A SPACE-EFFICIENT COOPERATIVE METHOD FOR DETECTING DDoS ATTACKS

A DISSERTATION

Submitted in partial fulfillment of the requirements for the award of the degree of

MASTER OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

By

NEHA GUPTA

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE -247 667 (INDIA)

JUNE, 2010

CANDIDATE'S DECLARATION

I hereby declare that the work, which is being presented in the dissertation entitled "**A SPACE-EFFICIENT COOPERATIVE METHOD FOR DETECTING DDoS ATTACKS**" towards the partial fulfillment of the requirement for the award of the degree of **Master of Technology** in **Information Technology** submitted in the Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, Roorkee (India) is an authentic record of my own work carried out during the period from July 2009 to June 2010, under the guidance of **Dr. Manoj Mishra, Professor, Department of Electronics and Computer Engineering, IIT Roorkee.**

I have not submitted the matter embodied in this dissertation for the award of any other degree or diploma.

Neha Gupta

Date: 13-6-10 **(NEHA GUPTA)**

Place: Roorkee

CERTIFICATE

This is to certify that the above statement made by the candidate is correct to the best of my knowledge and belief.

Date: 13-6-10

Manoj Misra

Place: Roorkee **(Dr.MANOJ MISHRA)**

Professor

Department of Electronics and Computer Engineering

IIT Roorkee – 247 667

ACKNOWLEDGEMENTS

I would like to take this opportunity to extend my heartfelt gratitude to my guide and mentor **Dr. MANOJ MISHRA**, Professor, Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, for his trust in my work, his able guidance, regular source of encouragement and assistance throughout this dissertation work. I would state that the dissertation work would not have been in the present shape without his inspirational support and I consider myself fortunate to have done my dissertation under him.

I also thanks to **Dr. S. N. Sinha**, Professor and Head of the Department of Electronics and Computer Engineering for providing facilities, inspiration and support for the work.

Finally, I would like to say that I am indebted to my parents for everything that they have done for me. All of this would have been impossible without their constant support. And I also thank to God for being kind to me and driving me through this journey.

Neha Gupta

NEHA GUPTA

ABSTRACT

Today, Internet is the prime medium for communication which is used by the number of users across the globe. At the same time, its commercial nature is causing increasing vulnerability to cyber crimes and there has been an enormous increase in the number of DDoS attacks on the Internet over the past decade. Network resources such as network bandwidth, web servers and network switches are mostly the victims of DDoS attacks.

Current Internet architecture allows the attacker to spoof the source address of the IP packet by rewriting the packet header. This gives provision to conceal the identity of the source of attack. IP spoofing is the most popular form of Distributed Denial of Service attack. A large number of schemes have been proposed and implemented for the defense against DDoS attacks. Some defend the attack by filtering and dropping packets and some defend the attack by tracing back to the source of attack after experiencing it. Both mechanisms have their own drawbacks. However these schemes require a large amount of space and do not use cooperation.

In this dissertation "*A Space-Efficient Cooperative Method for Detecting DDoS Attacks*", we proposed a space-efficient cooperative scheme which produces warning of the DDoS attack at an early stage. It uses a Bloom filter-based detection scheme to generate accurate detection results and at the same time consumes less space and computational resources.

The proposed scheme has been simulated using NS-2 (Network Simulator) on a Linux platform. Various test cases have been designed, for which simulations were performed by varying different parameters. The comparison of results with the existing schemes has overcome with some of the limitations like space, hash collisions etc.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 Introduction

Communication is the key in changing the way the world looks, thinks, and works. Hardwired telex and telephone were the only source for communication before computers. But now Internet has revolutionized the computer and communications world. Defense Advanced Research Projects Agency (DARPA) introduced the Advanced Research Projects Agency Network (ARPANET) to provide convenient sharing of specialized computing resources across different defense institutions. A significant landmark was the introduction of TCP/IP as a set of protocols for internetworking which created the nucleus of the Internet [1]. Internet was commercialized as the TCP/IP standard was adopted by hardware and software vendors allowing organizations to interconnect heterogeneous systems. The mass production of Internet-capable personal computers and an unprecedented growth in the number of Internet Service Providers (ISP) enabled Internet to be commonly accessible to everyone.

Today, the Internet is an essential part of our everyday life and many important and crucial services like banking, shopping, transport, health, and communication are partly or completely dependent on the Internet. Similarly it has also enhanced the criminal's ability to perform unlawful or unethical activities including attacks on servers and other Internet devices. One of the most common is the TCP SYN flood attacks. Recent trends in the Internet [2, 3] show that, at some point the Web sites were getting up to 50,000 fake hits per second from illegitimate machines and the total amount of the DDoS attacks reached over 40 gigabit per second barrier. It also shows that the amount of DDoS attack traffic has been increasing in size (doubled) year by year. The Figure 1.1 shows the year wise pattern of attack traffic. It shows the Size of the Internet users in the world by various Geographic Regions. This is the recent information according to the survey of Mini Watts Marketing Group [6]. According to this survey, the estimated Internet users are 1,802,330,457 for December 31st 2009.

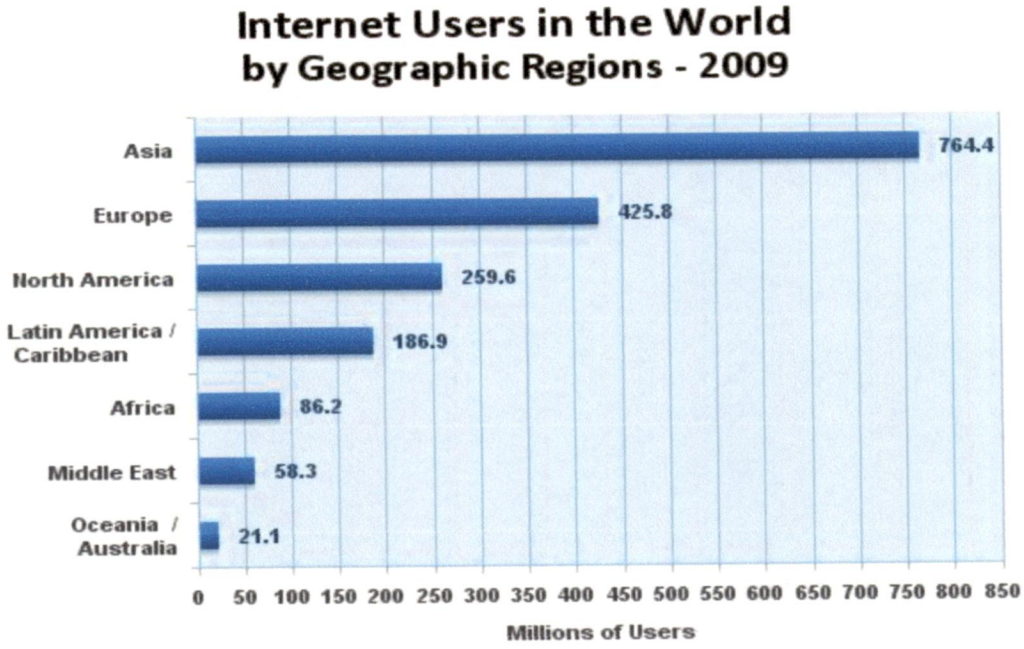

Figure1.1: Estimated Internet Users in the World by Geographic regions according the survey of Mini Watts Marketing Group

An attempt to make a computer resource or a service unavailable to its intended users is called as **Denial of Service (DoS)**. Denial of Service in its distributed form is called Distributed Denial of Service (DDoS). In DDoS attacks, the attacker first takes control of a large number of vulnerable hosts on the internet by compromising them. The attacker then uses those hosts to simultaneously send a huge number of packets to the victim, thereby exhausting all of the victim's resources. During DDoS attack, massive amounts of traffic arrive at the target of attack (i.e., victim). This target is either the network service or the network itself. Due to the huge amount of traffic, the computational overhead increases on the victim and the victim services get disrupted. The main purpose of the DDoS attacks is to reduce or eliminate the availability of a service provided over the Internet, to its legitimate users. As Figure 1.2 illustrates an example of how DDoS attack happens.

Current Internet architecture allows the attacker to spoof source address of the IP packet by rewriting the packet header. This process is called IP spoofing and it gives the provision to conceal the identity of the source of attack. IP spoofing is usually employed in conjunction with DDoS attacks in the Internet. In present Internet environment DDoS

attack [4, 5] is a serious security problem and it causes severe damages on the targeted servers.

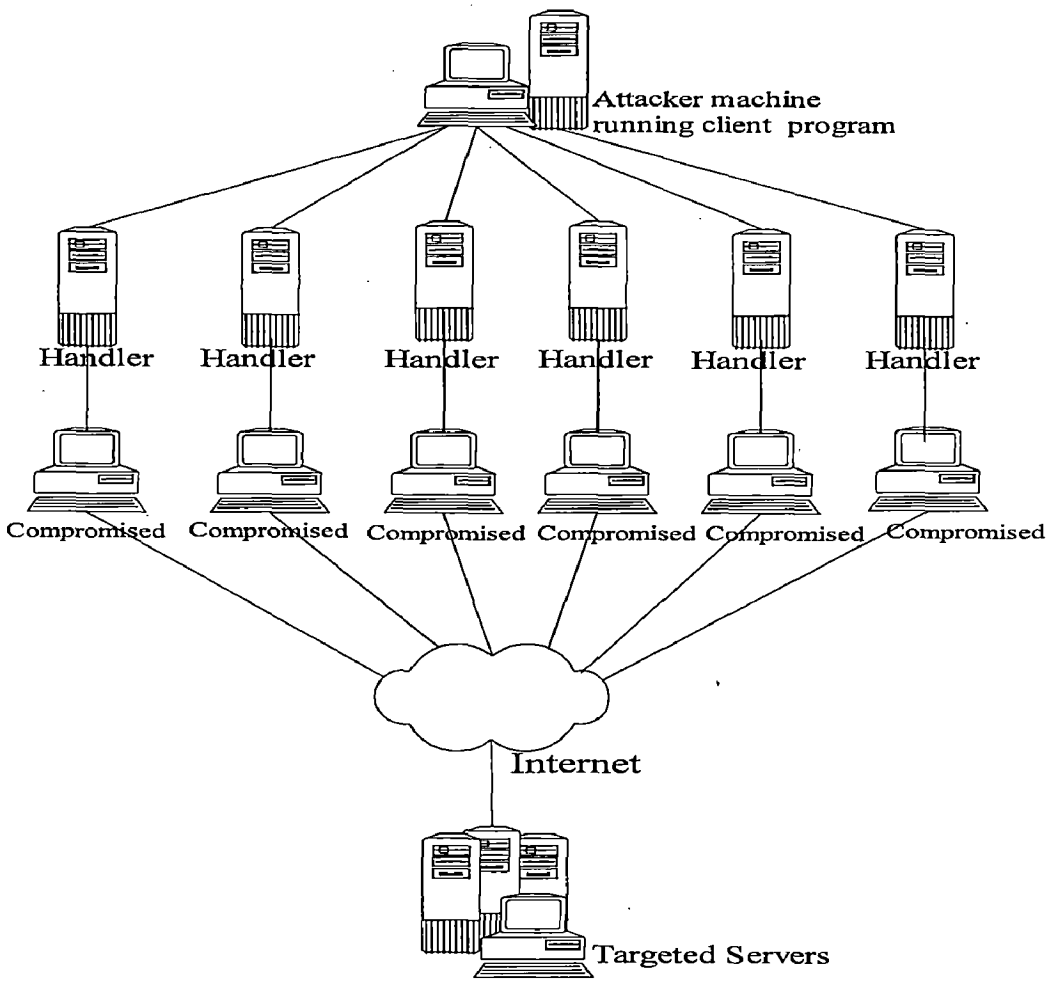

Figure 1.2: An example scenario of DDoS Attack

1.2 Motivation

The usage of internet is growing day by day and so, the number of cyber crimes is also increasing worldwide. A large number of security breach incidents are affecting many organizations and individuals. The crimes committed are becoming more and more sophisticated. Law enforcement is in a perpetual race with the cyber criminals to ensure that they are in a level playing field.

A recent study conducted by Arbor Networks [2] shows the year by year increase of the DDoS attack traffic on the Internet, from the year 2001 to 2008 as shown in Figure 1.3.

This indicates that DDoS attack traffic size has (in gigabits-per-second) nearly doubled in year 2008 from the year 2007.

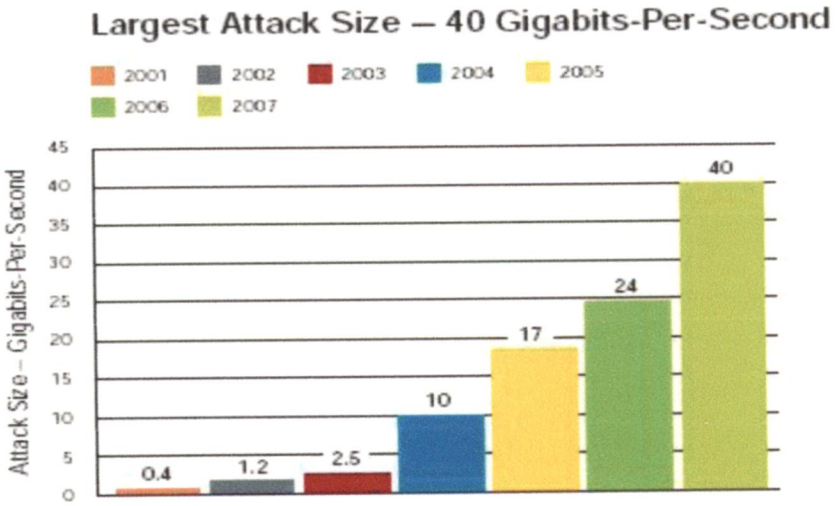

Figure1.3: DDoS attacks threaten ISP infrastructure

Various methodologies and frameworks have been proposed to defend DDoS attacks. However they have several limitations, like high computational overhead on the intermediate routers, large memory, space requirements, false positives etc. The challenge is to improve on these aspects and give a comprehensive solution.

There are two ways of defending the DDoS attacks. One way is to identify the attack packets and filter those packets using firewalls. The other way is to find out the attack source and take punitive action against them in order to avoid further attacks.
Motivation of our work is to develop efficient mechanisms to defend the DDoS attacks, in which we filter IP spoofed packets and reduce hash collisions by making it space efficient.

1.3 Statement of the Problem

The aim of this dissertation is to propose an effective and efficient scheme which promises defense against DDoS attacks in a robust way. This scheme covers the following activities simultaneously:

(i) Filter IP spoofed packets on client side using the client detector with the cooperation of the server detector.

(ii) Make an alarm to inform victim for DDoS attack.

(iii) Reduce the hash collisions.

1.4 Organization of the Dissertation

This report comprises of seven chapters including this chapter that introduces the topic and states the problem. The rest of the dissertation report is organized as follows.

Chapter 2 gives an overview of the DoS and DDoS attacks and discusses the types of DDoS attack and its attack tools and finally the DDoS defense mechanisms classifications.

Chapter 3 gives literature review of the different scheme and observes the research gaps in the existing defense against DDoS attacks are stated

Chapter 4 gives the details of the proposed scheme for defense against DDoS attacks.

Chapter 5 describes the simulation model that includes the system components. The implementation details are also explained out in terms of the topology used for simulation, simulation parameters and performance metric evaluation

Chapter 6 discusses the simulation results in terms of score and displays the comparison of the proposed scheme for defense against DDoS attacks with an existing scheme namely marking based detection and filtering scheme (MDADF). The experiments consist of studying the behaviors of attack under protocols, namely, TCP.

Chapter 7 concludes the work and gives the directions for future work.

CHAPTER 2
DDOS ATTACKS AND DEFENSE APPROACHES

According to the WWW Security FAQ a DoS attack is described as an attack designed to render a computer or network incapable of providing normal services. A DoS attack is considered to take place only when access to a computer or network resource is intentionally blocked or degraded as a result of malicious action taken by another user. These attacks don't necessarily damage data directly or permanently, but they intentionally compromise the availability of the resources. The most common DoS attacks target the computer network's bandwidth or connectivity. Bandwidth attacks flood the network with such a high volume of traffic that all available network resources are consumed and legitimate user requests cannot get through, resulting in degraded productivity. Connectivity attacks flood a computer with such a high volume of connection requests, that all available operating system resources are consumed and the computer can no longer process legitimate user requests.

2.1 DDoS attacks

2.1.1 Definition

Distributed Denial of Service (DDoS) attacks: ''A DDoS attack [7] uses many computers to launch a coordinated DoS attack against one or more targets. Using client/server technology, the perpetrator is able to multiply the effectiveness of the DoS significantly by harnessing the resources of multiple unwitting accomplice computers, which serve as attack platforms''.

The DDoS attack is the most advanced form of DoS attacks. It is distinguished from other attacks by its ability to deploy its weapons in a ''distributed'' way over the Internet and to aggregate these forces to create lethal traffic. DDoS attacks never try to break the victim's system, thus making any traditional security defense mechanism [4] inefficient. The main goal of a DDoS attack is to cause damage on a victim either for personal reasons, or for material gain, or for popularity.

2.1.2 DDoS strategy

A Distributed Denial of Service Attack is composed of four elements [4]. As shown in Figure 2.1 consists of:

1. The real attacker.
2. The handlers or masters, which are compromised hosts with a special program running on them, capable of controlling multiple agents.
3. The attack daemon agents or zombie hosts who are compromised hosts that are running a special program and are responsible for generating a stream of packets towards the intended victim.
4. A victim or target host.

The following steps take place while preparing and conducting a DDoS attack:

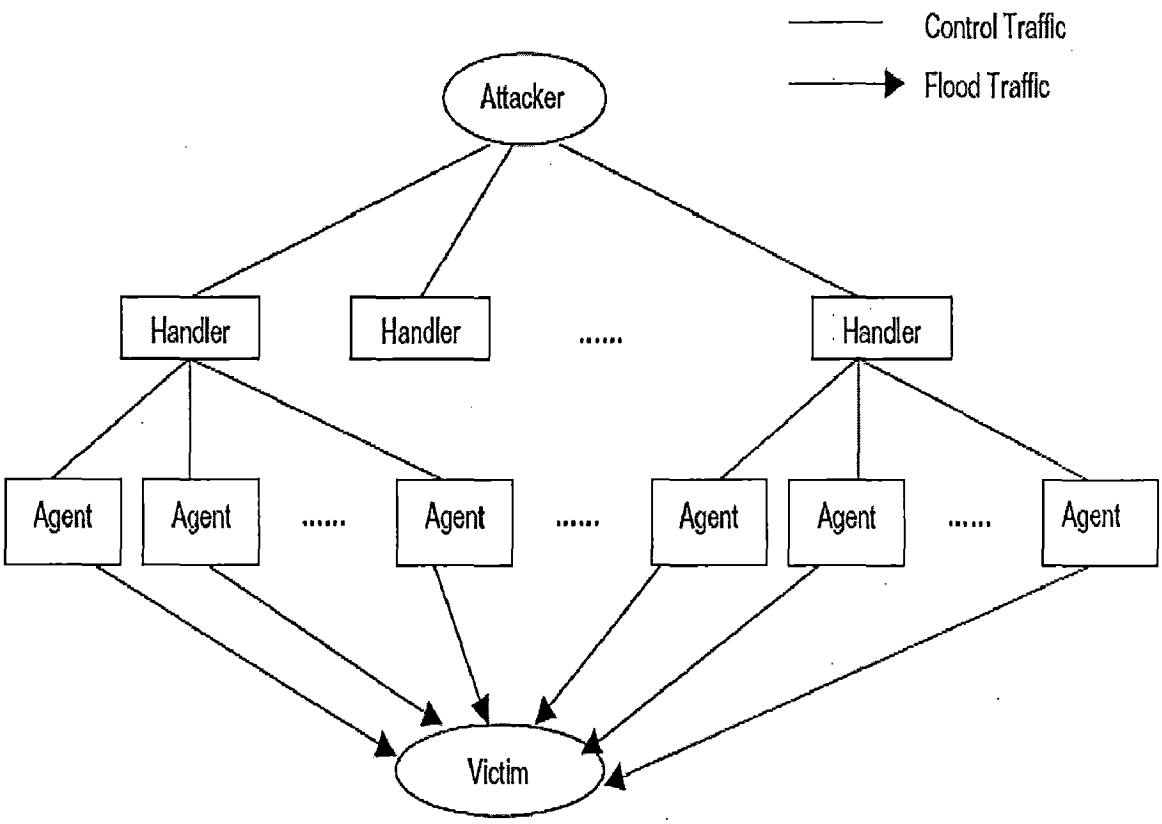

Figure 2.1: Architecture of DDoS Attacks

1. Selection of agents. The attacker chooses the agents that will perform the attack. These machines need to have some vulnerability that the attacker can use to gain access to them.

2. Compromise. The attacker exploits the security holes and vulnerabilities of the agent machines and plants the attack code. Furthermore he tries to protect the code from discovery and deactivation.

3. Communication. The attacker communicates with any number of handlers to identify which agents are up and running, when to schedule attacks, or when to upgrade agents. The communication between attacker and handler and between the handler and agents can be via TCP, UDP, or ICMP protocols.

4. Attack. At this step the attacker commands the onset of the attack.

2.2 Types of DDoS attacks

DDoS attacks can be classified into two main categories:

1) Flood attacks

A remote system is overwhelmed by a continuous flood of traffic designed to consume resources at the targeted server (CPU cycles and memory) and/or in the network (bandwidth and packet buffers). These attacks result in degraded service or a complete site shutdown.

2) Logic or software attacks

A small number of malformed packets are designed to exploit known software bugs on the target system. These attacks are relatively easy to counter either through the installation of software patches that eliminate the vulnerabilities or by adding specialized firewall rules to filter out malformed packets before they reach the target system.

2.2.1 Types of Flood attacks

1) TCP SYN Flood Attack

Taking advantage of the flaw of TCP three-way handshaking behavior, an attacker makes connection requests aimed at the victim server with packets with unreachable source addresses. The server is not able to complete the connection requests and, as a result, the victim wastes its network resources.

2) Smurf IP Attack

An attacker sends forged ICMP echo packets to broadcast addresses of vulnerable networks. All the systems on these networks reply to the victim with ICMP echo replies. This rapidly exhausts the bandwidth available to the target, effectively denying its services to legitimate users.

3) UDP Flood Attack

A UDP Flood Attack is possible when an attacker sends a UDP packet to a random port on the victim system. When the victim system receives a UDP packet, it will determine what application is waiting on the destination port. When it realizes that there is no application that is waiting on the port, it will generate an ICMP packet of destination unreachable to the forged source address. If enough UDP packets are delivered to unused ports on victim, the system will go down.

4) ICMP Flood Attack

An ICMP flood is usually accomplished by broadcasting either a bunch of pings or UDP packets. The idea is to send so much data to victim's system, that it slows the system down so much that victim is disconnected from IRC due to a ping timeout.

2.2.2 Types of Logic or Software attack

1) Ping of Death

An attacker sends an ICMP ECHO request packet that is much larger than the maximum

IP packet size to victim. Since the received ICMP echo request packet is bigger than the normal IP packet size, the victim cannot reassemble the packets. The OS may be crashed or rebooted as a result.

2) Teardrop

An attacker sends two fragments that cannot be reassembled properly by manipulating the offset value of packet and cause reboot or halt of victim system. Many other variants such as targa, SYNdrop, Boink, Nestea Bonk, TearDrop2 and NewTear are available.

3) Land

An attacker sends a forged packet with the same source and destination IP address. The victim system will be confused and crashed or rebooted.

2.3 TCP/SYN flood attack [8]

As our dissertation work mainly deals with TCP/SYN attacks, we therefore describe it in detail.

The basis of the SYN flooding attack lies in the design of the 3-way handshake that begins a TCP connection.

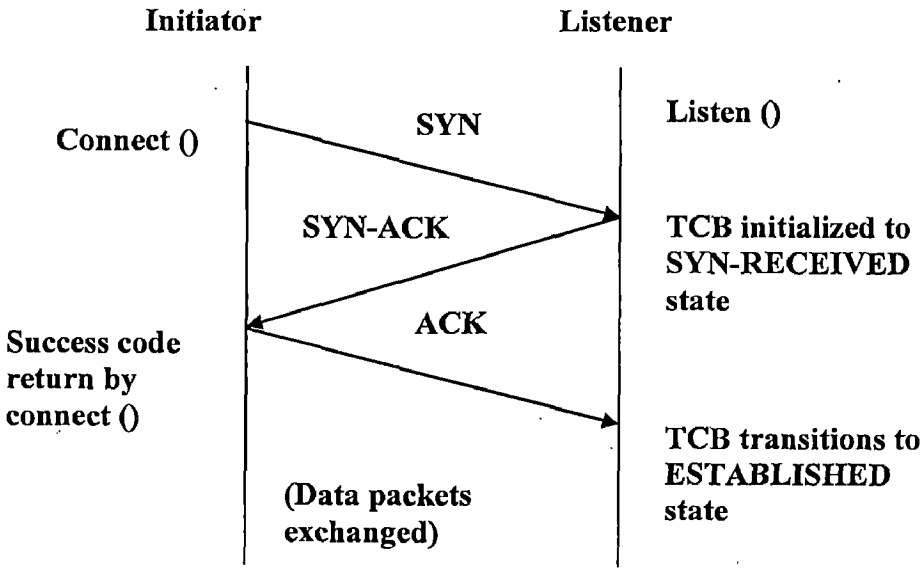

Figure 2.2: Normal TCP 3-Way Handshake

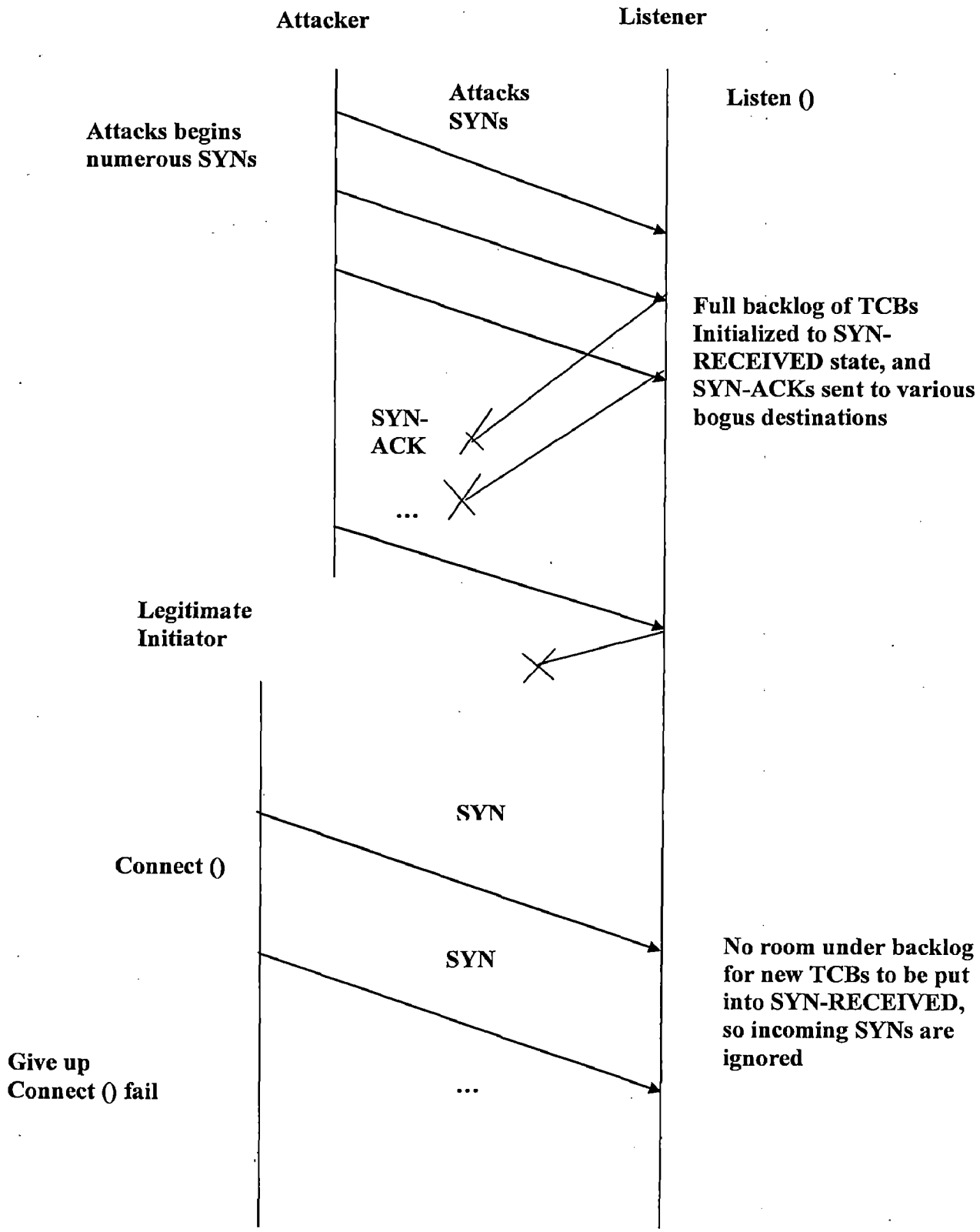

Figure2.3: Attack Demonstration

As Figure 2.2 shows the sequence of packets exchanged at the beginning of a normal TCP connection. The *Transmission Control Block* (TCB) is a transport protocol data structure (actually a set of structures in many operating systems) that holds all the information about a connection. Usually, each TCB exceeds at least 280 bytes, and in some operating systems currently takes more than 1300 bytes. The important aspect to note is that the TCB is allocated based on reception of the SYN packet— before the connection is fully established or the initiator's return reach ability has been verified. This situation leads to a clear potential of DoS attack where incoming SYNs cause the allocation of so many TCBs that a host's kernel memory is exhausted. The attacker uses source IP addresses in the SYNs that are not likely to trigger any response that would free the TCBs from the SYN-RECEIVED state. Because TCP attempts to be reliable, the target host keeps its TCBs stuck in SYN-RECEIVED for a relatively long time before giving up on the half connection and reaping them.

In the meantime, service is denied to the application process on the listener for legitimate new TCP connection initiation requests. As shown in Figure 2.3, a simplification of the sequence of events involved in a TCP SYN flooding attack.

The scenario pictured in Figure 2.3 is a simplification of how SYN flooding attacks are carried out in the real world, and is intended only to give an understanding of the basic idea behind these types of attacks.

2.4 Some DDoS attack tools [4]

There are several known DDoS attack tools.

2.4.1 Agent-based DDoS attack tools

1. Trinoo:- Trinoo is a bandwidth depletion attack tool that can be used to launch coordinated UDP flood attacks against one or many IP addresses. The attack uses constant size UDP packets to target random ports on the victim machine.

2. Tribe Flood Network (TFN): - It is a DDoS attack tool that provides the attacker with the ability to wage both bandwidths depletion and resource depletion attacks. It uses a

command line interface to communicate between the attacker and the control master program but offers no encryption between agents and handlers or between handlers and the attacker.

3. Stacheldraht: - It combines features of Trinoo (handler/agent architecture) with those of the original TFN. It also has the ability to perform updates on the agents automatically.

4. Mstream: - The mstream tool uses spoofed TCP packets with the ACK flag set to attack the target. Mstream is a simple point-to-point TCP ACK flooding tool that, as a side effect, can overwhelm the tables used by fast routing routines in some switches.

5. Shaft: - Shaft is a derivative of the Trinoo tool. It uses UDP communication between handlers and agents. The attacker communicates with the handlers via a TCP telnet connection. UDP is used for communication between handlers and agents, and messages are not encrypted. Shaft provides UDP, ICMP, and TCP flooding attack options.

2.4.2 IRC-based DDoS attack tools

1. Trinity v3: - Trinity v3 besides the up to now well-known UDP, TCP SYN, TCP ACK, TCP NULL packet floods introduces TCP fragment floods, TCP RST packet floods, TCP random flag packet floods, and TCP established floods, while randomizing all 32 bits of the source IP address.

2. Knight: - The Knight DDoS attack tool provides SYN attacks, UDP Flood attacks, and an urgent pointer flooder. It is designed to run on Windows operating systems and has features such as an automatic updater via http or ftp, a checksum generator and more. The Knight tool is typically installed by using Trojan horse program called Back Orifice.

2.5 DDoS defense mechanisms classification

DDoS attack is a hard problem to solve. First, there are no common characteristics of DDoS streams that can be used for their detection. Furthermore, the distributed nature of DDoS attacks makes them extremely difficult to combat or trace back. Moreover, the

automated tools that make the deployment of a DDoS attack possible can be easily downloaded. Attackers may also use IP spoofing in order to hide their true identity, and this makes the trace back of DDoS attacks even more difficult. Finally, there is no sufficient security level on all machines in the Internet, while there are persistent security holes in Internet hosts.

2.5.1 Classification by activity

DDoS defense mechanisms can be classified based on the activity deployed. This has following four categories [4]:

1. Intrusion Prevention,
2. Intrusion Detection,
3. Intrusion Tolerance and Mitigation, and
4. Intrusion Response.

In the following, we discuss above categories in brief.

1) Intrusion prevention

The best mitigation strategy against any attack is to completely prevent the attack. In this stage we try to stop DDoS attacks from being launched in the first place. There are many DoS defense mechanisms that try to prevent systems from attacks.

Using globally coordinated filters, attacking packets can be stopped, before they aggregate to lethal proportions.

Prevention approaches offer increased security but can never completely remove the threat of DDoS attacks because they are always vulnerable to new attacks for which signatures and patches do not exist in the database.

2) Intrusion detection

Intrusion detection has been a very active research area. By performing intrusion detection, a host computer and a network can guard themselves against being a source of network attack as well as being a victim of a DDoS attack. Intrusion detection systems

detect DDoS attacks either by using the database of known signatures or by recognizing anomalies in system behaviors.

Intrusion patterns can be any packet features, conditions, arrangements and interrelationships among events that lead to a break-in or other misuse. These patterns are defined as intrusion signatures.

3) Intrusion tolerance and mitigation

Research on intrusion tolerance accepts that it is impossible to prevent or stop DDoS completely and focuses on minimizing the attack impact and on maximizing the quality of its services. Intrusion tolerance can be divided in two categories: fault tolerance and quality of service (QoS).

Fault tolerance is a well-developed research area whose designs are built-in in most critical infrastructures and applied in three levels: hardware, software and system. The idea of fault tolerance is that by duplicating the network's services and diversifying its access points, the network can continue offering its services when flooding traffic congests one network link.

Quality of service (QoS) describes the assurance of the ability of a network to deliver predictable results for certain types of applications or traffic. Many Intrusion Tolerant QoS Techniques and Intrusion Tolerant QoS systems have been developed in order to mitigate DDoS attacks.

4) Intrusion response

Once an attack is identified, the immediate response is to identify the attack source and block its traffic accordingly. The blocking part is usually performed under manual control since an automated response system might cause further service degradation in response to a false alarm. Automated intrusion response systems do exist, but they are deployed only after a period of self-learning or testing.

2.5.2 Classification by deployment location

Based on the deployment location, we divide DDoS defense mechanisms to those deployed at the victim network, at the intermediate network (which forward the attack traffic to the victim) or at the source network.

1) Victim-network mechanisms

Historically, most of the systems for combating DDoS attacks have been designed to work on the victim side, since this side suffered the greatest impact of the attack. The victim has the greatest incentive to deploy a DDoS defense system, and maybe sacrifice some of its performance and resources for increased security.

2) Intermediate-network mechanisms

DDoS defense mechanisms deployed at the intermediate network are more effective than a victim network mechanisms since the attack traffic can be handled easily and traced back to the attackers. However these defense mechanisms present several disadvantages that prevent their wide deployment such as the increase of the intermediate network's performance and the greater difficulty to detect the attack since the intermediate network usually does not feel any effect from the attack.

3) Source network mechanisms

DDoS defense mechanisms deployed at the source network can stop attack flows before they enter the Internet core and before they aggregate with other attack flows. Being close to the sources, they can facilitate easier trace back and investigation of the attack. A source network mechanism has the same disadvantage as the intermediate network mechanism of detecting the occurrence on an attack, since it does not experience any difficulties. This disadvantage can be balanced by its ability to sacrifice some of its resources and performance for better DDoS detection. However, such a system might restrict legitimate traffic from a network in case of unreliable attack detection.

CHAPTER 3
LITERATURE REVIEW

This chapter gives an extensive literature survey on the existing defense mechanisms. It also presents an analysis of the existing methods. The limitations of the existing techniques are also described.

3.1 Existing Defense Approaches against IP spoofed DDoS attacks

Several mechanisms exist to defend against IP spoofed DDoS attacks. DDoS attacks make the computer resources unavailable to the legitimate users and give unauthorized access to a malicious user. In order to fulfill the requirements or the services of legitimate users, the victim must defend the attacks. These defense approaches are as follows:

3.1.1 Preventive Defense

Preventive schemes aim at improving the security level of a computer system or network, thus preventing attacks from happening, or enhancing the resistance to attacks. Proactive server roaming [9], Ingress filtering [11] and Outgress filtering fall into this category.

- **Proactive server roaming**

  A proactive server roaming scheme [9] belongs to preventive mechanisms. This system is composed of several distributed homogeneous servers and the location of active server changes among them using a secure roaming algorithm. Only the legitimate users know the server's roaming time and the address of new server. All connections are dropped when the server roams, so that the legitimate users can get services at least in the beginning of each roaming epoch before the attacker finds the active server out again. Such solutions are generally costly and difficult to re-ally prevent attacks.

➢ **Ingress filtering**

In this filtering technique, the border router filters the incoming packets, which comes with the source IP address, belongs to the same autonomous system domain. To do this the border router is configured with an access control list (ACL) that blocks the private addresses on the downstream interface. The principal problem with ingress filtering is that its effectiveness depends on widespread, if not universal, deployment. Unfortunately, a significant fraction of ISPs, perhaps the majority, do not implement this service either because they are uninformed or have been discouraged by the administrative burden, potential router overhead or the border router may be in the hands of private organizations or attackers who intentionally supports attack environment. A secondary problem is that even if ingress filtering were universally deployed at the customer-to-ISP level, attackers could still forge addresses from the hundreds or thousands of hosts within a valid customer network.

➢ **Outgress or Egress filtering**

In this filtering technique the border router filters out the outgoing packets which are going with a source IP address that does not belongs to the same Autonomous System domain. Here the border router is configured with an access control list that blocks the outgoing packets with the source IP address, that doesn't belongs to the same autonomous system domain. It is also having the same problem like ingress filtering as described above.

3.1.2 Reactive Solutions

Reactive solutions aim at improving the security of the computer system or network by detecting an ongoing attack and react to it by controlling the flow of attack by mitigating its effects. Pushback [13] method, filtering method [14] and other method proposed by Yaar et al [10] falls into this category.

The Pushback [13] method generates an attack signature based on the pattern of dropped packets after detecting congestion, and applies a rate limit on corresponding incoming

traffic. This information is then propagated to upstream routers, and the routers help to drop such packets, so that the attack flow can be pushed back.

3.1.3 Filtering Mechanisms

The existing mechanisms for filtering the IP-spoofed packets are as follows:
Wang et al. [16], uses flooding detection system (FDS) which lies in its statelessness and have low computation overhead. Instead of monitoring the ongoing traffic at the front end (like firewall or proxy) or the victim server, it detected SYN flooding attacks at leaf routers that connect end hosts to the Internet. This method detects DoS attacks by monitoring statistical changes. The ratio of SYN packets to FIN and RST packets is used and a variety of parameters, such as TCP and UDP traffic volume, are used. The attack detection is based on the following assumption. First, the random sequences are statistically homogeneous. Second, there will be a statistical change when an attack happens.

This detection scheme is based on the fact that a SYN packet will end with a FIN or RST packet during normal TCP connection. When the SYN flood starts, there will be more SYN packets than FIN and RST packets.

Jin et al. [17] utilized the TTL (Time-To-Live) value in the IP header to estimate the Hop-Count of each packet. He proposed a mechanism to filter out the spoofed packets based on the number of hops the packet traverses. This technique uses a mapping between IP address and hop-counts, and the victim uses this mapping to distinguish spoofed IP packets from legitimate ones. The drawback of this mechanism is that the hop-counts are calculated based on the value of TTL field, where the attacker may spoof this field too.

Lemon [18] incorporated SYN cache and cookies to prevent DDoS attacks, using cache or cookies to evaluate the security status of a connection before establishing the real connection with a protected server. All of these defense mechanisms are installed at the firewall of the victim server or inside the victim server, thereby providing no hints about the sources of the SYN flooding. They have to rely on the expensive IP traceback to

locate the flooding sources. Because the defense line is at, or close to, the victim, the network resources are also wasted due to transmission of flooding packets.

Keromytis et al. employed the secure overlay service (SOS) [19, 20] to proactively prevent DDoS. The architecture is constructed using a combination of secure overlay tunneling, routing via consistent hashing, and filtering. SOS architecture is composed of SOAP, overlay nodes, beacon, secret servlet and filtered region, which makes it difficult for an attacker to target nodes along the path to a specific SOS-protected destination. Based on SOS, researchers from Columbia University continued their proactive defense research.

Snoeren et al. [12] proposed a hash-based IP traceback approach, called Source Path Isolation Engine (SPIE), to realize log-based IP traceback in practice. Their approach reduces the attacks overhead but not significantly through recording packet digests in a data structure called a Bloom filter [15]. SPIE has made a significant improvement on the practicality of log-based IP traceback. However, its deployment at high-speed networks has still been a challenging task due to the high storage overhead and access time requirement for recording packet digests. Considering the effectiveness of log-based IP traceback in tracing both flooding and software exploit attacks, there is a need to develop more scalable solutions to facilitate its deployment at high-speed networks.

Chen et al. [27] extended the pushback model for detection and prevention of IP spoofed DDoS attacks by using the packet marking mechanism (MDADF). In this mechanism all the intermediate routers marks the packets with their identification. When the packet reaches the destination it contains the marking which is marked by all the routers on the path that the packet traverses. Further the firewall at the victim site filters the packets based on the marking the packet contains. According to this scheme a packet traverses on the same path contains the same mark. Based on this information the firewall filters out the packets. And also it learns the signatures from the dropped packets and pushback the list to upstream routers to rate limit the traffic before reach the victim.

Bin Xiao et al. [22] proposed a novel cooperative system for producing warning of DDoS attack. It define the detection mechanism that makes use of valuable information

obtained at the innocent host whose IP is utilized as the spoofed IP and which receives abnormal TCP control packets during the three-way handshake. The cooperative system approach introduced by Bin Xiao et al performed pretty well but it could not stop the legitimate users to suffer. The legitimate users didn't get network services when the hash table is full which cause the priority for collision of synchronization request.

3.2 Research Gaps

In this section we give the research gaps found in different DDoS attack mechanisms.

In Flooding detection system scheme (FDS) [16] the simplicity lies in its statelessness and the weakness of the SYN-FIN pairs lies in its vulnerability to simple counter-measure. SYN floods are detected by monitoring statistical changes. The attacker can avoid detection by sending the FIN or RST packet in conjunction with the SYN packets. To beat the detection scheme, the attacker can carefully mix different types of traffic to ensure that the proportion of each traffic is same as it is in normal traffic. Therefore, separating different types of traffic cannot make the attack behavior obvious or conspicuous.

In Hop-count filtering [17] method hop-counts can be calculated based on the value of TTL field where the attacker may spoof this field too.

In syncookies and syncache method [18] When syncookies are enabled, the existing code does not drop any entries from the syncache, choosing to send a syncookie response instead. However, this leads to the syncache being full of bogus entries from a SYN flood, and forces all legitimate connections to be handled by syncookies. Essentially, the system ends up behaving as if there is no syncache, which is not an ideal situation. I also observed that an unmodified machine provides unacceptable response times under a simple SYN flood attack.

In SOS (Secure Overlay Service) [19, 20] the architecture is composed of SOAP, overlay nodes, beacon, secret servlet and filtered region, which makes it difficult for an attacker to target nodes along the path to a specific SOS-protected destination.

In SPIE [12], ingress filtering had implementation issues. Partial implementation of these solutions caused loopholes in system which makes the solution ineffective.

In Bloom filter [22] there is a large priority for hash collisions and it is not space efficient. The storage cost and memory consumption is large. So, it cannot reduce potential false negatives and positives rates.

There is no scheme proposed which can effectively filters the incoming IP-spoofed packets. Therefore, we require a mechanism which fulfils the following needs.

1) Proposed solution should reduce storage cost, memory and computing cost.
2) It should require only a small hardware change on routers.
3) It must be well scalable to handle a large number of attackers.
4) It must perform real time filtering of incoming spoofed packets.
5) Compare the existing scheme namely MDADF with proposed scheme which is explained below in brief.

Marking based detection and filtering scheme (MDADF): -The MDADF [27] scheme is based on a firewall that distinguishes attack packets from the normal packets which is sent by the legitimate users and thus filters out the most of the attack packets before reaching to the victim. In this scheme the MDADF is deployed at each of the intermediate routers where the cooperation of the 20% routers is required in marking process. It also requires the participation of the third party network in its defense scheme which are deployed at the perimeter routers or the firewall of the networks.

In this scheme the source IP addresses is spoofed by attackers; the paths packets take to the destination are totally decided by the network topology and routers in the Internet, which are not controllable by the attackers. By recording the path information, the packets from different sources can be precisely differentiated, no matter what the IP addresses appeared in the packets. A router puts its IP address into the marking space of each packet it receives; if there is already a number in that space, it calculates the exclusive-or (XOR) of its address with the previous value in the marking space and puts the new value back. It ensures that the marking does not change its length when a packet

travels over the Internet, so the packet size remains constant. The marking is not used to find the source of attack, but it is used to separate attack packets.

In this scheme each cooperating router on the path of an IP packet would insert a mark on the ID-field of the packet. It uses CRC-16 hash function to mark IP address which is easy to compute and has low collision rate. So, to distinguish the spoofed packets, the firewall needs to keep a record of the genuine markings. The (IP-address, marking) pairs are stored in a Filter table, which are later used to verify each incoming packet and filter-out the spoofed ones. There are some drawbacks in MDADF scheme:

[1] By employing the filtering scheme, the firewall can protect the victim Web site by filtering out attack packets. However, sometimes the attack flow may be too large and the firewall may not have enough resources to handle it.

[2] When the firewall adds new entries or updates old entries in the Filter Table, these entries are sent as updates to the upstream routers.

To fulfill the above mentioned requirements, we propose a space-efficient Bloom-Filter detection scheme that enables the victim to perform real time filtering of incoming IP spoofed traffic at the edge routers placed at innocent host side

CHAPTER 4
PROPOSED DDoS DEFENSE MODEL

4.1 Design Methodology

A defense system must be able to differentiate between legitimate and attack traffic. In simple attacks, the traffic is generally somewhat differentiable from legitimate traffic. But in most of the cases we need to gather enough information before the attack can be detected this makes responding to attack and prevention of attack almost impossible. We must strike a balance between gathering enough information to characterize the attack and not overload logging and analysis capability. Moreover responding to the attack requires fast detection and accurate characterization of the attack streams so that they can be filtered or rate limited. DDoS attack exhausts host resources or the network bandwidth. It is consequently important to detect resource usage changes and reduce detection time. Choosing a set of parameters to monitor for anomalies directly affects detection and prevention accuracy and time. A properly chosen set of parameters will not generate too many false positives, while still detecting the majority of attacks early.

In this chapter, a Bloom Filter based detection scheme is presented which incorporates the methods for filtering with independent hash functions. The main goals of this scheme is to: i) Make a space-efficient data structure, ii) Distinguish the IP Spoofed packets from the legitimate packets, and finally iii) Reduces hash collisions.

In this scheme, each edge router filters the incoming packets and forwards to the downstream router. The basic aim of this scheme is to defend the DDoS attack. That is why we combine the features of detection and filtering mechanisms into a single mechanism. This kind of methodology has not been attempted in earlier reported works.

4.2 Efficient Approach at the Source-End

We first explain the difference between the three-way handshake of normal TCP connection and that of abnormal half-open connection designed. Based on this difference

of handshakes, our DDoS detection method is designed. To save the storage cost and to reduce the computation overhead, a Bloom filter based hash data structure is used. A simple but efficient detection scheme is proposed. Our method is expected to attract more ISPs to participate in the source-end DDoS defense because detection method does not bring evident performance degradation to network infrastructures.

4.2.1. Analysis of Half-open Connection

We first analyze the difference between normal traffic and attacking traffic. The different three-way handshake scenarios of normal TCP connection and abnormal half-open connections caused by spoofed IP DDoS attack are compared. The normal three-way handshake is shown in Figure 4.1. First the client sends a *SYN* request to the server. After receiving such request, server replies with a packet, which contains both the acknowledgement *ACK* and the synchronization request *SYN* (denoted as *ACK/SYN*). Then the client sends *ACK* back to the established the connection. In the Figure 4.1, k and j are sequence numbers produced randomly by the server and client respectively during the three-way handshake. All the three-way handshake control packets will be observed at the side of the client.

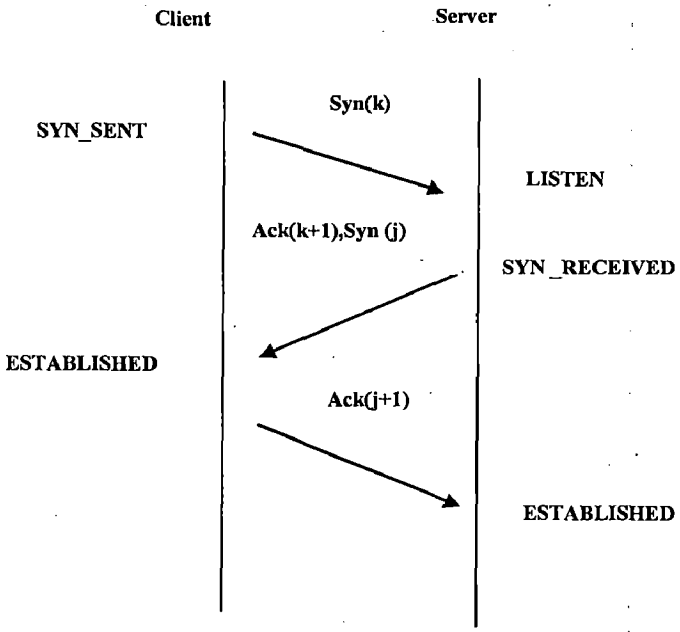

Figure 4.1: Normal three-way handshake

In a spoofed IP DDoS attack, the three-way handshake is not same Figure 4.2 shows the difference. The attacker usually uses an unreachable spoofed source IP in the attacking packet. The packet does not trigger the third round of handshake. The detector at the source only observes the first round of handshake, but will never find the second and the third round handshake.

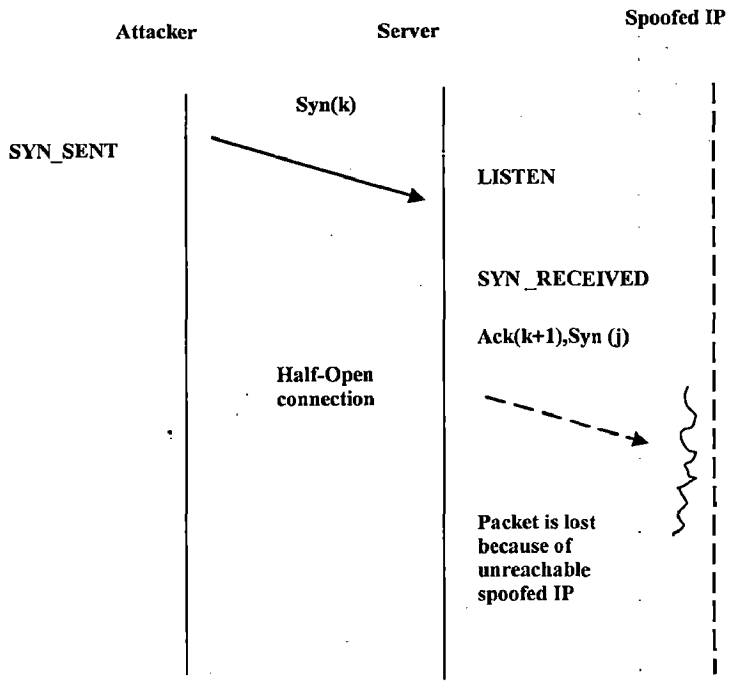

Figure 4.2: Abnormal Half-Open connection caused by spoofed source IP

4.3 Space–Efficient monitoring table

In order to capture abnormal handshake at the client side, the traffic is analyzed and recorded. Considering the volume of traffic on the Internet, this requires significant memory and computational resources to record behavior of each packet. Therefore based on Bloom filter, a space-efficient hash data structure is used to record the behavior of each packet. We first introduce the original Bloom filter and then present our modified Bloom filter.

4.3.1 Bloom Filters

Bloom filter is first described in 1970 by Burton Bloom [15] and they have been widely used in many applications such as database applications, peer to peer networks, resource allocation and packet routing, to reduce the disk access to differential files and other applications, e.g. spell checkers.

A Bloom filter is a space-efficient data structure which is used to test whether an element is a member of a set. It is an array of m bits $\{b_1, b_2 \ldots bm\}$, initialized to zero, used to represent a set of n elements, $S = \{x_1, x_2, \ldots, x_n\}$. The filter uses k independent and uniform hash functions, $h_1, \ldots h_k$ each returning a value between 1 and m. To add an element $x_i \in \{x_1, \ldots, x_n\}$ to the filter, k hash functions are applied on the input x_i and the corresponding bits in the filter are set see figure 4.3. The following is the pseudo-code for adding an element x to the filter.

```
ADD ELEMENT X
For j = 1 to k
  Do
    Filter [hj(X)] ←1
```

It should be noted that when a bit is already set to "1"-the n additional settings do not change it. The existing "1" is just over written which is a simple OR operation of all hash values. To test the membership of an element y, the k hash functions are applied to y and the corresponding bits are checked. If one of the bits is "0" then clearly the element is not in the set $S=\{y_1,y_2....y_n\}$. If all the bits are equal to "1" then we could say that the element belongs to the set. The following pseudo-code checks if y is an element of the set.

```
CHECK ELEMENT Y
For j = 1 to k
  Do
    If filter [hj(Y)] = 0 return False
Return True
```

If an element z has all the corresponding bits equal to "1" without the element itself belonging to the set then we can call that a false positive. The false positive rate can be calculated as follows.

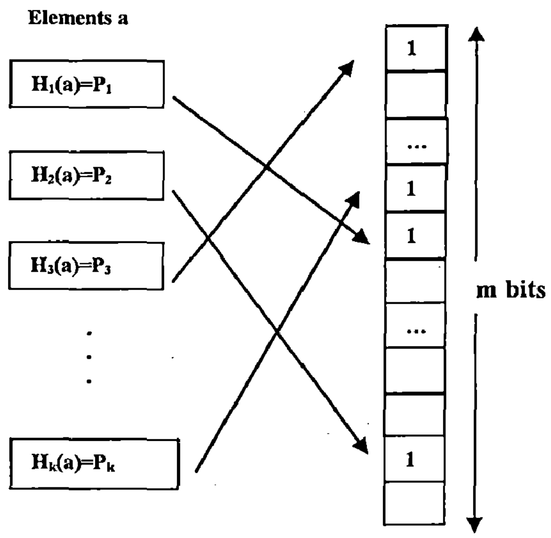

Figure 4.3: Original Bloom Filter uses independent hash functions to map input into corresponding bits

When a given hash function h_i is applied to an input x_i the result is a value between *1* and *m*. Since the hash functions are uniform, the probability that this result is equal to a particular number *b* is $\frac{1}{m}$. Therefore the probability of the bit at position *b* being 1 after one hash function is $\frac{1}{m}$. The probability that it is 0 is $1-\frac{1}{m}$. The probability that it is 0 after all *k* hash functions are applied is $(1-\frac{1}{m})^k$. Since there are *n* elements in the set, the probability that the bit *b* is equal to 0 after we process all n elements is $(1-\frac{1}{m})^{kn}$. Hence $1-(1-\frac{1}{m})^{kn}$ is the probability that a given bit *b* is set to 1 after all input elements $x_1, \ldots, x_n$ are processed. Since we want the false positive rate, we need the probability that for an arbitrary input *y* the corresponding *k* bits are 1 without *y* belonging to the set. This probability is

$$f_p = \left(1-\left(1-\frac{1}{m}\right)^{kn}\right)^k \qquad (1)$$

$$\approx \left(1 - e^{-\frac{nk}{m}}\right)^k \quad \text{..... (2)}$$

From the above equations we can say that the false positive rate depends on k and the ratio $\frac{m}{n}$.

Now, Bloom filter with hash function has been extended to defend DDoS attack [23, 24, 26]. We propose a modified Bloom filter in order to construct a hash table that can record three-way TCP control packets at a limited storage cost. The modified structure of the hash table makes it possible to capture abnormal handshakes even where the volume of traffic is large and also avoid hash collisions with fixed space efficient data structure.

4.3.2 Modified Monitoring table of Bloom Filter

Considering numerous IP addresses in network traffic, using limited m bit array to record IP address is not sufficient and may bring high false positive. We make two main modifications to original Bloom filter as shown in Figure 4.4: First, we use large array of hash table to substitute m bit array. Second we split the IP address into several segments and hash them separately into hash table in which counts are initialized to 0. It is split into segments because it keeps track of the recent arrival rates of packets of different destination IP addresses passing through router. After using counts table to replace m bit array, all the counts are initially o. When a key is inserted or deleted, the value of count is incremented or decremented by 1 accordingly. When a count changes from 0 to 1, the corresponding bit is turned on. When a count changes from 1 to 0 the corresponding bit is turned off. The value in the count indicates the current statistic results of traffic. The advantage of modified Bloom Filter is that it reduces the space from using a counter for each of the possible destination IP addresses.

The IP address is split into k segments and here i have set $k = 4$. Then each segment is an octet in IP address, which is more convenient to process. Since the value range for each octet is from 0 to 255, the m is set to 256 i.e. each table contains 256 counts. If the IP address is directly hashed into monitoring table [26], there will be more hash collision.

The reason is the number of counts is relatively limited compared to numerous values of IP addresses the internet. When the IP address is separated into several segments, the value range becomes small for each segment. This may reduce collisions and also has low false positive rates.

In the proposed scheme, both the source IP and destination IP are recorded in the hash table. In the Bloom filter, k tables by m bins with k independent hash functions are used to record the IP address of the three-way handshakes. Although it is possible that some segments of the two IP addresses are mapped into the same count in one table, the probability is rather very low than that of segments of two different IP addresses are mapped to the same count in all k tables.

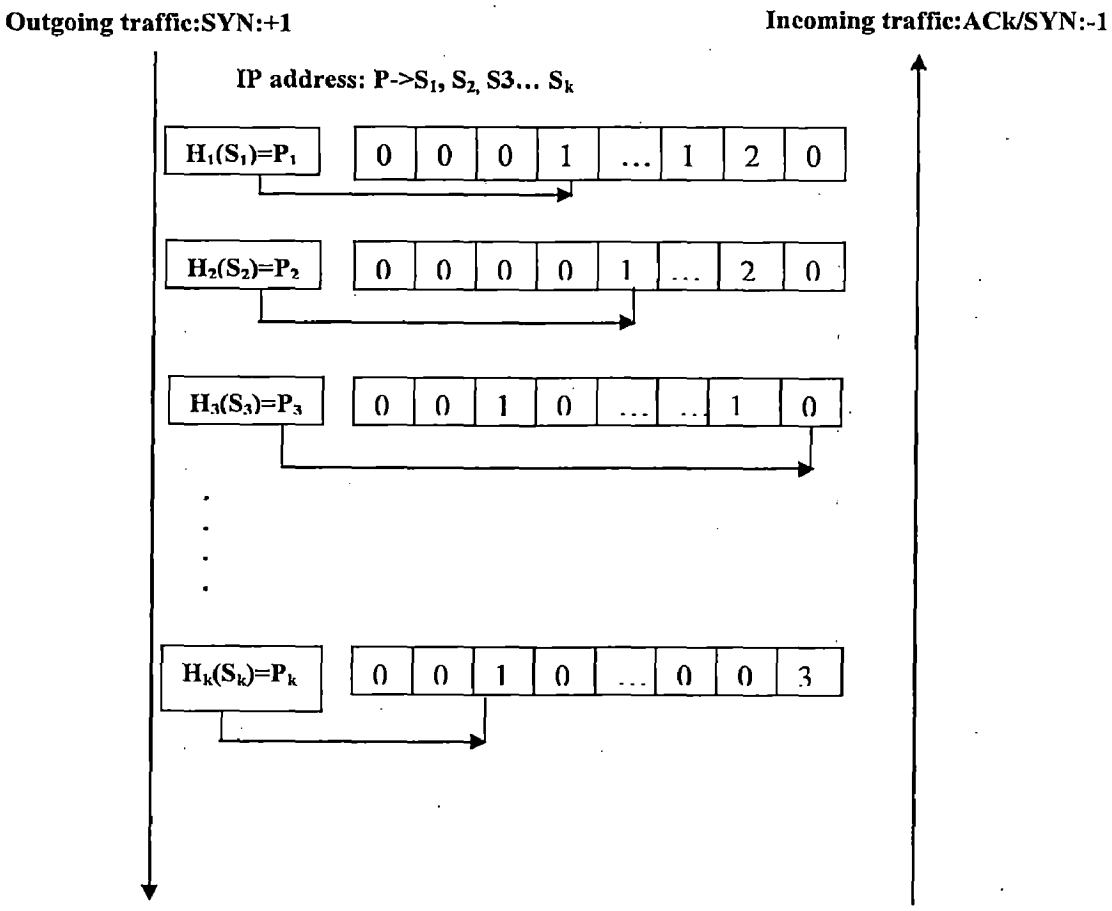

Figure 4.4: Modified Bloom Filter uses independent hash functions to map input into corresponding bits

4.3.3 Detection Scheme

To detect a DDoS attack at an early stage, the cooperation scheme is proposed which consist of client detector and server detector. The client detector, deployed at the edge router of innocent hosts, checks the TCP control packets flowing through the edge router. When it captures suspicious events, it notifies the protected server of a potential DDoS attack. The server detector, deployed by the protected server, detects attacks not only by passively listening for warnings from the client detector, but also by actively sending queries to the client detector to confirm alarms. So, the attacks are not only detected by client side but it needs the cooperation of the server to warn the DDoS attacks.

The Client Detector: - It is deployed at the edge router on the innocent client side. One of the main tasks of the client detector is to monitor the TCP control packets entering and leaving a domain. The detection scheme is developed from a modified hash table. We have designed the new hash table based on the Bloom filter method. Each states of the TCP three-way handshake are recorded in the hash table and the abnormal asymmetric three-way handshake is recorded and seen inside the client detector. After the accumulation of suspicious alarms at the client detector at the certain threshold score issues a DDoS attack and warning will be sent to the server.

To detect attacking traffic with spoofed source IP, the destination IP is recorded in the monitoring table. When a *SYN* packet, for the first round handshake, is captured from the outgoing traffic, the destination IP (the server's IP) is spited into several segments and then hashed into the monitoring table. If the corresponding count is 0, the corresponding count is turned on. If the count is already turned on, the count is incremented by 1. If corresponding *ACK/SYN* packet for the second round of handshake is soon captured in the incoming traffic, the source IP (the server's IP) is hashed into the hash table again. But this time the corresponding count is decremented by 1. When a count changes from 1 to 0, the corresponding bit is turned off i.e. there is space made for the other packets. The count will remain unchanged if the first two rounds of three-way handshake are completely captured at the client and server detector at the source side. These counts are reset to 0 to restart monitoring process for every period of t.

Client _Detection Scheme

```
Void Outgoing_Packet_Process (Input: P) {
R= (source IP, destination IP)   //R is record to maintain database for monitoring table
if P is a SYN packet then
for  (i=0; i<k; i++ ) do
j= Hash_i (P)
counter_j ++
end for
end if
Void Incoming_Packet_Process (Input: P) {
if P is ACK/SYN packet then
R= (destination IP, source IP)            //map record with the database
for (i=0; i<k; i++ ) do
j= Hash_i (P)
//check whether it meets turn- off counter i.e. counter ==0
if counter==0  then
Suspicious Alarm (SA) is reported
end if
end for
end if
// if it does not meet turn-off counters
for  (i=0; i<k; i++ ) do
j= Hash_i (P)
counter_j --
end for
end if
Return
}
```

Figure 4.5: Client detection scheme

If there is no *ACK/SYN* packet sent back to respond to previous *SYN*, the count has no chance to be decremented by 1 for this handshake. The value in the count will grow large because it is increased by 1 for each spoofed *SYN* packet. When a DDoS attack happens, an exceptional heavy volume of packets are sent toward the victim server. If there is at least one count in table containing suspicious value, then it is recorded in database for further analysis. So, when the value of a count exceeds the predefined threshold during period t, then this value is regarded as suspicious and the DDoS attack alarm will be warned. As shown in figure 4.5 for the client detection scheme.

The detection scheme only requires a simple hash operation and addition/subtraction operations. These operations bring little overhead to the computers.

When a new Suspicious Alarm (SA) is reported, the client detector will analyze the source IP distribution of SAs in its database. During a DDoS attack, the client detector find asymmetric *ACK/SYN* packets sent from a victim server. When SAs are reported from packets with the same source IP (server's IP) in a short period, there is probably a DDoS attack targeting the host. However, each SA comes from a different source IP. To evaluate the distribution of the source IP of SAs, a score is calculated as follows:

$$\text{Score}=\sum_{S \in \text{IP list}} (|Xs| - 1)^2$$

where *Xs* stands for a subset of the IP list that contains all IPs from reported SAs. All elements in *Xs* have the same IP value *s* in a certain period. The score will increase, when the number of SAs containing the same source IP increases. On the other hand if each of the SAs has a different source IP, the score will be 0. This score value can be an indicator for DDoS attacks. To save computation when a new SA comes, we use the following expression to calculate the score value:

$$\text{Score}_{\text{current}} = \begin{cases} \text{Score}_{\text{previous}} & s \text{ not in the history IP list} \\ \text{Score}_{\text{previous}} + 2 \times |Xs| - 1 & s \text{ in the history IP list} \end{cases}$$

The equation describe that upon the arrival of a SA whose source IP is not in the history IP list, the score remains unchanged while if it is in the history lists we have a new increased score. Because the score is the sum of $(|Xs| - 1)^2$, the new score is equal to the previous score adding the current $(|Xs| - 1)^2$ and minus the previous $(|Xs| - 1 - 1)^2$, i.e., $Score_{previous} + 2 \times |Xs| - 1$. When the score exceeds a predefined threshold, the reported SAs with the IP of victim is sent to the server detector at victim address.

On the other hand, whenever a query is received from a server detector, the number of SAs with the IP of the server in the database of the client detector is sent back.

The Server Detector: - The server detector is deployed at the protected server. With the assistance of client detectors, a server detector can detect a forthcoming DDoS attack at an early stage. As shown in Figure 4.6 the two parts of the server detection scheme. Both parts operate independently and concurrently and issues a confirmed DDoS attack alarm. Since this confirmation has no negative effect on the protected server, the server can perform a query as soon as any suspicious half-open connections are observed.
This is a distinct advantage over many other DDoS detection methods, which must wait to capture sufficient DDoS attack evidence before taking any further action, a requirement which delays DoS attack detection and prevention. In our scheme, cooperation between the client and server detectors ensures that the server detector launches DDoS alarms at a very early stage. The server detection scheme is composed of two parts. Part one shows that the server detector may passively wait for the potential DDoS alarm from client detectors. When enough potential DDoS attack alarms arrive, it will send the server a confirmed DDoS attack alarm. Part two shows that the server detector also performs more active detection by sending queries to client detectors when there are too many half-open connections observed. It is possible, however, that the source IPs of the spoofed packets is widely distributed with the result that the number of SAs (Suspicious alarm) at a client detector is insufficient to provoke the sending of an SA to the server detector. In this scheme, the server detector will select several cooperative client detectors to query about the number of SAs. The selection of client detectors depends on the source IPs of current half-open connections reserved by the server. A query is first sent to a client detector that is in a routing domain containing the most

pending connections. After receiving replies, the server detector can tell whether a half-connection is caused by a spoofed DDoS packet and an alarm is made or whether it is caused by something else and no action is required.

Server _Detection Scheme

If a Suspicious Alarm comes from the client detector then
Store it in the database of the Server detector
Sum the no. of Suspicious Alarm (N)
IF N> threshold (T) then
Confirmed DDoS Attack alarm
End if
End if
If half-open connection is found on the server the
Send queries to the client detector
Wait for the reply (reset)
Sum the no. of SAs (N) reply by client detector
If N> threshold (T) then
Confirmed DDoS Attack alarm
End if
End if

Figure 4.6: Server detection scheme

As shown in figure 4.7, its shows the flowchart of the detection scheme using Bloom filter with hash functions describing the proposed scheme.

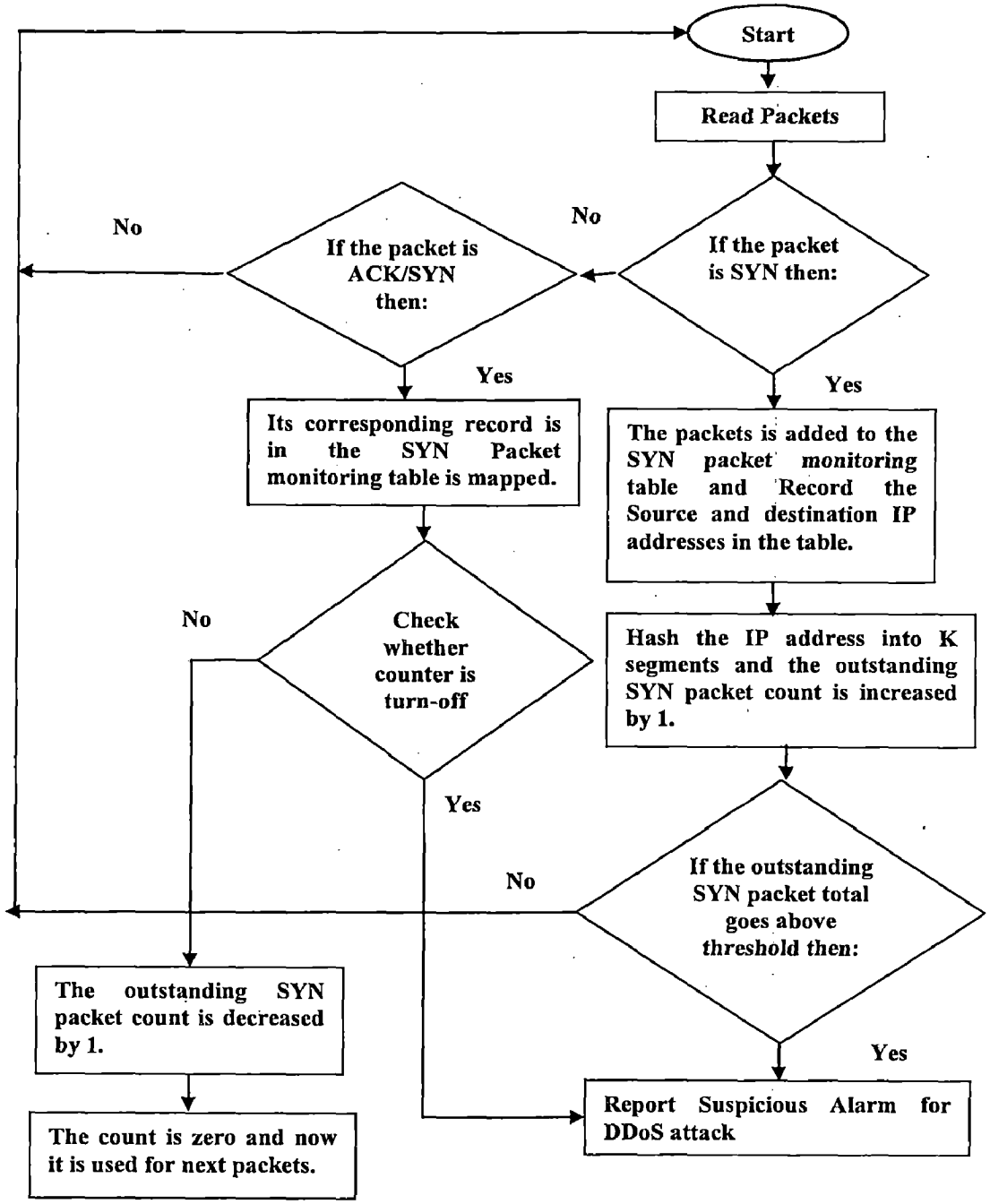

Figure 4.7: Flow chart showing detection scheme using Bloom-filter

CHAPTER 5
SIMULATION MODEL

To investigate the effectiveness of the proposed scheme in defending against DDoS attacks, the simulation on a simplified topology was carried out using Network Simulator version (ns-2.34) [23]. As my work was focused in wired networks, Ns2 was the best option to choose.

5.1 Simulation Model

5.1.1 System Components

The system consists of the following components:

Clients: - Two types of clients are considered: Legitimate Clients and Attackers. The legitimate clients are modeled by FTP applications running on **TCPNewReno** (a flavor of TCP). They obey the constraints imposed by the TCP protocol. The attackers are modeled by half-open connection of TCP. They try to establish TCP connection with the server using spoofed IP addresses results into an half open connections.

Server: - The service provided by the server is a generic TCP-based service. The server is modeled by a simple TCPSink which sends out ACK packets for packets it receives. The legitimate clients connect to the server with the aim of achieving file downloads, whereas the attackers aim at clogging the bottleneck link leading to the server in order to make the service unavailable to the legitimate clients.

Agents on Edge Routers: - One new agent (Bloom-Filter) is created in order to provide for the functionality of the proposed scheme. They are deployed at the edge routers.

Bloom-filter based hash function: - Agent is deployed on the routers that are located at a certain pre-determined distance from the client and server. This agent receive packets from the clients (legitimate and attackers) that are actually aimed for the server and store them in monitoring table before sending to the server.

5.1.2 Simulation Topology

Figure 5.1 illustrates the simulated network topology. The topology considered is similar to the one used traditionally to depict a typical client-server scenario in the Internet for simulation and validation purposes [26].

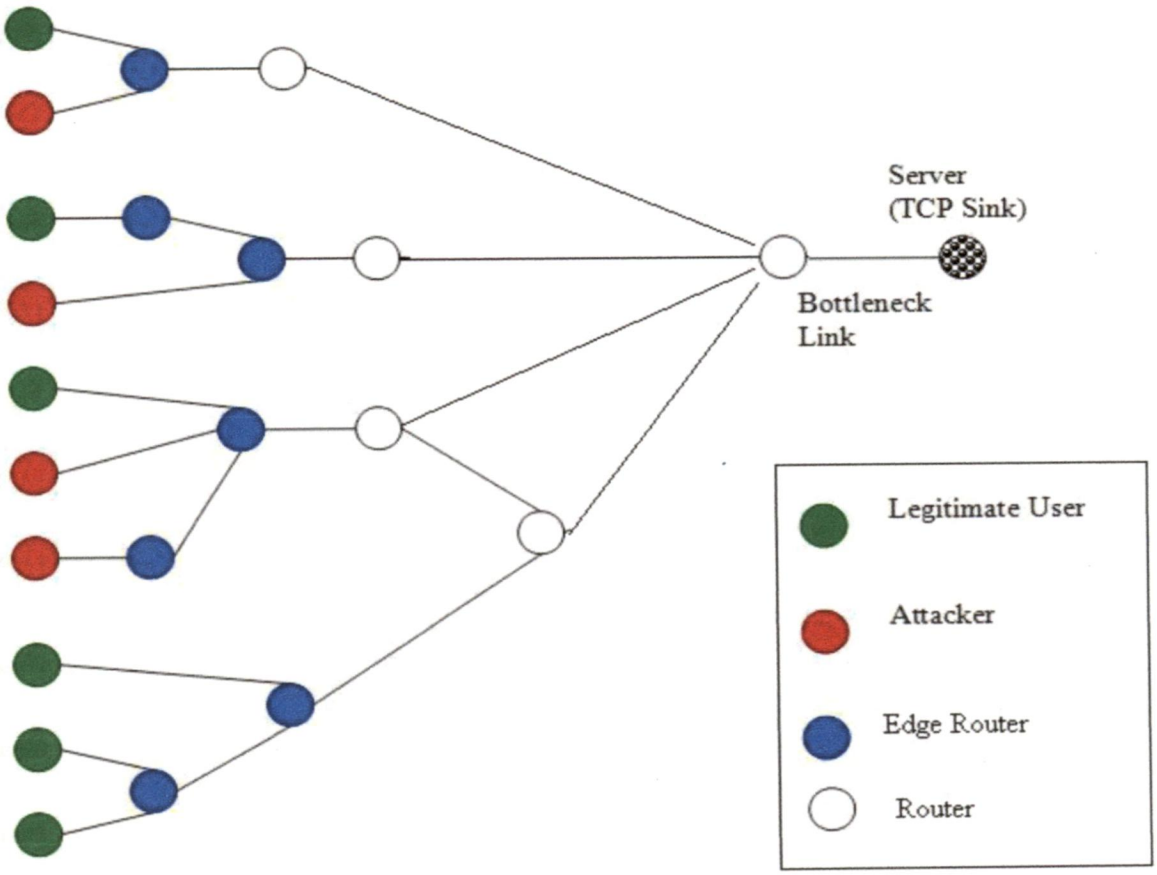

Figure 5.1: Topology used for Simulation.

The legitimate clients are TCP agents that request files of size 1 Mbps each. The rate of attack is kept very high which is very typical of an attack flow. Each of the links is a duplex link of 5 Mbps bandwidth, with the exception of the high bandwidth bottleneck link which is modeled by a combination of two simplex links of 20 Mbps bandwidth each. The Flow Monitor tailored for the purpose of Characterization is attached to the bottleneck link. Figure 5.1, the legitimate clients, the attackers and the agent edge router each using different color. The figure shows the topology of this Simulation. Legitimate user is one who sends legitimate traffic and Attacker is one who sends illegitimate traffic

from a zombie machine which is coordinated from a master and server is the victim. Edge router is placed at boundaries of an autonomous system. This edge router is directly attached with the client that monitors the traffic destined to the server. In order to separate attack traffic from legitimate traffic, the analysis of the incoming traffic going towards the target server is monitored and stored in monitoring table. Once the type of attack that is taking place is identified, we use our scheme to warn the DDoS attack.

5.1.3 Simulation Parameters

Table 5.1 lists the simulation parameters, their values and description of these parameters used in the simulation.

Parameter	Value	Description
Number of Nodes	25	Network nodes
Client Load	0. 1 - 0. 4	Relative load issued by client requests
Attack Load	0-0. 9	Relative load due to attack traffic.
Simulation time	0-80 sec	Simulation duration
Attack time	20 sec	Attack begins
Legitimate Traffic type	TCP	File Transfer Protocol
Attack Traffic Type	Spoofed IP-address	File Transfer Protocol
Client-Router link BW	5 Mbps	Bandwidth
Attacker-Router link BW	5 Mbps	Bandwidth
Router-Router link BW	5 Mbps	Bandwidth
Router-Server link BW	20 Mbps	Bandwidth

Table 5.1: Simulation Parameters.

5.1.4 Performance Evaluation Metrics

For comparing the performance of our scheme with the existing scheme, we used the following performance metrics:

1. Acceptance ratio of packets:- This is defined as the ratio of number of packets accepted, which is calculated according to the number of attackers at different thresholds. Acceptance ratio is calculated in terms of percentage of the packets accepted as legitimate and spoofed, when the filtering scheme is applied on the edge router under different magnitude of attacks. As the number of the attackers increased, there is slightly decrease in the acceptance ratio of legitimate packets as due to heavy congestion some of the legitimate packets are dropped. When the legitimate packets acceptance ratio decreases a little with the increasing number of attackers, then the spoofed packets acceptance ratio stays at a very low level.

2. Mean False Postive rate:- A false positive occurs when a benign event is declared as an attack. It is observed as :

$$\text{Mean False Positive rate} = \frac{\text{Number of false positive attacks detected}}{\text{Total number of attacks}}$$

CHAPTER 6
RESULTS AND DISCUSSION

An analysis of the results of the simulation experiments is given.

6.1 Results

To evaluate the detection performance, three scenarios are designed: there is no attacking traffic, the total traffic contains 1% attacking traffic and the total traffic contains 5% attacking traffic. The network delay from the source to the victim server is set to 100ms and the bottleneck bandwidth for victim server is 10Mb. The attacking traffic begins at simulation time 20 seconds and the whole simulation lasts for 80 seconds. The results shown from figure 6.1 to 6.3 are the counter values in the hash table. When this value goes beyond a threshold an attack is detected. The threshold was chosen based on a number of nodes and size of hash table.

As shown in Figure 6.1, the value of the count fluctuates between 0 and threshold when there is no attack traffic.

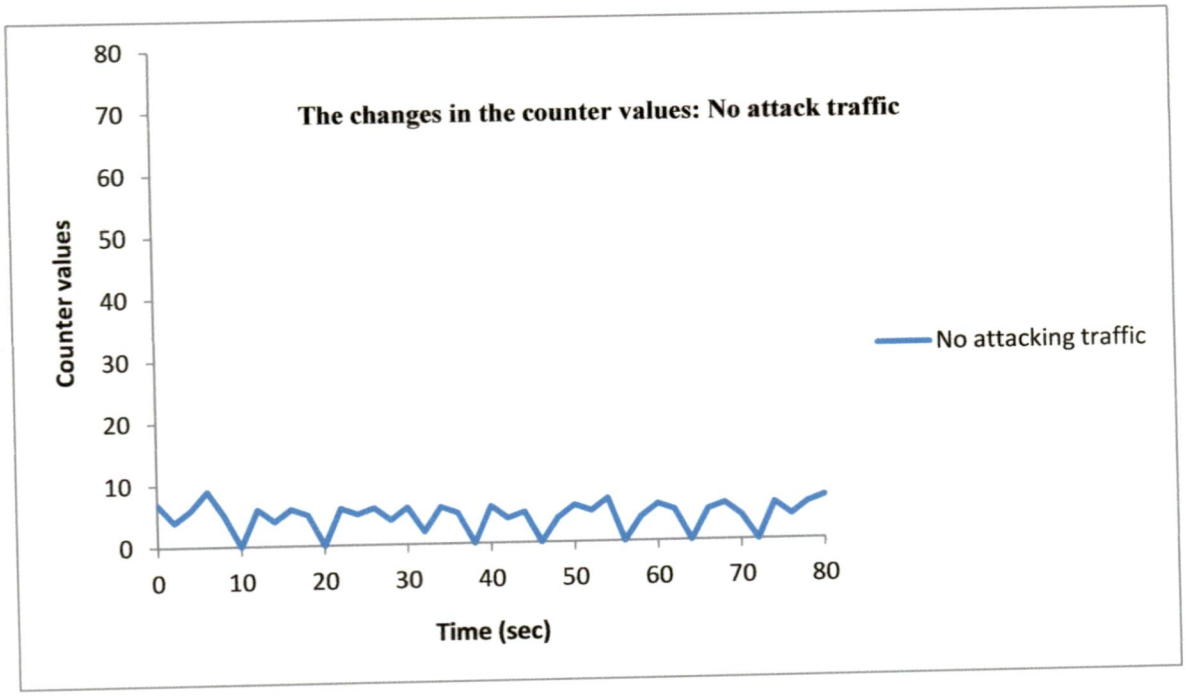

Figure 6.1: No attacking traffic

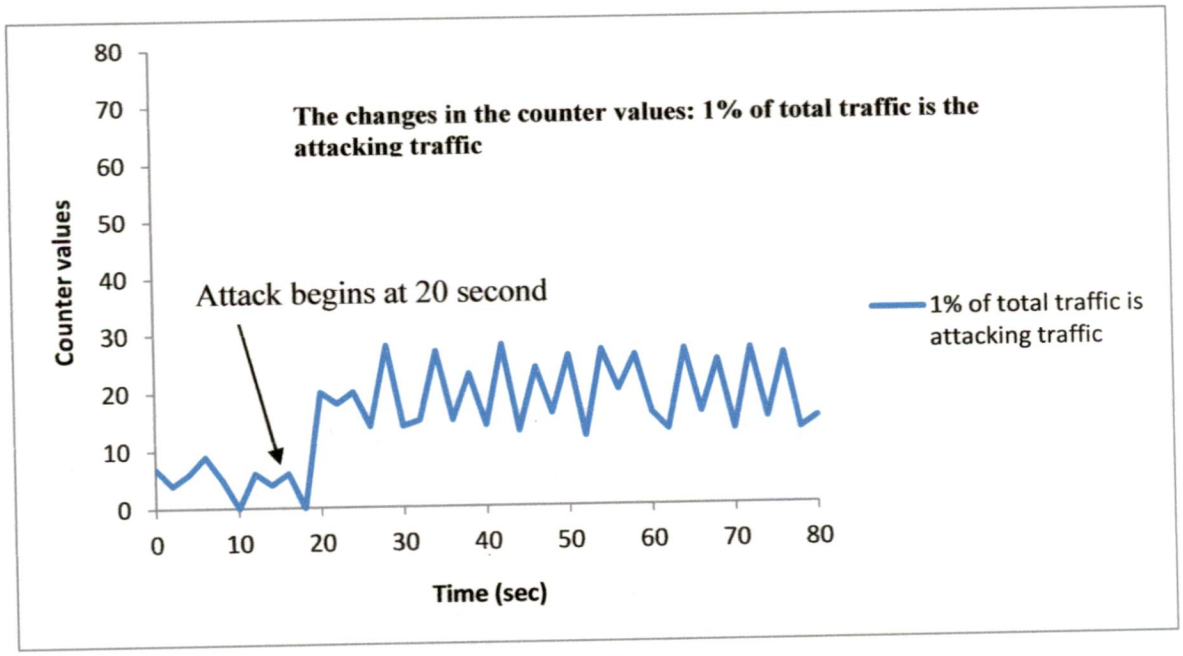

Figure 6.2: The total traffic contains 1% attacking traffic

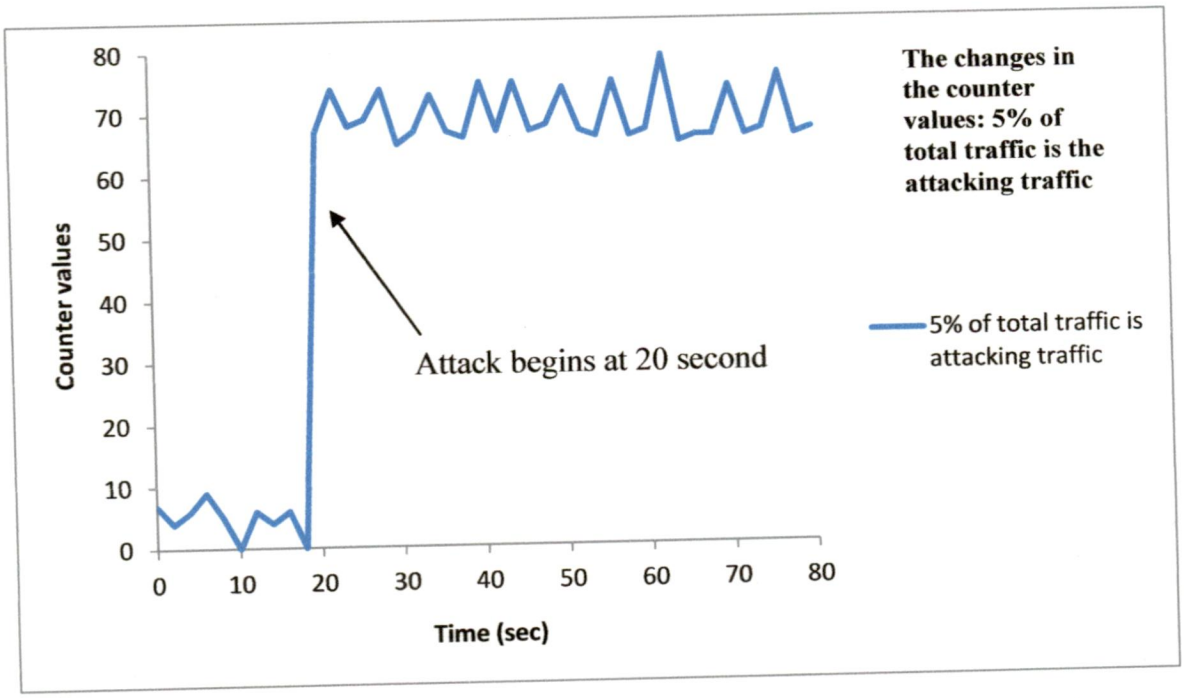

Figure 6.3: The total traffic contains 5% attacking Traffic

In case of attack, the counter value increases rapidly when the attack starts and crosses the threshold. The 5% attacking traffic triggers a much larger increase in counter value than that of 1% attacking traffic as shown in Figure 6.2 and 6.3. This shows that our method can accurately detect DDoS attack with fixed-length monitoring table.

In the next set of experiments we studied the effect of change in number of attackers for different threshold values. Each time, the percentage of legitimate packets accepted and the percent of spoofed packets accepted were observed and the results are plotted in figures 6.4 to 6.6. The values plotted are the mean values from 20 independent simulation runs.

We compared the results of our method with the MDADF [27] method. MDADF method is explained in chapter 3.

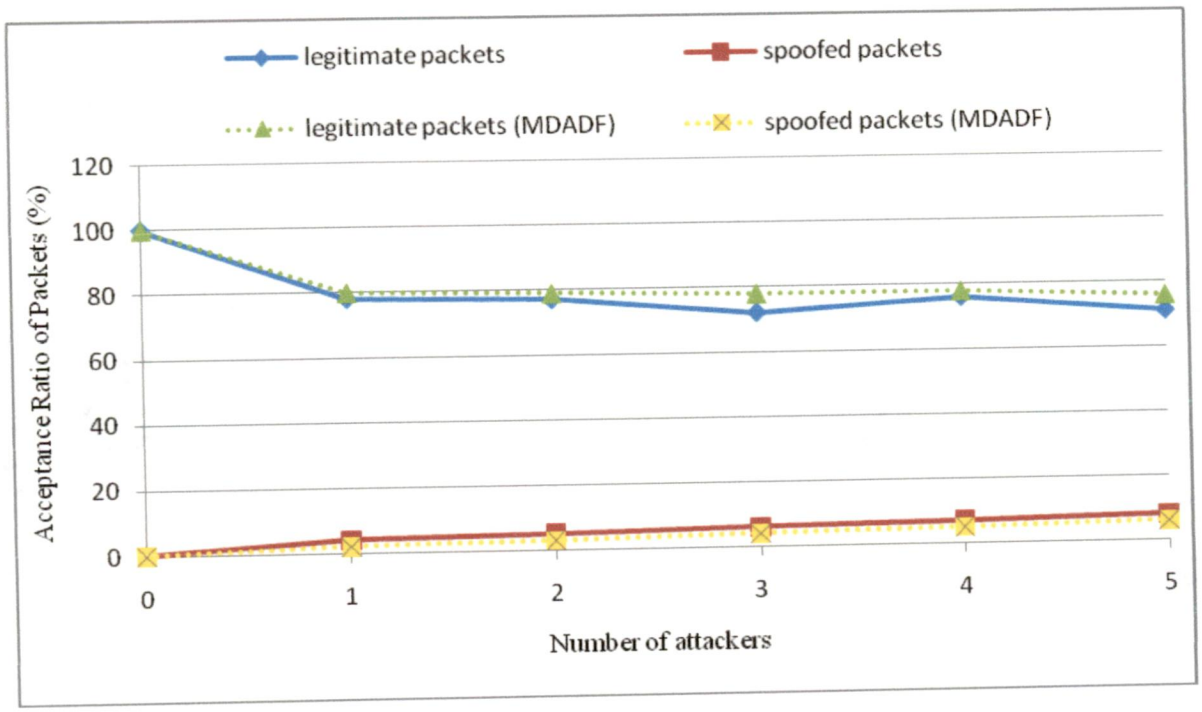

Figure 6.4: Acceptance ratio of packets vs. Number of attackers with threshold 1.

Figure 6.4 shows that the percentage of legitimate and spoofed packets accepted under different magnitudes of attack. Here we observed that around 25% of the legitimate

packets from innocent host are dropped and around 6% of the spoofed packets from attacker are accepted, when threshold value is 1. When these packets are passed through the filter, then the values of the counter in hash table is incremented. When the counter value goes above the threshold value then some of the legitimate packets are dropped and some of the spoofed packets are accepted.

As the number of attackers is increased, there is a slight decrease in the acceptance ratio of legitimate packets. The solid lines in Figure 6.4, 6.5,6.6 shows the acceptance ratio of packets of our proposed scheme and dashed lines shows the acceptance ratio of packets of MDADF method [27].

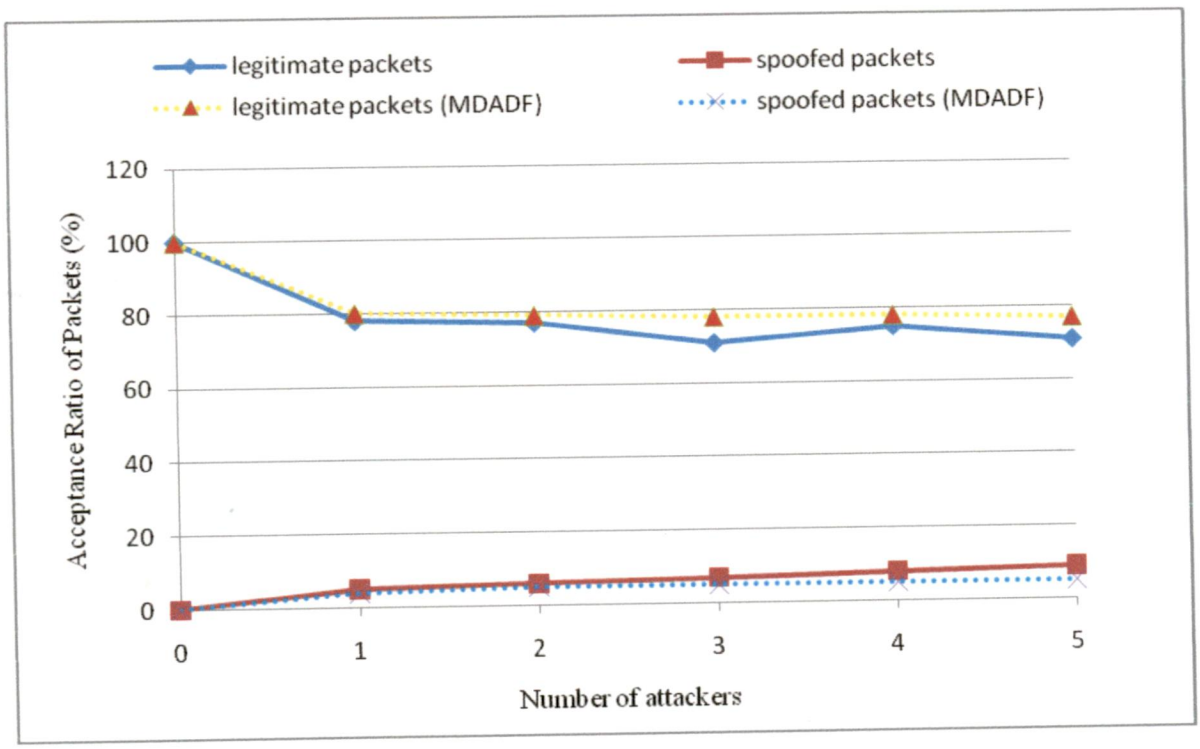

Figure 6.5: Acceptance ratio of packets vs. Number of attackers with threshold 2.

Figure 6.5 shows that the ratio of packets accepted under different magnitudes of attack when threshold value is 2. Here, we observe that around 20% of the legitimate packets are dropped and the acceptance ratio of spoofed packets is 7% which is only 1% more compared to spoofed packets accepted at threshold 1. It can also be observed that there is an increase in acceptance of the number of legitimate packets.

Figure 6.6 shows the acceptance ratio of packets under different magnitudes of attack for threshold value 3. Here we observed that only around 9% of the legitimate packets are dropped and the acceptance ratio of spoofed packets is 9% which is only 2% more compared to spoofed packets accepted at threshold 2.

These experiments show that as the threshold value increases the acceptance ratio of legitimate packets also increases.

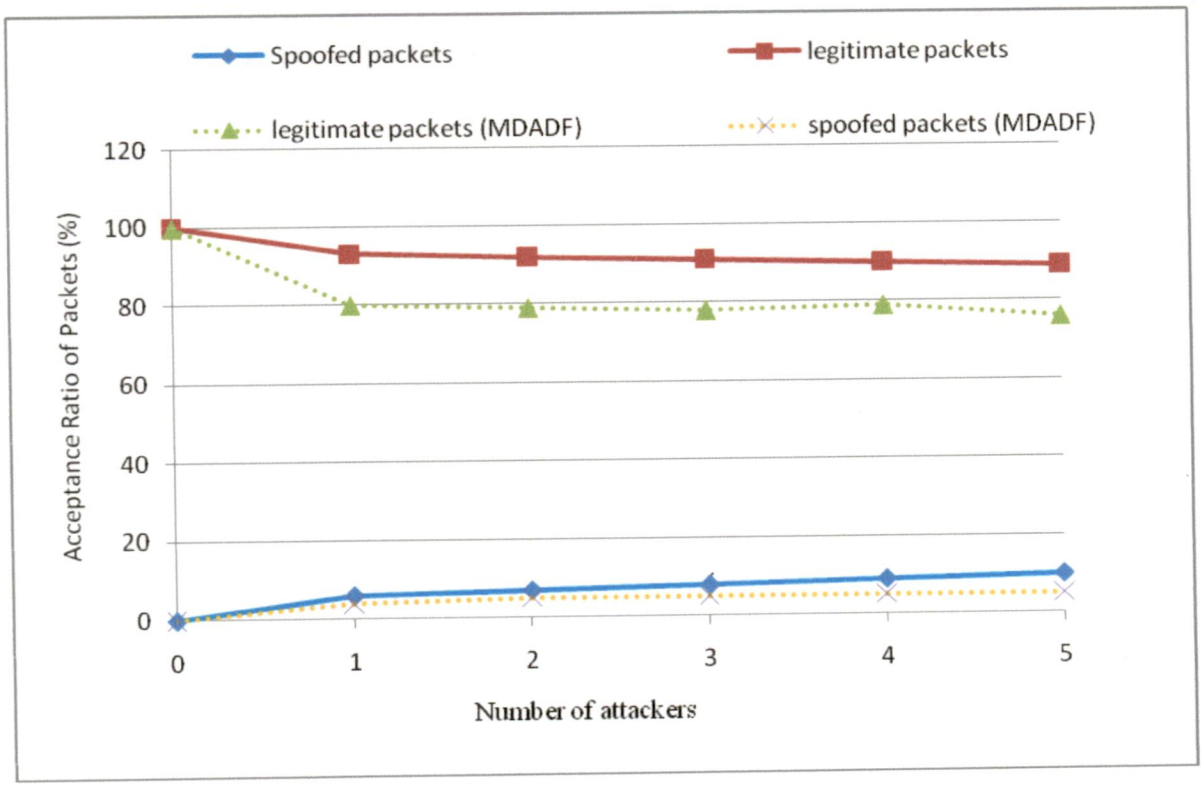

Figure 6.6: Acceptance ratio of packets vs Number of attackers with threshold 3.

Figure 6.7 shows, the mean false positive rate of our proposed scheme under different magnitudes of attack for different threshold values of counter. A false positive occurs when a benign event is declared as an attack. By observing figure 6.7 we say that the false positive rate of the filtering procedure decreases when threshold value increases.

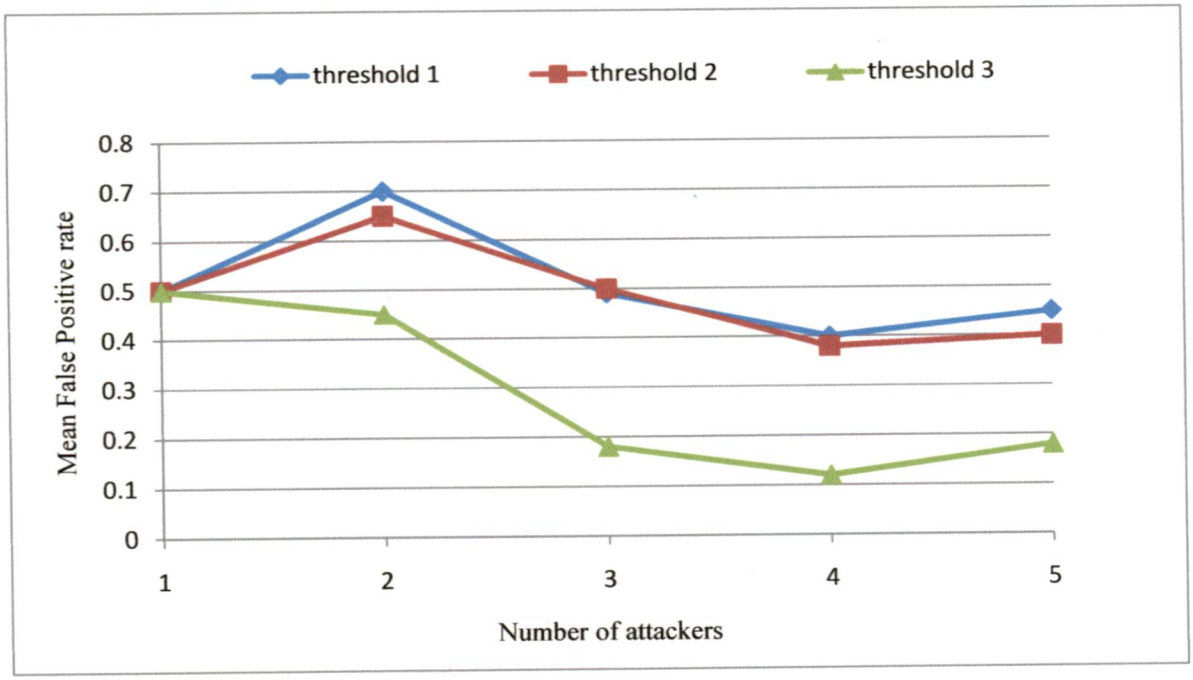

Figure 6.7: Mean false positive rate vs Number of attackers for different threshold

6.2 Comparison of various existing schemes with the proposed scheme

Our method, which uses Bloom-Filters for filtering process, is best suitable for small scale as well as large scale DDoS attacks because edge router on the path participates cooperatively at both sides. Table 6.1 shows the comparison between the proposed methods and the existing mechanisms based on different parameters. Following are the different parameters we have used to compare the proposed and existing mechanisms.

I. Support for Filtering

II. Support for Traceback

III. Small scale DDoS attacks prevention

IV. Large scale DDoS attacks prevention

V. Work fine for any path length of routers

Method	I	II	III	IV	V
FDS [16]		X		X	X
HCF [17]	X		X	X	X
SYN cache & cookies [18]	X		X		
SOS [19, 20]	X			X	X
SPIE [12]		X	X	X	
MDADF [27]	X		X	X	X
Proposed scheme using Bloom-Filters with hash function	X		X	X	X

Table 6.1: Comparison of various existing schemes with the proposed scheme

From Table 6.1 we can conclude that our proposed mechanisms have overcome some of the major limitations of the existing mechanisms. For example, it does not update the old entries in the hash table for the new entries of the packets, when the attacks flow is too large then it has enough resource to handle it.

CHAPTER 7
CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

Efficiency and scalability are the key requirements in design of defense against DDoS Attacks. The scheme proposed as a part of the dissertation provides an end-to-end solution for defense against DDoS attacks. Very little attention has been paid on defense using valuable information from innocent client whose IP is utilized in attacking packets.

The dissertation proposes a novel scheme for detecting DDoS attacks which is based on Bloom Filter. The scheme proposed is space efficient and computation efficient. The scheme uses fixed length table data structure to monitor traffic at the source end which is efficient in terms of space. It is simple and efficient detection scheme which have little computation overheads. The client and the server cooperate efficiently to issue warning of a DDoS attack. The system makes use of a hash table, derived from the Bloom filter, in order to monitor the three-way handshake.

Our proposed mechanism has overcome some of the limitations like it has the feature of filtering mechanisms; it greatly reduces the memory, storage cost and computational overheads at the edge routers by using simple calculations and also reduces the occurrence of hash collisions. By using this mechanism we can detect the attack DDoS attack at an early stage.

7.2. Suggestions for Future Work

The assumptions that the client detector initializes the filter area of the packet can be discarded and an effective strategy can be implemented to initialize the marking area of the packet when bloom-filters are used for traceback process.

The highlight of this technique is its simplicity which in turn makes it highly efficient in terms of computational cost as well storage space. The detection is done in real-time.

The scheme can be easily implemented at edge router of the client, without any need for change in the existing transport or network protocols.

This scheme work fine for high rate attacks. Moreover we are successful in detecting and filtering attacks; however our defense can be extended in multiple ISPs and the scheme can be used for the traceback for the source of attacks. In future work the detection scheme will be applied to real internet to evaluate the feasibility and effectiveness.

REFERENCES

[1] S. Moore, "Evolution of the Internet," Electro International 1994, Hynes Convention Center, Boston, MA, pp. 263–265. May. 1994.

[2] Robert Vamosi, "Study: DDoS attacks threaten ISP infrastructure," Online at http://news.cnet.com/8301-1009_3-10093699-83.html, CNET News, Nov. 2008.

[3] Elinor Mills, "Radio Free Europe DDOS attack latest by hactivists," Online at http://news.cnet.com/8301-10784_3-9933746-7.html, CNET News, May. 2008.

[4] Christos Douligeris and Aikaterini Mitrokotsa, "DDoS Attacks And Defencemechanisms: A Classification," *in Proceedings of the 3rd IEEE International Symposium on Signal Processing and Information Technology,* (ISSPIT'03), pp. 190-193, Dec 2003.

[5] Rocky K. C. Chang," Defending against Flooding-based Distributed Denial-of-service Attacks: A Tutorial," IEEE Communications Magazine, pp. 42-51, Oct. 2002.

[6] Internet World Stats, Internet User Statistics – The Big Picture: World Internet Users and Population Stats, http://www.internetworldstats.com/stats.htm

[7] L.D. Stein, J.N. Stewart, The World Wide Web Security FAQ, version 3.1.2, February 4, 2002, Available from http://www.w3.org/Security/Faq.

[8] <http://cisco.com> viewed on 15 may 2010.

[9] S. M. Khattab, C. Sangpachatanaruk, R. Melhem, D. Mosse, and T. Znati, "Proactive Server Roaming for Mitigating Denial-of-Service Attacks," in *Proceedings of the 1st International Conference on International Technology: Research and Education (ITRE'03),* pp. 286-290, Aug. 2003.

[10] A. Yaar, A. Perrig, and D. Song, "PI: A path identification mechanism to defend against DDoS attacks," in *proceedings of the IEEE symposium on Security and Privacy*, pp. 93-109, May 2003.

[11] P. Feruson and D. Seine, "Network Ingress Filtering: Defeating Denial Of Service Attacks Which Employ IP Source Address Spoofing," RFC2827, May 2000.

[12] A. C. Snoeren, C. Partridge, L. A. Sanchez, C. E. Jones, F. Tchakountio, B. Schwartz, S. T. Kent, and W. T. Strayer, "Single-Packet IP Traceback," IEEE/ACM Transactions on Networking, Vol. 10, No. 6, pp. 721-734, Dec. 2002.

[13] J.Ioannidis and S. M. Bellovin, "Implementing Pushback: Router-Based Defense Against DDoS Attacks," in *Proceedings of the Network and Distributed System Security Symposium (NDSS'02),* pp. 6-8, Feb. 2002.

[14] Yao Chen1, Shantanu Das, Pulak Dhar, Abdul-motaleb El Saddik, and Amiya Nayak, "Detecting and Preventing IP-spoofed Distributed DoS Attacks," *International Journal of Network Security*, Vol.7, No.1, pp.70–81, Jul. 2008.

[15] B. Bloom, "Space/Time Trade-Offs in Hash Coding with Allowable Errors," Comm. ACM, vol. 13, no. 7, pp. 422-426, 1970.

[16] Wang H, Zhang D, Shin KG (2002) Detecting SYN flooding attacks. In: *Proceedings of Annual Joint Conference of the IEEE Computer and Communications Societies(INFOCOM),* vol. 3, pp 1530–1539

[17] Jin C,Wang HN, Shin KG (2003) Hop-count filtering: An effective defense against spoofed DDoS traffic. *In: Proceedings of the 10th ACM Conference on Computer and Communication Security (CCS), ACM* Press, pp 30–41

[18] J.Lemon, "Resisting SYN Flooding Dos Attacks with A SYN Cache", *Proceeding of USENIX BSDCon*'2002,February, 2002.

[19] Keromytis A, MisraV, RubensteinD(2002) SOS: Secure overlay services. *In:ACMSIGCOMMComputer Communication Review, Proceedings of the 2002 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, Pittsburgh, PA*, vol. 32, pp 61–72

[20] Keromytis A, Misra V, Rubenstein, D (2004) SOS: An architecture for mitigating DDoS attacks. *IEEE Journal on Selected Areas in Communications* 22:176–188

[21] A. C. Snoeren. Hash-based IP traceback. In *Proceedings of the ACM SIGCOMM Conference*, pages 3–14. ACM Press, August 2001

[22] Bin Xiao · Wei Chen · Yanxiang He A novel approach to detecting DDoS attacks at an early stage* C_ Springer Science+Business Media, LLC 2006

[23] NS-2 Network Simulator, available at, http://www.isi.edu/nsnam/ns/, 2006, last accessed on 10-11-2009.

[24] Chan, E., Chan, H., Chan, K., Chan, V., Chanson, S., etc.: IDR: an intrusion detection router for defending against distributed denial-of-service(DDoS) attacks. *In: Proceedings of the 7th International Symposium on Parallel Architectures, Algorithms and Networks* 2004(ISPAN'04). (2004) 581–586

[25] S. Abdelsayed, D. Glimsholt, C. Leckie, S. Ryan, and S. Shami. An efficient filter for denial-of-service bandwidth attacks. In *IEEE Global Telecommunications Conference, 2003. GLOBECOM '03*, volume 3, pages 1353–1357, Dec 2003

[26] Bhavana Gandhi, R. C. Joshi, "An Integrated Framework for Proactive Mitigation, Characterization and Traceback of DDoS Attacks," *International Journal of Computer Science and Network Security IJCSNS*, Vol. 7, No. 3, March 2007, pp. 274-282.

[27] Yao Chen1, Shantanu Das, Pulak Dhar, Abdul-motaleb El Saddik, and Amiya Nayak, "Detecting and Preventing IP-spoofed Distributed DoS Attacks," International Journal of Network Security, Vol.7, No.1, pp.70–81, Jul. 2008.

LIST OF PUBLICATIONS

[1] Neha Gupta, "Detecting SYN Flooding Attack at an Early Stage*," In the Proceedings of International Conference on Advance in Commmunication, Network, and Computing (CNC'2010), Calicut (Kerala), India, Oct. 2010. (Accepted).

APPENDIXES

Network Simulator: NS-2 [23]

The *NS-2* simulator is a discrete event simulator widely used in the networking research community. It was developed at the University of California at Berkeley and extended at Carnegie Mellon University to simulate wireless networks. These extensions provide a detailed model of the physical and link layer behavior of a wire/wireless network and allow arbitrary movement of nodes within the network. NS-2 Simulator is usually a software package that simulates a real system scenario. Through the simulation we can test how a device or a system will perform in terms of timing and result. In addition to that it can be used to explore new policies, operating procedure without interrupting the system in real time.

Network simulation is very important because the network designer can test a complex network and make the right decisions about the designing in order the network will not face any problems in the future. New network devices can be added and testing without disturbing the existing network. NS-2 has high performance and it is very easy to use because of the combination of two languages. NS architecture follows the OSI model.

Node in a network is a point that connects other points, either a distribution point or an end point for data transmissions. A node can send or receive data. All kind of nodes in ns-2 are separated in two types of nodes. A unicast node that sends packets to only one node and a multicast node that sends packets to more than one node. Attack traffic source node is a node that sends malicious data (spoofed data) to other nodes. Traffic agents such as TCP or UDP are assigned on those nodes.

The receiving node is called sink node. A sink node can be an end node of the network. It can receive from different type of traffic source node. In case of a sink receives data form a TCP traffic node is defined as Agent/TCP Sink and as Agent/Null if it is received from a UDP traffic node.