PROTECTION FROM DISTRIBUTED DENIAL OF SERVICE (DDoS) ATTACKS IN ISP DOMAIN

A THESIS

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

DOCTOR OF PHILOSOPHY

in

ELECTRONICS AND COMPUTER ENGINEERING

By

KRISHAN KUMAR

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

JULY, 2007

INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the thesis entitled **PROTECTION FROM DISTRIBUTED DENIAL OF SERVICE (DDoS) ATTACKS IN ISP DOMAIN** in partial fulfilment of the requirements for the award of the degree of Doctor of Philosophy and submitted in the Department of Electronics and Computer Engineering of the Indian Institute of Technology Roorkee, Roorkee, is an authentic record of my own work carried out during the period from July, 2004 to July, 2007 under the supervision of Dr. R. C. Joshi, and Dr. Kuldip Singh, Professors, Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, Roorkee.

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other institute.

K. Saluja

(KRISHAN KUMAR)

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

(Dr. R. C. Joshi)
Professor
(Supervisor)

(Dr. Kuldip Singh)
Professor
(Supervisor)

Date: 23.7.07

The Ph.D. Viva-Voce Examination of **Mr. Krishan Kumar**, Research Scholar, has been held on 23/2/08

Signature of Supervisors

Signature of External Examiner

Abstract

The Internet is used extensively for important services such as banking, transportation, medicine, education, stock trades, defense, etc. Most of these transactions must be processed in a timely manner. However, these services are delayed, degraded and sometimes completely disrupted because of attacks on the Internet. The inherent vulnerabilities of the Internet architecture provide opportunities for a lot of attacks on its infrastructure and services. The problem is aggravated because of huge base of unprotected hosts on the Internet. These hosts are used in an unauthorized manner by attackers, as slaves called zombies, to launch attacks against high profile sites. Flooding Distributed denial-of-service (DDoS) is one such kind of attack in which a large number of unwitting hosts are used as an army against the victim site.

Flooding DDoS attacks consist of an overwhelming quantity of packets being sent from multiple attack sites to a victim site. These packets arrive in such a high quantity that some key resource at the victim (bandwidth, buffers, CPU time to compute responses) is quickly exhausted. The victim either crashes or spends so much time handling the attack traffic that it cannot attend to its real work. Thus legitimate clients are deprived of the victim's service for as long as the attack lasts. While services are restored as soon as the attack subsides, the incidents still create a significant disturbance to the users and costs victim sites millions of dollars.

The traditional security technologies such as firewalls, Intrusion detection systems (IDSs) and access control lists in routers are unable to defend networks from these attacks. The stumbling barrier against these attacks is that it is almost impossible to differentiate

between genuine and attack packets. The seriousness of DDoS problem and growing sophistication of attackers have led to development of numerous defense mechanisms in research and commercial communities. In order to be effective, these defense mechanisms need global deployment, normal traffic models, infrastructural changes, and minimal collateral damage. However, these requirements are difficult to accomplish because of decentralized Internet management, unpredictable user behaviour and variety of network environments, sophisticated and user friendly attack tools, high computational overheads at core of Internet, and distributed nature of DDoS attacks.

In this study, an ISP domain has been chosen to place various defense nodes of the proposed system. This provides advantage of more resources to fight against DDoS attacks. Moreover, single administrative control in an ISP domain, allows defense nodes to collaborate in a cohesive manner to achieve synergistic effect. Transit-stub model of GT-ITM topology generator is adopted for creating simulation topology consisting of four ISPs. The major contributions of the work are as follows. The present work is divided into three parts.

In the first part, an overview of DDoS problem, its basic cause, DDoS defense challenges and principles are presented. Core problems in existing DDoS defense techniques are identified on the basis of common DDoS defense principles and an array of DDoS attack types.

Second part of the thesis proposes an automated approach to detect flooding DDoS attacks and filter attack traffic at ingress edges of the protected ISP domain. A time series analysis of observed traffic detects flooding DDoS attacks by characterizing asymmetry in traffic distributions. The approach is validated using simulations in NS-2 testbed. Low rate flooding DDoS attacks, which slowly degrade services to legitimate clients, are detected reliably and accurately. Simulation experiments carried out at various attack strengths show

detection of very meek rate attacks. High rate flooding DDoS attacks, which completely disrupt services to legitimate clients, are easily detected at point of presence (POP) near the victim in ISP domain. High rate attacks whose intensity per flow slowly rises are also detected at an early stage. So a proactive detection of high rate flooding DDoS attacks is also exhibited in the proposed approach, which helps in timely recovery from attack. The filtering of attack traffic is done at ingress links of POPs in the protected ISP domain to save core bandwidth and reduce filtering overheads at single point. The selection of detection threshold and its impact on detection accuracy is analyzed using receiver operating characteristics (ROC) curves. The comparison of legitimate service level achieved with volume based existing techniques manifests supremacy of the approach.

In the third part of the thesis, high computational overheads of analyzing flooding DDoS attacks near the victim are tackled by proposed distributed approach in ISP domain. Analytical solution well supported by simulation experiments is presented to distribute computational overheads of detection system among POPs of the ISP domain without compromising accuracy. The computational complexity of proposed distributed scheme at POP connected to victim server is very less as compared to existing schemes. It makes our approach robust against high volume and high computational overheads of monitoring and analsysing traffic near the victim. Errors are also computed by removing assumptions. Regression and correlation analysis is used to find relationship between number of zombies used to launch the attack and deviation from detection threshold. Standard error of estimate, sample coefficient of determination and coefficient of correlations are calculated to describe the relationship. A tolerance based proactive approach is proposed to regulate traffic such that server resources are allocated in a fair manner to all traffic sources under a high rate flooding DDoS attack. The proposed algorithms rate limit traffic at edges of protected ISP domain depending upon share of traffic passing through it.

Acknowledgements

I would like to express my deepest gratitude to my learned supervisors **Prof. R.C. Joshi** and **Prof. Kuldip Singh** for their encouragement, painstaking supervision, innovative suggestions and invaluable help during the entire period of my Ph.D. studies. It has been a great honor and pleasure for me to work under their supervision. I learned a great deal from them, not only about research but also about matters touching many other aspects which will benefit me in future.

Prof. R.C. Joshi possesses a rare talent of being flexible but not lenient. This gave me an unparalleled freedom to explore bold ideas and limits procrastination. He is very active and alert person who can listen to you at any time and react meaningfully all the times. Most importantly he helped me understand that worry is not directly proportional to effort. **Prof. Kuldip Singh's** valuable suggestions, encouragement and moral support in every phase of my Ph.D. made it possible for me to complete the work in the scheduled time.

The cooperation and help extended by the Head and faculty members, Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee is gratefully acknowledged. I particularly thank Prof. Manoj Misra, for his technical suggestions. My thanks are also due to my research committee members for their patience during evaluations and discussions.

I am highly obliged to the Principal, SBS College of Engineering and Technology, Ferozepur (Punjab) for sponsoring me under Quality Improvement Programme. I am also grateful to my colleagues Mr. Rakesh Kumar Sharma, Mr. Neel Kant Grover, and Mr. Kulbushan Agnihotri at my parent college for their direct or indirect support.

My friends Mr. Vineet Prabhakar, Mr. Mahesh Jat, Dr. Zahid, Mr. T. P. Sharama, and Mr. R. C. Gangwar at IIT Roorkee were of invaluable help. I would never have made it without them. They offered their shoulder whenever I wanted to complain about stress of life. Mr. Vineet Prabhakar, potentially a brilliant scholar made my life joyous and filled my days with laughter. Mr Mahesh Jat, a friend and a true critique always listened to my technical problems and suggested good solutions. The social support of Mr. Baldev Saluja, Mr. Sunil Kumar, and Sachdeva family at Ferozepur always made me feel secure and concentrate on work during my stay at Roorkee.

Finally, I could not reach the important milestone of my life without the support and encouragement of my family. Thanks to my parents who have made it possible for me to reach where I am today. I express my sincere appreciation and gratitude to my wife Monika and sons Kshitij and Soumil for their patience and encouragement when it was most required. My adorable wife has taken care of my parents and sons in a way that I was never worried about my home at Ferozepur. She has been a friend to me when I am under stress and colleague when I want technical suggestions.

And above all, I am thankful to the Almighty whose divine grace gave me the required courage, strength and perseverance to overcome various obstacles that stood in my way.

K. Saluja

Krishan Kumar

Contents

List of Figures

List of Tables

DiDDeM	Distributed DoS Detection Mechanism
DiffServ	Differentiated Services
DNS	Domain Name System
DoS	Denial-of-Service
DPM	Deterministic Packet Marking
DRDoS	Distributed Reflector Denial-of-Service
DRL	Dynamic Rate Limiting
DS	Differentiated Services
DSR	Dynamic Source Routing
DV	Distance Vector
D-WARD	DDoS Network Attack Recognition and Defense
EMERALD	Event Monitoring Enabling Responses to Anomalous Live Disturbances
EWMA	Exponential Weighted Moving Average
FBI	Federal Bureau of Investigation
FIFO	First in First out
FQ	Fare Queuing
FRED	Flow Random Early Drop
FTP	File Transfer Protocol
Gb	Giga Bits
GT-ITM	Georgia Tech Internetwork Topology Models
HIDS	Host-based Intrusion Detection System
HRFD	High Rate Flooding DDoS attacks
HTTP	Hyper Text Transfer Protocol
IAD	Internet Protocol Address Database

ICMP	Internet Control Message Protocol
IDIP	Intrusion Detection and Isolation Protocol
IDS	Intrusion Detection System
IntServ	Integrated Services
IP	Internet Protocol
IPSec	Internet Protocol Security
ISP	Internet Service Provider
iTrace	ICMP Traceback
LRFD	Low Rate Flooding DDoS attacks
LS	Link state
Mbps	Mega bits per second
MIB	Management Information Base
MULTOPS	MUlti-Level Tree for Online Packet Statistics
NAT	Network Address Translation
NIDS	Network-based Intrusion Detection System
NPSR	Normal Packet Survival Ratio
NS	Network Simulator
OTcl	Object Tcl
PD	Preferential Dropping
PDA	Personal Digital Assistant
PKI	Public Key Infrastructure
POP	Point of Presence
PPM	Probabilistic Packet Marking
QoS	Quality of Service

RED	Random Early Detection
ROC	Receiver Operating Characteristics Curve
RPF	Route based Packet filtering
RSVP	Resource reservation Protocol
RTT	Round Trip Time
SAVE	Source Address Validity Enforcement
SCO	Santa Cruz operation
SFQ	stochastic fair queuing
SOAP	Secure Overlay Access Points
SOS	Secure Overlay System
SPIE	Source Path Isolation Engine
SRED	Stabilized Random Early Detection
TCP	Transmission Control Protocol
TDSAM	TCP-Dropping Statistic Analysis Module
tIP	Transport IP
UDP	User Datagram Protocol
UIPF	Ubiquitous Ingress/Egress Packet Filtering
VBA	Volume Based Approach
VIP	Virtual Private Network
WATCHER	Watching for Anomalies in Transit Conservation: a Heuristic for Ensuring Router Security
WFSA	Weighted Fair Share Algorithm
WWW	World Wide Web

List of Abbreviations

ACC	Aggregate based Congestion Control
AIMD	Additive Increase Multiplicative Decrease
AODV	Ad hoc On Demand Distance Vector
ARP	Address Resolution Protocol
AS	Autonomous System
ASSYST	Active Security System
ATA	Algebraic based Traceback
BA	Bandwidth Aggregate
BGP	Border Gateway Protocol
BSD	Berkeley Software Distribution
CBQ	Class Based Queuing
CDN	Content Distribution Network
CERT	Computer Emergency Response Team
CHOKe	CHOose and Keep for Responsive Flows, CHOose and Kill for Unresponsive Flows
CNN	Cable News Network
COSSACK	Coordinated Suppression of Simultaneous Attacks
CSI	Computer Security Institute
CUMSUM	Cumulative Sum
D-DCFI	DDoS-Detection Characterization and Filtering in ISP Domain
DDoS	Distributed Denial-of-Service
DefCOM	Defensive Cooperative Overlay Mesh

DiDDeM	Distributed DoS Detection Mechanism
DiffServ	Differentiated Services
DNS	Domain Name System
DoS	Denial-of-Service
DPM	Deterministic Packet Marking
DRDoS	Distributed Reflector Denial-of-Service
DRL	Dynamic Rate Limiting
DS	Differentiated Services
DSR	Dynamic Source Routing
DV	Distance Vector
D-WARD	DDoS Network Attack Recognition and Defense
EMERALD	Event Monitoring Enabling Responses to Anomalous Live Disturbances
EWMA	Exponential Weighted Moving Average
FBI	Federal Bureau of Investigation
FIFO	First in First out
FQ	Fare Queuing
FRED	Flow Random Early Drop
FTP	File Transfer Protocol
Gb	Giga Bits
GT-ITM	Georgia Tech Internetwork Topology Models
HIDS	Host-based Intrusion Detection System
HRFD	High Rate Flooding DDoS attacks
HTTP	Hyper Text Transfer Protocol
IAD	Internet Protocol Address Database

Chapter 1

Introduction and Statement of the Problem

The phenomenal growth and success of Internet has changed the way traditional essential services such as banking, transportation, medicine, education and defense are operated. Now they are being progressively replaced by cheaper and more efficient Internet-based applications. In present era, the world is highly dependent on the Internet and it is considered as main infrastructure of the global information society. Therefore, the availability of Internet is very critical for the socio-economic growth of the society. However, the inherent vulnerabilities of the Internet architecture provide opportunities for a lot of attacks on its infrastructure and services. Distributed denial-of-service (DDoS) attack is one such kind of attack, which poses an immense threat to the availability of the Internet. In this chapter, we shall provide an introduction to DDoS attacks, motivation for DDoS defense and finally statement of problem with a strategy followed in solving the problem.

1.1 Introduction

The "availability" means that the information, the computing systems, and the security controls are all accessible and operable in committed state at some random point of time [67]. Threat to the Internet availability is a big issue which is hampering growth and survival of E-business and other Internet based applications. The Internet like any other product is also prone to failures. Internet failures can be accidental or intentional. The Internet design concentrates mainly on providing functionality though a little attention has

been given on designing strategies for controlling accidental failures. On the other hand, intentional attacks by malicious users/hackers/crackers have no answer in the original Internet design. A denial-of-service (DoS) is such an intentional attempt by malicious users/attackers to completely disrupt or degrade (compromise) availability of service/resource to legitimate/authorized users [37]. Some well-known DoS attacks are SYN Flood, Teardrop, Smurf, Ping of Death, Land, Finger Bomb, Black Holes, Octopus, Snork, ARP Cache Poisoning and the Misdirection. DoS attacks exploit weaknesses in Internet protocols, applications, operating systems, and protocol implementation in operating systems.

Distributed denial-of-service attacks (DDoS) degrade or completely disrupt services to legitimate users by expending communication and/or computational resources of the target. DDoS attacks are amplified form of DoS attacks where attackers direct hundreds or even thousands of compromised hosts called zombies against a single target [89, 132]. These zombie hosts are unwittingly recruited from the millions of unprotected computers accessing the Internet through high-bandwidth and always available connections.

There are varieties of DDoS attacks as classified in [89, 22]. However, the most common form of DDoS attacks is a packet-flooding attack, in which a large number of seemingly legitimate TCP, User Datagram Protocol (UDP), or Internet Control Message Protocol (ICMP) packets are directed to a specific destination. Defending against these attacks is challenging for two reasons [154]. First, the number of zombies involved in a DDoS attack is very large and deployment of these zombies spans large geographical areas. The volume of traffic sent by a single zombie might be small, but the volume of aggregated traffic arriving at the victim host is overwhelming. Second, zombies usually spoof their IP addresses under the control of attacker, which makes it very difficult to trace the attack

traffic back even to zombies. According to the Internet architecture working group [106], the percentage of spoofed attacks is declining, but the sheer volume and distributed nature of DDoS attack traffic still thwart design of an effective defense.

1.2 Motivation

The attacker/malicious users waste their energy and effort to create attack network called botnet, which comprises of weakly secured machines to launch such attacks. The main motives behind DDoS attacks are either of criminal, commercial or ideological nature. Broadly speaking, there are usually four types of attackers:

- Criminals who blackmail their victims and demand high ransom payments.
- Competitors who aim to damage their rivals business and reputation.
- Terrorists who carry out ideologically motivated attacks.
- Script kiddies who are testing their abilities or for publicity.

Extremely sophisticated, user friendly and powerful DDoS toolkits [41, 42, 43, 44, 77, 123] are available to potential attackers increasing the opportunity of launching of these attacks. The DDoS attack tools are so simple to use that nothing more than the whim of a 13-year old hacker is required to knock any user site, or server off the Internet. Moreover, DDoS attacking programs have very simple logic structures and small memory sizes making them relatively easy to implement and hide. Therefore, DDoS has emerged as the weapon of choice for disruption on the Internet.

Various DDoS attacks against high-profile websites such as Yahoo, CNN Amazon and E Trade in early 2000, series of attacks on grc.com in May, 2001 [143] and mydoom virus attack on SCO website in Feb. 2003 demonstrate how devastating DDoS attacks are and how defenseless the Internet is under such attacks. The services of these websites were unavailable for hours or even days as a result of these attacks. Therefore, the already grown

dependence on the Internet makes the impact of successful DDoS attacks, financial and otherwise increasing painful for service providers, enterprises, and government agencies. Beginning from simple DoS security incidents, some of other well known packet flooding attacks and their impact are given below.

Real DoS incidents in the Internet between the years 1989 and 1995 were investigated in [82]. The three most typical effects were the following: 51% of these incidents filled a disk, 33% of the incidents degraded network service, and 26% of the incidents deleted some critical files. A single incident was able to cause several types of damages at the same time (the sum of percentages is more than 100%).

The first reported large-scale DDoS attack occurred in August, 1999, against a university [103]. This attack shut down the victim's network for more than two days. In February 7, 2000, several Web sites were attacked, which caused them to go offline for several hours [103]. In some cases these DDoS attacks were able to produce about 1 Gbit/s of attack traffic against a single victim [94].

The backscatter analysis was used to assess the number, duration, and focus of DoS attacks in the Internet [51]. Backscatter is called the unsolicited response traffic which the victim sends in response to attack packets with spoofed IP source addresses. The results indicate more than 12,000 attacks against more than 5,000 distinct victims during the 3-week period examined in February, 2001.

The Coordination Center of the Computer Emergency Response Team (CERT) was attacked in May, 2001. A DDoS attack caused its web site to be available only intermittently for more than two days [74].

The Domain Name System (DNS) is a continuous target for DoS attacks. In October, 2002, all root name servers experienced an exceptionally intensive DoS attack. Some DNS

requests were not able to reach a root name server due to congestion caused by the DoS attack. Another major DoS attack on June 15, 2004 [25], against name servers in Akamai's Content Distribution Network (CDN) blocked nearly all access to many sites for more than two hours. The affected sites included Apple Computer, Google, Microsoft, and Yahoo. These companies have outsourced their DNS service to Akamai to enhance service performance.

In UK online bookmaking, betting, and gambling sites have been extorted with DoS attacks during 2004 by unidentified attackers [15]. The Internet-based business service of Al Jazeera was brought down due to a DoS attack in January, 2005 [71]. Al Jazeera provides many Arabic-language news services. The text-to-speech translation application running in the Sun Microsystem's Grid Computing system was disabled with a DoS attack in March, 2006 [120]. This attack was carried out during the opening day of this service.

Using updated backscatter analysis [50], presence of roughly 2000-3000 active DoS attacks are established per week. The study of attacks over a three-year period revealed 68,700 attack on over 34,700 distinct Internet hosts belonging to more than 5,300 distinct organizations.

As proof of these disturbing trends, 2003 to 2006 FBI/CSI surveys [39, 100] concluded that DDoS attacks are one of the major causes of financial losses resulting from cyber crime.

The traditional security technologies such as firewalls [36, 113, 130], Intrusion detection systems (IDSs) [55, 174] and access control lists [144] in routers are unable to defend networks from these attacks. The stumbling barrier against these attacks is that it is almost impossible to differentiate between genuine and attack packets. Since the potency of flooding DDoS attacks does not depend upon exploitation of software bugs or protocol

vulnerabilities, it only depends on the volume of attack traffic. Consequently, flooding DDoS packets do not need to be malformed, such as invalid fragmentation field or a malicious packet payload. As a result, the flooding DDoS traffic looks very similar to legitimate traffic [91]. Also IP spoofing [19] and stateless routing reduces the chances of attacker being caught. Moreover, flooding DDoS attacks are very dynamic to elude existing defense systems [60, 89]. Therefore, it has become a real challenge to defend against these attacks. The seriousness of DDoS problem and growing sophistication of attackers have led to development of numerous defense mechanisms [22, 89]. These defense mechanisms are classified into four broad categories in [22]:

- Prevention
- Detection and characterization
- Traceback
- Tolerance and Mitigation

Attack prevention schemes [87, 98, 109, 119, 165, 167, 171] aim to either stop IP spoofing or strengthen the hosts by fixing security holes, such as insecure protocols, weak authentication schemes and vulnerable computer systems, which can be used as stepping stones to launch a DDoS attack. This approach aims to improve the global security level and is the best solution to counter DDoS attacks in theory. However, the disadvantage is that it needs global cooperation to ensure its effectiveness, which is extremely difficult in reality. Hence, the challenge is how to develop a scalable mechanism with low implementation cost. Attack detection aims to detect an on going DDoS attack and characterization helps to discriminate attack traffic from legitimate traffic. The challenge is how to detect every attack quickly without misclassifying any legitimate traffic. Traceback aims to locate the attack sources regardless of the spoofed source IP addresses in either process of attack

(active) or after the attack (passive). Stateless nature of IP routing further helps the cause of attackers. It is a crucial step to minimize the attack damage and provide deterrence to potential attackers. The challenge for attack traceback techniques [1, 3, 4, 6, 14, 40, 53, 86, 111, 131, 137, 147, 157, 158, 162] is how to locate attack sources quickly and accurately without changing current Internet infrastructure at minimum possible overheads. Overall research direction in this field has been limited mostly to finding zombies and path characterization up to zombies.

Tolerance and mitigation aims to eliminate or curtail the effects of an attack and try to maximize the quality of services under attack. The challenge for tolerance is how to filter the attack traffic without disturbing legitimate traffic.

The main aim of a DDoS defense system is to relieve victim's resources from high volume of counterfeit packets sent by attackers from distributed locations, so that these resources could be used to serve legitimate users [108]. In order to achieve it, timely detection of attack and accurate characterization of attack traffic followed by complete filtering or rate limiting of suspicious traffic is required. Though, detection of high rate flooding DDoS attacks is easy at the victim end and the loss is also maximum here, so victim end defense techniques [8, 9, 28, 31, 59, 85, 115, 150, 176] have maximum motivation of deployment. But due to excessive amassed DDoS traffic at the victim end, response is initiated manually in most of the cases. The offending packets actually consume the finite bandwidth available on the connection between victim end and the ISP. Even traditional security technologies [36, 55, 113, 130, 144, 174] are not able to escape from this problem.

The work in this thesis mainly concentrates on detecting flooding DDoS attacks, characterizing attack flows and filtering attack flows in ISP domain. In addition, a tolerance

based scheme to fairly allocate victim's resources to legitimate flows has also been proposed.

The special feature of current DDoS attack packets is that each packet is perfect legitimate packet but in combination, correlating these packets monitored at different network locations can give some signs of uniqueness from legitimate packets. The schemes [10, 26, 69, 70, 76, 93, 102, 116, 124, 133, 153] aim to detect DDoS attack by analyzing network based traffic and try to find deviation from normal behaviour. In almost all of these schemes the common challenge for DDoS detection system is that it is difficult or impossible for the training data to provide all types of normal traffic behavior [60, 89]. As a result, a lot of false alarms called false positives are generated when there is actually no attack. To minimize false positive rate, a larger number of parameters are used to provide more accurate normal profiles. However, with the increase in the number of parameters, the computational overhead to detect attack increases. This becomes a bottleneck, especially for volume-oriented DDoS attacks that are aggravated by the computational overhead of the detection scheme. In case of high rate flood based DDoS attacks, the detection system itself becomes a victim. So a DoS resistant detection System is choice of the hour at the moment. Moreover, detecting low rate attacks and characterizing low rate malicious flows accurately is not completely addressed in detection schemes [89]. The study in this area is totally disarrayed i.e. different detection and characterization methods are proposed using different topologies and different attacks [60]. Moreover, detection parameter variations and their effect on detection performance are not highlighted in existing methods. Even benchmarks and common evaluation criteria to compare existing approaches do not exist [60, 92].

The last but mostly used strategy assumes that because of limitations of prevention, detection, and finally tracing, it is almost impossible to prevent, accurately detect and

characterize, and trace back to ultimate attacker when attack is in progress. Tolerance and mitigation schemes either allocate resources of the victim in a fair manner or thrive to rate-limit traffic from malicious and mostly aggressive sources. Though FQ [7, 72], SFQ [121], QoS based techniques [97, 138] are good solutions as also proposed by Crocker [149] to allocate victim resources in fair manner, but excessive state monitoring, calculation of proper rate limits and testing for defaulters cause appreciable overheads considering rich resource based Internet of present age used for launching flood based attacks. Router based solutions like ACC [84, 129], are available for detecting high bandwidth aggregates based on destination address. Collateral damage measured in terms of ratio of number of legitimate packets to total packets received at victim called normal packet survival ratio (NPSR), is too high because of difficulty in finding narrow attack signatures and distributed nature of attack sources. A router throttle [46] based scheme however performs better than [84]. But in case of flash crowds [75] and very meek rate geographically distributed attacks, NPSR is not sufficient. In server based DDoS attacks, servers are normally attached near the backbone routers so that the flood traffic cannot create congestion at access link. However in this case, the processing capacity of the server becomes target of DDoS attackers. Proactive server roaming [30, 146] and SOS [6] works very well in providing service to legitimate clients, but requirement of client end programs to collaborate overall defense, has really hampered their popularity. DefCOM [62] claims to provide motivation for wide deployment by proposing an economic model both for victims and sources and warrant further research into cooperative DDoS defense.

Despite a significant breadth of research into countermeasures, DDoS attacks remain a major threat today [126]. So an attempt to contribute towards DDoS defense is need of the hour.

1.3 Statement of the Problem

The proposed work is to develop a real time detection system, which can generate an automated response to DDoS attacks and maintain legitimate traffic level as per quality of service (QoS) requirements at the victim. The problem can be divided into following sub problems.

1. Detection of flooding DDoS attacks in ISP domain and automated filtering attack packets at ISP boundary to provide reasonable performance in terms of normal packet survival ratio (NPSR) at the victim.
2. Excessive traffic of high rate flooding DDoS attacks results in high overheads of monitoring and analyzing traffic at single point of presence (POP) of the ISP domain. A distributed approach is proposed to minimize computational overheads at a single POP connected to victim by distributing overheads at all POPs of the ISP without compromising detection accuracy.
3. A relationship between number of non spoofed zombies and deviation from detection threshold is proposed using regression analysis for making a real time estimate of number of non spoofed zombies used to launch flooding DDoS attack.
4. A tolerance based proactive approach is proposed to regulate traffic at edges of network such that server resources are allocated in a fair manner to all traffic sources under a high rate flooding DDoS attack.

A general strategy to solve the above said problems is as follows:

1. Investigating the key existing methods of DDoS Defense.
2. Exploring the major inadequacies of the existing systems.
3. Proposing an efficient method that addresses the gaps in the existing methods.
4. Exploring the correctness of the proposed approach.
5. Evaluating the performance using NS-2 simulator test bed.

1.4 Organization of the Thesis

The thesis is organized as follows. Chapter 2 presents a brief overview and literature survey on DDoS defense. An automated approach to detect and react to DDoS attacks in ISP domain is presented in chapter 3. Chapter 4 discusses single point monitoring and analysis problems, and demonstrates a distributed approach to tackle the problem both analytically and experimentally. In chapter 5, a regression and correlation based analysis is used to predict number of zombies. Chapter 6 presents a Dynamic rate limiting based approach to minimize impact of DDoS attacks. Chapter 7 finally concludes the thesis by presenting our contributions and directions for the future work.

Chapter 2

Literature Survey

DDoS attacks appeared as a serious threat to the Internet in 1999 and since then many security efforts aim to constrain their affects are made. Unfortunately, the attackers closely follow developments in the security field and are often able to modify features of attacks and thus bypass the security system. We present here an overview of DDoS problem, and discuss strengths and weaknesses of state-of-art mechanisms based on common defense principles and an array of DDoS attack types. A summary of pending concerns highlights core problems in existing techniques.

2.1 An Overview

Operating systems and network protocols are developed without applying security engineering which results in providing hackers a lot of insecure machines on Internet. These insecure and unpatched machines are used by DDoS attackers as their army to launch attack. An attacker or hacker gradually implants attack programs on these insecure machines. Depending upon sophistication in logic of implanted programs these compromised machines are called Masters/Handlers or Zombies and are collectively called bots and the attack network is called botnet in hacker's community. Hackers send control instructions to masters, which in turn communicate it to zombies for launching attack. The zombie machines under control of masters/handlers (running control mechanism) as shown

in Figure 2.1 transmit attack packets, which converge at victim or its network to exhaust either its communication or computational resources.

DDoS attacks can be classified into two broad categories: flooding attacks and vulnerability attacks [89]. Flooding DDoS attacks consume resources such as network bandwidth by overwhelming bottleneck link with a high volume of packets. Vulnerability attacks use the expected behaviour of protocols such as TCP and HTTP to the attacker's advantage. The computational resources of the server are tied up by seemingly legitimate requests of the attackers and thus prevent the server from processing transactions or requests from authorized users.

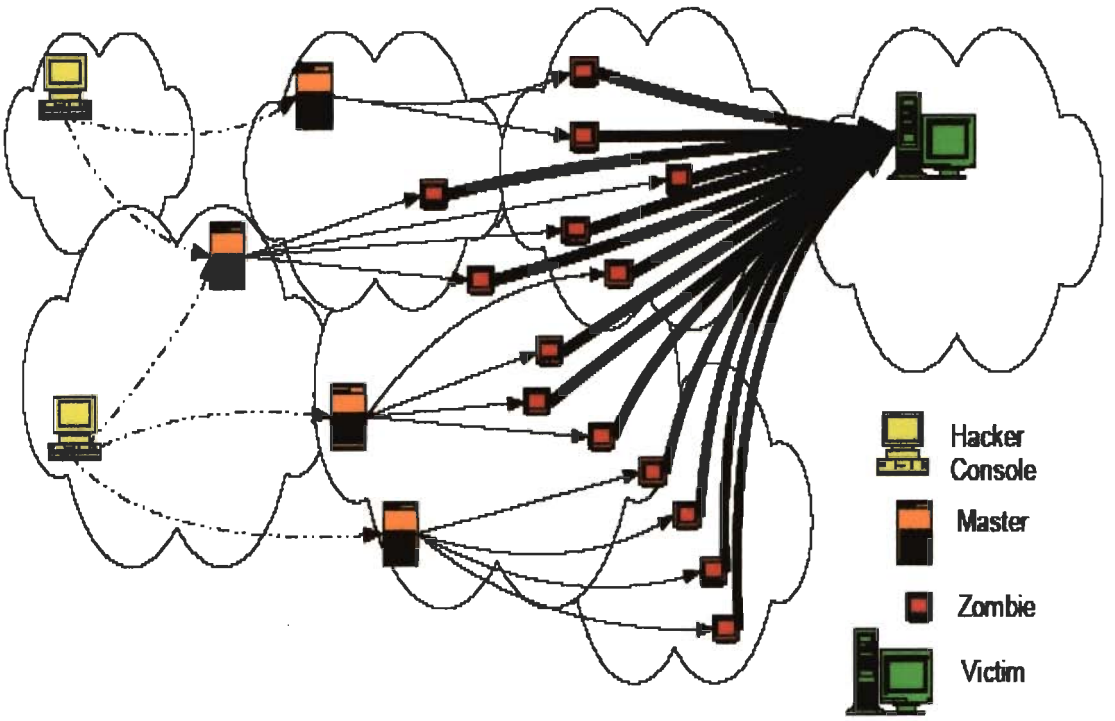

Figure 2.1: Attack Modus Operandi

Flooding DDoS is basically a resource overloading problem. The resource can be bandwidth, memory, CPU cycles, file descriptors and buffers etc. The attackers bombard the scarce resource(s) by sheer flood of packets.

In Figure 2.2, a flood of packets is shown, which congests the link between ISP's edge router and border router of victim domain. Attack packets keep coming as per distribution fixed by attacker, whereas legitimate clients cut short their packet sending rates as per flow control and congestion signals in [107, 140]. A situation comes when whole of bottleneck bandwidth is seized by attack packets. Thus, service is denied to legitimate users due to limited bottleneck bandwidth.

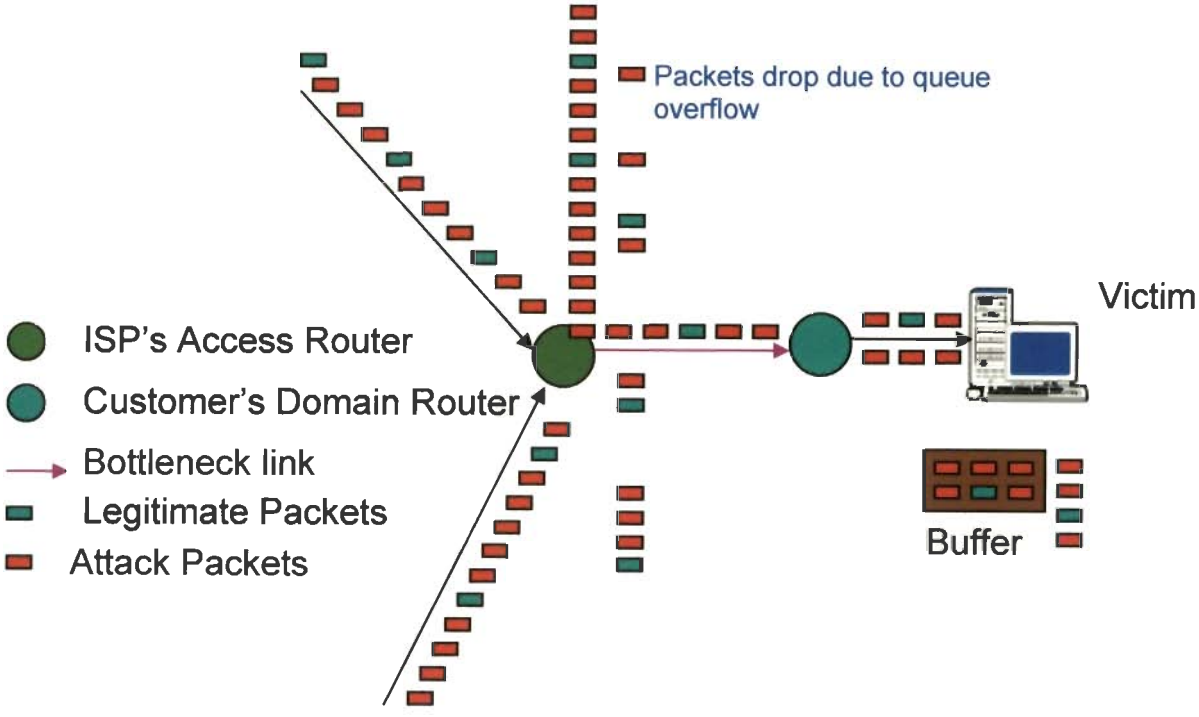

Figure 2.2: Packets drop under DDoS attack

However, resources of connecting network are not a problem in case of commercial servers as these are hosted by the ISPs, quite close to their backbone network with high bandwidth access links. But server resources such as processing capacity, buffer limit etc., are put under stress by flood of seemingly legitimate requests generated by DDoS attack zombies. Each request consume some CPU cycles. Once the total request rate is more than the service rate of server, as shown in Figure 2.2, the requests start getting buffered in the server, and

after some time due to buffer over run, incoming requests are dropped. The congestion and flow control signals force legitimate clients to decrease their rate of sending requests [107, 140], whereas attack packets keep coming. Finally, a stage comes when only attack traffic is reaching at the server. Thus, service is denied to legitimate clients. Moreover, as attack strength grows by using multiple sources, the computational requirements of even filtering traffic of malicious flows become a burden at the target [154].

The design of Internet architecture has many constraints, which actually provide motivation for attackers to launch flooding DDoS attacks. The protection of Internet from DDoS attacks i.e. DDoS defense has to compromise with these Internet design constraints and still provide a solution, which can offer Internet services to legitimate clients as per QoS requirements. The vulnerabilities in the Internet design and DDoS defense are explained below.

2.1.1 Inherent Vulnerabilities of the Internet Architecture

The Internet was designed with functionality, not security, in mind [64]. So its architecture has some inherent weaknesses and bugs, which result in successful origin and execution of DDoS attacks. Some of these are detailed below [89, 154]:

- **Connectivity and resource-sharing** — The Internet is designed as an open public infrastructure to share information resources. This has two consequences. First, the potential victims, such as web servers, must connect to the Internet and be visible to the public in order to provide public service. The visibility is made via a globally routable IP address. Second, the Internet is based on packet-switching, unlike its counterpart, the public telecommunication networks, which are based on circuit-switching. For circuit-switched networks, each service (e.g. a phone call) is allocated a separate channel until the end of the service. A user's service is not being

interfered by other users' behavior. In contrast, for packet-switched networks, users share all the resources and one user's service can be disturbed by other users' behavior. Flooding attacks take advantage of these features. First, attack packets are delivered to the victim before knowing whether they are malicious or not. Second, by occupying most of the shared resources, flooding attacks manage to disrupt the services for the legitimate users.

- **Authentication, Integrity and Traceability** — The Internet is equipped with no inbuilt authentication scheme, which leads to a serious problem, called IP spoofing. IP spoofing [19] refers to creating an IP packet containing fake information. IP source address spoofing occurs when one IP packet is generated without using the source IP address that is assigned to the computer system. Without an integrity check for each IP packet, attackers can spoof any field of an IP packet and inject it into the Internet Moreover, the routers generally do not have packet tracing functions, for example, keeping all previous connection records. They only receive and forward the packets. In practice, this cannot be done due to the huge amount of traffic that needs to be stored. Therefore, once an IP packet is received by the victim, there is no way to authenticate whether the packet actually comes from where it claims and what it contains. By hiding their identities and integrity using IP spoofing, the attacker can launch flooding attacks without being responsible for the damage.
- **Internet security is highly interdependent** — The Internet is a huge community, where many insecure systems exist. Unfortunately, the number of vulnerabilities reported each year is increasing according to CERT statistics [38]. We can secure our system but we cannot force others to do so. Hence an attacker can control a large

number of insecure systems by exploiting their vulnerabilities. By launching flooding attacks from these controlled systems, the attack power is tremendously increased.

- **Intelligence and resources asymmetry** — Most of Intelligence needed for service guarantees is located in end hosts. But high bandwidth links and routers are in the intermediate network. So attackers can exploit the abundant resources of intermediate unwitting network to send malicious packets to explode processing, memory and bandwidth capacity of victims.
- **Lack of centralized control on Internet** — The Internet is an aggregation of numerous networks, connected with each other to provide global access to end users. Each network is run according to local policies defined by its owners. There is no central authority or management hierarchy, which has overall control over all networks on the Internet. Consequently, the most obvious disadvantage to DDoS defenders is that no security policy can expect its global deployment due to privacy and other commercial concerns. Moreover, different modules of distributed security systems cannot cross their administrative boundaries on the Internet without explicit cooperation.

2.1.2 DDoS Defense

The main aim of a DDoS defense system is to relieve victim's resources from high volume of counterfeit packets sent by attackers from distributed locations, so that these resources could be used to serve legitimate users. Although a lot research in DDoS has been seen but still no comprehensive solution to tackle DDoS attacks exist [89, 126, 154]. The design and implementation of a comprehensive solution which can defend Internet from variety of DDoS attacks is hindered by following challenges:

- Large number of unwitting participants [50, 89]
- No common characteristics of DDoS streams [91]
- Use of legitimate traffic models by attackers [154]
- No administrative domain cooperation
- Automated DDoS attack tools [123, 42, 43, 77, 41, 44]
- Hidden identity of participants because of source address spoofing [19]
- Persistent security holes on the Internet [100]
- Lack of attack information [89]
- Lack of standardized evaluation and testing approaches [60, 92]

In order to build a comprehensive DDoS defense solution in light of these challenges, Robinson et al. [108] recommended following DDoS defense principles:

- As DDoS is a distributed attack and because of high volume and rate of attack packets distributed instead of centralized defense is the first principle of DDoS defense.
- High normal packet survival ratio (NPSR) (ratio of number of normal packets received to total number of packets reaching at the server) i.e. less collateral damage is the prime requirement for a DDoS defense.
- A DDoS defense method should provide secure communication for control messages in terms of confidentiality, authentication of sources, integrity and freshness of exchanged messages between defense nodes.
- A partially and incrementally deployable defense model is successful as there is no centralized control for autonomous systems (AS) in Internet.
- A defense system must take into account future compatibility issues such as interfacing with other systems and negotiating different defense policies.

Based on these defense principles and variety of DDoS attacks [89], existing DDoS defense approaches are reviewed in the next section.

2.2 Review of Existing DDoS Defense Approaches

Before discussing DDoS defense approaches, it is necessary to highlight here that traditional security technologies such as router access lists [144], firewalls [36, 130], and intrusion detection systems (IDS) [55, 174], which are important components of an overall security strategy, do not by themselves provide comprehensive DDoS protection because of following reasons:

- Routers access control list cannot filter DDoS based on protocols as DDoS traffic use valid protocols. Moreover random spoofing and allowing traffic for at least listening 80 port of web server hinder statistical decisions on IP flows and cannot restrict port 80 data.
- Firewalls on the other hand are placed so deep in the network that ISP resources from perimeter to defense get wasted even if Firewall could filter DDoS attacks. Also as WWW, DNS, FTP should allow request to come so at high rate DDoS attacks exhaust the processing capabilities of firewall.
- IDS only detect, so they need to be complemented with mitigation. Moreover novel attacks are not detected with accuracy.

Since, these security components are unable to restrict DDoS attacks, so specific DDoS defense approaches are being developed to combat DDoS attacks. A category wise review of DDoS defense approaches is explained below.

2.2.1 Prevention

Prevention is a mechanism which stops the attacks before they are actually launched. There are three precautions against DDoS attacks.

- First, the ISPs are strongly recommended to install ingress filters to stop IP address spoofing [87, 98, 119].
- Second, the end host should repair their security holes as soon as possible, especially for some well-known software and protocol bugs [171].
- Third, the end hosts are encouraged to install the Intrusion Detection System (IDS) to prevent from being compromised by the adversary [49, 167].

Generally speaking, if all the schemes mentioned above can be implemented effectively, the Internet could be much relieved from DDoS attacks. The work done so far in this direction with strengths and weaknesses is given below.

Ubiquitous Ingress/Egress packet Filtering (UIPF) proposed by Ferguson et al. [119] is a well known strategy to prevent IP spoofing based DDoS attacks. Forged IP addresses are assumed as attack signature in this case. Ingress filtering is mainly concerned with filtering malicious traffic coming into local network or ISP, and egress filtering means filtering malicious traffic leaving local network or an ISP. The traffic entering your network or ISP must not have source address of your domain. Ingress filtering is based on this logic. In case of egress filtering, all those source addresses, which do not belong to your domain, but are going out in the Internet from your domain, deserve to be filtered at exit point of your domain only.

With the wide deployment of ingress/egress filtering, we can greatly limit the Attacker's ability to spoof addresses and hence reduce the risk of having denial of Service and distributed denial of service attacks. However, without universal deployment, ingress/egress filtering cannot completely stop IP source address spoofing. Even if this scheme is implemented globally, the DDoS attacks launched from compromised systems, which do employ subnet spoofing or don't employ spoofing at all can still be successfully launched.

Even Distributed reflected denial-of-service (DRDoS) attacks, which use legitimate addresses, are not prevented. Moreover, checking of every incoming and outgoing packet also cause extra overheads and increased delays. Installing ingress/egress filtering at source networks and ISPs, without incentives and legal enforcement, is very difficult to achieve at global level.

Park et al. [98] suggested Route based Packet filtering (RPF) to filter IP spoofing based attacks. RPF is an extension of UIPF [119] to the internet core. This approach employs a number of distributed packet filters at border routers between autonomous systems (ASs) to examine whether each received packet comes from a correct link according to the inscribed source and destination addresses, and the border gateway protocol (BGP) [177] routing information. A packet is considered an attack packet (or illegitimate packet) if received from an unexpected link and is dropped. As per simulation based experiment it was found that by implementing the packet filters in about eighteen percent of ASs in the Internet, almost all IP spoofing can be detected.

As compared to Ingress/Egress filtering [119] lesser global co-operation is required. But implementation in even eighteen percent of border routers is difficult, so attack still can be launched from unprotected ISPs. The effectiveness of the approach is also quite sensitive to underlying Internet-AS connectivity structure. A new spoof based problem called En–route spoofing creep in as deployment is not hundred percent. Rest of problems like subnet spoofing, legitimate address use by zombies and DRDoS attacks remain completely unsolved. Moreover, approach in [98] requires BGP messages to carry source addresses. The inclusion of source information significantly increases the BGP message size and the message processing time. Consequently, in the event of recent route change, some legitimate packets also get dropped.

RPF [98] is vulnerable to asymmetrical and dynamic Internet routing as it does not provide a scheme to update the routing information. To overcome this disadvantage, Li et al. [87] have proposed a new protocol called the Source address validity enforcement (SAVE) Protocol, which enables routers to update the information of expected source IP addresses on each link and block any IP packet with an unexpected source IP address. Similar to the existing routing protocols SAVE constantly propagates messages containing valid source address information from the source location to all destinations. Hence, each router along the way is able to build an incoming table that associates each link of the router with a set of valid source address blocks. En –route spoofing problem is also solved if global deployment is enforced.

However, SAVE [87] requires change in well established routing protocols. Moreover, as SAVE propagates messages and filters the spoofed packets to protect other entities, it does not provide direct implementation incentives. If SAVE is not universally deployed, attackers can always spoof the IP addresses within networks that do not implement SAVE. Moreover, even if SAVE were universally deployed, attackers could still launch DDoS attacks using non-spoofed source addresses. So, subnet spoofing based attacks, DRDoS attacks, and attacks not employing spoofing were still hurting problems.

Prevention techniques, which aims to solve IP spoofing [87, 98, 119], suffer from a fundamental weakness of the Internet i.e. all require appreciable global deployment. Even though Internet awareness is increasing but its rate of growth is such that unsecured hosts are also increasing at alarming rates, so global deployment of any scheme seems impossible. Moreover, lack of deployment incentive and extra overheads in filtering traffic after checking IP/BGP headers are big hurdles on the way of stopping IP spoofing. IPv6, a well known network layer protocol is still not deployed globally, because of overheads and

flexibility of already being used IPv4. Moreover, Non-spoofing, subnet spoofing, and DRDoS based attacks are not countered at all by these approaches.

Researchers now have belief that it is impossible to achieve global security because of Internet openness and fast growth. Therefore, DDoS research direction goes into securing individual infrastructure. Peng et al. [156] approach relies on securing edge network to combat DDoS attacks. The basic idea used in [156] is "history repeats itself". In DDoS context, legitimate clients access same website regularly. As per this approach all the IP addresses of the previous successful connections are recorded in order to compile an IP address database (IAD). The stale IP addresses are left out as per sliding window based on timestamps and pre-fixed window time. When protected network or website experiences a high level of congestion or load, the edge routers admit the incoming packets according to pre-built IAD. Hash or Bloom filter based techniques are used to fastly search IP in IAD. Malicious traffic from zombies using non-spoofed IP addresses is filtered using valid addresses in IAD. Moreover, TCP based spoofed packets are filtered at edge by keeping state information of all connections whose final ACK is yet to return. This scheme is robust, and does not need the cooperation of the whole Internet community.

If attacker get estimate of window time, then IAD can become database of attackers. A new legitimate client in the event of attack is not entertained. To improve percentage of legitimate traffic (NPSR) getting through, the size of IAD is increased, but lookup time will also increase in the process. Even in normal operation, all packets need to be checked for their source legitimacy, so consumption of resources to store connection history cause memory overheads. Moreover, when large amounts of expected or unexpected traffic from legitimate clients suddenly arrive at a system (e.g., flash crowd [75]), the performance of [156] falls sharply. Knowledge of web access and usage patterns can be used to distinguish

flash crowds from attack traffic [117, 166]. Defense is also vulnerable to high rate attack traffic.

Geng et al. [171] have proposed changing IP address, a simple solution to a DDoS attack in order to invalidate the victim computer's IP address by changing it with a new one to switch the concentration/congestion point. This is called moving target defense. Once the IP address change is completed, edge routers start dropping the attack packets based on destination address. Although this action leaves the computer vulnerable because the attacker can launch the attack at the new IP address, but this option is practical for local DDoS attacks, which are based on IP addresses.

On the other hand, attackers can easily render this technique a futile process by adding a domain name service tracing function to the DDoS attack tools [41, 43, 44, 77].

Geng et al. [171] also suggested some common preventive measures for individual servers and ISPs. Some of preventive measures are: if a network service is not required, the particular services port(s) should be disabled to prevent attacks. The host computers should update themselves with the latest security patches for the bugs present and use the latest security techniques available to minimize the effect of DDoS attack. Installation of regular security patches and updates help to fix security holes in host machines. Moreover, by disabling IP broadcasts, host computers can no longer be used as amplifiers in Smurf Attack [35] and other name server based attacks [34]. However, a defense against this attack will be successful only if all the neighboring networks disable IP broadcasts. Host based Intrusion detection system (HIDS) i.e. Snort [109] [167] and some network based techniques (NIDS) [20, 49, 167] aim to find intrusions attempted by attackers using signature based and anomaly based techniques respectively.

Prevention approaches [171] involving installation of patches and IDS to stop intrusion offer increased security but can never completely remove the threat of DDoS attacks because they are always vulnerable to new attacks for which patches and signatures do not exist in the database.

Summary: Prevention approaches to stop IP spoofing [87, 98, 119] repairing security holes by patches [171], and stopping intrusion [109, 167] have lot of hurdles in terms of global deployment, host based incentives, installation of patches as soon as they are developed and released, overheads to check extra packet headers, and inability to detect new attacks. Moreover, non-spoofing, subnet spoofing, En-route and DRDoS based attacks have no reliable solution in prevention techniques. According to Internet architecture working group [106], the percentage of spoofed attacks is declining. Only four out of 1127 customers had impact of DDoS attacks on a large network due to use of spoofed addresses. Moreover, on an average security awareness is still not enough [100], so expecting installation of security technologies and patches in large base of Internet looks an ambitious goal in near future. Therefore, relying only on attack prevention schemes is not enough to stop DDoS attacks.

2.2.2 Detection and Characterization

The process of identifying that a network or server is under attack after launch of the attack is called detection. Detection can be passive if logs are analyzed after attacker fulfills his/her desire and attack is over. For analysis mining of collected packet traces can be helpful [145, 163]. Detection can be on time if we detect when attack is ongoing. It can be even proactive if either attack is detected before it reaches target or before appreciable degradation of service, we can detect signs of attack. Characterization means differentiating attack packets from legitimate packets by looking at some feature/header of packets which are derived from monitoring and analysis at various times and points of the Internet. The

special feature of current DDoS attack packets is that individually each packet is a perfect legitimate packet but in combination, correlating these packets monitored at different points can give some signs of uniqueness from legitimate packets. The study in this area is totally disarrayed i.e. different detection and characterization methods are proposed using different topologies and different attacks [60]. No benchmarks and evaluations criteria exist which can compare different approaches [60, 92].

Basically there are three methods to detect DDoS attacks [101]:

1. Signature based which are normally used in prevention. Signature based detection Bro [165], Snort [109] is useful to detect an important class of attacks (e.g., known worms and viruses) but is not helpful in detecting other attacks (e.g., scans, DDoS attacks) which are not characterized by a signature within a single packet, but exhibit unusual behavior across a set of packets . The communication between attackers and their zombies can be detected using signature based techniques for known DDoS attack tools as demonstrated by Cheng [61]. Data mining can be applied for scan detection [63]. However, state-of-art attack tools easily escape signature based detection as encrypted communication is used for control instructions between attacker and zombies.
2. Anomaly based which are used in on line attack detection uses normal detection models. Building a normal profile is the first step for all anomaly based detection techniques. Since there is no clear definition of what is normal, statistical modeling plays a crucial role in constructing the normal profile. Statistics-based anomaly detection includes two major parts. This first part is to find effective parameters to generate similarity measures. The parameters can be IP packet length, IP packet rate, management information base (MIB) variables, resource usage variables, and packet

header fields etc. Building a normal profile for detecting all types of attacks [22, 89] is the biggest hurdle before researchers because of variety of Internet services, protocols, and expected traffic load conditions. As a result, legitimate traffic activity is sometimes flagged as attack, called false positive or attack is sometimes not identified, called false negative. The second part is to calculate the similarity distance between normal and current profile. Statistical methods, such as χ^2 [102] and Kolmogrov-Smirnov tests [27] have been used to compare current profile with normal profile. If the distance between the current monitored traffic and the normal traffic profile is larger than a prefixed threshold, a DDoS attack is detected. Carl et al. survey [60] on DDoS detection schemes reveal that in most cases, researchers offer no guidance on setting thresholds. Infact, researchers often choose thresholds to suite their own experimental test cases and do not illustrate effect of threshold variations on performance.

3. Congestion based which are now days used in tolerance. These methods identify attack and malicious traffic effectively only when aggregate traffic induces congestion on the monitored links. Current DDoS attackers use a large number of zombies to launch attack. So characterizing attack sources on the basis of congestion is very difficult. Moreover, low rate degrading flooding DDoS attacks [89] cannot be detected by congestion based schemes as these attacks consume victim's resources gracefully and do not cause congestion to go so high that it can be detected. On a busy server where load of legitimate transactions are high, congestion based schemes cannot be used to detect DDoS attacks.

Keeping in view nature of DDoS attacks and available methods to detect, anomaly based detection seems promising. A few key strategies in anomaly based detection are surveyed with strengths and weaknesses as below.

Gil et al. [153] propose a heuristic data-structure called MULTOPS to detect denial of service attacks by monitoring the packet rate in both the up and down links. MULTOPS assumes that packet rates between two hosts are proportional during normal operation. A significant, disproportional difference between the packet rate going to and from a host or subnet is strong indication of a DoS attack.

MULTOPS assumes that the incoming packet rate is proportional to outgoing packet rate, which is not always the case. For example, real audio/video streams are highly disproportional, and with the widespread use of on-line movie and on-line news, where the packet rate from the server is much higher than from the client, false positive rates will become a serious concern for this scheme. Moreover, MULTOPS is vulnerable to attacks with randomly spoofed IP source addresses. The simplest way to cripple MULTOPS is to use randomly spoofed IP addresses, which makes the calculation based on genuine IP addresses inaccurate and consumes resources by storing spoofed IP address information. Another countermeasure is to connect to the target from a large number of attack sources in a legitimate manner (e.g. downloading a file from a ftp server). Therefore, the packet rate ratio between in flows and out flows during the attack will appear to be normal and undetected by MULTOPS. Some more disadvantages of MULTOPS are requirement of router reconfiguration and new memory management schemes.

Wang et al. [70] proposed SYN detection to detect SYN floods, and Blazek et al. [124] proposed batch detection to detect DoS attacks. Both methods detect DoS attacks by monitoring statistical changes. The first step for these methods is to choose a parameter for

incoming traffic and model it to be a random sequence during normal operation. In [70], the ratio of SYN packets to FIN and RST packets is used, while in [124] a variety of parameters, such as TCP and UDP traffic volume, are used. The attack detection is based on the following assumptions. First, the random sequence is 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. 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 traffic is the same as it is in normal traffic. Therefore, separating different types of traffic cannot make the attack behavior more conspicuous or obvious.

The attack flows do not follow flow and congestion control signals in [107, 140], so are not regulated as normal TCP traffic. So, attack flows have different statistical features compared with normal flows. Based on this assumption, Cheng et al. [26] propose to use spectral analysis to identify DoS attack flows. In this approach, the number of packet arrivals in a fixed interval is used as the signal. In the power spectral density of the signal, a normal TCP flow will exhibit strong periodicity around its round-trip time in both flow directions, whereas an attack flow usually does not.

First of all, spectral analysis is only valid for TCP flows. As UDP and ICMP are connectionless protocols, the periodic traffic behavior is unexpected. Attackers can use UDP or ICMP traffic to confuse the detection scheme. Moreover, the attacker can mimic the periodicity of normal TCP flows by sending packets periodically. For example, a large number of zombies can be directed to make legitimate TCP connections to the target.

Though assumption strength is considered strong in this case, but computational complexity is high. High computational complexity of a detection approach is vulnerability in itself under high rate flooding DDoS attacks.

Kulkarni et al. [10] observed that a large numbers of similar packets (in terms of their destination address, protocol type, execution pattern etc.) are used for launching most of DDoS attacks. Thus, there is a lot of similarity in the traffic pattern. Basically they assumed that similar DDoS attack codes are used for launching DDoS attacks from different zombies. On the other hand, legitimate traffic flows tend to have many different traffic types. Hence, traffic flows are not highly correlated and appear to be random.

The assumption to use same attack tool suggests that the resulted traffic is highly correlated. But, there are many types of attack tools which do not observe this feature. Attacker sources can be orchestrated to break the correlation by sending attack traffic at different times, with different traffic types, packet sizes, and sending rates. This is easy to achieve. For example, attackers can use the IP address of a compromised computer as the random seed to generate a set of parameters for configuring attack traffic. By doing this, attack traffic will appear random, which can bypass detection.

Another detection method of DDoS attacks uses the Management Information Base (MIB) data from routers using time series analysis [54]. The MIB data from a router includes parameters that indicate different packet and routing statistics. Cabrera et al. [76] has focused on identifying statistical patterns in different parameters, in order to achieve the early detection of DDoS attacks. It looks promising for possibly mapping ICMP, UDP and TCP packet statistical abnormalities to specific DDoS attacks. There are three steps to this scheme. The first step is to extract the key variables from the target e.g., the number of ICMP echo packets is the key variable for Ping Flood attacks. The second step is to use

statistical tools (e.g., AutoRegressive Model) to find the variables from the potential attackers that are highly related to the key variable. For example, the number of ICMP echo reply packets at the potential attackers is highly correlated with the key variable for Ping Flood attacks. The third step is to build a normal profile using the found variables from the potential attackers. Any anomalies from potential attackers compared with the normal profile are regarded as strong indications of an attack. Step one and two are completed during the off-line training period and step three is done on-line.

The vulnerability of this scheme is that the effectiveness of training is based on the features of known attacks. The attacker can disturb or disable the detection scheme by inventing new attacks. As DDoS attacks do not necessarily need to use any particular type of traffic, it is easy for the attacker to create a new type of attack just by combining different types of attack traffic. This causes multiple key variables from the target, and the correlations between the variables from the potential attackers and the target will become extremely complex, which complicates the process of building a normal profile and makes the detection less effective.

Mirkovic et al. [93] proposed a system called DWARD that does DDoS attack detection at the source based on the idea that DDoS attacks should be stopped as close to the sources as possible. D-WARD is installed at the edge routers of a network and monitors the traffic being sent to and from the hosts in its interior. If an asymmetry in the packet rates generated by an internal host is noticed, D-WARD rate limits the packet rate.

The drawback of this approach is that there is a possibility of numerous false positives while detecting DDoS conditions near the source, because of the asymmetry in the packet rates for a short duration. False negatives can also occur because of distributed nature of

DDoS traffic and use of large number of zombies to launch attacks. Furthermore, some legitimate flows like real time UDP flows do exhibit asymmetry.

To conclude so far, we can say all these defense schemes are attack specific. Most assumptions are not strong, since attackers can change their attack patterns to exploit the assumptions and evade detection. Although the assumption for spectral Analysis proposed by Cheng et al. [26] is strong, it only works for TCP flows and it is complicated to implement. D-WARD [93] employed at source is revolutionary in the sense that it detects and filters attacks at source but DDoS attacks characteristics and lack of implementation incentives, decreases motivation for global deployment.

Few more detection schemes which rely on source address distribution and Cumulative sum (CUMSUM), are discussed below.

Feinstein et al. [102] focus their detection efforts on activity level and source address distribution using sample entropy. They cluster flows according to the addresses of the destination machines located behind the monitoring point. The first cluster contains the single most frequently seen source address, the second cluster contains the next four most frequent, the third cluster the next 16, the fourth the next 256, and the fifth the next 4,096; the sixth cluster encompasses all remaining traffic. The researchers compare each cluster's activity level to the expected amount using a chi-square statistic, thus providing a "goodness of fit" result. A deviation from the expected traffic profile suggests anomalous activity.

Change-point detection algorithms operate on continuously sampled data. An example here is cumulative sum (CUMSUM) algorithms. To identify and localize a DoS attack, the CUMSUM identifies deviations in the actual versus expected local average in the traffic time series [70, 124, 133]. If the difference exceeds some upper bound, the CUMSUM's recursive statistic increases for each time-series sample. During time intervals containing

only normal traffic, the difference is below this bound, and the Cusum statistic decreases until reaching zero. Using an appropriate threshold against the Cusum statistic, the algorithm identifies an increasing trend in the time-series data, which might indicate a DoS attack's onset. Through the settings of the threshold and upper bound, the Cusum algorithm can trade off detection delay and false-alarm rates. Wavelet analysis [116] describes an input signal in terms of spectral components. Although Fourier analysis is more common, it provides a global frequency description and no time localization. Wavelets provide for concurrent time and frequency description, and can thus determine the time at which certain frequency components are present. Wavelet energies in the high-band spectral window identified change points within an input signal. Brooks et al. [133] enhanced CUMSUM change-point detection approach by using discrete wavelet analysis. Wang et al. [69] uses CUMSUM to detect SYN flooding attacks. Based on the inherent protocol behaviour of the TCP connection establishment and tear down, they utilize two types of packet pairs: SYN versus FIN and SYN versus SYN/ACK. The SYN versus SYN/ACK pair method is employed at first-mile (egress) router, whereas SYN versus FIN pair is installed at last–mile (ingress) routers. First pair method detects flooding attacks and flooding sources, whereas second pair method detects flooding attack near the victim.

CUMSUM based detection approach [69, 124] has very less computational complexity. Scheme in [102] uses very simple metric called sample entropy to summarize traffic distribution. Moreover it uses only six bins to analyze its address distributions. The complexity in this case is higher than CUMSUM, but overall it is lesser than lot of approaches which uses complex metrics to detect attack. The combined CUMSUM and wavelet-based approach incurs an extra $O(2^n)$ complexity over CUMSUM, where $5 < n < 11$ is the spectral resolution level. The approach in [116] is at least two times as

complex as [133] because it uses two redundant wavelet filter stages. A popular backscatter technique [50] analyzes the largest address distribution, so their method consequently requires most computation and memory use. However, in a single ISP domain if traffic meant for only protected servers is analyzed then computational complexity is quite lesser than before. Computational complexity can be further decreased by doing sampling of traffic at ingress edges of an ISP, provided sampling has minimum effect on chosen computational metric to detect the attack.

Summary: Important problems in existing detection and characterization schemes are listed below:

- Availability of user friendly attack tools [41, 43, 44, 77] and their source codes give flexibility to attackers to create a variety of new attacks by error and trial. Most of detections schemes can easily be defeated by developing attacks through this error and trial method. Even existing variety of attacks are sufficient to disguise most of prevailing detection methods [60, 89]. For change-point detectors [124] that monitor changes in packet volume over time, an initially low, slowly ramping attack rate dynamic might be obscured by the background traffic's high variability.
- In all detection schemes, researchers have yet to develop nominal-traffic measures that encompass the range of possible network conditions. Network services have different activity levels and availability, in keeping with users' variable time-of-day interactions. At this point, it is unclear whether suitable training algorithms or guidelines, exist that can adequately model normal traffic's irregular behaviour.
- Most of detection systems are under tested against varying network and attack conditions. Comprehensive testing is a highly complex, and time-consuming process. Existing studies employed little variation in network topology, number of

legitimate clients and attackers, attack strengths to emulate a realistic deployment setting. This under testing problem is mainly due to lack of comprehensive test data, testing environments, and standards. Availability of attack data, keeping in view reputation of affected organization is also an issue, which stands in the way of DDoS researchers. Most recently Mirkovic et al. [90, 92] are in the process of recommending benchmarks and evaluation criteria, which may give proper shape to DDoS research in future.

- Detection models normally have tunable parameters like clustering level (traffic aggregation for monitoring), sampling window size, and thresholds etc. In most cases, researchers offer no guidance on parameter variations or their effects on detection performance [60]. Ad hoc training is typically required to tune parameters as per desired detection performance. But actually researchers often optimize parameters to their own experimental test cases so as to show better results.
- In order to provide all types of normal traffic behavior, a large number of parameters are used to provide accurate normal profiles. Similarly to trap attackers' tricks to mimic legitimate traffic patterns, so that false positives and negatives should be minimized, sophisticated detection and characterization algorithms are required. However, with the increase of the number of parameters and sophistication in algorithms, the computational overhead to detect attack increases. This becomes a bottleneck, especially for volume-oriented DDoS attacks that are aggravated by the Computational overhead of the detection scheme. More importantly, unlike sophisticated network intrusions that depend on malformed packets or special packet sequences, DDoS attacks only need the massive traffic volume generated by a number of compromised hosts, A prominent example is that during the spread of the

"Code Red" worm [33], over 300,000 "zombie" machines were compromised to launch a denial of service attack on the White House Web site [45]. Due to large number of zombies traffic volume generated by a DDoS attack can exceed 10Gb/s [170].

2.2.3 Traceback

DDoS is a well coordinated attack where attacker employ a network of weekly secured machines (Handlers/masters/zombies or bots) called botnet. Once an attack has been detected, an ideal response is to block the attack traffic at its source and identify complete botnet. Because of limited cooperation between ISP's, bugs in operating systems, and attacker's growing technical ability to exploit these bugs, it is very difficult if not impossible to track this botnet. In best of the work done so far, reaching up to zombies and hence limiting the attack army and then thorough investigation of these zombies to find traces of communication with other parts of botnet is done. Unfortunately, there is no easy way to track even IP traffic to Zombies and characterize the path used by packets to reach from zombies to victim. This is due to two aspects of the IP protocol. The first is the ease with which IP source addresses can be forged. The second is the stateless nature of IP routing, where routers normally know only the next hop for forwarding a packet, rather than the complete end-to-end route taken by each packet. This design decision has given the Internet enormous efficiency and scalability, albeit at the cost of traceability and network security in terms of DoS/DDoS attack. In order to address this limitation, many schemes based on enhanced router functions or modification of the current protocols has been proposed to support IP traceability. Some of these schemes include probabilistic packet marking (PPM)[147] and authenticated PPM [53], ICMP traceback (iTrace) [137] and intention-driven ICMP traceback [14], source path isolation engine (SPIE, also called hash

based traceback) [4] and sampling based traceback [86], algebraic based traceback (ATA) [40], deterministic packet marking (DPM) [1], deterministic edge router marking (DERM) [135, 136], intelligent traceback [111], linkage information based DPM [3], an overlay-based solution (center-track) [131] and secure overlay system SOS [6], and controller-agent model in single ISP domain [158] and multiple ISP domains [157].

A survey conducted by Gao et al. [182], evaluate the performance of traceback schemes on the basis of various evaluation metric recommended in [2]. These metrics are minimum number of marked packets required for path reconstruction, processing burden, bandwidth overhead, memory overhead, robustness, scalability, and ISP involvement. Moreover, an analysis of traceback techniques is also performed in [160].

Summary: Overall in all of these traceback techniques high computational overheads are involved. Moreover security of communication framework so that these control messages should not be forged in terms of Confidentiality, Authentication, Integrity, and freshness is a also big hurdle to tackle with. Co-operation between ISPs is always bump to bear with. Overall research direction in this field has been limited mostly to finding Zombies and path characterization up to Zombies. However, some passive approaches also worked for separating communication between attacker and master, and master and zombies. Integrating detection and characterization with traceback is also one of the main concerns. Currently traceback in combination with tolerance and mitigation is a popular methodology to defend DDoS attacks.

2.2.4 Tolerance and Mitigation

DDoS attack traffic (not control traffic between master and zombies) can be either semantic or unintelligent and useless. Semantic attacks are affected by specially crafted intelligent

packets. These types of attacks however can be controlled by timely installing patches for applications/OS and protocols. However, the second type where bogus but valid packets are used to exploit the inherent vulnerabilities of Internet i.e. asymmetry of intelligence and resource, and best effort packet forwarding model, are very difficult to contain. The target of this thesis is to curb these distributed flood based DDoS attacks. The main aim of tolerance and mitigation is to provide an optimum level of service as per QoS requirements to legitimate clients while the service provider is under attack. This is not a comprehensive solution in any way, however it can complement other approaches to work in parallel and achieve their goals by providing sufficient assurance and cushion in terms of time to providers that the legitimate clients are being served. Moreover this approach can itself get complemented from others approaches and then synergistic effect is best performance for clients.

Tolerance and mitigation based techniques mainly deal with link or network based and node or server based flooding DDoS attacks. Under tolerance and mitigation the objectives are: First, finding agent addresses from where attack traffic is originating and secondly, applying throttles as per requirements of link and server sustainability. It has been observed that in network based attacks, it is finally congestion at access links which cut the connectivity between legitimate clients and requested services, however in server based attacks normally there is not appreciable congestion at the links but it is sheer volume and rate at which requests for service keep coming to servers, which the server is not able to handle at the requested rate. Thus, health monitor at server [175] shows very high loads beyond thresholds and finally buffer overflows at server and hence denial of service. So DDoS is basically a resource overloading problem where resource can be access link, server data structures / handles, and memory or CPU cycles. First tolerance and mitigation

techniques for network based flooding DDoS attacks are reviewed and it will be followed by defense for server based attacks.

2.2.4.1 Network based Attacks

Tolerance and mitigation techniques to stop network based flooding attacks mainly concentrate on controlling intentional and malicious congestion. Flooding DDoS attack traffic does not observe end-to-end congestion control [107, 140], so the router control plays a predominant role in defending DDoS attack. The two types of router algorithms for network based congestion control are scheduling and queue management. The classic example for scheduling algorithms is fair queuing algorithm (FQ) [7]. FQ requires the router to partition the input traffic into separate queues and use a separate buffer space for each queue. Router keeps the state for each flow and manages them individually. One flow cannot degrade the quality of another.

However FQ needs, complex per-flow state, which makes it too expensive to be widely implemented. To reduce the cost of keeping per-flow state in every router, Stoica et al. [72] proposed a new scheduling algorithm called core stateless FQ to categorize the routers into edge routers and core routers. An edge router maintains the per-flow state information and estimates the arriving rate for per-flow. These estimates are inserted into the packet headers and passed on to the core routers. The core router keeps a simple stateless FIFO queue and drops packet according to the estimates in the packet header during the congestion. Although the scheme simplifies FQ, it is still very expensive to keep per-flow state information. However because of lesser metering in core routers, it is better than previous one. Still extracting packet information from the packet in core, adds to the complexity of this scheme. Mckenny [121] proposes stochastic fair queuing (SFQ) to approximate FQ at a smaller implementation cost. SFQ classifies packets into a smaller number of queues than

FQ using a hash function. Although it reduces the complexity of FQ, it still needs 1,000 to 2,000 queues in a typical router to approximate FQ performance.

Scheduling algorithms [7, 72, 121] can ensure the fairness between the traffic flows, but they are too expensive in terms of delays, state monitoring and don't scale well to a large number of users. Moreover large number of slow rate DDoS traffic flows can still prove lethal to victims.

In contrast, queue management algorithms are usually simple but it is weak in proving fairness. Random Early Detection (RED) [141] exemplifies this class of algorithms. A router only maintains a simple FIFO queue for all traffic flow and drops the arriving packet randomly during congestion. The probability of a packet drop increases with growth in queue. By keeping the output queue size small, RED can reduce the delay time for most of the traffic flow.

RED [141] cannot penalize the misbehaving traffic flows. In order to improve the RED's probability to penalize the misbehaving traffic flows, Floyd et al. [142] have proposed a technique to use a lightweight detection algorithm to identify unresponsive flows and then explicitly manage the bandwidth of these flows. Their technique performs tests that identifies flows that are unresponsive, TCP-friendly, or high-bandwidth and regulate them. Lin et al. [47] propose a technique called Flow Random Early Drop (FRED) to maintain the fairness between the traffic flows. It only keeps the states for the flows that have packets buffered in the router. Once queue length of one flow in the buffer is between minimum (min) and maximum (max), the packet for this flow is dropped randomly. Once the queue length is larger than max, the incoming packet is dropped. The router also keeps the information to count the number of times the flow has failed to respond to congestion notification. Penalties are taken to the unresponsive flows.

These variants of RED incur extra implementation overhead since they collect certain type of state information. Ott et al. [152] set up another interesting variant called stabilized RED (SRED). SRED stabilizes the FIFO buffer occupancy independently of the number of active flows. It maintains a data structure called Zombie list, which serves as a proxy for information about the recent flows. By doing this, it can estimate the number of active flows and identify the candidates for the misbehaving flows. Although SRED can identify misbehaving flows, it hasn't set up a scheme to penalize them. To improve this scheme, a stateless active queue management scheme called CHOKe (CHOose and Keep for responsive flows, CHOose and kill for unresponsive flows) was proposed to approximate fair bandwidth allocation. CHOKe draws a packet from the FIFO buffer at random and compares it with the arriving packet If they both belong to the same flow, they are both dropped; else the randomly chosen packet is left intact and the arriving packet is dropped with a probability p which depends on the congestion level.

Although this scheme is effective in defending the unresponsive traffic flow, it performs poorly for a large number of small traffic flows. So it is still vulnerable in defending flash crowds [75] and DDoS attacks.

Mahajan et al. [128] propose a new variant of RED called RED with Preferential Dropping (RED-PD) to identify high bandwidth flows and control the bandwidth obtained by these flows. However, it controls the high-bandwidth flows by estimating their arriving rate, which is not very accurate. Furthermore, the test for unresponsive traffic flows need to be more accurate to maintain fairness.

Lau et al. [59] have investigated the performance of various queuing algorithms implemented in a network router under flooding DDoS attack, and tried to find whether legitimate users can obtain desired service or not. The simulations show RED and Class

Based Queuing (CBQ) are successful in providing a part of bandwidth requested by the legitimate user during high rate flooding DDoS attack. The topology, traffic generation model and applications used in the simulation are very simple compared to the realistic network topology.

To guarantee fairness, it was suggested that DDoS attacks could be countered by applying resource allocation techniques on network bandwidth. Integrated Services (IntServ) [169] and Differentiated Services (DiffServ) [138] are two approaches aimed at isolating flows with specific QoS requirements from lower-priority traffic. IntServ uses the resource reservation protocol (RSVP) to coordinate the allocation of resources along the path that a specific traffic flow will pass. The link bandwidth and buffer space are assured for that specific traffic flow. Taxonomy of approaches to per-class QoS differentiation is presented in [183].

DiffServ [138] has provided a novel architecture for implementing scalable service differentiation in the Internet. This architecture maintains the scalability by aggregating traffic classification state which is conveyed by the means of IP-layer packet marking using DS field [97]. Packets are classified and marked to receive a particular per-hop forwarding behavior on nodes along their path. The internet is classified into boundary nodes and core networks. All the sophisticated operations, such as marking, policing, shaping, are deployed in the boundary nodes. Per-application flow or per-customer forwarding state need not be maintained within the core of the network. Network resources are allocated to traffic streams by the service provisioning policies, which govern how traffic is marked and conditioned upon entry to a differentiated service-capable network. DiffServ can classify traffic according to the policies. It protects the high-priority traffic as compared to lower

priority or best-effort traffic. Thus, it can be an effective approach to defend flash crowds or DDoS attacks.

Global co-operation is must in routers (edge/core) of multiple domains so as to implement these preference based approaches. Moreover whenever we discriminate, we need to police and authenticate sources to prevent cheating, which may turn out to be complex and costly. Flow or aggregate based monitoring and metering, definitely have computational overheads. Web traffic, which is a significant fraction of network traffic, is likely to remain best-effort as per TCP congestion control spirit and freedom to access, so it is not protected by QoS requirements.

Wang et al. [68], propose that all types of packets don't deserve same treatment as done in best-effort model and even in packets of same class under DiffServ .e.g. an ACK packet containing acknowledgement of several packets is more important than a simple TCP data packet So packets can be differentiated into various bandwidth aggregates (BA) (TCP, UDP, ICMP and TCP control etc) deserving different allotment of bandwidth at edge and core routers. In this work, a transport-aware IP router (tIP) architecture is provided in which layer-4 service differentiation and resource isolation is done. The key components of the tIP router architecture are the fine-grained QoS classifier and the adaptive weight based resource manager. A two-stage packet classification mechanism is devised to decouple the fine-grained QoS lookup from the routing lookup at core routers. Moreover one lightweight and heavyweight classification algorithm is implemented at core and edge routers respectively to classify the aggregates. Then by using separate queues and adaptive-weighted bandwidth allocation, better service differentiation and resource isolation are achieved for these aggregates. This architecture is also compatible with DiffServ [138]

model. The main drawback of this approach is extra overheads in dealing with transport layer headers as routers are three-layered devices.

Xiong et al. [178] also took the defense of DDoS attack as a congestion control problem. They propose to use backward pressure propagation, feedback control scheme to defend DDoS attack. They used rate-based and queue-length-based algorithms to create the feedback signal accordingly. Once the input traffic rate or the output queue length has exceed the desired threshold, a feedback signal is sent to adjust the admitted portion of traffic in different input and output ports to put the rate and queue length below the threshold. The method is effective to make sure that the network traffic works in a tolerable level during DDoS attack. However, they don't set up a scheme to discriminate good traffic from bad traffic.

All the schemes up to this stage can punish a small number of large unresponsive traffic flows to maintain a certain level of fairness at the expense of increased delay, state monitoring, or marking. However, no scheme works well during the time of DDoS attack which is characterized by a large number of small traffic flows.

After this it was thought to characterize groups or aggregates of agent addresses responsible for DDoS attack and punish them by applying rate limiting or throttling. Technically speaking a aggregate can be taken as a collection of packets from one or more flows with a common property where a flow represents stream of packets being exchanged between a client application (IP address + port address) and a server application (IP address + port address). For example, all TCP SYN packets going to 128.250.*.* make up one aggregate and client with IP address 202.20.2.9:1120 interacting with server 128.250.10.7:80 using HTTP protocol represents a flow.

DDoS traffic mostly belongs to high-bandwidth traffic aggregate. It blocks the services for other users by sending huge traffic to a particular link and causes a server network congestion, thus prevent normal users from reaching the server. So the direction went into finding high bandwidth ,TCP unfriendly and unresponsive aggregates consisting of traffic from small or high rate flows , so that state monitoring overhead should decrease and DDoS flows contributing to these aggregates should be accurately detected as for as possible. An optimum point and rate to throttle these aggregates need to be decided afterwards.

Mahajan et al. [129], propose aggregate based congestion control (ACC) which gives some relief to congested links due to DDoS attacks and flash crowds. ACC is offered in two ways namely local ACC and pushback based ACC.

The ACC itself is broken into two phases namely detection and control. In the detection phase, the ACC agent is responsible for identifying the aggregate and calculating the rate limit for them. The rate limit in the control phase determines whether a packet is to be discarded or forwarded. In local ACC, both phases are applied on the same congested router, whereas the other referred as a pushback scheme extends the local ACC to upstream routers. The advantage is that the aggregates can be completely stopped or rate-limited to a particular bandwidth. Once the aggregates are identified the router can ask the upstream routers to prevent it and hence reach the source of the attacker. When pushback is applied upstream, there is more bandwidth for legitimate traffic in downstream routers. In some sense, pushback is similar to trace route. Later architecture for Pushback and its implementation under FreeBSD is done by Ioannidis et al. [84].

Here definitely high bandwidth aggregates are rate limited but DDoS traffic is not always high rate even slow rate DDoS traffic from a large number of zombies or agents can also cause denial of service. In this case, if distribution of agents is isotropic then DDoS

traffic enters at different ingress edge routers, hence aggregate formation is not proper for pushback. Moreover for DRDoS attacks, the collateral damage is more as rate limiting is applied based on victim address, so a lot of legitimate traffic is also throttled. Among other disadvantages associated with this approach, requirement of strong authentication to perform a pushback is important. As without authenticated pushback, the scheme itself can be used by attackers to launch DoS attacks on the network. Even after implementing strong authentication, if the attacker can compromise a single ACC implemented router, severe damages can be caused. When a pushback enabled edge router is not able to discriminate attack traffic, and upstream routers are in different domain where ACC is not enabled then collateral damage is more [179]. Moreover global cooperation is must otherwise rate limiting is possible only till edge routers of the ISP.

As compared to previous work, state monitoring is reduced to only those flows, which contribute to high aggregates for rate limiting. At least concentrated attackers or attacker's traffic flow which contribute to high bandwidth aggregates are rate limited. But extra computational and communication overheads to implement authentication, confidentiality, integrity and freshness of pushback messages is an added expense. In case of wireless networks in which low power devices are used for communication [88], voluminous nature of flooding DDoS attacks can cripple the services severely.

2.2.4.2 Server based Attacks

Now we switch our attention towards server based DDoS attacks, where it is assumed that the servers are attached near the backbone routers so that the flood traffic cannot create congestion at access link. So in this case there is no congestion on links in most of the cases. However the traffic to process at server is more than it can handle in terms of either rate or volume. On the servers, memory in which data structures required for running server

process and network based requests is a scarce resource. Buffers or data structures like backlog are limited for network based protocols. Process control blocks, file descriptors and handles are limited in case of operating systems. So only a finite number of requests can be handled per unit time and hence number of requests should arrive only within a limit. In most of the cases, the detection on server based attack is done by server itself or the firewall/IDS shielding the server however handling of DDoS attacks are done by either reconfiguring router access lists, dynamic firewall rules , over provisioning of resources or throttling requests .

Kargl et al. [58], suggested load balancer based technique in which a cluster of web servers are protected by firewall and load balancer. Server clustering and load balancing are discussed in [32]. Firewall implements traditional prevention measures and filters suggested by load balancer time to time. Load balancer act as translator and also allocates requests to appropriate web server as per load. Moreover traffic monitors at web servers and load balancer in consultation with manager, deduce classification for packets to be treated by load balancer using class based queuing. The same CBQ is also used at web servers for sending response to various classes. Sairam et al. [16] also worked for fair bandwidth allocation using load balancing. Even at physical layer level fair bandwidth allocation similar to [148] can be investigated.

The biggest challenge however is the power of many against few which is inherent feature of DDoS attacks. Moreover, long delays are caused to legitimate packets because of CBQ and slow rate attacks also go unnoticed.

An architecture that relies on the provision of QoS mechanisms in intermediate routers is VIPnets that was proposed by Brustoloni [78]. In VIPnets legitimate traffic is assumed to

be the traffic coming from networks implementing the VIPnet service. All other traffic is considered as low-priority and can be dropped in the case of an attack.

Proactive server roaming a similar to technique highlighted by khattab et al. [146], which was further extended by Sangpachatanaruk et al. [30] using large simulated network, is an anticipation based technique to defend DDoS attacks. Here one server from cluster of servers is made active server at a particular time. The timing and actual address of server is calculated by legitimate clients with the help of preloaded client module. The client module is loaded through secure communication on authenticated clients. The incomplete connections and sessions are replicated on the roamed servers also using secure migration protocols. So by this way, only legitimate clients can access the server whereas all others are filtered either through dynamically configured router access lists or firewall. Moreover attackers' packets are logged for further analysis.

The firewall is not given proper protection from high volume of packets. During roaming and replication, even legitimate packets suffer. This methodology should be tested on real internet like topology using internet like traffic models running various types of services. Moreover, various secure communication methods and roaming strategies can also be explored in simulations to get better results.

Using the techniques employed in QoS regulation Garg et al. [8], proposed a defensive approach against DDoS attacks by regulating resource consumption, which belong to the category of resource accounting. They suggest that resource regulation can be done at the flow level, where each flow gets a fair share of the resource much in the same way as round robin scheduling in CPUs. However, it is still possible to mount a Denial of Service attack by having a large number of hosts connecting to the server each claiming their slice of the resource, thus causing resource starvation.

Juels et al. [9] propose a pricing mechanism, where the client has to solve a cryptographic problem (puzzle) with varying complexity before the server allocates resources to the requests and starts servicing it. Client puzzles allow for the graceful degradation of services when an attack occurs. A server can even increase complexity of the puzzles so as to get more time between requests.

The main disadvantage of approach in [9] is requirement of specialized software or plug-in to be installed on the clients. The public services available on the Internet have lesser motivation towards these alterations and requirements.

Resource pricing is another approach that was proposed by Mankins et al. [48] in order to mitigate DDoS attacks. They noted that DDoS attacks work because the cost falls overwhelmingly on the server, and during an attack, the attack traffic is virtually impossible to tell apart from legitimate traffic. They propose distributed gateway architecture and a payment protocol that imposes dynamically changing prices on network, server, and information sources in order to push some cost of initiating service requests in terms of monetary payments and computational burdens back onto the requesting clients. By employing different price and purchase functions, the architecture can provide service quality differentiation and furthermore, select good client behavior and discriminate against adversarial behavior. They identify allotting a priority mechanism to desirable clients based on a key, and punish clients that cause load on the server.

The drawback of this approach is that a malicious user can populate the system with fake request at a low price, thus driving up the price for legitimate users. Mankins et al. [48] recommends solving this problem by partitioning resources into classes and using different pricing functions for each class. The obvious disadvantage is social acceptability and excessive state monitoring and analysis at intermediate routers.

Xu et al. [95] also suggested means to sustain availability of web services under DDoS attacks. The first step to distinguish spoofed packets is accomplished by redirecting a client to a new IP address and port number (to receive web service) through a standard HTTP redirect message. Part of the new IP address and port number serve as message authentication code for the client's source IP address. Packets from spoofed IP sources will not have the correct MAC since the attacker will not be able to receive a redirect message. Further to curb non spoofed sources fair bandwidth allocation is done using fair queuing. Defaulters are blacklisted and filtered at firewall.

Flash crowds [75] are little difficult to handle as there will be large number of legitimate SYN requests at once but only very few will get through , so performance will be little poor. Moreover flash crowd increase state keeping overheads. Under DDoS with non spoofed addresses, flash traffic will get long delays.

Yau et al. [46] used router throttles to combat DDoS attacks against Internet servers. A proactive approach is followed in the sense that before aggressive packets can converge to overwhelm a server, routers along forwarding paths, regulate the contributing packet rates to more moderate levels, thus forestalling an impending attack. The basic mechanism is for a server under stress, to install a router throttle at an upstream router several hops away. The throttle limits the rate at which packets destined for server will be forwarded by the router. Traffic that exceeds the rate limit can either be dropped or rerouted to an alternate server. However, if the current throttle fails to bring down the load to below threshold, the throttle rate is reduced. On the other hand, if the server load falls below a low-water mark, the throttle rate is increased (i.e., relaxed). If an increase does not cause the load to significantly increase over some observation period, then the throttle is removed. The throttle rate is determined by two strategies: Just half or Farley equal (fair throttling) at all routers. The

goal of the control algorithm is to keep the server load within lower and upper thresholds whenever a throttle is in effect.

Here no pushback and response messages are required as in pushback technique [84, 129]. Moreover in case of evenly distributed attackers, this approach yields better results as throttling is carried at k hops away, so concentrated good traffic near server is not dropped. In fact the effectiveness increases with value of k. Attackers can exploit communication part as no secure ways are used to send throttle messages in same and different domain. In case of meek slow rate attack, NPSR is very low. Control parameters should be set more dynamically and intelligently. Oscillations and more convergence time can also become bottleneck. Static selection of throttling routers can also become headache. Moreover no consideration of bandwidth, queue length of available links and routers between server and throttling routers/ingress points of ISP is considered in calculation of throttling rates as it is assumed that server is attached to backbone routers, so high bandwidth is available near server. Hence there is no chance of congestion of any link close to the server.

Summary : Though FQ [7, 72], SFQ [121], QoS based techniques [97, 138] and identifying thinner Bandwidth aggregates [68] and then regulation are good solutions as also proposed by Crocker [149], but excessive state monitoring, calculating proper rate limits and testing for defaulters cause appreciable overheads considering rich resource based Internet of present age used for launching flood based attacks. So better monitoring policies (local or distributed), dynamic rate limits as per legitimate traffic models and algorithms for classifying defaulters to test only suspicious clients are main challenges upfront. Computational burden on core routers are decreased [72, 97] but still more ways can help the cause. Some algorithms are available for detecting high bandwidth aggregates based on destination address [84, 129]. If somehow we can find source characteristics to narrow

down these attack/congestion signatures then normal packet survival ratio can improve in leaps and bounds. Finding efficiently without false positives unresponsive ,TCP unfriendly flows is in itself is a big challenge because round trip time (RTT) , timeout time, route changes and normal congestion packet drops at other router on the path also affect response from legitimate clients. Isotropic (Highly distributed), slow rate attacks which even cause congestion at links are not identified in congestion signature without large number of false positives and false negatives. Attack agents (Zombies) which keep on regularly changing their source addresses without wrapping randomly are not identified. Moreover, slow rate degrading flooding DDoS attacks which do not cause congestion at links cannot be grouped in any congestion aggregates without high number of false positives and false negatives.

Server based attacks are addressed using resource accounting and QoS based solutions. These solutions result in high delays because of scheduling and queuing approaches to handle traffic. Moreover slow rate attacks where large number of attackers consume lot of bandwidth has no proper answer available so for. Client based programs required to be loaded for proactive server roaming has really hampered its popularity. However in limited attack scenario in terms of topology, number of attackers and different server based applications, its performance needs to be evaluated. Throttling techniques have assumed that web servers are attached to backbone routers so bandwidth of path links to server is not a concern in evaluating rate limits at k hops away which in recent literature is assumed to depend only on arriving rate of traffic at server/victim. Slow rate attacks using isotropic distribution yield very low NPSR in throttle techniques [46]. Still proper secure messaging system for control messages need to be found, which has perfect blend of security (confidentiality, authentication, integrity and freshness) and lesser overheads.

Accurate Characterization of Flash Crowds from DDoS attack traffic for better NPSR is still a pending issue.

There are many schemes which involve more than one kind of defense approach. These schemes fight against DDoS attacks in a distributed manner where different components of are placed at different points of the topology. DDoS is a distributed problem and must be addressed in a distributed manner. A review of distributed defense schemes in next section highlights state-of-art issues in DDoS defense.

2.3 Distributed Defense

Various DDoS defense techniques reviewed so far combine actions of victim-end, source-end and intermediate-network defense systems. Voluminous and distributed nature of DDoS traffic demands a distributed DDoS solution because centralized solutions cannot handle high overheads of monitoring, analyzing and filtering. Components of distributed defense system cooperate with each other to combat the attacks. Compared with the centralized defense systems, distributed defense systems can discover and fight the attacks with more resources and at more than one point of the Internet. It is very difficult for the centralized defense system to detect the attack at the beginning. When the attacks are full-fledged, it becomes more difficult for defense system to resist the flooding. And centralized defense systems themselves are more vulnerable to be attacked by hackers. The centralized defense systems are mostly deployed on the victim network because of economic reasons. Thus such defense systems are irresponsible systems which could only respond to the attacks, but not to stop the attacks.

Distributed defense systems overcome the shortcomings of centralized and isolated defense systems. Deployed on all around the Internet, distributed defense systems can detect the attacks before they are launched by inspecting the traffic on many edge networks in

which the computers are compromised by hackers. The most important and attractive feature of the distributed defense system is that the components in the distributed defense system can cooperate with each other to fight against DDoS attacks.

The advantage of distributed over centralized defense has been recognized in [6, 29, 172]. Some recently proposed defenses use collaborating source-end and victim-end nodes [29], while others deploy collaborating nodes at the victim and core networks [46]. While they perform well against a variety of attacks, they do not completely handle the flooding DDoS threat. Specifically, source/victim defenses fail to handle large attacks launched from legacy networks, while victim/core defenses inflict high collateral damage to legitimate traffic. A few defenses combine defense nodes at all three locations [6, 172]. These defenses achieve higher effectiveness, but focus on a single approach to defense (e.g., a capability mechanism in [172], victim-hiding in [6]), which ultimately discourages integration with other defenses and wide deployment. A review of some of well known distributed defense techniques is given below.

Pushback [84] enables routers to identify high-bandwidth aggregates that contribute to congestion rate limit them. If the congested router cannot control the aggregate itself, it requests its upstream neighbor's help in rate limiting. The performance of Pushback is good when attackers are collocated on a path separate from the legitimate traffic, otherwise it inflicts collateral damage. Further, Pushback cannot work in non-contiguous deployment and cannot detect attacks that do not congest core routers. By pushing the defense frontier towards attack sources, more legitimate traffic can be protected. An improved version of this pushback scheme called Selective pushback [155] sends pushback messages to the routers closest to the attack sources directly by analyzing the traffic distribution change of all upstream routers at the target. The benefit of this scheme is twofold. First, traffic

distribution analysis can locate attack sources more accurately than purely volume-based approaches, especially during a highly distributed denial of service attack. Second, the pushback message can be sent to the routers closest to the attack sources directly, which can mitigate the attack damage more quickly than the original pushback scheme.

Tupakula et al. [158], propose a controller agent model to counteract DoS attacks within one ISP domain which they later extended to multiple domains [157]. In this model, agents represent the edge routers and controllers represent trusted entities owned by the ISP. Once a target detects an attack, it sends a request to the controller, asking all agents to mark all packets to the target After checking the marking field, the target can find out which agent (edge router) is the entry point for the attack traffic. The target then sends a refined request to the controller, asking some particular agents to filter attack traffic according to the attack signature provided by the target. So attack traffic originating from zombies is filtered at ingress edges of the protected ISP, but legitimate traffic is allowed to enter the domain. In [157] designated controllers of multiple domains interact to decrease the impact of attack and Traceback the attack path till attack zombies. One point to be noted in this model is that it uses third party detection for detecting and characterizing attack traffic.

This is a good model in terms of number of packets required to find ingress edges of attack, but attack signature should be as narrow as possible to lessen collateral damage. The communication required between victim and controller as well between agents and controllers should be first possible in state of DDoS and should also be confidential, authentic, integral and fresh. Moreover single point failure at controller due to DDoS attack centered at controller or intrinsic fault can really damage the whole scene. Also filtering techniques are used to stop the attack. Instead adaptive rate limit would be better if attack signatures are not accurate.

SOS [6] uses access points (SOAPs) close to source networks to verify legitimate users and send their traffic on the overlay to secret servlets that tunnel it to a distributed firewall protecting the victim. SOS offers good protection to the server but the traffic experiences a significant delay because it is routed on the overlay. SOS approach involves a variety of authentication and overlay routing mechanisms and suffers from routing related drawbacks. Moreover, if attackers can gain massive attack power, for example, via worm spread, all the SOAPs can be paralyzed, and the target's success will be disrupted.

Active Security System (ASSYST) [125] supports distributed response with non-contiguous deployment, with nodes equivalent to classifiers being deployed only at edge networks. CROSSACK [29] similarly forms a multicast group of defense nodes that are deployed at source and victim networks and cooperate in filtering the attack. Both [125] and [29] cannot handle attacks from legacy networks that do not deploy their defense mechanisms. Parameter Based Defense [139] constructs a multicast group at an ISP that rate limits an attack originated from one of its customer networks. It requires wide deployment and does not perform well in non-contiguous deployment. Yau et al. [46] propose a router throttle mechanism installed at the routers that are close to the victim. This defense system incorporates only victim-end and core defense mechanisms, and thus inflicts collateral damage to legitimate traffic. Some router based solutions consists of an overlay of routers with added functionality, which helps them trace and stop the attacks close to the source. Tracing is done using signatures assigned to each source network, and inflicts collateral damage on legitimate users that share a network with an attacker.

DefCOM [62] provides added functionality to existing defenses so they can collaborate in DDoS detection and response though a dynamically-built overlay. There are three types of DefCOM functionalities that are added to existing routers or defense nodes. A single

physical node can host more functionality at a time. The functionalities are: (1) A classifier functionality is added to existing defenses that is capable of differentiating the legitimate from the attack traffic. A classifier marks packets recognized as legitimate with a HIGH-priority mark that guarantees priority handling by downstream DefCOM nodes. (2) A rate limiter functionality is deployed by routers. During an attack, a rate limiter runs a weighted fair share algorithm (WFSA) to prioritize traffic it forwards to the victim, and it rate limits this traffic to preserve victim's resources. (3) An alert generator functionality is added to defenses that can detect a DoS attack. An alert generator propagates the attack alert to other DefCOM nodes using the overlay. The alert contains the IP address of the attack's victim and specifies a desired rate limit, e.g., the size of the victim's bottleneck link. Extra infrastructure for overlay and cooperation at all points of the Internet are big concerns. Collateral damage depends upon accuracy of classifier.

Some of distributed systems to defend DDoS attacks are evaluated in terms of deployment, detection, response, security, robustness and implementation. Defense Systems compared are ACC [84, 129], SOS [6], Controller-agent [157, 158, 161], Throttling [46], DiDDeM [83], MANANet [112], CROSSACK [29], IDIP [52], ASSYST [125], and DefCOM [62]. Table 2.1 shows the summary of comparison among these systems.

Different strategies are employed by these systems. We can not say one system is better than the other because different systems are applied onto different scenarios. However, we can compare the strategies used by these systems to get the insight of what strategies are more useful in the campaign with DDoS attacks.

2.3.1 Deployment

Since a distributed defense system has many nodes that can be homogenous or heterogeneous nodes, these nodes must be deployed at different locations in the network.

Table 2.1: Summary of comparisons among distributed defense systems

	ACC	Controller-Agent	Throttle	DiDDeM	SOS
Deployment	Throughout the network	ISP domain	Routers close to victim	Multiple ISP domains	Source/ Victim
Detection	Congestion based	Third party Intrusion detection system	N/A	Congestion based	Filtering
Response	Rate limiting	Dropping all packets	Rate limiting	Dropping all the packets till number of packets dropped less than threshold	Rate Limiting
Security	N/A	Analyzed in later versions	N/A	N/A	IPSec
Robustness	Weak	Dynamic generation of Agent IDs	Weak	---------- -	Moderate
Implementation	Difficult	Practical provided incentives are given to ISPs.	Difficult	Difficult	Difficult

Table 2.1: Summary of comparisons among distributed defense systems (contd.)

	MANANet	CROSSACK	IDIP	ASSYST	DefCOM
Deployment	Cooperative routers at the Victim	Source/ Victim	Distributed groups	Throughout the network	Throughout the network
Detection	PEIP	Spectral analysis	Intrusion detection	Intrusion detection	Traffic tree discovery
Response	Rate Limiting	Dropping all packets	Dropping all packets	Dropping all packets	Rate limiting
Security	N/A	CA	IPSec	N/A	PKI
Robustness	Weak	Weak	Weak	Weak	Weak
Implementation	Difficult	Easy	Easy	Difficult	Difficult

The functionalities of defense nodes include detection of potential attack, alarm generating and multicasting, attack source finding, and attack traffic controlling. Different nodes can be deployed at the edge networks and core networks. Some approaches such as DefCOM [62], ACC [129] and ASSYST [125] deploy their nodes throughout the network. This deployment requires that every participating node must be able to perform the detection and traffic controlling functions, communicate and coordinate well with each other. It could raise the unnecessary traffic burden at the intermediate nodes. Moreover, it could not be the best place to detect the attack at the intermediate nodes. The best deployment is the mixture deployment at both source end and victim end. The reason for this deployment is that first, the victim end aggregates the most information for the detection and can achieve the most accurate detection true positive rate. Second, detecting preliminary attack signatures at source end allows the defense system to mitigate a DDoS attack at its initial phase. Third, the source end traffic controlling can protect the network's availability to a max degree because not only the victim but also the rest of network can be free of network congestion. But practically DDoS traffic is so low at sources that it is not easy to characterize attacks packets at the source. At victim end automation of real time response by generating alerts and collaborating with other defense nodes is difficult against high volume of DDoS attack traffic. Besides wide base of vulnerable machines on the Internet and distributed Internet control, global deployment at source is too difficult without explicit incentives.

Though, ISPs do not have enough financial motivation to install DDoS defense systems currently, but being infrastructure wise adequate and under single administrative control, ISPs seem to be the best place to handle DDoS attacks. Detection module can be put near the victim and filtering can be done at all ingress links of the ISP. It will not only distribute filtering overheads but also saves expensive core bandwidth.

2.3.2 Detection

Flooding DDoS attacks bring network anomaly such as the sudden surge of network traffic volume, increase of the packets with random source IP addresses, and asymmetric amount of packets associated with some network protocol such as TCP SYN. Detection and filtering is a straightforward approach to defend such attack. One of the main objectives of a successful distributed defense system is the fast and sensitive detection by using a fine granularity detection method. In most of distributed defense systems which are either traceback based or tolerance based uses either congestion based or third party intrusion detection systems.

Though, detection of high rate flooding DDoS attacks is easy at the victim, but due to excessive DDoS traffic, response is initiated manually in most of the cases. So a real time detection and automated response needs to be dealt more carefully at appropriate point of network where excessive traffic can be handled in a better manner. Moreover, selection of thresholds and their impact on detection accuracy needs to be analyzed properly so as to give meaningful direction to DDoS research [60].

2.3.3 Response

Rate-limiting and throttling are the most popular strategies used in the current distributed defense systems, such as in DefCOM [62], SOS [6], ACC [129], MANANet [112] and Throttle [46]. Because no defense systems can detect the attacking packets with 100 percent accuracy, it is advisable to limit the rate of high-bandwidth flows rather than to drop all the suspicious packets.

It also gives the defense system flexibility to adjust the limit to which the suspicious network traffic is suppressed. The disadvantage is that it allows a certain amount of attacking packets to pass through and some legitimate packets are either delayed or

dropped. This will bring problems when rate limiting is deployed on the network in which there are resource-demanding applications (e.g. video stream) and the bandwidth is not big enough. However, currently there seems to be no better solutions unless the detection accuracy can be improved to a satisfying extent.

2.3.4 Security

A distributed defense system must be able to protect the information to be exchanged from being intercepted by the hackers. Current security mechanisms such as IPSec, PKI etc. are used to meet the requirement. The examples of security implement are highlighted in [168]. Some research has been done to deal with the denial of service problems in the security protocols [85, 118]. An analysis of DDoS defense in terms of security is also done in [159] for controller agent model [158]. Here we do not specifically consider how to defend the security architecture because we assume the motivation of the DDoS attacks is to prevent the legitimate users from accessing the desired resources, but not to crash the security architecture, which is more difficult to achieve.

2.3.5 Robustness

Here robustness means the degree to which the distributed defense system itself can resist the attacks. When the distributed defense system is deployed and is known to the hackers, they will launch attacks to the distributed defense system so that the defense systems cause denial of service to protected systems. Although the distributed defense system is less vulnerable to such attacks than the centralized defense system, it is still possible that the distributed defense system fails due to the attacks targeting it. Unfortunately this issue is less concerned in the design of the current distributed defense system.

The concentration point of flooding DDoS traffic is victim so more attack evidence is also available near the victim. Detection of attack and characterization of attack sources can

be done best near the victim. However, state monitoring and sophisticated analysis to capture all kinds of attack require higher computational complexity, which is vulnerability in case of high rate flooding DDoS attacks near the victim at single point. Distributed defense systems in which detection and characterization is done at single point, higher computational complexity vulnerability can really cripple detection system which is an integral part of whole distributed defense system.

A detection system which can quickly detect (Low rate or start of high rate attack) and react to flooding DDoS attacks such that it does not give chance of accumulation of attack traffic at victim can solve a lot of problems. At the same time, computational complexity of detection scheme should either be minimized or distributed without compromising complete view of attack traffic.

2.3.6 Implementation

If a distributed system is in good design and has good experimental results, such system still cannot be accepted by the security community if they are not easy to be implemented or even impossible to be implemented under current Internet infrastructure. Most of distributed defense systems e.g., DefCOM [62], SOS [6], ASSYST [125], ACC [129] and MANANet [112] need routers to support specific functions. So a large portion of current distributed defense systems require the Internet infrastructure to be modified. That is one of the reasons why no successful solutions which can defeat DDoS attacks are available up to now. In the medium term, it is expected that ISPs will begin to deploy more distributed defense systems at the ingress and egress points of their networks. DefCOM [62] has worked on economic model for wide deployment but still the longer-term challenge for defense against these attacks is to find technical and economic models to achieve cooperation between ISP [154] to combat DDoS attacks collaboratively. An ISCP protocol (Inter-Domain Security

Management Agent Coordination Protocol) is one of the efforts being done in this direction [181].

2.4 Conclusion

In this chapter, a review of DDoS defense approaches is presented. Later in the chapter important distributed defense systems to combat DDoS attacks are reviewed on the basis of characteristics of defense systems. The gaps in existing work are identified and highlighted. Efforts are made to address some of these gaps as part of our work in subsequent chapters.

Chapter 3

D-DCFI: DDoS Detection Characterization and Filtering in ISP Domain

Distributed denial-of-service (DDoS) traffic is highly distributed, and usually consists of legitimate packets, generated in huge quantity. These characteristics make real time detection of DDoS attacks and identification of attack packets some of the hardest problems for security experts. Most of the existing solutions [8, 9, 28, 31, 59, 85, 115, 150, 176] for detecting DDoS attacks are aimed at aiding end-node victims under attack. However, very little attention has been given to this problem from an Internet service provider (ISP) perspective. In this chapter, we propose a traffic feature distribution based approach to detect, characterize and filter DDoS attacks in ISP domain. Simulation experiments are carried out in NS-2 testbed at different attack strengths to validate our countermeasure at very low and high rate flooding DDoS attacks. An Internet type topology is used to test the proposed scheme. Detection thresholds and efficiency are justified using receiver operating characteristics (ROC) curves [13]. Normal packet survival ratio (NPSR) is computed from offline traces to evaluate relative effectiveness of proposed and existing schemes.

3.1 Introduction

Currently, the majority (90-94%) of DDoS attacks are performed using TCP, and a large portion (52-57%) of them is targeted to bandwidth exhaustion [50]. The work in this thesis

concentrates on TCP low rate flooding DDoS called LRFD and high rate flooding DDoS called HRFD attacks. It proposes a DDoS defense system to curb LRFD and HRFD attacks in an ISP domain called D-DCFI. Simulation scenarios consist of TCP based server so, only TCP part of the legitimate traffic is protected in D-DCFI. Considering that many public servers provide services through TCP, protecting the TCP portion of the bandwidth is sufficient in protecting most of services available on the Internet. However, the work can be easily extended to include legitimate UDP traffic as it does not involve TCP specific models at any stage.

The Flooding DDoS traffic completely blends itself with small amount of legitimate traffic in such a manner that no differentiation can be made on packet-by-packet basis [91]. It forces DDoS detection systems to install a real time packet monitoring process. The traffic features (i.e., source address, destination address, source port, destination port, and protocol type) in real time traffic are monitored and further analyzed to find attack signs within amassed DDoS and legitimate traffic. The existing methods [17, 18, 79, 104, 110, 116, 124, 134, 153] use volume based metrics (number of packets and bytes count per unit time) to detect and characterize DDoS attacks. These schemes are better suited to HRFD attacks, which completely disrupt the services to legitimate clients. LRFD attacks consume a small portion of victim's resources, and are not detected using these schemes. However, the accurate detection of LRFD attacks is very important. First, it enables the early detection of high rate attacks whose intensity slowly increases. Secondly, detection closer to source is possible, which is otherwise very difficult because of lesser volume of attack traffic at source. Moreover, the response provided by these detection schemes suffers from large collateral damage especially for LRFD attacks and highly distributed (Isotropic) HRFD attacks at source.

Lakhina et al. [12] observed that most of traffic anomalies such as port scan, network scan, worms, and DoS described in [11], in spite of their different nature induce a change in distributional aspects of traffic features. D-DCFI uses anomaly based detection in ISP domain. It interprets flooding DDoS attacks as events that disturb the distribution of traffic flows. Here the traffic flow is a set of packets satisfying a 5-tuple (source address, destination address, source port, destination port, and protocol type) qualifier, monitored in a time window. Though, D-DCFI uses traffic feature distributions to detect flooding DDoS anomalies at various attack strengths, but the effort in [12] has demonstrated that these distributions have the potential to detect various network anomalies [11].

Feinstein et al. [102] uses sample entropy for DDoS detection. Source IP is the only traffic feature used for computation of sample entropy, whereas actual flooding DDoS attacks also use random ports in their packets. Accordingly, for low rate attacks the dispersion in traffic flow distribution does not increase too much if random ports are used for a set of source IP addresses. However, the proposed D-DCFI uses 5-tuple flows and treats different port number for same source address as different flow. As a result, number of traffic flows having a small share of arrived packets increases, which in turn increases sample entropy. Though D-DCFI induces high state monitoring overheads, but a small observation window, sampled traffic [24], and fast monitoring adapters [81] can handle the traffic load. Moreover, characterization of attack traffic is based on anomalous bins in [102] whereas bins also contain legitimate traffic. A packet window of 10,000 packets for finding frequency of occurrence of unique SourceIP is used in [102]. The increase in sample entropy value indicates presence of attacks in [102]. But the detection of LRFD and HRFD attacks separately is not demonstrated with results. The choice of threshold as per different network environments and effect of tunable parameter variations on performance is not

presented with results. D-DCFI however, provides a formal definition of detection model. D-DCFI uses time window instead of packet window in packet monitoring process. A systematic study of threshold setting as per network environment is discussed using ROC curves [13]. The variation of performance with change in tunable parameter is discussed with results in this work.

The NS-2 [114] testbed is used for implementation and evaluation of our approach. An ISP level topology is used for simulation experiments. Transit-stub model of GT-ITM [65] topology generator is adopted for creating topology consisting of four ISPs.

The chapter is organized as follows. Section 3.2 describes motivation for DDoS defense in ISP domain. Section 3.3 elaborates the utility of traffic feature distributions for diagnosing anomalies in general and DDoS in specific and introduces sample entropy metric to summarize traffic feature distributions. Section 3.4 presents design of the D-DCFI and details key components of the system. Design of simulation experiments is given in section 3.5. Section 3.6 illustrate detection of attack using simulation results and portrays tradeoff between detection accuracy and false positives to set detection thresholds. In section 3.7 impact of LRFD attacks are discussed. Section 3.8 describes comparison with existing techniques. Finally, Section 3.9 concludes the chapter.

3.2 DDoS Defense in ISP Domain

A typical DDoS defense system consists of detection of attack, characterization of attack sources, and filtering of attack traffic. It can either be deployed as a single-point system or as a distributed system. A single-point system consists of a single defense node that observes the attack, analyses the traffic and applies the response. Distributed systems consist of multiple defense nodes that are deployed at various locations on the network.

Defense nodes communicate through the network and coordinate their actions to achieve a better overall performance.

On the Internet, DDoS attack streams originate from geographically distributed machines, are forwarded by core routers and converge at the victim network. There is interaction of three types of networks: source networks that unwittingly host attack machines, several intermediate networks that forward attack traffic to the victim, and the victim network that hosts the target. Figure 3.1 depicts this interaction [91]. Each of the involved networks (source, intermediate, and victim) can host DDoS defense systems. We analyze feasibility of DDoS defense deployed at each of these individual points.

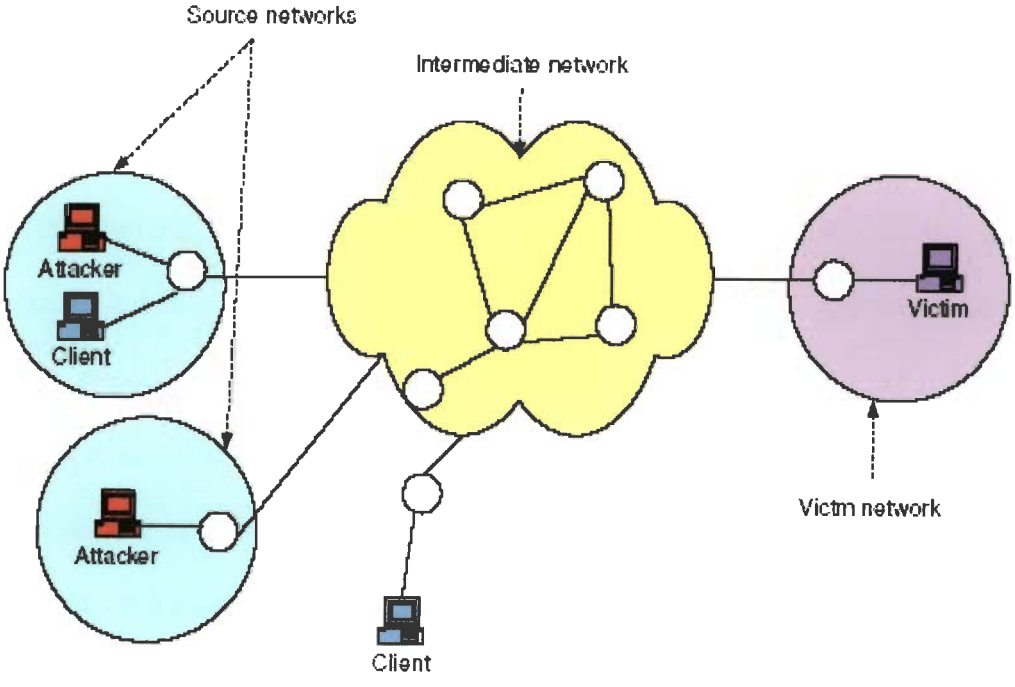

Figure 3.1: Points of DDoS defense

The placement of DDoS defense logic at a particular point of the Internet is an important concern, as Internet has decentralized management.

Prevention methods, such as Ingress/Egress Filtering [119] and repairing security holes [171], are implemented at source networks to stop origin of DDoS traffic. Absence of

incentives, per packet filtering overheads, and security measures awareness stand in the way of DDoS defense deployed at the source network. D-WARD [91] is also a source-end defense scheme. It faces two hard challenges. First, in a highly distributed attack (i.e. isotropic DDoS attack), each source network is responsible for only a small fraction of the attack traffic, which is unlikely to generate anomalous statistics. Secondly, a witty DDoS attacker can also control the attack traffic from each source network to be within normal range because ultimately it is the aggregation of attack traffic and not individual source traffic which is going to inflict damage to the victim. Moreover, the biggest problem in source-end defense is requirement of global deployment, which is impossible to achieve as Internet has no central control.

Historically, most of existing DDoS defending systems: resource accounting [8, 9, 59, 115, 176] and protocol security mechanisms [28, 31, 85, 150] have been designed to work on the victim side. DDoS attacks have maximum impact on the victim, so the motivation for DDoS defense deployment on the victim side is also justified. However, under a sustained high bandwidth DDoS attack, it is not possible to contain the attack at border gateway and/or firewall [113] in the victim side. The offending packets actually consume the finite bandwidth available on the connection to the ISP. Therefore, the legitimate packets are not able to even reach at the victim side. Hence, filtering on victim side has no meaning as it cannot protect legitimate traffic.

Many solutions, such as pushback [129], SOS [6], and traceback [5, 40, 53, 137, 147] are deployed at the intermediate network i.e. in the core of Internet. They all put burden on core routers, which are meant for forwarding packets at high speeds as per Internet design. Besides, intermediate network is not owned by single administrative domain. So, establishing cooperation and trust relationships between different domains, such that

requests originating from one domain will be honoured by the other or the module to be installed in other domain will be allowed, are the concerns that have practically no answer. A similar effort in preventing privacy violation using non disclosure agreements (NDAs) is made for achieving cooperation between provider and receiver organization [164].

Distributed defense techniques are likely to be the proper solution for handling the DDoS threat [108]. However, they are infrastructural solutions i.e. they span multiple networks and administrative domains and represent major undertakings of many Internet participants. Such systems are difficult to deploy and maintain. Further, the required cooperation of defenses is hard to achieve due to distributed Internet management and strictly autonomous operation of administrative domains. Securing and authenticating the communication channels also incurs a high cost if the number of participants is large. In light of above said issues and Internet design vulnerabilities [89], a practical DDoS defense system deployment should have following important characteristics:

- Autonomous system i.e. whole defense location under one administrative control so that different defense nodes can collaborate in a secure manner.
- Large and infrastructure wise rich enough to handle high voluminous traffic from evenly distributed flood sources.
- Capability to evolve DDoS defense in incremental fashion.
- Sufficient financial motivation for value-added DDoS security service.

The Internet consists of thousands of Autonomous Systems (ASes) i.e., networks that are each owned and operated by a single institution. Usually each ISP operates one AS, though some ISPs may operate multiple ASes for business reasons (e.g. to provide more autonomy to administrators of an ISP's backbones in the United States and Europe) or historical reasons (e.g. a recent merger of two ISPs) [105]. An ISP has total autonomy to collaborate

defense nodes in a secure manner. Enough infrastructures can be provided for DDoS defense to handle high volume at ingress points. Moreover, once agreement is reached between various ISPs then inter co-operation among ISPs is also possible [139, 157]. Accordingly, there is scope of incremental DDoS defense. If a provider's infrastructure is attacked (routers, DNS, etc.), all services to its customers fail, resulting in service level agreement (SLA) violations. Moreover, ISPs normally host most of the services available on the Internet. The cost of DDoS protection is insurance against catastrophic failures that would cost the business orders of magnitude more in terms of both revenue and negative customer relations. However, Cost-avoidance is not the only motivation to implement a complete DDoS solution in ISP domain. For the users, DDoS protection can also be offered as a value-added service that creates new revenue streams and provides competitive differentiation for ISPs. In nutshell, ISP level DDoS defense is most practical and viable at this stage. Though, longer term objective "how to achieve inter ISPs cooperation" still remains as the biggest challenge. D-DCFI is a distributed approach to defend DDoS attacks. It detects flooding DDoS attacks at the POP connected to victim server and filter DDoS traffic at responsible POPs closer to source of the attack in the same ISP domain. All these characteristics make D-DCFI a practically viable and feasible. The POPs of the ISP connected to victim network filter the offending traffic and allow legitimate traffic to pass through to the server. By blocking the attack traffic at the POPs, D-DCFI offers protection from DDoS attack without requiring an upgrade of the core routers or coordination with other service providers. Specialized monitoring adapters [81] can be used for very high-speed network links or edge routers can take the burden of analysis within the POP. The extra cost on the ISP in terms of analysis and packet filtering overheads is much less than the benefit an ISP can offer to its clients. Moreover network analyzers are available as

freeware on the Internet [57]. PickPacket [21] is a network monitoring tool that can be used for packet analysis.

3.3 Traffic Feature Distributions

A traffic feature is a field in the header of a packet. The focus in D-DCFI is on five fields of an IPv4 packet viz. source addresses denoted as srcIP, destination address denoted as dstIP, source port denoted as srcPort, destination port denoted as dstPort, and protocol type denoted as proType. The packets having the same 5-tuple of 104 bits are considered in one traffic flow in the same observation window.

The underlying idea is to exploit changes observed in the distribution of addresses or ports under attack to characterize important traffic anomalies. Table 3.1 lists a set of anomalies commonly encountered in backbone network traffic with their affect on traffic feature distributions [11].

Each of these anomalies affects the distribution of certain traffic features. In some cases, feature distributions become more dispersed e.g. when source addresses are spoofed in DoS attacks, or when ports are scanned for vulnerabilities. In other cases, feature distributions become concentrated on a small set of values e.g. when a single source sends a large number of packets to a single destination in an unusually high volume flow. These can be taken as strong signatures for the purpose of detecting anomalies. Clearly dispersion of the traffic feature distributions plays a major role in detecting all of these anomalies.

In case of DDoS attacks, a cohort of malicious zombies (unprotected computers accessing the Internet on which attack daemon programs are installed) transmits flooding traffic using random packet header fields towards the victim server. At initial stage of attack, flooding traffic is normally low but with time the sheer volume can cripple even well secured systems. Quite evidently, it is vital to pick the attack as early as possible.

Table 3.1: Effect of anomalies on feature distributions

Anomaly Label	Definition	Traffic Feature Distributions Affected
Alpha Flows	Unusually large volume point to point flow	Source and destination address (possibly ports)
Flash Crowd	Unusual burst of traffic to single destination, from a "typical" distribution of sources	Destination address, destination port
Port Scan	Probes to many destination ports on a small set of destination addresses	Destination address, destination port
Network Scan	Probes to many destination addresses on a small set of destination ports	Destination address, destination port
Outage Events	Traffic shifts due to equipment failures or maintenance	Mainly source and destination address
Point to Multipoint	Traffic from single source to many destinations, e.g., content distribution	Source address, destination address
Worms	Scanning by worms for vulnerable hosts (special case of Network Scan)	Destination address and port
DoS	Denial-of-service attack	Destination address

The number of zombies in a typical DDoS attack can vary from one hundred to more than 100,000 [154]. These zombies either use own or spoofed IP addresses. The use of large number of zombies and spoofed addresses triggers change in source IP distribution. Moreover, victim address is also common, thus distribution of destination IP will be more concentrated on victim address. Similarly, due to random selection of source and destination ports by popular DDoS attack tools, the distribution of source and destination ports also gets affected. D-DCFI makes use of flow based distribution to include all of affected traffic features. In case of LRFD or at start of HRFD attack, the total traffic volume received at the server is very less. But distribution of traffic features like source address, source port, destination address, and destination port, and hence flow is affected appreciably. Moreover,

it is a well known fact that positively and negatively skewed distributions have more concentration of the observations towards the higher and lower values respectively [66]. Accordingly, HRFD and LRFD attacks can also be differentiated.

In general, two properties of traffic distributions namely dispersion and skewness are used to characterize attack anomalies. A metric that captures the degree of dispersal or concentration of a distribution is called sample entropy [23].

Let $X = \{n_i, i = 1,....., N\}$ is the frequency distribution consisting of N features where feature i occurs n_i times in the sample.

Let $S = \sum_{i=1}^{N} n_i$ be the total number of observations in the distribution. It actually represents total number of packets observed in the sample.

Let $p_i = n_i / S$ be the probability of occurrence each feature i in the sample.

Then, the sample entropy $H(X)$ is calculated as

$$H(X) = -\sum_{i=1}^{N} (p_i) \times \log_2 (p_i) \tag{3.1}$$

where X is the frequency distribution consisting of N flows. It actually gives number of packets arrived per flow in the sample.

Flow i has n_i packets arrived in the sample.

The value of sample entropy lies in the range $(0 - \log_2 N)$. The metric takes on the value 0 when the distribution is maximally concentrated, i.e. all observations are the same. Sample entropy takes on the value $\log_2 N$ when the distribution is maximally dispersed, i.e. $n_1 = n_2 = ...n_n$.

Figure 3.2 depicts maximum sample entropy values for number of flows. The number of legitimate flows except SYN requests cannot increase more than valid number of connections for a server that uses TCP service at transport layer. Accordingly, keeping an account of valid number of connections can put a maximal constraint on sample entropy value to detect any kind of anomaly.

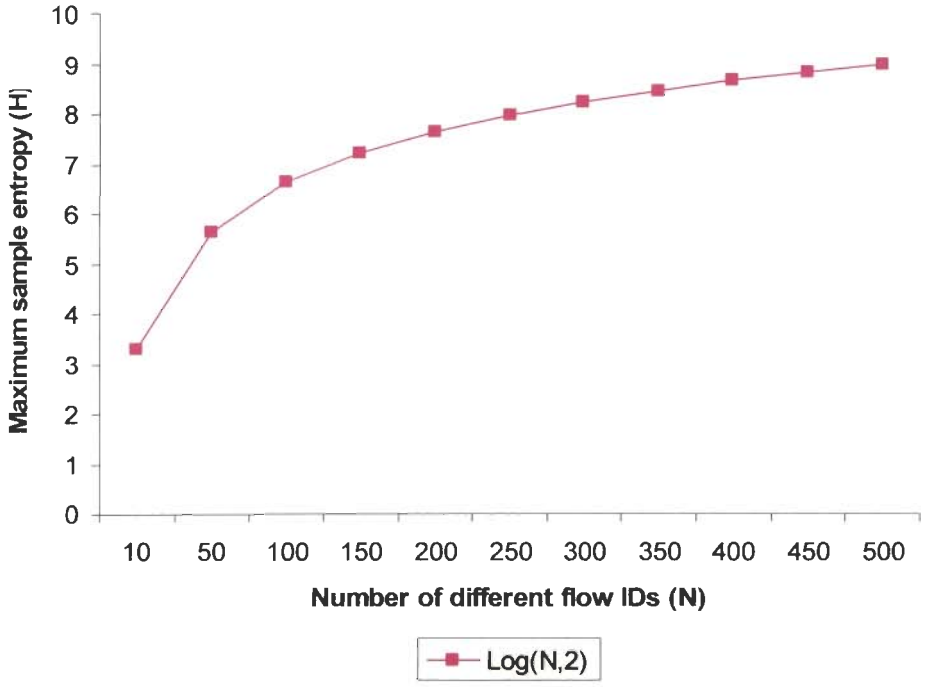

Figure 3.2: Maximum value of sample entropy

Moreover, a LRFD consists of a lot of flows having lesser number of packets than legitimate flows in an observed sample of packets. This results in a negatively skewed distribution. Similarly, in HRFD attack, mostly attack flows contribute to more number of packets than legitimate flows, which result in a positively skewed distribution. Sample entropy $H(X)$ for negatively skewed distribution is more whereas for positively skewed distribution its value is lesser than normal $H(X)$ without attack [66]. The above said facts

indicate that sample entropy $H(X)$ is an effective summary statistic for characterizing a distribution and detection of flooding DDoS attacks.

3.4 D-DCFI: DDoS Detection Characterization and Filtering in ISP Domain

The end systems or hosts (users PCs, PDAs, web Servers, and mail servers etc) connect to each other through a tiered hierarchy of ISPs in the Internet. Each tier of hierarchy is different in terms of service coverage, and bandwidths of internal links. Upper tier ISP is service provider, and lower tier ISP that connects to it is its customer. In the ISP's network, the point at which the ISP connect to other ISP (whether below, above, or at the same level in the hierarchy) is known as a point of presence (POP). The interconnection of POPs of an ISP through high bandwidth links is called ISP backbone. In an ISP's network a POP is actually a group of connected core and access routers. POPs are connected to each other at core routers (private/public peer or NAT). The customer domains are attached to POP at access routers.

An ISP level Internet topology, consisting of four ISP domains is considered for simulation. Each ISP domain has 10 POPs. The POPs are represented as single node in Figure 3.3. One customer domain is attached to each POP. Customer domains consist of legitimate and attacking hosts. Two POPs in each ISP are attached to other ISPs and ISP domain 4 has one additional POP. The additional POP P_s is connected to the protected server. The aim of D-DCFI is to protect ISP domain 4 from DDoS attacks. Detection of flooding DDoS attacks and characterization of attack flows are performed at POP P_s. The framed filtering rules are transmitted to all POPs of ISP 4 through multicasting. Hence the DDoS attacks originated in ISP domain 4 are filtered near the source whereas attacks coming from ISP domains 1, 2 and 3 are filtered at peering points. D-DCFI can be extended

to propagate filtering logic in multiple ISP domains by using communication framework given in [157] and [139]. The legitimate and attack traffic is generated from all four ISPs. In the very first step, time series [54] monitoring of traffic directed to protected server is performed at P_s. The procedural flowchart of the proposed approach D-DCFI is given in Figure 3.4.

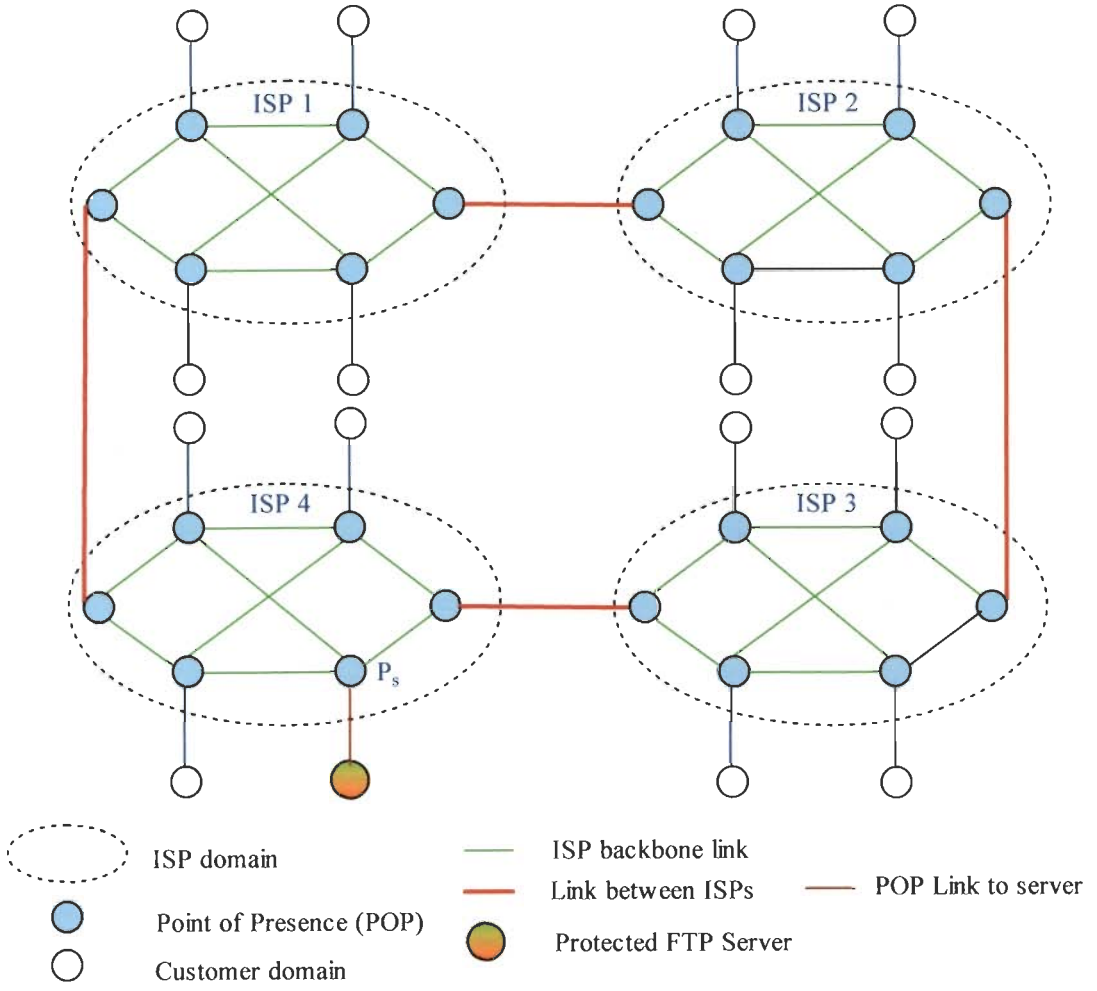

Figure 3.3: A short scale simulation topology

The monitoring process yields packets arrival distribution X of flows, and total number of packets S. In flowchart shown in Figure 3.4, ANFA array indexed by flow ID i represents X, and APA represents S. Second step computes the sample entropy $H(X)$ by using equation (3.1). In third step the computed sample entropy value in step 2 is compared

with already profiled normal value of sample entropy. The deviation of computed and profiled entropy beyond threshold marks a flooding DDoS attack. The sign of deviation distinguishes LRFD and HRFD attacks. A variable ATTACK is used to represent LRFD attack (ATTACK=0), HRFD attack (ATTACK=1), and no attack (ATTACK=-1).

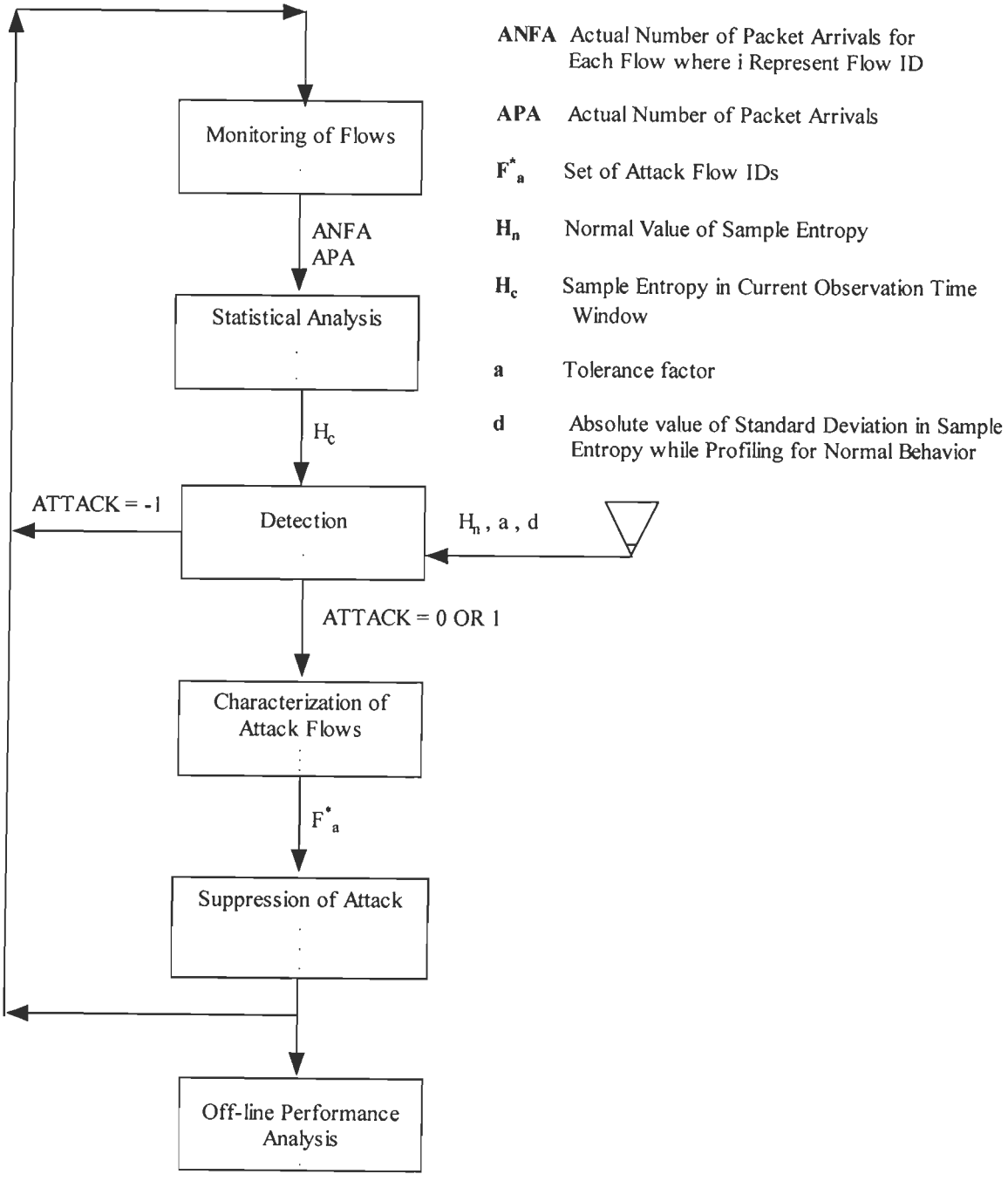

Figure 3.4: Procedural flowchart of D-DCFI

Detection of attack is followed by Characterization of attack flows in next step. Direction of skewness in X and total increased traffic after launch of flooding DDoS attack are used to isolate set of anomalous flows F_a^* . Suppression of attack module explains attack response provided in D-DCFI. Filtering rules are framed using F_a^* got from previous step, and are communicated to all POPs of ISP 4 through multicasting. The POPs of ISP 4 are registered to a multicast group. The address of the group is local to the ISP 4 only. The POP P_s is responsible of coordinating the DDoS defense. The direct support of multicasting in NS-2 helped P_s to send filtering rules to all POPs of ISP 4.

As per filtering rules, packets are filtered at all responsible POPs of ISP 4. Thus, victim server is relieved from flooding DDoS attacks in real time. Hence, D-DCFI is a distributed approach in which various functional modules are placed at different points of the same ISP domain.

The last stage is responsible for measuring effectiveness of the approach. Legitimate traffic service level is one of the main DDoS defense effectiveness measure as recommended in DDoS evaluation project [90]. Normal packet survival ratio (NPSR) is measured as a fraction of legitimate traffic in total arrived traffic. It is a good metric to represent legitimate traffic service level as it represent legitimate traffic with respect to attack strength. An offline trace generated by NS-2 is used to compute NPSR at desired time granularity. Subsequent subsections details each step of procedural flowchart in Figure 3.4.

3.4.1 Monitoring

The overhead of monitoring packets at POP makes it infeasible to keep traffic statistics of all destinations of an ISP. The traffic destined to protected server is monitored at link

between P_s and the server. Practically, sample and hold algorithm [24] is used instead of monitoring all traffic to protected servers. Packets are monitored in a short sized time window to minimize memory overheads. In D-DCFI, random packet process $\{X(t), t = j\Delta, j \in I\}$ monitors traffic between POP P_s and protected server where Δ is a constant time interval called time window, I is the set of positive integers, and for time $t, X(t)$ is a random variable that represents frequency distribution consisting of actual number of packet arrivals for all flows in $\{t - \Delta, t\}$. The detailed flowchart for monitoring process is given in Figure 3.5.

The headers of each packet are detached to classify the packet to a particular flow ID i using an inbuilt traffic classification mechanism. The array TNFA indexed by flow i is incremented by 1 as shown in Figure 3.5. The packet monitoring loop continues till time t is less than $j\Delta$. Once time t equals $j\Delta$, the actual number of packets ANFA are calculated for each flow i. The output ANFA is actually $X(t)$ and APA represents actual number of packet arrivals S for all flows in current time window $\{t - \Delta, t\}$ as shown in table 3.2.

Table 3.2: Frequency distribution of packet arrivals $X(t)$

Flow ID i	**Actual Number of Packet Arrivals n_i**
1	n_1
2	n_2
3	n_3
.	.
.	.
N	n_N

In table 3.2 N represents total number of flows and $S = \sum_{i=1}^{N} n_i$

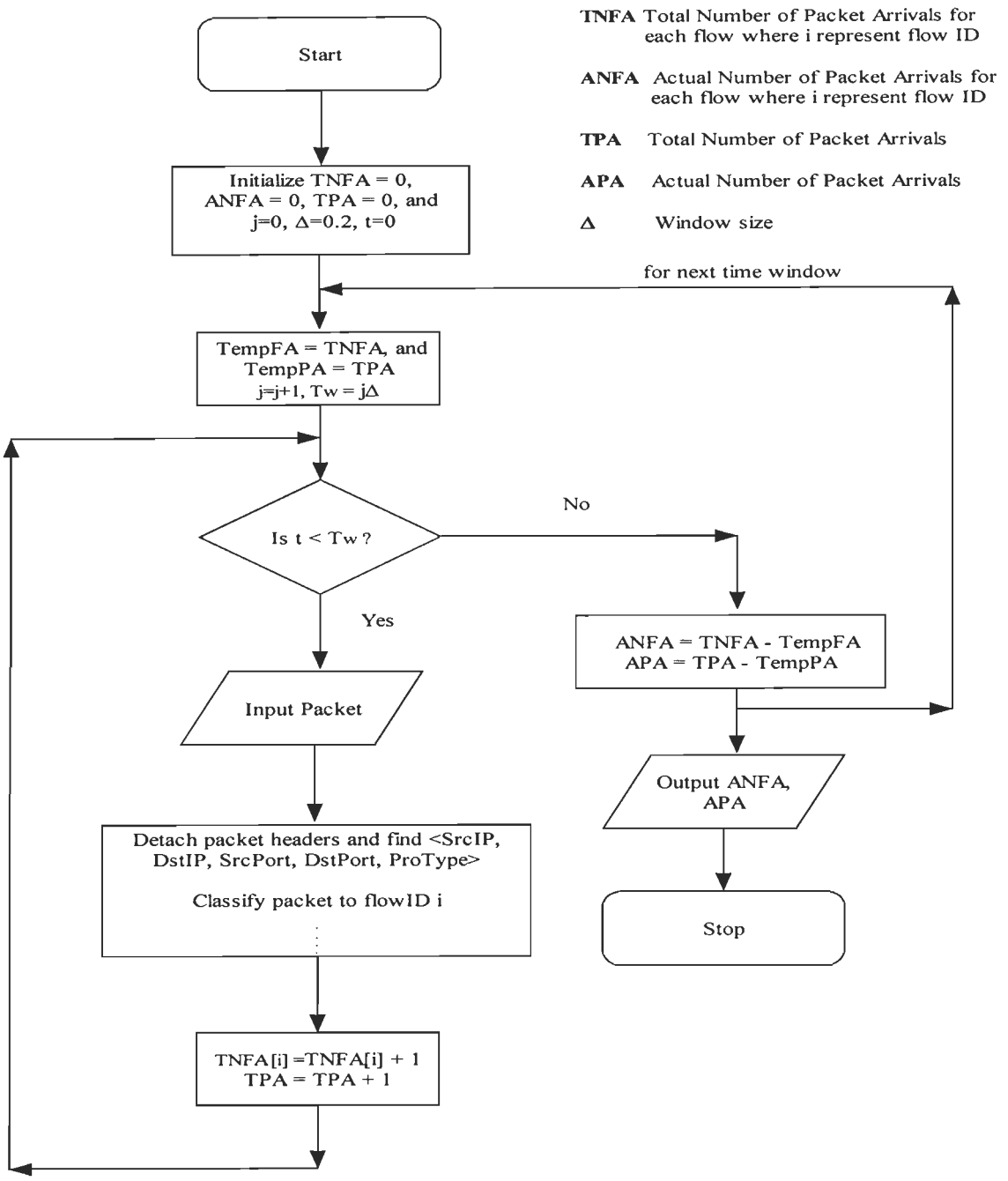

Figure 3.5: Flowchart for packet monitoring process

3.4.2 Statistical Analysis

The frequency distribution $X(t)$ generated in the previous step is used for computing sample entropy $H(X)$ by using equation (3.1). Detailed flowchart for the same is shown in

Figure 3.6. First p_i and then $p_i \times \log_2(p_i)$ are calculated for all N flows. The sum of $p_i \times \log_2(p_i)$ for each flow ID i gives sample entropy $H(X)$.

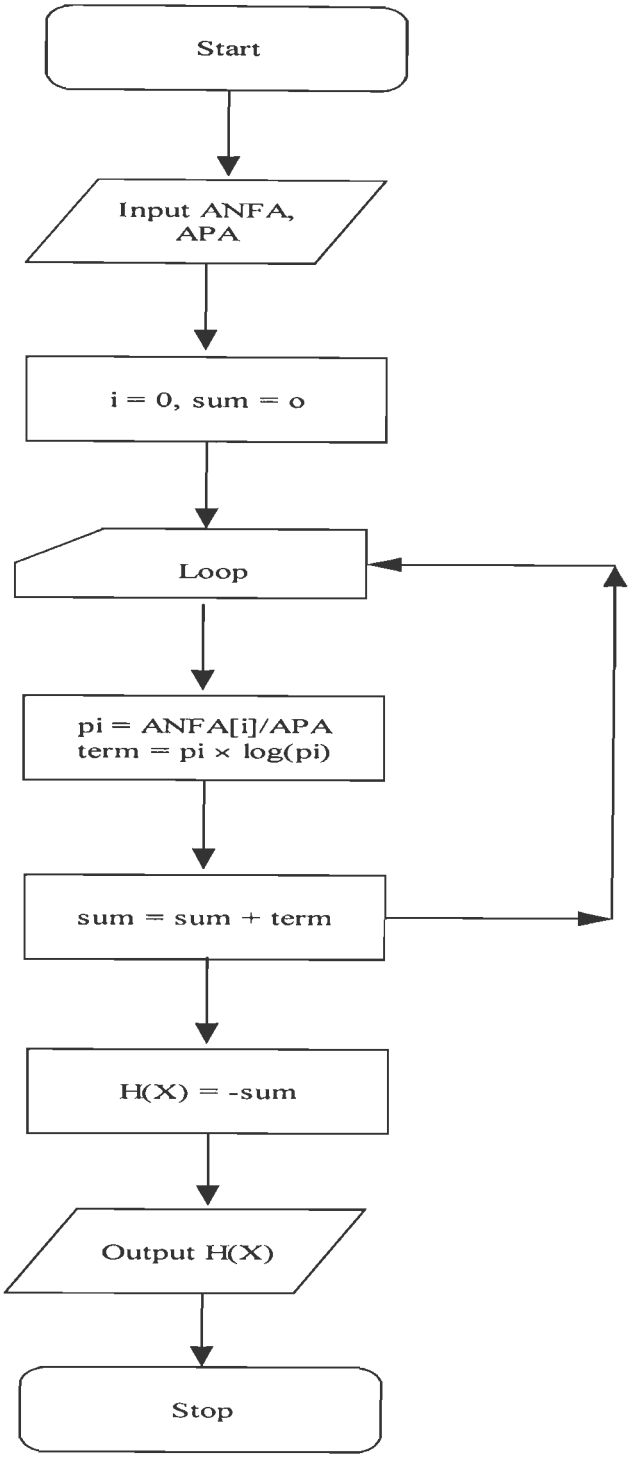

Figure 3.6: Flowchart for computation of sample entropy

The negation of sum indicates conformance with equation (3.1) for calculation of sample entropy $H(X)$ in current time window $\{t-\Delta,t\}$.

3.4.3 Detection of Attack

The observed time series [54] of sample entropy $H(X)$ without mixing attack traffic reveals that sample entropy distribution lack kurtosis [66] or have negative kurtosis. The frequency distribution of sample entropy has high hump in the middle. Quite evidently, sample entropy $H(X)$ varies within very narrow limits after slow start phase is over. The study on NZIX traces conducted in [102] also reveals the same facts. The variation in sample entropy becomes narrower if we increase monitoring period Δ. We take average of $H(X)$ and consider that as normal Entropy $H_n(X)$. The basic idea to increase Δ is to remove small scale perturbations by averaging over slightly longer-intervals of time. At the same time, it is also desirable that Δ should not exceed a limit as Internet traffic shows large variations across different times of the day. The normal profile of traffic is summarized with computation of average sample entropy $H_n(X)$ from observed time series of sample entropy $H(X)$. The detection of attack in real time is realized with continuously computing sample entropy $H_c(X)$ called current sample entropy at time $\{t-\Delta,t\}$ where Δ is a very short time window. Deviation in sample entropy $(H_c - H_n)$ more than a predefined threshold signals a flooding DDoS attack where as sign of deviation indicates LRFD or HRFD attack.

D-DCFI assumes that the system is under attack at time t_a. It means attack sources start emitting packets at time t_a. The network is in normal state for time $t < t_a$ and turns into attacked state at time t_a. Let t_d denotes estimate of t_a.

At time t_d following event triggers in D-DCFI

$$(H_c(X) > (H_n(X) + a \times d))$$
$$OR$$
$$(H_c(X) < (H_n(X) - a \times d)) \tag{3.2}$$

where $a \in I$, I is set of integers, and d is absolute value of standard deviation in observed time series of sample entropy $H(X)$ without attack.

Tolerance factor a is a tunable parameter. The value of a is chosen after conducting simulations at different attack strengths. The tradeoff between detection rate and false positive rate using ROC curves provides guidelines for deciding value of a for a particular network environment and hence help in setting thresholds. The performance of the detection method in D-DCFI is studied with variation of tunable parameter a in results and discussion section. Hence, the anomaly based detection approach in D-DCFI has reasonable foundation to adapt according to different environments and traffic conditions. The flowchart for detection of flooding attack is given in Figure 3.7.

High value of current sample entropy $H_c(X)$ gives sign of a LRFD attack or start of HRFD. LRFD attacks are carried with large number of zombies. However, a few numbers of packets per flow are flooded to the protected server as zombies use random packet header fields in their packets. Accordingly, sample entropy rises because of increased dispersion in $X(t)$ at time t_d. Distribution $X(t)$ is also negatively skewed at time t_d. HRFD attacks normally start with a few number of packets per flow, but with time the intensity of attack increases. At this time, the service of protected server definitely starts getting degraded but is not completely disrupted. Thus, the rise in value of sample entropy gives an early indication of HRFD attack. Moreover, it is worth mentioning here that even though in

LRFD attack, the volume per flow is less, but aggregate volume may be large enough to completely cripple down the protected server.

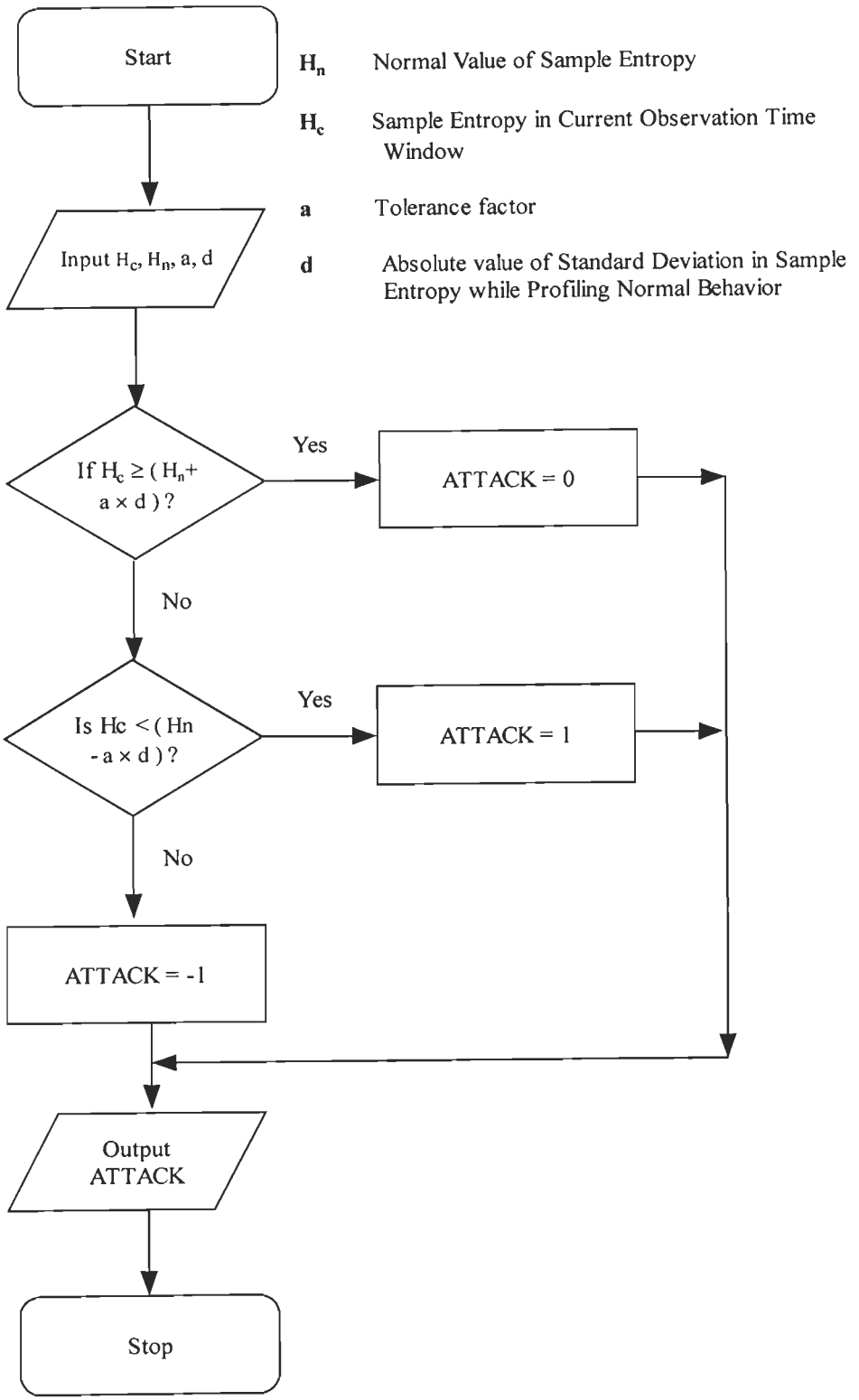

Figure 3.7: Flowchart for detection of flooding DDoS attack

Also highly distributive nature of DDoS traffic increases dispersion resulting in detection of attack.

Downfall in $H_c(X)$ value however, alarms a HRFD attack. Distribution $X(t)$ is positively skewed to a few high frequency values. Clearly, dispersion tends to decrease and hence, a sudden fall in sample entropy $H_c(X)$ is observed.

The average entropy $H_n(X)$ can also be computed using an exponential weighted moving average (EWMA) [66] of previous measurements as given below:

$$H_n(X) = \beta H_{n-1}(X) + (1-\beta)H_c(X) \tag{3.3}$$

where β is the EWMA factor and $H_{n-1}(X)$ is average value computed in previous time window. The objective here is to adapt in accordance with trends and periodic behavior of normal traffic i.e. the load is much higher in peak hours as compared to off-peak hours. In fact, in actual run of the system, adaptive threshold based scheme can be used for computing $H_n(X)$.

The complete system will be operable in different traffic conditions without choosing fixed value of thresholds. Even in case of profiling if ROC curves give high false alarm rate at required detection accuracy or vice versa, a simple modification of our detection model given in equation (3.2) can be made. The modification includes another tunable parameter b, where b is minimum number of consecutive event triggers required to alarm an attack.

The flowchart for detection given in Figure 3.7 is quite straight forward. Computed values of H_n, a, and d are used as input and H_c taken from previous step is compared as per

equation (3.2). The variable ATTACK =0 || 1 indicates LRFD or HRFD attack respectively whereas, ATTACK=-1 signs normal activity.

3.4.4 Characterization of Attack Traffic

The aim of DDoS defense is to maximize number of legitimate packets reaching at the server. To achieve the same, it is necessary to distinguish the traffic coming from attacking sources. In detection phase, if $H_c(X)$ value is more than threshold value, then suspected malicious flows tend to have lower frequency values of packet arrivals. The attack is termed as LRFD attack. A lower value of $H_c(X)$ than threshold hints that suspected malicious flows have high values of number of packet arrivals. The attack is called HRFD.

At any time $t > t_a$ in observation window $\{t-\Delta, t\}$, the traffic in ISP domain consists of both legitimate and attack flows. Let F represent set of active flows in ISP domain 4.

$$F = F_n \cup F_a \quad (F_n \cap F_a = \phi) \tag{3.4}$$

In equation (3.4), set F_n represents normal or legitimate flow IDs and F_a is set of attack flow IDs. The main task of this module is to find $F_a^* = \{f_1, f_2, \ldots\ldots f_m\} \subset F$ i.e., the set of m malicious flows. Skewness observed in packet arrivals distribution $X(t)$ in time window $\{t-\Delta, t_d\}$, decides computation of flows in F_a^*. As for as low rate attacks are concerned, m number of least measured packet arrival flows constitute F_a^* and for high rate attacks, m number of highest measured packet arrival flows form F_a^*. Ideally

$$(F_a^* \cap F_a = F_a) \quad AND \quad (F_a^* \cap F_n = \phi) \tag{5}$$

The collateral damage caused by any DDoS response module depends on accurate identification of attack flows F_a^*. The severity of collateral damage can be categorized into three cases as detailed next.

If $(F_a^* \cap F_a \neq F_a)$

{

if $(F_a^* = F_a \cup F_x)$ *where* $F_x \subset F_n$

{

few normal flows are characterized as attack flows.

}

else if $(F_a^* = F_s)$ *where* $F_s \subset F_a$

{

all the actual attack flows are not characterized as attack flows

}

else if $(F_a^* = F_s \cup F_x)$

{

few normal flows are classified as attack and all attacks flows are not identified

}

}

The actual collateral damage depends upon number of normal flows classified to attack flows i.e. F_x. Characterization of attack flows in D-DCFI depends on finding correct value of m, as either low or high frequency flows constitute attack flows. An estimate of total attack traffic Φ_a is used to compute m and F_a^*. The expected value of attack traffic is computed as $\Phi_a = \Phi_{t_d} - \Phi_n$ where Φ_{t_d} is the total traffic received in $\{t_d - \Delta, t_d\}$ and Φ_n is averaged total traffic. The value Φ_n is calculated by averaging total traffic observed from the time bottleneck link utilization is 1 up to time $t = t_d - \Delta$. The condition in equation (3.6) is used to find flows responsible for asymmetrical skewness in packet arrival distribution i.e. malicious flows.

$$\sum_{j=1}^{m} X_i^j(t_d + \Delta) \leq \Phi_a \tag{3.6}$$

Distribution $X(t_d+\Delta)$ represents packet arrivals for flows in next time window after attack is detected, i is designated flow, and j varies from 1 to m. The condition given in equation (3.6) helps characterization module to segregate m flows, which have either least or highest packet arrivals. The m flows having least packet arrivals are put in set F_a^* for low rate DDoS attacks as these flows cause current sample entropy $H_c(X)$ to increase. Set F_a^* constitutes m flows having highest packet arrivals in case of high rate DDoS attacks as these flows decreases $H_c(X)$.

The flowchart in Figure 3.8, details characterization of attack flows. The variable ATTACK and the array ANFA generated from detection module together with values of Φ_{t_d} and Φ_n are used to determine m number of malicious flows. After sorting ANFA in ascending for low rate and descending for high rate attacks, equation (3.6) is used to find F_a^* by employing a loop with terminating condition $s < \phi_a$. The set F_a^* is further pruned by removing flows from F, which have been active at time $t_d - \Delta$.

There are normal flows that also contribute to asymmetry in current sample entropy $H_c(X)$. For example, when $H_c(X)$ is high as compared to expected then those legal flows which have suffered drops or just have started will also have lower rate than other normal flows which have not suffered drops. Also when $H_c(X)$ is too low then some valid flows may have higher number of packet arrivals due to traffic bursts. Therefore, there is probability of misclassification of flows.

To improve the accuracy of characterization, flows in F_a^* are further investigated to confirm their illegitimacy. So far most DDoS attacks we know never attempt to establish connections with the target. The legitimate clients can be identified by monitoring their

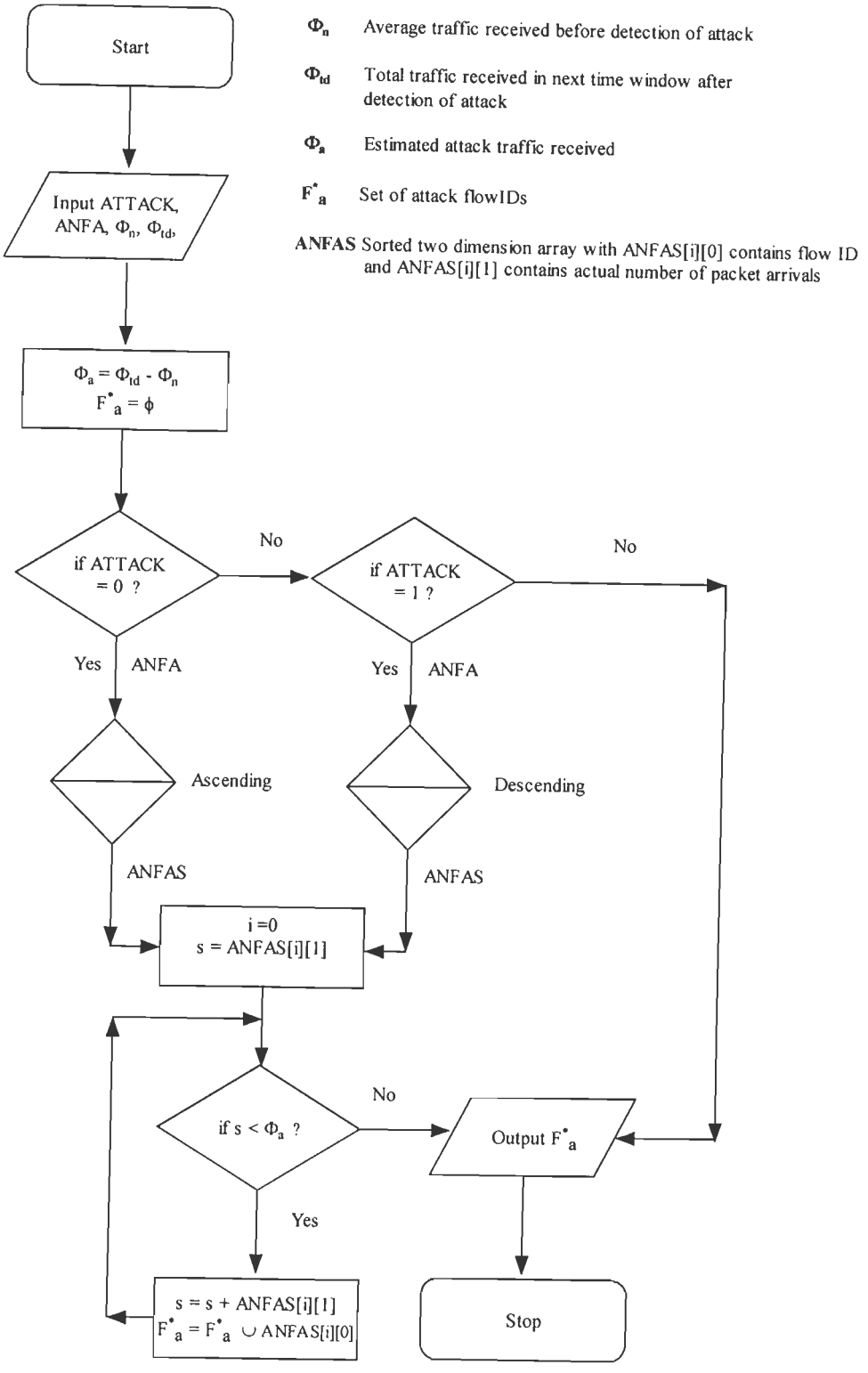

Figure 3.8: Flowchart for characterization of attack traffic

connection status with the server. Such status can be easily obtained from the server. However, as under attack, server is not able to furnish information so to make the system practical and automated, we simply look for ACK packets coming from the server at monitoring POP. TCP servers send ACK packets to only those clients that successfully completes three-way handshake as shown in Figure 3.9.

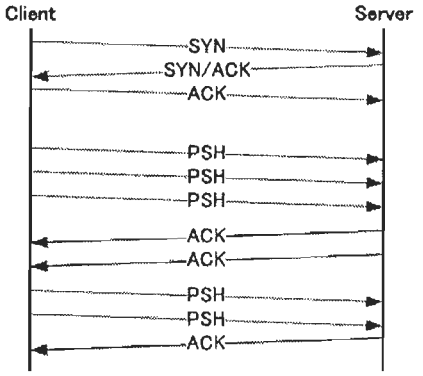

Figure 3.9: A typical TCP connection

ACKs coming from the server are monitored in time $\{t_d + \Delta, t_d + u\Delta\}$ where u is the least positive integer such that $(u\Delta - \Delta) > RTT_g$, where RTT_g represents maximum round trip time (RTT) calculated using packet arrival time of different flows. If destination addresses and port in every ACK coming from the server is found in any flow of F_a^* as source address and source port then that flow is deleted from F_a^*. The flows that use spoofed IP addresses are also trapped as they use IP address of inactive machines. The final ACK required for three-way handshake is not sent and so no connection is established.

Finally, a TCP server sends ACKs for only those flows that have established connections. Though currently most of ISP routers use Ingress and Egress [119] filtering, but still finding sources which use IP spoofing especially subnet spoofing is important in context of defending from DDoS attacks.

3.4.5 Suppression of Attack Traffic

The ideal response to DDoS attacks is filtering attack traffic close to their origin. The POP P_s in ISP 4 as shown in Figure 3.3 coordinates the filtering process. The suppression module is normally integrated with routing module as while routing packet header fields are detached. There are three points in the proposed topology, where attack traffic can be dropped:

1. POP P_s connected to the server
2. All POPs of the ISP domain 4
3. All POPs of other cooperative ISP domains

D-DCFI filters attack at second point. The disadvantage with the first point is increased load on POP P_s. Core bandwidth is also wasted to carry attack packets up to P_s. The filtering at third point logically seems to be the best, but cooperation between inter ISPs is still a big challenge [154]. The communication framework in [157] however helps to install filters at edge routers of cooperative ISPs. Moreover, secure communication framework [159] also makes communication framework hard enough to deny DDoS attacks on the framework itself. Figure 3.10 explains suppression of attack traffic. Set F_a^* is first put in payload of a multicast packet. It is sent to the registered multicast group comprising of all POPs of ISP 4. After communication delay, the packet reaches at all POPs where PF array is populated with F_a^*. Now Flow ID of each packet is scanned in the filter array PF and if flow ID exists then packet is dropped, otherwise it is enqued for the next hop. All suspicious packets are dropped at the boundary of ISP 4 and hence victim server is relieved from DDoS attack.

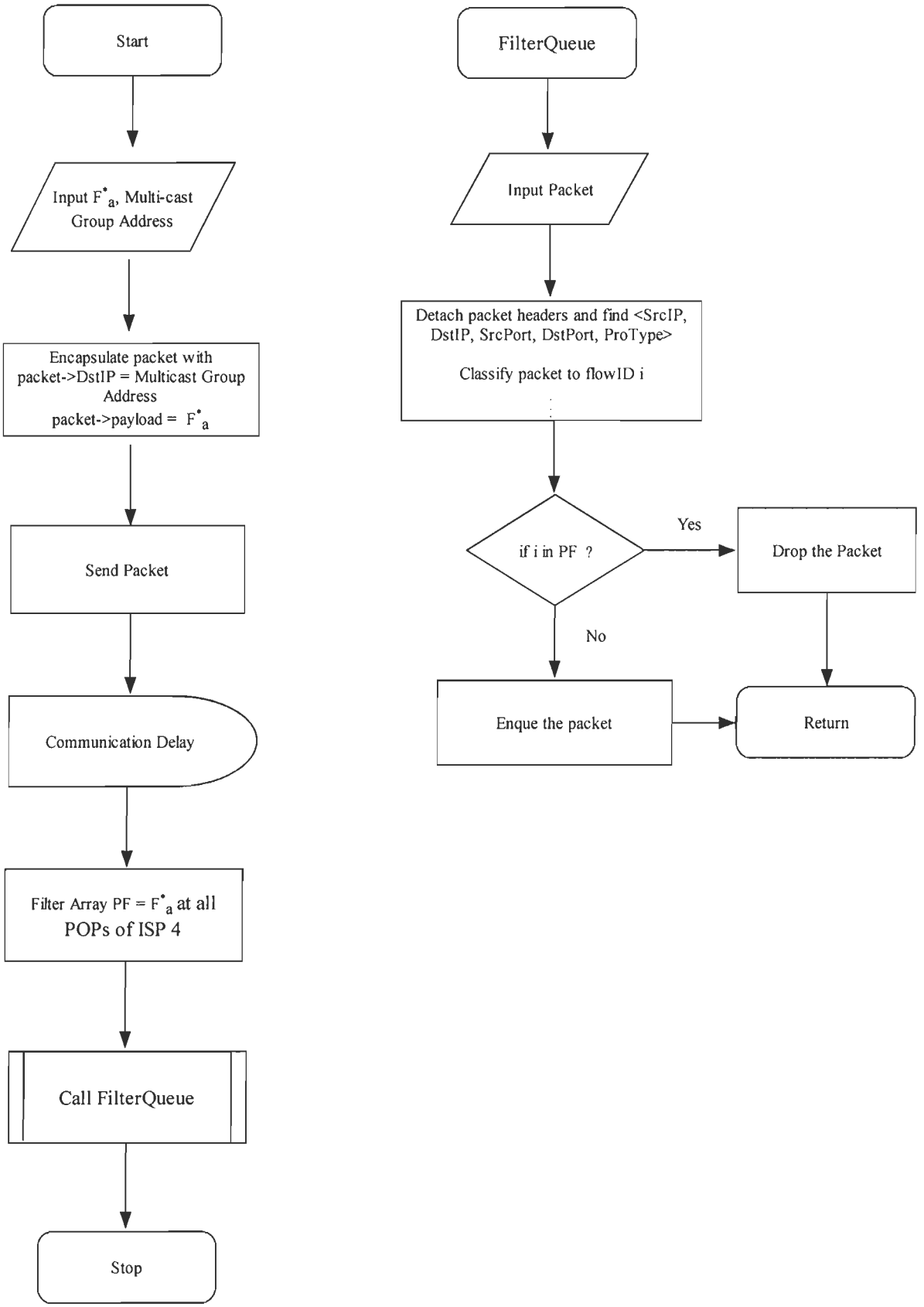

Figure 3.10: Flowchart for suppression of attack traffic

3.5 Design of Simulation Experiments

Simulation is performed using NS-2 [114] network simulator. NS-2 is a discrete event driven simulator used for wired cum wireless network research. NS-2 is an open source tool. The use of open source tools in networking laboratory is discussed in [56]. In fact NS-2 is used as testbed for validation and performance comparison of different approaches. The best part of NS is that it simulates a variety of IP networks. It has support for network protocols such as TCP and UDP, network traffic sources such as FTP, Telnet, CBR, and Ping, router queue management mechanisms such as Drop Tail, RED and CBQ, and routing algorithms such as DV, LS, AODV, and DSR. Moreover its two tier interface of OTcl and C++ give flexibility to work at designing and running simulations in Tcl using simulated objects in OTcl library whereas, C++ deals with hard core implementation of various protocols, event schedulers, agents and networking components. A C++ user in NS can modify and/or create protocols, agents, and nodes etc. as per requirements of proposed approach. A typical simulation scenario in NS consists of an OTcl script that initiates the events scheduler, sets up the topology, communicates traffic sources when to start and stop through the events scheduler, and instructs traffic sinks what to do with received packets. This OTcl script may contain modified or new objects created in C++ or may have calls to functions written in C++ using interface between OTcl and C++. We have used trace driven simulations. Various components and parameters of simulation are explained next.

3.5.1 Topology

Network topology often influences outcome of the simulation. Realistic topologies are needed to produce reliable simulation results. GT-ITM [65] topology generator is used to create topology. Transit-stub model of GT-ITM is applied to create an ISP level topology. The topology generated in the form of graph is later converted to Tcl format of NS-2. A

simplified view of topology is shown in Figure 3.3 (section 3.4). The parameters used in GT-ITM [65] to create topology are listed in table 3.3.

Table 3.3: Topology generator parameters

Parameter	Value
ISP domains	4
No. of transit routers	12 (1 more in ISP 4 for connecting servers)
Edge probability	0.85
Number of stub domains	10 per ISP
Number of hosts	10 per stub domain
Backbone link bandwidths	2.5GhZ
Backbone link delays	0 seconds

Four transit domains created are marked as ISP domains 1, 2, 3 and 4. Twelve transit domain nodes in each ISP are having a link to each other called edge with probability 0.85. All the four ISPs have two peer links at transit nodes with adjacent ISPs. The other ten transit nodes are connected to one stub domain node. Stub domain node is used here for connecting the ISP to customer domain.

Each combination of transit-stub node is represented as point of presence (POP) in the simplified topology shown in Figure 3.3. Stub node in all POPs is used for connecting customer domain whereas transit node is used for interconnection between POPs of the same and neighboring ISPs. In each customer domain there are 10 legal clients i.e. 100 legal clients per ISP domain. Twenty-five attackers are used to launch DDoS attacks in every ISP. A total of one hundred attackers flood DDoS traffic from all four ISPs. The backbone is consisting of POPs interconnections each having bandwidth of 2.5Gbps. ISP 4 contains protected server connected to additional POP P_s.

3.5.2 Basic Parameters of Simulation

Simulation time is taken as 60 seconds. DDoS attack is launched from 20 to 50 seconds. However for pulsing attacks, attack periods are 20-25, 30-35, and 40-45 seconds. Table 3.4 provides the basic parameters used in simulation.

Table 3.4: Basic parameters of simulation

Parameter	Value
Simulation Time	60 seconds
Access bandwidth	1Mbps
Bottleneck Bandwidth	310Mbps
Mean attack rate per attack host	0.1 to 1.0Mbps (LRFD) 2.7 to 3.7Mbps (HRFD)
Attack period	20-50 seconds

Legitimate hosts have access bandwidth of 1Mbps. Four hundred legitimate hosts generate enough traffic to saturate the bottleneck bandwidth of 310Mbps between server and POP P_s. The server as per its capacity planning has an estimate of maximum number of clients to be served at any instant of time. On this basis, four hundred legitimate clients are used to generate traffic so as to saturate bottleneck of 310Mbps for better utilization of link all the time. Once slow start phase is over, bottleneck link utilization is always one without generating the attack. The total legitimate traffic generated after slow start phase is 320Mbps (approx.). Legitimate packets are dropped even without attack, so congestion cannot be taken as attack signature. Attack strength is varied in the range (0.1-1.0Mbps) per attack host for LRFD and (2.7-3.7Mbps) per attack host for HRFD attacks. As number of attack hosts are one hundred so some of LRFD attacks have strength even less than five percent of legitimate traffic. Some of generated HRFD attacks have strength more than hundred percent. Here attack hosts mean zombies used to launch the attack. In this way, a

perfect attack scenario is generated for launching low and high rate flooding DDoS attacks against the server.

3.5.3 Traffic Parameters

Finalizing packet arrival process at legitimate clients is itself a topic of research. Cao et al. [80] at bell labs concluded in their research that Internet traffic tends toward Poisson distribution as the load increases. An extensive empirical and theoretical study of packet traffic variables: arrivals, sizes, and packet counts, demonstrates that the number of active connections has a dramatic effect on traffic characteristics. At low connection loads on un-congested link i.e. with little or no queuing on bottleneck link, the traffic variables are long-range dependent. As traffic load increases, the laws of superposition of marked point processes push the arrivals toward Poisson, the sizes toward independence, and reduce the variability of the counts relative to the mean. Poisson distribution is used in simulation for packet arrival process as shown in table 3.5.

Table 3.5: Traffic parameters

Parameter	Value
Traffic arrival process at legitimate clients	Poisson
Traffic generation at attackers	Mean attack rate given in table 3.4
Connection startup time	Random 1-8seconds
Packet Size	1040bytes

LRFD and HRFD attacks are generated with the help of UDP traffic. The UDP traffic in NS-2 is used to generate attack traffic at different rates. The generated file contains inter arrival time between packets with size of the packet. Size is kept same as that of legitimate packet i.e. 1024 bytes. These files are attached in traffic generation module used for conducting simulations in NS-2. UDP traffic is used for attack generation as like DDoS

attack traffic, it does not follow congestion and flow control signals [140]. Moreover, all the legitimate TCP connections are not initiated at the same time as SYN backlog is limited in size as shown in table 3.5. The legal TCP connections are initiated after 20 seconds so as to check whether proposed method picks them as legal or not.

3.5.4 Attack Detection Parameters

Window size Δ is an important parameter that determines smoothening of short term fluctuations in legitimate traffic. A high value of Δ means lesser false positive rate, but more detection time, whereas low value of Δ may falsely detect legitimate fluctuations as attack i.e., more false positives but, less detection time. Similarly in terms of false negative rate, higher values of Δ may generate more false negatives as a very short duration attack may not change traffic metrics to cross thresholds. In addition, round trip time also affects proper selection of parameter Δ. As a TCP packet drop induces decrease in packet rate, the effect of a TCP packet drop should not be reflected in the same time window Δ otherwise true picture of network traffic is not reflected. Thus, Δ should be more than maximum round trip time (RTT) of any TCP flow. Moreover, the impact of multicast communication between detection node and filtering nodes should not be shown in same windows otherwise the cause and effect relation will not be properly reflected. D-DCFI has chosen value of Δ after consideration all these factors. Table 3.6 shows finalized detection parameters.

Table 3.6: Attack detection parameters

Parameter	Value
Time window Δ	0.2 seconds
Tolerance factor a for sample entropy deviation	6-7

Tolerance factor a is computed by using tradeoff between false positives and true positives by drawing ROC curves. The ROC curve gives optimum value of both false positives and

true positives in interval $a = (6-7)$ for LRFD attacks. Simulations are conducted at various attack strengths to finalize the tolerance factor a as shown in table 3.6. The detailed explanation of ROC curve is given in results and discussion section.

3.6 Results and Discussion

The proposed method is implemented in NS-2 testbed using simulation parameters discussed in the previous section. The results are categorized as follows:

- Detection of attack
- Setting thresholds

3.6.1 Detection of Attack

The real time anomaly based detection scheme of D-DCFI comprises of two steps. The first step is profiling for normal behavior and the second step is raising detection alarms when current profile deviates from normal behavior more than a threshold. Sample entropy $H_n(X)$ of traffic flows characterized by flow IDs summarizes normal behavior and time series of sample entropy $H_c(X)$ represents current profile of traffic in ISP 4. Deviation $H_c(X) - H_n(X)$ at time t_d exceeding $a \times d$ raises a flooding DDoS attack alarm as per equation (3.2). Simulation experiments are conducted for finding $H_n(X)$ i.e. normal sample entropy without mixing attack with legitimate traffic. The time series of sample entropy $H(X)$ with $\Delta = 0.2$ seconds is shown in Figure 3.11.

The sample entropy $H(X)$ is computed in intervals of Δ time by using equation (3.1) where $X(t)$ is represented in table 3.2. It is quite evident from the graph in Figure 3.11 that the values of $H(X)$ lies in a narrow range. Statisticians describe this phenomenon with Kurtosis [66]. The sample entropy distribution lack Kurtosis or have negative Kurtosis i.e., the middle part of distribution will have high hump. Simulation is also carried by taking

longer time window $\Delta = 1.0$ seconds. The sample entropy range is still narrower as expected however average is almost same.

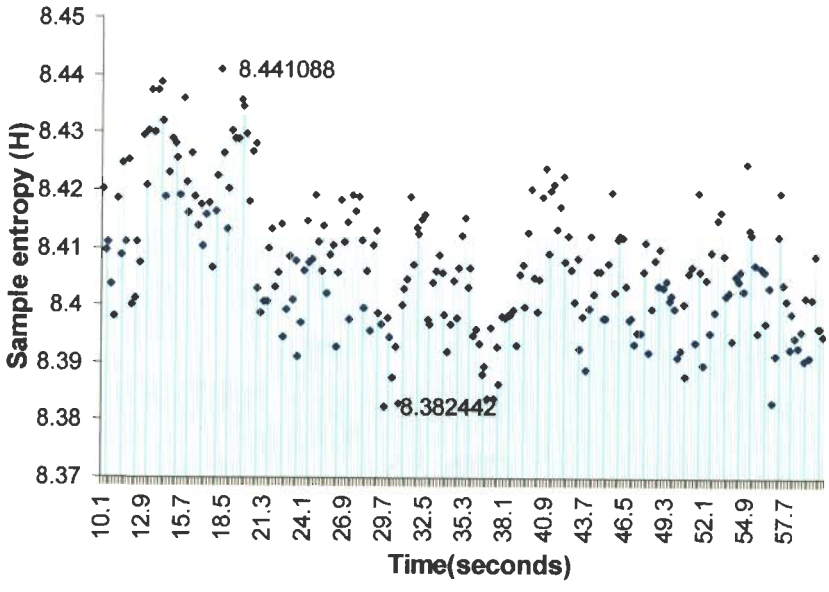

Figure 3.11: Temporal variation of sample entropy without attack

Simulation starts at time 1.5 seconds. The normal TCP traffic first goes through the slow start phase [140], and with time TCP traffic grows. The TCP traffic coming from various ISP domains aggregates at bottleneck link. The bottleneck link depicted in Figure 3.3 has bandwidth of 310Mbps. It is found in Figure 3.12 that the utilization of bottleneck link is less than 100% when most of flows are in slow start phase.

But once utilization become 100% as shown in Figure 3.12, the sample entropy $H(X)$ value also lies in small range as depicted in Figure 3.11. `The $H(X)$ values are shown from 10.1 seconds onwards and at this time bottleneck utilization is 100%. It is found that the range of $H(X)$ without attack is 8.382442 to 8.441088, whereas this may vary depending upon network environment and type of applications running in ISP e.g. in NZIX traces the range was 7.0 to 7.5. The normal value of sample entropy $H_n(X)$ is the average of $H(X)$.

The value of $H_n(X)$ is 8.407158. Standard deviation d is 0.012, and maximum absolute deviation from average is 0.03393.

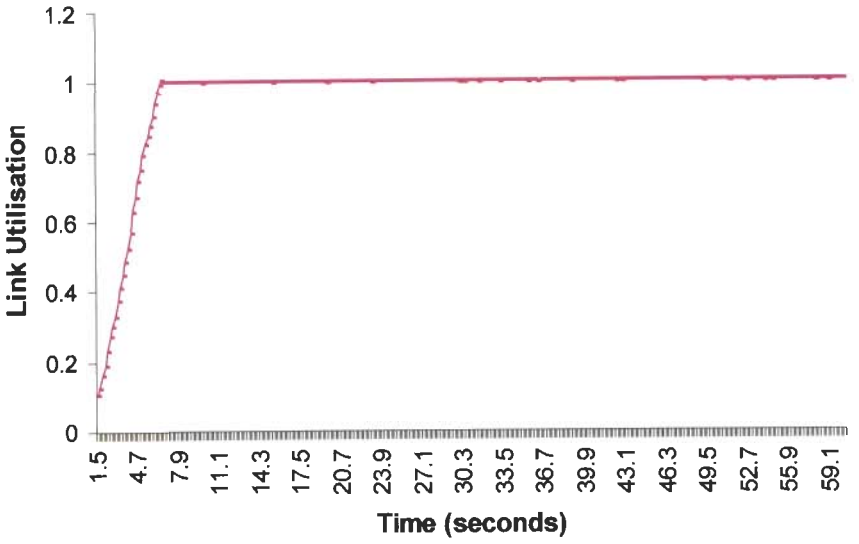

Figure 3.12: Link utilization

Finalized simulation parameters are:

Normal Entropy Value ($H_n(X)$): 8.407158

Standard deviation (d): 0.012

Tolerance factor (a) is computed using ROC curves, which is explained after detection of attack.

Legitimate traffic is severely affected at maximum load for even very low rate attacks. Traffic congestion level does not indicate that the network is under attack, but D-DCFI generates attack alarms as dispersion in traffic distributions exceeds set thresholds. Though the work is supported using simulation, but in actual network conditions the same kind of experiments can be performed to profile normal behaviour of the system by computing

$H_n(X)$ and d. Moreover, to adapt as per actual traffic load conditions of the ISP, equation (3.3) can be used to compute $H_n(X)$.

The real time detection of flooding DDoS attacks start with time series monitoring of traffic at POP P_s of ISP 4. Time series monitoring starts at 1.5 seconds. Frequency distribution of packet arrivals $X(t)$ per flow is computed at P_s in intervals of $\Delta = 0.2$ seconds. Sample entropy $H_c(X)$ is computed by using equation (3.1). The process of detecting flooding DDoS attacks starts with triggering of any event in equation (3.2). Attack is launched at 20th second and lasts till 50th second. Figure 3.13 shows sample entropy $H_c(X)$ profile when LRFD attack is launched. LRFD attack is launched using 100 attackers with mean rate 0.3Mbps per attacker.

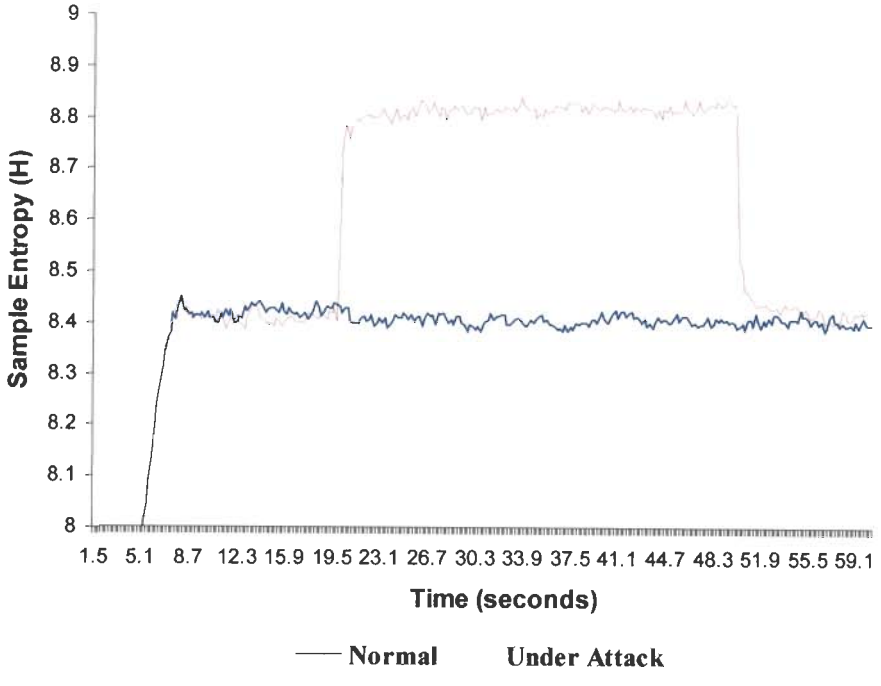

Figure 3.13: Sample entropy for low rate DDoS (LRFD) attack

Though legitimate TCP users have access bandwidth of 1Mbps, it is observed that average legitimate traffic rate per flow is 0.8Mbps. The main cause behind it is limited bottleneck

capacity, as all 400 TCP flows are not able to transmit at 1Mbps. As per Figure 3.13, in first time window after attack is launched at 20 seconds, there is a jump in sample entropy value. The positive jump and persistent high value as compared to normal reflects that it is a low rate attack (LRFD) and the flows, which are causing this anomaly have comparatively lesser frequency than already existing legitimate flows. It is found that frequency distribution of packet arrivals $X(t)$ per flow consists of new flows with packet arrivals lesser than already existing flows. Actually, $X(t)$ get negatively skewed due to lesser number of packets contributed by attack flows. Further, due to arrival of low rate flows, the dispersion of $X(t)$ measured as $H_c(X)$ increases. It is highlighted here that even some legitimate flows, which have just started or are in slow start phase also contribute to increase in $H_c(X)$ value. Overall, a large number of new flows have lesser p_i values due to low rate attacks. Thus, $H_c(X)$ increases more than threshold as per equation (3.1). Even attackers which do IP spoofing and uses random packet header fields contribute to more number of flows with lesser p_i and add to the value of $H_c(X)$.

We repeated low rate attacks in the range from 0.1 to 1.0Mbps (mean rate) per attacker using 100 attackers. In all cases the trend was similar as shown in Figure 3.14, though deviation $H_c(X) - H_n(X)$ is different. As the attack rate is very low, traditional volume based techniques [17, 18, 79, 104, 110, 116, 124, 134, 153] are not able to distinguish between attack and normal condition as the traffic load does not increase much. However, Figure 3.14 clearly indicates the deviation in sample entropy, which justifies our claim of picking even very meek rate attacks. The main reason behind it is that D-DCFI relies on dispersion rather than volume to detect flooding DDoS attacks. Though total volume contributed by low rate attacks is low, but contribution of low rate attack flows towards

increasing dispersion of $X(t)$ is more. Hence, the value of $H_c(X)$ is more than normal as shown in Figure 3.14.

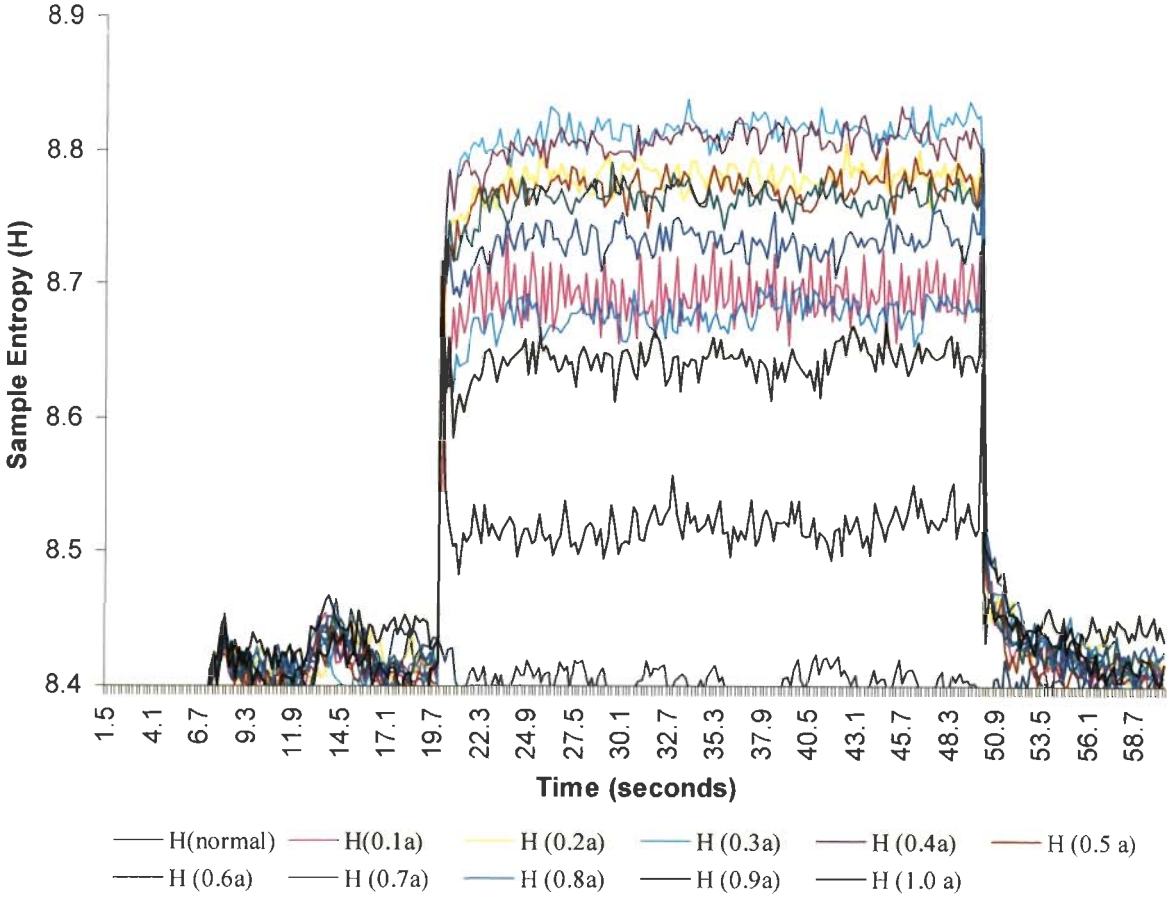

Figure 3.14: Sample entropy at variable attack strengths

In another set of simulation experiments, we launched HRFD attacks with mean attack rate in the range from 2.7 Mbps to 3.7 Mbps per attacker using 100 attackers. As the attack traffic aggregates at bottleneck link, legitimate packets drop at faster rate. The distribution $X(t)$ is dominated by new aggressive flows. In effect sample entropy $H_c(X)$ tends to be lower than normal $H_n(X)$. Figure 3.15 shows comparison of $H_c(X)$ under attack strength of 300Mbps (3Mbps per attacker using 100 attackers) with $H(X)$ without attack. However,

$H_c(X)$ do rise initially as shown in Figure 3.16. At 20.1 seconds, the number of packets reached at POP P_s is very less, so there is rise in sample entropy $H_c(X)$.

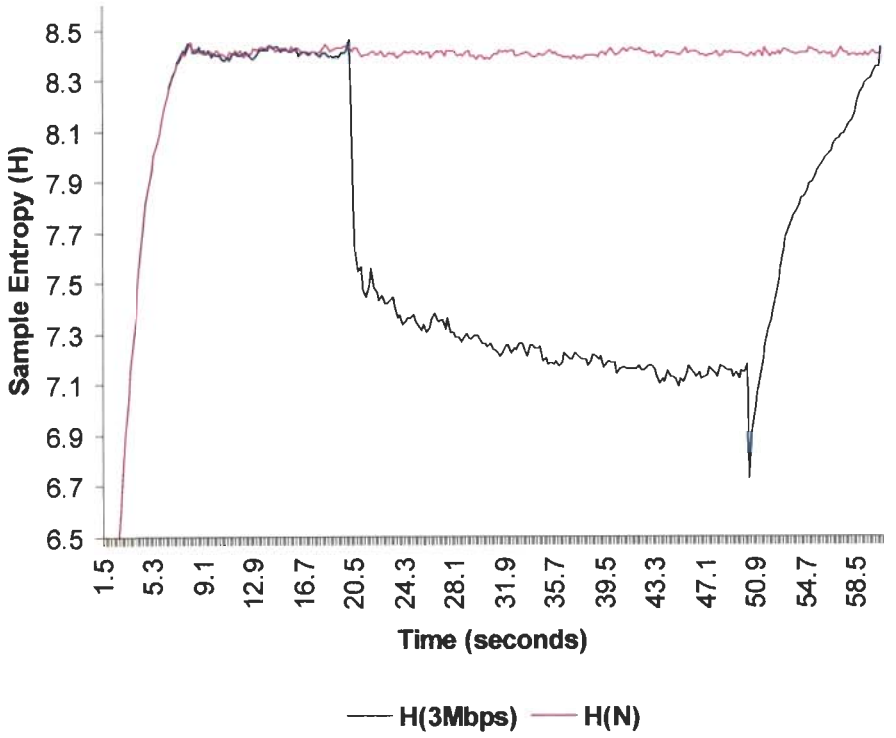

Figure 3.15: Sample entropy for high rate DDoS (HRFD) attack

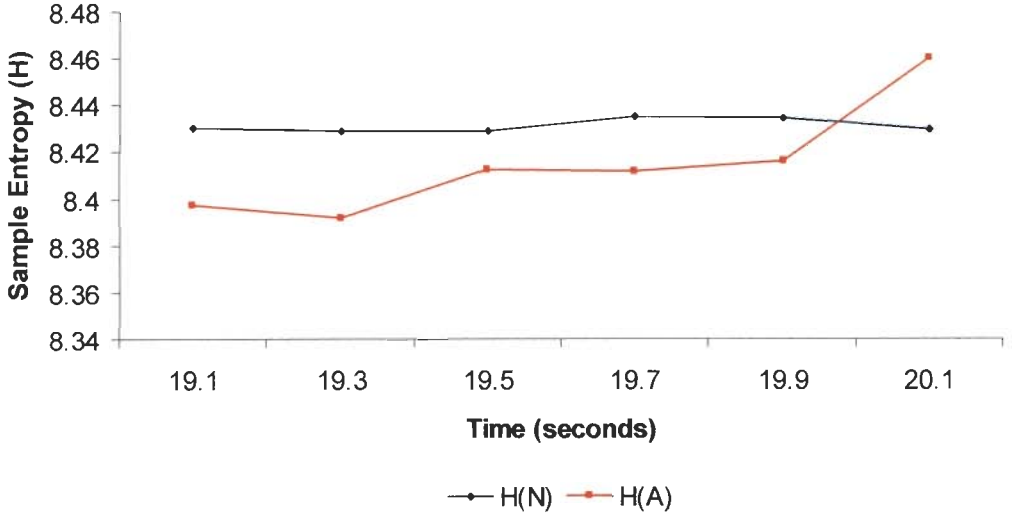

Figure 3.16: Initial rise in sample entropy under high rate DDoS (HRFD) attack

Clearly at the onset of even a high rate flooding (HRFD) attacks $H_c(X)$ rises giving a proactive signal of highly aggressive attack. In this case, the flows that have comparatively higher share of packets are reasons of anomaly. Similar trends exist for high rate attacks at different attack strengths. The only difference is in deviation from normal value.

The detection delay i.e. $t_d - t_a$ is 0.3 seconds in LRFD attacks, since we look for persistent rise in entropy, whereas it is 0.5 seconds for HRFD attacks as at 20.1 seconds, it first rises and after that it persistently drops. The persistent here means that any event in equation (3.2) triggers at least two times continuously. However, it is also a tunable parameter which depends on network environment. In current simulation scenario, for LRFD attacks, same event in detection model given in equation (3.2) has triggered more than once continuously for all cases. It is worth mentioning here that detection delay also depends upon simulation parameters such as size of time window, start time of simulation, and attack generation tool characteristics.

3.6.2 Setting Thresholds

In the last sub-section, $H_n(X)$, and d are computed after carrying out simulations without attack. The tolerance factor a that determines how much deviation $H_c(X) - H_n(X)$ is appropriate to alarm a flooding DDoS, is decided in this sub-section. As per equation (3.2), $a \times d$ value above or below $H_n(X)$ actually arbitrates an attack. The choice of $a \times d$ is however, a keen-witted task. A particular value of $a \times d$ may not be suitable for all network environments. Hence, for a particular network environment, a careful investigation of performance measures is done to find an optimal value of $a \times d$. The performance measures considered in this work are detection rate R_d and false positive rate R_{fp}. Detection rate R_d can be defined as below:

$$R_d = p / At_n \tag{3.7}$$

where p is the number of detected attacks and At_n is total number of actual attacks. The formal definition of false positive rate R_{fp} is given as:

$$R_{fp} = f / Nt_n \tag{3.8}$$

where f is total number of attack alarms when actually there were no attack i.e. false positive alarms and Nt_n gives total number of normal traffic events. In $a \times d$, value d is constant for a network environment. Therefore, the only tunable parameter is a. The optimal value of a for low rate attacks is found by observing simulation statistics at attack strengths ranging from 10Mbps to 100Mbps. The total number of sample points in time series of H_c for every run of simulation from time 1.5 seconds to 60.3 seconds with $\Delta = 0.2$ seconds were 295. In the simulation scenario, we have not mixed variable attacks with normal traffic. However, each simulation run consists of legitimate traffic and attack traffic at a particular attack strength as described before. As attacks were launched from 20.0 seconds to 50.0 seconds, the number of attack time windows was 150. The number of normal windows considered was 117 (from 7.1 seconds to 19.9 seconds and 50.1 second to 60.3 seconds). The value of a is varied from 1 to 10. Table 3.7 is prepared for different attack strengths ranging from 10.0Mbps to 100Mbps.

Table 3.7: Attack strength

Tolerance factor a	$H_n(X) + a \times d$	Number of detected attacks p	Detection Rate $R_d = p / At_n$ where $At_n = 150$	Number of false alarms f	False positive rate $R_{fp} = f / Nt_n$ where $Nt_n = 117$

R_d and R_{fp} calculated as per above table for different attack strengths for every value of a is averaged. It was observed that as a increases (i.e., $H_n(X) + a \times d$ increases) detection rate R_d which was stable at 1 tends to decrease after $a > 5$ as depicted in Figure 3.17. But for $a < 6$, legitimate fluctuations in normal traffic are also signaled as attack i.e., false positives are more for initial values of a. False positive rate R_{fp} at $a = 1$ is around 49% as shown in Figure 3.17.

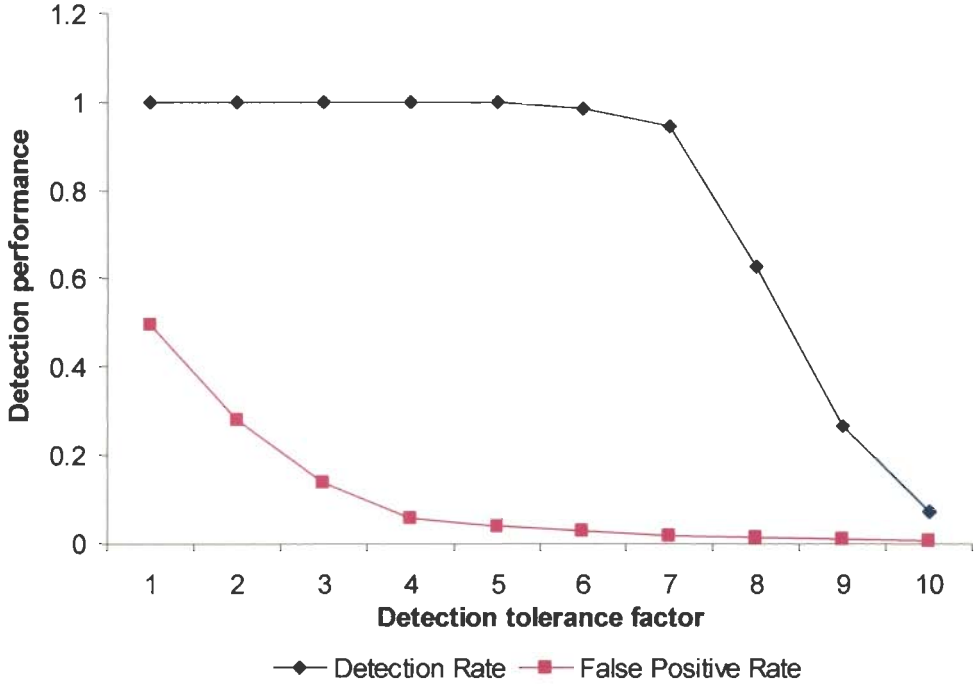

Figure 3.17: Effect of tolerance factor on detection accuracy

The increase in a factor decreases R_{fp} value. The threshold bar for event triggering in equation (3.2) gets advanced by a factor d which tends to decrease the chance of legitimate activity being termed as attack activity. The careful investigation of Figure 3.17 reveals that detection rate falls steeply after $a = 7$ whereas false positive rate becomes stable. At the same time decrease in false positive rate is faster before $a = 5$. So, to have an optimal value of a we need to settle two conflicting goals namely maximizing detection rate and

minimizing false positive rate. The Figure 3.18 called Receiver operating Characteristic curve shows tradeoff between detection rate and false positive rate. The high concentration of values towards lower right half clearly depicts the dominance of our approach. The lower right part of ROC actually gives the best region to operate the network.

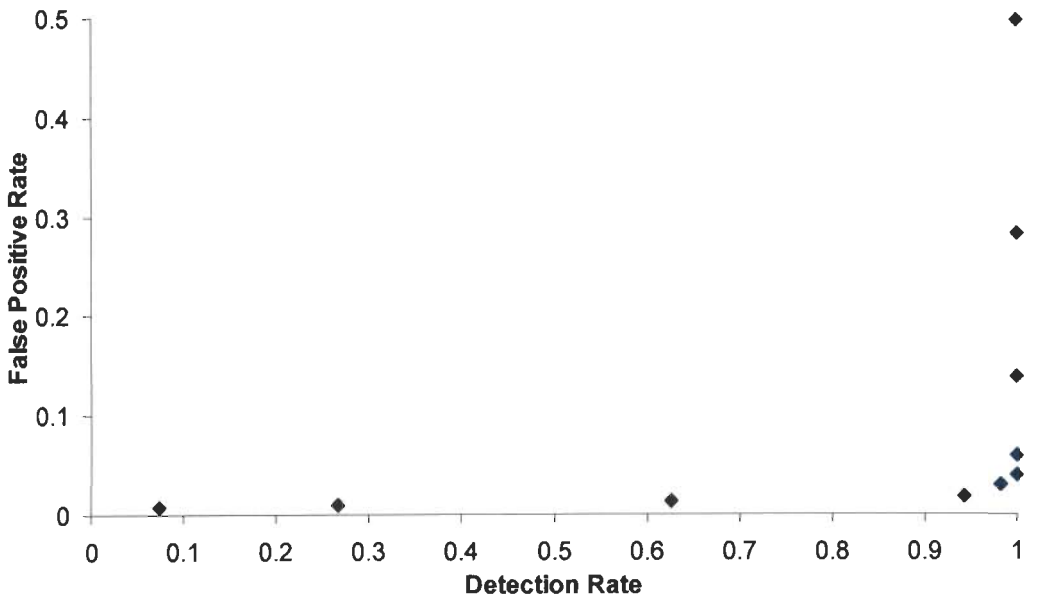

Figure 3.18: Receiver Operating Characteristic Curve (ROC)

As shown in Figure 3.18, when detection rate is 98% false positive rate is 2.93% and at 94% detection rate false positive rate is 1.85%, so the value of a is taken from 6 to 7 in D-DCFI. Accordingly, for any network environment by drawing ROC curves, one can reach at proper threshold values which generate optimum results for that particular environment.

3.7 Degradation of Goodput with Attack

The aim of any DDoS attack is to minimize legitimate traffic reaching at the server. Goodput is a measure of legitimate traffic reaching at server and is calculated as sum of bits received per flow at server of all normal flows per unit time i.e. $\sum F_n / \Delta$. Goodput at different attack rates are shown in Figure 3.19. LRFD attacks are conducted at strengths

ranging from 10Mbps to 100Mbps against protected server in ISP 4 connected to POP P_s. The bottleneck bandwidth is 310Mbps for all LRFD attacks.

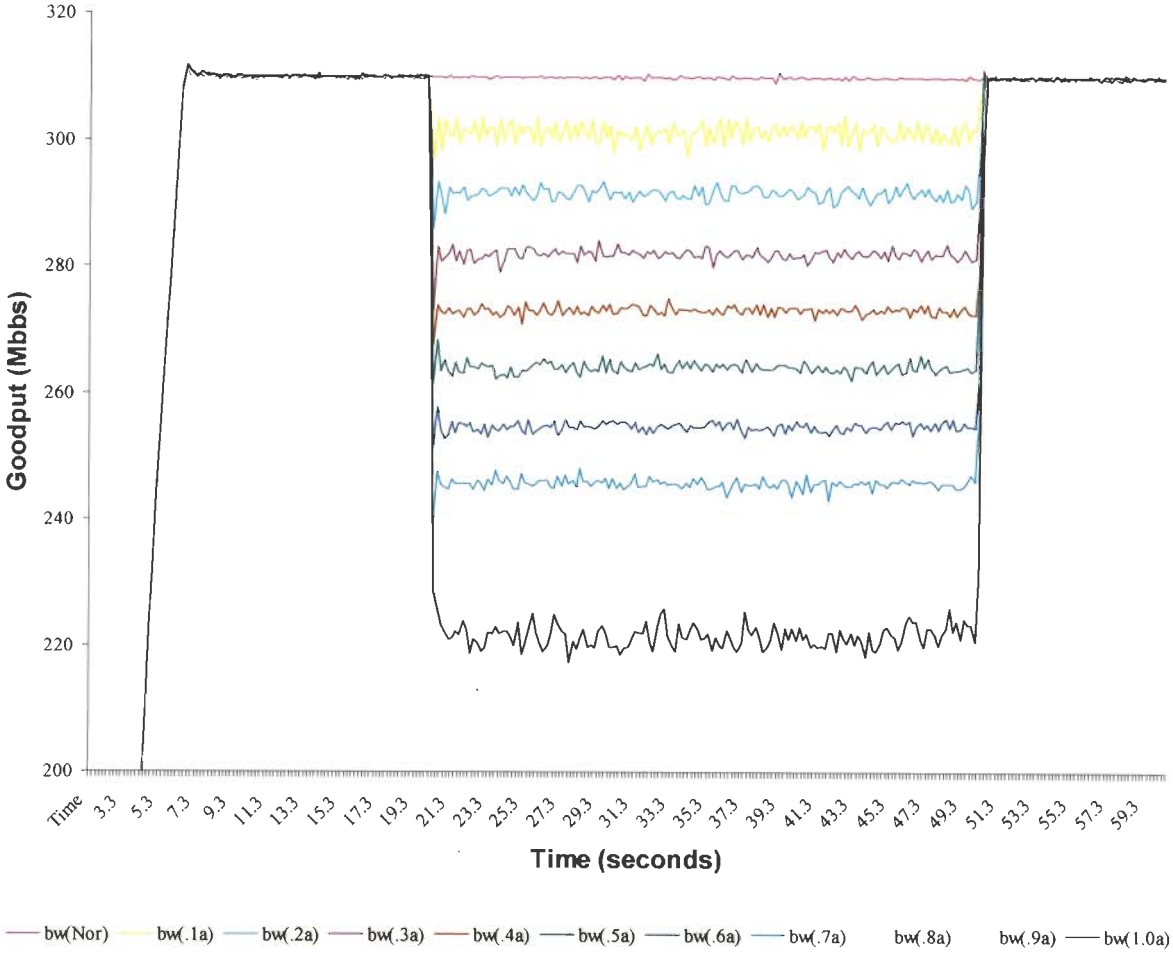

Figure 3.19: Relative degradation of goodput at different attack strengths

Attack duration is constant from 20 seconds to 50 seconds. As shown in Figure 3.19, for time less than 20 seconds the total traffic seen at bottleneck link is legitimate, hence goodput is same as the bottleneck bandwidth i.e. 310Mbps. In slow start phase bottleneck link is not fully utilized so goodput also grows, but once link utilization is 100% then goodput is 310Mbps. Once attack starts at 20 seconds, goodput decreases as attack traffic starts reaching at the bottleneck link between POP P_s and protected server. An analysis of simulation trace is conducted after each LHRD attack carried against protected server. First,

traffic between P_s and protected server is filtered. Next, all received traffic is filtered out. Thereafter TCP traffic is separated as UDP traffic has been used for launching attack. The final step is to sum bytes of all packets in the left trace based on some time granularity using a Perl Script [96]. Selected granularity value is 0.2 in D-DCFI. The results are depicted in Figure 3.19.

At 20.2, there is dip in legitimate traffic for all LHRD attacks carried at different attack strengths. Moreover, at meek attack rates, number of attack packet drops is almost negligible so they degrade to their full strength, however as attack strength increases number of legitimate as well as attack packet drops also increases. As for as high rate attacks are concerned, they almost bring the legitimate goodput to zero as is depicted in subsequent sub-section.

3.8 Comparison

The comprehensive evaluation of DDoS is still a pending issue due to unavailability of comprehensive test data, testing environments, and standards [60]. An ongoing effort by Mircovik et al. [92] is however a good step towards developing benchmarks for DDoS defense evaluation.

D-DCFI is compared with a traffic volume based approach [18] termed as VBA to counter DDoS attacks. VBA is implemented on our topology. It is found that for HRFD attacks, results are comparable. Though D-DCFI performs better, but in case of LRFD attacks, D-DCFI is far ahead and better than VBA. The detection modules of both the approaches are compared using ROC curves [13]. The response however, is compared by measuring NPSR at protected server. NPSR can be defined as $npsr = (L_n / (L_n + A_n))$

where L_n and A_n are number of legitimate and attack packets received in current time window $\{t-\Delta, t\}$.

VBA is based on measuring traffic levels at the bottleneck queue. Two events that mark the beginning of attack are as follows:

Event 1: The queue length exceeds the upper threshold L_1.

Event 2: $\lambda_c(t_d) > (1+r)\lambda_n$ where λ_c is aggregate traffic at the time of detection of attack and λ_n is already profiled normal traffic and r (tolerance factor) is a design parameter.

It is found that for LRFD attacks, the performance of VBA is not good. For LRFD attacks, simulations are carried at different attack strengths in the range (0.1 – 0.5) Mbps per attacker using 100 attackers. First λ_n, which is an average of arrived bits at the server, is calculated without attack and λ_c gives arrived bits in 0.2 seconds time windows. Traffic statistics for VBA are given in table 3.8.

Table 3.8: Values for volume based approach

Attack Strength/per attacker (Mbps)	λ_c **(Mbps)**	$\lambda_c - \lambda_n$ **(Mbps)** λ_n **=318.5Mbps**	r
0.1	327.3	8.8	0.02
0.2	339.2	20.7	0.06
0.3	348.7	30.2	0.09
0.4	356.5	38.0	0.11
0.5	368.4	49.9	0.15

Bencsath et al. [18] have reported that VBA is effective for $r > 0.6$. At $r = 0.2$, number of false detections is 1117 in VBA. This is mainly due to the fact that even in normal arrived traffic there are fluctuations and in case of very low rate degrading (LRFD) attacks, the total arrived traffic does not exceed even normal fluctuations.

VBA however, gives better results for HRFD attacks than LRFD attacks. The comparison with D-DCFI is made using ROC curves. ROC curve given in Figure 3.20 compares two approaches for HRFD attacks at different thresholds. Value r is tunable parameter in VBA whereas a is parameter of concern in D-DCFI. We conducted HRFD attacks at strengths in the range 270Mbps-370Mbps.

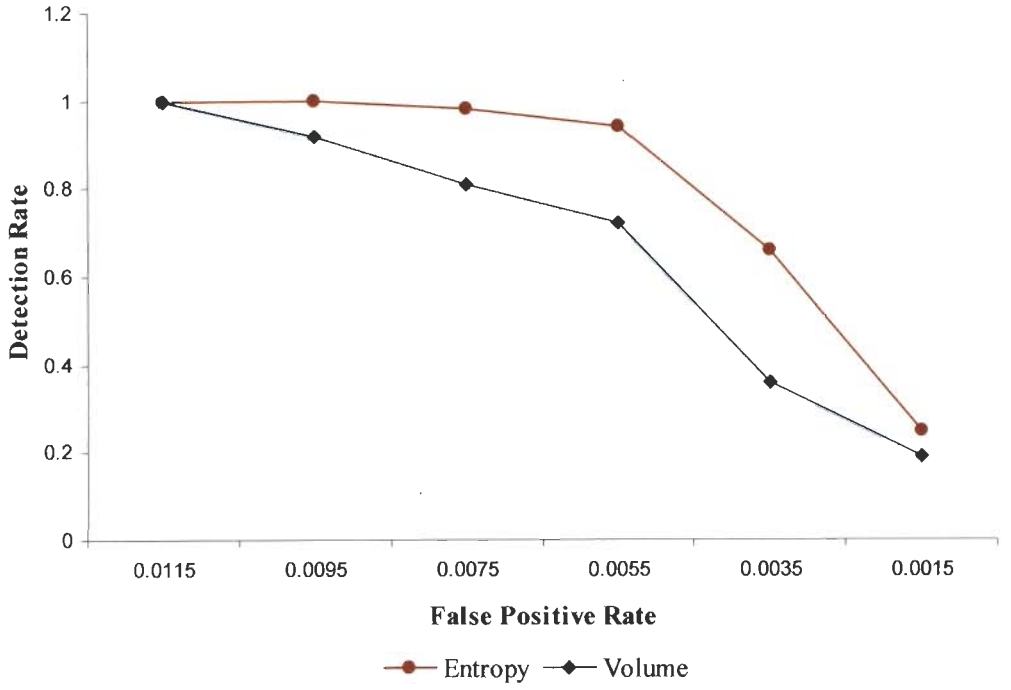

Figure 3.20: Comparative detection accuracy

Figure 3.20 reveals that at 100% detection rate, false positives are too high in both cases. D-DCFI performs better even up to 0.0075 false positive rates. Detection rate is almost 0.98 at this point. But after that performance of both schemes falls. Though, detection rate of VBA falls more sharply. The reason behind downfall is high value of thresholds. Traffic thresholds levels are increased to achieve better false positive rates. In case of VBA, normal fluctuations are picked as attacks at lower threshold values. Dispersion rather than volume used in D-DCFI helps to pick attacks with lesser false positive rates. The testing on real state-of-art attack trace is still a pending issue.

After detection, characterization of attack flows is done using direction of skewness observed in packet arrival distribution. Attack flows are filtered at ingress POPs of ISP 4. Results are compared with existing schemes [18] and [141].

Average attack strength of 300Mbps is used to test active queue management techniques RED [141] and DropTail. RED [141] is known for its fair share of bandwidth to all flows for tackling aggressive flows. Goodput and attack traffic is shown in Figure 3.21 for DropTail and RED.

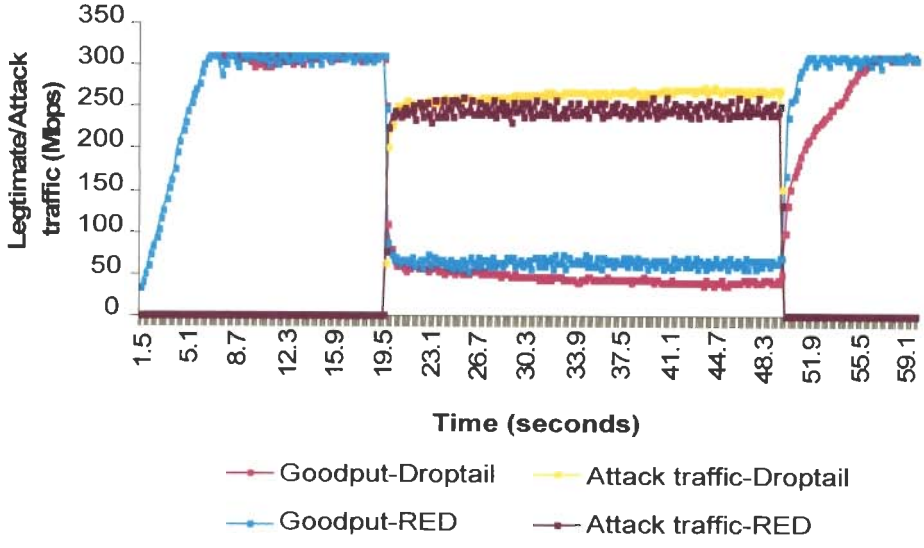

Figure 3.21: Legitimate and attack traffic under DDoS attack

Attack traffic generated during the attack has mean inter arrival time between packets set to 0.00266 seconds. The attack is generated in interval 20-50 seconds. The dip in goodput at 20.1 seconds indicates that attack traffic has reached POP P_s. Thus, attack traffic has seized most of bottleneck bandwidth. Quite evidently, sum of attack traffic and legitimate traffic (goodput) received per unit time is 310Mbps i.e., link utilization is always 1. RED [141] is better than DropTail as for as share of bottleneck bandwidth to legitimate traffic is

concerned. But, goodput achieved in case of RED with large topology and large number of legitimate and attacker clients is not good. Better techniques are required to curb high rate flows. Response provided in VBA is also implemented on topology used in D-DCFI. Pulsing attacks are generated to better characterize the effectiveness of dynamic filtering mechanism. It is found that filtering mechanism is taken off and again triggered whenever network is put under attack in D-DCFI. Attack periods are 20-25, 30-35, and 40-45 seconds. Attack traffic generated during on periods has mean inter arrival time of 0.00266 seconds as taken earlier.

VBA gives reasonable performance at $r = 0.59$, as at this value false positive rate is low. The status of attack traffic with and without defense is shown in Figure 3.22. It compares total attack traffic received and dropped with defense and without applying defense. During attack periods, the total attack traffic generated is around 300Mbps. Detection latency i.e. delay in detecting attack is 0.5 seconds for both D-DCFI and VBA. A delay of 0.1 seconds is caused in communicating and installing filters at POPs of ISP 4. D-DCFI has an extra delay of 0.4 seconds to further refine characterization of attack flows. During this time network is under complete stress of attack. Initially, attack traffic is also dropped because normal TCP traffic has already occupied major share on bottleneck link. But, with time it can be seen in Figure 3.22 that under attack almost 250Mbps is occupied by the attack traffic out of 310Mbps bottleneck capacity and only 50 Mbps of attack traffic is getting dropped. This is due to the fact that attack traffic does not follow congestion and flow control signals, whereas normal TCP traffic flows decrease their flow rates following congestion and flow control signals [140]. Without D-DCFI and VBA, attack traffic received and dropped are around 250Mbps and 50Mbps respectively. Similarly, with VBA and D-DCFI, total attack traffic received and dropped is shown in Figure 3.22. VBA starts

dropping attack packets at 20.6 seconds, whereas D-DCFI starts the same at 21.0 seconds. D-DCFI compensates delay in filtering by dropping whole of attack traffic due to better characterization of attack flows as compared to VBA as shown in Figure 3.22. The overall goodput achieved in D-DCFA is better as compared to VBA. To compare legitimate traffic level, normal packet survival ratio (NPSR) for D-DCFA and VBA is computed. The NPSR is computed with granularity 0.2 seconds. The comparison of NPSR achieved by both schemes is depicted in Figure 3.23. Though, VBA starts filtering malicious traffic earlier, but due to better characterization of attack traffic NPSR achieved in D-DCFA is better.

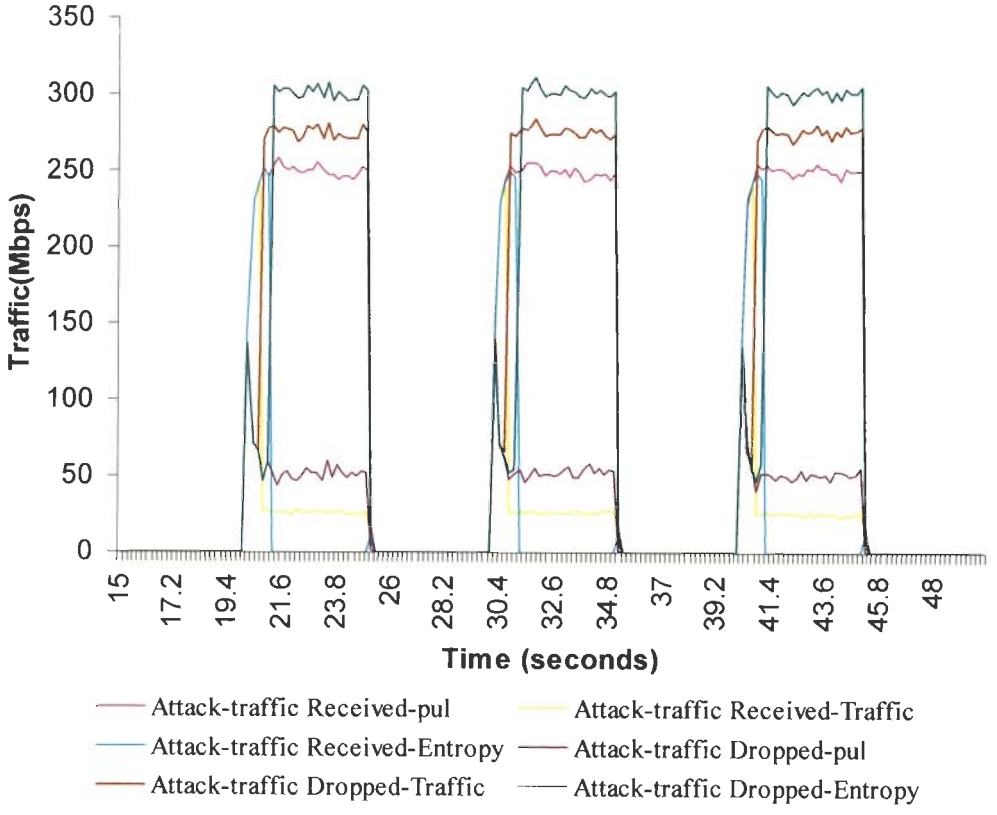

Figure 3.22: Comparative attack traffic received and dropped

The extra time consumed for characterization of attack flows pays its cost in terms of protecting whole of legitimate traffic afterwards. The value of $npsr = 1$ illustrate the effectiveness of D-DCFA. Moreover, when protected server is not under attack, monitoring

of dropped packets at POPs, triggers removal of filters. The detection of attack pulses, and subsequent dropping of complete attack traffic, manifests supremacy of the proposed approach.

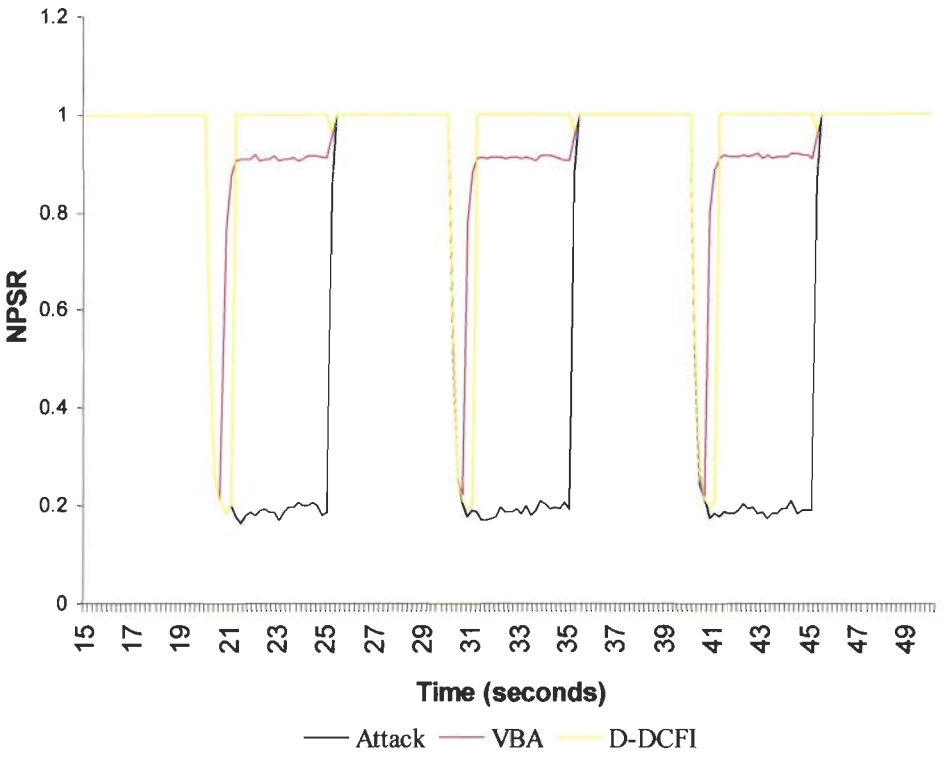

Figure 3.23: Comparative NPSR

3.9 Conclusion

An automated approach to detect both low rate degrading and high rate disruptive flooding DDoS attacks in ISP domain is presented. The validation in NS-2 justifies our claim of detecting DDoS attacks with accuracy. The tradeoff between detection rate and false positive rate, using ROC curves sets the guidelines for profiling normal behavior of any network environment. The response to flooding DDoS attacks is evaluated by computing NPSR at protected server. The NPSR achieved proves the effectiveness of our attack filtering mechanism. The dynamic filtering at ingress POPs near the attack sources also

saves the expensive ISP core bandwidth. Even attack flows, which use spoofed source IP addresses are identified and filtered at ingress POPs. The placement of all modules in single ISP domain makes the approach practical. The proposed approach can be incrementally deployed in multiple ISPs with help of trusted entities acting as interfaces between two ISPs.

Chapter 4

A Distributed Approach to Detect DDoS Attacks in ISP Domain

DDoS attacks are best detected near the victim's site as maximum attack traffic converges at this point. In most of current solutions, monitoring and analysis of traffic for DDoS detection have been carried at a single link, which connects victim to ISP. However the mammoth volume generated by DDoS attacks pose the biggest challenge in terms of memory and computational overheads. These overheads make DDoS solution itself vulnerable against DDoS attacks as detection module is unable to handle aggregate traffic. D-DCFI proposed in chapter 3 is a distributed approach in ISP domain. It detects flooding DDoS attacks at the POP connected to victim server and filter DDoS traffic at responsible POPs closer to source of the attack in the same ISP domain. The detection functionality in D-DCFI is extended in this chapter by distributing monitoring and analysis overheads amongst all POPs of the ISP using a traffic feature distribution based approach.

4.1 Introduction

High rate flooding (HRFD) DDoS attacks generate an overwhelming volume of packets directed to victim server. These packets arrive in such a high quantity that some key resource is quickly exhausted at the victim [91]. Statistically, network bandwidth, system memory, and CPU processing capacity are the most common targets of HRFD attacks.

When any of these resources form a bottleneck, performance of victim is severely affected, impeding legitimate use of a service. To provide a better response against these attacks, it is necessary to detect the attack in real time and filter attack packets in an automated manner at perimeter of the protected domain. Real time detection of HRFD attacks requires on line packet monitoring. As HRFD attacks make use of seemingly legitimates packets, so simply checking payload of a packet cannot provide signs of an attack. A thorough investigation of collected packets is needed to find patterns of attack within amassed traffic near the victim. In actual HRFD attack scenario where server is attached to fast backbone, access control lists [144], firewalls [36, 113, 130] and intrusion detection systems [55, 174] deployed are unable to handle voluminous traffic due to limited memory and processing capacity thresholds. As a result, they start dropping the packets. So instead of protecting the network, security systems themselves become a reason for causing denial of service. Consequently to resume services, either victim networks wait for the attacker to command zombies to stop flooding attack traffic or a responsible official manually inform upstream ISPs using other communication services e.g. telecom services to filter traffic destined towards victim network. Hence, the actual purpose of real time detection and automated response to filter attack traffic is totally defeated.

D-DCFI explained in chapter 3 monitors real time traffic at POP P_s, which is connected to victim server as shown in Figure 3.3. Detection scheme in D-DCFI is extended in this chapter by monitoring and analyzing traffic at all POPs of the protected ISP domain. A formula is derived for distributing overheads among multiple POPs of the protected ISP domain. Thus, space and computational overheads are distributed at multiple points in extended D-DCFI. The rest of the chapter is organized as follows. Section 4.2 discusses effectiveness of DDoS attack detection at various locations on the Internet. Section 4.3

highlights conflicting requirements of an ideal DDoS attack detection system. Section 4.4 explains proposed approach. Section 4.5 derives the formula for distributing overheads among multiple POPs. In section 4.6, computational complexities are computed for detection approach in D-DCFI and extended distributed approach to detect flooding DDoS attacks. Section 4.7 provides discussion. Finally, section 4.8 concludes the chapter.

4.2 Effectiveness of DDoS Attack Detection

Figure 4.1 shows that the DDoS detection can be performed in four places on the paths between the victim and the zombie agents or reflectors [132]. As depicted in the diagram, a DDoS attack resembles a funnel in which attack packets are generated from a dispersed area, like the top of a funnel.

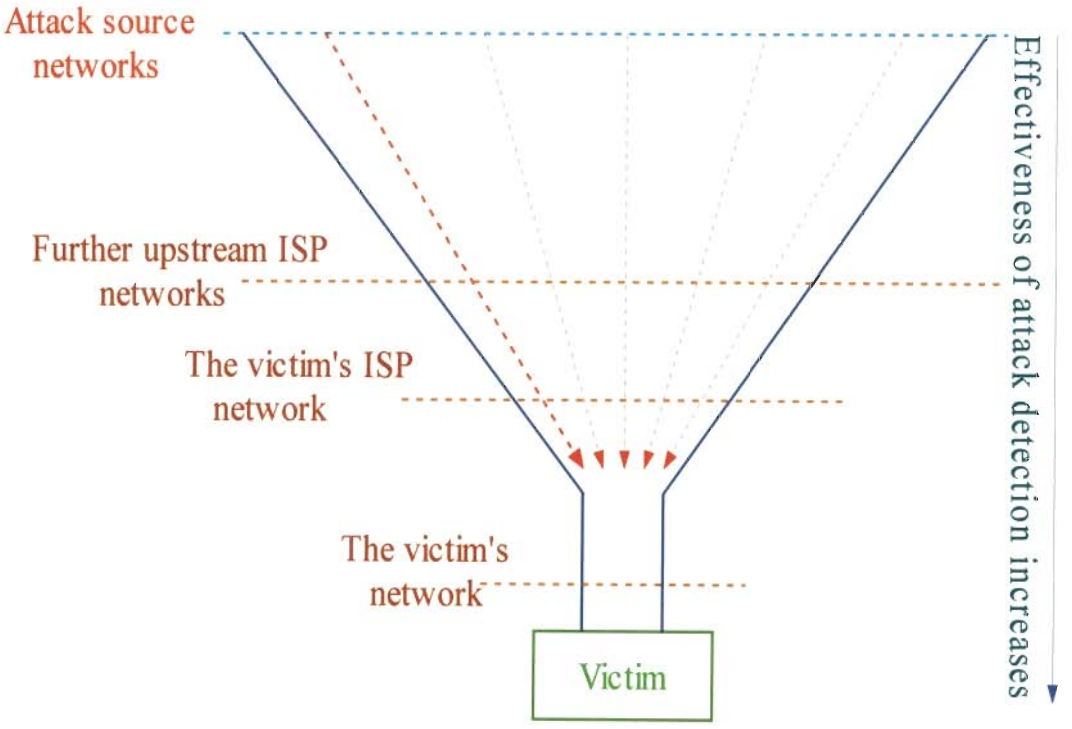

Figure 4.1: Possible locations for DDoS attack detection

The victim, like the narrow end of a funnel, receives all the attack packets generated. Thus, detecting a DDoS attack is "relatively" easy at the victim network, because it can observe

all the attack packets. In contrast, it is less likely for an individual source network, where attack sources (zombie agents and reflectors) are located, to detect the attack unless a large number of attack sources are located in that network.

A discussion of effectiveness of DDoS attack detection process at various locations is given next.

- **At Source Networks** — The basic cause of almost all security problems in Internet is huge base of vulnerable machines on the Internet. These unpatched machines are used as slaves by attackers to launch attacks against high profile sites [89]. In highly isotropic attacks, each zombie generates very less amount of attack traffic to launch a flooding DDoS attack. To handle this attack scenario, it is a big question whether we have reliable models to detect DDoS attacks at source end. Secondly, characterization of attack sources requires substantial attack data as flooding DDoS traffic completely blends itself with legitimate traffic in such a manner that no differentiation can be made on packet-by-packet basis [91]. Hence, detection of attack and characterization of attack sources in case of flooding DDoS cannot be tackled reliably at source end. Moreover, global deployment is required for source end detection to be completely effective. As Internet is decentralized management system, expecting global deployment looks an ambitious goal in near future.
- **At the Victim's Network** — Unlike the case for source networks, a DDoS victim can detect (by a router, intrusion detection system, or network operator) a DDoS attack based on an unusually high volume of incoming traffic (of certain packet types) or degraded server and network performance. In fact, a number of companies have offered products based on this traffic anomaly approach to detect DDoS attacks. These commercial detection systems are usually placed in a network under

protection or in a service provider's network. Although the details of the attack detection algorithms are not disclosed, statistical approaches employed by other intrusion detection systems, notably EMERALD, have been presented and discussed in the past [122]. However, if an incoming link is jammed by attack packets, a victim practically cannot do anything but shut down its network and ask the upstream ISP to filter the packets involved [143].

- **At a Victim's Upstream ISP Network** — Frequently, an upstream ISP is requested (through telephone calls) by a DDoS victim to filter attack packets. To speed up and automate this process, a victim network may send to an upstream ISP router an intrusion alert message, which specifies the signatures of the attack packet flows, as soon as detecting a DDoS attack. Such intrusion alert protocol needs to be designed carefully. The messages also have to be protected by strong authentication and encryption algorithms. In general as we move away from victim's network, total attack traffic at any single point is always lesser than aggregate traffic. Without collaborative efforts, it is not easy to detect flooding DDoS attacks at this point. Extra burden of state monitoring on core routers also hinders detection schemes to be installed in victim's ISP domain. However, if sufficient motivation in terms of incentives is provided in ISP domain then Victim's ISP network can provide DDoS detection and filtering capability to curb DDoS attacks as they are managed by a single administrative authority and have resources to handle traffic. The future looks bright for implementation of collaborative approaches to detect DDoS attacks.
- **At Further Upstream ISP Networks** — In principle, one could extend the backpressure approach just described to further upstream ISP networks. The victim network is responsible for detecting DDoS attacks, and the upstream ISPs are then

notified to filter packets which contain signatures of the attack packet flows. In other words, packet filtering is pushed as upstream as possible [83, 139, 157]. Similar to the ubiquitous ingress filtering, this approach is effective only if ISP networks are willing to cooperate and allow installation of packet filters upon receiving intrusion alerts. As per current Internet architecture, detection is not at all effective at further upstream ISP networks.

In brief, funnel like structure of flooding DDoS attacks and Internet architecture vulnerabilities suggest decrease in effectiveness of detection scheme as we move away from victim towards source networks via intermediate network consisting of interconnected ISPs.

4.3 Conflicting Requirements of an Ideal DDOS Detection System

Real time detection of flooding DDoS attacks involves monitoring and analysis of network traffic. Either time windows [70, 124] or packet windows [102] are used to monitor real time traffic. Statistics about header fields of the packets are recorded in a data structure. For detecting DDoS attacks, these data structures are maintained in featured machines or routers on the Internet. Core routers in the intermediate network are responsible for forwarding packets at high speeds using routing tables. As per original Internet design, these routers do not keep statistics about packets due to limited memory. Thus, maintaining extra data structures in memory of core routers generates expensive memory overheads. In case of HRFD attacks, volume of monitored traffic is so high that even security systems meant to protect network are unable to handle excessive traffic due to limited memory resources.

Moreover, analysis of monitored traffic requires complex algorithms to detect DDOS attacks and characterize attack traffic. These complex algorithms generate highly expensive computational overheads. Time complexity represented in terms of asymptotic notations like $O, \Omega,$ *and* Θ, describes the scalability of algorithms i.e. as the size of the input to an

algorithm increases, how the running time of the algorithm change [151]. The volume of traffic in Internet backbones is so high that applying complex algorithms can have serious repercursions. Especially under HRFD attacks , security systems itself can come down to its knees in handling voluminous traffic. A prominent example is that during the spread of the "Code Red" worm [33], over 300,000 "zombie" machines were compromised to launch a denial of service attack on the White House Web site [45]. Due to large number of zombies traffic volume generated by a DDoS attack can exceed 10Gb/s [170].

Therefore, a detection system must incur low computational overheads, otherwise it can itself become target of a voluminous DDoS attack. In view of this and section 4.2, requirements for DDoS attacks detection and characterization are:

- Whole traffic instead of partial traffic should be monitored and analyzed for detecting and characterizing attacks in real time.
- Detection system must incur low computational overheads.

Quite evidently, these two are conflicting requirements as in case of flooding DDoS attacks, high volume of seemingly legitimate packets are generated. Monitoring and analyzing whole of aggregated traffic incur high computational overheads. Moreover, availability of variety of DDoS attacks tools and growing intelligence of hackers demand higly complex algorithms to detect and charterize DDoS attacks. It makes the situation worse for DDoS defenders.

In nutshell, a scheme which can monitor all traffic destined to victim and analyze the same at single point gives best detection results. But in this case computational overheads are also centered at single point which is itself vulnerability as for as huge volume of traffic generated by DDoS attack is concerned. So a technique, which can monitor and analyze traffic at distributed points, but actually behave as if the total traffic is monitored and

analyzed at single point, is good for DDoS attack detection. So by applying this technique computational overheads can be distributed from single point to multiple points.

4.4 Proposed Distributed Approach

Detection of flooding DDoS attacks in D-DCFI proposed in previous chapter is done using traffic feature distributions. Sample entropy of traffic flows defined in equation (3.1) is used to summarize traffic feature distributions. An anomaly based detection model with threshold limits defined in equation (3.2) is used for detecting LRFD and HRFD attacks.

An ISP level topology given in Figure 3.3 is used. For details of simulation parameters refer to section 3.5. In D-DCFI POP P_s is responsible for monitoring all traffic originated in four ISP domains. The proposed extension in detection approach of D-DCFI considers all POPs of the protected ISP domain 4 for monitoring traffic in real time. A similar packet monitoring process as in D-DCFI is used to monitor traffic at all POPs of the ISP. Refer to subsection 3.4.1 for details. Sample entropy E_i is computed at all POPs in intervals of Δ seconds (refer subsection 3.4.2 for computation of sample entropy). A part of the ISP level topology in which proposed approach is implemented is shown in Figure 4.2.

Let n is the number of POPs in the protected ISP domain.

Let $\{N_1, N_2 \ldots\ldots\ldots N_k\}$ N_i is number of flows in POP P_i.

Let $\{H_1, H_2, \ldots\ldots\ldots H_n\}$ H_i is frequency histograms (i.e. number of packets arrivals for each flow) associated with POP P_i in time window Δ at time t where $H_i = \{X_{i1}, X_{i2}, \ldots\ldots\ldots X_{iN_i}\}$ and X_{ij} represent number of packets for POP P_i and flow j. Let $\{S_1, S_2, \cdots\cdots\cdots S_n\}$ $S_i = \sum_{j=1}^{N_i} X_{ij}$ i.e. total number of packets observed in a time interval at a particular POP P_i.

Let E_i represent sample entropy for POP P_i at any time t.

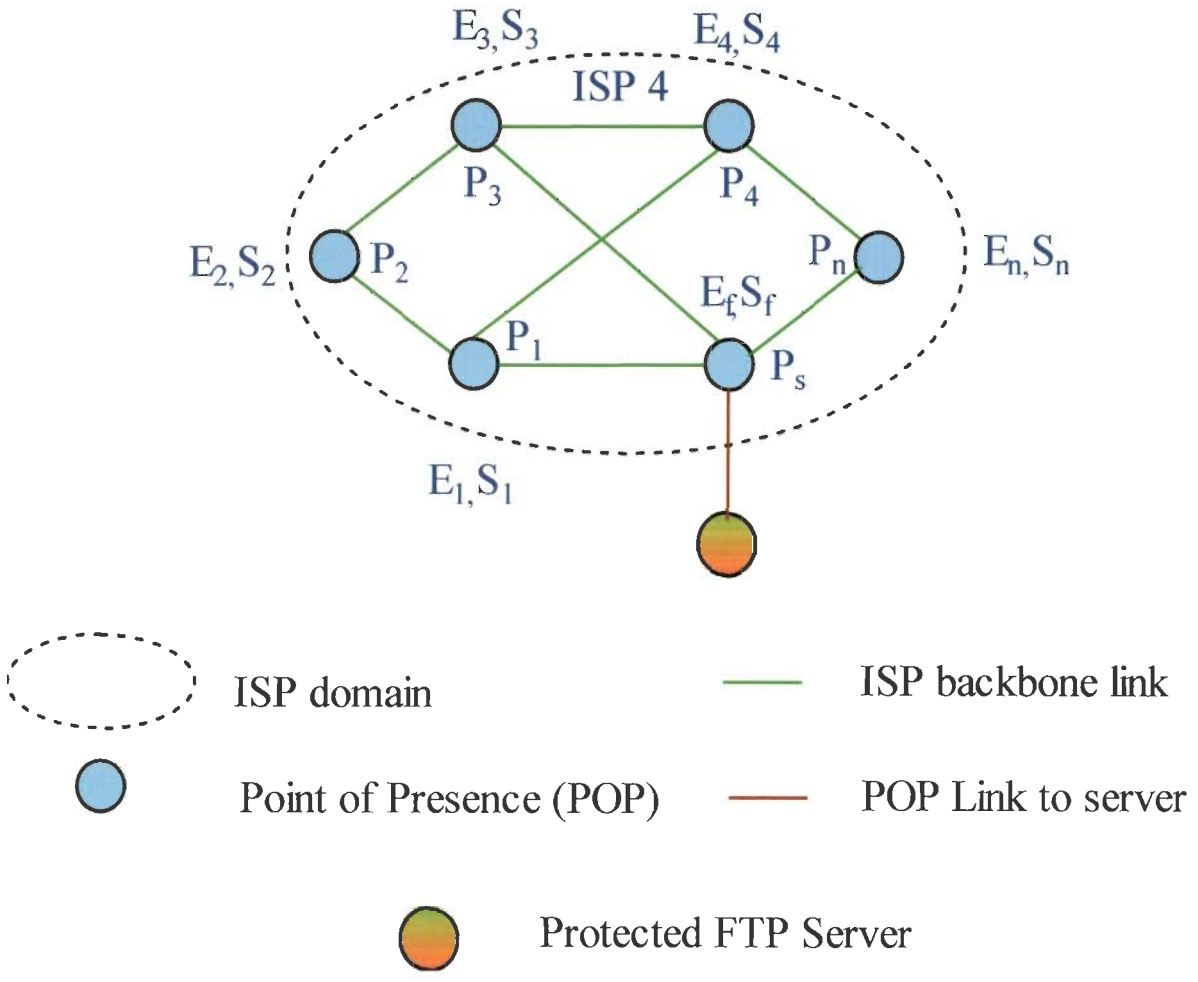

Figure 4.2: Architecture of distributed detection approach at ISP 4

Table 4.1 summarizes statistics monitored and analyzed at all POPs. Each row of the table provides statistics related to POP i.Each column except last represents number of packet arrivals for each flow. Last column represents computed sample entropy at all POPs. Flows $\{1,2,3,4,\ldots\ldots N_i\}$ are different for each POP P_i. Though some flows can be same. A time series of these statistics give sample entropies computed at all POPs in intervals of Δ seconds. Sample entropy E_i computed at P_i with S_i is sent to P_s (POP connected to the protected server) where final sample entropy E_f is calculated using equation (4.1) defined as follows:

$$E_f = (1/S_f)\sum_{i=1}^{n} S_i(E_i - \log(S_i)) + \log(S_f) \quad (4.1)$$

where $S_f = \sum_{i=1}^{n} S_i$ is total number of packets observed at all POPs in Δ seconds.

Table 4.1: Statistics collected at all POPs of the ISP 4

	1	**2**	**3**	**4**	**5**	.....	.....	N_i	**Total number of arrived packets** (S_i)	**Sample Entropy** (E_i)
1	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	.....	.....	X_{1N_1}	S_1	E_1
2	X_{21}	X_{22}	X_{23}	X_{24}	X_{25}	.....	.....	X_{2N_2}	S_2	E_2
3	X_{31}	X_{32}	X_{33}	X_{34}	X_{35}	.....	.....	X_{3N_3}	S_3	E_3
4	X_{41}	X_{42}	X_{43}	X_{44}	X_{45}	.....	.....	X_{4N_4}	S_4	E_4
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮		⋮	⋮
n	X_{n1}	X_{n2}	X_{n3}	X_{n4}	X_{n5}	.....	.....	X_{nN_n}	S_n	E_n

Final sample entropy E_f computed using equation (4.1) gives the same value as if whole traffic is monitored and analyzed at single point P_s. Analytical and experimental proofs for the same are given in subsequent section. Sample entropy E_f computed using equation (4.1) is put in current sample entropy $H_c(X)$ of detection model given by equation (3.2) and is compared with already profiled normal value of sample entropy $H_n(X)$ at POP P_s. The communication delay to notify P_s about individual sample entropies computed at ingress POPs E_i introduces extra detection delay.

In short, we do not monitor traffic at single link instead it is done at all ingress points of the ISP. Sample entropy E_i computed at these points is sent to the coordinator P_s (a router responsible for analysis). The final entropy E_f is computed using equation (4.1). At the cost

of communication overhead, memory and computational overheads are distributed. Thus, proposed distributed approach handles high volume of packets generated by DDoS attacks by distributing monitoring and analysis among all ingress POPs of the ISP. So memory and computational overheads are distributed among all ingress POPs of the protected ISP domain.

The values of E_i and S_i are not communicated in a secure manner in extended detection approach of D-DCFI. Moreover, extended D-DCFI is limited to ISP domain 4 only. The ISP network may also be jammed under sufficiently large-scale DDoS attacks [132]. So, a comprehensive solution against flooding DDoS attacks must include multiple ISP domains.

4.5 Proof

The equation (4.1) is proved both analytically and experimentally in next subsections.

4.5.1 Analytical

The proof of equation (4.1) is given as follows:

Table 4.1 shows statistics collected at all POPs of the ISP 4. The sample entropy E_i at POP P_i is computed using equation (3.1) as follows:

For $\log_2$, log is used in derivation.

$$-E_i = \sum_{j=1}^{N_i}((x_{ij}/S_i)\log(x_{ij}/S_i))$$

$$\Rightarrow -E_1 = \sum_{j=1}^{N_1}((x_{1j}/S_1)\log(x_{1j}/S_1))$$

$$\Rightarrow -E_1 = (x_{11}/S_1)\log(x_{11}/S_1) + (x_{12}/S_1)\log(x_{12}/S_1) + \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots + (x_{1N_1}/S_1)\log(x_{1N_1}/S_1)$$

$$\Rightarrow -E_1 = \log(x_{11}/S_1)^{(x_{11}/S_1)} + \log(x_{12}/S_1)^{(x_{12}/S_1)} + \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots + \log(x_{1N_1}/S_1)^{(x_{1N_1}/S_1)}$$

$$\Rightarrow -E_1 = \log((x_{11}/S_1)^{(x_{11}/S_1)}(x_{12}/S_1)^{(x_{12}/S_1)}\ldots\ldots\ldots\ldots\ldots\ldots(x_{1N_1}/S_1)^{(x_{1N_1}/S_1)})$$

Using $\log(x+y) = \log(xy)$

$$\Rightarrow -E_i = \log((x_{i1}/S_i)^{(x_{i1}/S_i)}(x_{i2}/S_i)^{(x_{i2}/S_i)}\ldots\ldots\ldots\ldots(x_{iN_i}/S_i)^{(x_{iN_i}/S_i)})$$

$$\Rightarrow 2^{-E_i} = (x_{i1}/S_i)^{(x_{i1}/S_i)}(x_{i2}/S_i)^{(x_{i2}/S_i)}\ldots\ldots\ldots\ldots(x_{iN_i}/S_i)^{(x_{iN_i}/S_i)}$$

Using $a = \log_b(x) \Rightarrow b^a = x$

As $S_f = \sum_{i=1}^{n} S_i$

Considering all flows are observed at POP P_s. The traffic destined to only protected server is monitored at all ingress POPs of the ISP 4. Moreover, we have assumed that flows observed by all POPs directed to protected server are different at different POPs.

$$2^{-E_f} = (x_{11}/S_f)^{(x_{11}/S_f)}(x_{12}/S_f)^{(x_{12}/S_f)}\ldots\ldots\ldots\ldots(x_{1N_1}/S_f)^{(x_{1N_1}/S_f)}$$

$$(x_{21}/S_f)^{(x_{21}/S_f)}(x_{22}/S_f)^{(x_{22}/S_f)}\ldots\ldots\ldots\ldots(x_{2N_2}/S_f)^{(x_{2N_2}/S_f)}$$

$$(x_{31}/S_f)^{(x_{31}/S_f)}(x_{32}/S_f)^{(x_{32}/S_f)}\ldots\ldots\ldots\ldots(x_{3N_3}/S_f)^{(x_{3N_3}/S_f)}$$

$$\vdots$$

$$\vdots$$

$$\vdots$$

$$\vdots$$

$$(x_{n1}/S_f)^{(x_{n1}/S_f)}(x_{n2}/S_f)^{(x_{n2}/S_f)}\ldots\ldots\ldots\ldots(x_{nN_n}/S_f)^{(x_{nN_n}/S_f)}$$

Multiply numerator and denominator by S_i where i varies from 1 to n.

$$\Rightarrow 2^{-E_f} = ((x_{11}/S_f)(S_1/S_1))^{((x_{11}/S_f)(S_1/S_1))}((x_{12}/S_f)(S_1/S_1))^{((x_{12}/S_f)(S_1/S_1))}\ldots\ldots\ldots\ldots$$

$$\ldots\ldots\ldots\ldots((x_{1N_1}/S_f)(S_1/S_1))^{((x_{1N_1}/S_f)(S_1/S_1))}$$

$$((x_{21}/S_f)(S_2/S_2))^{((x_{21}/S_f)(S_2/S_2))}((x_{22}/S_f)(S_2/S_2))^{((x_{22}/S_f)(S_2/S_2))}\ldots\ldots\ldots\ldots$$

$$\ldots\ldots\ldots\ldots((x_{2N_2}/S_f)(S_2/S_2))^{((x_{2N_2}/S_f)(S_2/S_2))}$$

$$\vdots$$

$$\vdots$$

$$((x_{n1}/S_f)(S_n/S_n))^{((x_{n1}/S_f)(S_n/S_n))}((x_{n2}/S_f)(S_n/S_n))^{((x_{n2}/S_f)(S_n/S_n))}\ldots\ldots\ldots\ldots$$

$$\ldots\ldots\ldots\ldots((x_{nN_n}/S_f)(S_n/S_n))^{((x_{nN_n}/S_f)(S_n/S_n))}$$

$$\Rightarrow 2^{-E_f} = ((x_{11}/S_1)(S_1/S_f))^{((x_{11}/S_1)(S_1/S_f))}((x_{12}/S_1)(S_1/S_f))^{((x_{12}/S_1)(S_1/S_f))}\ldots\ldots\ldots\ldots\ldots$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots((x_{1N_1}/S_1)(S_1/S_f))^{((x_{1N_1}/S_1)(S_1/S_f))}$$

$$((x_{21}/S_2)(S_2/S_f))^{((x_{21}/S_2)(S_2/S_f))}((x_{22}/S_2)(S_2/S_f))^{((x_{22}/S_2)(S_2/S_f))}\ldots\ldots\ldots\ldots\ldots$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots((x_{2N_2}/S_2)(S_2/S_f))^{((x_{2N_2}/S_2)(S_2/S_f))}$$

$$\vdots$$

$$\vdots$$

$$((x_{n1}/S_n)(S_n/S_f))^{((x_{n1}/S_n)(S_n/S_f))}((x_{n2}/S_n)(S_n/S_f))^{((x_{n2}/S_n)(S_n/S_f))}\ldots\ldots\ldots\ldots\ldots$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots((x_{nN_n}/S_n)(S_n/S_f))^{((x_{nN_n}/S_n)(S_n/S_f))}$$

Replacing 2^{-E_i} for

$$(x_{i1}/S_i)^{(x_{i1}/S_i)}(x_{i2}/S_i)^{(x_{i2}/S_i)}\ldots\ldots\ldots\ldots(x_{iN_i}/S_i)^{(x_{iN_i}/S_i)} \quad \forall\, i = 1\ to\ n$$

$$2^{-E_f} = (2^{-E_1})^{(S_1/S_f)}(2^{-E_2})^{(S_2/S_f)}\ldots\ldots\ldots\ldots\ldots\ldots(2^{-E_n})^{(S_n/S_f)}$$
$$(S_1/S_f)^{(S_1/S_f)}(S_2/S_f)^{(S_2/S_f)}\ldots\ldots\ldots\ldots\ldots\ldots(S_n/S_f)^{(S_n/S_f)}$$

Taking Log_2 on both sides

$$\Rightarrow -E_f = \log((2^{-E_1})^{(S_1/S_f)}(2^{-E_2})^{(S_2/S_f)}\ldots\ldots\ldots\ldots\ldots(2^{-E_n})^{(S_n/S_f)})$$
$$+\log((S_1/S_f)^{(S_1/S_f)}(S_2/S_f)^{(S_2/S_f)}+\ldots\ldots\ldots\ldots+(S_n/S_f)^{(S_n/S_f)})$$

$$-E_f = -E_1(S_1/S_f) - E_2(S_2/S_f) - \ldots\ldots\ldots\ldots\ldots\ldots - E_n(S_n/S_f)$$
$$+(S_1/S_f)\log(S_1/S_f) + (S_2/S_f)\log(S_2/S_f) + \ldots\ldots\ldots + (S_n/S_f)\log(S_n/S_f)$$

$$\Rightarrow E_f = E_1(S_1/S_f) + E_2(S_2/S_f) + \ldots\ldots\ldots\ldots\ldots\ldots + E_n(S_n/S_f)$$
$$-(S_1/S_f)\log(S_1/S_f) - (S_2/S_f)\log(S_2/S_f) - \ldots\ldots\ldots - (S_n/S_f)\log(S_n/S_f)$$

$$\Rightarrow E_f = 1/S_f(E_1S_1 + E_2S_2 + \ldots\ldots\ldots\ldots\ldots\ldots + E_nS_n)$$
$$-1/S_f(S_1\log(S_1/S_f) + S_2\log(S_2/S_f) + \ldots\ldots\ldots\ldots + S_n\log(S_n/S_f))$$

$$\Rightarrow E_f = 1/S_f((E_1S_1 + E_2S_2 + \ldots\ldots\ldots\ldots\ldots\ldots + E_nS_n)$$
$$-(S_1\log(S_1/S_f) + S_2\log(S_2/S_f) + \ldots\ldots\ldots\ldots + S_n\log(S_n/S_f))$$

$$\Rightarrow E_f = 1/S_f((E_1S_1 + E_2S_2 \ldots\ldots + E_nS_n) - (S_1\log(S_1) - S_1\log(S_f) + S_2\log(S_2) - S_2\log(S_f) \ldots\ldots + S_n\log(S_n) - S_n\log(S_f))$$

$$\Rightarrow E_f = 1/S_f(S_1(E_1 - \log(S_1)) + S_2(E_2 - \log(S_2)) \ldots\ldots + S_n(E_n - \log(S_n)) + S_1\log(S_f) + S_2\log(S_f)) \ldots\ldots + S_n\log(S_f))$$

$$\Rightarrow E_f = 1/S_f(S_1(E_1 - \log(S_1)) + S_2(E_2 - \log(S_2)) \ldots\ldots + S_n(E_n - \log(S_n)) + \log(S_f)(S_1 + S_2 + \ldots\ldots + S_n))$$

As $S_f = S_1 + S_2 \cdots\cdots S_n$

$$\Rightarrow E_f = 1/S_f(S_1(E_1 - \log(S_1)) + S_2(E_2 - \log(S_2)) + .. + S_n(E_n - \log(S_n)) + S_f\log(S_f))$$

$$E_f = (1/S_f)\sum_{i=1}^{n}(S_i(E_i - \log(S_i)) + \log(S_f)$$

This is same as equation (4.1).

4.5.2 Experimental

Simulations are conducted in NS-2 without attack and with attack (at 0.1Mbps per attacker) using 100 attackers. Details of simulation parameters are given in section 3.5. Monitoring and computation of sample entropy E_i are done at POPs of ISP 4 as shown in Figure 4.2. Monitoring and computation of sample E_f are performed at POP P_s.The values of E_i and S_i are put in equation (4.1) for all POPs of ISP domain 4. The final computed value matches E_f directly computed at POP P_s. Thus complexity of analyzing total traffic is distributed among all POPs of the protected ISP 4. Table 4.2 shows all computed E_i values at POPs and stepwise calculations for computing final entropy using equation (4.1) without attack.

Table 4.2: Sample entropies at POPs and step wise calculations for equation (4.1) without attack

POP number	Sample entropy (E_i)	S_i	$\log(S_i)$	$S_i(E_i - \log(S_i))$
1	3.317282	231	7.851749	-1047.46
2	3.295047	223	7.8009	-1004.81
3	2.969538	148	7.209453	-627.507
4	3.114191	199	7.636625	-899.964
5	3.102673	161	7.330917	-680.747
6	2.842068	166	7.375039	-752.473
7	3.321733	242	7.918863	-1112.51
8	3.115487	186	7.539159	-822.803
9	3.100056	184	7.523562	-813.925
10	3.169925	216	7.754888	-990.352
11	5.570005	998	9.962896	-4384.11
12	7.711671	4674	12.19044	-20933.8

$$S_f = \sum_{i=1}^{i=12} S_i = 7628 \text{ and } \log_2(S_f) = 12.89709$$

$$\sum_{i=1}^{n} (S_i(E_i - \log(S_i)) = -34070.4$$

$$1/S_f \sum_{i=1}^{n} (S_i(E_i - \log(S_i)) = -4.4665$$

$$(1/S_f) \sum_{i=1}^{n} (S_i(E_i - \log(S_i)) + \log(S_f) = 8.430594$$

Sample entropy E_f computed at POP P_s is also 8.430594.

Hence equation (4.1) computes final sample entropy E_f from sample entropies E_i computed at ingress points of ISP 4 as if whole traffic is monitored and sample entropy is computed at single POP P_s. A graphical representation of sample entropies computed at all ingress POPs of ISP domain 4 is shown in Figure 4.3.

POPs 1-10 are responsible for traffic originated in ISP domain 4 only. POP 11 and POP 12 are connected to POPs of ISP domain 1 and 3 respectively and are responsible for traffic

coming from other ISP domains. Figure 4.3 shows traffic is evenly distributed within ingress POPs of ISP domain 4. POP 11 and POP 12 are responsible for mixed traffic of ISP 1 and 2, and ISP 3 and 2 respectively. Sample entropy computed by using equation (4.1) and sample entropy computed at POP P_s proves correctness of proposed approach.

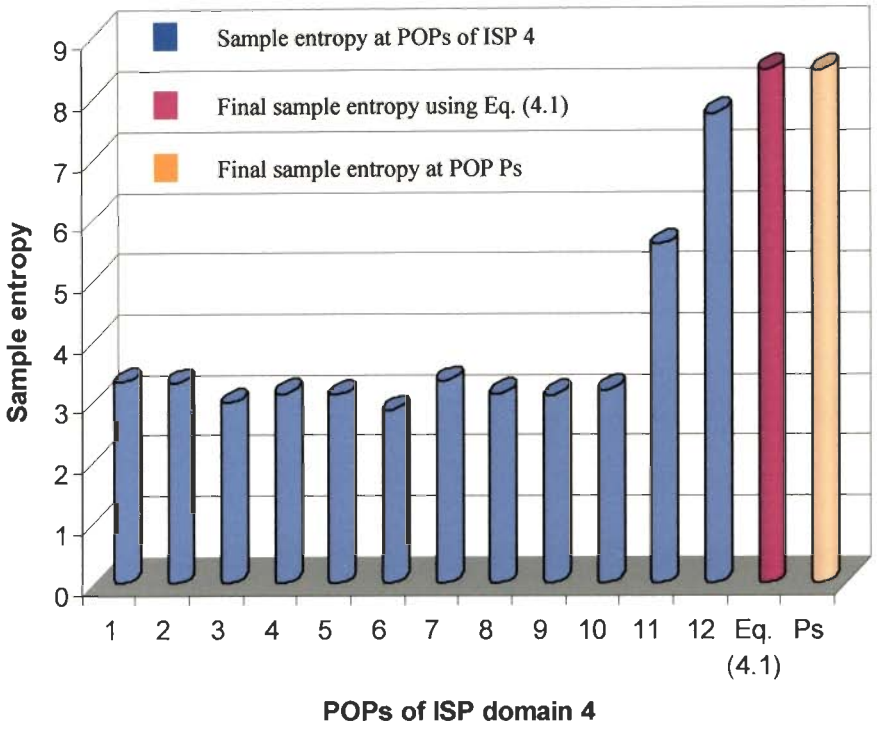

Figure 4.3: Distribution of sample entropy among POPs of ISP domain 4 without attack

The simulation experiments are now carried at attack strength of 0.1Mbps per attacker. Table 4.3 repeats step wise calculations under attack. The attackers are also uniformly distributed among all ISPs. Twenty five attackers launch attack against protected server from all ISPs. Here attackers mean zombie machines which are exploited by attackers to launch attacks against the protected servers. The attack as well legitimate traffic from ISP domain 1, 2 and 3 enter in protected ISP domain 4 through ISP-ISP peer links. This is evident from Figure 3.3. Figure 4.4 shows distribution of sample entropies among POPs of ISP domain 4 under attack.

Table 4.3: Sample entropies at POPs and step wise calculations for equation (4.1) with attack strength 0.1Mbps per attacker using 100 attackers

POP number	Sample entropy (E_i)	S_i	$\log(S_i)$	$S_i(E_i - \log(S_i))$
1	3.722487	231	7.851749	-953.86
2	3.696628	244	7.930737	-1033.12
3	3.583207	212	7.72792	-878.679
4	3.604573	222	7.794416	-930.145
5	3.695111	206	7.686501	-822.226
6	3.686409	224	7.807355	-923.092
7	3.544096	178	7.475733	-699.831
8	3.570212	211	7.721099	-875.837
9	3.611803	207	7.693487	-844.909
10	3.715312	268	8.066089	-1166.01
11	5.816956	1063	10.05393	-4503.9
12	7.916632	4781	12.2231	-20589.2

$$S_f = \sum_{i=1}^{i=12} S_i = 8047 \text{ and } \log_2(S_f) = 12.97424$$

$$\sum_{i=1}^{n} (S_i(E_i - \log(S_i)) = -34220.8$$

$$1/S_f \sum_{i=1}^{n} (S_i(E_i - \log(S_i)) = -4.25262$$

$$(1/S_f) \sum_{i=1}^{n} (S_i(E_i - \log(S_i)) + \log(S_f) = 8.721617$$

Sample entropy E_f computed at POP P_s is also 8.721617.

The same value of sample entropy using equation (4.1) and simulation once again proves that distributed approach computes sample entropy by distributing the computational burden without sacrificing in precision of final sample entropy computation. We repeated the simulation using eight flows common between POP 11 and POP 12. Equation (4.1) assumes these flows to be separate and takes them as sixteen flows, whereas at POP P_s the common flows coming from different ingress POPs are taken as one rather than two, so it considers them as total eight flows only. It introduces an error as depicted in Figure 4.5.

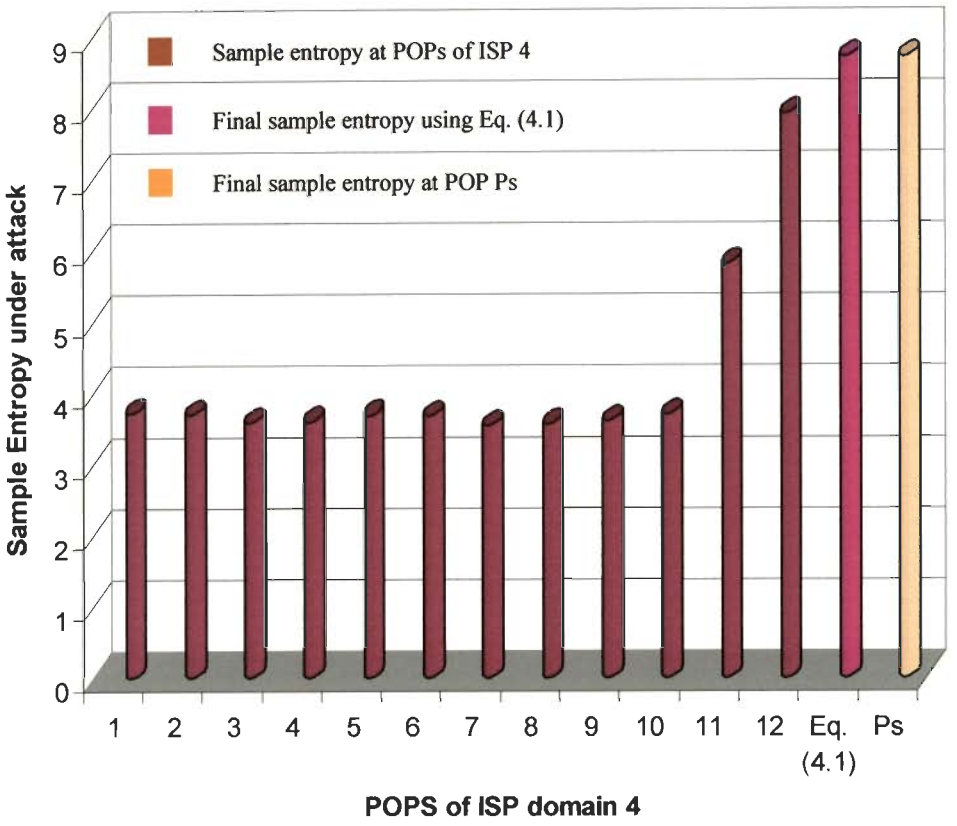

Figure 4.4: Distribution of sample entropy at POPs of ISP domain 4 under attack

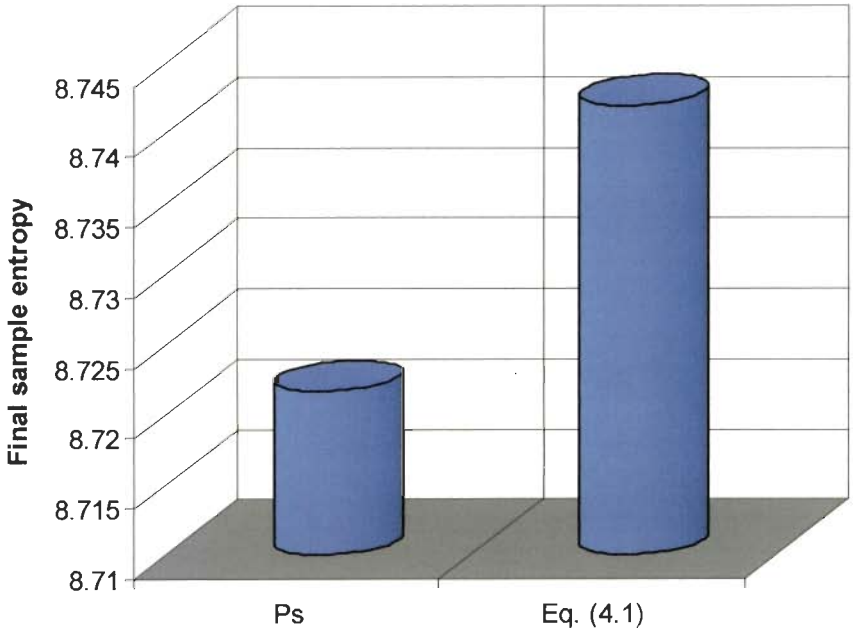

Figure 4.5: Error due to common flows

As equation (4.1) considers each common flow as two flows, it increases the dispersion and hence final sample entropy E_f.

The equation (4.1) gives accurate results when flows are not common at different POPs of the ISP domain. Though, because of spoofing of packet header fields and dynamic routing, the flows can be common in practical scenario. The equation (4.1) requires extension in order to handle this scenario.

4.6 Computational complexity

Computational complexity of an algorithm describes "As the size of the input to an algorithm increases, how the running time and memory requirements of the algorithm change and what are the implications and ramifications of that change?"[151]. The time complexity of an algorithm is mesured as number of steps it takes to solve an instance of the problem as a function of the size of the input. The space complexity of a problem is a related concept, that measures the amount of space, or memory required by the algorithm.

Sample entropy is a very simple metric to summarize traffic feature distributions. Computation of sample entropy at POP P_s does not involve complex computations as depicted in figure 3.5. The time complexity of the algoritham is $O(N)$ where N is number of traffic flows monitored in a particular time window $\{t-\Delta, t\}$. As for as space requirements are concerened as shown in Figure 3.5, a two dimentional array of size $(N \times 2)$ is used as input to compute sample entropy of traffic flows distribution. It is quite evident that time and space requirents are dependent on volume of traffic. On the other hand in case of distributed appraoch as shown in table 4.2 and 4.3, time complexity to compute final sample entropy at POP P_s by using equation 4.1 is $O(n)$, where n is number of POPs in the protected ISP domain. Time complexities to compute sample entropies at all

ingress POPs of the ISP domain are $O(N_1), O(N_2), O(N_3), O(N_4) \ldots\ldots\ldots O(N_n)$ where N_i is number of flows at POP P_i .So time complexity of distributed detection approach is as follows:

$$\text{Time complexity} = O(n) + O(N_1) + O(N_2) + O(N_3) + O(N_4) + \ldots\ldots\ldots + O(N_n)$$

$$\text{Time complexity} = O(n) + O(N) \tag{4.2}$$

$$\text{where } N = N_1 + N_2 + N_3 + \ldots\ldots\ldots N_n$$

The equation (4.2) reveals that time complexity of the distributed approach depends upon number of POPs n and total number of flows N directed towards victim server in protected ISP domain. The number of flows N observed by an ISP is very high as compared to number of POPs present in the ISP. Moreover, packet headers spoofing [19] in popular DDoS attack tools result in a lot of flows aggregated near the victim side. So, equation (4.2) can be reduced to:

$$\text{Time complexity} = O(N) \text{ as } N >> n$$

A two dimensional array $(n \times 2)$ to store POP number and entropies computed at POPs is used to compute final entropy E_f at POP P_s using equation (4.1). However $(N_i \times 2)$ sized arrays are required at different POPs to compute individual entropies E_i. Moreover, an additional communication overhead of n packets to notify POP P_s about E_i and S_i computed at all ingress POPs is incured after each time interval of Δ seconds.

It is well known fact that the number of POPs is far less than total number of flows encountered in an ISP domain. Therefore, proposed distributed detection has very less computional overheads at POP P_s, and all ingress POPs share computational overheads.

Though, communication overheads to send sample entropies E_i to POP P_s for final sample entropy E_f computation are extra in the distributed approach.

4.7 Discussion

It is found that real time detection of flooding DDoS attacks can be accomplished by distributed monitoring and analysis at ingress POPs of an ISP domain. The computed sample entropies (E_i) at ingress POPs of the ISP domain are sent to POP P_s where final sample entropy E_f is computed by using equation (4.1). The detection is performed at POP P_s by comparing E_f with already profiled normal sample entropy using equation (3.2). The time and space complexities of of single point and distributed approach are summarized in table 4.4. Here n is number of POPs and N is total number of flows directed to the victim server in protected ISP domain 4.

Table 4.4: Summary of computational complexities

Approach	Computational complexity at POP P_s		Overall computational complexity of all ingress POPs P_i of ISP $\{i = 1,2 \ldots n\}$		Overall complexity i.e. sum of complexities at POP P_s and all ingress POPs P_i of ISP	
	Time	Space	Time	Space	Time	Space
Single point	$O(N)$	$O(N)$	-------	--------	$O(N)$	$O(N)$
Distributed	$O(n)$	$O(n)$	$O(N)$	$O(N)$	$O(n)+O(N)$ $\Rightarrow O(N)$ as $N >> n$	$O(n)+O(N)$ $\Rightarrow O(N)$ as $N >> n$

The conflicting requirements mentioned in section 4.3 are satisfied by the detection approach as detection is performed by taking whole of traffic into consideration at POP P_s and computational overheads are also vey less at the POP.

The time complexity of single point detection approach and distributed approach is $O(N)$. It means by employing distributed approach, time complexity has not varied much as compared to single point approach in D-DCFI. However, the time complexity is $O(n)$ for distributed approach at POP P_s, whereas it is $O(N)$ for single point approach in D-DCFI at POP P_s. Similarly, space complexity of distributed approach, which is sum of space complexity, at ingress POPs of the ISP domain to compute E_i and at POP P_s to compute final entropy E_f, are also almost same as single point approach. In other words, computational complexity of distributed approach are approximately equal to single point detection approach used in D-DCFI. But whole of computational complexity is centred at POP P_s in single point detection approach in D-DCFI, whereas it is distributed among all POPs of the ISP in proposed distributed approach. Therefore, it is concluded here that in distributed approach, computational complexity i.e. time and space requirements for monitoring and analyzing traffic at POP P_s is very less. All ingress POPs take the load of monitoring and analysis for the traffic passing through them towards victim instead of single POP P_s. Thus, overall burden of state monitoring and analysis is distributed among ingress POPs of the protected ISP domain. POP P_s where whole of traffic converges, is almost relived from computational burden.

High rate flooding DDoS (HRFD) attacks generate large volume of traffic from zombies distributed across the Internet. State monitoring based solutions e.g. RealSecure [73], WATCHER [99], and TDSAM [173] need to maintain tremendous states to determine whether or not a packet is malicious are vuleranable against HRFD based attacks. Volume based schemes [18, 104] located in victim side though easily detect these attacks by measuring traffic levels (i.e. bytes or packets per second). But the required response to filter

or rate limit attack traffic is highly ineffective as high volume of attack packets either seize whole of bottleneck bandwidth or processing capacity thresholds of defense systems are exceeded. As a result, manual intervention is required to relieve victim from these DDoS attacks. In the proposed approach, the point of defense, where whole of traffic aggregates (i.e. POP P_s) has very less computational complexity. So it is not vulnerable against HRFD attacks. On the other hand, low rate flooding DDoS (LRFD) attacks gracefully degrade victim's resources. At initial stage, even HRFD attacks generate less traffic, but attack traffic slowly grows as zombies increase their intensity of packet flooding. LRFD attacks and proactive detection of HRFD attacks is not accomplished by volume based techniques as traffic level does not exceed normal fluctuations. D-DCFI detects LRFD attacks with accuracy as shown in Fig. 3.13. The proposed approach is an extension of detection approach in D-DCFI, so it not only solves the problem of excessive overheads at traffic aggregation point for HRFD attacks but also detects LRFD attacks with accuracy. On the other hand, the communication overhead to inform sample entropy values to P_s is an extra burden in the distributed approach. Moreover it is worth mentioning here that, computational complexity at ingress points of the ISP can also be decreased by using stream sampling algorithms [24]. Lakhina et al. [12] have demonstrated that the bias due to sampling is lesser if we summarize traffic distributions using sample entropy as compared to volume based metrics (i.e. bytes or packets per second).

The equation (4.1) is based on the assumption that set of flows directed towards protected server at ingress POPs are mutually exclusive from each other. We intend to modify equation (4.1) to remove this assumption in future work. Moreover communication required to notify POP P_s about E_i, S_i computed at ingress POPs is also not secure in proposed detection approach. This is itself a vulnerability, which can cause DoS attacks. A secure communication framework for distributed approach is also included in future work.

4.8 Conclusion

We presented a solution to detect flooding DDOS attacks at point of aggregation of amassed traffic with minimum computational overheads. The complexity of monitoring and analyzing huge volume of flooding DDoS attacks is managed by distributing the tasks among ingress POPs of the protected ISP domain. The sample entropies computed at ingress POPs are combined at point of aggregation of traffic using our derived equation. The equation computes final sample entropy as if whole traffic is monitored at single POP connected to victim server. The computational complexities of single point and distributed approach show that the overall complexity of distributed approach is almost same as single point, though communication delay introduces detection latency in the proposed detection approach. Characterization of attack flows in a distributed manner and subsequently filtering malicious traffic near attack sources in an ISP domain are our near future goals. However the longer-term goal is to achieve the same in multiple ISP domains in presence of decentralized management of the Internet (in absence of centralized control on the Internet).

Chapter 5

Predicting Number of Zombies Using Regression and Correlation Analysis

A lot of study has been done to find relationship between variables using regression and correlation analysis. In this chapter an effort is made to predict number of zombies based on its relationship with deviation from detection threshold at the time of detecting the DDoS attacks.

5.1 Introduction

In the existing literature a significant attempt is made by Moore et al. [50, 51] to estimate number of spoofed addresses involved in a DDoS attack. They have used backscatter analysis for the estimation. This is an offline analysis based on unsolicited responses. In an anomaly based detection method, deviation from normal behaviour beyond a threshold marks beginning of an attack. This extent of deviation is normally not utilized. We propose to estimate number of zombies using this extent of deviation from detection threshold. A real time estimation of the number of zombies in DDoS scenario is helpful in narrowing down set of suspected zombies generated after characterization of attack flows is over. Predicted number of most suspicious sources can be chosen for filtering or rate limiting to suppress the effect of attack. It is assumed here that zombies have not used spoofed headers.

Sample entropy is used as a metric to detect flooding DDoS attacks using equation (3.2). In order to predict number of zombies z from deviation $(H_c - H_n)$ in sample entropy value, another set of simulation experiments are conducted. This time simulations experiments are done at same attack strength 20Mbps in total, but number of zombies are varied from 20-100. Accordingly, mean attack rate per zombie is varied from 0.2Mbps-1.0Mbps. The deviations in sample entropy values from normal for variation of five numbers of zombies are given in result and discussion section. This section also includes variation in sample entropy with 25, 50, 75, and 100 numbers of zombies at same attack strength in total of 20Mbps. It is found that there is strong relationship between deviation in sample entropy and number of non spoofed zombies involved in launching the attacks. In this chapter an attempt is made to find the relationship between number of zombies involved in a flooding DDoS attack and deviation in sample entropy value from normal value of sample entropy.

5.2 Regression and Correlation Analysis

Regression and correlation analysis [127] is used to determine relationship between two variables. Chi-Square tests of independence can also be used to determine whether statistical relationship exist between variables, but regression and correlation not only shows nature of dependence between two variables but also strength of relationship is determined numerically. The known variable (or variables) is called independent variable(s). The variable we are trying to predict is the dependent variable. The value of dependent variable can be estimated based on past observations of independent variables. Multiple regression is used to describe a process by which several variables are used to predict another.

In regression analysis, we develop an estimating equation i.e. a mathematical formula that relates the known variables to the unknown variable. Then, after we apply correlation

analysis to determine the degree to which the variables are related. Correlation analysis, then give us how well the estimating equation actually describes the relationship. We examine linear relationship between number of zombies y' and observed deviation in sample entropy $(H_c - H_n)$.

A straight line of the form $y' = mx + c$ for the relation between predicted number of zombies and entropy deviation is tried for the relationship. The results are promising as we are able to fit in straight line for the required relationship and errors are also very less. Though, in this case simulation experiments have been used but real time experiments can also be done to establish this kind of relationship.

5.3 Simulation Setup

The topology in Figure 3.3 and simulation parameters discussed in chapter 3 are used again in this work. However, the simulation experiments are done in a different manner i.e. earlier number of zombie were kept constant, but in this case attack strength is kept same and number of zombies is varied. The mean attack rate per zombie is 1.0Mbps for the case of 20 zombies and 0.2Mbps for the case of 100 zombies. So total attack strength is 20Mbps. Similarly for 20-100 zombies, mean attack rate is decided.

5.4 Results and Discussion

As soon as any event in equation 3.2 triggers, attack is said to have occurred. Figure 3.13 shows sample entropy profile when our network is put under low rate attack. In this case attack is launched with 100 zombies with mean rate 0.3Mbps per zombie. Clearly in first time window after attack is launched at 20 seconds, there is jump in sample entropy value. The positive jump and persistent high value as compared to normal reflects that it is a low

rate attack and the flows which are causing this anomaly have comparatively lesser number of packets than already existing ones.

In order to predict number of zombies (y') from deviation ($H_c - H_n$) in sample entropy value, another set of simulation experiments are conducted as discussed in simulation setup. The deviations in sample entropy values from normal for variation of 5 numbers of zombies are given in table 5.1. Figure 5.1 shows variation in sample entropy with 25, 50, 75, and 100 numbers of zombies at same attack strength in total of 20Mbps.

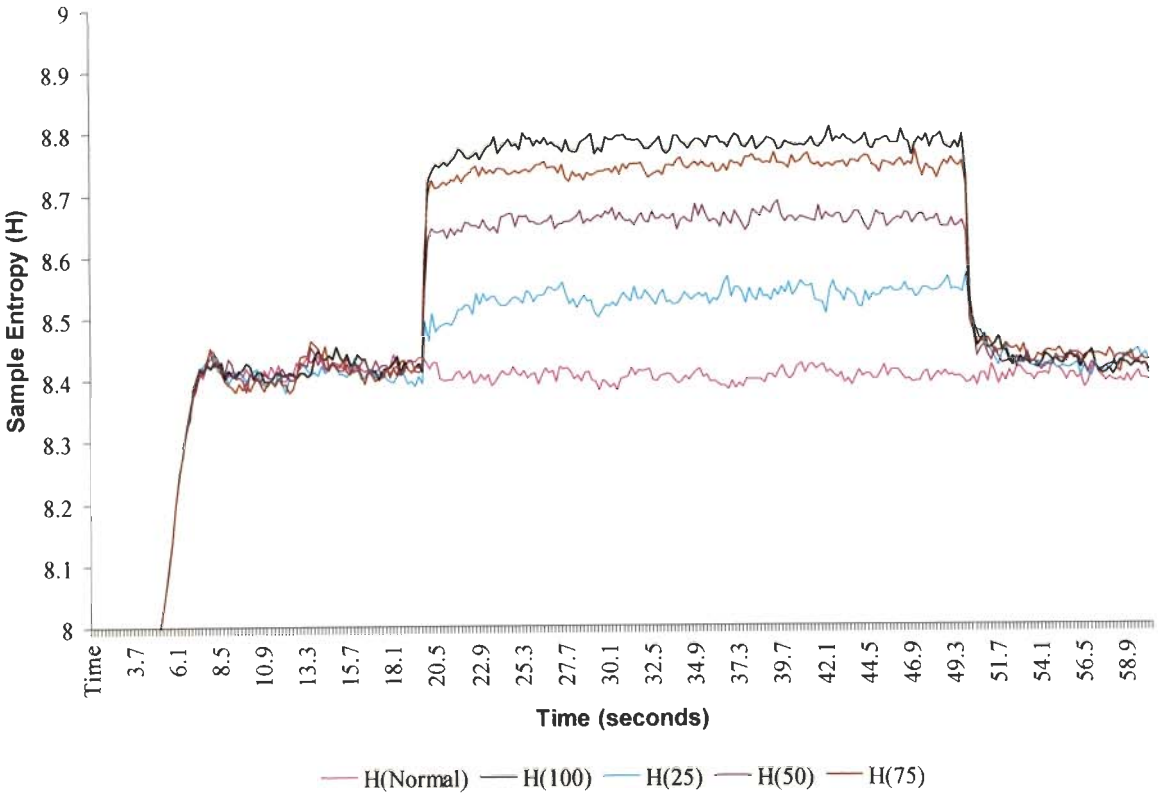

Figure 5.1 Variation in sample entropy with number of zombies

Using regression analysis, we tried to fit in straight line $y' = mx + c$ for the relation between predicted number of zombies and sample entropy deviation. Here y' is predicted number of zombies and x is sample entropy deviation $(H_c - H_n)$. m and c are constants to be

computed by regression analysis. Let y be the corresponding observed value for observed value of x.The table 5.1 gives some of observed values of y and x.

Table 5.1: Deviation in sample entropy with actual number of zombies

Actual Number of zombies (y)	**Deviation in Entropy** $(H_c - H_n)$
20	0.062871
25	0.117888
30	0.148609
35	0.180394
40	0.194598
45	0.227069
50	0.241797
55	0.264329
60	0.271391
65	0.292619
70	0.308269
75	0.324733
80	0.328367
85	0.348394
90	0.356384
95	0.36808
100	0.38807

Regression analysis is applied on all observed values of y and x to fit in a straight line. We are able to fit in straight line as follows:-

$$y' = 263.274(H_c - H_n) - 8.5111 \qquad (5.1)$$

The reliability of regression line in equation (5.1) also needs to be checked and the overall trend should be justified. To measure the reliability of the estimation equation, statisticians have developed the standard error of estimate S_e. The value of S_e found in our case is 5.2966873. The standard error of estimate S_e, measures the variability, or scatter, of the observed values around the regression line. The larger the standard error of estimate, the greater is the scattering (or dispersion) of points around the regression line. As per statistics

theory, we can expect 68 percent of points within $\pm 1S_e$, 95.5 percent of points within $\pm 2S_e$ and 99.7 percent of points within $\pm 3S_e$.

The extent or strength of the association between two variables x and y is measured by coefficient of determination. Since, a sample of points is used to develop a regression line, the measure is referred as sample coefficient of determination R^2. Value R^2 only measures strength of a linear relationship between variables. The R^2 value for regression line in equation (5.1) is 0.9587428. It indicates that variation in y values with change in x values is reflected well by the regression line in equation (5.1).

The coefficient of correlation is the second measure that is used to describe how well one variable is explained by another. When samples are dealt, the sample coefficient of correlation denoted by R is used and is the square root of the sample coefficient of determination [127]:

$$R = sqrt(R^2) \tag{5.2}$$

The value of R for regression line in equation (5.1) as per equation (5.2) is ± 0.979154. As slope of regression line in equation (5.1) is positive so value of R is 0.979154. Thus, sign of R indicates the direction of relationship between two variables x and y i.e. as x increases y also increases.

The constant in equation (5.1) depend upon network environment. In actual experimentation by using our approach, the expected number of zombies can be predicted in real time. Using equation (5.1), the number of zombies is computed and is compared with actual number of zombies. The comparison is depicted in Figure 5.2.

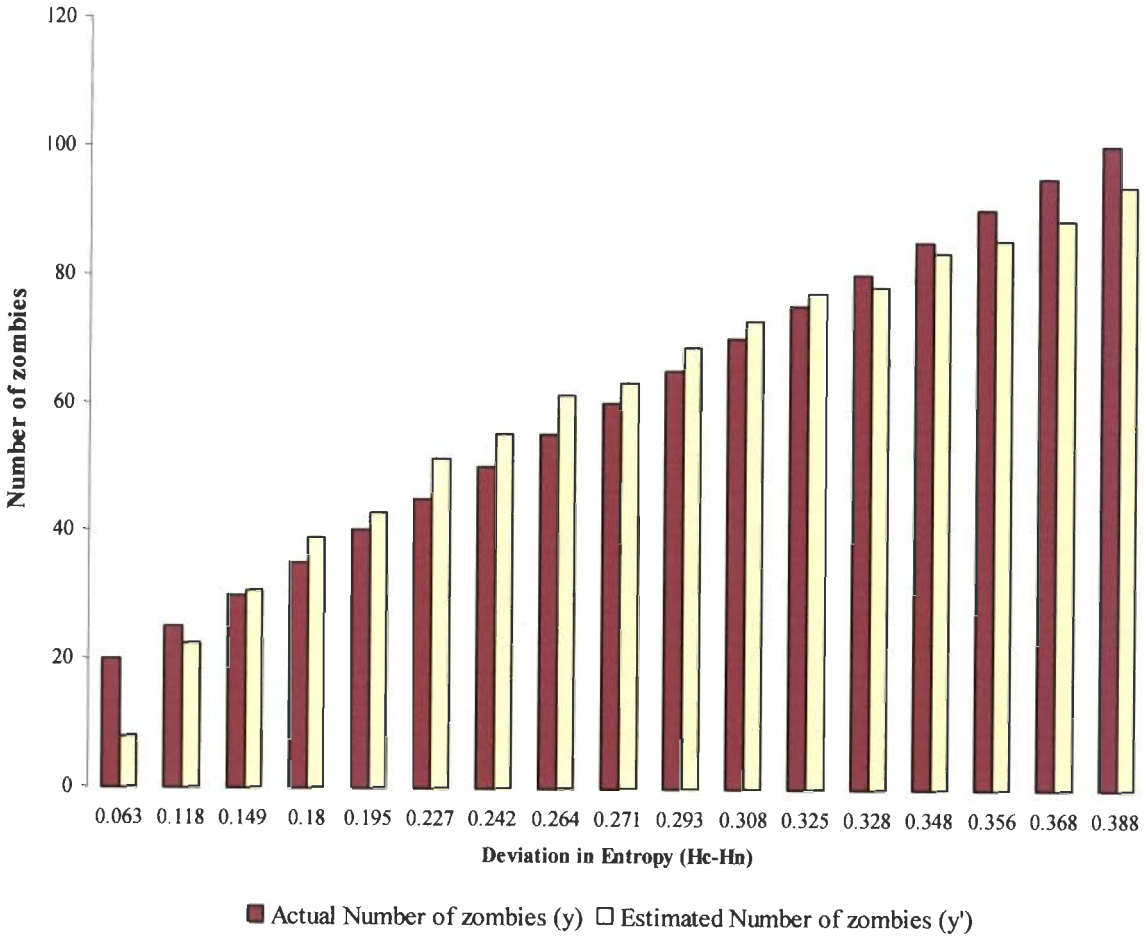

Figure 5.2: Comparison of estimated and actual number of zombies

5.5 Error

The error in relationship is shown in the form of falsely predicted flows as zombies (false positives) and zombie flows identified as legitimate (false negatives) in Figure 5.3. The value of standard error of estimate S_e (5.2966873) summarizes Figure 5.3.

Quite evidently, our estimation has some errors because it is not able to pick all legitimate flows and zombie flows in case of low rate attacks with the help of deviation in sample entropy. The error in relationship is due to the fact that zombies send attack packets at varying rates. Those zombies which overall contribute to increase the dispersion are

trapped but the flows which are very similar to legitimate flows are not identified in this case.

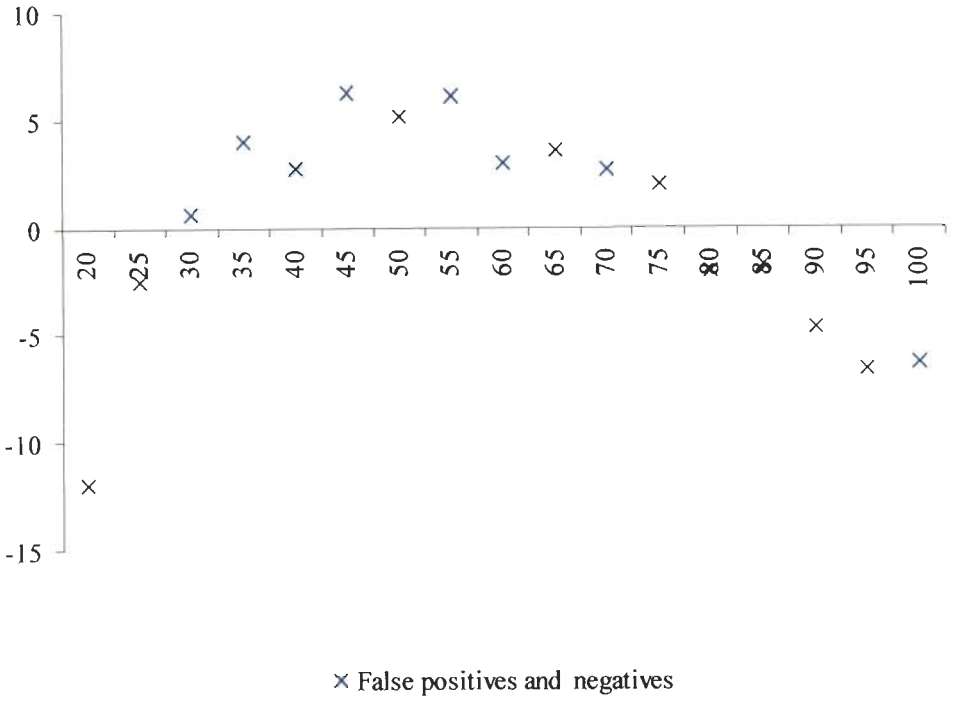

Figure 5.3: Error in regression line given in equation (5.1)

5.6 Conclusion

We detected DDoS attacks in ISP domain using sample entropy as traffic feature distribution metric. After properly setting the thresholds for normal entropy, extent of deviation in sample entropy gives strong signals for estimation of number of attack flows. Regression and correlation analysis proved a straight line relationship between number of zombies and deviation in sample entropy. The standard error of estimate S_e is 5.2966873 and value of sample coefficient of determination R^2 is 0.9587428 for the obtained regression line. It means that variation in y values with change in x values is represented well by the regression line. Moreover, value of sample coefficient of correlation R is

0.979154. It indicates that with rise in sample entropy deviation, number of estimated zombies increases. Attack strength can also be predicted from deviation in sample entropy value by using the same method if number of zombies is kept constant and attack strength is varied. Though, simulation experiments in NS-2 test bed are used for estimation of number of zombies but experimentation using a real time test bed can strongly validate our claim.

Chapter 6

Tolerating DDoS attacks using Dynamic Rate Limiting

Distributed Denial-of-Service (DDoS) attacks pose a serious threat to availability of Internet Services. Several schemes have been proposed for countering DDoS attacks directed at an Internet Server, but they suffer from a range of problems. Some of them are impractical and others are not effective against these attacks. Dynamic rate limiting techniques are proposed that minimizes the impact of attack. The basic mechanism relies on monitoring and rate limiting at edges of the protected ISP. The rate limiting function is triggered at participating edge routers after they get signal from the server under attack. Server instructs edge routers to rate limit the traffic according to the share of traffic that has recently passed through it.

6.1 Introduction

The goal of DDoS attacks is to completely tie up the resources of the victim server, which prevents legitimate users from accessing its services. DDoS attack is a resource management problem. The goal is to protect server from excessive service request arrivals over a global network in such a manner that minimum computational resources of the server should be wasted in handling the illegitimate requests. Ferry [180] can handle more number of incoming requests, but still volume of DDoS attacks is such that even for these architectures it is not easy to withstand the high traffic load.

Existing DDoS solutions are classified into four broad categories in [22]: Prevention, Detection and characterization, Traceback, and Tolerance and mitigation. Tolerance and mitigation aims to eliminate or curtail the effects of an attack and try to maximize the quality of services under attack. Active queue management [152], load balancer [58], resource accounting [78], proactive server roaming [30, 146], throttling [46], and pushback [84, 129] are some of the techniques in the realm of tolerance. The proposed approach aims to sustain the services of the protected server under high bandwidth DDoS attacks using tolerance based scheme. In this work the impact of DDoS attacks is minimized by dynamically rate limiting of incoming traffic at edges, based on share of traffic per edge router, and per flow.

A defense approach similar to dynamic rate limiting has been proposed in [46]. Yau et al. [46] identifies DDoS attacks based on traffic level at the server. It calculates rate limits and sends control signals to participating router located at K hops away from the server. These control signals contain throttle values to be used by the participating routers so as to limit traffic directed towards the protected server. The main advantage of dynamic rate limiting as compared to approach in [46] is that server sends different throttle values for all routers. Moreover, each router's throttle value has been calculated according to the share of traffic, it is contributing to the total attack traffic. Also this scheme has been enhanced by going to the depth of flows passing through each router, whereas same throttle values are used for all routers in [46]. Per router throttle scheme is comparable to [46], whereas per flow scheme surpasses, though at the cost of increased overheads. A general framework for identifying and controlling high bandwidth aggregates in a network is described in [129]. In order to protect good user traffic from attack traffic destined for the victim server, a recursive pushback starting from the victim server is carried out up to upstream routers.

Pushback mechanism always starts at the point of congestion near the server under attack. At this point good user traffic is perfectly blended with attack traffic and thus can be severely punished.

An ISP level solution has been proposed in [139] and in [158]. In [139] perimeter based defense mechanisms, which provide anti-DDoS services to its customers are discussed. Solution lacks effectiveness in case of attack within the ISP. In [158] protection is limited within the domain of single ISP though later they extended it in multiple domains [157]. Also the efficiency of the scheme decreases as number of edge routers increases in the same ISP. Following contributions are made in this chapter.

- An Internet type topology is generated using GT-ITM topology generator.
- A dynamic rate limiting algorithm is presented that can effectively protect a server from resource overloading and minimize collateral damage at the same time.
- A scheme is presented that not only defends victim server from high rate flooding DDoS attacks but also take care of the ISP's core bandwidth.
- Multicasting is used to send control messages.
- This is an ISP level solution where incremental deployment to other ISPs is also possible.

The rest of the chapter is organized as follows. In section 6.2 we discuss the dynamic rate throttling algorithms. Simulation and results are discussed in section 6.3. Section 6.4 finally concludes the chapter.

6.2 Dynamic Rate Limiting

The basic principle in this scheme is that the router which is contributing more in total traffic should be punished more. Dynamic rate limiting (DRL) throttles the traffic at edge routers of the protected ISP domain in proportion to traffic passing through it i.e. fraction of

traffic observed at any edge router actually determines rate limit value for that particular edge router. Three levels of traffic have been considered:

- Normal Level (L_N): It is the level of traffic that the server can handle or serve smoothly.
- Congestion Level (L_C): It is the traffic level at which server becomes suspicious of attack and start calculating the rate limiting function.
- Attack level (L_A): It is the traffic level at which monitoring routers start rate limiting according to the function sent to them by victim server.

The dynamic rate Limiting (DRL) algorithm in general can be considered of following steps.

1. Check the traffic level, if it is less than L_N then go to step 2 else go to step 3.
2. If traffic level is less than L_N then increase the traffic by a factor β additively till it becomes >= L_N and then go to step 1.
3. If traffic level is greater than L_A then send a control message to router for rate limiting the traffic by half till traffic doesn't becomes less then L_N.
4. Request edge routers to start marking the packets with the router ID and compute the share of each router's traffic in total traffic.
5. A function at server calculates to what degree an edge router should limit the traffic (or of each flow in case of per flow throttling) passing through it.
6. A feedback from edge routers at regular intervals about traffic drop rate per router and per flow is sent back to server for fresh calculation of rate limit values. Server decides whether it has to keep up rate limiting at routers or not? If yes go to step 5 else go to step 1.

6.2.1 Terminology

As explained above in terms of traffic experienced the server fixes three limits namely Normal Level (L_N), Congestion Level (L_C), and Attack Level (L_A). There are four states in which a system moves depending upon the traffic values. These states are Normal State (NS), Ready State (RS), Action State (AS) and Rigorous State (RGS). The flow state diagram for DRT mechanism in Figure 6.1 shows the relation between states and traffic value being experienced by the server (symbolized by symbols *X, Y, Z,* and *W).*

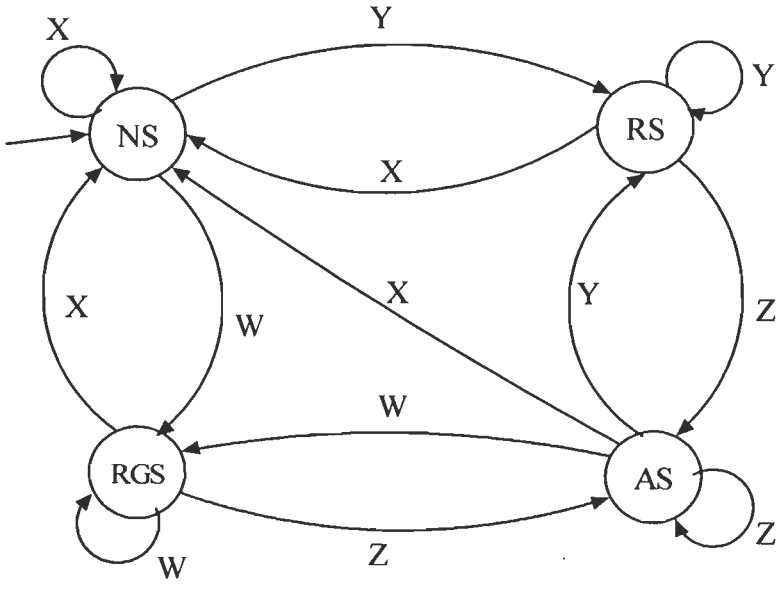

Figure 6.1: Flow state diagram for DRL mechanism

Table 6.1 shows the value of *X, Y, Z* and *W* symbols in terms of traffic at server and corresponding action being taken in the mechanism, where ρ is the amount of traffic being experienced by the server S in time window t.

The observed traffic less than or equal to L_N signifies that server is not under attack and network is in normal state. Traffic above L_N and less then L_C represents a suspicious state as server resources are consumed more as compared to normal state. The traffic level above L_C and less then L_A confirms a DDoS attack on the server.

Table 6.1: Value of *X, Y, Z* and *W* variables and corresponding action

Variables	Traffic Value	Action
X	$\rho <= L_N$	Do nothing and stop throttling if already applied
Y	$L_N < \rho <= L_C$	Send Message to Participating Routers to mark with router ID the traffic being converging from them at server end.
Z	$L_C < \rho <= L_A$	Server sends the rate throttling values to edge routers and subsequently routers start throttling.
W	$\rho > L_A$	Cut half of the traffic at every edge router.

The difference between states when traffic is between L_N and L_C (represented by symbol *Y* in Table 6.1) and between L_C and L_A (represented by symbol *Z* in Table 6.1) is that, former can still serve the packets but it has reached its optimum resource level in process of serving incoming request packets. In the later one server is unable to serve all the incoming legitimate packets since the resource limit has already been crossed. The value above L_A signifies that server is under severe attack it needs a rigorous (or harsh) rate limiting to curb aggressive traffic.

In Figure 6.1, NS which is symbolizing Normal State (NS) is also the initial state. Other states are Ready State (RS), Action state (AS) and Rigorous state (RGS). As the value of traffic at server changes the state also changes. An amount of total traffic converged at the server is represented by *X, Y, Z* and *W* variables.

Initially the system is in NS State. If the value of the traffic remains *X* it will remain in NS state. Once the traffic value changes to *Y,* it goes into RS state. In this state, server sends

control message to monitoring routers to start marking the packets passing through them. The server calculates each router's traffic share in total traffic in RS state. If the value of traffic is *Y* it remains in RS state and goes back to NS state if the value becomes *X*. In NS state throttling is also stopped.

Once the traffic crosses from *Y* and takes the value of *Z*, the state is changed to AS. Server sends rate limit values for each router (or for each flow) calculated through a function *f(x,y)*, where x is router's share and *y* is the feedback value. This feedback value *y* depends upon the total traffic experienced by an edge router. The value is sent to protected server for computation of rate limit value for the edge router. Routers at their end rate limit traffic as per rate limit values sent to them by the server. These edge routers continuously send feedback to server. They remain in same state till the value of traffic remains *Z*. If the value of traffic reduces to *Y*, the state of the system moves back to RS. Rate limiting is stopped but packet marking goes on till traffic reduces to *X* which represents NS state.

If the traffic changes to *W*, it moves to RGS state and it remains in the same state till the value is *W*. In this state server sends the control message to all monitoring routers to start rate limiting the traffic with throttle value twice of present rate limiting value. This process of increase in rate limit continues till the traffic doesn't come under *Y* or *X*. It moves back from RGS to AS if the traffic reduces to *Z*.

If the traffic reduces to *X*, it moves to NS. If the traffic suddenly crosses the value L_A, system moves directly from NS to RGS state bypassing all the states. The edge routers are then instructed to start rate limiting the traffic and this value keep on increasing till the traffic doesn't come under L_N or L_C. Also at any stage if traffic reduces to less than L_N then server sends control message to edge routers for additive increase in allowing the traffic. This process continue till the traffic crosses L_N.

6.2.2 Identifying Responsible Flows

This section covers the details about the two algorithms namely Per Router and Per Flow rate limiting. Two algorithms differ in strategy of applying the rate limiting scheme. These are explained below.

6.2.2.1 Per Router Throttling

Initially the server under attack samples the traffic converging to it. The traffic emerges through different edge routers. The server raises suspicion about high rate attack depending upon level of traffic being received per unit time. When traffic level crosses L_N, it triggers the DRL mechanism by sending the signal to monitoring routers for marking the packets passing through them. It then starts calculating each router's traffic contribution in total traffic. It keeps on doing the same till traffic level is less then L_C. Once the traffic crosses L_C, server sends different rate limiting values (depending upon their contribution) to edge routers. The traffic level beyond L_A triggers half rate throttling as per table 6.1.

After each router I gets its due rate limit values L_{RI} , which is sent to it by the server , each router starts rate limiting the traffic accordingly. Routers keep track of the traffic being experienced at their end and send this feedback to the victim server. Before sending the control signal for rate limiting again, server do considers the feedback sent to it from routers and accordingly takes decision for continuing the rate limiting or to stop it.

Let TR_I be the traffic being sent to the server from router I in time window t, so TR which is total traffic converging at server from different routers can be shown in equation (6.1).

$$TR = \sum TR_I \tag{6.1}$$

Where $I = 1 \ldots n$ for all routers

$$\phi_I = TR_I / TR \tag{6.2}$$

Where Φ_I signifies the rate limit for the router R_I in equation (6.2). Let router R_I is still dropping the packet at D_I rate. So this new information is given back to server. Let at any time window server experiences traffic rate of TR_I from router R_I and let D_I be the drop rate of router being sent to server from router I. Server chooses a value according to function $f(x,y)$, which suggests the server whether to send back a control signal for continuing the router throttling signal or to stop it for future. This can be shown by equation (6.3).

$$\phi_{IN} = f(D_I, \phi_I) \tag{6.3}$$

Where Φ_{IN} on left side of equation (6.3) is the new control signal for router, which is to be sent next time. On the right side of the equation, the second parameter (Φ_I) of function is the old value for router throttling rate and D_I is the feedback sent to server from the router I about drop rate. In simple terms it can be shown that new rate limiting value, Q_{IN} is represented by equation (6.4) as follows.

$$Q_{IN} = (D_I + TR_I)/TR \tag{6.4}$$

This process continues till the traffic does not come under the level of L_C or L_N. If the value becomes less than L_N then server sends an additive control value β to router for increasing the traffic till it crosses L_N.

6.2.2.2 Per Flow Throttling

Per Router rate limiting has some disadvantages associated with it. The throttling rate sent by the victim server to the router I, is commonly applied for all the flows passing through the router R_I. All the flows passing through a router irrespective of whether it is an attack flow or a genuine traffic get scrutinized by same value. It considers the attacker and genuine flows in same category. Thus, high false positive rate may result if the good and attack traffic is passing through same router.

Per flow rate limiting deals with per flow (instead of per router) rate limiting for different flows passing through different routers. As in per router rate limiting, in this scheme also victim server samples the traffic flows converging to it. It adds up attack traffic of all the flows and checks level of traffic consistently. Let TF_{JI} be the traffic contribution of flow J coming from router I and TF is total traffic of all the flows coming from different routers in a time window t. The relation between TF_{JI} and TF can be shown by equation (6.5).

$$TF = \sum TF_{JI} \tag{6.5}$$

Where $I = 1 \ldots n$ are routers and $J = 1 \ldots k$ are flows under router I

$$\phi_{JI} = TF_{JI} / TF \tag{6.6}$$

The traffic limit for per flow can be calculated by equation (6.6) where ϕ_{JI} signify the rate limit for flow J in router I. For detecting the packet coming from different routers one can use source IP, source port, Destination IP, destination port to categorize it under a particular flow. Now in this case server sends a rate limit value back to routers. The calculation of the hash function has been explained with the help of following example. Please refer to Figure 6.2 for better understanding of the algorithm. Let S is the victim server and *R1* and *R2* are two routers. Flows *f1*, *f2* and *f3* are passing through *R1* and flows *f4*, *f5* and *f6* are passing through router *R2*. *f1*, *f3* and *f5* are attack flows and *f2*, *f4* and *f6* are genuine flows.

Now every flow converging at victim server is analyzed by the server by creating a data base as shown in Table 6.2. This database has fields for Flow ID, Router ID, and categorization field (for categorizing the flow as attack flow or genuine flow). Now depending upon this data base, server creates a hash function for distinguishing between

attacks and genuine flows. Subsequently, the computed values are sent to different routers for rate limiting the flows passing through them.

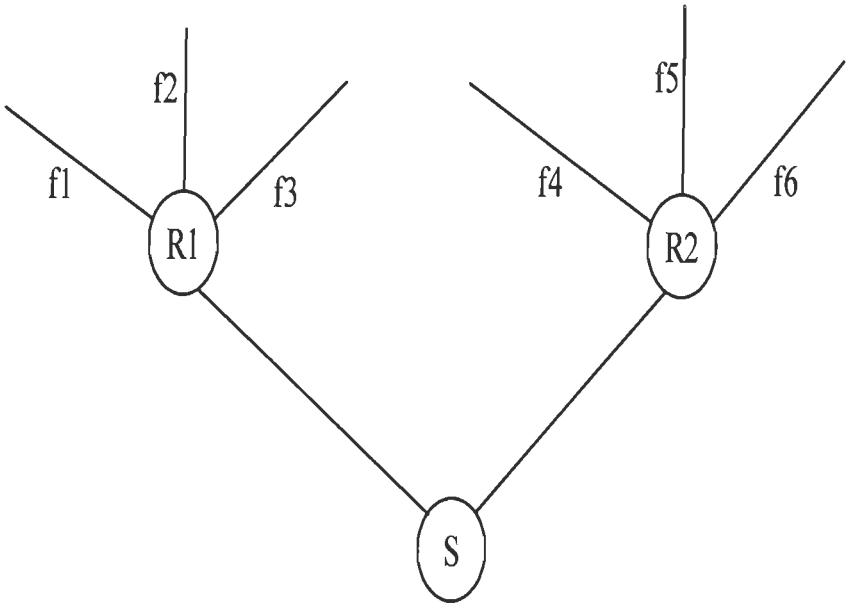

Figure 6.2: Flow based representation of traffic to victim server

Table 6.2: Database created at server end for making a hash function

Flow ID	Router ID	Attack/Genuine
F1	R1	Attacker
F2	R1	Genuine
F3	R1	Attacker
F4	R2	Genuine
F5	R2	Attacker
F6	R2	Genuine

Hash function h(IP) has Source IP, Source port, Destination IP, Destination port address as inputs. It classifies flow either as attack or legitimate flow based on number of individual packets of the flow in monitored time window. If the level of traffic is in between L_C and L_A, then a rate limit of α can be suggested by the server depending upon the contribution in total traffic. If the level of traffic at server end is higher then L_A then routers can be requested to drop all the packets belonging to attack flows.

Regular feedbacks are also sent by routers to server for the traffic they are experiencing at their end for each flow after applying rate throttling using hash functions. In this scenario each router sends the detail drop rate of each flow passing through it. Let router R_I is still dropping the packet at DF_{JI} rate for flow J. So this new information is given back to server again. Server chooses a value according to function f(x,y), which suggests the server whether to send back a control signal for continuing the rate limiting signal or to stop it (for per flow separately) for future. This is shown in equation (6.7) as below

$$\phi_{JI} = f(DF_{JI}, \phi_{JI}) \tag{6.7}$$

In simple terms it can be shown that new rate limiting values for each flow can be represented by equation (6.8) as follows.

$$Q_{JI} = (DF_{JI} + TF_{JI}) / TF \tag{6.8}$$

This process goes on till state of system becomes NS. If the traffic level becomes less then L_N then server sends an additive control value β to routers for increasing the traffic till it crosses L_N.

6.3 Simulations

The simulation model consists of several components such as a victim server connected to backbone consisting of interconnected core routers. Edge routers are connected at one end to core routers in the backbone and to other end they are connected to legitimate and attacking hosts. Though bandwidth in the core is sufficient but for the purpose of simulations instead of taking processing capacity of server as bottleneck, link bandwidth between server and access router is taken as 10Mbps. The total traffic generated from legitimate clients after slow start phase finishes is 10Mbps whereas total 9Mbps of attack traffic is generated from 2 to 4 seconds in simulation time of 6 seconds. Simulation topology, mechanism and results are discussed in the next subsections.

6.3.1 Topology

The topology is generated using GT-ITM [65] topology generator. The topology contains one ISP domain consisting of five core and five edge routers. Both core and edge routers are represented in Figure 6.3 by circles.

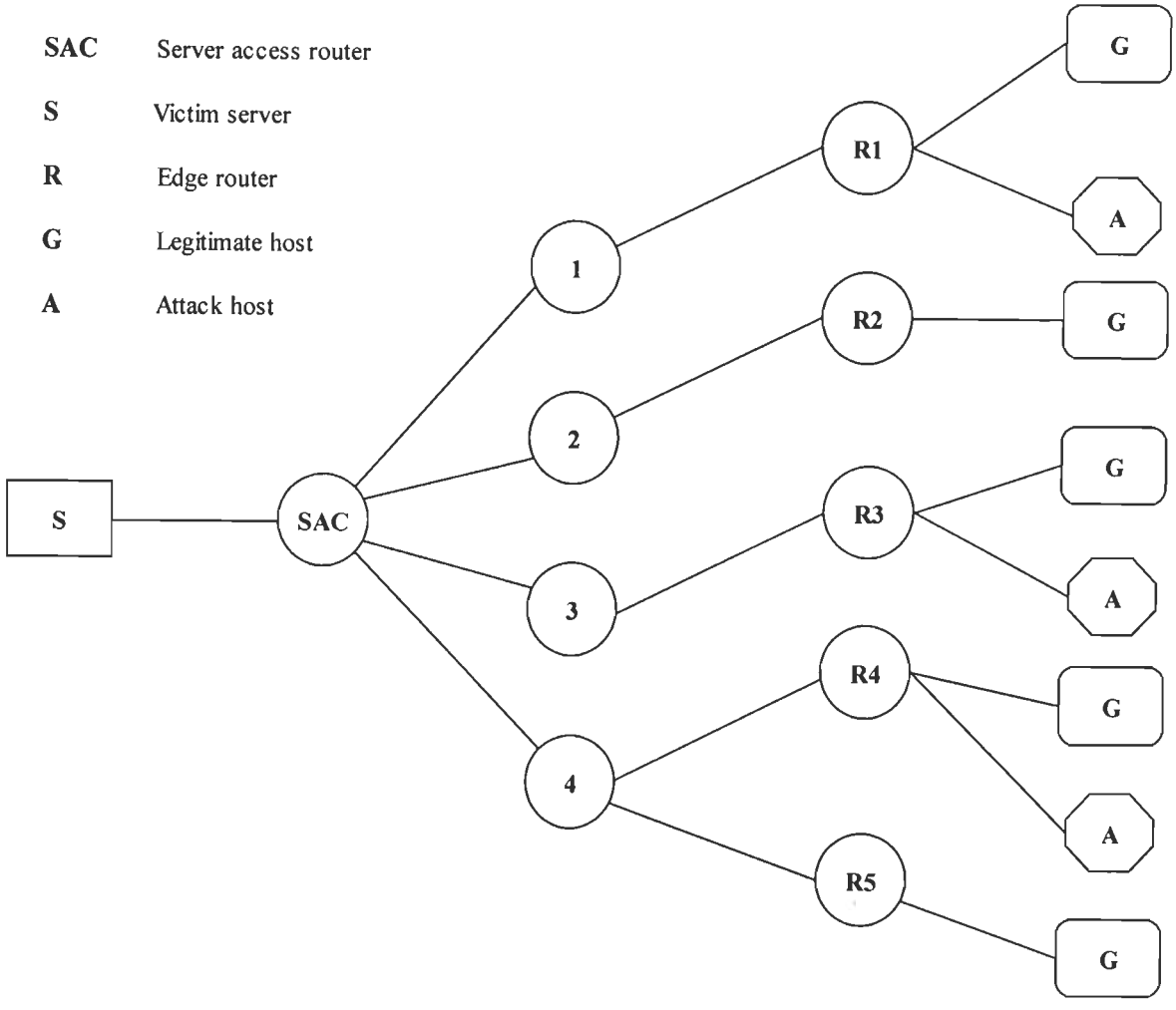

Figure 6.3: Simulation Topology

The edge routers are connected to customer domains comprising of legitimate and attack hosts. Group of legitimate hosts are represented as rounded rectangles and octagon is used for group of attack hosts. The server represented by rectangle is connected to backbone through server access router C which is further connected to four core routers. These core routers in turn are connected to edge routers. The topology used in our simulation, is shown

in Figure 6.3. It shows a victim server *S* connected with one core router *C*. Routers *R1, R2 R3, R4* and *R5* represent five participating/monitoring routers. *G* indicates nodes sending genuine traffic and *A* represents attack nodes. The simulations are performed using Network NS-2 Simulator [114] on Linux Operating System.

6.3.2 Mechanism

The traffic is generated from nodes represented by G and A, and it converges at S. The server keeps track of the total traffic being received. When the traffic increases by limit L_N server sends a control signal to monitoring routers, requesting for to start marking the packets passing through them with a router ID. Router ID is generated randomly at server similar to agent IDs produced by controller in [158] and is sent to monitoring routers. The server calculates the amount of share of each traffic flow (as required in two different algorithms) and accordingly sends a different dynamic rate limiting values depending upon their respective shares to all the routers. Multicasting is used for sending control messages and rate throttling values.

6.3.3 Results and Discussion

Figure 6.4 shows the comparison between different throttling schemes. All the schemes show same results during initial phase i.e. till the attack is detected. Once the attack is detected proposed schemes surpass the static throttling schemes. Both of schemes are compared with each other. Per flow throttling performs better than per router but at the cost of increased overheads. In case of throttling done on per flow, the ratio of false positive values is very low as compared to per router throttling. Per router based throttling does not consider individual flows thus both legitimate and attack flows are punished. It makes no distinction between the legitimate and attack flows passing through same router. The difference between two schemes is that Per flow requires more processing time at server

side thus decreasing its efficiency and Per router is faster but throttles genuine packets as well thus leading to more false positives. The static router throttling scheme shows more deterioration in throughput since it does not takes into account per router attack traffic contribution. It throttles the traffic coming from different router with same value.

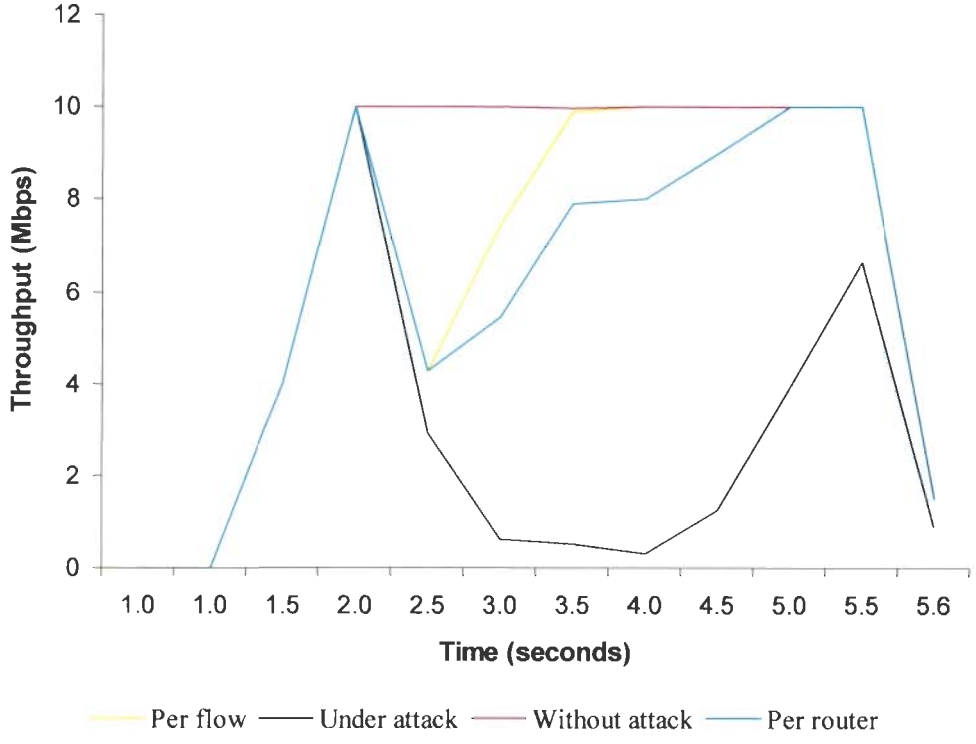

Figure 6.4: Comparison of different throttling schemes

The testing of two algorithms is done on a small topology consisting of few legitimate and attack clients. Once slow start phase is over, legitimate traffic occupies whole of bottleneck bandwidth as shown in Figure 6.4 without attack. The attack period is from 2 to 4 seconds. Clearly under attack without defense attack traffic seizes whole of bottleneck bandwidth as legitimate traffic follows congestion and flow control signals whereas attackers keep on sending traffic at the predefined rates. Per router scheme however throttle attack traffic as per share of each router in total traffic. At edge routers both legitimate and traffic is mixed so some part of legitimate traffic is also dropped by this scheme. In case of Per flow scheme, attack flows are identified and throttled at edge routers.

6.4 Conclusion

The scheme presented here is a server centric approach to protect it under attacks. Dynamic rate limiting schemes are used to achieve maximum legitimate traffic level at the victim. In per router dynamic rate limiting scheme, the edge routers are informed about rate limits that each edge router has to enforce as per its share in total traffic aggregated near the server. On the other hand, in case of per flow dynamic rate limiting, the edge routers are informed about rate limits of flows. Since it goes to the depth of flow level, the real culprits passing through edge routers towards the server are nabbed without increasing false positives.

The rate throttling scheme presented here protects ISPs from DDoS attacks. The main focus in this chapter has been on aggressive DDoS attacks, in which attackers try to overwhelm a victim server by directing an excessive volume of traffic to the server. By performing dynamic rate limiting at edge routers it is successful in limiting the traffic by a sufficient rate and thus enabling victim server to provide services to legitimate clients during high traffic.

Chapter 7

Conclusions and Scope for Future Work

7.1 Conclusions

In this thesis, various approaches are proposed for protection from Distributed denial-of-service (DDoS) attacks. The major findings of this work are summarized as follows:

1. The proposed approach to detect flooding DDoS attacks and filter attack traffic at ingress edges of the protected ISP domain generates an automated response against these attacks. Due to distributed functionality and better infrastructure to handle DDoS traffic at one hop upstream from victim network in ISP domain, our approach is successful in reacting to flooding DDoS attacks without manual intervention. The infrastructure however improves with tier level of the ISP.

2. ISP level approaches proposed in chapter 3 and 4, do not require whole Internet infrastructure to be modified and are independent in the sense that they can work in isolation though cooperation improves the results. Moreover single administrative control is an advantage to make approach feasible in the large Internet scene. Transit-stub model of Internet topology generator is adopted for creating topology consisting of four ISPs domains. The victim server however is protected in single ISP domain (section 3.4 and 3.5).

3. Low rate flooding DDoS attacks, which slowly degrade services to legitimate clients, are detected reliably and accurately. Simulation experiments carried out at various attack strengths show detection of very meek rate attacks (section 3.6). High rate flooding DDoS attacks, which completely disrupt services to legitimate clients, are easily detected at POP near the victim in ISP domain. High rate attacks whose intensity per flow slowly rises are detected at an early stage in the proposed approach. It also provides proactive detection of high rate flooding DDoS attacks, which helps in timely recovery from attack.

4. Detection model based on sample entropy of traffic flow distributions has been defined. The choice of threshold as per different network environments and effect of tunable parameter variations on detection performance is presented. A systematic study of threshold setting as per network environment is discussed using ROC curves. This sets the guidelines for profiling normal behavior of any network environment (section 3.6). The trade off between detection rate and false positive rate has been highlighted which helps in effectively tuning the parameters of the detection algorithm in order to achieve specific requirements in terms of detection rate and false positive rate.

5. The dynamic filtering at ingress edges of the protected ISP not only distributes the look up and checking overheads but also saves the expensive core bandwidth. Normal packet survival ratio (NPSR) is used in the proposed approach (Refer to chapter 3 for details) to measure legitimate traffic level. The comparison with volume based technique (Refer to section 3.8) manifests supremacy of the proposed approach.

6. The complexity of monitoring and analyzing huge volume of flooding DDoS attacks is managed by distributing the tasks among all ingress POPs of the protected ISP domain. The equation (4.1) computes final sample entropy as if whole traffic is monitored at single POP connected to victim server. The computational complexity of our distributed scheme at POP connected to victim server is very less (Time complexity $O(n)$ and space complexity $O(n)$ where n is the number of POPs in protected ISP domain) as compared to existing schemes, which makes proposed approach robust against high volume and high computational overheads of monitoring and analsysing traffic near the victim. A summary of time and space complexities is provided in table 4.4. The validity of equation (4.1) has been further established by simulation based experiments.

7. The unutilized deviation of sample entropy from detection thresholds has association with number of zombies involved in flooding DDoS attacks. A regression and correlation analysis based on observed values of number of zombies used to launch the attack and sample entropy deviation, revealed a strong relation as given in equation (5.1). The standard error of estimate S_e is 5.2966873 and value of sample coefficient of determination R^2 is 0.9587428. The value of sample coefficient of correlation R is 0.979154 (chapter 5). It indicates that with rise in sample entropy deviation, number of estimated zombies increases.

8. Two algorithms Per router and Per flow are proposed, which try to allocate victim's resources in a fair manner to legitimate clients (chapter 6). The algorithms are successful in maintaining appropriate legitimate service level in case of highly aggressive attackers.

7.2 Scope for Future Work

This study opens up a number of avenues for future work. A number of research issues need to be addressed. Some of them are as follows:

1. Though monitoring and analysis of traffic at flow level has improved proposed DDoS defense to a great extent but at the same time it induces high state monitoring overheads. A small observation window used in chapter 3 and further distribution of computational complexity at ingress POPs of ISP domain in chapter 4, has greatly reduced state monitoring overheads at any single point of defense. But traffic sampling without inducing bias and fast monitoring adapters can handle the traffic load in practical deployment of the defense. Better data structures like bloom filter can also be used to minimize storage complexity. Moreover, clustering of traffic flows in bins so that collateral damage does not increase much can really improve DDoS defense.

2. Better hashing and flow classification techniques would reduce packet handling overheads, thus enabling DDoS defense to handle higher packet rates in a better manner.

3. The DDoS defense proposed in chapter 3 does not identify ingress edges of specific malicious flows. A simple packet marking scheme can be used to characterize ingress edges responsible for specific malicious flows. This will greatly reduce lookup overheads for filtering malicious flows at ingress edges of the protected ISP domain though at the cost of packet marking overheads.

4. The equation (4.1) is based on the assumption that set of flows directed towards protected server at ingress POPs are mutually exclusive from each other. The

analytical solution to remove this assumption would be very beneficial to make proposed approach more practical and robust. Characterization of attack flows in a distributed manner and subsequently filtering malicious traffic near attack sources in an ISP domain are also our future goals.

5. Distributed defense systems proposed in chapter 3 and 4 are not able to protect the information to be exchanged from being intercepted by the hackers. Moreover, authentication of information sources like POP connected to server for defense proposed in chapter 3 and ingress POPs for distributed detection system proposed in chapter 4 is not done. Current security mechanisms such as IPSec, PKI, CA, Symmetric and Public key based authentication can be used to meet the requirements. The authentication, confidentiality and integrity of information and its sources is also an issue in tolerance based approach proposed in chapter 6.

6. Currently proposed DDoS defense approaches are limited in single ISP domain. But they can be extended in multiple ISP domains using perimeter based approach or controller agent model discussed in chapter 2.

7. Simulation experiments in NS-2 test bed are used for estimation of number of zombies in chapter 5, but investigation using real time test beds or real attack traces will be more useful.

The remaining open issues do not pertain exclusively to our work but to DDoS defense in general:

1. The longer-term issue for defense against flooding DDoS attacks is to find technical and economic models to achieve cooperation between ISP to combat DDoS attacks collaboratively.

2. The availability of user friendly attack tools and their source codes provide flexibility to attackers to create a variety of new attacks by error and trial. It is almost impossible to predict all attack variations and design defenses that will work for all cases.

Bibliography

[1] A. Belenky and N. Ansari, "IP Traceback with Deterministic Packet Marking," *IEEE Communication Letters*, Vol. 7, No. 4, pp. 162-164, April 2003.

[2] A. Belenky and N. Ansari, "On IP Traceback," *IEEE Communication Magzine*, Vol. 41, No. 7, pp. 142-153, July 2003.

[3] A. Belenky and N. Ansari, "Tracing Multiple Attackers with Deterministic Packet Marking (DPM)," *In Proceedings of IEEE Pacific Rim Conference on Communication, Computers and Signal Processing*, Vol. 1, pp. 49-52, Victoria, BC, Canada, Aug. 2003.

[4] A. C. Snoeren, "Single Packet IP Traceback," *IEEE/ACM Transactions on Networking*, Vol. 10, No. 6, pp.721-734, Dec. 2002.

[5] A. C. Snoeren, C. Partridge, L. A. Sanchez, C. E. Jones, F. Tchakountio, S. T. Kent, and W. T. Strayer, "Hash-Based IP Traceback," *In Proceedings of ACM SIGCOMM*, pp. 3-14, August 2001.

[6] A. D. Keromytis, V. Misra, and D. Rubenstein, "SOS: An Architecture For Mitigating DDoS Attacks," *IEEE Journal on Selected Areas in Communication*, Vol. 22, No.1, pp. 176-188, 2004.

[7] A. Demers, S. Keshav, and S. Shenkar, "Analysis and simulation of a fair queuing algorithm," *Journal of Internetworking Research and Experience*, pp. 3-26, 1990.

[8] A. Garg and A. L. N. Reddy, "Mitigation of DoS attacks through QoS Regulation," *In Proceedings of 10_{th} International Workshop on Quality of service (IWQOS)*, pp. 45-53, 2002.

[9] A. Juels and J. Brainard, "Client puzzles: A cryptographic countermeasure against connection depletion attacks," *In Proceedings of the 1999 Networks and distributed system security symposium*, pp. 134-149, March 1999.

[10] A. Kulkarni, S. Bush, and S. Evans, "Detecting distributed denial of-service attacks using Kolmogorov complexity metrics," Technical Report 2001CRD176, GE Research & Development Center, 2001, URL http://www.crd.ge.com/ cooltechnologies/pdf/2001crd176.pdf.

[11] A. Lakhina, M. Crovella, and C. Diot, "Characterization of Network-Wide Anomalies in Traffic Flows," *In Proceedings of ACM SIGCOMM Internet Measurement Conference*, pp. 201-206, 2004.

[12] A. Lakhina, M. Crovella, and C. Diot, "Mining Anomalies Using Traffic Feature Distributions," *ACM SIGCOMM,* pp. 217-228, Philadelphia, Pennsylvania, USA, 2005.

[13] A. Lazarevic, L. Ertoz, V. Kumar, A. Ozgur, and J. Srivastava, "A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection," *In Proceedings of 3rd SIAM International Conference on Data Mining*, San Francisco, CA, May 2003.

[14] A. Mankin, D. Massey, C. Wu, S. F. Wu, and L. Zhang, "On Design and Evaluation of 'Intention-driven' ICMP Traceback," *In Proceedings of 10th International Conference on Computer Communications and Networks*, pp. 159-165, 2001.

[15] A. McCue, "Bookie reveals $100,000 cost of denial-of-service extortion attacks," June 2004. [Online]. Available: http://software.silicon.com/security/0,39024655, 39121278,00.htm, [last accessed Jan. 4, 2007].

[16] A. S. Sairam, G. Barua, "Effective Bandwidth Utilisation in Multihoming Networks," *In First International Conference on Communication System Software and Middleware (COMSWARE)*, pp. 1-8, Jan. 2006.

[17] Arbor Networks. At http://www.arbornetworks.com/.

[18] B. Bencsath and I. Vajda, "Protection against DDoS Attacks Based on Traffic Level Measurements," *In Proceedings of Western Simulation Multi Conference*, San Diego, pp. 22-28, California, *USA*, 2004.

[19] B. Haris and R. Hunt, "TCP/IP security threats and attack methods," *Computer Communications Review*, Vol. 22, No. 10, pp. 885-897, June 1999.

[20] B. Mukherjee, L. T. Heberlein, K. N. Levitt, "Network intrusion detection," *IEEE Network*, Vol. 8, No. 3, pp. 26-41, 1994.

[21] B. Pande, D. Gupta, D. Sanghi, and S. K. Jain, "The Network Monitoring Tool - PickPacket," *In Proceedings of International Conference on Information Technology and Applications (ICITA)*, pp. 191-196, 2005.

[22] C. Douligeris and A. Mitrokotsa, "DDoS attacks and defense mechanisms: classification and state-of-the-art," *Computer Networks,* Vol. 44, No. 5, pp. 643–666, April 2004.

[23] C. E. Shannon and W. Weaver, The Mathematical Theory of Communication, University of Illinois Press, 1963.

[24] C. Estan and G. Varghese, "New directions in traffic measurement and accounting," *In Proceedings of ACM SIGCOMM*, pp. 323-336, 2002.

[25] C. Gonsalves, "Akamai DDoS attack whacks Web traffic," June 2004. [Online]. Available: http://www.eweek.com/article2/0,1895,1612739,00.asp, [last accessed April. 18, 2007].

[26] C. M. Cheng, H.T. Kung, and K.S. Tan, "Use of spectral analysis in defense against DoS attacks," *In Proceedings of IEEE GLOBECOM 2002*, pp. 2143-2148, 2002.

[27] C. Manikopoulos and S. Papavassiliou, "Network Intrusion and Fault Detection: A Statistical Anomaly Approach," *IEEE Communications Magazine*, pp. 76-82, October 2002.

[28] C. Meadows, "A formal framework and evaluation method for network denial of service," *In Proceedings of the 12th IEEE Computer Security Foundations Workshop*, pp. 4-13, June 1999.

[29] C. Papadopoulos, R. Lindell, J. Mehringer, A. Hussain, and R. Govindan, "CROSSACK: Coordinated Suppression of Simultaneous Attacks," *In Proceedings of DISCEX*, pp. 2-13, 2003.

[30] C. Sangpachatanaruk, S.M. Khattab, T. Znati , R. Melhem, and D. Mosse, "Design and analysis of a replicated elusive server scheme for mitigating denial of service attacks," *The Journal of Systems and software*, Vol. 73, No. 1, pp. 15-29, Sep. 2004.

[31] C. Schuba, I. Krsul, M. Kuhn, G. 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*, pp. 208-223, May 1997.

[32] C. Xu, Scalable and Secure Internet Services and Architecture, Chapman & Hall/CRC Press, ISBN 1-58488-377-4, 2005.

[33] CERT Advisory CA-2001-19: "Code Red" worm exploiting buffer overflow in IIS indexing service DLL. Go online to http://www.cert.org/advisories/CA-2001-19.html, [last accessed May 20, 2007].

[34] CERT Coordination Center, "*DoS using nameservers*," http://www.cert.org/incidentnotes/IN-2000-04.html.

[35] CERT Coordination Center, "*Smurf attack,*" http://www.cert.org/advisories/CA-1998-01.html, [Last accessed May 25, 2007].

[36] Cheswick WR, Bellovin SM. Firewalls and internet security: repelling the Wily Hacker: Addison-Wesley Pub Co; 1994.

[37] Computer Emergency Response Team. CERT Advisory CA-2000-01 Denial-of-Service developments. Available at: http://www.cert.org/advisories/CA-2000-01.html, Jan.2000.

[38] Computer Emergency Response Team. CERT Statistics, 2006 http://www.cert.org/stats/cert_stats.html, [last accessed June 28, 2007].

[39] CSI/FBI Computer Crime and Security Survey. Computer Crime Research Center, 2004, Available at: http://www.crime-research.org/news/11.06.2004/423/, [last accessed May 5, 2007].

[40] D. Dean, M. Franklin, and A. Stubblefield, "An Algebraic Approach to IP Traceback," *ACM Transactions on Information and System Security*, Vol. 5, No. 2, pp. 119-137, May 2002.

[41] D. Dittrich, "The "Stacheldraht" distributed denial of service attack tool," University of Washington, December 1999, Available from http://staff.washington.edu/dittrich/misc/stacheldraht.analysis.txt, [last accessed Feb. 11, 007].

[42] D. Dittrich, "The DoS Projects ''trinoo'' Distributed Denial of Service attack tool," University of Washington, October 21, 1999, Available from

<http://staff.washington.edu/dittrich/misc/trinoo.analysis.txt>, [last accessed Feb. 11, 2007].

[43] D. Dittrich, "The Tribe Flood Network Distributed Denial of Service attack tool, University of Washington, October 21, 1999, Available from <http://staff.washington.edu/dittrich/misc/trinoo.analysis.txt>, [last accessed Feb. 11, 2007].

[44] D. Dittrich, G. Weaver, S. Dietrich, and N. Long, "The Mstream Distributed Denial of Service attack tool," May 2000, Available from <http://staff.washington.edu/dittrich/misc.mstream.analysis.txt>, [last accessed Feb. 11, 2007].

[45] D. Evans and D. Larochelle, "Improving security using extensible lightweight static analysis," *IEEE Software*, Vol. 19, No. 1, pp. 42–51, Feb. 2002.

[46] D. K. Y. Yau, J. C. S. Lui, F. Liang, and Y. Yam, "Defending against distributed denial of service attacks with Max-Min fair server-centric router throttles," *IEEE Transactions on Networking*, Vol. 13. No. 1, pp. 29-42, Feb. 2005.

[47] D. Lin, and R. Morris, "Dynamics of Random Early Detection," *In Proceeding of ACM SIGCOMM*, pp. 127-137, New York,1997.

[48] D. Mankins, R. Krishnan, C. Boyd, J. Zao, M. Frentz, "Mitigating Distributed Denial of Service Attacks with Dynamic Resource Pricing," *In Proceedings of 17th Annual Computer Security Applications Conference (ACSAC'01)* p. 0411, 2001.

[49] D. Marchette, "A Statistical Method for Profiling Network Traffic," *In Proceedings of the Workshop on Intrusion Detection and Network Monitoring*, pp. 119-128, Santa Clara, California, USA, April, 1999.

[50] D. Moore, C. Shannon, D. J. Brown, G. Voelker, and S. Savage, "Inferring Internet Denial-of-Service Activity," *ACM Transactions on Computer Systems*, Vol. 24, No. 2, pp 115–139, May 2006.

[51] D. Moore, G. M. Voelker, and S. Savage, "Inferring Internet denial-of-service activity," *In Proceedings of the 10th USENIX Security Symposium*, Vol. 10, Washington, D.C., 2001.

[52] D. Schnackenberg, K. Djahandari, and D. Sterne, "Infrastructure for Intrusion Detection and Response," *In Proceedings of the DARPA Information Survivability Conference and Exposition*, pp. 3-11, 2000.

[53] D. Song and A. Perrig, "Advanced and Authenticated Marking Schemes for IP Traceback," *In Proceedings of IEEE INFOCOM*, pp. 878-886, 2001.

[54] D.S.G. Pollock, A Handbook of Time Series Analysis, Signal Processing and Dynamics, Academic Press.

[55] Debar H, Dacier M, Wespi A., "Towards a taxonomy of intrusion detection systems," *Computer Networks*, Vol. 31, 1999.

[56] E. Yaprak and L. Anneberg, "Utilizing Open Source Tools in the Networking Laboratory," *In Proceedings of ASEE Annual Conference*, Portland, Oregon, June 2005.

[57] E. Yaprak, L. Anneberg, "The Use of Freeware Network Analyzers in a Networking Laboratory," *In Proceedings of ASEE Annual Conference*, Montreal, Canada, June 2002.

[58] 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. 514–524, Hong Kong, 2001.

[59] F. Lau, S. H. Rubin, M. H. Smith, and L. Trajkovic, "Distributed Denial of Service Attacks," *In Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, pp. 2275-2280, October 2000.

[60] G. Carl, G. Kesidis, R. R. Brooks, and S. Rai, "Denial-of-Service Attack - Detection Techniques," *IEEE Internet Computing*, Vol. 10, No. 1, pp. 82-89, Feb. 2006.

[61] G. Cheng, Malware FAQ: Analysis on DDOS tool Stacheldraht v1.666, 2006. Available at http://www.sans.org/resources/malwarefaq/stacheldraht.php, [last accessed June 14, 2007].

[62] G. Oikonomou, J. Mirkovic, P. Reiher, and M. Robinson, "A Framework for a Collaborative DDoS Defense," *In Proceedings of the 22nd Annual Computer Security Applications Conference*, pp. 33-42, 2006.

[63] G. Simon, H. Xiong, E. Eilertson, and V. Kumar, "Scan Detection - A Data Mining Approach," *In Proceedings of SIAM International Conf. on Data Mining (SDM)*, pp. 118-129, 2006.

[64] G. Sivakumar, Cryptographic protocols and Network Security, 2004. Available at http://www.cse.iitb.ac.in/~siva/talks/crypto.pdf, 2004, [last accessed May 25, 2007].

[65] GT-ITM Documentation and tool. Available at http://www.cc.gatech.edu/fac/Ellen.Zegura/graphs.html, [last accessed April 23, 2006].

[66] H. E. Klugh, Statistics: Essentials of Research, 1994.

[67] H. F. Tipton and M. Krause, *Information Security Management Handbook*, CRC Press, 2004.

[68] H. Wang and K. G. shin, "Transport-Aware IP routers: a built-in protection mechanism to counter DDoS attacks," *IEEE Transactions on Parallel and Distributed Systems*. Vol. 14, No. 9, pp. 873-884, 2003.

[69] H. Wang, D. Zhang, and K.G. Shin, "Change-Point Monitoring for the Detection of DoS Attacks," *IEEE Transactions on Dependable and Secure Computing*, Vol. 1, No. 4, pp. 193-208, 2004.

[70] H. Wang, D. Zhang, and K.G. Shin, "Detecting SYN flooding attacks," *In Proceedings of IEEE INFOCOM 2002*, pp. 1530-1539, 2002.

[71] Haymarket Media, "Al-Jazeera hacked in DoS attack," Jan. 2005. [Online]. Available: http://www.itnews.com.au/newsstory.aspx?CIaNID=17603, [last accessed Mar. 16, 2007].

[72] I. Stoica, S. Shenker, and H.Zhang, "Core-Stateless Fair Queuing: Achieving Approximately Fair Bandwidth Allocations in High Speed Networks," *In Proceeding of ACM SIGCOMM,* New York, 1998.

[73] Internet System, RealSecure User's Guide and Reference Manual, 1996.

[74] ITworld.com, "CERT hit by DDoS attack for a third day," May 2001. [Online]. Available:http://security.itworld.com/4339/IDG010524CERT2/pfindex.html, [last accessed May. 20, 2007].

[75] J Jung, B. Krishnamurthy, and M. Rabinovich, "Flash Crowds and Denial of Service Attacks: Characterization and applications for CDNS and Web Sites," *In Proceedings of International World Wide Web Conference,* ACM Press, pp. 252-262, 2002.

[76] J. B. D. Cabrera, L. Lewis, X. Qin, W. Lee, R. K. Prasanth, B. Ravichandran, and R. K. Mehra, "Proactive detection of distributed denial of service attacks using

MIB traffic variables a feasibility study," *In Proceedings of the 7th IFIP/IEEE International Symposium on Integrated Network Management*, pp. 609-622, 2001.

[77] J. Barlow, TFN2K - An Analysis, 2000 Available at http://packetstormsecurity.org/distributed/TFN2k_Analysis-1.3.txt, [last accessed Jan. 23, 2007].

[78] J. Brustoloni, "Protecting electronic commerce from Distributed Denial of Service attacks," *In Proceedings of the 11th International World Wide Web Conference, ACM*, pp. 553–561, 2002.

[79] J. Brutlag, "Aberrant behavior detection in time series for network monitoring," *In Proceedings of USENIX LISA*, New Orleans, pp. 139-146, December 2000.

[80] J. Cao, W. S. Cleveland, D. Lin, D. X. Sun, "Internet Traffic Tends Toward Poisson and Independent as the Load Increases," *Nonlinear Estimation and Classification, eds.*, *Springer*, New York, 2002.

[81] J. Coppens, "SCAMPI-a scalable monitoring plateform for the Internet," *In Proceedings of 2nd International Workshop on Inter-Domain Performance and Simulation (IPS)*, March, 2004.

[82] J. D. Howard, "An analysis of security incidents on the Internet 1989–1995," Ph.D. Dissertation, Carnegie Mellon University, 1997.

[83] J. Haggerty, Q. Shi, and M. Merabti, "Early Detection and Prevention of Denial-of-Service Attacks: A Novel Mechanism with Propagated Traced-Back Attack Blocking," *IEEE Journal on Selected Areas in Communication*, Vol. 23, No. 10, pp. 1994-2002, Oct. 2005.

[84] J. Ioannidis, and S. M. Bellovin, "Implementing Pushback: Router-Based Defense against DDoS Attacks," *In Proceedings of Network and Distributed System Security Symposium*, Catamaran Resort Hotel San Diego, California, 2002.

[85] J. Leiwo, P. Nikander, and T. Aura, "Towards network denial of service resistant protocols," *In Proceedings of the 15th International Information Security Conference*, pp. 301-310, Aug. 2000.

[86] J. Li, "Large-Scale IP Traceback in High-Speed Internet: Practical Techniques and Theoretical Foundation," *In Proceedings of 2004 IEEE Symposium on Security and Privacy*, pp. 115-129, Oakland, CA, 2004.

[87] J. Li, J. Mirkovic, M. Wang, and P. Reither, "L. Zhang. Save: Source address validity enforcement protocol," *In Proceedings of IEEE INFOCOM*, pp. 1557-1566, 2002.

[88] J. M. Conrad and I. Howitt, "Introducing Students to Communications Concepts Using Optical and Low-Power Wireless Devices," *Invited paper to the Journal Elektrik: Special Issue on Electrical and Computer Engineering Education in the 21st Century: Issues, Perspectives and Challenges*, Vol. 14, No. 1, 2006.

[89] J. Mirkovic and P. Reiher, "A Taxonomy of DDoS Attack and DDoS Defense Mechanisms," *ACM SIGCOMM Computer Communications Review*, Volume 34, No. 2, pp. 39-53, April, 2004.

[90] J. Mirkovic and P. Reiher, A University of Delaware Subcontract to UCLA, Available at: http://www.lasr.cs.ucla.edu/Benchmarks_DDoS_Def_Eval.html, [last accessed March 23, 2007].

[91] J. Mirkovic, D-WARD: Source-End Defense Against Distributed Denial-of-service Attacks, Ph.D. Thesis, University of California, Los Angeles, 2003.

[92] J. Mirkovic, E Arikan, S. Wei, R. Thomas, S. Fahmy, and P. Reiher, "Benchmarks for DDOS Defense Evaluation," *In Proceedings of Military Communications Conference (MILCOM)*, pp. 1-10, Washington, DC, Oct. 2006.

[93] J. Mirkovic, G. Prier, and P. Reiher, "Attacking DDoS at the source," *In Proceedings of ICNP 2002*, Paris, France, pp. 312–321, 2002.

[94] J. Molsa, Mitigating Denial of Service Attacks in Computer Networks, Doctoral Dissertation, Helsinki University of Technology, 2006.

[95] J. Xu and W. Lee, "Sustaining availability of web services under distributed denial of service attacks," *IEEE Transactions on Computers*, Vol. 52. No. 2, pp. 195-208, 2003.

[96] K. Fall and K. Vardhan, The NS Manual (Formerly ns Notes and Documentation), 2005.

[97] K. Nichols, S. Blake, F. Baker, and D. Black, "Definition of the Differentiated Services Field (DS field) in the Ipv4 and Ipv6 Headers," RFC 2474, December 1998.

[98] K. Park, and H. Lee, "On the effectiveness of router-based packet filtering for distributed DoS attack prevention in power-law Internets," *In Proceedings of the ACM SIGCOMM Conference*, pp. 15-26, 2001.

[99] K.A. Bradley, S. Cheung, N. Puketza, B. Mukherjee, and R.A. Olsson, "Detecting Disruptive Routers: A Distributed Network Monitoring Approach," *In Proceedings of IEEE Symposium on Security and Privacy*, pp. 50-60, 1998.

[100] L. A. Gordon, M. P. Loeb, W. Lucysgyn, and R. Richardson, *CSI/FBI Computer Crime and Security Survey*, CSI Publications, 2006, pp. 14-15.

[101] L. Chen, T. A. Longstaff, and K. M. Carley, "Charaterizations of defense mechanisms against distributed denial of service attacks," *Computer & Security*, Vol. 23, No. 8, pp. 665-678, Dec. 2004.

[102] L. Feinstein, D. Schnackenberg, R. Balpuari, and D. Kindred, "Statistical Approaches to DDoS Attack Detection and Response," In *Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX'03),* Vol. 1, pp. 303-314, 2003.

[103] L. Garber, "Denial-of-service attacks rip the Internet," *IEEE Computer*, Vol. 33, No. 4, pp. 12–17, Apr. 2000.

[104] L. Limwiwatkul and A. Rungsawang, "Distributed Denial of Service Detection using TCP/IP Header and Traffic Measurement Analysis," *In Proceedings of International Symposium on Communications and Information Technologies (ISCIT 2004)*, Sapporo, Japan, pp. 605-610, 2004.

[105] M. Caesar and J, Rexford, "BGP routing policies in ISP networks,"

[106] M. Handley, Internet Architecture WG: DoS-resistant Internet subgroup report, 2005 Available online at http://www.communications.net/object/download/1543 /doc/mjh-dos-summary.pdf. [last accessed April 25, 2007].

[107] M. Kisimoto, Studies on Congestion Control Mechanisms in the Internet – AIMD-based Window Flow Control Mechanism and Active Queue Management Mechanism, Master Thesis, Osaka University, 2003.

[108] M. Robinson, J. Mirkovic, M. Schnaider, S Michel, and P. Reiher, "Challenges and principles of DDoS defense," *ACM SIGCOMM*, 2003.

[109] M. Roesch, "Snort - Lightweight Intrusion Detection for Networks," *In Proceedings of USENIX Systems Administration Conf. (LISA '99)*, pp. 229-238, 1999.

[110] M. Roughan, T. Griffin, Z. M. Mao, A. Greenberg, and B. Freeman, "Combining Routing and Traffic Data for Detection of IP Forwarding Anomalies," In *ACM SIGCOMM NeTs Workshop*, Portland, pp. 416-417, August 2004.

[111] M. Sung and J. Xu, "IP Traceback-based Intelligent Packet Filtering: A Novel Technique for Defending Against Internet DDoS Attacks," *IEEE Transactions on Parallel and Distributed. Systems*, Vol. 14, No. 9, pp. 861-872, 2003.

[112] MANAnet DDoS White Papers, available at http://www.cs3-inc.com/mananet.html, [last accessed March 26, 2007].

[113] McAfee. Personal Firewall. http://www.mcafee.com.

[114] NS Documentation. Available at: http://www.isi.edu/nsnam/ns, [last accessed March 26, 2007].

[115] O. Spatscheck and L. L. Petersen, "Defending Against Denial of Service Attacks in Scout," *In Proceedings of the 3rd Symposium on Operating Systems Design and Implementation*, pp. 59-72, February 1999.

[116] P. Barford, J. Kline, D. Plonka, and A. Ron, "A Signal Analysis of Network Traffic Anomalies," *In Proceedings of ACM SIGCOMM Internet Measurement Workshop*, pp. 71–82, 2002.

[117] P. D. Shenoy, K. G. Srinivasa, K. R. Venugopal, and L M Patnaik, "Soft Computing Approach for Mining Top-k Ranked webpages from WWW," *GESTS International Transactions on Computer Science and Engineering*, Vol. 3, No. 1, pp. 185-196, 2005.

[118] P. Eronen, "Denial of Service in Public Key Protocols," *In Proceedings of the Helsinki University of Technology Seminar on Network Security*, 2000.

[119] P. Ferguson, D. Senie, "Network ingress filtering: Defeating denial of service attacks which employ IP source address spoofing," RFC 2267, the Internet Engineering Task Force (IETF), 1998.

[120] P. Galli, "DoS attack brings down Sun Grid demo," Mar. 2006. [Online]. Available http://www.eweek.com/article2/0,1895,1941574,00.asp, [last accessed Feb.7, 2007].

[121] P. Mckenny, "Stochastic Fairness Queuing," *In Proceeding of IEEE INFOCOMM*, Piscataway, *N.J.*, pp. 733-740, 1990.

[122] P. Porras and A. Valdes, "Live Traffic Analysis of TCP/IP Gateways," I*n Proceedings of Networks and Distributed. Systems Security Symposium*, Mar. 1998, http://www.sdl.sri.com/projects/emerald/livetraffic.html.

[123] P.J. Criscuolo, "Distributed Denial of Service Trin00, Tribe Flood Network, Tribe Flood Network 2000, and Stacheldraht CIAC-2319," Department of Energy Computer Incident Advisory (CIAC), UCRL-ID-136939, Rev. 1, Lawrence Livermore National Laboratory, February 14, 2000, Available at <http://ftp.se.kde.org/pub/security/csir/ciac/ciacdocs/ciac2319.txt>. [last accessed May 13, 2006].

[124] R. B. Blazek, H. Kim, B. Rozovskii, and A. Tartakovsky, "A novel approach to detection of denial-of-service attacks via adaptive sequential and batch sequential change-point detection methods," *In Proceedings of IEEE Systems, Man and Cybernetics Information Assurance Workshop,* pp. 220-226, 2001.

[125] R. Canonico, D. Cotroneo, L. Peluso, S. P. Romano, and G. Ventre, "Programming Routers to Improve Network Security," *In Proceedings of the OPENSIG Workshop Next Generation Network Programming*, 2001.

[126] R. Chen, J. Park, and R. Marchany, "A Divide-and-Conquer Strategy for Thwarting Distributed Denial-of-Service Attacks," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 18, No. 5, pp. 577-588, May 2007.

[127] R. I. Levin and D. S. Rubin, Statistics for Management, PHI, 1996.

[128] R. Mahajan and S.Floyd, "Controlling High Bandwidth Aggregates at the Congested Router," *AT&T Center for Internet Research at ICSI (ACIRI) and AT&T Labs Research, Technique Report TR-01-001(Draft)*, 2001.

[129] R. Mahajan, S. Bellovin, S. Floyd, V. Paxson, and S. Shenker, "Controlling high bandwidth aggregates in the network," *ACM Computer Communications Review*, Vol. 32, No. 3, pp. 62-73, July 2002.

[130] R. Oppliger, "Internet Security: firewalls and beyond," *Communications of the ACM*, Vol. 40, pp. 92-102, 1997.

[131] R. Stone, "CenterTrack: an IP Overlay Network for Tracing DoS Floods," *In USENIX Security Symposium*, pp.199-212, 2000.

[132] R.K.C. Chang, "Defending against Flooding-Based Distributed Denial-of-Service Attacks: A Tutorial," *IEEE Communication Magazine*, Vol. 40, No. 10, pp. 42-51, Oct. 2002.

[133] R.R. Brooks, Disruptive Security Technologies with Mobile Code and Peer-to-Peer Networks, CRC Press, 2005.

[134] Riverhead Networks. At http://www.riverhead.com/.

[135] S K. Rayanchu and G. Barua, "Defending Against Slave and Reflector Attacks with Deterministic Edge Router Marking (DERM)," Invited Lecture, *In Proceedings of the National Conference on Communications, NCC*, Kharagpur, 2005.

[136] S K. Rayanchu and G. Barua, "Tracing Attackers with Deterministic Edge Router Marking (DERM)," *In proceedings of Distributed Computing and Internet Technology (ICDCIT)*, Vol. 3347 of LNCS, pp. 400-409, 2004.

[137] S. Bellovin, *ICMP Traceback Messages*, IETF draft, 2000 [online] Available at: http://www.research.att.com/smb/papers/draft-bellovin-itrace-00.txt.

[138] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated services," IETF, RFC 2475, 1998.

[139] S. Chen and Q. Song, "Perimeter-Based Defense against High Bandwidth DDoS Attacks," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 16, No. 6, pp. 526-537, June 2005.

[140] S. Floyd and K. Fall, "Router Mechanisms to Support End-to-End Congestion Control," Lawrence Berkeley Laboratories Technical Report.1997.

[141] S. Floyd and V. Jacobon, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Transactions on Networking*, Vol. 1, No. 4, pp. 397-413, 1993.

[142] S. Floyd, and K. Fall, "Promoting the use of End-to-End Congestion Control in the Internet," *IEEE/ACM Transactions on Networking*, Vol. 7, No. 4, pp. 458-472, Aug. 1999.

[143] S. Gibson, The Strange Tale of the Denial of Service Attacks Against GRC.COM, Mar. 2002. Available at http://grc.com/dos/grcdos.htm, [last accessed May 29, 2007].

[144] S. Hazelhurst, "Algorithms for Analyzing firewalls and Router Access Lists," *In Proceedings of Workshop on Dependable IP Systems and Platforms (ICDSN)*, 2000.

[145] S. K. Gupta, V. Bhatnagar, and S. K. Wasan, "On Mining of Data," *IETE Journal of Research, Special issue on Data and Knowledge Engineering*," Vol. 47, No.1, pp. 5-18, 2001.

[146] 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)*, Newark, NJ, pp. 500–504, August 2003.

[147] S. Savage, "Network Support for IP Traceback," *IEEE/ACM Trans. Net.*, Vol. 9, pp. 226-237, 2001.

[148] S. Srivastava, S. Tripathi, D. Sanghi and A. K. Chaturvedi, "A Code Allocation Protocol for Maximizing Throughput in CDMA based Ad Hoc Networks," *In Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, pp.1385-1390, 2003.

[149] S.D. Crocker, "Protecting the Internet from Distributed Denial of service attacks: A proposal," *In Proceedings of IEEE*, Vol. 92 No.9, pp. 1375-1381, Sep., 2004.

[150] T. Aura, P. Nikander, and J. Leiwo, "DOS-Resistant Authentication with Client Puzzles," *Lecture Notes in Computer Science*, Vol. 2133, 2001.

[151] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, Introduction to Algorithms, McGrwaw-Hill, 2001.

[152] T. J. Ott, T.V. Lakshman, and L. H. Wong, "SRED: stabilized RED," *In Proceedings of IEEE INFOCOM*, New York, USA, pp. 1346-1355, March, 1999.

[153] T. M. Gil and M. Poletto, "Multops: a data-structure for bandwidth attack detection," *In Proceedings of the 10th USENIX Security Symposium*, 2001.

[154] T. Peng, C. Leckie, and K. Ramamohanarao, "Survey of Network-Based Defense Mechanisms Countering the DoS and DDoS Problems," *ACM Computing Surveys*, Vol. 39, No. 1, Article 3, April 2007.

[155] T. Peng, C. Leckie, and K. Ramamohanarao, "Defending against distributed denial of service attack using selective pushback," *In Proceedings of the 9th IEEE International Conference on Telecommunications (ICT)* (Beijing, China). 411–429, 2002.

[156] T. Peng, C. Leckie, and K. Ramamohanarao, "Protection from Distributed Denial of Service attack using history-based IP filtering," *In Proceedings of IEEE International Conference on Communications (ICC)*, pp. 482-486, Anchorage, AL, USA, 2003.

[157] U. K. Tupakula and V. Varadharajan, "A controller agent model to counteract DoS attacks in multiple domains," *In Proceedings of IFIP/IEEE Eighth International Symposium on Integrated Network Management*, pp.113-116, 2003.

[158] U. K. Tupakula and V. Varadharajan, "A practical method to counteract denial of service attacks," *In Proceedings of the 26th Australasian computer science conference*, Volume 16, pp. 275-284, Adelaide, Australia, 2003.

[159] U. K. Tupakula and V. Varadharajan, "Analysis of automated model against DDoS Attacks," 2003, Available at: http://citeseer.ist.psu.edu/664761.html, [last accessed May 5, 2007]

[160] U. K. Tupakula and V. Varadharajan, "Analysis of traceback techniques," *In Proceedings of the 2006 Australasian workshops on Grid computing and e-research*, Volume 54, pp. 115-124, 2006.

[161] U. K. Tupakula and V. Varadharajan, "Tracing DDoS Floods: An Automated Approach," *Journal of Network and Systems Management*, Volume 12, pp. 111-135, 2004.

[162] U. K. Tupakula, V. Varadharajan, A. K. Gajam, S. K. Vuppala, P. N. S. Rao, "DDoS: design, implementation and analysis of automated model," *International Journal of Wireless and Mobile Computing*, Vol. 2, No.1 pp. 72 – 85, 2007.

[163] V. Chandola, E. Eilertson, L. Ertoz, G. Simon, and V. Kumar, "Data Mining for Cyber Security," *Book Chapter in Data Warehousing and Data Mining Techniques for Computer Security*, Springer, pp. 1-20, 2006.

[164] V. Goyal, S. K. Gupta, I. Meshram, and A. Gupta, "PRINDA: Architecture and Design of Non-Disclosure Agreements in Privacy Policy Framework," p. 90, I*n Proceedings of the 22nd International Conference on Data Engineering Workshops (ICDEW'06)*, 2006.

[165] V. Paxson, "Bro: A System for Detecting Network Intruders in Real-Time," *International Journal of Computer and Telecommunication Networking*, Vol. 31, No. 24, pp. 2435-2463, 1999.

[166] V. U. Maheswari, A. Siromoney, and K. M. Mehata, "Mining Web Usage Graphs," *In Proceedings of International Conference on Knowledge Based Computer systems (KBCS)*, pp. 186-192, Dec. 2000.

[167] W. Lee, S. J. Stolfo, and K. W. Mok, "A data mining framework for building intrusion detection models," *In Proceedings of the 1999 IEEE Symposium on Security and Privacy*, May 1999. http://citeseer.nj.nec.com/article/lee99data.html.

[168] W. Shi, Y. Xiang, and W. Zhou, "Distributed Defense Against Distributed Denial-of-Service Attacks," *In Proceedings of ICA3PP Springer-Verlag*, LNCS 3719, pp. 357- 362, 2005.

[169] W. Zhao, D. Olshefski, and H. Schulzrinne, "Internet Quality of Service: an overview," *Columbia Technical Report CUCS-003-00*, 2000.

[170] Worldwide ISP security report. Whitepaper. Arbor Networks, Lerington, MA http://www.arbor.net/downloads/Arbor_Worldwide_ISP_Security_Report.pdf, [last accessed March 3, 2007].

[171] X. Geng and A. B. Whinston, "Defeating Distributed Denial of Service attacks," *IEEE IT Professional*, Vol. 2, No. 4, pp. 36–42, 2000.

[172] X. Yang, D. Wetherall, and T. Anderson, "A DoS-limiting network architecture," *In Proceedings of ACM SIGCOMM*, pp. 241-252, 2005.

[173] X. Zhang, S. F. Wu, Z. Fu, and T. Wu, "Malicious Packet Dropping: How it might impact the TCP performance and How we can detect it," *In Proceedings of IEEE ICNP*, pp. 263-272, 2000.

[174] Y. Bai and H. Kobayashi, "Intrusion Detection Systems: Technology and Development," *In Proceedings of the 17th International Conference on Advanced Information Networking and Applications (AINA)*, pp. 710-715, March, 2003.

[175] Y. Hu, A. Nanda, and Q. Yang, "Measurement, Analysis and Performance Improvement of the Apache Web Server," *The International Journal of Computers and Their Applications*," Vol. 8, No. 4, pp. 217-231, 2001.

[176] Y. L. Zheng and J. Leiwo, "A Method to Implement a Denial of Service Protection Base," *Information Security and Privacy*, volume 1270 of LNCS, pp. 90-101, 1997.

[177] Y. Rekhter, and T. Li, "A Border Gateway Protocol 4 (BGP-4)," *RFC 1771, the Internet Engineering Task Force (IETF),* 1995.

[178] Y. Xiong, S. Liu, and P. Sun, "On the defense of the Distributed Denial of Service Attacks: An On-Off Feedback Control Approach," *IEEE Transactions on System, Man and Cybernetics-Part A: Systems and Humans*, Vol. 31 No.4, pp. 282-293, 2001.

[179] Y. Xu and R. Guerin, "On the Robustness of Router-based Denial-of-Service Defense Systems," *ACM SIGCOMM Computer Communication Review*, Vol. 35, No. 3, pp. 47-60, July 2005.

[180] Y. Zhu and Y. Hu, "Ferry: A P2P-Based Architecture for Content-Based Publish/Subscribe Services," *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, Vol.18, No.5, pp. 672-685, 2007.

[181] Z. Fu, H. Huang, T. Wu, S. Wu, F. Gong, C. Xu, and I. Baldine, "ISCP: Design and Implementation of an Inter-Domain Security Management Agent (SMA) Coordination Protocol," *In Proceedings of Network Operations and Management Symposium (IEEE/IFIP NOMS)*, pp. 565-578, 2000.

[182] Z. Gao and N. Ansari, "Tracing cyber attacks from the practical perspective," *IEEE Communications Magazine*, Vol. 43, No. 5, pp. 123-131, May 2005.

[183] Z. Xiaobo, W. Jianbin, C. Xu, "Quality-of-service differentiation on the internet: A taxonomy," *Journal of network and computer applications*, Vol. 30, No. 1, pp. 354-383, Jan. 2007.

List of Publications

International Refereed Journals:

1. K. Kumar, R. C. Joshi, and K. Singh, "Tolerating DDoS Attacks using Dynamic Rate limiting," *International Review on Computers and Software* ,ISSN 1828-6003, Volume 1 N.2, pp. 156-163, Sep. 2006.
2. K. Kumar, R.C. Joshi, and K. Singh, "Detecting DDoS attacks using Traffic Feature Distributions in ISP Domain," *International Journal of Digital Information & Management*, (under review- Document control No. RP 235).
3. K. Kumar, R. C. Joshi, and K. Singh, "Distributing overheads in ISP domain to detect DDoS attacks," *International Journal of Computers & Applications*, 2007 ACTA press, (Accepted- Paper 202-2326).
4. K. Kumar, R. C. Joshi, and K. Singh, "An ISP level Integrated Approach to Combat DDoS Attacks," *An International Journal of Computing & Informatics, ACM* Slovenija, (Communicated- Paper ID: H200610050106).

International Conferences:

1. K. Kumar, R. C. Joshi, and K. Singh, "Predicting Number of Attackers using Regression analysis," *In Proceedings of **IEEE** International Conference on Information and Communication Technology*," DOI 10.1109/ICICT.2007.375402 ISBN: 984-32-3394-8, pp. 319-322, Dhaka, Bangladesh, March 2007,
2. K. Kumar, R. C. Joshi, and K. Singh, "A Distributed Approach using Entropy to Detect Attacks in ISP Domain," *In Proceedings of **IEEE** International Conference on Signal Processing, Communications and Networking*, DOI 10.1109/ICSCN.2007.350758, ISBN 1-4244-0997-7, pp. 331-337, MIT, Chennai, Feb. 2007.

3. K. Kumar, R. C. Joshi, and K. Singh, "An ISP Level Distributed approach to detect DDoS Attacks," *In Proceedings of On Line International Conference on Telecommunications and Networking*, (TeNe 06) , SPRINGER VERLAG, USA, Dec. 2006

4. K. Kumar, R. C. Joshi, and K. Singh, "Detecting Low Rate Degrading and High Bandwidth Disruptive DDoS attacks in ISP Domain," *In Proceedings of International Conference on Information Security and Computer Forensics*, ISBN 81-8284-141-0, pp 83-88, Chennai, Dec. 2006.

5. K. Kumar, R. C. Joshi, and K. Singh, "Filtering High Bandwidth DDoS Attacks using Traffic Level Measurements," *In Proceedings of International Conference of Next Generation Communications ICONGENCOM-06*, pp.210-213, Allahabad , Dec. 2006.

6. K. Kumar, R. C. Joshi, and K. Singh, "A Distributed Approach to Detect DDoS Attacks in ISP Domain," **21st IEEE**, International Conference on Advanced Information and Networking Applications, Canada, (AINA-2007), accepted.

7. B. Gandhi, K. Kumar, R. C. Joshi, "An Efficient DSP-Based Technique to Detect the Signature of Shrew Attacks," *In Proceedings of IEEE International Conference of Signal and Image Processing ICSIP*, Vol. 2, pp. 897-902, Coimbatore, Dec. 2006.

8. B. Gandhi, K. Kumar, R. C. Joshi, "A Novel EPSD Based Approach for Characterization of DDoS Attacks," *In Proceedings of International Conference of Next Generation Communications ICONGENCOM*, pp. 214-218, Allahabad, Dec. 2006.

9. A. Sardana, K. Kumar, R. C. Joshi, and K. Singh, "Detection and Honeypot Based Redirection to Counter DDoS Attacks in ISP Domain," *Third IEEE,CS International Symposium on Information Assurance and Security (IAS07)*, Domain, UK, (Accepted – Paper No. 4780-to be published on IEEE eXplore).

National Conferences/Seminars:

1. K. Kumar, R. Sharma, V. K. Allu, K. Singh, and R. C. Joshi, "An ISP level solution to defend Distributed Denial of Service attacks," *In Proceedings of National Conference on Mathematical Techniques (MATEIT)*, pp 53-57, Delhi, 2006.
2. K. Kumar, R. C. Joshi, and K. Singh, "An Integrated Approach for Defending against Distributed Denial-of-Service (DDoS) Attacks," *IRISS-2006*, IIT Madras. Available at http://www.cs.iitm.ernet.in/~iriss06/paper.html .