congested so no packets are dropped at the output queue. However, SRPD can still detect the attack by computing the average response time of the high-rate flows and control the attack by preferential dropping.

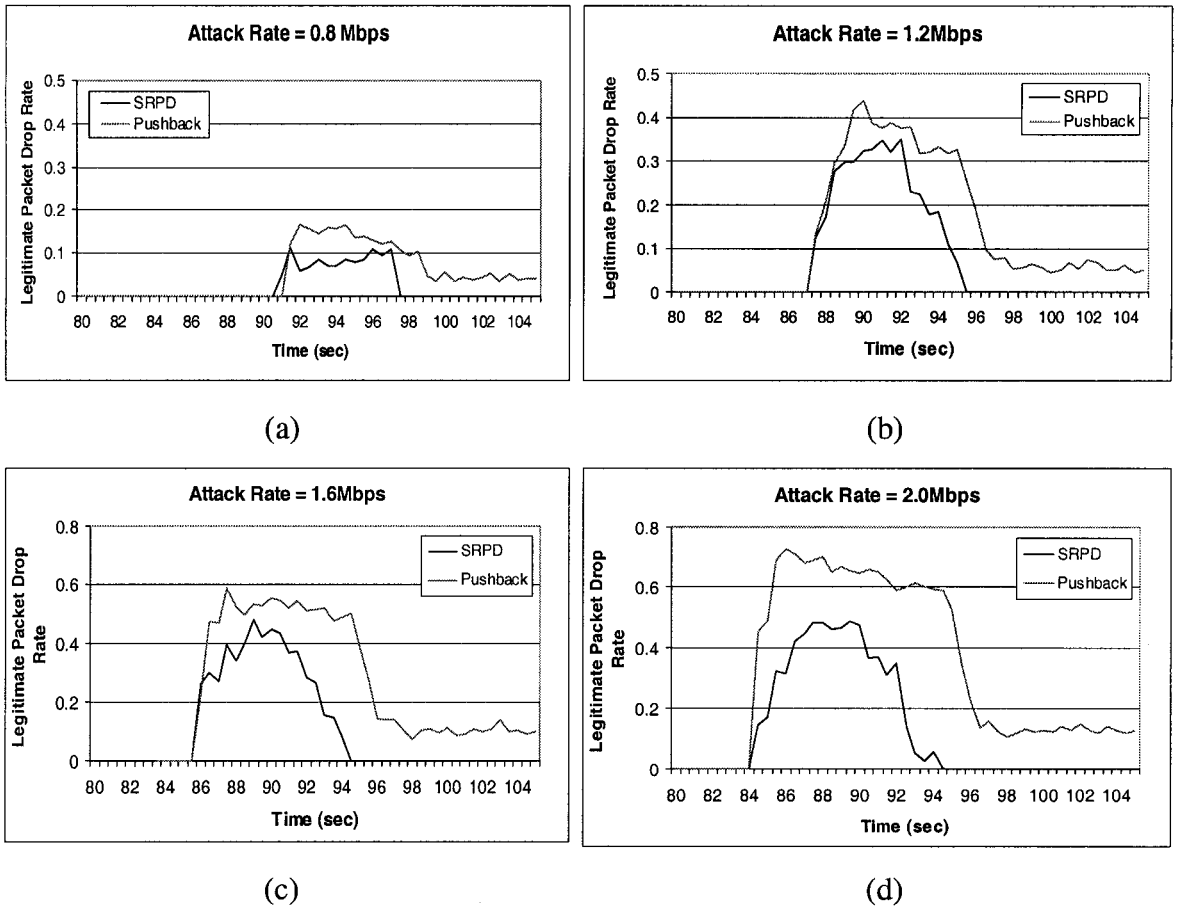

Figure 4.3 Legitimate Packet Drop Rate (Compare with Pushback)

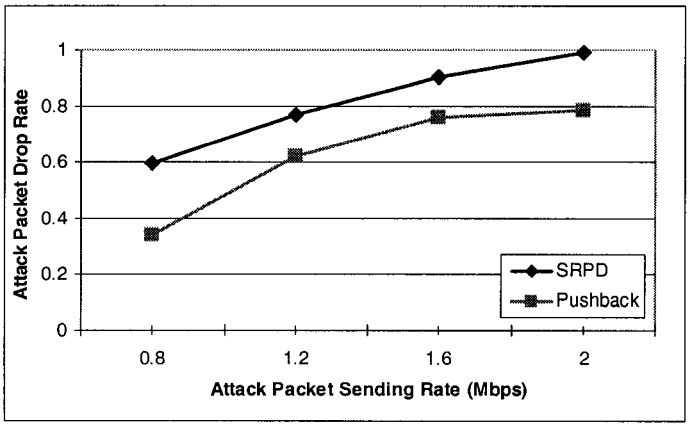

Figure 4.4: Attack Packet Drop Rate (Compare with Pushback)

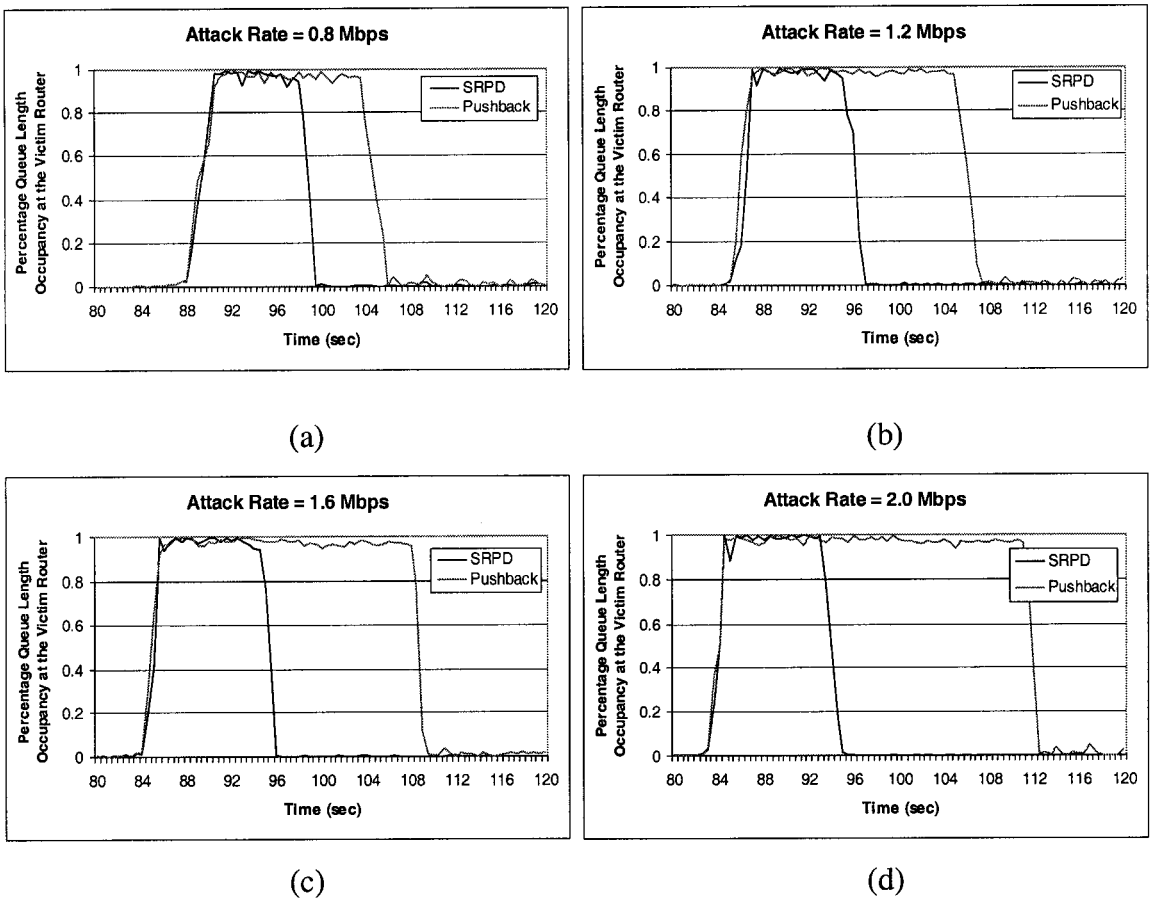

Figure 4.5: Percentage of Queue Length Occupancy at the Victim Router (Compare with Pushback)

4.6.1.2 Effect of the High-rate Identifying Threshold

Choosing a proper value for the identification threshold, $Th_{identify}$, is one of the most important issues for SRPD. Improper setting of $Th_{identify}$ may result in monitoring more flows than necessary or failure to identify attack traffic. The parameter settings in the experiments with different high-rate identification thresholds are given as follows:

Bandwidth between victim router and victim server:	10Mbps
Propagation delay between source router and victim router:	100ms

High-rate identification threshold:	0.5~2.5Mbps
Average response time minimum threshold:	200ms
Average response time maximum threshold:	500ms
Victim router to transmission time for each packet:	0.4ms

Figures 4.6 and 4.7 show the effect of different identification threshold settings under various attack intensities. Each source network configures its own threshold $Th_{identify}$ based on its traffic components and patterns. The higher the threshold is set, the less the attack flows are identified. We vary the threshold between 0.5Mbps and 2.5Mbps in different source networks and then compare the results when different numbers of attack flows are identified. For example, if the attack rate is at 1.6Mbps, the source router with its identification threshold set to 2.5Mbps cannot detect the attack flow coming out of its network. In an experiment where 8 attack flows are identified, the detection rate is 80% since there are altogether 10 attack flows in our simulations. Figure 4.6 illustrates the good packet drop rates when different percentage of attack flows is detected. The more the attack flows are identified, the less the good packets suffer drops and the shorter the attack is conquered. If all attack flows are detected by the source routers, it takes SRPD 6~10 seconds to control the attack. If 80% of flows are detected, SRPD needs 9~12 seconds to defeat the attack.

A successful defense against a DDoS attack in SRPD needs the cooperation of enough source routers. If too few source routers are involved, defeating DDoS takes a longer time and may even fail. Figure 4.7 shows the attack packet drop rate when different numbers of attack flows are recognized. The average attack packet drop rate

is calculated the same way as mentioned in the previous section. As expected, the more attack flows are identified, the more bad packets are dropped. At the attacking rate of 2Mbps, if all attack flows are identified, 98.9% of the malicious packets can be dropped at their sources.

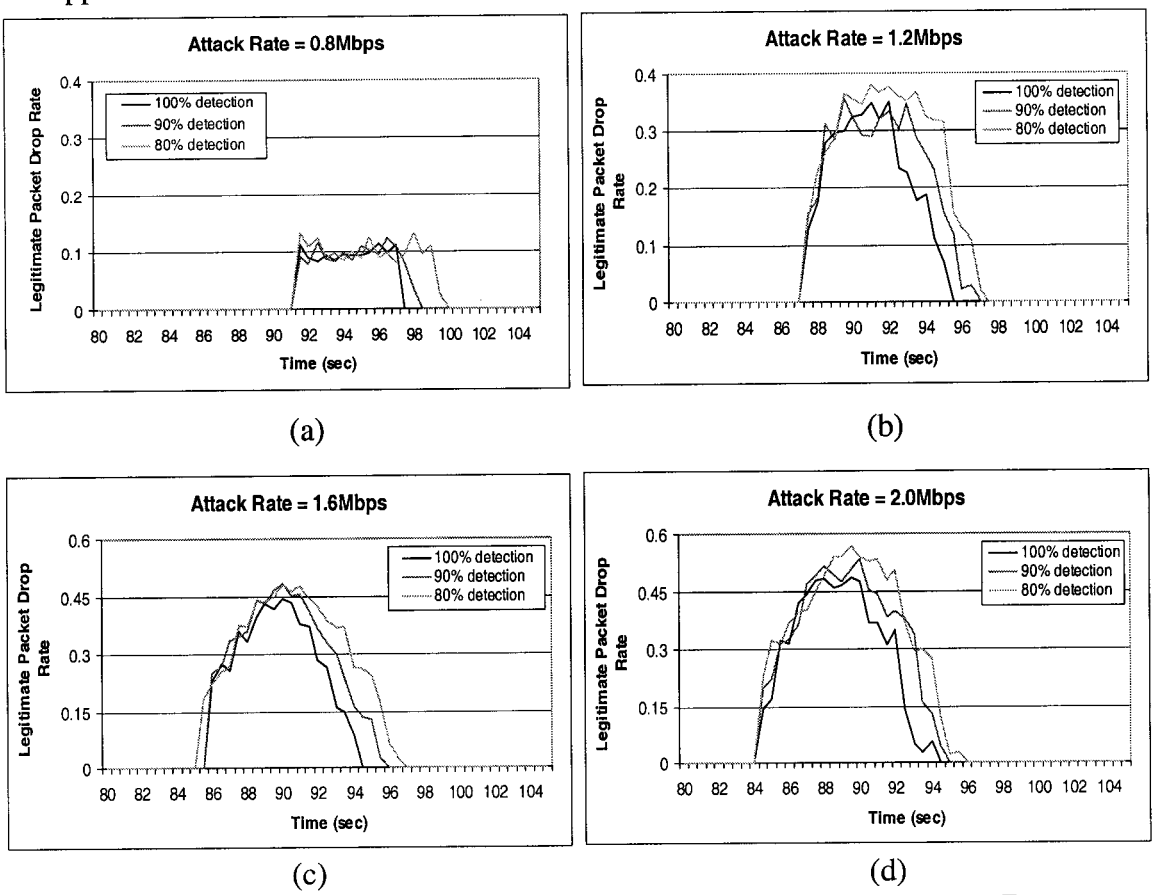

Figure 4.6: Legitimate Packet Drop Rate with Different Detection Rate

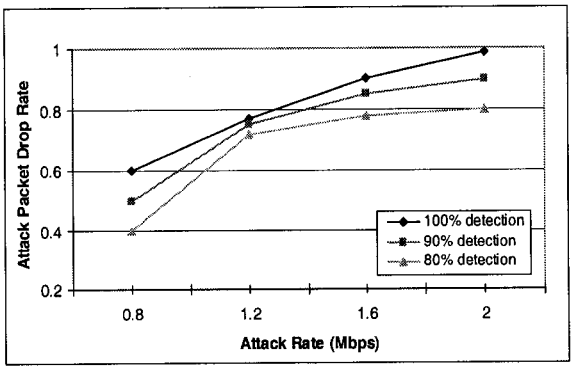

Figure 4.7: Attack Packets Drop Rate with Different Detection Rate

4.6.1.3 Effect of the Minimum Response Time Threshold

A flow's drop probability is computed based on its average response time. Setting proper thresholds for the average response time is of utmost importance. The minimum and maximum thresholds can decide when probabilistic dropping starts at a source router and how many packets are dropped. In sections 4.6.1.3 and 4.6.1.4, we discuss the effects of these two threshold settings, respectively. Some parameters that we used in the experiments to study the effect of minimum response time threshold settings are shown below.

Bandwidth between victim router and victim server:	10Mbps
Propagation delay between source router and victim router:	100ms
High-rate identification threshold:	0.5Mbps
Average response time minimum threshold:	100~400ms
Average response time maximum threshold:	500ms
Victim router to transmission time for each packet:	0.4ms

In this experiment, the round trip end-to-end delay is set to 100ms. The minimum threshold for the average response time varies from 100ms to 400ms while the maximum threshold ranges from 400ms to 900ms to observe their influences on performance. We set the maximum threshold to 500ms in order to study the effect of minimum threshold settings. Figure 4.8 and 4.9 illustrate the good and bad packet drop rate for different minimum threshold settings. Results show that the minimum threshold of the average response time does not significantly affect the performance. This is because the minimum threshold is not the only factor in making the decision to start dropping at the source router. SRPD does not immediately start to probabilistically drop once the average response time exceeds the minimum

threshold. When the response time is greater than the minimum threshold, a source router sends a Queue Length Enquiry Message to the victim router asking about the congestion situation. If no congestion is found at the victim router, the source router does not immediately drop packets.

Small values of the minimum threshold mean that source routers send Queue Length Enquiry Messages earlier than that of large values. For instance, when the minimum value is set to 100ms, a source router sends a Queue Length Enquiry Message to the victim router as soon as the average response time exceeds 100ms. However, at this time, there may not be any congestion at the victim router. The source router then does not conduct further action on the monitored flows. Unless this minimum threshold value is set very high, SRPD is not very sensitive to it. The side effect of improper minimum threshold is that source routers send Queue Length Enquiry Messages more or less than necessary. Through the simulation experiments, we found that setting this threshold to twice the round trip end-to-end link delay (200ms in this case) between source router and victim router results in best performance.

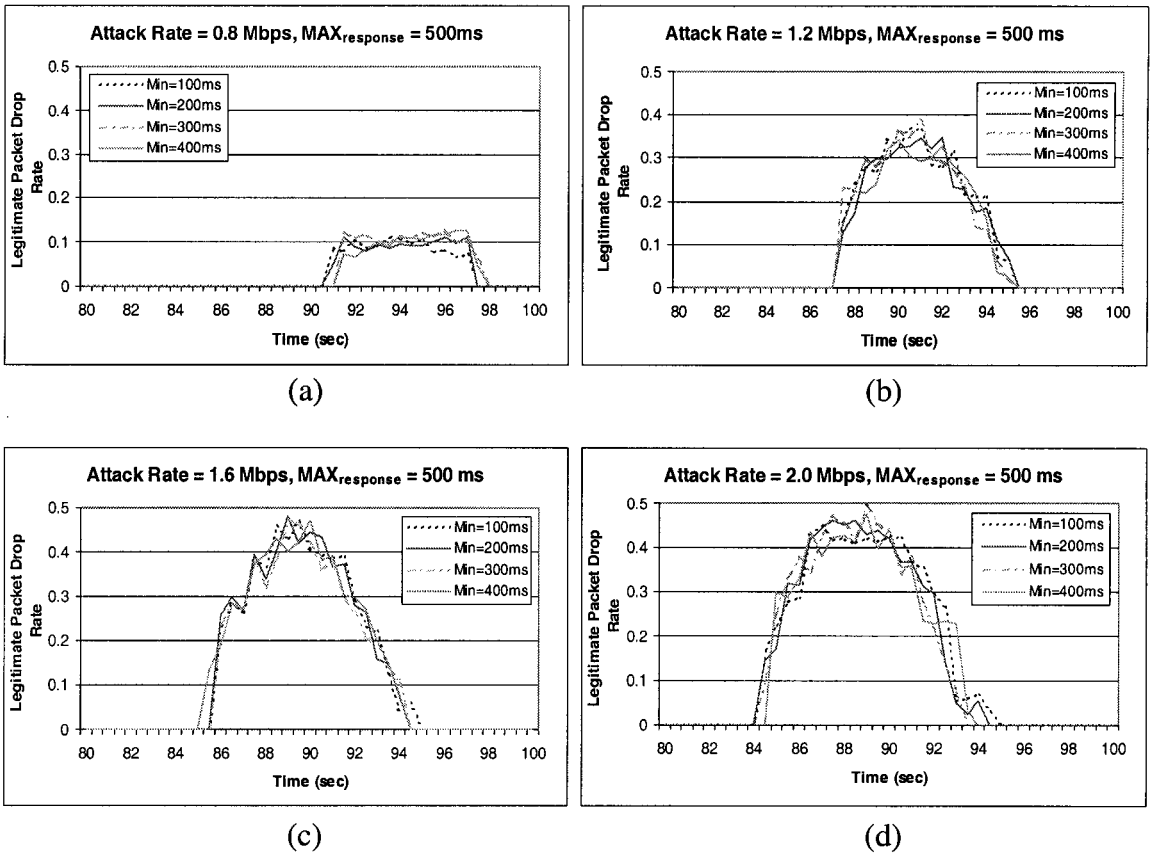

Figure 4.8: Good Packet Drop Rate at $Max_{response}$=500ms

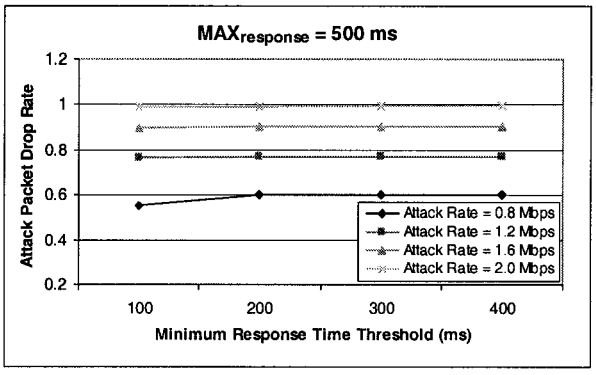

Figure 4.9: Attack Packet Drop Rate at $Max_{response}$=500ms

4.6.1.4 Effect of the Maximum Response Time Threshold

In this section, we set the minimum threshold to 200ms and study the effect of the maximum threshold on the performance of SRPD. Various values for maximum threshold are used ranging from 400ms to 900ms. The minimum threshold for

average response time has a basic range: (0, $Max_{response}$). The maximum threshold, however, has no upper limit. Theoretically, it can be set to any value in the range of ($Min_{response}$, ∞). Therefore, maximum threshold is set to a wide range to study its effect. The following table shows the parameters that are used in the experiments of examining maximum response time threshold's effect.

Bandwidth between victim router and victim server:	100Mbps
Propagation delay between source router and victim router:	100ms
High-rate identification threshold:	0.5Mbps
Average response time minimum threshold:	200ms
Average response time maximum threshold:	400~900ms
Victim router to transmission time for each packet:	0.4ms

Figures 4.10-4.14 show simulation results for different maximum values. Figure 4.10 and 4.11 show the legitimate packet drop rate with relatively smaller and larger threshold values, respectively. The effect of the maximum threshold is very minimal when its value is lower than 700ms. For threshold values larger than 700ms, there is noticeable service degradation as the threshold value increases. The good packets experience a much longer period of dropping. SRPD is able to control a DDoS attack in about 10 seconds with appropriate threshold settings (say 700ms). However, with improper parameters (high maximum threshold), it may not be able to control the DDoS attack.

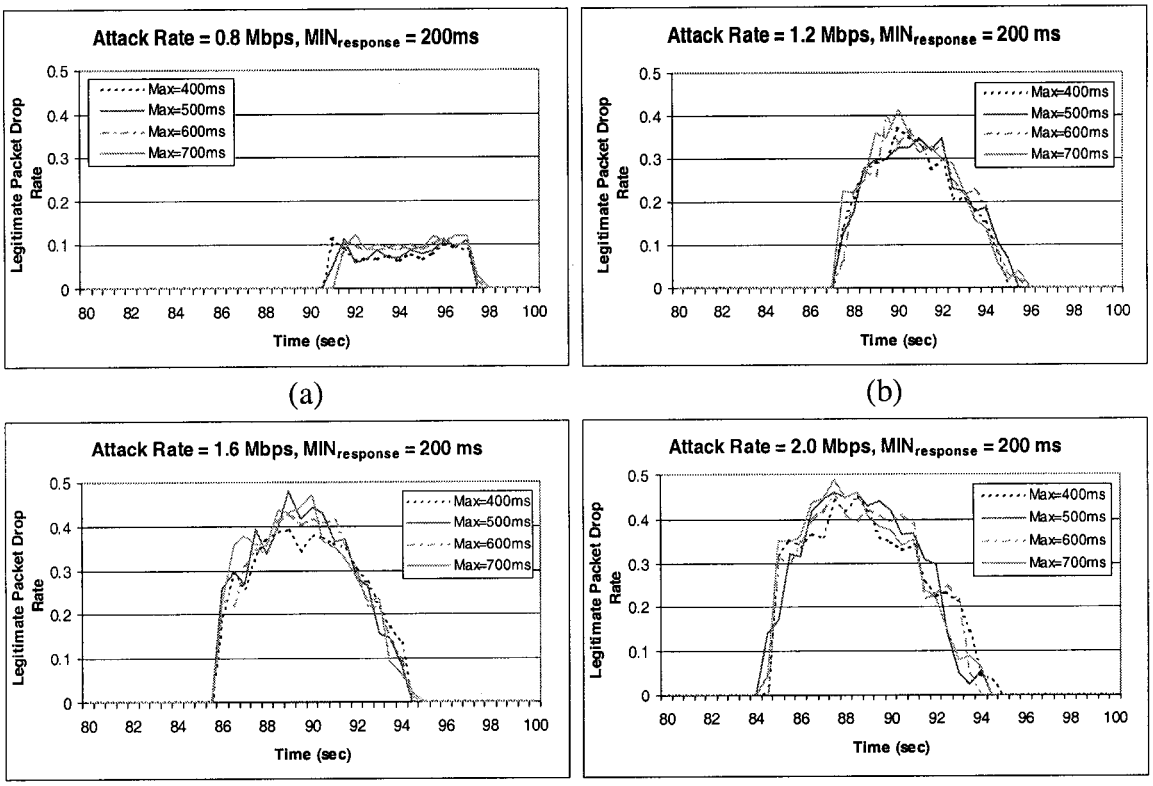

Figure 4.10: Legitimate Packet Drop Rate with Lower Max$_{response}$

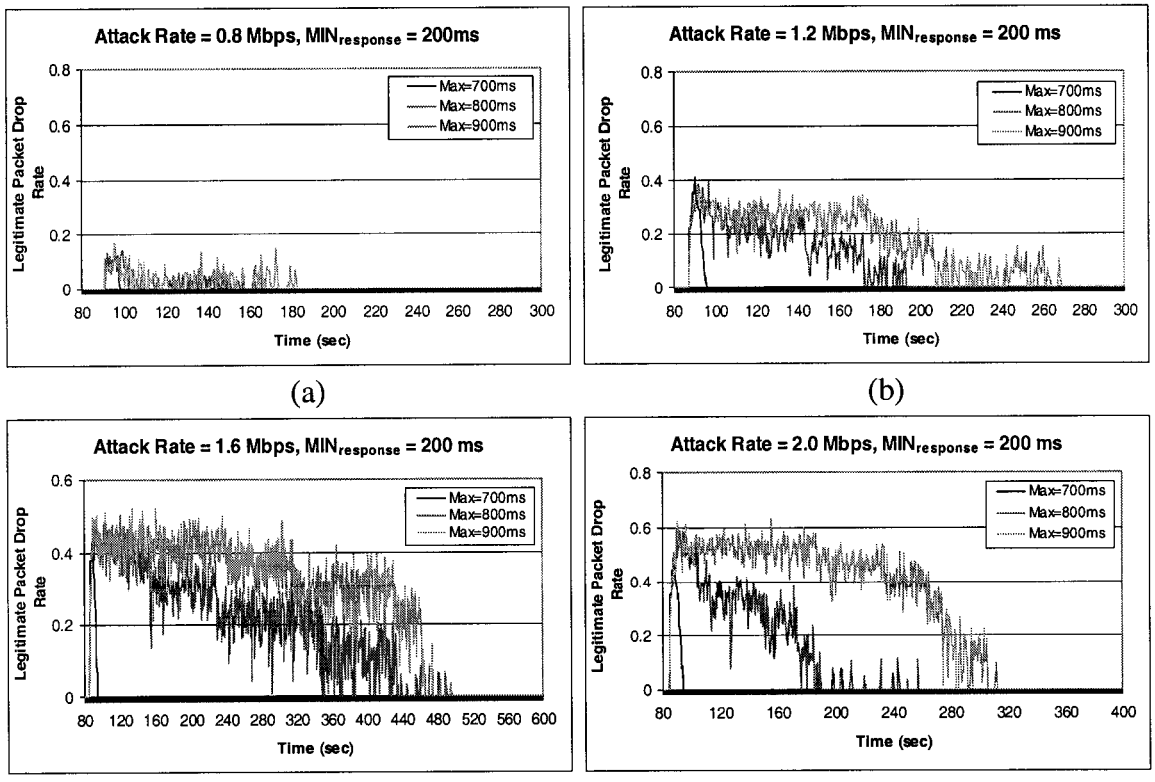

Figure 4.11: Legitimate Packet Drop Rate with Higher Max$_{response}$

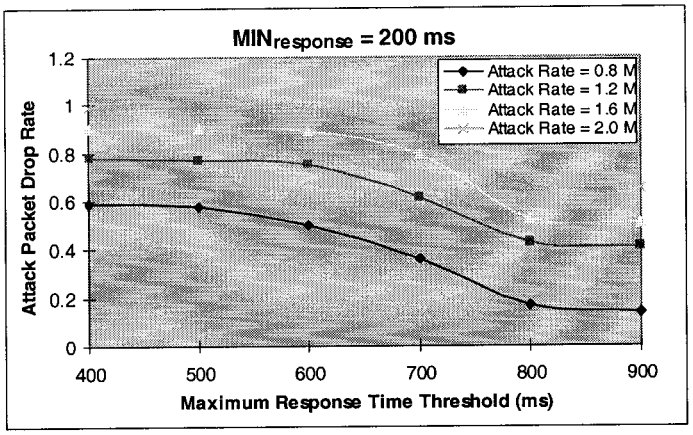

Figure 4.12: Attack Packet Drop Rate with Various $Max_{response}$

Figure 4.12 gives the attack packet drop rate with different maximum thresholds. The performance decreases to different degrees when the value of the maximum threshold increases. SRPD may not drop enough attack traffic to control an attack if the threshold is too large. Figures 4.13 and 4.14 show the average response time for various maximum thresholds. If the threshold is smaller than 700ms, there is little difference in the results. The average response time goes back to normal in about 10 seconds after the SRPD algorithm is activated. However, SRPD takes much longer time to reduce the response time to normal if the maximum threshold is too large. As Figure 4.14 shows that inappropriate maximum threshold settings can result in long average response time that may even be in minutes.

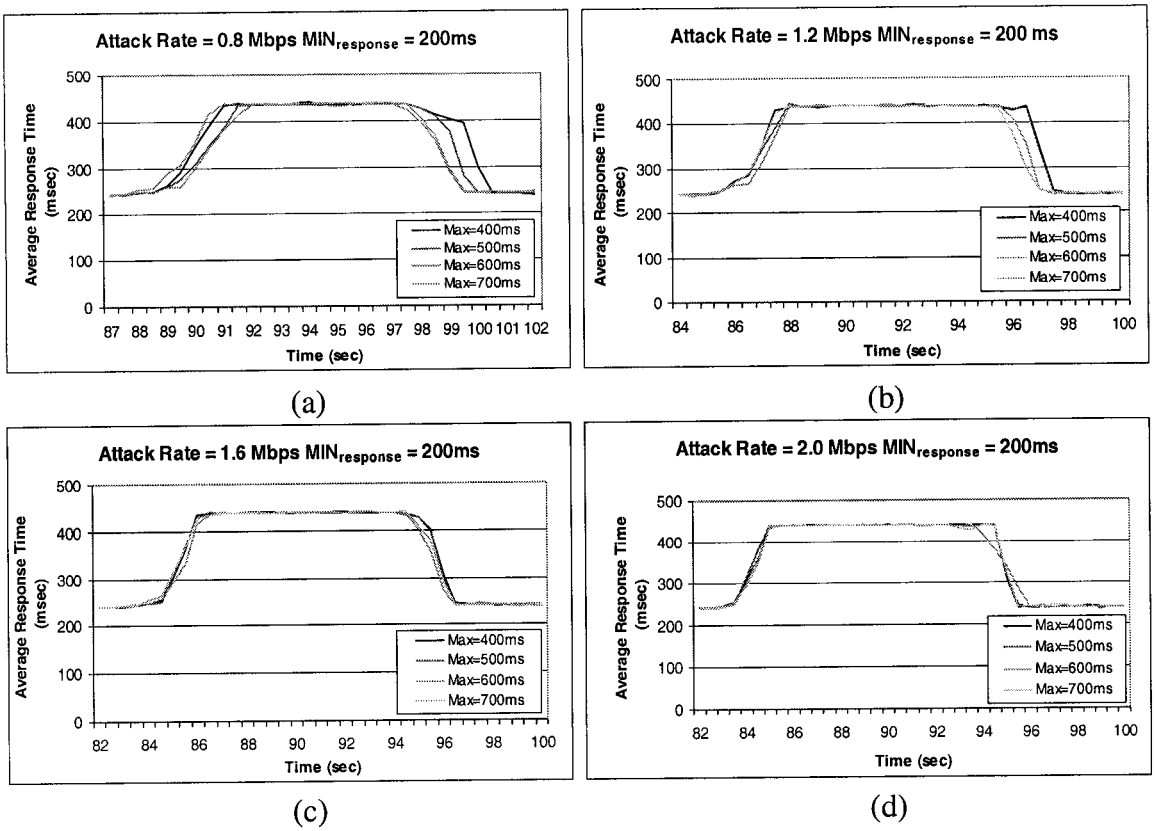

Figure 4.13: Average Response Time with Lower $Max_{response}$

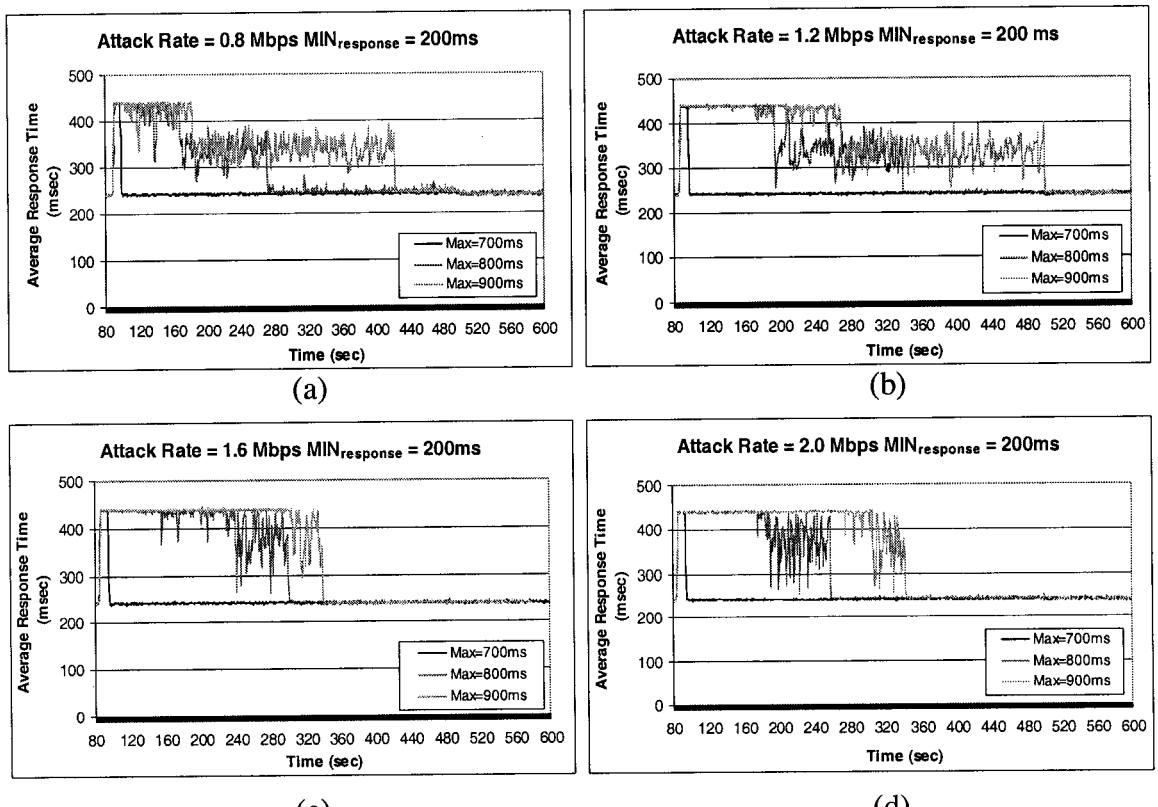

Figure 4.14: Average Response Time with Higher $Max_{response}$

Similar results are observed when the end-end-link link delay between source routers and the victim router is increased to 300ms and the minimum threshold for average response time is set to 600ms, see Figures 4.15 and 4.16.

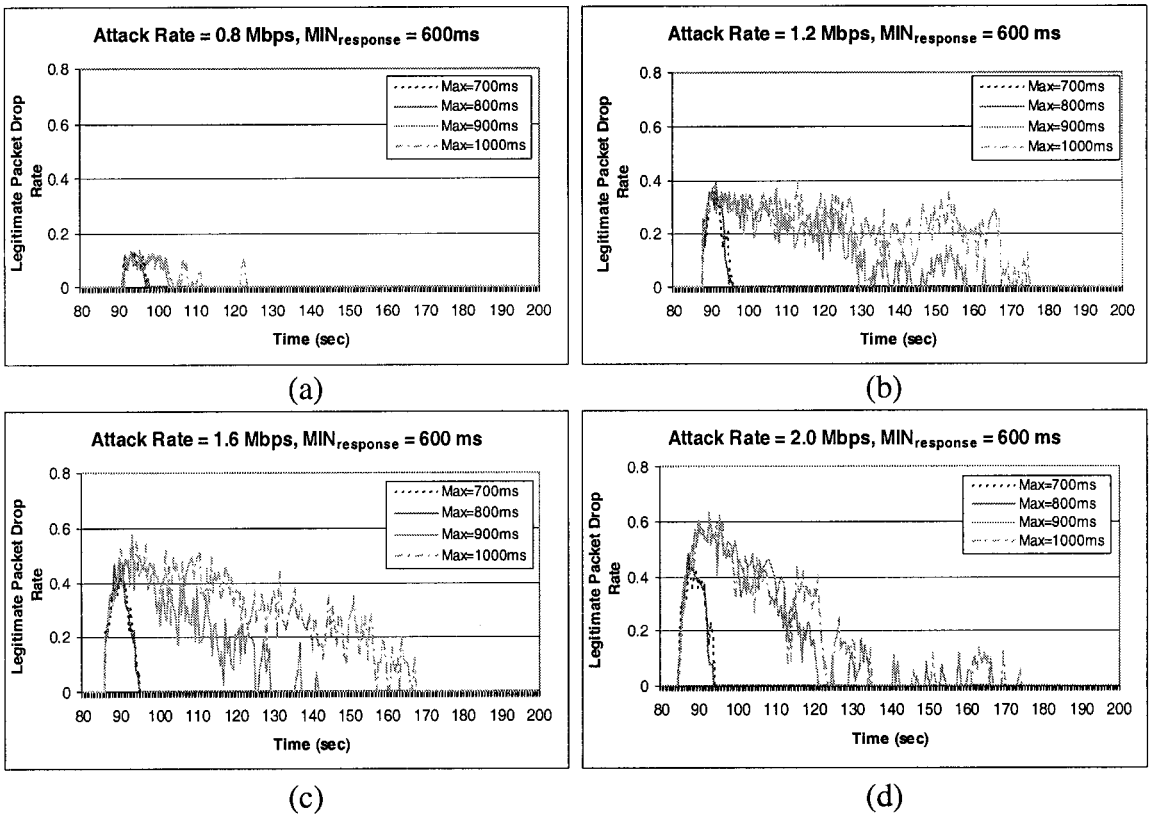

Figure 4.15: Legitimate Packet Drop Rate at $Min_{response}$=600ms

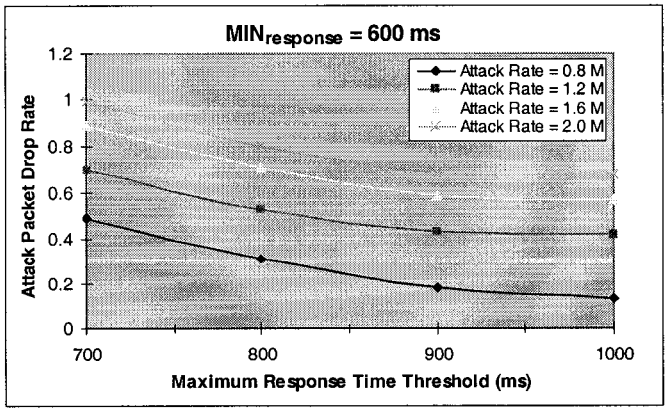

Figure 4.16: Attack Packet Drop Rate at $Min_{response}$=600ms

From our simulation experiments, we have found that the minimum threshold does not affect the performance as much as the maximum threshold. The minimum threshold for the average response time is suggested to be twice that of the round trip end-to-end link delay. The maximum threshold for the average response time is the maximum expected value for the response time that clients can endure in times of congestion. In chapter 3, the calculation of average response time was outlined. Regardless of the minor transmission and routing time at the victim router and the victim server, the average response time is mainly composed of twice the link delay, queuing delay at the victim router, and the service time at the victim server when the victim router is congested as following equation 4.1 shows:

$$T_{response} \approx 2*Latency_{sr-vr} + ST_{vs} + QT_{vr} \quad (4.1)$$

Since FIFO scheme is used at the output queue of the victim router, the queuing delay is equal to one packet transmission time multipled by the number of packets in the queue. Therefore, we have the following equation to describe the maximum response time:

$$Max_{response\_vr} \approx 2*Latency_{sr-vr} + aveST_{vs} + maxNum_{queue} *Trans_{vr} \quad (4.2)$$

where $aveST_{vs}$ is the average service time at the victim server and $maxNum_{queue}$ is the maximum allowable packet number in output queue at the victim router in times of congestion.

When the victim server is congested while the victim router is not, there is no extra queuing delay at the victim router. Then the maximum response time can be described as follows:

$$Max_{response\_vs} \approx 2*Latency_{sr\_vr} + maxST_{vs} \quad (4.3)$$

Where $maxST_{vs}$ is the maximum service time at the victim server. Theoretically, maximum threshold for average response time can be chosen as the larger value between (4.2) and (4.3) -- $Max\{Max_{response\_vr}, Max_{response\_vs}\}$. However, in practice, we may choose a smaller value than $Max\{Max_{response\_vr}, Max_{response\_vs}\}$ to limit the average response time under a modest DDoS attack, and hence we do not allow severe congestion.

4.6.2 Simulations of Flooding the Victim Server

In this section, we investigate the case where the victim server is overloaded. In this situation, both the victim router and the destination link have enough capacity to process and forward arrival packets. However, the victim server is not able to service incoming packets due to the lack of computing resources. The output queue length at the victim router is maintained normally low throughout the attack period. SRPD detects an ongoing attack by a persistent long response time.

To simulate this scenario, the bandwidth of the destination link between the victim router and the victim server is set to 100Mbps so that congestion only happens at the server. The link delay is 100ms. The minimum and maximum threshold for average

response time is set to 200ms and 500ms, respectively. The victim server can service no more than 500 concurrent requests at the same time. Note that in this case, the average response time is the only way for SRPD to detect an attack and from the previous experiments. The related parameter settings are listed bellow:

Bandwidth between victim router and victim server:	100Mbps
Propagation delay between source router and victim router:	100ms
High-rate identification threshold:	0.5Mbps
Average response time minimum threshold:	200ms
Average response time maximum threshold:	500ms
Victim router to transmission time for each packet:	0.04ms

Figure 4.17 and 4.18 show the good and bad packet drop rate under different attack intensities. With properly set parameters, good packets are only dropped during the first few seconds of the attack. More than 99.5% of attack packets can be discarded at their sources for different attack rates. Figure 4.19 demonstrates the workload on the victim server. Figure 4.20 illustrates the average response time. The workload on the server increases when a DDoS attack starts and it keeps high during the attack because the arrived requests have to stay in the system longer than usual due to the lack of CPU and memory resources, which results in long average response time as shown in Figure 4.20. Both the workload of the victim server and the average response time go back to normal level when the attack is under control and congestion is relieved.

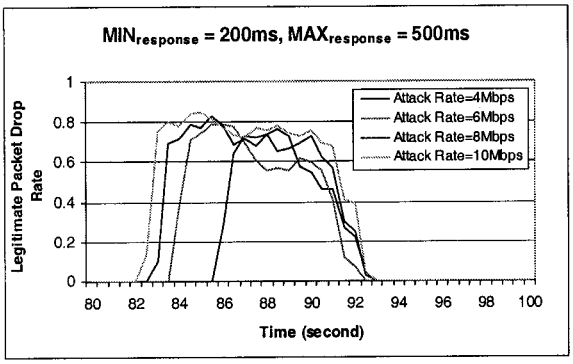

Figure 4.17: Legitimate Drop Rate

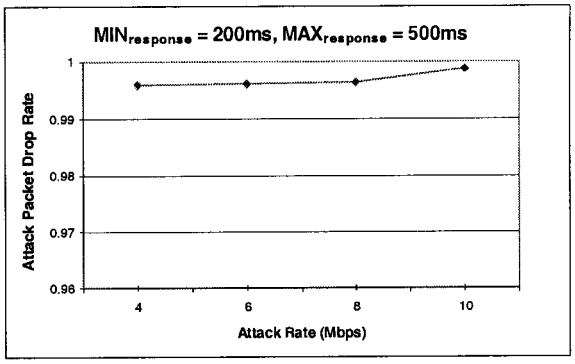

Figure 4.18: Attack Packet Drop Rate

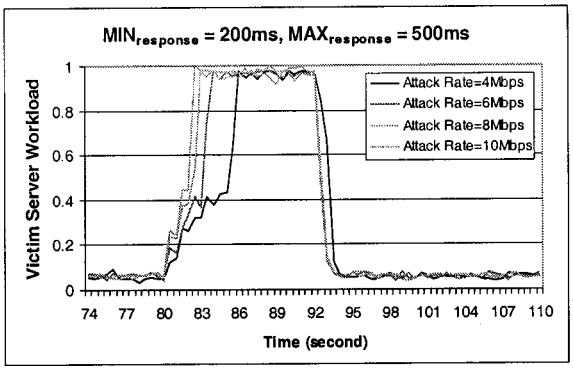

Figure 4.19: Victim Server Workload

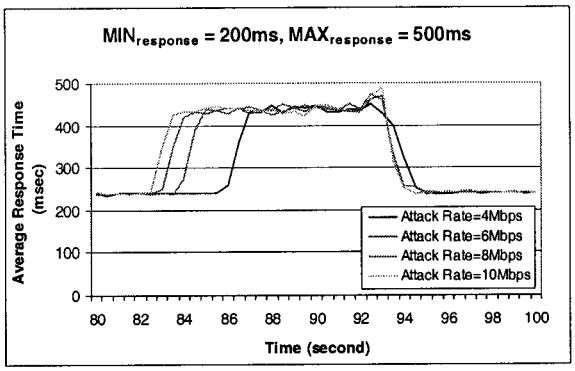

Figure 4.20: Average Response Time

4.7 Summary

This chapter examines and evaluates the performance of the proposed SRPD DDoS-attack-prevention scheme using simulation. The performance of SRPD is compared with the Pushback mechanism, which is currently considered one of the most promising mechanisms to defend against DDoS attacks. The network models of SRPD and Pushback, parameter settings, evaluation criteria, and simulation software implementation have been described. Two types of attack scenarios have been discussed -- the victim router is congested and the victim server is overloaded.

The simulation experiments show that SRPD outperforms Pushback in terms of collateral damage to legitimate traffic, attack packet drop rate, and the ability to reduce the queue length at the congested router. SRPD drops less good packets and more bad packets than Pushback. SRPD also reduces the victim router's queue length to normal faster than Pushback does. SRPD can also effectively handle the case in which the victim server is congested while the victim router is not. In this case, the victim router's queue length is maintained normally low. SRPD is still able to detect ongoing attack by persistent large average response times. Pushback, however, has no way to detect and manage this situation since there are no packet drops at the victim router.

The effects of different parameter settings in SRPD have been investigated as well. Different attack intensity, high-rate flow identification threshold, and minimum and maximum response time thresholds are used. Simulation results show that SRPD is parameter sensitive. SRPD responds to DDoS attacks faster but takes longer time to relieve congestion for higher attack intensities. For instance, and in a typical parameter setting SRPD takes 7 seconds to control an attack at the rate of 0.8 Mbps per source and 11 seconds to control an attack at the rate of 2.0 Mbps per source. Effectiveness of SRPD needs the cooperation of all source routers. The fewer the number of source routers involved, or the fewer the number of attack flows identified, the longer the time it takes SRPD to control an attack. The minimum threshold of the average response time does not significantly affect the performance of SRPD. However, SRPD is sensitive to the maximum threshold of the average response time.

The choice of the minimum and maximum thresholds for the average response time has been discussed in section 4.6.1.4. The minimum threshold should be set twice of the round trip end-to-end link delay between source routers and the victim router. Equations 4.2 and 4.3 provide a method to estimate the maximum average response time so that the maximum threshold can be set accordingly.

However, if an attack is well distributed, even though the parameters are properly set, SRPD routers may not be able to detect and control it. In that case, each attack flow coming out of daemon hosts could be even lower than or just like normal traffic flows. As long as a source router does not identify any high-rate flows, SRPD algorithm is not triggered. The attack is then not detected by SRPD. This is a shortcoming for SRPD. However, SRPD can be used together with other reactive mechanisms, like Pushback, to increase robustness. This is left for further study.

Chapter 5

Conclusion

5.1 Concluding Remarks

The currently vulnerable internetworking system and the growing number of Internet crimes inspire much research to increase Internet security and protect it from intrusions. The goal of our research is to investigate an effective approach to detect possible ongoing DDoS attacks and defend against them. The proposed Source Router Preferential Dropping (SRPD) scheme detects ongoing attacks at their sources by monitoring traffic arrival rate and response time and defeats them by preferentially dropping packets from high rate flows. It minimizes the collateral damage to legitimate traffic and more precisely identifies and discards attack traffic. SRPD can be considered as a proactive approach to stop DDoS attacks at their originating networks and to control them in their early stages.

SRPD uses a number of parameters to detect and control DDoS attacks. These parameters include a high-rate flow identification threshold, minimum and maximum average response time thresholds, victim server load and victim router queue length.

To achieve better performance, SRPD requires the cooperation of more source routers, and information exchange with the destination/victim routers.

A simulation model is constructed to study the performance of SRPD, and compare it with the Pushback approach. Simulation results show that SRPD outperforms Pushback in terms of legitimate packet drop rate, attack packet drop rate, and the output queue length at the congested victim router. SRPD shows more adaptability than Pushback in that it is able to handle not only the situation that the victim router is congested but also the situation that the victim server is congested.

The "distribution" attribute of DDoS attacks makes it very hard to arrive at a complete solution for defeating DDoS attacks. Till now, there is no perfect approach that can deal with all types of DDoS attacks, has no adverse effects on legitimate traffic, and identifies and drops all attack packets. Several properties of SRPD make it a promising and a feasible scheme for defeating DDoS attacks. The most attractive feature is that it reduces the collateral damage to a very low level. As well, SRPD only needs to be deployed at edge routers close to a client's network. Core routers in backbone networks need not be involved. This feature makes it more practical than Pushback, which needs the collaboration of all routers along the path from the victim network to the source network.

5.2 Discussion and Future Work

There are a few aspects that need future discussion and some implementation and operational issues need to be addressed before SRPD could be more robust in combating against DDoS attacks and deployed in the Internet.

SRPD trusts every packet that passes the preprocessing "egress filtering" procedure. This brings vulnerability. In the case of VLAN networks, two machines in different VLANs can share a same physical interface at their edge router. Figure 5.1 gives an example of this situation. Machine A and machine B are located in two different VLANs but share a same physical interface with separated port configured at their edge router. Machine A could send out packets with their source IP address set to B. Then the edge router is not able to block this type of traffic by egress filtering or even by "Unicast Reverse Path Forwarding" [12] mentioned in chapter 2. In this situation, SRPD algorithm's ability to identify attack flows may decline. Fortunately, if machine A sends attack traffic using B's source IP address constantly during the whole period of the attack, SRPD can still recognize this flow.

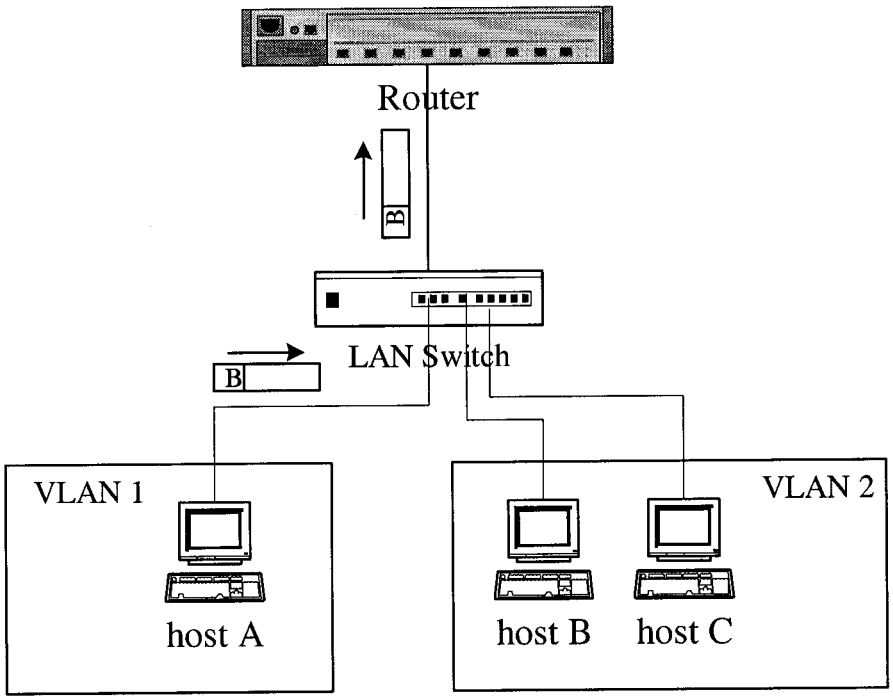

Figure 5.1: Same Physical Interface Sharing

SRPD is less effective in identifying attack traffic when the attack is well distributed. If there are hundreds of thousands of attack sources involved in a single attack, each attack source does not have to generate a high-rate flow and aggregate still can overwhelm a victim machine. It is possible that an attack flow's rate is comparable to a normal flow's rate in this case. SRPD then may punish good traffic as well. In fact, no existing approaches can effectively deal with well-distributed attacks.

In SRPD, source routers send Queue Length Enquiry Messages to victim routers asking about the congestion situation when they are not sure whether congestion is going on at the victim routers. However, attackers may generate ICMP response messages themselves. It is possible that attackers generate Queue Length Response Messages with a very small value set in the packets' payload, representing a low

output queue occupancy rate at the victim router. Source routers do not drop packets if they are given information that there is no congestion at the victim router. Although source routers still start probabilistically dropping when they detect a long lasting average response time, SRPD responds to the attack more slowly than it should and performance is then reduced. Therefore, some form of authentication is desired. Assigning higher priority for this type of ICMP messages can also improve the reliability of the message transmission.

There are other approaches that can be further implemented at source routers to increase their confidence of dropping ongoing attack traffic. Gathering attack history information may help source routers to make more precise decisions of packet dropping. A source router can increase the packet drop probability if the detected attack flows are recorded in its attack flow membership table.

Bibliography

[1] CERT Coordination Center, "Denial of Service Attacks," http://www.cert.org/tech_tips/denial_of_service.htm

[2] CERT Coordination Center, "Trends in Denial of Service Attack Technology," October 2001.

[3] D. Moore, G. M. Voelker, and S. Savage, "Inferring Internet Denial-of-Service Activity," In *Proceedings of 10th USENIX Security Symposium*, August 2001.

[4] D. Dittrich, "The DoS Project's "trinoo" distributed denial of service attack tool," Technical Report, University of Washington, October 1999. http://staff.washington.edu/dittrich/misc/trinoo.analysis

[5] D. Dittrich, "The "Tribe Flood Network" distributed denial of service attack tool," Technical Report, University of Washington, October 1999. http://staff.washington.edu/dittrich/misc/stacheldraht.analysis

[6] D. Dittrich, "The "stacheldraht" distributed denial of service attack tool," Technical Report, University of Washington, December 1999. http://staff.washington.edu/dittrich/misc/stacheldraht.analysis

[7] J. Mirkovic, J. Martin, and P. Reiher, "A Taxnomy of DDoS Attacks and DDoS Defense Mechansims," UCLA Technical Report #020018, 2002.

[8] CERT Coordination Center, "TCP SYN flooding and IP spoofing attacks," http://www.cert.org/advisories/CA-1996-21.html

[9] CERT Coordination Center, "Smurf IP Denial-of-Service Attacks," http://www.cert.org/advisories/CA-1998-01.html

[10] CERT Coordination Center, "Denial of Service Attacks using Name servers," http://www.cert.org/incident_notes/IN-2000-04.html

[11] P. Ferguson and D. Senie, "Network ingress filtering: Defeating denial of service attacks which employ IP source address spoofing," RFC 2267, January 1998.

[12] Cisco Web Page, "Strategies to Protect Against Distributed Denial of Service (DDoS) Attacks," February 2000. http://www.cisco.com/warp/public/707/newsflash.html

[13] H. Wang, D. Zhang, K. G. Shin, "Detecting SYN Flooding Attacks," In *Proceedings of INFOCOM*, June 2002.

[14] C. Schuba, I. Krsul, M. Kuhn, E. Spafford, A. Sundaram and D. Zamboni, "Analysis of a Denial of Service Attack on TCP," In *Proceedings of the 1997 IEEE Symposium on Security and Privacy, pp208 –223*, May 1997.

[15] J. Lemon, "Resisting SYN Flooding DoS Attacks with a SYN Cache," In *Proceedings of USENIX BSDCon'2002*, February 2002.

[16] D. J. Bernstein and E. Schenk, "Linux Kernel SYN Cookies Firewall Project," http://www.bronzesoft.org/projects/scfw

[17] Check Point Software Technologies Ltd. "TCP SYN Flooding Attack and the FireWall-1 SYNDefender, October 1996.

[18] Netscreen 100 Firewall Appliance, http://www.netscreen.com/

[19] Tripwire, "Tripwire for servers," http://www.tripwire.com/products/servers/

[20] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, "Practical network support for IP Traceback," In *Proceedings of ACM SIGCOMM Conference*, August 2000.

[21] S. M. Bellovin, "ICMP traceback messages," Internet draft: draft-bellovin-itrace-00.txt, March 2000.

[22] D. X. Song and A. Perrig, "Advanced and authenticated marking scheme for IP Traceback," In *Proceedings of IEEE INFOCOM*, 2001.

[23] D. Dean, M. Franklin and A. Stubblefield, "An algebraic approach to IP Traceback," In *Proceedings of the 2001 Network and Distributed Sysmte Security Symposium*, February 2001.

[24] A.C. Snoeren, C. Partridge, L. A. Sanchez, C. E. Jones, F. Tchakountio, S. T. Kent, W. T. Strayer, "Hash-Based IP Traceback," In *Proceedings of ACM SIGCOMM 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, August 2001.

[25] R. Stone, "CenterTrack: An IP Overlay Network for Tracking DoS Floods," In *Proceedings of 9th USENIX Security Symposium*, August 2000.

[26] K. Park and H. Lee, "On the effectiveness of probabilistic packet marking for IP traceback under denial of service attack," In *Proceedings of IEEE INFOCOM*, 2001.

[27] J. Li, J. Mirkovic, M. Wang, P. Reiher and L. Zhang, "SAVE: Source address validity enforcement protocol," In *Proceedings of INFOCOM*, June 2002.

[28] T. M. Gil and M. Poleto, "MULTOPS: a data-structure for bandwidth attack detection," In *Proceedings of 10th USENIX Security Symposium*, August 2001.

[29] S. Floyd, S. Bellovin, J. Ioannidis, K. Kompella, R. Mahajan and V. Paxson, "Pushback Messages for Controlling aggregates in the Network," Internet draft: draft-floyd-pushback-messages-00.txt, July 2001.

[30] D. K. Y. Yau, J. C.S. Lui, and F. Liang, "Defending Against Distributed Denial-of-Service Attacks with Max-Min Fair Server-centric Router Throttles," In *Proceedings of IEEE International Workshop on Quality of Service, pp. 35-42,* 2002.

[31] D. Dittrich, S. Dietrich, and N. Long, "An analysis of the 'Shaft' distributed denial of service tool," March 2000.
http://home.adelphi.edu/~spock/shaft_analysis.txt

[32] J. Postel, "Internet Control Message Protocol – DARPA Internet Program Protocol Specification," Request for Comments 792, September 1981.

[33] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Transaction on Networking, Vol. 1(4): 397-413,* August 1993.

[34] S. Floyd and K. Fall, "Promoting the Use of End-to-End Congestion Control in the Internet," *IEEE/ACM Transactions on Networking, Vol. 7(4): pp.458-473,* August 1999.

[35] J. Ioannidis and S. Bellovin, "Implementing Pushback: Router-Based Defense Against DDoS Attacks," In *Proceedings of the Network and Distributed System Security Symposium*, February 2002.

[36] J. Postel, "Transmission Control Protocol – DARPA Internet Program Protocol Specification," Request for Comments 793, September 1981.

[37] J. Postel, "User Datagram Protocol," Request for Comments 768, August 1980.

[38] H. L. Flanagan, "Egress Filtering – Keeping the Internet Safe from Your Systems," April 2001.
http://www.sans.org/rr/sysadmin/egress.php

[39] S. Floyd, K. Fall and K. Tieu, "Estimating Arrival Rates from the RED Packet Drop History," unpublished, August 1998.
www.aciri.org/floyd/end2end-paper.html

[40] F. Kargl, J. Maier, and M. Weber, "Protecting Web Servers from Distributed Denial of Service Attacks," In *Proceedings of the Tenth International Conference on World Wide Web, pp. 514-523*, May 2001.

[41] O. Spatscheck, "Defending Against Denial of Service Attacks in Scout," In *Proceedings of the 3rd Symposium on Operating Systems Design and Implementation*, February 1999.

[42] D. Schnackenberg, K. Djahandari, and D. Sterne, "Infrastructure for Intrusion Detection and Response," In *Proceedings of DISCEX*, January 2000.

[43] J. Mirkovic, J. Martin, and P. Reiher, "A Taxonomy of DDoS Attacks and DDoS Defense Mechanisms," UCLA Technical Report #020018, 2002.

[44] F. Lau, S. H. Rubin, M. H. Smith, and L. Trajkovic, "Distributed Denial of Service Attacks," In *Proceedings of IEEE GlOBECOM Conference, Volume 3, pp. 2275-2280*, 2000.

[45] A. Piskozub, "Denial of service and distributed denial of service attacks," *TCSET, pp303-304*, February 2002.

[46] R. K. C. Chang, "Defending against Flooding-Based Distributed Denial-of-Service Attacks: A Tutorial," *IEEE Communications Magazine, pp42-51*, October 2002.

Appendix A

Flow Charts of Main Methods in SRPD

In our simulations, there are several important methods running on source routers to process different types of packets. Three of them are illustrated in the Figure A.1, A.2 and A.3, which represent the key procedures of SRPD algorithm. Basically, there are three types of packets at source routers:

1. Request packets coming out from clients in source networks;
2. Their corresponding reply packets coming back from the global Internet;
3. Queue Length Response Messages from victim routers.

SRPD source routers first identify the type of incoming packets and then manipulate them with different operations accordingly. When a SRPD source router receives a packet coming from its inside client, it checks whether the packet's source IP address is unauthorized. If it is from an authorized address, it also checks whether the packet belongs to a suspicious flow (a flow with its state set other than "Unmonitored"). Further actions are manipulated based on these attributes of the packet. Figure A.1 describes the detailed procedure that how SRPD source routers process packets from inside clients.

Figure A.2 and A.3 demonstrate SRPD source router's procedures of processing reply packets originated from the victim server and ICMP Queue Length Response Messages from the victim router, respectively. The method shown in Figure A.2 describes that when a source router receives a reply packet from outside Internet, it updates the average response time of the corresponding flow and changes the flow's state based on the range of the average response time and sends Queue Length Enquiry Messages to the victim router if the average response time is located in a specified range. The method displayed in Figure A.3 tells that when a source router receives an ICMP response message from the victim router informing it about the output queue length occupancy rate, it adjusts the packet drop probability accordingly.

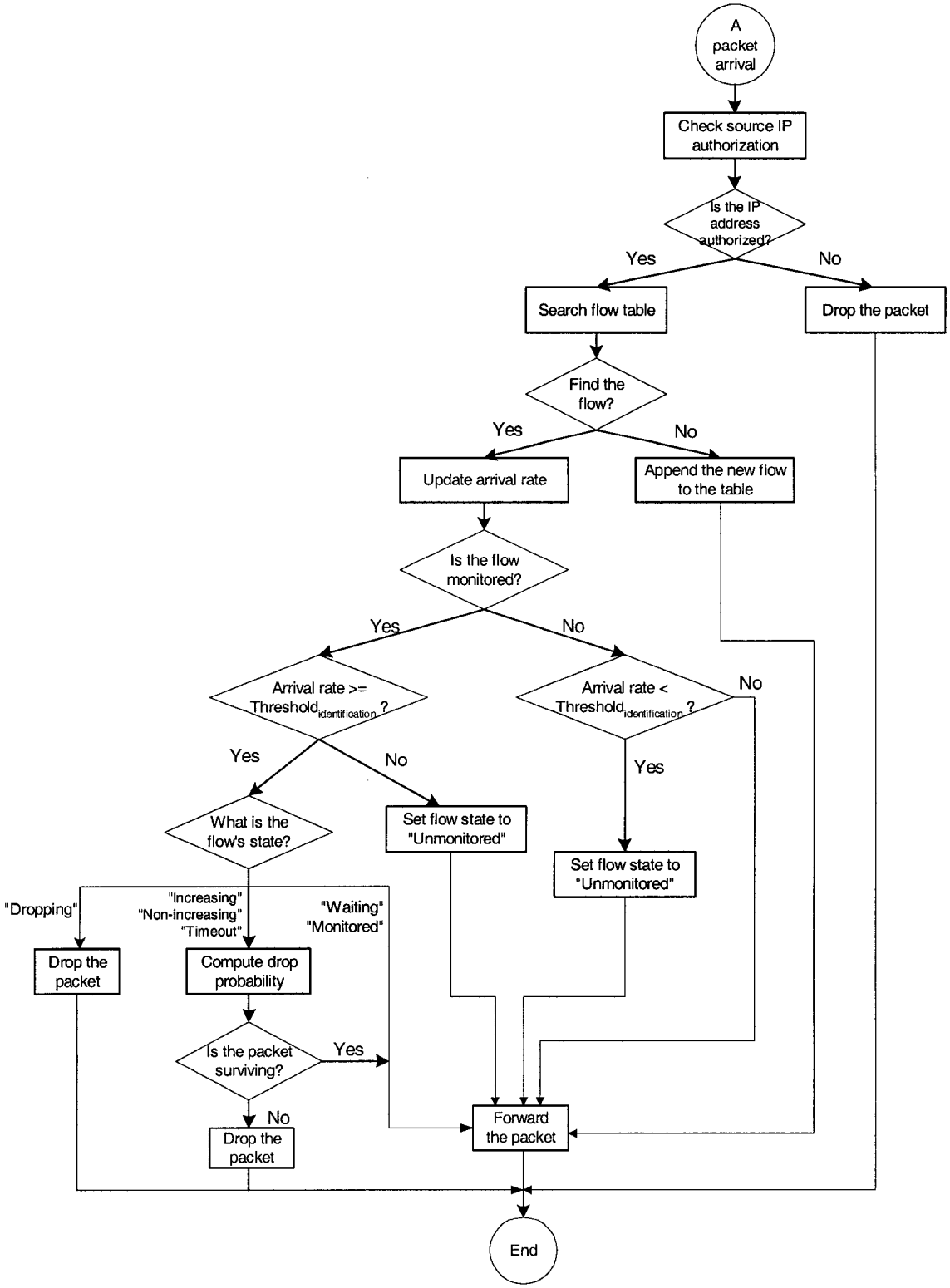

Figure A.1: Method of Processing Arrival Packets from Inside Clients

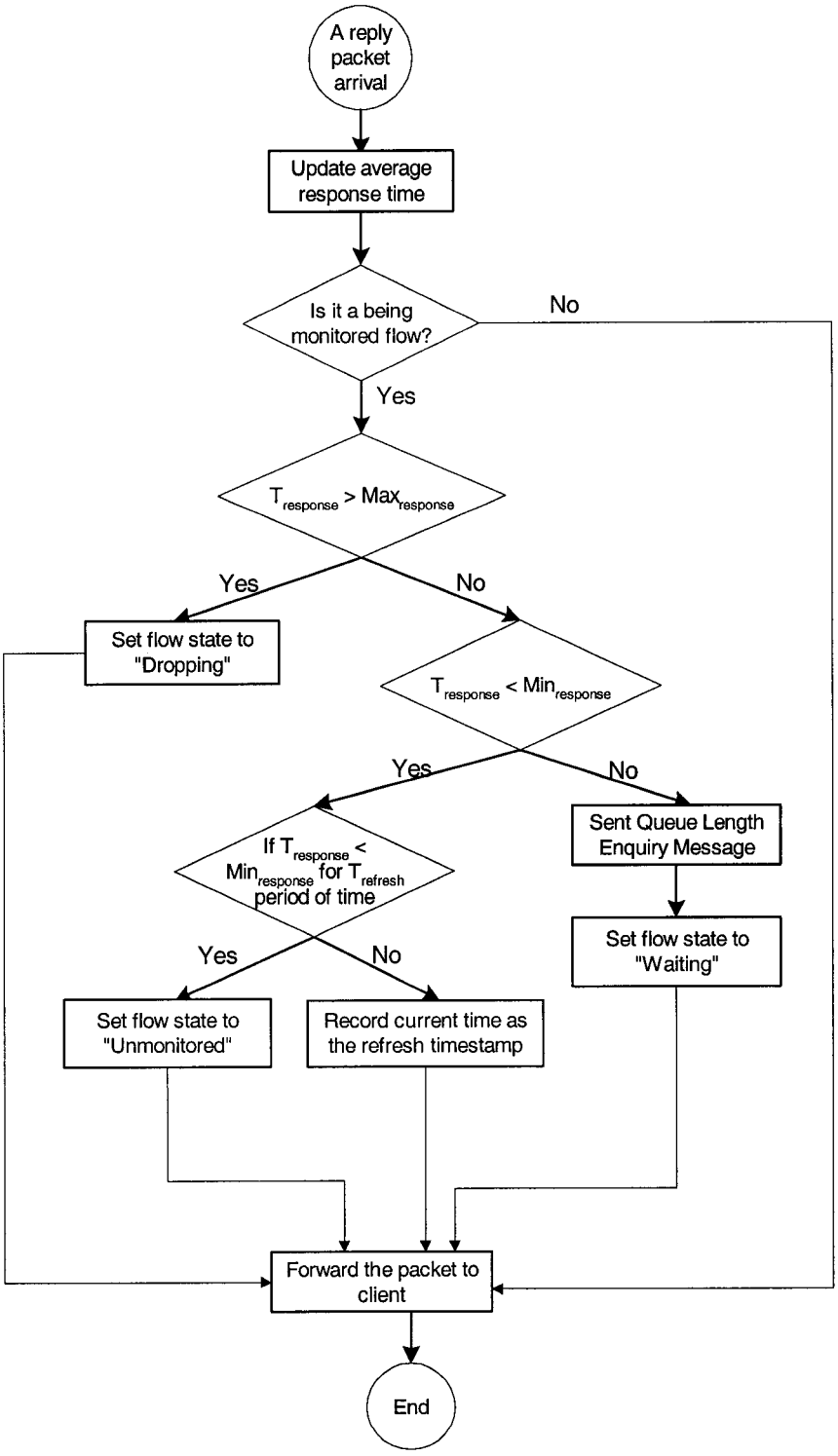

Figure A.2: Method of Processing Reply Packets from Global Internet

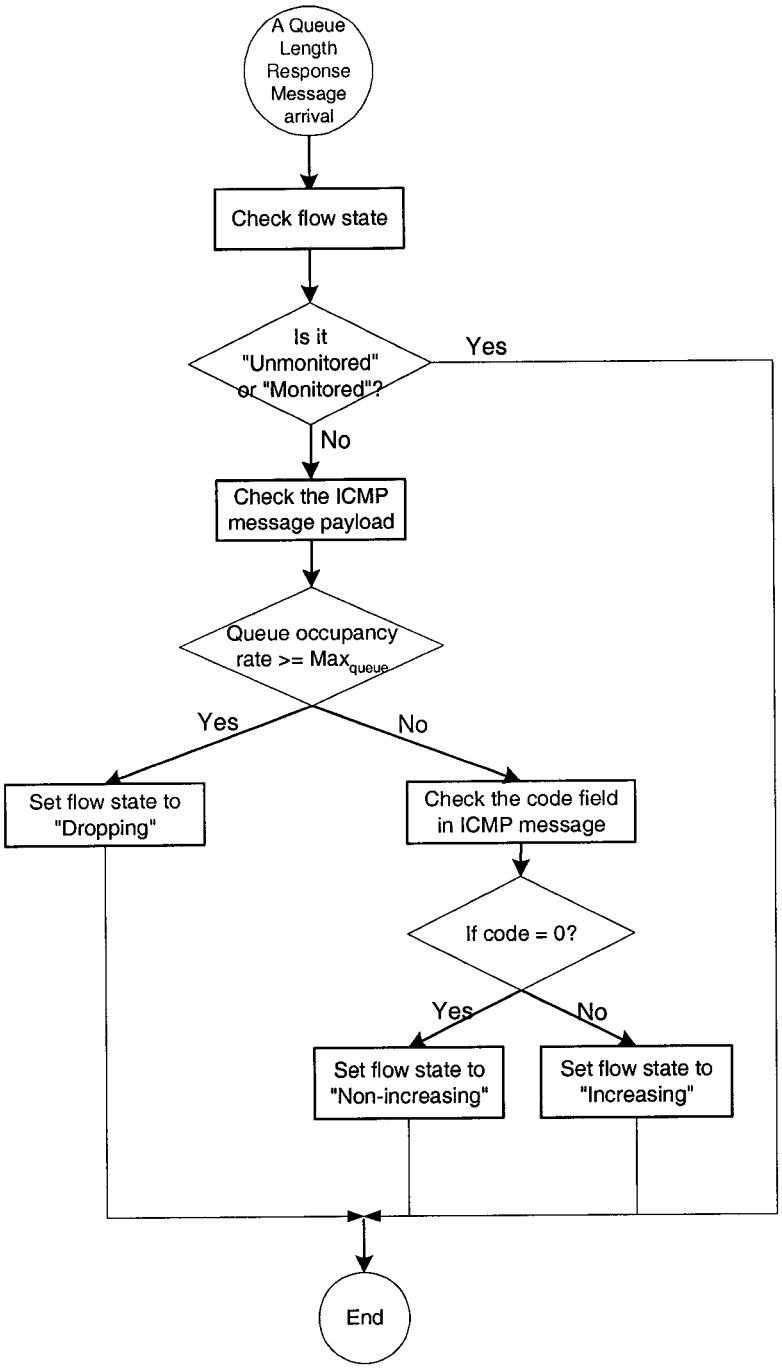

Figure A.3: Method of Processing ICMP Response Messages from Victim Routers

Appendix B

Confidence Intervals

Normally, confidence intervals placed on the mean values of simulation results can be used to describe the accuracy of the simulation results. Consider the results of N statistically independent simulation runs for the same experiment: X_1, X_2, ..., X_N. The sample mean, $\overline{X}$ is given as:

$$\overline{X} = \frac{\sum_{i=1}^{N} X_i}{N}$$

The variance of the distribution of the sample values, S_x^2 is:

$$S_x^2 = \frac{\sum_{i=1}^{N} (X_i - \overline{X})^2}{N-1}$$

The standard derivation of the sample mean is given by: $\frac{S_x}{\sqrt{N}}$.

Under the assumption of independence and normality, the sample mean is distributed in accordance to the T-Distribution, which means the sample mean of the simulation runs fall in the interval $\pm\varepsilon$ within the actual mean with a certain probability drawn from the T-Distribution.

$$\varepsilon = \frac{S_x t_{\alpha/2,N-1}}{\sqrt{N}}$$

where $t_{\alpha/2,N-1}$ is the value of the T-distribution with N-1 degrees of freedom with probability $\alpha/2$.

The upper and lower limits of the confidence interval regarding the simulation results are:

$$\text{Lower Limit} = \overline{X} - \frac{S_x t_{\alpha/2,N-1}}{\sqrt{N}}$$

$$\text{Upper Limit} = \overline{X} + \frac{S_x t_{\alpha/2,N-1}}{\sqrt{N}}$$