Early Detection System for Distributed Denial of Service Attacks

Thaneswaran Velauthapillai

Submitted in total fulfilment of the requirements of the degree of Doctor of Philosophy

Department of Computing and Information Systems
THE UNIVERSITY OF MELBOURNE

September 2014
Abstract

In today's environment, the online services of any organization can become a target of Distributed Denial of Service (DDoS) attacks and the attacks continue to become more complex and sophisticated. Large scale attacks against enterprises and governments around the world have revealed the inability of existing solutions to effectively address DDoS attacks. The main tasks of DDoS defence systems are to detect the DDoS attacks accurately at an early stage and to respond pro-actively to stop the oncoming attack. Detection of the attack is most reliable when undertaken closer to the victim because attack traffic concentrates at the victim, however early attack detection, i.e. detecting that the attack is underway before the attack traffic reaches the victim, is clearly required to be undertaken closer to the attack sources. Assuming that an attack has been detected then protection can be achieved with the help of backbone routers, e.g. by blocking attack traffic from identified attacking sources. However detecting an attack closer to the sources, or in other words far from the victim, is less reliable because attack traffic is more sparse at these points.

A naive solution is to measure traffic throughout the network and report all of the measurements to a centralized system. The central system would raise an attack alert if the total detected traffic to a given victim is greater than some threshold. This traditional client-server model represents a central point of failure and a performance bottleneck. Therefore we propose a distributed and cooperative defence system that we call Gossip Detector can provide early attack detection, in the intermediate network between the attack sources and the victim.
tack sources and the victim.

*Gossip Detector* uses a gossip-based information exchange protocol to share network traffic information in a cooperative overlay network; with regard to a given victim for which the overlay is setup to defend. The defence system is distributed and deployed at intermediate network routers. Through the exchange of traffic measurements, each node in the cooperative system can reach a decision as to whether an attack is underway or not; and thereby the entire system can reach a consensus as to whether an attack is underway or not. In this way a response can be instigated to try and reduce the impact of the attack on the victim. Detecting an attack at an early time is challenging because network delays become a dominant factor.

The research approach adopted in this dissertation includes mathematical analyses of the proposed defence system and the evaluation of the system in a simulated network under different flooding-based attacks using the ns-2 simulator. The simulation results show that *Gossip Detector* can detect attacks within 0.5 seconds with a probability of attack detection as high as 0.99 and probability of false alarm below 0.01 on a topology of average router delay 12 ms. This compares favourably against other widely known methods including change-point detection, TTL analysis and wavelet analysis. In both analytical results and simulations based results, we demonstrate the effectiveness of the defence system in terms of early attack detection and we show the tradeoffs with consumed bandwidth.
Declaration

This is to certify that

1. the thesis comprises only my original work towards the PhD,

2. due acknowledgement has been made in the text to all other material used,

3. the thesis is less than 100,000 words in length, exclusive of tables, maps, bibliographies and appendices.

Thaneswaran Velauthapillai, September 2014
Acknowledgements

Completing my PhD degree has probably been the most challenging activity of my life. The best and worst moments of my doctoral journey have been shared with many people. It has been a great opportunity to spend several years in the Department of Computing and Information Systems at the University of Melbourne, and its members will always remain dear to me.

My first debt of gratitude must go to my supervisors, Dr. Aaron Harwood and A/Professor Shanika Karunasekera. They patiently provided the vision, encouragement and advice necessary for me to proceed through the doctoral program and complete my dissertation. I would also like to thank my PhD committee member, Professor Ramamoohanarao Kotagiri, who provided encouraging and constructive feedback on my research. I also want to thank you all for enabling my defence to be an enjoyable moment, and for all of your brilliant comments and suggestions.

I feel extremely grateful for the strong and selfless support from my parents and my wife Janaki Thaneswaran. Without the sacrifices of my parents, I would never have had the opportunity to undertake this study. Also, I am thankful to my wife who gave her encouragement, trust, and love to me during the most difficult time of my Ph.D study, which I will never forget.

I would like to thank Dr John Papandriopoulos for providing a wonderful Latex template for my thesis format. Furthermore, there were many people who gave interesting feedback and valuable suggestions, for which I am most appreciative. Finally, I am
grateful for the financial support that I have received from the University of Melbourne through the Australian Postgraduate Awards (APA).
## Contents

1 Introduction 1
1.1 Background and Problem Statement 2
  1.1.1 Distributed Denial of Service Attack 2
  1.1.2 DDoS Attack Trends 4
  1.1.3 Internet Security 5
1.2 DDoS Defence Systems 5
  1.2.1 Requirements for a Defence System 6
  1.2.2 Existing Solutions 7
  1.2.3 Motivation for Distributed Solution 10
1.3 Research Question 11
  1.3.1 Legitimate Traffic versus Attack Traffic 12
  1.3.2 Our Approach 12
1.4 Research Methodology 15
  1.4.1 Evaluation Approaches 15
  1.4.2 Test Scenarios 16
  1.4.3 Objective Statement 16
1.5 Contributions of the Thesis 17
1.6 Publications 18

2 A Survey of DDoS Attacks and Defence Mechanisms 19
2.1 Denial of Internet Service ................................................. 20
  2.1.1 DoS and DDoS .................................................. 20
  2.1.2 Target of DDoS ................................................ 22
2.2 Challenges in DDoS Defence ............................................. 23
  2.2.1 Internet Architecture Based Challenges ......................... 23
  2.2.2 Other Challenges .............................................. 25
2.3 Types of Attack ....................................................... 26
  2.3.1 Bandwidth Depletion Attacks ................................ 26
  2.3.2 Resource Depletion Attacks ................................ 27
  2.3.3 Attack Taxonomy .............................................. 30
2.4 DDoS Attack Mitigation Phases ........................................ 30
  2.4.1 Attack Prevention ............................................. 32
  2.4.2 Attack Detection ............................................. 38
  2.4.3 Attack Source Identification ................................ 40
  2.4.4 Attack Reaction Mechanisms ................................ 42
2.5 DDoS Defence Proposals ................................................ 45
  2.5.1 Victim-End Defence ......................................... 45
  2.5.2 Intermediate Network Defence ................................ 46
  2.5.3 Source-end Defence ......................................... 49
2.6 Summary ............................................................... 50

3 Theoretical Model for Gossip Detector .................................. 53
  3.1 Overview ............................................................ 53
  3.2 Abstract Defence System .......................................... 54
  3.3 Basic Analytical Model ............................................. 55
    3.3.1 Probability of Attack Detection .............................. 55
    3.3.2 Probability of False Alarm .................................. 57
    3.3.3 Tradeoff ..................................................... 58
## 3.4 Extended Analytical Model

- **3.4.1 Probability of Attack Detection by Time $\tau$**
- **3.4.2 Detection Delay**
- **3.4.3 Probability of False Alarm by Time $\tau$**
- **3.4.4 Tradeoff**

## 3.5 Summary

### 4 Cooperative Distributed Defence System

- **4.1 System Architecture**
- **4.1.1 Parameters of Gossip Detector**
- **4.1.2 Overlay Construction**

- **4.2 Detailed Gossip Detector Design**
  - **4.2.1 Phases, Rounds and Local Measurement**
  - **4.2.2 Gossip-based Averaging**
  - **4.2.3 Local Adjustment**
  - **4.2.4 Total Victim Bitrate**
  - **4.2.5 Windows and Detection**
  - **4.2.6 Overall Attack Decision**

- **4.3 Performance Evaluation**
  - **4.3.1 Simulation Implementation**
  - **4.3.2 Performance Metrics**

- **4.4 Fundamental Experimental Results**
  - **4.4.1 Traffic Estimation**
  - **4.4.2 Local Versus Global Information**
  - **4.4.3 Round Time and Distance**
  - **4.4.4 Bandwidth Consumption**
  - **4.4.5 Lost and Late Packets**

- **4.5 Summary**
5 Comprehensive Analysis of DDoS Attacks

5.1 Experimental Setup ........................................ 100
5.1.1 Legitimate Traffic ...................................... 100

5.2 Flooding-based DDoS Attacks ................................. 102
5.2.1 Constant Rate Attack .................................. 103
5.2.2 Increasing Rate Attack ................................ 103
5.2.3 Subgroup Attack ....................................... 104
5.2.4 Pulsing Attack ....................................... 105

5.3 Performance Evaluation .................................... 106
5.3.1 Performance Factors .................................. 106
5.3.2 Performance against Flooding-based Attacks .......... 110
5.3.3 Overlay Topology ..................................... 110
5.3.4 Number of Gossip Rounds .............................. 113
5.3.5 Bandwidth Consumption ................................ 114
5.3.6 Network Topology .................................... 116
5.3.7 Deployment Distance .................................. 117

5.4 Comparison to Existing Work .............................. 118
5.5 Summary .................................................. 120

6 Conclusion .................................................. 123

6.1 Summary of the DDoS Problem ............................ 123
6.2 Gossip Detector solution .................................. 124
6.3 Future Work ............................................... 126
6.4 Closing Remarks ......................................... 127
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Types of DDoS attacks</td>
<td>3</td>
</tr>
<tr>
<td>1.2</td>
<td>DDoS attack bandwidth growth reported each year</td>
<td>4</td>
</tr>
<tr>
<td>1.3</td>
<td>A simplified network illustrating different locations for deploying the DDoS defence</td>
<td>7</td>
</tr>
<tr>
<td>1.4</td>
<td>Intuitive understanding of attack traffic</td>
<td>14</td>
</tr>
<tr>
<td>2.1</td>
<td>DDoS agent-handler attack model</td>
<td>21</td>
</tr>
<tr>
<td>2.2</td>
<td>Simple example for a reflector Distributed Denial-of-Service Attack</td>
<td>28</td>
</tr>
<tr>
<td>2.3</td>
<td>TCP SYN attack model</td>
<td>28</td>
</tr>
<tr>
<td>2.4</td>
<td>IP spoofed type of DDoS attack</td>
<td>29</td>
</tr>
<tr>
<td>2.5</td>
<td>DDoS attack taxonomy</td>
<td>31</td>
</tr>
<tr>
<td>2.6</td>
<td>Taxonomy of DDoS mitigation mechanisms</td>
<td>33</td>
</tr>
<tr>
<td>2.7</td>
<td>The communication between an authorized source and a protected site in SOS</td>
<td>37</td>
</tr>
<tr>
<td>3.1</td>
<td>A simple system model to analyse performance of Gossip Detector</td>
<td>54</td>
</tr>
<tr>
<td>3.2</td>
<td>A probability tree diagram shows all the possible expected events at a defence node</td>
<td>55</td>
</tr>
<tr>
<td>3.3</td>
<td>Probability of attack detection versus probability attack detection for indi-vidual defence node</td>
<td>56</td>
</tr>
</tbody>
</table>
### Section 3

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.4 Comparing the variation of probability of attack detection with number of defence nodes</td>
<td>57</td>
</tr>
<tr>
<td>3.5 Probability of false alarm versus probability false alarm for individual defence node</td>
<td>59</td>
</tr>
<tr>
<td>3.6 Probability of false alarm with number of defence nodes in Gossip Detector</td>
<td>59</td>
</tr>
<tr>
<td>3.7 Tradeoff between probability of attack detection and probability of false alarm, $p = 0.9, q = 0.1$</td>
<td>60</td>
</tr>
<tr>
<td>3.8 Tradeoff between probability of attack detection and probability of false alarm, $p = 0.75, q = 0.25$</td>
<td>60</td>
</tr>
<tr>
<td>3.9 Compares the probability of attack detection when the number of defence nodes varies in Gossip Detector</td>
<td>63</td>
</tr>
<tr>
<td>3.10 Compares the probability of attack detection when the number of defence nodes varies in the defence system</td>
<td>63</td>
</tr>
<tr>
<td>3.11 Compares the probability of attack detection when the number of defence nodes varies in Gossip Detector</td>
<td>64</td>
</tr>
<tr>
<td>3.12 Compares the average detection delay when the number of defence nodes varies in Gossip Detector</td>
<td>64</td>
</tr>
<tr>
<td>3.13 Compares the average detection delay when the number of defence nodes varies in the defence system</td>
<td>66</td>
</tr>
<tr>
<td>3.14 Compares the average detection delay when the number of defence nodes varies in Gossip Detector</td>
<td>67</td>
</tr>
</tbody>
</table>

### Section 4

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 Example Gossip Detector deployed on a simple network</td>
<td>71</td>
</tr>
<tr>
<td>4.2 Example timeline for Gossip Detector</td>
<td>76</td>
</tr>
<tr>
<td>4.3 Attack detection procedure</td>
<td>78</td>
</tr>
<tr>
<td>4.4 Number of defence nodes at distances</td>
<td>81</td>
</tr>
<tr>
<td>4.5 Experimental topology</td>
<td>82</td>
</tr>
<tr>
<td>4.6 Estimated total bit rate to victim during an attack</td>
<td>86</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>-----------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>4.7 Performance recovery of <em>Gossip Detector</em></td>
<td>87</td>
</tr>
<tr>
<td>4.8 Comparing performances in all distances for different round times and the number of rounds (1,2 and 3)</td>
<td>89</td>
</tr>
<tr>
<td>4.9 Comparing performances in all distances for different round times and the number of rounds (4 and 5)</td>
<td>90</td>
</tr>
<tr>
<td>4.10 Comparing performance of <em>Gossip Detector</em> for different rounds (1,2 and 3) for CBR attack</td>
<td>92</td>
</tr>
<tr>
<td>4.11 Comparing performance of <em>Gossip Detector</em> for different rounds (4 and 5) for CBR attack</td>
<td>93</td>
</tr>
<tr>
<td>4.12 Comparing performances in all distances for different round times and the number of rounds</td>
<td>94</td>
</tr>
<tr>
<td>4.13 Comparing performance of <em>Gossip Detector</em> for different rounds (2 and 4) for VBR attack</td>
<td>95</td>
</tr>
<tr>
<td>4.14 Bandwidth consumption at different distances and packet drop rate in a phase interval</td>
<td>96</td>
</tr>
<tr>
<td>5.1 Sample experimental topology</td>
<td>101</td>
</tr>
<tr>
<td>5.2 Pareto on/off Internet traffic model</td>
<td>102</td>
</tr>
<tr>
<td>5.3 Attack rate variation for a constant rate attacking model</td>
<td>103</td>
</tr>
<tr>
<td>5.4 Increasing rate attacking model</td>
<td>104</td>
</tr>
<tr>
<td>5.5 Subgroup attacking model</td>
<td>104</td>
</tr>
<tr>
<td>5.6 Pulsing attacking model</td>
<td>105</td>
</tr>
<tr>
<td>5.7 Factors affecting performance of <em>Gossip Detector</em></td>
<td>107</td>
</tr>
<tr>
<td>5.8 Histogram of background and attack traffic of a defence node at a distance</td>
<td>108</td>
</tr>
<tr>
<td>5.9 Experimenting <em>Gossip Detector</em> against different flooding-based attacks</td>
<td>111</td>
</tr>
<tr>
<td>5.10 Examples of overlay topologies</td>
<td>112</td>
</tr>
<tr>
<td>5.11 Comparing Constant-rate attack’s performance for different overlay topologies and number of rounds</td>
<td>114</td>
</tr>
</tbody>
</table>
5.12 Comparing trade offs between detection delays and bandwidth consumption
5.13 Comparing all type of attacks
5.14 Detection delay variation with Gossip Detector deployment distance
5.15 Comparing probability of attack detection at different deployment distance
5.16 Comparing probability of false alarm at different deployment distance
## List of Tables

1.1 Comparison of defence systems based on the deployment locations . . . . 9  
2.1 The Signature-based versus and Anomaly-based attack detection . . . . 38  
2.2 Comparing different distributed defence System . . . . . . . . . . . . . 51  
4.1 Notation used in proposal of *Gossip Detector* . . . . . . . . . . . . . . . . . 73  
5.1 Configuration of Pareto on/off traffic generator . . . . . . . . . . . . . . 100  
5.2 Comparing *Gossip Detector* with other defence proposals . . . . . . . . . 119
Chapter 1

Introduction

Distributed Denial of Service (DDoS) attacks on Internet services are a major operational threat compared to other network security problems [1]. The increasing trend in the scale and frequency of DDoS attacks is an alarming threat to the availability of Internet services. A survey from Arbor Network [10] revealed that the peak scale of DDoS attacks in 2010 reached 100 Gigabits per second against a single target, while it peaked at 300 Gigabits per second in 2013 [47]. One of the fundamental reasons for the rapid growth of DDoS attacks is the lack of availability of effective solutions to detect and to identify the attacker, especially in the case of newer and smarter DDoS attacks. Furthermore, an attacker does not require deep technical knowledge as the attacks can be undertaken with user-friendly attacking tools such as Trinoo, TFN, TFN2K, Shft, Stacheldraht, etc. [109], and attack packets appear as legitimate requests which are actually coming from an overwhelming number of zombies and spoofed identities, making it hard to identify and block them.

From an Internet topology perspective, DDoS attacks come from a large number of distributed routers at the edges of the network. It is accepted that defence against these attacks must address their distribution nature to be successful [62]. Thus, addressing DDoS threats requires a distributed solution which contains defence nodes located at multiple places in the Internet that can work cooperatively. The attacks ideally should be stopped as close to the sources as possible, saving network resources and reducing congestion. Distributed collaboration is emerging as an effective way to defend
1.1 Background and Problem Statement

1.1.1 Distributed Denial of Service Attack

The Internet consists of millions of computers around the world and a lot of people use the Internet daily. A Denial of Service (DoS) attack is an attack with the purpose of denying intended users from utilising a network resource such as an Internet service. A Distributed Denial of Service (DDoS) attack is a coordinated attack on the availability of
services in a given target system or a network that is launched indirectly through many compromised computing systems.

DDoS attacks are broadly classified into two categories: bandwidth depletion attacks and resource depletion attacks [43]. A bandwidth depletion attack aims to flood the victim’s network with unwanted traffic, preventing legitimate traffic from reaching the victim’s system. E.g. flooding based attacks using UDP packets or ICMP echo requests. A resource depletion attack is designed to tie up the resources of the victim’s system by targeting a server or process within the system, making it impossible for legitimate requests to obtain service. TCP SYN flooding is the most prevalent attack and is an ideal example of a resource depletion attack. As illustrated in Fig. 1.1 the most common type of DDoS attack is the bandwidth depletion attack and our research focuses on this type of attack [52].
According to the Symantec Internet Security Report [22] a new attack is launched, on average, every minute of every day. Attack tools becomes more sophisticated as time passes and they easily find new security flaws on holes in the Internet connected PCs and other resources. Thus recruiting a large number of attacking zombies is not a hard task and large-scale DDoS attacks are a rapidly emerging problem. The Fig. 1.2 shows the attack growth from 2002 to 2013 [10] [47]. The maximum attack size was 300Gbps against a single target on 2013. Also, the lack of common characteristics of DDoS streams makes it hard to differentiate between attack traffic and legitimate traffic.
1.1.3 Internet Security

The primary reason for the prevalence of DDoS attacks is that the Internet was originally designed for openness and scalability without much consideration of security [71]. Malicious users exploit the design weaknesses of the Internet and develop attacking tools to easily perform a DDoS attack. Many attack tools are based on misuse of ordinary network protocols [72]. Many hosts on the Internet are vulnerable to DDoS attacks, particularly hosts that are either running no antivirus software or out-of-date antivirus software, or those that have not been properly patched. The Symantec Internet security report [97] shows that 61% of malicious sites are regular web sites that have been compromised and infected with malicious code and hence more than 250000 zombies are created each month. As a result, almost all Internet services are vulnerable to distributed denial of service attacks at sufficient scale. In most cases, sufficient attack scale can be achieved by compromising enough end-hosts or routers, and using those compromised hosts to launch the attack.

In principle it is not possible to distinguish between a DDoS attack and a flash crowd. There are no common characteristics of DDoS streams that could be used for their detection and filtering. The attackers achieve their desired effect by increasing the volume of attack packets, and can alter all packet fields to avoid characterisation. The cooperation of distributed sources makes DDoS attacks hard to combat or trace back. Unexpected heavy and non-malicious traffic has the same effect as a DDoS attack.

1.2 DDoS Defence Systems

The seriousness of the DDoS problem and the increased frequency and strength of attacks has led to a wide variety of defence mechanisms.
1.2 DDoS Defence Systems

1.2.1 Requirements for a Defence System

The design of an effective DDoS defence system contains a significant challenge in successfully detecting and stopping attacks while at the same time preserving performance of the network. The following are widely accepted requirements that a good DDoS defence system must satisfy:

**Early attack detection** [1, 27, 90]: DDoS attacks flood victims with unwanted packets. This means that victims cannot contact anyone else for help during the attack. Consequently, any potential reaction can be instigated if the attack is detected early. Usually traffic flow increases suddenly and without any warning [1]. For this reason defence mechanisms must detect and react quickly to avoid collateral damage to the victim.

**Reliable attack detection and response** [62, 71, 80, 109]: The system should detect all attacks that occur and should not generate false alarms.

**Low impact on performance cost** [60]: Performance cost is an important factor in designing a defence system. Resource requirements of a defence system must not degrade the performance of the deploying network.

**Effective response** [33, 61]: When an attack is detected, the system should engage in a response that significantly reduces the effectiveness of the attack, regardless of the attack characteristics.

**Realistic design** [62]: Defence systems should have practical deployment on the Internet. Thus the system should require little or no changes to the existing Internet infrastructure.
**High security** [1,54]: A DDoS defence system must ensure that it cannot be misused to degrade or deny service to legitimate clients. It also must be resistant to attempts by attackers to bypass or disable it.

### 1.2.2 Existing Solutions

As shown in Fig 1.3, the attacking nodes can be in the victim’s network, the intermediate network and the source(s) networks [54]. The victim’s network is the administrative domain that contains the victim of the attack: the intermediate network is (are) the administrative domain(s) that forward attack packets to the victim and the source networks are the administrative domain(s) that contain most of the attacking nodes. With regard to deployment location, the existing defence mechanisms are categorised as victim-end defence, intermediate network defence and source-end defence.
1.2 DDoS Defence Systems

**Introduction**

**Victim-End Defence**

Most defence systems for combating DDoS attacks work from the victim’s side. The victim-end defence typically cannot provide complete protection from DDoS attacks because the defence system may itself be overwhelmed by the attack traffic. Most of the proposed systems attempt to characterise the attack traffic and install filtering rules in the upstream routers of the victim network. In some cases this can successfully defend against the attack’s effects. For example a web server that is under a UDP flood attack can request that all UDP traffic be filtered. However, this technique fails when the amount of traffic is sufficiently high enough to cause the filtering mechanism in the upstream router to fall behind. Further, the Internet core has much greater resources than its edges [71], and thus a DDoS can fully utilize the core resources to overwhelm the victim network and exhaust its limited resources.

**Intermediate Network Defence**

These types of DDoS defences are deployed in the intermediate network. Response to a DDoS attack by the intermediate network is obviously more effective than Victim-End Defence response since large volumes of attack traffic can be handled easily and attacks can be traced back to the sources. This is because defence nodes are not as overwhelmed as those of the victim’s network. However, there are several challenges that prevent widespread deployment of these approaches. For example difficulties in detecting the attack and identifying the attackers arise, as the defence nodes in the intermediate network usually observe only partial effects of the attack. Furthermore, the lack of interdomain cooperation, as well as the deployment costs, are also major issues.

**Source-End Defence**

This type of DDoS defence is deployed at networks that are hosting some of the attack machines, potentially on the machines themselves. Source-End defence systems can
monitor the outgoing traffic from these networks and control attack traffic. Furthermore Source-End defence systems provide a highly selective response to DDoS attacks and almost no damage to legitimate traffic [58]. However the distributed communication costs are the highest of these three placement strategies; potentially requiring numerous communications across the entire Internet.

Comparison of Deployment Locations

Table 1.1 compares the advantages of defence systems with different deployment locations. Compared with the victim-end defence systems, the distributed defence systems, deployable at intermediate networks or source-end, can discover the attacks in a more timely fashion, as multiple nodes can share information and reach an accurate attack decision. Components of the distributed defence system can cooperate with each other to combat the attacks. Mostly distributed defence systems locate their defence nodes either at edge routers, or in the core routers. Distributed defence at the Internet core has a definite advantage over single point defence. Our proposed defence system is a decentralised and distributed defence system deployable at intermediate networks or source networks.

Table 1.1: Comparison of defence systems based on the deployment locations

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Victim-End</th>
<th>Intermediate</th>
<th>Source-End</th>
</tr>
</thead>
<tbody>
<tr>
<td>distributed</td>
<td>typically not</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>ease of detection</td>
<td>easy</td>
<td>moderate</td>
<td>hard</td>
</tr>
<tr>
<td>cooperation</td>
<td>none needed</td>
<td>autonomous systems in router hardware possible</td>
<td>organizations and users</td>
</tr>
<tr>
<td>deployment</td>
<td>only at victim</td>
<td>some</td>
<td>directly at source</td>
</tr>
<tr>
<td>trace back</td>
<td>not possible</td>
<td>large</td>
<td>highly selective</td>
</tr>
<tr>
<td>selectivity</td>
<td>small</td>
<td>moderate</td>
<td>large</td>
</tr>
<tr>
<td>scale</td>
<td>none</td>
<td>medium</td>
<td>instant</td>
</tr>
<tr>
<td>early detection</td>
<td>impossible</td>
<td>possible</td>
<td>high</td>
</tr>
<tr>
<td>overhead</td>
<td>none</td>
<td></td>
<td>possible</td>
</tr>
<tr>
<td>response</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1.2 DDoS Defence Systems

**Introduction**

**distributed** – ability to detect an attack at multiple locations

**ease of detection** – significance of attack

**cooperation** – cooperation among defence nodes

**deployment** – location of the defence system

**trace back** – identifying source of attack

**selectivity** – ease of attack response

**scale** – strength of attack

**early detection** – time between start and detection of an attack

**overhead** – complexity of defence system deployment

**response** – defending attack after detection

1.2.3 Motivation for Distributed Solution

Although some current solutions can successfully detect and drop attacking packets against well known attacks, there are no commercial or research solutions, that we know of, to DDoS attacks that make any guarantees of continued service to legitimate clients of the victim during the attack [75]. Furthermore, most of the proposed solutions are centralised and isolated systems, which are used to protect a single target or network. Centralised defence systems are mostly deployed close to the victim because of economic reasons [80] and so that attacks may be identified using more accurate attack characteristics.

By definition a DDoS attack comes from distributed attacking sources (e.g. zombie computers), which suggests that a distributed defence mechanism is the best for complete defence [65]. Distributed defence systems that are positioned either at the Intermediate or Source-End can potentially detect the attack before the attack packets reach the victim. While attack detection is more effective closer to the victim network, responses
to the attack are more selective closer to the source [75]. Generally, distributed defence systems detect an attack at multiple points and combine attack decisions from each point to make a global decision about whether an attack is underway or not. This can consume bandwidth and incurs communication delays. However, this type of collaborative decision-making reduces the probability of false alarms in such a distributed system and increases the probability of attack detection. In this thesis we consider a distributed defence system and show the trade off in overheads versus attack detection reliability and early detection time.

1.3 Research Question

We address the following research question:

Can we reliably and efficiently detect a DDoS attack, at a time significantly before the attack is observed (detected) at the victim’s end?

Ideally we would want to detect the attack at exactly the same time that the attack begins; let’s say time $t_0$. Obviously we cannot detect the attack any earlier than $t_0$. Also, assuming it to be possible, to detect the attack at exactly $t_0$ we would need to have defence nodes at the sources of the attack, i.e. Source-End. Generally we define the attack detection delay, $\tau$, as the time from $t_0$ to when a global decision is made that an attack is underway, i.e. $t_0 + \tau$ is the time that the attack was detected. The worst-case value of $\tau$ is effectively the delay between when the attack starts and when the victim detects the attack.

The sub-questions that we answer become:

- What is the best placement of the defence nodes? I.e., at what distance from the victim should they be placed? In our work we consider that they are Intermediate Network defence nodes, but in general we abstractly consider the defence nodes to be some number of router hops from the victim.
1.3 Research Question

• How do the defence nodes communicate their individual attack decisions and/or measurements of potential attack traffic, in order to arrive at a global decision?

Answers to the above questions affect both $\tau$ and other aspects such as the bandwidth overheads and the reliability of attack detection.

1.3.1 Legitimate Traffic versus Attack Traffic

In our work we do not distinguish between legitimate traffic and attack traffic. We count all traffic going to a given victim as potential attack traffic. In this case, our attack detection system is detecting flash crowds as attacks. Techniques for distinguishing legitimate traffic from attack traffic could be applied orthogonally to our work, in the sense that the technique could enable us to not count legitimate traffic. This is a direction that we did not pursue in our research.

1.3.2 Our Approach

In this research we proposed to use a cooperative overlay network with nodes that are distributed throughout the Intermediate Network. A gossip-based scheme (i.e. information sharing) is used to compute an estimate of the total amount of attack traffic that is headed towards the victim. We refer to our proposed system as Gossip Detector.

Intuitive Understanding

Fig 1.4 provides an intuitive understanding of a distributed defence system that is placed in the Intermediate Network. The attack traffic arrows represent packets of the attack, close to time $t_0$. A sample period of $\epsilon$ seconds is considered as the time for which traffic is sampled and the bit-rate of the traffic is estimated. At the victim’s end, relatively fewer traffic samples are taken than in a distributed defence system which has multiple sample points. In this case, the distributed defence system has a better view of the overall
traffic going to the victim and therefore will potentially reach a decision that an attack is underway before the victim would reach the decision.

**Characteristics and general operation of Gossip Detector**

*Gossip Detector* addresses the requirements listed in Section 1.2.1:

- Early attack detection is obtained through close defence deployment to the attacking sources.

- *Gossip Detector* consists of the distributed defence nodes at intermediate networks to detect DDoS attacks independently at defence nodes.

- To make reliable and rapid attack detection, *Gossip Detector* utilizes a peer-to-peer overlay network which removes central points of failure and the associated performance bottleneck.

- The simple gossip based communication mechanism is used to exchange attack information among detection nodes and to conclude the overall network-wide attacks effectively and accurately.

*Gossip Detector* makes attack decisions based on the estimate of global traffic bitrate towards the victim, called “global victim bitrate”, estimated at the Intermediate Network. First, each defence node makes a local measurement of attack traffic bitrate to the victim; this is called “local victim bitrate”. Then all nodes participate in a distributed averaging algorithm whereby they arrive at the average of their local measurements. The average is then multiplied by the number of overlay nodes and the result is taken to be the estimated global victim bitrate. If the estimated global victim bitrate exceeds the victim’s capacity (which is a predefined threshold) then the defence nodes flag an attack alert. If more than a threshold number of defence nodes flag attack alerts then the victim is considered “under attack” at that time.
1.3 Research Question

Introduction

Figure 1.4: Intuitive understanding of attack traffic in the network going towards the victim. The attack traffic is typical coordinated to start at the same time, $t_0$. The length of the black arrow indicates how far the traffic is routed through the network over a short time interval, $\epsilon$. Some of the traffic passes through the defence nodes and is measured in that time interval.
1.4 Research Methodology

Distributed denial of service attacks occur in distributed and complex network environments and potentially involve a large numbers of entities. Reproduction of similar environments in a DDoS defence experiment is a challenging task [55]. It is desirable when testing systems for DDoS defence, that they include a reproduction of DDoS attacks and the same Internet-like legitimate traffic. However, this is exceedingly difficult. Reproduction of the scale of the attack is infeasible, and scalability of any DDoS defence system is similarly unfeasible or assessment through experiments [57].

1.4.1 Evaluation Approaches

Theoretical Method: Theory is one of the methods used to analyse performances of a proposed defence system achieved by making assumptions. We used theoretical analysis to explore the performances of Gossip Detector. Furthermore, theory may be useful to evaluate the robustness of a given defence.

Simulation: Simulation is a highly accepted and popular method for addressing network performance experiments. For example, Internet devices, such as switches and routers, are only modelled at a high level in popular packet-level simulators such as ns-2 [15]. The challenge in using a simulation is to correctly set remaining router parameters, such as buffer size. Hence, many research papers report results that are highly sensitive to the default setting in the simulator.

Emulation: Emulation involves testing in a mini-network, such as a network lab. Three test beds have been popular for DDoS defence testing: Emulab [95], DETER [14] and Planetlab [73].

Real Network Deployment: Testing a defence system in an operational network enables the most realistic testing environments. The system deploys in a real topology with real traffic and attacks. The main drawback of this approach is that many
1.4 Research Methodology

Introduction

...researchers lack access to a large network to initiate the deployment of new defences. To the best of our knowledge, none of surveyed papers used this type of deployment for evaluation.

We follow the ns-2 [24] simulation approach that has been used to evaluate top recent publications such as TVA [102], StopIt [50] and PSP [20]. The ns-2 simulator uses two different languages TCL and C++. We implemented Gossip Detector in C++ to support relatively large-scale simulations. Also we use some theoretical methods to analyse Gossip Detector. As mentioned [57], our experience using ns-2 also suggests that these results may sometimes overestimate and sometimes underestimate the attack impact.

1.4.2 Test Scenarios

For realistic experimental results we used widely accepted methods for generating the Internet topology, background Internet traffic and flooding based attack models. Because denial of service is experienced by legitimate clients in presence of attack traffic, legitimate traffic settings will greatly influence the success of an attack. A Pareto model with shape parameter 1.4 is used to represent the web object size in order to simulate legitimate traffic [42]. We used the Brite [53] topology generator with Waxman model [94] and random node placement to generate an Internet-like network topology.

1.4.3 Objective Statement

In order to clearly represent the objective of Gossip Detector, we mathematically define an objective statement. If we consider \( \tau \) as the attack detection delay of the system then the objective of Gossip Detector is defined as a constrained optimization problem,

\[
\begin{align*}
\min_{p_1, p_2, \ldots, p_n} & \quad E[\tau] \\
\text{Subject to} & \quad P_{fa} \leq \chi & P_{ad} \geq \psi
\end{align*}
\]

(1.1)
Where $\chi$ and $\psi$ are the acceptable false alarm probability ($P_{fa}$) and the expected attack detection probability ($P_{ad}$) respectively and $p_1, p_2, ..., p_n$ are the parameters of Gossip Detector.

## 1.5 Contributions of the Thesis

In this section, we outline the contributions that have been made through each chapter.

### Chapter 2

We present a detailed study of DDoS attacks and existing major defence proposals. We explore the challenges in defending against DDoS attacks and we categorise the existing proposals, highlighting the limitations of each defence mechanism. Furthermore, we illustrate the attack taxonomy and we present significant attack defence responses.

### Chapter 3

We propose an abstract model of Gossip Detector and analyse probabilistically how Gossip Detector can achieve early attack detection using decentralized defence nodes. First we define performance metrics which are used to measure the performance of Gossip Detector. Then by considering the attack detection rules of Gossip Detector, we analyse its performance.

### Chapter 4

We propose detailed implementation of Gossip Detector and explain the experimental set up to perform an initial evaluation of Gossip Detector under a simple attack model. The results of this chapter are presented in a publication [91].
Chapter 5

In this chapter, we concentrate on producing more realistic results by experimenting with Gossip Detector under different types of flooding-based attacks in variety of Internet topologies. The experimental results of this chapter agree with the expected defence performance of our theoretical analysis.

Chapter 6

This section summarises and concludes the thesis and provides suggestions for future work.

1.6 Publications


- Thaneswaran Velauthapillai, Shanika Karunasekera and Aaron Harwood, “Early Detection of Flooding-Based DDoS Attacks Using a Cooperative Overlay Network”, This Journal paper is submitted to the Journal of Parallel and Distributed Computing.
Chapter 2

A Survey of DDoS Attacks and Defence Mechanisms

In this chapter, we present a study of denial of service attacks in the Internet, and a survey of existing defence mechanisms. We investigate how the attackers successfully exploit the security holes of the Internet to disturb online services by bypassing existing defence systems successfully. We explore a number of existing DDoS attack defence proposals from the literature and we present the defence systems according to their deployment locations and investigate their strengths and weaknesses. We present related works in the three categories defined from Section 1.2.2: Victim-End Defence, Intermediate Network Defence, and Source-End Defence.

From early on, a number of DDoS attack defence techniques have been proposed by academics and security experts. Generally defend systems deployed in the commercial Internet services are not published due to security reasons. For example, to the best of our knowledge, there are no publications about the DDoS defence strategies deployed in Google services, Yahoo servers or Facebook cluster machines. However DDoS attacks against major Internet sites are usually widely published. Furthermore, attack details against popular sites are not disclosed, to mitigate the affects on customer confidence about the Internet service, and thus there is generally a huge challenge to security researches to explore and develop a good system solution [60].
2.1 Denial of Internet Service

The main objective of computer security is to provide information availability, confidentiality and integrity to the intended users [6]. Availability can be defined as the provision of the information or resource in a reliable and timely manner. Denial of Service is a potential security threat that denies the availability of a resource in a system. A Denial of Service Attack is an action executed by a malicious entity to make a resource unavailable to its intended users.

2.1.1 DoS and DDoS

Malicious users who attempt to exhaust Internet services and deny their services to intended users is known as a Denial-of-Service (DoS) attacks. When attack from a single host of the network, it constitutes a DoS attack. It is also possible that a lot of malicious hosts coordinate to flood the victim with useless attack packets simultaneously from multiple points. This type of attack is called a Distributed DoS (DDoS) attack. The victim can be a network server, client or router, a network link or an entire network, an Internet Service Provider or a country [32].

The Agent-Handler model of a DDoS attack consists of clients, handlers, and agents as shown in Fig. 2.1. The attacker communicates with the rest of the DDoS attack system. The handlers are software packages, located throughout the Internet, that the attacker uses to communicate with the agents. The agent software exists in compromised systems that will eventually carry out the attack. The attacker communicates with any number of handlers to identify which agents are up and running, when to schedule attacks or to upgrade agents. The machine running the agent systems does not know that the system has been compromised and can take part in a DDoS attack.
2.1 Denial of Internet Service

Figure 2.1: DDoS Agent-Handler Attack Model.
2.1 Denial of Internet Service

A Survey of DDoS Attacks and Defence Mechanisms

2.1.2 Target of DDoS

The victim of a DDoS attack can be an end system, a router, a communication link or an entire network, an infrastructure, or any combination of these [32].

**DDoS on Application** In application DDoS attacks, an attacker attempts to prevent the application from performing its intended tasks by causing the application to exhaust a resource. The resources for applications may be the maximum number of processes and the maximum number of simultaneous connections that an application can create, etc. Examples of application level DDoS include: attempts to flood web applications, blocking user access by repeated invalid login attempts, opening huge number of application-database connections by CPU intensive SQL queries.

**DDoS on Operating System** Operating system DDoS attacks are very similar to application DDoS attacks. A well-known DDoS attack on an operating system is the Transmission Control Protocol (TCP) SYN flooding attack [4]. The attacker sends a flood of TCP SYN packets to the victim without completing the TCP handshake, thereby exhausting the victim’s connection state memory.

**DDoS on Router** Many of the DDoS attacks against an end system can also be launched against an IP router. The simplest attack on a router is to overload the routing table with a sufficiently large number of routes that the router runs out of memory, or the router has insufficient CPU power to process the routes [17].

**DDoS on Ongoing Communication** Instead of attacking the end system, an attacker may attempt to disrupt an ongoing communication. If an attacker can observe a TCP connection, then it is relatively easy to spoof packets to either reset that connection or to de-synchronize it so that no further progress can be made [36].

**DDoS on Links** The simplest form of DDoS attack on links is to send enough non-congestion-controlled traffic (e.g. UDP traffic) such that a link becomes excessively congested, and legitimate traffic suffers unacceptably high packet loss [32].
DDoS on Infrastructure  Many communication systems depend on some infrastructure for their normal operations. For example, infrastructure can be a global domain name system or a global public key infrastructure, or can be a local area Ethernet infrastructure or a wireless access point. For example denying access to a DNS server effectively denies access to all services, such as Web, email, public keys and certificates etc., that are being served by that DNS server [96].

DDoS on Firewalls and IDS  Firewalls are intended to defend the systems behind them against outside threats by restricting data communication traffic to and from the protected systems [81]. Firewalls may also be used in defending against denial of service attacks. Meanwhile, firewalls themselves may become the targets of DDoS attacks.

2.2 Challenges in DDoS Defence

Despite the huge effort by researchers and security experts to address the denial of service problem, Internet denial of service attacks still remain an unsolved problem. There are various technical and non-technical challenges that need to be understood in order to design good solutions. We first explore the design principles of the Internet and their implication to the denial of service problem. We also look at other technical challenges and how they effect the solution to the denial of service problem.

2.2.1 Internet Architecture Based Challenges

We look at various Internet design principles and give an analysis of how each design choice translates into a challenge in addressing the denial of service problem. The order of these principles reflects the importance of different design goals of the Internet presented in [21].

On-demand Resource Sharing  The fundamental structure of the Internet is a packet
2.2 Challenges in DDoS Defence A Survey of DDoS Attacks and Defence Mechanisms

switched communications facility in which networks are inter-connected together using store and forward packet communications [21]. Packet switching allocates link to use on demand among the users. In such environment, a misbehaving user can disrupt service for other users by occupying most of the shared resources. Such resource sharing based on users’ demand creates an inter-user dependency. Gligor [31] points out that inter-user dependency is a fundamental factor that enables denial of service to occur.

**Simple Core and Complex Edge** One of the design principles is that the Internet should keep the core networks simple and push any complexity into the end hosts [56]. This means that intermediate routers or the core routers only need to deliver IP packets without needing to understand services above the network layer. Most changes to the Internet are implemented at the end hosts. This encourages the development of new protocols and hence DDoS tools.

**Multi-path Routing** The Internet routing infrastructure is designed with the ability to route packets along alternative paths that bypass failing portions of the network. The possibility of multipath routing reduces routers’ ability to determine spoofed source address makes it more challenging to trace the origin of attack packets in the Internet.

**Decentralized Management** The Internet consist of numerous networks, interconnected to provide global access to the end users [56]. There is no central management in the Internet and each interconnected network is managed locally. This management approach allows the Internet has grown rapidly. However, this has also provided attackers with easy-to-access resources, and made cooperative DDoS attack defence across multiple subnetworks. Many DDoS defence approaches need to be deployed at numerous locations to be effective. However, it is extremely difficult to enforce global deployment in the Internet.
2.2.2 Other Challenges

In addition to the architectural challenges described above, there are several other challenges that make the Internet DDoS hard to defend against. A brief description of these challenges is given next.

**Difficulty of Distinguishing Malicious Requests** It is difficult to distinguish between malicious requests and legitimate ones. This is true for packets, network flows, transport layer segments, or application service request messages.

**DDoS Research Challenges** The advance of DDoS defence research has been interrupted by the lack of attack information, the absence of standardized evaluation, and the difficulty of large-scale testing [56]. They argue that very limited information about DDoS incidents are publicly available because organizations’ unwillingness to disclose the occurrence of an attack to avoid the business reputation of the victim. Without detailed analysis of real-world DDoS attacks, it is difficult to design good solutions to the problem.

**Lack of Common Characteristics of DDoS Streams** There are no common characteristics of DDoS streams that can be used for their detection and filtering. The attacks achieve the desired effect by increase volume of the attack packets, and can afford to vary all packet fields to avoid characterisation. In addition to this, attackers follow advances in the security field and adjust their tools to bypass new security defence systems.

**Lack of Cooperation Across Administrative Domains** The cooperation of distributed sources makes DDoS attacks hard to combat or trace back. At the same time, there is no cooperation between participating administrative domains (source, target and intermediate domains that carry DDoS traffic) that would enable fast, efficient and distributed response to the attack.
2.3 Types of Attack

The awareness of DDoS tricks and methods is a fundamental key in implementing defense system to combat attacks. Many experts have tried to classify the DDoS defense mechanisms in order to clarify them. This classification gives users an overall view of the situation and helps defense-mechanism developers cooperate against the threat. DDoS attacks are broadly classified into two categories: bandwidth depletion and resource depletion attacks [84].

2.3.1 Bandwidth Depletion Attacks

A bandwidth depletion attack or flooding based attack floods the victim network with unwanted traffic that prevents legitimate traffic from reaching the victim’s system. Bandwidth depletion attacks can be divided into two types: flood attacks and amplification attacks [84].
Flood Attacks

A flood attack involves zombies sending large volumes of traffic to a victim system, to congest the victim system’s network bandwidth with IP traffic. The victim system slows down, crashes, or suffers from saturated network bandwidth, preventing access by legitimate users. Flood attacks have been launched using both UDP (User Datagram Protocol) and ICMP (Internet Control Message Protocol) packets. When ICMP is used to flood a victim, the attacker uses the zombies to send large volumes of ICMP ECHO REPLY packets (ping) to the victim system.

Amplification Attacks

In an amplification attacker, as shown in Fig. 2.2, the zombies send messages to a broadcast IP address and this to cause all systems in the subnet reached by the broadcast address to send a reply to the victim system. In this attack, the broadcast IP address is used to amplify and reflect the attack traffic, and thus reduce the victim system’s bandwidth. The Smurf attack is an example of an amplification attack where the attacker sends packets to a network amplifier – a system that supporting broadcast addressing, with the return address spoofed to the victims IP address [5]. Another example is the DDoS Fraggle attack. A DDoS Fraggle attack is similar to a Smurf attack in that the attacker sends packets to a network amplifier. Fraggle is different from Smurf in that Fraggle uses UDP ECHO packets instead of ICMP ECHO packets.

2.3.2 Resource Depletion Attacks

A resource depletion attack ties up the resources of a victim system making the victim unable to process legitimate requests for services. The attacker in resource depletion attacks sends packets that misuse network protocol communications or are malformed.
2.3 Types of Attack

A Survey of DDoS Attacks and Defence Mechanisms

Figure 2.2: Simple Example for a Reflector Distributed Denial-of-Service Attack.

Figure 2.3: TCP SYN Attack Model.
Protocol Exploit Attacks

Misusing the *TCP SYN* (Transfer Control Protocol Synchronize) packet, misusing the *PUSH+ACK* protocol, and *IP spoofed* are examples of this type of attack. In a DDoS TCP SYN attack, as shown in Fig. 2.3, the attacker instructs the zombies to send bogus TCP SYN requests to a victim server in order to tie up the server’s processor resources, and hence prevent the server from responding to legitimate requests.

In a PUSH+ACK attack, the attacking agents send TCP packets with the PUSH and ACK bits set to one. These triggers in the TCP packet header instruct the victim system to unload all data in the TCP buffer and send an acknowledgement when complete. If this process is repeated with multiple agents, the receiving system cannot process the large volume of incoming packets and the victim system will crash. In IP spoofed attacks, the attacker sends DNS queries to the DNS server to reply to the victim as illustrated in Fig. 2.4.

![Figure 2.4: IP spoofed Type of DDoS Attack.](image)
2.4 DDoS Attack Mitigation Phases

A Survey of DDoS Attacks and Defence Mechanisms

Malformed Packet attacks

In this type of attack the attacker instructs the zombies to send incorrectly formed IP packets to the victim system in order to crash it. There are at least two types of malformed packet attacks. In an IP address attack, the packet contains the same source and destination IP addresses. This can confuse the operating system of the victim system and can cause the victim system to crash. In an IP packet options attack, a malformed packet may randomize the optional fields within an IP packet and set all quality of service bits to one so that the victim system must use additional processing time to analyse the traffic. If this attack is multiplied, it can exhaust the processing ability of the victim system.

2.3.3 Attack Taxonomy

A taxonomy allows us to reason about attacks and provides a classification system that ideally suggests ways to mitigate attacks by prevention, detection, and recovery. It can help risk management by identifying vulnerabilities and making attacker characteristics explicit. Based on the previous discussions of various attack types, we derived a taxonomy of denial of service attacks. Fig. 2.5 illustrates the taxonomy of denial of service attacks.

2.4 DDoS Attack Mitigation Phases

The denial of service mitigations can be divided into four phases: prevention, detection, source identification, and response. Prevention approaches attempt to eliminate the possibility of DDoS attacks or prevent the attack from causing any significant damage. Attack detection monitors and analyses events in a system to discover malicious attempts to cause denial of service. It is an important step before directing further actions to counter an attack. Attack source identification aims to locate the attack sources regardless of whether the source address field of malicious requests contain spoof...
2.4 DDoS Attack Mitigation Phases

- Resource Depletion
  - Malformed Packet Attack
    - IP Packet Option
    - IP Address
    - PUSH + ACK
  - Protocol Exploit Attack
    - TCP SYN
    - Fraggle
- Bandwidth Depletion
  - Amplification Attack
    - ICMP
    - UDP
  - Flood Attack

Figure 2.5: DDoS Attack Taxonomy.
2.4 DDoS Attack Mitigation Phases: A Survey of DDoS Attacks and Defence Mechanisms

Response mechanisms are usually initiated after the detection of an attack. Fig. 2.6 illustrates the taxonomy of defence mechanisms that we created to classify the existing DDoS solutions.

2.4.1 Attack Prevention

Denial of service prevention mechanisms aim to stop attacks before they actually cause damage. Prevention mechanisms include spoofed packet filtering, self-certifying addresses, and secure overlays.

Filtering Spoofed Packets

Almost all DDoS attackers spoofing IP address to hide the origin of an attack. Reflection and amplification attack techniques rely on IP address spoofing. Filtering mechanisms are designed to prohibit DDoS attack traffic with spoofed source addresses from reaching the target, by dropping packets with false IP addresses.

**Ingress/Egress Filtering** The purpose of ingress/egress filtering is to allow traffic to enter or leave the network only if its source addresses are within the expected IP address range. Ingress filtering refers to filtering the traffic coming into a network, and egress filtering refers to filtering the traffic leaving the network. Using network ingress filtering to counter DDoS attacks is introduced in RFC 2827 [25] as a best current practice.

**Route-Based Filtering** A route-based distributed packet filtering (DPF) approach to filtering out spoofed packet flows is proposed in [67]. DPF uses routing information to determine if a packet arriving at a router with an AS is valid with respect to its marked source/destination addresses. DPF uses information about the Border Gateway Protocol (BGP) [74] routing topology to filter traffic with spoofed source addresses.
2.4 DDoS Attack Mitigation Phases

Figure 2.6: Taxonomy of DDoS Mitigation Mechanisms.
2.4 DDoS Attack Mitigation Phases
A Survey of DDoS Attacks and Defence Mechanisms

Source Address Validity Enforcement (SAVE) Protocol To overcome the disadvantages of route-based filtering, Source Address Validity Enforcement protocol has been proposed in [49]. SAVE constantly propagates messages containing valid source address information from the source location to all destinations. Thus, each router along the way builds an incoming table that associates each link of the router with a set of valid source address blocks. When a packet arrives on an interface, a router consults its incoming table to determine whether this packet comes from the proper direction.

Hop-Count Filtering (HCF) HCF [34] introduces filtering packets with spoofed IP addresses using a method called Hop-Counter Filtering. They argue that although an attacker can forge any field in the IP header, they cannot falsify the number of hops an IP packet takes to reach its destination. They propose a method to infer this hop-count information from the Time to Live (TTL) value in the IP header. Using the TTL based hop-count computation method, a victim builds a hop-count to source IP address mapping table. When a victim receives a packet, it computes a hop-count for its IP address and compares it to the hop-count stored in the mapping table to identify address-spoofed packets. The advantage of HCF is that it requires deployment at the victim, which is much easier to deploy compared with network based filtering approaches. Moreover, a potential victim has a much stronger incentive to deploy defence mechanisms that the intermediate network service providers. However, HCF suffers from high false positives and false negatives.

IPv4 Source Guard IPv4 Source Guard [11] provides source IP version 4 (IPv4) address filtering on a Layer 2 port to prevent a malicious host from spoofing IP addresses. IP Source Guard snoops DHCP address assignments and uses static IPv4 source bindings to automatically configure each Layer 2 port to discard traffic if the source IP address is different from the IP address assigned to that port.

Passport Passport [50] is a source address validation framework and aims to ensure that
no host or AS can spoof the address space of an AS that deploys Passport. When a packet leaves its source AS, the border router stamps one Message Authentication Code (MAC) for each AS on the path to the destination. Each MAC is computed using a secret key shared between the source AS and the AS on the path. When the packet enters an AS on the path, the border router verifies the corresponding MAC using the secret key shared with the source AS.

Self-certifying Addresses

The accountability of IP addresses is the one of the main problems that needs to be addressed in preventing DDoS attack [60]. As explained in the following subsections, Host Identity Protocol and Accountable Internet Protocol are the proposed solution for the accountability of IP.

Host Identity Protocol (HIP) The HIP architecture [63] proposes a new namespace called Host Identity namespace and a new protocol layer called Host Identity Protocol [64] between the internet and transport layers. The Host Identity namespace consists of Host Identifiers (HIs), where a Host Identifier is an public key of an symmetric key-pair. HIP architecture is effective in preventing IP address spoofing and preventing DDoS.

Accountable Internet Protocol (AIP) AIP [8] is to provide Internet layer accountability using selfcertifying addresses. AIP is designed to address the lack of secure binding of a host to its IP addresses, and lack of secure binding of an AS number to the IP prefixes owned by that AS. The simplicity and DDoS attack prevention effectiveness of AIP make it a very attractive candidate for future generation Internet protocols. However, routing scalability and traffic engineering scalability of AIP’s at addressing scheme for use in the Internet is a very big concern.
2.4 DDoS Attack Mitigation Phases: A Survey of DDoS Attacks and Defence Mechanisms

**Secure Overlays**

Secure overlay prevention strategy protects a victim by routing traffic to a protected network through an overlay network that is built on top of IP. Since the overlay network only admits authorized users and it is difficult for attackers to cause DDoS on the protected servers or networks. A protected victim is separated from the Internet by hiding the IP addresses of the protected victim or by using distributed firewalls to filter all incoming traffic to the protected victim except for that traffic coming from the trusted nodes in the overlay network. *Secure Overlay Service* [39] and *Secure-i3* [7] are the two examples for secure overlay based DDoS defence.

**Secure Overlay Services (SOS)** prevent denial-of-service attacks on critical servers by routing requests from previously authenticated clients to those servers via an overlay network. All other requests are filtered by the overlay. SOS is a distributed system that offers excellent protection to the specified target at the cost of modifying the clients’ machines to be aware of the overlay and use it to access the target. A secure overlay is constructed by selecting a set of nodes distributed throughout the network, and are logically linked. As shown in Fig. 2.7, all the source traffic is verified by a secure overlay access point (SOAP). Authenticated traffic will be forwarded to a special overlay node called a beacon. The beacon then routes traffic to another special overlay node called a secret servlet for further authentication, and the secret servlet routes verified traffic to the victim. The identity of the secret servlet is only revealed to the beacon and remains a secret to the attacker.

SOS is robust against DDoS attacks because of the following reasons:

- if an access point is attacked, the source can choose an alternate access point.
- if a secret servlet’s identity is discovered by attackers the protected site can choose an alternate set of secret servlets.

The main drawback of SOS is that it is not suitable for a service available to the public, such as a web server. The clients of an SOS-protected sever must register their identity
before they can connect and a large numbers of overlay nodes are required to make the system resilient to DDoS attacks. Also, SOS offers no protection from insider attacks. Furthermore the deployment is also relatively difficult. Secure Overlay Forwarding System (SOFS) [93] is an extension to SOS, aimed at resisting intelligent DDoS attacks. SOS uses a three layer architecture while SOFS utilises more than three layers and thereby provides more protection to the victim. Deployment of SOFS requires many changes in the network system.

Secure-i3 is the name of an overlay network solution. End-to-end communication between two hosts is routed within the overlay based on identifiers instead of IP addresses. Secure-i3 utilizes the i3 [85] overlay network as its means to hide IP addresses of end hosts. There are multiple issues with Secure-i3. The assumption that the IP addresses of end hosts and portions of i3 nodes are unknown to the attacker. Attacker can obtain such information through various techniques such as footprinting and scanning. Secure-i3 requires large number of powerful i3 nodes with ample bandwidth to be deployed in the Internet, adding a very high extra cost. Since routing in Secure-i3 overlay network creates
2.4 DDoS Attack Mitigation Phases: A Survey of DDoS Attacks and Defence Mechanisms

extra level of indirection, it increases end-to-end delay and decreases network goodput.

2.4.2 Attack Detection

The DDoS detection strategies fall into three types: **Signature-based** detection, **Anomaly-based** detection, and **Hybrid systems**. Signature-based methods search for patterns or signatures in observed network traffic that match known attack signatures from a database. Anomaly-based methods compare the parameters of the observed network traffic with normal traffic and hence it is possible for new attacks to be detected. However, in order to avoid false alarms, the model of normal traffic must always be kept updated and the threshold of categorizing an anomaly must be properly adjusted. Finally, hybrid systems combine both these methods. Hybrid systems usually update their signature database with attacks detected by anomaly detection. Table 2.1 shows a comparison.

<table>
<thead>
<tr>
<th>Signature-based Attack Detection</th>
<th>Anomaly-based Attack Detection</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Detect attacks by matching a pattern or signature of known attacks.</td>
<td>• Detect attacks by monitoring the changes of traffic volume or features.</td>
</tr>
<tr>
<td>• Reliably and easy attack detection for an already known attacks.</td>
<td>• Hard to set up a threshold for attack flows.</td>
</tr>
<tr>
<td>• Signature database must always be kept up-to-date in order to retain the reliability of the system</td>
<td>• Unknown attacks can be discovered.</td>
</tr>
<tr>
<td></td>
<td>• Models update need to be updated with changes of the system.</td>
</tr>
</tbody>
</table>
Signature-based Attack Detection

A signature-based defence system tries to match a pattern or signature that can allow the detection of a specific to known attacks. Mostly this type of detection explores the content of packets and make attack decision. The advantage of these methods is that they can easily and reliably detect known attacks, but they cannot recognize new attacks. Moreover, the signature database must always be kept up to date in order to retain the reliability of the system. Bro [68] is a network intrusion detection system for detecting network intruders in real-time by monitoring a network link. The Bro system is high-speed, for large volume monitoring, with real-time attack detection mechanisms. Bro is conceptually divided into an event engine that maintains state for each connection based on source IP, source port, destination IP and destination port. Bro requires manual creation of attack signatures and event handling scripts, which requires meticulous effort and strong intrusion detection background from security administrators.

The Snort [76], relies on signatures that define byte patterns of known attack packets. They provide a low false positive rate but are not able to detect previously unknown attacks. Snort is an open-source light-weight network intrusion detection and prevention tool and widely deployed network intrusion detection and prevention systems worldwide.

Anomaly-based Attack Detection

Anomaly based DDoS attack detection mechanisms analyse the normal behavior in a system and aim to detect attacks via identifying significant deviation from the normal behavior. Mostly it detects attacks by monitoring the changes of traffic volume or features. It is efficient toward flooding DDoS attack with detectable features. Compared to signature based detection approaches, it can discovery previously unseen attacks. Anomaly based approaches faces a challenge when determining the threshold for new attack behavior.

Mechanisms that deploy anomaly detection have a model of normal system behavior,
such as a model of normal traffic dynamics or expected system performance. The current state of the system is periodically compared with the models to detect anomalies. The advantage of anomaly detection over pattern detection is that unknown attacks can be discovered. However, anomaly-based detection has to address two issues:

**Threshold setting** Anomalies are detected when the current system state differs from the model by a certain threshold. The setting of a low threshold leads to many false positives, while a high threshold reduces the sensitivity of the detection mechanism.

**Model update** Systems and communication patterns evolve with time, and models need to be updated to reflect this change. Anomaly based systems usually perform automatic model updates using statistics gathered at a time when no attack was detected. This approach makes the detection mechanism vulnerable to increasing rate attacks that can mistrial models and delay or even avoid attack detection.

Anomaly detection systems like NSOM [44] and [23] monitor network traffic and search for anomalous behavior by applying neural networks or threshold-based mechanisms. They are able to detect previously unknown attacks at the expense of a higher false positive and false negative rate.

### 2.4.3 Attack Source Identification

After detecting an attack, the response to the attack is more effective close to the attacking source [75]. Unfortunately, there is no easy way to trace IP traffic to its source. This is due to two characteristics of the IP protocol. The first is that the 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. In this section we discuss some source identification methods.
Probabilistic IP traceback (PPM) PPM scheme [78] traces anonymous packet flooding attacks in the Internet back to their source. The main idea of probabilistic packet marking is that each router probabilistically encodes distance to receiver and router address information into a marking field in the packet header, and the receiver reconstructs the path that a packet traveled from the encoded information.

ICMP traceback message A new type of ICMP message called iTrace packet is utilized in [13] to help receiver reconstruct the path that packets take through the Internet. Each router generates a traceback message with a very low probability for each packet it forwards, and sends the message to the receiver. The advantage of this scheme is that it is simple and easy to implement. However, it needs digital signatures to protect the integrity of the information contained in a traceback message.

Single-Packet IP Traceback (SPIE) SPIE [82] identifies the source of a particular IP packet given a copy of the packet to be traced. It finds IP destination and an approximate time of receipt. SPIE requires all routers to keep a hash digest of recently forwarded packets. When a traceback is needed, a query is dispatched to SPIE which in turn queries routers for packet digests of the relevant time periods. The main advantage of SPIE over PPM and ICMP traceback is that the receiver does not need to receive large number of attack packets to traceback to the attack source. However SPIE requires a large amount of memory for storing packet digests at routers.

Entropy variation scheme The entropy variations of network traffic [106] is used to implement a traceback scheme. The difference in entropy values between normal traffic and the traffic under DDoS attack is used to detect the attack. Once it is detected, the traceback is initiated through a pushback tracing procedure. The proposed scheme has an advantage over traditional packet marking schemes in terms of scalability and storage requirements in victim or intermediate routers. The method stores only short-term information of traffic entropy in order to detect the DDoS attack. The authors also claim that the method is able to implement accurate
2.4 DDoS Attack Mitigation Phases

A Survey of DDoS Attacks and Defence Mechanisms

traceback in a large-scale DDoS attack scenario within a few seconds.

**Cloud security defence** A mechanism [19] which involves the determinist packet marking and service oriented traceback architecture is used to protect from the XML DoS attack. This is a track back mechanism which identifies the source of the attack and filters it. Cloud traceback and cloud protector are used in this scheme. Cloud trace back marks the incoming packets and the cloud protector filters the packets. The cloud protector is a trained back propagation neural network.

**Traceback by using new information metrics** The generalized entropy metric and the information distance metric were proposed in [98] to detect low rate DDoS attacks. This approach has advantages in terms of detection speed and false positive rate. The impracticality of this approach is rather unfortunate with successful implementation relying on gaining full control of all the routers in the network.

### 2.4.4 Attack Reaction Mechanisms

After undertaking attack detection and attack source identification techniques in a timely manner, further actions are necessary to counter the attack. Reaction or response mechanisms are usually initiated after the detection of an attack to eliminate or minimize the impact of the attack. The filtering, rate-limiting and resource management are the popular reaction mechanisms.

**Filtering and Rate-limiting**

Filtering and rate-limiting mechanisms use the characterization of malicious traffic that is provided by detection mechanisms to filter out or rate-limit attack flows. Rate-limiting is usually used in cases where detection has many false positives or cannot precisely characterize the attack traffic.

The rate limiting algorithm [103] is based on the fact that the traffic rate at the victim
2.4 DDoS Attack Mitigation Phases

end is normal if traffic rates forwarded to the victim by all level-\( k \) routers are normal. One drawback is that it uses equal rate limit for all level-\( k \) routers and hence it is unfair for those routers that forward little or no attack traffic. Collateral damage for the legitimate traffic will be unavoidable in these routers.

The overlay-based distributed defence framework [35] detects attacks at the victim end. During source finding, SPIE is used. To control attack traffic at the source end, it combines the history of a flow into rate limit calculation by defining a reputation argument. A spoofing DDoS attack can make the flow-based rate limit algorithm ineffective. Moreover, the realization of the framework needs a relatively huge modification of current network.

Packet filtering [87] is usually accomplished at routers based on clearly defined attack signatures, e.g., obviously incorrect source addresses. The common drawback with packet filtering is that it needs to be deployed widely in order to protect the victim, and attack traffic cannot be filtered if it uses packets that request legitimate services [99].

Two methods are:

**Pushback high-bandwidth aggregates (ACC)** ACC [51] is for controlling high bandwidth aggregates in the network, where an aggregate is a collection of packets from one or more flows that have some property in common. ACC includes a local mechanism for identifying and controlling an aggregate at a single router, and a cooperative pushback mechanism in which a router can ask upstream routers to control an aggregate. ACC is triggered only when a link experiences sustained severe congestion, which can be determined by looking for an extended high packet loss rate period. After a severe congestion is detected, ACC identifies the aggregates that are responsible for the congestion. Unfortunately to pushback each source in a large distributed DDoS attack is likely to be relatively expensive in terms of network state, and requires each source to be identified and pushed back individually.

**StopIt** This is a filter-based DDoS defence framework [50]. StopIt aims to stop the un-
wanted traffic destined to a receiver without inflicting damage on legitimate hosts sending traffic to that receiver. A StopIt server learns the addresses of other StopIt servers by listening to Border Gateway Protocol (BGP) [74] updates.

**Resource Management**

A fundamental problem of the Internet with regard to DDoS attacks is that the receiver has not control over who can send how much traffic to it. A misbehaving sender can simply ignore any congestion control signals and send traffic at the maximum possible rate. Resource Management based DDoS response solutions aim to enable the receiver to stop misbehaving senders.

Approaches proposed in [29,37,46,83,110] illustrate resource accounting mechanisms. In fact, DDoS is a resource overloading problem, resource accounting approaches can address it. One drawback of these mechanisms is that clients must be aware of the defence and install special software, enabling them to reply to legitimacy tests. Another drawback is that resource accounting requires keeping state per user, thus storage requirements grow with the number of legitimate users. This indicates that the system must be installed at the end network. As discussed in Section 1.2.2, victim-end defence cannot handle high-volume attacks that overwhelm the defence system.

Resource multiplication mechanisms provide an abundance of resources to counter DDoS threats. For example a pool of servers with a load balancer and installs high bandwidth links between itself and upstream routers. The other approach is to acquire resources dynamically once the attack has been detected [101]. These approaches essentially raise the bar on how many machines must participate in an attack to be effective. While not providing perfect protection, for those who can afford the costs resource multiplication has often proved sufficient. Another approach is the use of Akamai services for distributed Web site hosting [2]. User requests for a Web page hosted in such a manner are redirected to an Akamai name server, which then distributes the load among
multiple, geographically distributed Web servers hosting replicas of the requested page. Stateless Internet Flow Filter and Traffic Validation Architecture are the two well known resource management schemes:

**Stateless Internet Flow Filter (SIFF)** SIFF [100] selectively stops unwanted attack flows from reaching the recipient’s network. SIFF assumes two Internet packet classes: privileged packets that are subject to receiver’s control and unprivileged packets that are used in legacy traffic and in SIFF handshake. The SIFF handshake protocol is used by senders to obtain capabilities to send privileged traffic. Sender starts the handshake process by sending a capability request packet with its capability initialized set to zero.

**Traffic Validation Architecture (TVA)** TVA [102] is a network architecture that limits the impact of DDoS floods. The TVA architecture builds on the previously proposed work [9] on capabilities, and tries to address the limitations of previous capability mechanisms such as SIFF. TVA aims to address the flooding attack against capability setup channel and attacks that flood the receiver using already acquired capabilities. There are several weaknesses of TVA scheme. For example, TVA assumes that the receiver can distinguish the malicious sender from the legitimate ones. Hence, TVA relies on an attack identification method to filter malicious traffic, and its DDoS prevention effectiveness depends on the accuracy of the identification method.

### 2.5 DDoS Defence Proposals

#### 2.5.1 Victim-End Defence

Most systems for combating DDoS attacks work on the victim side. As discussed in Section 1.2.2, victim-end defence cannot provide complete protection from DDoS attacks because the defence system may itself be overwhelmed by the attack traffic.
2.5 DDoS Defence Proposals

A Survey of DDoS Attacks and Defence Mechanisms

Several DDoS defence systems [110] perform anomaly detection (usually at the victim network) by observing numerous traffic parameters and defining a range of allowed values based on the analysis of packet trace data. The attack response is to impose a non-selective fixed rate limit to offending streams, thus likely damaging legitimate traffic.

A path signature (PS) based victim-end defence system has been proposed in [41]. The system requires all routers to flip selected bits in the IP identification field for all incoming packets. Based on these marking bits, a unique PS can be generated for all packets from the same location. At the victim end, the defence system separates traffic based on PS of each packet and detects DDoS attacks by monitoring anomalous changes of traffic amount from a PS. Then, a rate limit value will be set up on this traffic. However, it is hard to detect DDoS attacks if PS diversity is much greater than real router diversity of incoming traffic. Moreover, it is possible that a PS has been changed after an attack has been detected. For this situation, collateral damage for the legitimate traffic cannot be avoided.

A defence framework for flooding based DDoS attacks [105] detects attacks at the victim using distance based DDoS defence systems techniques and it responds to the attacks at the source ends. The scheme has two issues: communication with the source end during the attack time, and detecting the attack at the victim cannot prohibit damage to the victim.

2.5.2 Intermediate Network Defence

Distributed defence system at edge routers, or in the core routers has a definite advantage over single point defence [54]. The idea of cooperative defence against DDoS attacks has been proposed in a number of projects.

DefCOM [65] is a distributed cooperative system which combines diverse defence systems for cooperative response to DDoS. DefCOM contains three types of nodes, namely: Alert nodes, Core nodes and Classifier nodes. Alert nodes detect attacks close to the tar-
2.5 DDoS Defence Proposals

get and propagate an alarm. Core nodes and Classifier nodes handle different kinds of traffic, e.g. Classifier nodes discriminate between legitimate and attack packets. Different network administration policies make hard on deployment of this defence system.

MANAnet [3] forms cooperative neighbourhoods of defence nodes around the victim. These nodes stamp packets that they forward, encoding the path to the victim in the packet header. The router at the victim then offers a fair share of resources to each encoded path. Source networks can further deploy the Reverse Firewall to prevent outgoing attacks. MANAnet requires contiguous deployment of defence nodes near the victim and changes to the IP protocol to facilitate packet stamping. Both of these features are likely to hinder wide deployment.

COSSACK [66] is a distributed approach to DDoS detection and response and it uses a software subsystem known as a watchdog. Each network should deploy its own watchdog at the egress router. Watchdogs detect attacks using existing intrusion detection systems and share their decision with other watchdogs which increases detection confidence. Multicast communication is used to alert other watchdogs. COSSACK solely relies on other defence systems’ techniques to make local decisions and multicast communication consumes more network resources than the gossip protocol that we used in our defence system.

IDIP [79] is an application-level protocol that coordinates detection and responses of multiple intrusion detection systems. IDIP nodes are organised into neighbourhoods and communities. Coordinated detection, attack tracing and response within a community are managed by a component called a Discovery Coordinator. IDIP assumes contiguous deployment of neighbourhoods that share information, and thus its ability to prevent attacks is limited in a partial deployment scenario.

Due to IP address spoofing, there are a lot of fake IP addresses used in DDoS attacks. SIM [70] is a source IP address monitoring based detection scheme. SIM is deployed at the edge router that provides Internet access to the subnetwork in which the target resides. SIM assumes that the set of source IP addresses that is seen during normal op-
2.5 DDoS Defence Proposals

A Survey of DDoS Attacks and Defence Mechanisms

Operation is somewhat constant, and most of the previously unseen addresses that showed up during an attack belong to attackers. By using a pre-built database of IP addresses, SIM monitors the proportion of previously unseen source IP addresses, and detects any abrupt changes using the cumulative sum (CUSUM) sequential change point detection algorithm. An abrupt change of the proportion of new source IP addresses is labeled as a strong indication of a DDoS attack. They also try to improve the detection accuracy by simultaneously monitoring the traffic rate per IP address.

The Active Security System (ASSYST) \[16\] consists of defence nodes deployed at edge networks. Once the attack is detected, the nodes discover their peers using probe messages, and organise themselves into a peer-to-peer network. The blocking requests are then propagated to upstream peers and a response is installed as close to the sources as possible. ASSYST does not provide separation of legitimate from attack flows, and is likely to reduce collateral damage.

Distributed Change-point Detection (DCD) \[18\] uses change aggregation trees (CAT) to detect DDoS by collaborating between different domains in an ISP. Each domain constructs a CAT based on the traffic superflow of the routers. By collaboration between the CATs of each domain a global CAT tree is built. Once the global CAT tree exceeds a preset threshold then an attack is declared. Because the cat threshold value is not the same to all type of attach, setting the ideal threshold value is challenging in this defence system whereas in our defence system we makes decision based on victim’s capacity.

The FireCol \[28\] is a Collaborative Protection Network for the Detection of Flooding DDoS Attacks deployed at the Internet service providers (ISPs) level. This defence system forms virtual protection rings around the hosts to defend and collaborate by exchanging selected traffic information. However as in Secure Overlay Services (SOS), the client must register in advance to access FireCol protected Internet services and attack information sharing is insecure in the defence system.

Zhang et al. \[109\] proposed a distributed cooperative mitigation approach which functions in two stages. In the first stage, distributed defence nodes locally detect at-
A Survey of DDoS Attacks and Defence Mechanisms

2.5 DDoS Defence Proposals

attacks by using variety of existing attack detection tools such as Snort [76]. In the second stage, in order to enhance the accuracy of global detection decision, the local decisions are shared among defence nodes using a gossip based peer to peer communication. Gossip based protocols are robust and resilient in an attack environment and also less deployment complex. However they use signature based attack detection method which can not detect new attacks.

The distributed detection and response scheme proposed in [45] uses a Stub Agent (SA) deployed in a local ISP network detects anomalous changes of the traffic rate. Source-end SAs and transit network agents (TA) lower attack traffic in the network by setting different rate limits. Unfortunately, DDoS detection based on disproportionate TCP packet rates cannot cover proportional attacks, attacks with randomised forged IP addresses originating from a single machine, and attacks that use many zombies. Also, rate limiting at core routers definitely lowers the performance of the whole network. The entire scheme lacks an effective method to reconstruct the attack path when a spoofing attack happens. A more serious problem is collateral damage for the legitimate traffic.

The objective of the proposal in [77] is to utilize a global architecture for an efficient Intrusion Detection System where participants can exchange information following a P2P model. Thus, the solution proposed is designed to elaborate the defence against these large scale attacks by the correlation of the suspicious evidence provided and stored by the architecture entities from different geographical locations. This scheme uses existing defence strategies to conclude local attack decisions.

### 2.5.3 Source-end Defence

A DDoS defence system deployable at networks where most of the attack sources are located is known as source-end defence system. D-WARD and MULTOPS are two examples of Source-end defence systems.

D-WARD [58] is a source-end DDoS defence system that autonomously detects and
stops attacks originating from the attack source network. Attacks are detected by monitoring two-way traffic flows between the network and the rest of the Internet. Monitored flows are periodically compared with predefined models of normal traffic, and those flows classified as part of a DDoS attack are rate-limited. The TCP, ICMP, UDP normal traffic models are used as the TCP, ICMP, UDP normal traffic models. Once an attack flow is identified, it will be controlled under a rate limit value. Although D-WARD can detect some attacks at the source end, the detection may be error prone due to lack of communication between the source and the victim-end, and coordination among the source-end defence systems.

MULTOPS [30] is a heuristic and a data-structure that network devices can use to detect DDoS attacks. Each network device maintains a multi-level tree, monitoring certain traffic characteristics and storing data in nodes corresponding to subnet prefixes at different aggregation levels. The tree expands and contracts within a fixed memory budget. The attack is detected by abnormal packet ratio values, and offending flows are rate-limited. The system is designed so that it can operate as either a source-end or victim-end DDoS defence system. Non-TCP flows in a system using MULTOPS can either be misclassified as attack flows, or recognized as special and rate-limited to a fixed value. In the first approach, harm is done to a legitimate flow, while in the second approach, a sufficiently distributed attack can still make use of the allowed rate to achieve the effect.

Table 2.2 shows a comparison of different distributed defence systems.

### 2.6 Summary

From early on, DDoS attacks have attracted a lot of attention in research and commercial communities. Many DDoS attack solutions aim at exploiting a certain feature of current attacks to prevent them. Unfortunately, the attackers carefully follow developments in the security field and are able to bypass the security system. We present here related works in three categories namely Intermediate Defence, Source-End defence and
Victim-End system and discussed strong and weak points of each method. Single point deployment defence systems or the Victim-end defence systems can not achieve successful defence against distributed denial of service attacks. The DDoS problem requires a distributed solution in which defence nodes are located throughout the Internet and cooperate to achieve better overall defence. The idea of cooperative defence against DDoS attacks has been proposed in a number of projects. Anomaly based defence systems are preferred over signature-based defence system as they can detect new unknown attacks.

### Table 2.2: Comparing different distributed defence System.

<table>
<thead>
<tr>
<th>Defence Systems</th>
<th>Deployment</th>
<th>Detection</th>
<th>Response</th>
<th>Robustness</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOS</td>
<td>Source/Victim</td>
<td>Filtering</td>
<td>Rate limiting</td>
<td>Strong</td>
<td>Difficult</td>
</tr>
<tr>
<td>DefCOM</td>
<td>Throughout the network</td>
<td>Traffic tree discovery</td>
<td>Rate limiting</td>
<td>Weak</td>
<td>Difficult</td>
</tr>
<tr>
<td>MANANet</td>
<td>Victim end as a group</td>
<td>PEIP</td>
<td>Rate limiting</td>
<td>Weak</td>
<td>Difficult</td>
</tr>
<tr>
<td>COSSACK</td>
<td>Source/Victim</td>
<td>Spectral Analysis</td>
<td>Dropping all packets</td>
<td>Weak</td>
<td>Easy</td>
</tr>
<tr>
<td>ACC</td>
<td>Throughout the network</td>
<td>Congestion detection</td>
<td>Rate limiting</td>
<td>Weak</td>
<td>Difficult</td>
</tr>
<tr>
<td>ASSYST</td>
<td>Throughout the network</td>
<td>Intrusion detection</td>
<td>Dropping all packets</td>
<td>Weak</td>
<td>Difficult</td>
</tr>
</tbody>
</table>
Chapter 3

Theoretical Model for Gossip Detector

3.1 Overview

In our work, attack detection at an early stage is the most significant design requirement for a defence system. Even though some existing defence systems detect attacks successfully, preventing the attacks are difficult or impossible task because the attack detections are not early enough to execute the reaction mechanism [90]. In fact all attacks can be detected after some time by a defence system or by network administrators/users. However the early attack detection is essential to avoid service disturbance or damage to the victim. In this chapter we describe an abstract attack-detector and we define an attack detection system as consisting of a number of such detectors. We define some fundamental performance metrics which are used to characterise the performance of this abstract system – probability of attack detection, probability of false alarm and detection delay.

The chapter is organized in the following way:

- We define a basic analytical model that takes into account probability of attack detection probability, $p_{ad}$, and probability of false alarm, $p_{fa}$, but not attack detection delay. We show the tradeoff between $p_{ad}$ and $p_{fa}$.

- We modify the basic model to include attack detection delay, $\tau$. We then show its relationship to $p_{ad}$ and $p_{fa}$.
3.2 Abstract Defence System

We consider a network consisting of $N$ routers, connected in a static network. For convenience (and without loss of generality) we have at most one host attached to each router. One of the hosts is selected as a victim. The word victim refers to the host that the system is intending to protect and will be the target of the attack. We consider $n < N$ defence nodes or abstract attack-detectors. Fig. 3.1 illustrates a simple system model.

Let the attack start at time $t_0$. In this chapter we completely abstract away the notion of traffic. We consider each attack-detector, $i$, to either detect the attack with probability $P_{ad}^i$ given that an attack is underway, or to detect an attack with probability $P_{fa}^i$ given that an attack is not underway. In the basic model we are not concerned with time; while in the extended model we are.

We are interested in a collaborative defence system and so we consider that if more than a certain number of defence nodes, $n_c < n$, raise attack alerts then the victim is
considered as “under attack” at the time.

### 3.3 Basic Analytical Model

#### 3.3.1 Probability of Attack Detection

The probability of attack detection of the Gossip Detector, $P_{ad}$, is equal to the probability of any $n_c$ defence nodes successfully detecting attack given that there is an attack underway. As shown in Fig. 3.2, we define $P_{iad}^i$ to be the probability of attack detection of $i^{th}$ defence node, $i = 1, \ldots, n$, given that there is an attack underway. A Poisson binomial distribution with $n$ number of independent Bernoulli trails and the trail probabilities $P_{iad}^i$, $i = 1, \ldots, n$, can be used to compute $P_{ad}$.

For example to calculate the probability of attack detection of the Gossip Detector with $n = 3$, $n_c = 2$ and $P_{iad}^1$, $P_{iad}^2$, $P_{iad}^3$ be the probability of attack detection of each defence node...
3.3 Basic Analytical Model

Theoretical Model for Gossip Detector
during an attack then:

\[ P_{ad} = P_{ad}^1 P_{ad}^2 P_{ad}^3 + P_{ad}^1 P_{ad}^2 (1 - P_{ad}^3) + P_{ad}^1 (1 - P_{ad}^2) P_{ad}^3 + (1 - P_{ad}^1) P_{ad}^2 P_{ad}^3 \]

To simplify our analysis we assume that each defence node detects attack with equal probability, \( p \), i.e. \( P_{ad}^i = p \) for all \( i \), where \( p \) is a constant value. This assumption will not affect analytical results as the probability of attack detections at each defence node will be the same in an ideal function of the Gossip Detector. As illustrated in the Fig. 3.3, when the individual probability of attack detection \( P_{ad}^i \) increases the probability of attack detection of Gossip Detector increases. We can then write:

\[ P_{ad} = \sum_{k=n_{\varepsilon}}^{n} \binom{n}{k} p^k (1 - p)^{n-k} \tag{3.1} \]

Fig. 3.3 shows how \( P_{ad} \) changes as \( p \) changes, for \( n = 20 \) and various values of \( n_{\varepsilon} \).

Figure 3.3: Probability of attack detection versus probability attack detection for individual defence node, \( n = 20 \) and \( n_{\varepsilon} = 5, 10, 15 \).

Choosing the probability of attack detection for individual defence nodes as \( p = 0.9 \)
and threshold value $n_\varepsilon = \frac{n}{2}$. Fig. 3.4 shows the variation of $P_{ad}$ versus $n$. Clearly when the number of defence nodes increases, the probability of attack detection increases. This indicates the our defence system can improve probability of attack detection by increasing the number of defence nodes.

![Figure 3.4: Comparing the variation of probability of attack detection with number of defence nodes in Gossip Detector. $p = 0.9$.](image)

### 3.3.2 Probability of False Alarm

A false alarm is defined as including both false positives and false negatives. An individual false positive occurs at a defence node when there is no attack underway and the defence node raises an attack alert. An individual false negative happens at a defence node when there is an attack underway and the defence node fails to raise an attack alert. An individual false alarm does not constitute in itself a false alarm overall, since the overall system alarm is an aggregate of the individual defence nodes.
The Fig. 3.2 indicates the possibilities of false alarms at a defence node. That is:

\[ P(\text{False negative}) = q' (1 - p') \]

\[ P(\text{False positive}) = (1 - q') p' \]

The probability of false alarm at defence node \( i \), \( P_{fa}^i \), is equal to probability of false positive plus the probability of false negative:

\[ P_{fa}^i = q' (1 - p') + (1 - q') p' \]

The probability of false alarm of \textit{Gossip Detector} (\( P_{fa} \)) is equal to the probability of any \( n_e \) number of defence nodes raising false alarms out of \( n \) defence nodes. For simplicity, we assume that all defence nodes have the same probability of false alarm, \( q \), i.e \( P_{fa}^i = q \) for all \( i \), where \( q \) is a constant value. Using the Eq. 3.2 we calculate the probability of false alarm:

\[ P_{fa} = \sum_{k=n_e}^{n} \binom{n}{k} q^k (1 - q)^{n-k} \]  \hspace{1cm} (3.2)

Fig. 3.5 shows how \( P_{fa} \) changes as \( q \) changes, for \( n = 20 \) and various values of \( n_e \).

By choosing \( q = 0.1 \) and \( n_e = 0.5 n \), we plot the Fig. 3.6 to see the variation of \( P_{fa} \) as \( n \) changes. Fig. 3.6 clearly indicates that when the number of defence nodes increases, \textit{Gossip Detector} probability of false alarm decreases.

### 3.3.3 Tradeoff

Generally the system shows a trade-off between probability of attack detection and probability of false alarm. By choosing \( p = 0.9 \) and \( q = 0.1 \), we plot Fig. 3.7 to see the tradeoff between \( P_{ad} \) as \( P_{fa} \). Also by choosing \( p = 0.75 \) and \( q = 0.25 \), which is a worse detection system, we plot Fig. 3.8 to see the tradeoff between \( P_{ad} \) and \( P_{fa} \).
3.3 Basic Analytical Model

Figure 3.5: $P_{fa}$ with defence node false alarm probability $q$, $n = 20$ and $n_{\varepsilon} = 5, 10, 15$.

Figure 3.6: Comparing the trend of $P_{fa}$ with number of defence nodes in Gossip Detector. $q = 0.1$. 

59
3.3 Basic Analytical Model

Theoretical Model for Gossip Detector

Figure 3.7: Tradeoff between $P_{ad}$ and $P_{fa}$. Where the data points denote $\frac{n_0}{n}$ ratio. $p = 0.9$, $q = 0.1$ and $n = 10, 12, 14, 16, 18, 20, 22$, right to left.

Figure 3.8: Tradeoff between $P_{ad}$ and $P_{fa}$. Where the data points denote $\frac{n_0}{n}$ ratio. $p = 0.75$, $q = 0.25$ and $n = 10, 12, 14, 16, 18, 20, 22$, right to left.
3.4 Extended Analytical Model

In order to analyse the detection delay of the system, we let $T_i$ be the time at which the attack was detected at defence node $i$, given that the attack started at time 0. If we consider $P(T_i)$ as the probability of attack detection with detection delay $T_i$ at node $i$, then the cdf,

$$F_i(t) = P(T_i \leq t)$$

(3.3)

is the probability of attack detection with detection delay less than or equal to time $t$ at defence node $i$.

The detection delay of *Gossip Detector* is $\tau = t$ if at least $n_c$ defence nodes detect the attack on or before time $t$. A Poisson binomial distribution with $n$ independent Bernoulli trials and trial probabilities $F_i(t)$, can be used to compute the probability of attack detection by time $t$. The following Eq. 3.4 gives the probability of attack detection of *Gossip Detector* by time $t$, for general distributions:

$$P_{ad}(t) = \sum_{l=n_c}^{n} \sum_{A \in S_l} \prod_{i \in A} F_i(t) \prod_{j \in A^c} (1 - F_j(t)),$$

(3.4)

where $S_l$ is the set of all subsets of $l$ integers that can be selected from $\{1, 2, 3, ..., n\}$, and $A^c$ is the complement set.

Generally, as we increase the number of defence nodes in the defence system, the attack detection delay will be reduced. For a simple analysis, we assume that $P(T_i)$ is the same probability distribution at all defence nodes and therefore also $F_i(t)$ will be the same distribution for all $i$; $F_i(t) = F(t)$ $\forall i$. If we assume that $P_i(t)$ follows an exponential distribution with parameter $\mu$,

$$P_i(t) = \mu e^{-\mu t},$$
3.4 Extended Analytical Model

Theoretical Model for Gossip Detector

then Eq. 3.4 becomes:

\[ P_{ad}(t) = \sum_{k=n_e}^{n} \binom{n}{k} (F(t))^k (1 - F(t))^{n-k}. \]  \hspace{1cm} (3.5)

In order to compare the detection delay with Gossip Detector containing different number of defence nodes, we define the pdf of \( P_{ad}(t) \) as,

\[ p_{pdf}^{ad}(t) = \frac{P_{ad}(t)}{dt}, \]  \hspace{1cm} (3.6)

Then,

\[ E_{ad}[t] = \int_{0}^{\infty} p_{pdf}^{ad}(t) t \, dt. \]  \hspace{1cm} (3.7)

3.4.1 Probability of Attack Detection by Time \( \tau \)

In Figs. 3.9 and 3.10 we show Eq. 3.5 for \( \mu = \{ \frac{1}{10}, \frac{1}{15} \} \) resp. As the mean time for an individual node to detect an attack increases, the ability for the system as a whole to detect the attack increases correspondingly.

Fig. 3.11 shows the effect of \( n_e \) increases, i.e. as we require more nodes to agree that an attack is underway, for a fixed size defence network.

3.4.2 Detection Delay

we plot Fig. 3.12 using Eq. 3.7 to see the tradeoff between \( E_{ad}[t] \) and \( d \). As shown in Fig. 3.12, the average detection delay reduces as the number of defence nodes increase in Gossip Detector. We expect this detection delay behaviour with number of defence nodes in our real simulation experiments.
3.4.3 Probability of False Alarm by Time $\tau$

Similarly to the basic analytic model, we also consider that Gossip Detector will generate false alarms. In this case however we consider a false alarm rate $\lambda$ and use the same
Figure 3.11: Compares the probability of attack detection when the number of defence nodes varies in the defence system. $n_{\varepsilon} = 4, 8, 12, 16, 20$ and $n = 20$.

Figure 3.12: Compares the average detection delay when the number of defence nodes varies in Gossip Detector.

approach to compute the mean time to false alarm for Gossip Detector. In other words, we consider that Gossip Detector will generate some number of false alarms per time unit. In
order to analyze the false alarm rate of Gossip Detector we let $T_i$ be the time at which the false alarm raised at defence node $i$, given that the network monitoring time started at time 0. If we consider $Q(T_i)$ as the probability of false alarm with alert delay $T_i$ at node $i$, then the cdf,

$$H_i(t) = Q(T_i \leq t)$$

is the probability of false alarm with false alarm delay less than or equal to time $t$ at defence node $i$.

The false alarm delay of Gossip Detector is $\tau = t$ if at least $n_\epsilon$ defence nodes raise a false alarm on or before time $t$. A Poisson binomial distribution with $n$ independent Bernoulli trials and trial probabilities $H_i(t)$, can be used to compute the probability of false alarm by time $t$. The following Eq. 3.9 gives the probability of false alarm of Gossip Detector by time $t$, for general distributions:

$$P_{fa}(t) = \sum_{l=n_\epsilon}^{n} \sum_{A \in S_l} \prod_{i \in A} H_i(t) \prod_{j \in A^c} (1 - H_j(t)),$$

where $S_l$ is the set of all subsets of $l$ integers that can be selected from $\{1, 2, 3, ..., n\}$, and $A^c$ is the complement set.

Generally, as we increase the number of defence nodes in Gossip Detector the false alarm delay will be reduced. For a simple analysis, we assume that $Q(T_i)$ is the same probability distribution at all defence nodes and therefore also $H_i(t)$ will be the same distribution for all $i$; $H_i(t) = H(t) \forall i$. If we assume that $Q_i(t)$ follows an exponential distribution with parameter $\lambda$,

$$Q_i(t) = \lambda e^{-\lambda t},$$

then Eq. 3.4 becomes:

$$P_{fa}(t) = \sum_{k=n_\epsilon}^{n} \binom{n}{k} (H(t))^k (1 - H(t))^{n-k}.$$
In order to compare the false alarm delay with *Gossip Detector* containing different number of defence nodes, we define the pdf of $P_{fa}(t)$ as,

$$P_{pdf}^{fa}(t) = \frac{P_{fa}(t)}{dt},$$

Then,

$$E_{fa}[t] = \int_0^\infty P_{pdf}^{fa}(t)t \, dt.$$  \hspace{1cm} (3.12)

The trends for false alarm detection delay are obviously the same trends for attack detection delay as shown earlier, so we do not show all of the charts again. As shown in Fig. 3.13, the false alarm delay reduces as the number of defence nodes increase in *Gossip Detector*. We expect this false alarm delay behaviour with number of defence nodes in our real simulation experiments.

![Figure 3.13](image_url)  
Figure 3.13: Compares the average detection delay when the number of defence nodes varies in the defence system.
3.4.4 Tradeoff

Generally the system shows a trade-off between attack detection delay and false alarm rate. We plot Fig. 3.14 to see the tradeoff between $E_{ad}(t)$ as $E_{fa}(t)$.

![Figure 3.14: Compares the average detection delay when the number of defence nodes varies in Gossip Detector.](image)

3.5 Summary

The analytical results clearly shows that by increasing number of defence nodes in Gossip Detector not only system reduces the detection time but also it increases the probability of attack detection and reduces probability of false alarm. Furthermore, the performance of Gossip Detector directly depends on individual performance of defence nodes. That is, at least the threshold number ($n_\epsilon$) of defences node have to make the the correct attack decisions at a time to raise a right attack decision by Gossip Detector.

The main contributions of this chapter are:

- We propose the preliminary version of Gossip Detector and demonstrate how Gossip
3.5 Summary

Theoretical Model for Gossip Detector

Detector can achieve the proactive attack detection. We use a number of extensive numerical simulations for this purpose.

- Defined new metrics which are suitable to measure the performances of a defence systems which detects attack at multiple locations of the network, and

- Study of the interdependency of the system performance metrics and other parameters of Gossip Detector.
Chapter 4

Cooperative Distributed Defence System

In this chapter, we propose Gossip Detector in detail and evaluate against DDoS attacks using network simulation. Also we propose the use of a gossip-based averaging protocol in the attack detection process. Legitimate traffic is not differentiated from attack traffic in this work; our experiments generate both legitimate traffic and attack traffic and our defence nodes count all such traffic as the same. We leave aspects such as packet identification, that might be used to improve Gossip Detector to other research. Also, we do not consider responding to flagged attacks, e.g. by throttling or dropping packets. Instead we concentrate solely on early detection of attacks.

Gossip Detector detects DDoS attacks at an early stage using a deployment of defence nodes that reside on routers, as an overlay, and that use the router network for communicating measurement information. We assume that Gossip Detector is intended to defend a given node in the system called the victim. We locate all of the defence nodes at the same network distance (number of hops) from the victim. The defence nodes each locally measure the traffic bit rate that is being routed to the victim and use a gossip-based averaging protocol, as well as router topology properties and time-dependent averaging, to estimate the total (network-wide) traffic bit rate that is being routed to the victim. Each defence node then flags an attack or not based on whether the estimate exceeds a fixed threshold.
Networks–enabled devices constantly create Internet traffic which vary from time to time. This traffic variation will affect threshold setting in the Gossip Detector. In our work we compute what threshold will achieve desired delay/accuracy/bandwidth tradeoffs under a variety of attack traffic intensity and pattern settings, or in other words, for a given threshold what is the resulting tradeoff. While we do not categorize the traffic intensity as “peak” and ”non-peak”, we believe that our approach can be used to ascertain appropriate thresholds in these conditions. The system can in theory then apply these thresholds given the time of day, day of the week, etc.

4.1 System Architecture

In our work the victim is assumed to be running a web server that is receiving requests from other hosts in the network. There are both legitimate hosts that generate HTTP requests for the victim’s web server and attacking hosts that generate constant bit rate traffic to the victim during an attack. Fig. 4.1 shows a simple example. Further details concerning background traffic, attacker distributions, etc., are explained later in Section 4.3.

Let the victim’s maximum bit rate, i.e. that it can receive, be $\omega_{\text{max}}$, sometimes called the victim’s capacity. Let $\omega_i$ be the instantaneous bit rate of traffic that is generated at router $i$ (or more specifically at the host attached to router $i$) and that is destined to the victim. In this work, a flooding based attack is considered to be occurring at any time when:

$$\sum_i \omega_i > \omega_{\text{max}}.$$  

This is a necessary and sufficient condition for the victim’s capacity to be exceeded – we use it as our attack detection condition. This condition does not necessarily imply that the victim’s capacity will be exceeded, because packets generated at a distance from the victim take longer to reach the victim than packets generated closer to the victim, by which time such packets may already be processed. However, the victim’s capacity will
Figure 4.1: Example Gossip Detector deployed on a simple network. The connections from hosts to routers are not counted in distance calculations.
4.1 System Architecture

Cooperative Distributed Defence System

never be exceeded if the condition is never true.

Consider a minimum spanning tree rooted at the victim. Let $\omega_{i}^{\text{local}}$ be the instantaneous bit rate of traffic that is generated at router $i$ plus traffic that passes through router $i$ coming from $i$’s subtree. Let $\Gamma_d$ be the set of routers at distance $d$ from the victim. Then we may write:

$$\sum_{i} \omega_{i} = \sum_{i \in \Gamma_d} \omega_{i}^{\text{local}} + \sum_{i \in \Gamma_0 \cup \Gamma_1 \cup \ldots \cup \Gamma_{d-1}} \omega_{i},$$

for any $d \in \{0,1,2,\ldots,d_{\text{max}}\}$ where $d_{\text{max}}$ is the maximum distance from the victim to any other router. This principle is applied in Gossip Detector. The first component is collectively measured by defence nodes that are located at distance $d$ from the victim. The second component is estimated using knowledge of the router topology and attack statistics. More specifically, we assume that the router topology is known and that attackers are distributed uniformly at random across the router network. This allows us to estimate the second component from the first component:

$$\sum_{i \in \Gamma_0 \cup \Gamma_1 \cup \ldots \cup \Gamma_{d-1}} \omega_{i} \approx \frac{N_{d-1}}{N - N_{d-1}} \sum_{i \in \Gamma_d} \omega_{i}^{\text{local}},$$

where $N_{d} = |\bigcup_{j=0}^{d} \Gamma_j|$.

Our practical implementation is not as simple as the above suggests. The $\omega_{i}^{\text{local}}$ is measured over a time interval and the summation of $\omega_{i}^{\text{local}}$ over $i$ uses a distributed gossip approach which introduces additional delays. Therefore we make a number of compensating adjustments in our practical calculations.

4.1.1 Parameters of Gossip Detector

Table 4.1 shows a summary of notation which describe Gossip Detector parameters. The words node and router are used interchangeably. When we talk about victim traffic or victim bit rate we mean traffic that has the victim as its destination address and the bit rate of such traffic resp. A distance between two nodes in the network refers to the smallest
number of hops or edges between the nodes.

<table>
<thead>
<tr>
<th>symbol</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N$</td>
<td>number of nodes in the network</td>
</tr>
<tr>
<td>$n_i$</td>
<td>number of nodes at distance $i$ from the victim</td>
</tr>
<tr>
<td>$d$</td>
<td>the distance from all overlay nodes to the victim</td>
</tr>
<tr>
<td>$N_d$</td>
<td>number of nodes less than or equal to distance $d$ from the victim</td>
</tr>
<tr>
<td>$d_{\text{max}}$</td>
<td>the distance to the node(s) that are furtherest from the victim in the network</td>
</tr>
<tr>
<td>$n_{\text{nei}}$</td>
<td>number of other defence nodes known to each defence node</td>
</tr>
<tr>
<td>$n_e$</td>
<td>minimum number of alerts to make an attack decision</td>
</tr>
<tr>
<td>$\omega_{\text{max}}$</td>
<td>victim’s maximum bit rate</td>
</tr>
<tr>
<td>$\omega_{\text{local}}_{i,p}$</td>
<td>locally measured victim bit rate in phase $p$ at node $i$</td>
</tr>
<tr>
<td>$\omega_{i,p}^{\text{gossip}}$</td>
<td>averaged victim bit rate in phase $p$ at node $i$</td>
</tr>
<tr>
<td>$\omega_{i,p}^{\text{adj}}$</td>
<td>adjusted average victim bit rate in phase $p$ at node $i$</td>
</tr>
<tr>
<td>$\omega_{i,p}^{\text{total}}$</td>
<td>estimated total victim bit rate towards the victim from the entire network in phase $p$ at node $i$</td>
</tr>
<tr>
<td>$\omega_{\text{used}}$</td>
<td>percentage of bandwidth consumed by Gossip Detector</td>
</tr>
<tr>
<td>$s_{i,t}$</td>
<td>size of packet, at node $i$, that arrived at time $t$</td>
</tr>
<tr>
<td>$W_{\text{size}}$</td>
<td>number of previous phases used to make smooth attack decision</td>
</tr>
<tr>
<td>$\sigma_{i,p}$</td>
<td>number of test positives in the current window at node $i$ and phase $p$</td>
</tr>
<tr>
<td>$\tau_{\text{atk}}$</td>
<td>attacking time intervals in the experiments</td>
</tr>
<tr>
<td>$\tau_{p}$</td>
<td>phase interval or attack detection interval</td>
</tr>
<tr>
<td>$\tau_{r}$</td>
<td>round interval within phases</td>
</tr>
<tr>
<td>$\epsilon$</td>
<td>minimum expected performance of Gossip Detector</td>
</tr>
<tr>
<td>$A_{\text{rate}}$</td>
<td>constant bitrate of experimental attacks</td>
</tr>
</tbody>
</table>

4.1.2 Overlay Construction

We place a defence node at every router that is a distance $d$ from the victim. These defence nodes form a cooperative overlay network. Let $n_d$ be the number of such nodes. For
example, in Fig. 4.1, defence nodes are located at distance 2 from the victim, and \( n_d = 5 \). Any victim traffic, that originates at a router at distance \( \geq d \) from the victim will pass through exactly one overlay node and the overlay node will count these packets. All other packets, i.e. packets with destination address that are not the victim, are not counted. We assume that the overlay nodes communicate via the router network and that every node in the overlay knows of every other overlay node. A message sent from one overlay node to another may traverse several hops in the router network.

### 4.2 Detailed Gossip Detector Design

Our overall approach is outlined below:

1. Each node makes an independent, local measurement of the victim’s bitrate.

2. All nodes participate in a distributed averaging algorithm whereby they arrive at the average of their local measurements – ideally they would all arrive at the same value.

3. Since the distributed averaging algorithm takes some time to complete, each node locally adjusts the resulting average by combining it with its latest local measurement.

4. The adjusted average is then multiplied by the number of overlay nodes and the result is taken to be the total victim traffic that originates from distance \( \geq d \) to the victim. This is further corrected to account for victim traffic that cannot be measured, i.e. traffic that originates from distance \( < d \) to the victim, to obtain the total victim bit rate.

5. Each node then locally tests whether the victim bit rate exceeds the victim’s capacity. If at least 50% of a node’s local tests are positive within a given time window then the node flags that an attack is happening at that time.
We now expand the above bullet points.

### 4.2.1 Phases, Rounds and Local Measurement

Our overall algorithm consists of phases of duration $\tau_p$ seconds, where each phase consists of rounds of duration $\tau_r$ such that $\frac{\tau_r}{\tau_p}$ is the number of rounds in a phase, which we ensure is always integral. Fig. 4.2 shows a time line with phases and rounds; phases end at times $t_0, t_1, t_2, \ldots$, where phase $p$ ends at time $t_p$. We assume that all overlay nodes operate synchronously. Having said this, in the experiments we include the case when the round time is insufficient for packets to be communicated from all nodes in the overlay to all other nodes. During phase $p - 1$, which starts at time $t_{p-2}$ and terminates at time $t_{p-1}$, each defence node $i = 1, 2, \ldots, n_d$ counts all victim packets that pass through it and this is used to compute the locally observed victim bit rate at node $i$ during phase $p - 1$:

$$\omega_{i,p-1}^{local} = \sum_{t \in (t_{p-2}, t_{p-1})} \frac{s_{i,t} \tau_p}{\tau_p}, \quad (4.2)$$

where $s_{i,t}$ is the size of the packet counted at node $i$ that arrived at time $t$. The above measurement is simply the locally observed victim bit rate at overlay node $i$.

### 4.2.2 Gossip-based Averaging

We use a gossip-based averaging algorithm that enables all defence nodes to compute the average of local bit rate measurements:

$$\omega_{i,p}^{gossip} = \frac{1}{n_d} \sum_{i=1}^{n_d} \omega_{i,p-1}^{local}. \quad (4.3)$$

We consider the most general form of the push-sum algorithm for computing the average, described in [38]. In each round $r$, within a phase $p$, the defence node $i$ computes a sum $S_{r,i,p}$ and a weight $W_{r,i,p}$ which are initialised to $S_{0,i,p} = \omega_{i,p-1}^{local}$ and $W_{0,i,p} = 1$. Each
4.2 Detailed Gossip Detector Design

Cooperative Distributed Defence System

Figure 4.2: Example timeline for defence node $i$. Phase $j$ ends at time $t_j$. At the end of a phase, a local measurement is produced, $\omega_{\text{local}}^{i,j}$. A constant number of gossiping rounds occur within each phase, as indicated by the small tick marks, which is used to gossip the local measurement from the previous phase and produce $\omega_{\text{gossip}}^{i,j}$. An attack test is made at the end of each phase. If more than 50% of such tests are positive within a sliding window of $W_{\text{size}}$ phases then an attack alert is raised as indicated by the impulse, $a_i(t) \in \{0, 1\}$.

The node $i$ chooses non-negative contribution fractions $\alpha_{r,i,j}$ of $S_{r,i,p}$ and $W_{r,i,p}$ and sends to all neighbours $j$, where $\sum_j \alpha_{r,i,j} = 1$. At the end of round $r$, let $(S_{j}', W_{j}')$ be all value-weight pairs received by node $i$ in round $r - 1$ then:

$$S_{r,i,p} = \sum_j S_{j}'$$

and

$$W_{r,i,p} = \sum_j W_{j}'$$

The estimate of Eq. 4.3 at node $i$ is then:

$$\frac{S_{r,i,p}}{W_{r,i,p}}$$

i.e. the estimate average in phase $p$ of the values measured in phase $p - 1$.

In this work, each defence node knows every other defence node and so $\alpha_{r,i,j} = \frac{1}{n_d}$ is a constant over all defence nodes and all rounds. However we do not assume that there
is enough time for all packets to be communicated between all defence nodes in each round of the gossiping, i.e. the round time may be less than the required communication time. In this case, packets which arrive after the round are discarded. This leads to errors in the averaging process; however it also allows the process to take less time and is therefore a tradeoff that we explore experimentally. Each defence node undertakes \( \frac{\tau_p}{\tau_r} \) rounds. Increasing the number of rounds, either by increasing the phase time or by decreasing the round time, leads to a more accurate estimation for average bit rate but it wastes various network resources and increases the detection latency.

### 4.2.3 Local Adjustment

The average computed at the end of phase \( p \) is always one phase time old. By this time each node has a more up-to-date local measurement of the victim bit rate \( \omega_{i,p}^{\text{local}} \). To include the most up-to-date measurement \( \omega_{i,p}^{\text{local}} \), we use the following:

\[
\omega_{i,p}^{\text{adj}} = (1 - \beta) \omega_{i,p}^{\text{gossip}} + \beta \omega_{i,p}^{\text{local}} \tag{4.4}
\]

where \( 0 \leq \beta \leq 1 \) is a constant.

### 4.2.4 Total Victim Bitrate

The overlay does not measure packets that come from inside the overlay, i.e. traffic that comes from nodes at a distance less than the overlay distance from the victim. We apply the principle from Eq. 4.1 at each node \( i \) to obtain an estimate of the total victim bit rate:

\[
\omega_{i,p}^{\text{total}} = n_d \omega_{i,p}^{\text{adj}} \left( 1 + \frac{N_{d-1}}{N - N_{d-1}} \right) \tag{4.5}
\]

Note that \( n_d \omega_{i,p}^{\text{adj}} \) is effectively node \( i \)'s estimate of \( \sum_{i \in \Gamma_d} \omega_{i}^{\text{local}} \).
### 4.2 Detailed Gossip Detector Design

Cooperative Distributed Defence System

---

**Figure 4.3:** Attack detection procedure. Measurements from phase $p - 1$ are gossiped in phase $p$, measurements from phase $p$ are gossiped in phase $p + 1$, and so on. Marking the packets as counted avoids it being counted twice, though it is not strictly required in this work because a packet will never pass through more than one defence node.

1. Calculate $\omega_{\text{local}}^{i,p-1}$ using Eq. 4.2.
2. Mark the pkt as counted.
3. Send the pkt to outgoing link.

---

1. Using gossip based averaging algorithm [38], exchange $\omega_{\text{local}}^{i,p-1}$ value among defence nodes for $\tau_p$ rounds to obtain $\omega_{\text{gossip}}^{i,p}$.
2. Calculate $\omega_{\text{total}}^{i,p}$ using Eq. 4.5.

---

Add one more test positive to the current window

- Yes
  - \( \omega_{\text{total}}^{i,p} > \omega_{\text{max}} \)
  - Attack alert
- No
  - \( \text{test positives in the current window} \geq 50\% \) of window size
  - No attack alert
4.2.5 Windows and Detection

We provide the attack detection procedure as a flow chart in the Fig. 4.3. Each defence node executes the algorithm to make its own attack tests on its estimated total bit rate. If

\[ \omega_{i,p}^{total} > \omega^{\text{max}} \]

then a test positive is recorded in phased \( p \). In order to make smooth attack decisions and hence to reduce false alarm rates, an attack alert is raised based on the tests of the current phase and the tests of the previous phases. As illustrated in Fig. 4.2, we can use the previous three phases’ tests and current phase’s test (we call this a window of size 4) to make an attack alert decision on the current phase. If at least 50% of the attack tests in the window are positives then node flags an attack alert.

4.2.6 Overall Attack Decision

Gossip Detector makes local attack alert at each decentralized defense node. In order to make overall attack decision, the attack alerts from each defense node should be combined in a network place. It can be achieved by two different methods. First method is to send attack alerts of defense nodes to a centralized location (to the victim) and combine results to make overall attack decision. Second way is to propagate the local attack alerts among defense nodes via gossip rounds of next phase time and combine overall attack decision in every defense node and then can response attack from each defense node. If we use the last method, Gossip Detector remains as a decentralized defence system. Further, in this thesis we were considered only with detecting the attack, the actual response is outside our scope.
4.3 Performance Evaluation

4.3.1 Simulation Implementation

Simulating Internet Like Topology

There are several topology generators available to the network research. Some of them mainly aim to generate random topologies and others aim to the hierarchical properties of the Internet. An ideal topology generator should enable the use and development of generation models that produce accurate representations of Internet topologies. We used the Brite [53] topology generator that reproduce fundamental properties of the topology of the Internet. Our experimental topology, shown in Fig. 4.5, consists of 100 nodes and the bandwidth of all links set to 10Mbs [105]. One of the nodes is selected as the victim and remains constant for all experiments. In each experiment we randomly select 40 nodes to be attacking nodes and the remaining 60 nodes become legitimate nodes. Of the 60 legitimate nodes, 40 of them are clients of the victim and the rest are clients to a second server that is placed somewhere else in the network at random. Fig. 4.4 shows the number of defence nodes at different distances.

Simulating Internet Traffic

In the early days of Internet research, people assumed that Internet traffic generated by individual network hosts is Poisson distributed and so the aggregated traffic follows the same. However, a number of studies show that the data traffic in both local-area and wide-area IP networks clearly differs from Poisson process [69]. In the Internet, heavy-tailed distributions have been observed in the traffic flow and in the topological properties. The Pareto distribution has a heavy tail that is not shown in Poisson process and the real IP traffic is better modeled by Pareto model with density function.
Cooperative Distributed Defence System

4.3 Performance Evaluation

A Pareto model with shape parameter $\alpha=1.4$ is used to represent wide-area Internet traffic [42]. In our experimental implementation, HTTP clients follow the Pareto model with shape parameter 1.4 in HTTP data requests.

Simulating Attack Traffic

The attacking nodes do not follow any congestion control scheme and pump as much traffic as possible to exhaust the HTTP server [42]. We use a constant bit rate (CBR) traffic generator in ns-2 to generate UDP packets towards the victim to start the attack [105]. In most of our experiments each attacker uses a bit rate of 0.25Mbps. In some of the experiments the attacks use a bit rate selected uniformly at random from the interval $(0.01, 0.25)$ Mbps. In our experiments, we simulate 10 flooding based attacks lasting 10
Performance Evaluation

Cooperative Distributed Defence System

Defence nodes

Traffic measured in the overlay

Background users

Victim’s users

Attacking nodes

HTTP servers

Traffic estimated using Eq. 1.

Server1: Victim

Server2: Background traffic

Defence nodes

Figure 4.5: Experimental topology. All nodes at the same distance from the victim (left) are drawn in the same column. The defence nodes in this example are at distance 3. A second server for background traffic is shown at distance 4.
seconds each. The first attack is launched in the first 100 seconds after the experiment is started. Other attacks occur in time intervals of 40 seconds. Each particular experimental run is averaged over ten trials where the attacker locations, second server placement and random CBR if applicable, are different in each trial.

4.3.2 Performance Metrics

Individual defence nodes make their own attack decision based on the estimated total bit rate at the end of each phase. defence nodes can therefore make contradictory attack decisions to each other. In order to evaluate system wide performance of Gossip Detector we define the following measurements. Let \( a_i(t) \) represent the attack decisions of node \( i \) at time \( t \):

\[
a_i(t) = \begin{cases} 
1 & \text{if } \sigma_{i,p} \geq \frac{W_{size}}{2} \\
0 & \text{otherwise,}
\end{cases}
\]

where \( \sigma_{i,p} \) is the number of test positives in the window at node \( i \) and phase \( p \). Also we define the oracle:

\[
A(t) = \begin{cases} 
1 & \text{if } t \in [\tau_{atk}] \\
0 & \text{otherwise}
\end{cases}
\]

We then define the over all system wide performance, \( \tilde{A}(t) \), at time \( t \) as follows:

\[
\tilde{A}(t) = \frac{1}{n_d} \sum_{i=1}^{n_d} \left( 1 - a_i(t) \oplus A(t) \right),
\]

where \( \oplus \) is the exclusive or operator, i.e. the summand evaluates to 0 for defence node \( i \) iff its attack decision is incorrect at time \( t \).

If \( \tilde{A}(t) \geq \epsilon \) then we consider the performance of Gossip Detector is enough to protect the victim, where \( \epsilon \) is a performance threshold value, \( 0 \leq \epsilon \leq 1 \). In our all experiments we assume \( \epsilon \) is 0.8.

We introduce three evaluation metrics: Probability of Attack Detection, Probability of False Alarm, and Detection delay to measure performance.
### Probability of Attack Detection and Probability of False Alarm

Define $TP$ (True Positive), $TN$ (True Negative), $FP$ (False Positive), and $FN$ (False Negative) as follows:

$$TP = \sum_{t \in [\tau_{atk}]} \tilde{A}(t)$$
$$TN = \sum_{t \in [\tau_{atk}]} \tilde{A}(t)$$
$$FP = \sum_{t \in [\tau_{atk}]} \tilde{A}(t)$$
$$FN = \sum_{t \in [\tau_{atk}]} \tilde{A}(t).$$

Probability of Attack Detection and Probability of False Alarm are then defined:

$$\text{Probability of Attack Detection} = \frac{TP}{TP + FN}$$
$$\text{Probability of False Alarm} = \frac{FP}{FP + TN}.$$

### Detection Delay

We define the detection delay as the time between when the attack starts to when the systemwide performance, $\tilde{A}(t)$, recovers to be $\geq \epsilon$. At the start of the attack, the systemwide performance drops to near 0 by the end of the current phase because a number of nodes do not observe the attack in that phase. The Gossip Detector performance improves as gossiped information is taken into account.

### 4.4 Fundamental Experimental Results

For brevity we omit experimental results that cover all ranges of parameters. In particular, we do not include experimental results pertaining to window size – we experimentally found that $W_{size} = 2$ provides the best performance for Gossip Detector in the cases.
4.4.1 Traffic Estimation

We use the principle of Eq. 4.1 to estimate missing inside traffic of the victim. The overlay at distance \( d = 1 \) does not do any estimation of missing inside traffic since the victim does not generate traffic to itself. As the distance increases, Fig. 4.6(a) shows that the missing inside traffic becomes significant, i.e. less traffic is measured further from the victim. Fig. 4.6(b) shows the use of Eq. 4.1 to include the missing traffic in the total bit rate estimation; whereby roughly the same bit rate is obtained independent to the distance.

4.4.2 Local Versus Global Information

Fig. 4.7(b) experimental result shows that if each defence node uses only the local bit rates (\( \beta = 1.0 \)) for making attack decisions, the performance of *Gossip Detector* is not enough to detect attacks. In other words, without gossiping, about half the nodes are not aware of the attack. When we use both local and global attack information *Gossip Detector* successfully detects attacks as shown in Fig. 4.7(a). We experimentally determined \( \beta = 0.25 \) to be the best setting for the cases we examined in this work and we use that value for all other experiments.

4.4.3 Round Time and Distance

We call the packet exchange time of the routers as the round time. The gossip-based averaging algorithm plays a major role in establishing a correct attack decision at all defence nodes. The performance of *Gossip Detector* depends on the accuracy of the averaging algorithm. Accuracy of the gossip-based averaging algorithm in turn depends on the gossiping round time and the number of rounds within the phase. If the round time was large enough, and no packets were dropped by routers, all gossiped packets would be
4.4 Fundamental Experimental Results

Cooperative Distributed Defence System

Figure 4.6: Estimated total bit rate to victim during an attack that starts at time 100 and ends at time 110, for overlays at different distances, $\beta = 0.25, \tau_r = 5ms, \tau_p = 20ms, W_{size} = 2$. 
Figure 4.7: Performance, $\tilde{A}(t)$, for $\beta = \{0.25, 1.0\}$, $\tau_r = 5ms, \tau_p = 20ms, W_{size} = 2$. In (a) Gossip Detector performance quickly recovers after the start of the attack because of gossiped information, while in (b) under the same scenario, without gossiping, about half the nodes are not aware of the attack.
properly communicated and the defence nodes would converge to an exact average in a single round. However we are mainly concerned on early detection of attack and so we are concerned with round times that are as short as possible – the time taken roughly for a packet to be transmitted from one router to another. As round time is increased then more of the gossiped packets will reach their destination before the end of the round and thereby be included in the computation. As the number of rounds is increased, the gossiping algorithm will continue to converge, though it is not entirely defined what it converges to in the presence of lost information. Nevertheless our experimental results show the approach performs for a range of different round times and numbers of rounds, and for overlays at different distances.

Experimental results shown in Fig. 4.8 and Fig. 4.9, depicts detection delay, probability of attack detection and probability of false alarm in columns 1, 2 and 3 resp. Rows 1 to 5 show increasing number of rounds. Each chart has the round time as the independent variable and shows results for overlays at different distances. The total phase time is the number of rounds times the round time. We can make comparisons in Fig. 4.8 and and Fig. 4.9 between cases where the phase time is constant, e.g. comparing round time 6ms and 1 round, with round time 3ms and 2 rounds, and so on. The experiments in Fig. 4.8 and Fig. 4.9 were run with attacking nodes that each used 0.25Mbps CBR traffic to the victim. Fig. 4.12 shows a selection of these results for the case when the attacking nodes each select their CBR rate uniformly at random in the interval \((0.01, 0.25)\) Mbps.

Generally, when round time and/or number of rounds becomes bigger the detection rates increase and Gossip Detector takes a longer time to detect attacks. When the detection rate is low then the detection delay can sometimes be very large, much larger than what is useful. In some cases, e.g. when the overlay nodes are at distance 5 and no attackers appear at that distance (there are only 4 routers at that distance) then attacks are not detected at all. In this case we did not include the detection delay, in other words, the results show the detection delay assuming that the attack was detected.

Erratic behavior is observed at distance 4 and to a lesser extent at distance 5. There
Figure 4.8: Comparing performances in all distances for different round times and the number of rounds (1, 2 and 3), $\beta = 0.25$, and $W_{\text{size}} = 2$. The phase time, $\tau_p$, is the round time, $\tau_r$, by the number of rounds. CBR of all attackers is 0.25Mbps.
Figure 4.9: Comparing performances in all distances for different round times and the number of rounds (4 and 5), $\beta = 0.25$, and $W_{size} = 2$. The phase time, $\tau_p$, is the round time, $\tau_r$, by the number of rounds. CBR of all attackers is 0.25Mbps.
are various aspects of *Gossip Detector* that contribute to this behavior. The behavior at these distances is further elucidated in Fig. 4.12. Generally, at these distances there is less observed attack traffic and more reliance on estimation of the missing traffic which can lead to greater variation. Background traffic (which is not separated from attack traffic) therefore has a greater impact on the estimation. Fig. 4.12 shows that when the attackers are all using different bit rates, the estimation of missing traffic fails to work at large distances. Surprisingly distance 5 performance better than distance 4, though both have poor detection rates. Distance 5 has only 4 nodes in the overlay and so gossiping can be relatively more exact than distance 4 which has significantly more nodes in the overlay.

The probability of false alarms are generally unaffected by distance. Increasing the round time and increasing the number of rounds generally increases the probability of false alarm. This is because the phase time increases and *Gossip Detector* takes proportionally longer to recover when the attack finishes.

Fig. 4.10 and Fig. 4.11 show the performance of *Gossip Detector* for different rounds (1, 2, 3, 4 and 5) for CBR attack with $\tau_r = 8\text{ms}, \beta = 0.25, W_{\text{size}} = 2$ and indicate the competitive configuration in each case. Fig. 4.13 shows the performance of *Gossip Detector* for different rounds (2 and 4) for VBR attack with $\tau_r = 8\text{ms}, \beta = 0.25, W_{\text{size}} = 2$ and indicate the competitive configuration in each case.

### 4.4.4 Bandwidth Consumption

Bandwidth consumption of *Gossip Detector* depends on gossip round time and number of defence nodes. Fig. 4.14(a) shows the percentage of bandwidth consumed by gossip packets for different distances. The bandwidth consumed by *Gossip Detector* is negligible compared to the total bandwidth of the network.
Figure 4.10: Comparing performance of Gossip Detector for different rounds (1, 2 and 3) for CBR attack and $\tau_r = 8ms$, $\beta = 0.25$, $W_{size} = 2$. CBR of each attacker is selected uniformly at random in the range (0.01, 0.25) Mbps.
Figure 4.11: Comparing performance of Gossip Detector for different rounds (4 and 5) for CBR attack and $\tau_r = 8 \text{ms}, \beta = 0.25, W_{\text{size}} = 2$. CBR of each attacker is selected uniformly at random in the range (0.01, 0.25) Mbps.
Figure 4.12: Comparing performances in all distances for different round times and the number of rounds $\beta = 0.25, W_{size} = 2$. CBR of each attacker is selected uniformly at random in the range $(0.01, 0.25)$ Mbps.
Figure 4.13: Comparing performance of *Gossip Detector* for different rounds (2 and 4) for VBR attack and $\tau_r = 8\, ms$, $\beta = 0.25$, $W_{size} = 2$. VBR of each attacker is selected uniformly at random in the range $(0.01, 0.25)\, Mbps$. 

(a) Number of rounds=2

(b) Number of rounds=2

(c) Number of rounds=4

(d) Number of rounds=4
4.4 Fundamental Experimental Results

Cooperative Distributed Defence System

4.4.5 Lost and Late Packets

The Fig. 4.14(b) shows the drop rate of the control packets during an attack. The reasons for the packet drops are either from router drops or because they are discarded because they arrived at the destination after the round time. The affect of lost and late packets is partly the reason for poor detection rates observed in Fig. 4.8 and Fig. 4.9 when round time is short. It is also partly responsible for the case where distance 5 shows better detection rate than distance 4–distance 5 has significantly less lost and late packets than distance 4.

The limitation of Gossip Detector is that if all attacking nodes are inside the overlay ring then Gossip Detector does not detect attacks. But this scenario is very rare in real network and we can ensure to be more trusted hosts inside the defence ring. When we increase the distance between the victim and the defence overlays, the number of defence nodes will raise and hence communication between defence nodes takes longer time. It requires bigger gossiping round time and increased detection delay.
4.5 Summary

From the Fig. 4.10 and Fig. 4.11, the distance 4 appears to be the best choice for *Gossip Detector* deployment. Furthermore, it is difficult to compare with the theory because the theory does not include deployment distance. In the simulation nodes at a greater distance experience less background traffic and so they have a lower false alarm probability compared to nodes closer to the victim. Hence, the abstract model in Chapter 3, which has the same false alarm probability for all nodes, is not accurate in this respect.
Chapter 5

Comprehensive Analysis of DDoS Attacks

In this chapter, we present existing flooding based DDoS attack models and evaluate Gossip Detector against these attacking models. The performance of Gossip Detector depends on two types of factors: the parameters of Gossip Detector and the external factors such as deployment network topology, distribution of the attack sources, strength of attack, etc. We experimentally explore all of the factors to determine the optimal parameter settings that ensure the best performance of Gossip Detector. Also we analyse the bandwidth usages of Gossip Detector when we choose different configurations for Gossip Detector. Finally, we compare the performance of Gossip Detector with existing well known defence systems.

The fundamental objective of Gossip Detector is to minimise detection delay, while maintaining a higher attack detection probability and keeping false alarms below a required level. The theoretical analyses of Gossip Detector performed in Chapter 3, concluded that Gossip Detector could achieve the objective by simply increasing the number of defence nodes contains. In Chapter 4, we experiment with Gossip Detector by launching simple attacking models that have a constant network topology. Furthermore, in this chapter we run an extensive number of experiments on different network topologies for thorough evaluation of Gossip Detector.
5.1 Experimental Setup

In order to evaluate Gossip Detector with different parameter settings, we used the ns-2 [24] simulator. The ns-2 is an object-oriented, discrete event driven network simulator. It is primarily useful for simulating local and wide area networks. For realistic experimental results, we use widely accepted methods for generating the Internet topology and background Internet traffic. We deploy the same experimental topology as in Section 4.3.1. Fig. 5.1 shows a sample experimental topology.

5.1.1 Legitimate Traffic

In the experiments of previous works, we deployed a HTTP based traffic model for generating Internet like legitimate traffic. In this work, we used a widely accepted self-similar traffic model for simulating Internet traffic. Internet packet traffic exhibits statistically self-similar traffic characteristics [48]. To generate self-similar traffic, we used the method described in [88]. In this method, the resulting self-similar traffic is obtained by aggregating multiple sub-streams, each consisting of alternating Pareto-distributed on/off periods as illustrated in Fig. 5.2.

Pareto distribution is a heavy-tailed distribution with the probability-density function 
\[ f(x) = \frac{\alpha k^\alpha}{x^{\alpha+1}} \] for \( x \geq k \), where \( \alpha \) is a shape parameter, and \( k \) is a location parameter. Table 5.1 shows configurations of the Pareto On/Off traffic generator used in our ns-2 experiments.

<table>
<thead>
<tr>
<th>Pareto parameter</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>burst_time</td>
<td>1000ms</td>
</tr>
<tr>
<td>idle_time</td>
<td>400ms</td>
</tr>
<tr>
<td>rate</td>
<td>0.2mb</td>
</tr>
<tr>
<td>packetSize</td>
<td>120</td>
</tr>
<tr>
<td>shape</td>
<td>1.4</td>
</tr>
</tbody>
</table>

We used the shape parameters of the Pareto model at 1.4 because it represents a typ-
Figure 5.1: Sample experimental topology, $N=200$. 
The common feature of all flooding-based DDoS attacks is to create a huge volume of attack traffic to exhaust the link between the victim and the rest of the network. The UDP flood, TCP SYN flood, and Smurf attacks are some of the well-known flooding-based attacks. The UDP flood attack is more effective than other attacks as the size of the UDP packets can be enormous. The packet size could be set up to 65000 bytes which could easily flood a given target link when multiple zombies are set up [86]. Furthermore, UDP flood attack is supported by most attacking tools like Trinoo [109] and it is easy to implement in the ns-2 simulator. We generated UDP flood attacks with different packet rates to test four attack dynamics: constant rate, increasing rate, pulsing, and subgroup attacks [108].

We uniformly distributed 40% attacking sources out of total traffic sources throughout
the simulated network. Even though flooding-based attack sources generate as many packet as attacking resources allow, we avoided the full force of attack, such that the attack could not be detected easily at the defence nodes [59]. The attach rate for each attack source is set to be 0.15Mbps, and thus the total attack rate is 6Mbps (40x0.15). The average background is set to be around 8Mbps and the victim capacity is 10Mbps. After 100 seconds of the experiment’s start and before 300 seconds the of experiment’s end, all attack sources simultaneously launch an attack at a given time, and this attack continues for 300 seconds.

### 5.2.1 Constant Rate Attack

![Image](https://via.placeholder.com/150)

**Figure 5.3:** Attack rate variation for a constant rate attack at an attack source.

Constant rate attack is the simplest and most known DDoS attacking technique [60]. In this type of attack the intended attack rate is achieved immediately and maintained until the attack is stopped. In order to run an experiment under this type of attack, uniformly distributed attack sources generate constant rate packets simultaneously towards the victim. Fig. 5.3 shows the attack dynamics with a constant rate, starting from time $t_0$ with an attack intensity 0.15Mbps.

### 5.2.2 Increasing Rate Attack

Usually an abrupt change in traffic volume is an important signal to initiate anomaly detection. Avoiding abrupt changes in traffic volume is a strategy of sophisticated attackers
5.2 Flooding-based DDoS Attacks

Comprehensive Analysis of DDoS Attacks

5.2.3 Subgroup Attack

In this type of attack, the attacking sources are divided into several subgroups that coordinate so that one subgroup is always active, followed by successive attacks from the subgroups, which can still appear like a continuous denial of service at the attack victim. This method of attack hides the attack sources and makes it difficult for defence systems to trace back through the attack path. This method of attack dynamic changes the direction of the attack and makes it difficult to traceback and respond to the attacks.
To simulate a subgroup attack, we divided 40 attack sources into three subgroups. As shown in Fig. 5.5, each of the three subgroups (Gr$_1$: 13 sources, Gr$_2$: 13 sources, and Gr$_3$: 14 sources) is active in turn: the first subgroup attacking from $t_0$ to $t_1$, the second subgroup attacking from $t_1$ to $t_2$, and the third subgroup starting from $t_2$. In this type of attack, because one third of attack sources are active at a time, we set the attack rate to 0.45Mbps for each attack source so that the total attack strength would be approximately 6.0Mbps.

5.2.4 Pulsing Attack

In the pulsing attack, the attack rate is oscillating between a constant rate and zero, thus reducing chance of detection by a defence system. All attack sources start and stop periodically. This fluctuation occurs during a pulsing attack, when attack sources intermittently abort the attack and resume it later. In our experiments, the attack sources sent packets for 10 seconds and then stopped attack for 30 seconds before resuming the attack again.

![Figure 5.6: Pulsing attacking model.](image)

In pulsing attacks, the attack rate oscillates between 0.15 Mbps and zero, periodically reducing attack traffic in order to avoid detection. As shown in Fig.5.6, the dynamics of a pulsing attack appear as an on/off pattern with period $t_p$ and burst duration $l_p$. 

105
5.3 Performance Evaluation

We present the detailed evaluation results of Gossip Detector under different parameter configurations and operating environments. First we present how Gossip Detector respond to different types of flooding-based attacks and then the performance impact of overlay topology and the number of gossip rounded. Finally, we demonstrate how the deployment distance of Gossip Detector influences the performance of Gossip Detector.

5.3.1 Performance Factors

Since the main objective of Gossip Detector is early attack detection, detection delay is the significant performance metric. When we configure Gossip Detector to achieve early attack detection, attack detection probability decreases and the number of false alarms increases. In order to analyse the tradeoffs between the performance of Gossip Detector we use the following metrics: detection delay ($\tau$), probability of false alarm ($P_{fa}$) and probability of attack detection ($P_{ad}$). We refine the methods of computing the performance metrics from previous experiments so as to better reflect the initial definition of the metrics derived in the theoretical analysis chapter. The following subsections explain how we measure performance metrics in our experiments.

As illustrated in Fig. 5.7, there are two types of factors affecting the performance of Gossip Detector. The parameters that we can adjust in Gossip Detector and the external factors that we cannot control in Gossip Detector. We are interested in exploring how the performance varies with all of these factors, and therefore improve upon Gossip Detector so as to perform at a higher level. In order to avoid the influence of external factors while exploring Gossip Detector parameters, we analysed the averaged performance of a number of experimental trials, including all possible circumstances of the external factors. For example to avoid the influence of distribution in the attacking sources, we ran a number of experiments with different combinations of attacking nodes, calculating the average of the trial outcomes to analyse the performance of Gossip Detector.
We omit experimental results that have already been analysed in the Chapter 4. In particular, we do not include experimental results pertaining to phase time ($\tau_p$), i.e. we chose a fixed $\tau_p$ value for a topology and altered the $\tau_r$ value to test a different number of gossip rounds within a phase time.

![Factors affecting performance of Gossip Detector.](image)

**System’s Parameters**
- $n_d$
- $n_{nei}$
- $\tau_p$
- $\tau_r$
- $\beta$
- $W_{size}$
- $n_\epsilon$

**External Factors**
- Distri. of attacking nodes
- Distri. of legitimate nodes
- Deployment topology
- Attack type and strength
- Attack start time

**System’s Performance**
- $\tau$
- $P_{ad}$
- $P_{fa}$
- $\omega_{used}$

**Probability of False Alarm**

According to the attack detection rules of *Gossip Detector* false alarms can occur when at least $n_\epsilon$ numbers of defence nodes raise false alarms at one time, and there is no attack towards the victim. The false alarm at a defence node occurs when the total bitrate estimation of the background traffic exceeds the victim’s capacity. If we know the false alarm probabilities of the individual defence nodes then we can compute the false probability of *Gossip Detector*.

If we assume that $P_{fa}^i$ to be the false alarm probability of $i^{th}$ defence node, $i = 1, ..., n$
and given that there is no attack in the system then the false alarm probability of Gossip Detector is equal to the probability of \( n_e \) number of defence nodes raise false alarms out of \( n \) defence nodes. A Poisson binomial distribution with \( n \) number of independent Bernoulli trails and the trail probabilities \( P_f^{i}, i = 1, \ldots, n \) can be used to compute \( P_{fa} \). We utilized an algorithm, presented in [26], to compute the cumulative density function of the Poisson binomial distribution and hence find the probability of at least \( n_e \) false alarms from the defence nodes at a given time.

![Figure 5.8: Histogram of background and attack traffic of a defence node at a distance.](image)

To compute individual false alarm probability of each defence node, we utilize probability distribution of the background traffic estimations computed at phase time intervals of our experiments. As illustrate in the Fig. 5.8(a), we noticed that the background traffic estimations follow a Gaussian distribution at each defence node. The Central Limit Theorem also serve as a general motivation for such models [40]. i.e. the total background bitrate estimation,

\[
\omega_{i,p}^{total} \sim \mathcal{N}(\mu_i, \sigma_i) \quad i = 1, \ldots, n_d
\]

where \( \mu_i = E(\omega_{i,p}^{total}) \) and \( \sigma_i = Var(\omega_{i,p}^{total}) \).
The Probability Density Function (PDF) is given as,

$$f_{\mu_i, \sigma_i}(x) = \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{(x-\mu_i)^2}{2\sigma_i^2}}$$

If the total background bitrate estimation exceeds the victim’s capacity then false alarm occurs at a defence node. Hence we can compute the probability of false alarm at node $i$ using,

$$P_{fa}^i = \int_{\omega_{\text{max}}}^{\infty} f_{\mu_i, \sigma_i}(x) \, dx \quad i = 1, ..., n_d$$  \hspace{1cm} (5.1)

**Probability of Attack Detection**

The same method used in computing the false alarm probability is followed to calculate the probability of attack detection. The attack detection happens when at least $n_\epsilon$ number of defence nodes detects attack at a time and there is an attack towards the victim. Therefore we only use the total bitrate estimations during the attack time to fit a Gaussian distribution. As shown Fig. 5.8(b) attack traffic also follows a Gaussian distribution, and hence we compute the probabilities of attack detection at each defence node during the time of attack. From the individual attack detection probabilities we can calculate system wide attack detection probability.

**Detection Delay**

We consider there to be no attack prior to a time $t_0$, therefore an experimental attack starts at $t_0$. During this period, each of the attack nodes generates attack traffic towards the victim. Consider that at distance $d$ the attack is detected at time $t_d$. The detection delay, $\tau$ is given by,

$$\tau = t_d - t_0$$
5.3.2 Performance against Flooding-based Attacks

The intention of the flooding-based attacks is to exhaust the victim with a huge volume of attacking packets and deny the victim’s services to its legitimate users. Since Gossip Detector makes attack decisions based on the victim’s traffic volume, Gossip Detector is able to detect all types of flooding-based attacks. The statistical performance of the detection procedures is described via receiver operational characteristics (ROC). In order to explore the performance of Gossip Detector with parameters such as overlay topology, flooding-based attacks and number of gossip rounds, we computed the performances, \( P_{ad} \), \( P_{fa} \) and detection delay for all possible \( n_c (1, 2, \ldots, n_d) \) and plot ROC curves: \( P_{ad} \) versus \( P_{fa} \) and detection delay versus \( P_{fa} \) as shown in Fig. 5.9.

Fig. 5.9 illustrates, the performance of Gossip Detector under constant rate attacks, which are the same as subgroup attacks. Thus, when attacker changes the direction of the attacks Gossip Detector is able to successfully detect the attack. The performance of Gossip Detector under increase rate attack is the same as constant rate attack as shown in Fig. 5.9(a) and Fig. 5.9(b), except that detection delay is slightly above the constant rate attack because the increasing attacks gradually increase the traffic volume over a period of time to slow exhaustion of a victim’s resources. As shown in Fig. 5.9(d), Gossip Detector detects the pulsing attack with higher false alarm probability than other types of attacks.

5.3.3 Overlay Topology

We place defence nodes at every router that is a deployment distance \( (d) \) from the victim, so that the defence nodes form a cooperative overlay network. Any victim traffic, which originates at a router at distance \( \geq d \) from the victim will pass through exactly one overlay node and the overlay node will count these packets. Let \( n_d \) be the number of such defence nodes then we define overlay topologies as:

\[
G(n_d, n_{nei})
\]
3. Performance Evaluation

![Graphs comparing different types of DDoS attacks](image)

(a) Constant-rate attack
(b) Increase-rate attack
(c) Subgroup attack
(d) Pulsing attack

Figure 5.9: Comparing $G(n_d, n_d \times 20\%)$ performance of Gossip Detector against different flooding-based attacks. $P_{fa} < 0.01$, $A_{rate}$ of all attackers is 0.150Mbps.
5.3 Performance Evaluation

Comprehensive Analysis of DDoS Attacks

Figure 5.10: Examples of overlay topologies, $G(n_d, n_{nei})$. $n_d=8$ and $n_{nei}=2, 3, 4, \text{ and } 7$. 
Where \( n_{nei} \) is the number of other defence nodes known to every defence node. We can have different overlay topologies at a distance by changing \( n_{nei} \) value. Obviously, the performance of Gossip Detector depends on the \( n_{nei} \) value. The Fig. 5.10 shows some of the possible overlay topologies located at a distance \( d \) from the victim with \( n_d = 8 \) and \( n_{nei} = 2, 3, 4, \) and 7.

We measure the performances of Gossip Detector by setting up different overlay topologies, \( G(n_d, n_{nei}) \) where \( n_{nei} = 10\%, 20\%, ..., 100\% \) of \( n_d \) deployed at a distance. The Fig. 5.11 illustrates the performance results of Gossip Detector for different overlay topologies. The results clearly show that when we increase the \( n_{nei} \) value, Gossip Detector detects attack early and false alarms are reduced. Fig. 5.11(c) is the result for Gossip Detector with fully connected overlay topology. In this case, one round of gossip is enough to calculate average bitrate towards victim and thus more than one gossip rounds in Gossip Detector is useless. Even though it is the best performance among the results shown in Fig. 5.9(a) and Fig. 5.11(b), it consumes more network bandwidth than others. From the results, we can conclude that Gossip Detector with higher numbers of neighbours gives better performance than fewer numbers of neighbours. When we increase \( n_{nei} \) value in Gossip Detector the detection delay becomes lower but it causes in increased bandwidth consumption.

5.3.4 Number of Gossip Rounds

When we increase the number of gossip rounds within a phase interval, the total bitrate estimation of the averaging algorithm becomes more accurate. Thus, Gossip Detector achieves a better performance when we set higher number of gossip rounds in Gossip Detector. We experimentally investigated the performance with different numbers of gossip rounds for all types of flooding-based attacks. The Fig. 5.11 illustrates the performance of Gossip Detector with number of gossip rounds at 1, 2 and 3. Fig. 5.11(a) shows that detection delay decreases when Gossip Detector has 3 gossip rounds, rather than 1 and 2 gossip rounds. Furthermore, the detection probability increases when we set more gos-
5.3 Performance Evaluation

Comprehensive Analysis of DDoS Attacks

sip rounds in *Gossip Detector*. In Fig. 5.11(c) the same performance results are observed for all number of gossip rounds because in this scenario *Gossip Detector* contained fully connected overlay topology and one gossip round was enough to estimate total victim’s bitrate accurately.

![Graphs showing performance results](image)

**Figure 5.11:** Comparing Constant-rate attack’s performance for different overlay topologies and number of rounds in an experimental topology. $P_{fa} < 0.01$, $A_{rate}$ of all attackers is 0.150Mbps.

### 5.3.5 Bandwidth Consumption

We explored the various parameter settings of *Gossip Detector* and in each case, we measured the network resources that were consumed by *Gossip Detector*. Among the network resources, we concentrated on the bandwidth consumption of *Gossip Detector*. The ROC curve shown in Fig. 5.12 illustrates the trade offs between the percentage of bandwidth consumption of *Gossip Detector* and detection delay performance for various parameter settings. If bandwidth is not an issue in the network system then the *Gossip Detector* with higher number of neighbours will be a better configuration for the *Gossip Detector*. Ide-
Figure 5.12: Comparing trade-offs between detection delays and bandwidth consumption. The numbers along the symbol denote $P_{ad}$. $P_{fa} < 0.01$ for all results. $A_{rate}$ of all attackers is 0.150Mbps.
ally, lowest number of gossip rounds and maximum number of neighbours will be the balanced choice for the parameter settings.

### 5.3.6 Network Topology

Since *Gossip Detector* shares attack information via the underlying of the victim’s network for the attack detection process, the performance of *Gossip Detector* directly depends on the communication delay of the victim’s network. Fig. 5.13 compares the detection delay of *Gossip Detector* in two network topologies, which have different communication delay. When the average communication delay among network routers increases the detection delay of *Gossip Detector* also increases as illustrated in the Fig. 5.13.

![Detection Delay Comparison](image)

**Figure 5.13: Comparing all type of attacks, $P_{fa} < 0.01$, $n_{nei} = 100\%$, $\tau_p/\tau_r = 1$.**
5.3.7 Deployment Distance

We consider the distance between Gossip Detector and the victim as the deployment distance. The deployment distance of our defence system makes a great impact on the attack detection delay. When we increase the deployment distance, typically two things happen. Firstly, the defence nodes become closer to the attacking sources, and secondly, the number of defence nodes increases in the defence system. We already discovered in theoretical analyses that when we increase the number of defence nodes, detection delay comes down in our defence system.

The experimental result, illustrated in Fig. 5.14, clearly shows that detection delay comes down to deployment distance increases which is the same trend that we discovered in the theoretical model. This result is for constant rate attack and all other types of flooding-based attack also have the same results.

![Detection delay variation with defence deployment distance.](image)

Figure 5.14: Detection delay variation with defence deployment distance.

We also measured the probability of attack detection when we increase defence de-
5.4 Comparison to Existing Work

Comprehensive Analysis of DDoS Attacks

As shown in Fig. 5.15, the detection probability is increasing as distance increases, the same conclusion we already discovered in theoretical analysis.

![Graph showing probability of attack detection vs. distance](image)

Figure 5.15: Comparing probability of attack detection at different deployment distance.

As in Fig. 5.16, the probability of false alarm comes down when we increase deploy distance. This results conclusion that with increased number of distance or defence nodes, $P_{fa}$ and detection delay decreases and $P_{ad}$ increases. Thus, by increasing deployment distance of *Gossip Detector* we can achieve early attack detection and required $P_{ad}$ while keeping $P_{fa}$ under control.

### 5.4 Comparison to Existing Work

Table 5.2 compares *Gossip Detector* with other related defence systems. The change point detection schemes [89] [92] are easy to deploy and have low computational overhead. However such detection systems result in relatively higher detection delay than *Gossip Detector*. Wavelet analysis [12] also detects attack with relatively higher detection delay
5.4 Comparison to Existing Work

Table 5.2: Comparing Gossip Detector with other defence proposals

<table>
<thead>
<tr>
<th>Detection method</th>
<th>Experimental method</th>
<th>Attack model</th>
<th>Detection delay</th>
<th>Detection results</th>
<th>False alarm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change-point detection [89]</td>
<td>ns-2 simulation of 100 nodes</td>
<td>TCP, UDP, and ICMP floods by abrupt and linear increase</td>
<td>1 to 36 secs</td>
<td>*</td>
<td>0.01 to 0.06</td>
</tr>
<tr>
<td>Change-point detection [92]</td>
<td>Three private network data sets</td>
<td>TCP SYN constant rate flood attack</td>
<td>20 to 480 secs</td>
<td>0.7 to 1</td>
<td>*</td>
</tr>
<tr>
<td>Detection technique based on TTL value [104]</td>
<td>ns-2 simulation of 100 nodes</td>
<td>Constant Bit Rate (CBR) flows in ns-2</td>
<td>on average 0.10 secs</td>
<td>0.2 to 1</td>
<td>0.0023 to 0.0179</td>
</tr>
<tr>
<td>Wavelet analysis [12]</td>
<td>Three weeks’ worth of university data</td>
<td>flood attacks of 4x, 7x, and 10x intensities</td>
<td>on average 25 secs</td>
<td>0.47</td>
<td>approx. 0.21</td>
</tr>
<tr>
<td>Our defence system</td>
<td>ns-2 simulation of 100 nodes</td>
<td>Const. rate atk, Incr. rate atk, Pulse atk and Subgr. atk</td>
<td>Refer Fig. 5.13 for detection delays</td>
<td>$P_{ad} \geq 0.9$</td>
<td>$P_{fa} \leq 0.01$</td>
</tr>
</tbody>
</table>

* values are not given in the corresponding references
5.5 Summary

Comprehensive Analysis of DDoS Attacks

and lower detection rate. Furthermore it has more computational overhead. The defence technique in [104] utilizes the packets’ TTL value to calculate mean distance values of the packets with abnormalities considered to indicate flooding based attacks. The attacks are detected close to victim using single point detection with detection delay of about 100ms. Gossip Detector can detect attacks in distributed defence nodes with an average detection delay of 32ms and a detection rate of 0.99, when appropriate parameters are used.

5.5 Summary

Being able to detect DDoS attacks as early as possible is important as it provides more time to undertake appropriate defence procedures. Our proposed distributed solution detects attacks using an overlay network, deployed by the routers, at a given distance from the victim. It uses gossip-based averaging between the nodes in the overlay so that every node can compute an aggregate for the traffic rate in order to make a local decision.
Distributed responses to the attacks are thereby feasible to undertake and we intend to continue our research in this area. Other future work includes research into gossiping based approaches that can detect attacks to any destination in the network, rather than just a given destination.
Chapter 6
Conclusion

After analysing existing Distributed Denial of Service attack defence proposals in Chapter 2, we found that the major challenge of DDoS attack defence is to detect DDoS attacks reliably and efficiently at a time significantly before the attack is detected at the victim end. To address these challenges, we have proposed Gossip Detector which is a cooperative distributed defence system that successfully detects a broad range of flooding-based DDoS attacks. Gossip Detector achieves excellent performance and it provides proactive attack detection which is the most significant requirement of any DDoS defence system. In this chapter, we summarise the DDoS problem, the features and performance of Gossip Detector, and the lessons learned in the course of this dissertation work.

6.1 Summary of the DDoS Problem

Security experts are still struggling to devise an effective solution to the DDoS problem. The attackers are becoming more sophisticated and they use legitimate packets that a victim cannot ignore. DDoS attacks strategically avoid detection, while still performing successful attacks. Although many commercial and research defences have already been proposed, none of them provide complete protection from the threat. They detect a small range of attacks that either use malformed packets or create severe disturbances in the network and they handle those attacks by dropping a portion of the traffic destined for the victim.
In the meantime, more and more machines are connecting to the Internet via broad-
band connections and wireless devices. These machines are mostly lacking in proper
security against intrusions, and a high-bandwidth Internet connection makes them easy
zombie recruitment targets, subject to large-scale attacks. Furthermore, we increasingly
rely on the Internet for various services, as it offers great convenience and speed. This
makes many Internet sites significant resources that should be accessible at all times.

DDoS defence must be distributed, as a large number of defence nodes can only be
defeated with an equal amount of attacking power. Since it is difficult to assure wide
deployment of a single defence system, defences must be ready and able to interface
with one another to combat attacks. Ideally, a defence system should be independent so
that it can work in isolation and still achieve good performance. It should also be able
to combine its actions with other systems to further improve performance. Our Gossip
Detector meets all of these goals, as demonstrated in both the theoretical and experimental
evaluation results. At the same time a DDoS defence must be proactive to protect the
victim before damage occurs.

6.2 Gossip Detector solution

Gossip Detector approaches the DDoS problem from a new direction. It is a distributed
defence solution whose goal is to detect attacks at an early stage. It uses a cooperative
overlay network with nodes that are distributed throughout the Intermediate Network.
A gossip-based scheme is used to compute an estimate of the total amount of attack traffic
that is headed towards the victim. In its design and implementation, Gossip Detector
adheres to several principles that are main contributors to its good performance.

Generally, detecting the abnormal behaviour of attack near the victim is easy. How-
ever, it is often too late to detect the DDoS attack at the victim network. The attack should
ideally be stopped as close to the sources as possible, saving network resources and re-
ducing congestion. However, there are no common characteristics of DDoS streams,
which can be used to detect the attacks near the source [17]. To balance this trade off, in this research we try to detect the DDoS attacks at the intermediate network. As the traffic is not aggregated enough in the intermediate network, current single deployment detection systems cannot detect DDoS attacks with a high degree of accuracy. To improve the defence efficiency and accuracy, Gossip Detector consists of a diverse collection of independent defence nodes located in the intermediate network of the Internet. The Gossip Detector scheme provides reliable, rapid and widespread cooperation among individual detection nodes to improve the accuracy of DDoS detection in the intermediate network. The following points illustrate the main contributions of Gossip Detector:

- Early attack detection obtained through defence deployment close to the attacking sources. The evaluation results of Gossip Detector using the ns-2 simulator illustrate that the proposed approach is both efficient and feasible. The experimental results show that Gossip Detector can detect attacks within 0.5 seconds, with a detection probability as high as 0.99 and probability of false alarm below 0.01 on a topology of average router delay of 12 ms. This compares favourably against other widely known methods including change-point detection, TTL analysis and wavelet analysis.

- Given the large scale of the Internet and the purpose of this infrastructure, we need a resilient and scalable communication mechanism to exchange the attack information. The simple gossip based communication mechanism is used to exchange attack information among detection nodes and to conclude the overall network-wide attacks effectively and accurately.

- The nature of DDoS suggests that a distributed mechanism is necessary for a successful DDoS attack defence. The Gossip Detector is a distributed defence system which consists of the distributed defence nodes at intermediate networks to detect DDoS attacks independently.

- To make reliable and rapid attack detection, Gossip Detector utilises a peer-to-peer
overlay network which removes central points of failure and its associated performance bottleneck.

More than the design of *Gossip Detector*, this research has been an exploration of the DDoS problems and possible solutions. This effort involved in the creation of attack and defence taxonomies enhances the understanding of the problem and solution space, and facilitates the design of better defences. In parallel, we have also defined a new set of metrics that can be used to measure the performance of a distributed defence system.

### 6.3 Future Work

In this section we outlined our research plans and goals to be achieved in future research. Our proposed early detection defence system contains many parameters that should be optimized to work effectively. When we performed evaluations of the proposed defence scheme, we always deployed the same gossip-based algorithm and we did not test *Gossip Detector* with real network trace data, etc. In order to verify the effectiveness of *Gossip Detector* in all scenarios, we need to research further and find optimal parameter settings.

Furthermore, *Gossip Detector* was originally designed to protect only one victim. In future, we will improve *Gossip Detector* to protect multiple targets of a network. In order to minimise the loss caused by DDoS attacks, a reaction scheme must be employed when an attack is underway. In other words, after attack detection, the next step is to respond to an attack. Regardless, our defence system is incomplete because it only detects flooding-based attacks, it does not act to protect the target. In order to be a complete defence mechanism, we need further research to implement a new reaction scheme or deploy existing reaction models in order to throttle malicious attacking packets.

Sophisticated DDoS attacks attempt to avoid detection of the defence systems by mimicking the traffic patterns of flash crowds. This poses a critical challenge to those who defend against DDoS attacks. However the study [107] shows that the current attack flows are usually more similar to each other compared to the flows of flash crowds.
As a result, the aggregated attack flows possess a small standard deviation compared with that of a flash crowd. This characteristic can be easily deployed in Gossip Detector to differentiate the DDoS attacks from flash crowd.

Low-rate distributed denial of service attacks [106] have significant ability of concealing its traffic because it is very much like normal traffic. They have the capacity to avoid detection by Gossip Detector as it is a anomaly–based detection scheme. However Gossip Detector can detect pulsing type of attacks.

6.4 Closing Remarks

DDoS attacks are continuing to increase in both size and complexity and the motivations behind attacks have also broadened. Worries spread as a result, from social networks to governments at risk of attack. The number of DDoS attacks continues to increase, and DDoS remains a growing threat. Administrators need to understand that traditional victim-end defences are no longer enough to protect a network or the Internet services. Trying to extend the capabilities of traditional solutions to defend against DDoS attacks has proven to be ineffective. It is important to note that new solutions are urgently needed for protection against DDoS attacks.

Distributed Denial of Service requires a distributed solution. This thesis has presented crucial building block of this solution - a distributed defence system that provides a proactive detection system. It can offer excellent performance, even in autonomous operation. This performance is further enhanced if it is combined with a good responsive defence system. We believe that these features will lead to quick and widespread deployment of the system, and integration with other systems, and thus to further improvement of Internet security.


