Chapter 2

Related work and Background

In the previous chapter, I presented an introduction to different types DDoS attacks. In this chapter I, present a brief discussion of some of the related works from the literature. Some of the basic concepts used in the research are also discussed in this chapter. Towards the end of this chapter different datasets used in the experiments of the research are discussed.

2.1 Introduction

Over the years many researchers have proposed a wide range of different mechanisms to defend against different types of DDoS attacks. These solutions usually make different assumptions of the problem. For example, some of them assume the solution to be independent of the intermediate network while others assume cooperation from the intermediate routers. Similarly, the computational model used in these defense solutions also varies widely. This chapter presents a discussion of different types of DDoS defense systems proposed in the literature. This chapter also discusses some of the background concepts used in the thesis.

The rest of the chapter is organized as follows. Section 2.2 presents different types of DDoS defense mechanism based on various perspectives. Section 2.3 presents a brief discussion of the related works done based on their computational model. Section 2.4 presents some of the background concepts used in the research. Section 2.5 discusses and summarizes the chapter. Section 2.6 discusses different datasets used in the research.

2.2 Types of DDoS Defense Mechanisms

This section presents a brief discussion of different types of DDoS mitigation schemes based on different perspectives.

Based on activation time of the defense system they can be classified as

1. *Proactive*: A proactive solution attempts to make a potential victim DDoS immune. Such solutions typically apply techniques like admission control, packet dropping based on priority and on demand resource allocation. In [19], the authors propose a server centric router throttling mechanism where each border router dynamically adjust the incoming traffic towards the protected server. In [20], the authors propose a proactive surge protection mechanism to defend against volumetric DDoS attack. In [21], the authors provide an application layer solution using the concept of fair bandwidth allocation among the clients.

2. *Reactive*: A reactive solution allows the DDoS attack to occur first and then provide solutions to detect the attack [15, 23, 24, 25] in near real time and take necessary actions to mitigate the attack. The post attack detection actions normally include traceback [26, 27, 28, 29, 52] of the source IPs (spoofed or real), pushback [51] and packet dropping [53].

The researchers make another assumption while proposing a solution is the deployment location of their solution. Based on the deployment location, the defense mechanisms can be classified into the following categories

1. *Source end*: In source end solutions [30], the defense mechanism attempts to block suspicious/ anomalous traffic at the source itself in order to prevent the zombies participate in a DDoS attack. The most commonly taken approach is Ingress/Egress filtering at the edge routers of a network. Source end solutions are usually the most effective one. However, there is no deployment incentive for such approaches.

2. *Victim end*: In this approach, it is considered that the rest of the Internet is non-cooperative and the attack is tried to detect and mitigate at the victim site. Such an approach can be installed at the edge routers of a network or at IDS/firewalls protecting the subnetwork of the victim. Some of the victim end solutions in the literature are [5, 30, 31].

3. *Network based*: In this type of defense, it is assumed that at least some part of the Internet is cooperative against the DDoS attack [27, 29, 33].

4. *Hybrid*: In such an approach defense is normally distributed in nature and try to defend the attack by deploying defense at different locations of the network and maintaining cooperation among them [34, 35]

The solutions to DDoS attacks are generally layer specific too. Based on the layer at which the traffic is monitored to detect an attack, defense can be categorized as

1. *Application layer solution*: The application specific features like request rate, user interaction time and user profile are examined to detect DDoS attacks. Some of the application layer DDoS detection mechanisms are discussed in [4, 37, 38]

2. *Network layer solution*: Network layer information like packet rate, IP field and link utilization are used to detect and mitigate attack [5, 15, 23, 24, 25, 30, 31].

2.3 Related Work

In this section some of the related works from the literature have been reported. I group them based on the main feature of the defense model.

2.3.1 Statistical Approaches

One of the most commonly used statistical approaches in the literature is information theoretic approach. In [31], Feinstein et al. propose a detection mechanism which computes the entropy and frequency sorted distribution of selected packet features to detect an attack. Yu and Zhou [15] consider the variation in entropy to detect the DDoS attack surge. Lee and Xiang [23] use different information theoretic measures such as entropy, relative entropy, information gain, information cost and relative conditional entropy to detect DDoS attack. Yang et al. [33] use generalized entropy and information distance to detect low-rate DDoS attack. In [39], the authors compare the performance of different entropy measures such as Hartley entropy, Shannon entropy, Renyin's entropy and Renyin's generalized entropy in detecting DDoS attacks of different rate. In [5], the authors propose a detection based on TCP SYN vs FIN pairs. Although this method is capable of detecting

DDoS attack with a high accuracy and low detection time, however, the method will be ineffective if the attacker generates the attack traffic with well-proportioned SYN and FIN packets. In the work of [72], [30] the authors maintain a database of trustworthy IP addresses. The source IPs of the incoming packets are verified against the database. More number of new IPs indicate an attack in their detection mechanism. They use CUSUM to detect the change in the number of new IP addresses to detect a DDoS attack. Chen et.el [34] propose a distributed change point detection mechanism deployed over multiple Autonomous System (AS). Every AS maintains a server called Change Aggregation Tree (CAT) server. The routers in the cooperative AS detects the change in traffic surge and reports to the CAT server. The local CATs are then merged at the victim end to confirm and mitigate an attack. F-divergence is used as a detection mechanism in [118]. They calculate the Total Variation Distance (TVD) both horizontally and vertically. To calculate horizontal TVD they use the common connections between two conjugative windows. Vertical TVD is calculated from the difference between the incoming and outgoing traffic of a victim. Although they considered bidirectional communication as a necessary criteria for a valid communication, they did not consider the types of these bidirectional packets. Under an attack also bidirectional communication may be observed due to the response messages from the victim. For example, in case of a TCP SYN attack, the victim responds to the attack SYN packets by sending acknowledgment to the source IP of the packet. Thus at the victim end, the bidirectional pattern might not be disturbed significantly. An entropy based detection mechanism is presented in the work of [33]. They present a detailed comparison of different entropy measures and establish generalized entropy as a better measure over others. Their method needs the full control over all the routers, which might not be possible in a real time environment. Also this model assumes that the attack and normal traffic follows different but specific distributions. A classification based approach is presented in [42]. They extracted the distribution of different TCP flags into bands and used a Naive Bayesian classifier to detect DDoS attack. This method leaves scope for the attacker to mimic idle flag distribution in the attack traffic. Correlation based detection is proposed in [24]. Jin and Yeung [25] discuss the effects of multivariate correlation analysis on the DDoS detection and proposes a covariance analysis model for detecting SYN flooding. In [40], the authors propose a rank co-relation based method to detect low-rate DDoS attack. In [53], the authors propose a DDoS detection and mitigation system based on the aggregate traffic volume coming from a specific prefix. In [54] the authors propose a statistical technique to differentiate between high rate and low rate DDoS attacks. Authors in [126] introduce a shrew DDoS defense mechanism based on spectral analysis. Another shrew attack detection and mitigation technique is discussed in [108]. The proposed method uses frequency domain characteristics of the incoming flows to the protected server to detect and mitigate shrew attack. In [130, 131, 132], the authors propose different statistical filtering approaches to discriminate normal packets from benign packets. CBF[133] is another statistical approach where incoming packets are assigned a confidence score based on its different attribute values to defend DDoS attacks in the cloud. In [105], the authors evaluated the performances of different statistical distance measures such as the Jeffrey distance, the Sibson distance, and the Hellinger distance to discriminate DDoS attacks from flash crowd. In [78] the authors propose a DDoS mitigation technique based on correlation pattern suitable for cloud computing environment. In [79] authors propose an entropy based DDoS mitigation technique, which is capable of discriminating flash crowd in VOIP network. In [80] authors discuss a method to prevent DDoS attack which uses IP spoofing to perform the attack.

2.3.2 Supervised Machine Learning Based Approaches

In supervised machine learning approach a model is trained from historical data and then the trained model is used to forecast or classify the incoming packets/flows. In [41], the authors propose a learning algorithm to train such a model. Xiang et al. [35] use neural network at distributed points over the network to detect and filter the attack packets. Their method uses the mark field of the IP header which is used by different packet marking schemes to identify the attack packets. In [42], the authors propose a Naive Bayesian Classifier which models TCP and UDP traffic separately from historical data to classify the incoming traffic as attack or normal. In [43], the authors propose a ensemble classification technique, RBPBoost, which uses Neyman Pearson cost minimization technique to classify incoming traffic. Mansour et al.[44] use a multi-layer perception (MLP) neural classifier to discriminate the attack flows from benign flows. An application layer DDoS attack detection mechanism is proposed in [38] based on hidden semi-Markov model. History of different user's browsing behavior is used to train the model to classify the ongoing communications. In [55] the authors propose an artificial neural network based DDoS defense mechanism to detect both known and unknown DDoS attacks. Yi Xie et al. in [4] propose a system to differentiate application layer DDoS attack from flashcrowd. In their method a hidden semi-Markov model is used to model the access behavior of normal users. They used an entropy based measure is used to detect the application layer DDoS attack. In [68] the authors proposed a neural network based DDoS defense mechanism in SDN paradigm.

2.3.3 Unsupervised Machine Learning Based Approaches

Unsupervised machine learning techniques learn the characteristics of a given system from unlabeled example data. In [45], the authors discuss a clustering based DDoS detection mechanism. The incoming traffic is clustered based on different packet/flow level features in order to detect DDoS attacks. Cheng jin et al. [46] propose a DDoS defense mechanism where clusters are first extracted based on the hop count value of the historical Internet traffic. Incoming packets are then compared against these clusters to defend spoofed DDoS attacks. In [37], the authors propose an application layer anomaly detection method based on K-means clustering. An SDN network based DDoS defense mechanism is proposed in [129] where "Self Organizing Map(SOM)" technique is used for flow analysis in the SDN controller.

2.3.4 Packet Marking and IP Traceback Based Approaches

In packet marking and IP traceback based approach, the intermediate routers insert a mark in the ID field of the IP header. Such marks are used by the victim to perform traceback, packet dropping or attack path construction to defend against spoofing based DDoS attacks. In IP traceback, the goal is to reach the actual source of one or more packets from the victim end irrespective of the SIP addresses carried by the packets. Different IP traceback mechanisms have been proposed such as marking based IP traceback, logging based IP traceback and hybrid IP traceback mechanisms. In marking based traceback schemes such as [28],[75] and [74], one or more intermediate routers inscribe their IP addresses into the forwarded packets with some predefined probability. Such techniques are known as Probabilistic Packet Marking(PPM) schemes. Once the victim gathers enough attack packets, it can deduce the paths traversed by the attack packets. Such schemes put heavy computational overhead at the victim side. In [29],[113] and [91], the authors propose schemes in which the edge routers of the source network mark the outgoing packets deterministically with their IP addresses. Such schemes are called as Deterministic Packet Marking(DPM) schemes. From the received packets the victim can deduce the address of the first router traversed by the attack packets, even if the source IP addresses of the packets are spoofed. In both PPM and DPM schemes although the victim can infer the true identity of the attack sources however, the attacker cannot perform single packet level discrimination of spoofed packets from benign packets. In logging based traceback schemes such as those proposed in [114], [92], and [94], signature of the forwarded packets are maintained (logged) in the intermediate routers. When a victim detects an attack, the suspicious packets are traced back to their sources by following the logs maintained by the intermediate routers. The authors of SPIE[138] facilitate single packet traceback by logging unique digest for all passing packets using bloom filters in the intermediate routers along the path from the source to the destination. SPIE needs 0.5% of the total link bandwidth per unit time storage at a router. In high speed links this storage requirement becomes impractical. In HIT[127], the authors propose a hybrid of both packet marking and logging techniques which, needs only 50% of the routers to log a particular packet. Thus, an RIT enabled router needs to store digest of only 50% of the packets passing through it. In PPIT[100], the authors propose another hybrid traceback technique which reduces the number of packet digests needed to be logged at a router to 39%. The authors in MRT[122] and MORE[123] propose different techniques to reduce the frequency of log operations and thus, the storage requirement at a router. Some of the other such approaches are [109] and [93]. Both logging based and hybrid schemes are not capable of distinguishing the spoofed packets from benign packets at the victim end. In HCF[46], the authors demonstrate how the hop count value (derived from the TTL of a received packet) can be used to discriminate spoofed packets from benign packets. HCF is based on the observation that Internet routes are stable over a period of time hence, the hop count value associated with a SIP address with respect to a particular destination does not change frequently. Based on this idea, the authors propose to maintain a table in the victim side which maps an IP address to its hop count value. As soon as a packet is received by the victim, it's derived hop count value is compared against the stored hop count value. A mismatch is considered as a spoofing incident. The main advantage of this scheme is that it does not assume any support from the intermediate network. However, since the hop count values are distributed over a very small space, mostly between 0 to 31, an attacker can easily make the victim to accept large number of spoofed packets by randomly setting the initial TTL values of the attack packets. In [95], the authors propose another hop count based DDoS defense system. In PI[26], the authors propose a technique which is based on the observation that from a given source most of the packets travel the same sequence of routers to reach a particular destination. In their approach the 16 bit ID field of a packet is used to carry the signature of the traveled path by the packets. Each intermediate router inserts a specific number of bits in the ID field of the forwarded packets based on their TTL values. In StackPI[27], the authors propose an enhanced version of the PI scheme. In their approach, each intermediate router pushes k bits (k=1 or 2) into the ID field of the forwarded packets, considering the 16 bit ID field as a stack. Thus, only the last $16/k$ routers decide the final mark of the received packets. Since the routers within the destination network are less informative in terms of path information than the routers closer to the sources, this method assumes that the routers in the destination network do not participate in the marking process. In ANTID[107], the authors propose a marking scheme where the mark carried by a packet comprises of two fields, a n bit path identification field which encodes the path traversed by the packet to reach the destination from its source and a d bit distance field which represents the distance of the received packet from its source. Typically, the 16 bit ID field of a packet is used for marking process. As most of the Internet paths are within the range of 1-32 hops, 5 bits are required for the distance field. The authors suggested the allocation of the 16 bits between n and d to be 11 and 5 respectively.

2.3.5 Rate Limiting Based Approaches

In this category of defense system, the network resources are allocated among different traffic aggregation based on historical values. Any traffic aggregation which uses more resources than its fair share are either discarded as anomalous or marked as low priority packets which the intermediate routers can drop in case of congestion. In [19], the authors propose a server centric router throttling mechanism. In their approach, the protected server communicates its current load with a set of border routers. The border routers in turn applies dynamic traffic throttling to protect the victim from a DDoS attack. In [20] the authors propose a proactive surge protection technique to defend a potential victim against volumetric DDoS attack. They maintained a link utilization matrix to represent the traffic volume between any entry and exit point of the network. The incoming packets are marked as either low priority or high priority based on the current link utilization of the specific link. When a router detects congestion it performs a priority based dropping. In [21] the authors provide an application layer solution using concept of fair bandwidth allocation among the clients. They use a game theoretical approach to perform the bandwidth allocation.

2.3.6 SDN and Cloud Based approaches

In [59], the authors discussed a distributed DDoS defense system deployed over a set of SDN controllers. In [136], the authors provide an SDN based DDoS defense system by monitoring the flow characteristics at the controller. In [135], the authors discussed the potential DDoS threats towards SDN controlled network due to its centralized controller architecture. In [47], the authors proposed a resource management based DDoS defense mechanism for cloud services. In [137] the authors demonstrated a cloud based DDoS mitigation system where clones of IDSes are created based on the intensity of the attack. They used queuing theory based technique to estimate the need of the resources. Another cloud based technique is proposed in [104] where the defense system shuffles the clients among the attacked servers to create a moving target effect. In FACT [125], the authors proposed a authentication based cloud traceback framework. In [60], the authors demonstrate a collaborative DDoS defense system by using the concepts of blockchain and smart contracts.

2.3.7 Surveys

A detailed classification of different DDoS defense mechanisms can be found in [2, 16], where the authors discussed the pros and cons of different categories of defense mechanisms. Also, the authors pointed out different performance measurement and desired characteristics of a defense system. In [134] the authors discussed different filtering based DDoS attack defense mechanisms in detail. In [73], the authors discussed different cloud based DDoS mitigation systems to mitigate large DDoS attacks. In [49, 67, 70], the authors address different taxonomy, security threats and probable DDoS defense systems in cloud computing environment. In [82, 85], the authors discussed DDoS from the combined perspective of SDN and cloud computing.

2.4 Background

In this section I discuss some of the background concepts used in the research.

2.4.1 Cryptographic Hash Function

A cryptographic hash function is a a mathematical operation that transforms an arbitrary size data to a fixed size bit pattern, called as hash value. The operation one-way in the sense that retrieving the original data from the hash value is infeasible. Following are the main properties of a cryptographic hash function.

1. It should be a deterministic function. i.e., for the same input data the out put hash value is always same.
2. The function should be computationally light.
3. Reversing the function to generate the original data from its hash value should be infeasible.
4. The hash values should not be correlated with the input data, i.e., a small change in the input data should change the output hash extensively.
5. it is computationally infeasible to find two instances of different data which land in the same hash value.

the

Cryptographic hash functions are widely used information security related applications such as digital signatures and message authentication codes. Details about cryptographic hash function can be found in [56].

2.4.2 Hash Table

A hash table [57] is a data structure which allows to store and retrieve data based on their associated key. A hash table usually implements a hash function to compute a hash value of the given key which is then used to index an array where the actual data is stored. In ideal situation, the hash function should assign a different index for each given key. However, in practice hash functions usually map one or more keys to the same index. Such a situation is called as hash collision. Hash tables are usually more efficient than search trees which makes them more suitable for different computer applications such database indexing and caches.

A hash table can be evaluated by its load factor, which is defined as

$$load\ factor = \frac{n}{k}$$

where n is the number of entries inserted in the hash table and k is the number of slots in the hash table. A high load factor indicates a overloaded hash table which in turn may cause slow update and search operation.

2.4.3 Bloom Filter

A Bloom [58] filter is a space-efficient but probabilistic data structure that supports membership queries against a set of elements. To represent and support membership queries for a set $X = \{x_1, x_2, ...x_n\}$ of n elements, a Bloom filter uses m bits and a set of k hash functions, $H\{h_1, h_2, , , , h_k\}$ with range $(1, 2, ...m)$. For theoretical analysis it is assumed that each hash function spreads the n items in the set uniformly over the range $(1, 2, ...m)$, independent of each other. For practical implementation we chose the hash functions from a family of universal hash function. For each element $x_j \in X, j = 0, 1, ...n$, $m[h_i(x_j)]$ is set to 1, where $i = 0, 1, ..k$. For a membership query of an element y, if $\exists h_l \in H, l = 0, 1, ..k$ such that $m[h_l(y)] = 0$, then the element y is certainly not there in the set. If $\forall h_l, l = 0, 1, ..k$ $m[h_l(y)] = 1$, then we assume that the element is a member of the set with a certain probability of being wrong, referred to as *false positive rate*. Let $X = \{x_1, x_2, ...x_n\}$ be the set of n elements, m be the number of bits in the Bloom filter, k is the number of hash function used by the Bloom filter and fpr is the maximum tolerable false positive rate of the Bloom filter then the following equations can be used to show the relationship among the parameters.

$$k = ln\frac{m}{n} \tag{2.1}$$

$$fpr = (0.6185)^{\frac{m}{n}} \tag{2.2}$$

From Equation (6.2), we see that the *false positive rate* of a Bloom filter decreases if bits per element *i.e.*, $\frac{m}{n}$ increases. Also, from Equation (6.1) we see that the number of hash functions in a Bloom filter increases logarithmically with respect to $\frac{m}{n}$.

2.4.4 Datasets Used in the Research

The datasets used in my research are discussed below.

2.4.4.1 Normal Traces

These traces are used to study the characteristics of normal Internet communication in specific situations. Different parameters involved in the proposed defense systems are estimated using these traces. These traces are also used as normal background traffic to study the effectiveness of the proposed defense systems. Following are the normal traces used in the experiments.

1. AUK-VIII [61]: It contains wide area traffic between Auckland university and rest of the world. The AUK-VIII trace is collected from an edge router of the Auckland university network. The link is bi-directional in the sense that it carries both incoming and outgoing traffic to/from the network. An arrangement of the capturing point is shown in figure 2.1

2. WAIK-VIII [62] : It contains wide area traffic between Waikato university and rest of the world. Similar to AUK-VIII, WAIK-VIII trace contains the bi-directional traffic observed at an edge router of the Waikato university network. We use libtrace tool [63] to split AUKVIII and WAIK-VIII traces into corresponding incoming and outgoing traces.

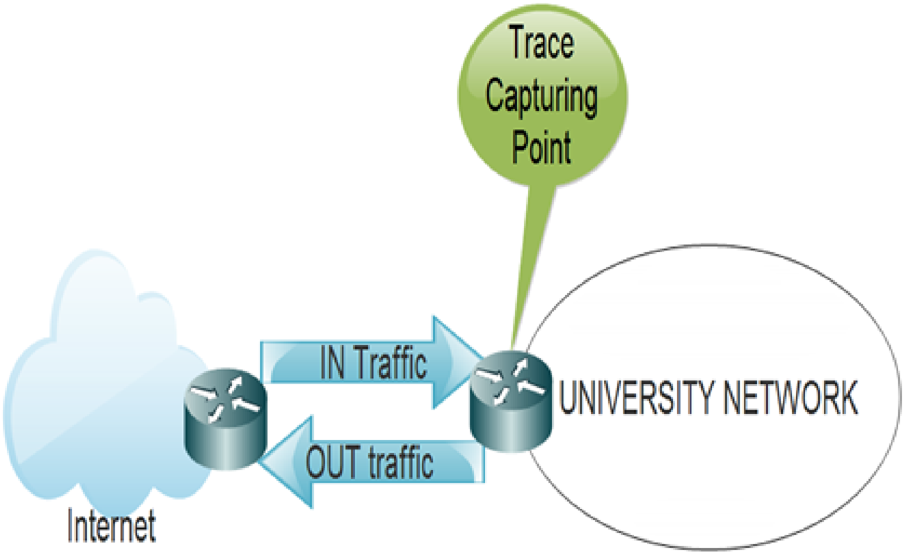

Figure 2.1: Packet capturing arrangement of AUK-VIII and WAIK-VIII network traces.

2.4.4.2 Attack Traces

Attack traces contain DDoS attack traffic, which are used to learn the attack traffic characteristics as well as to evaluate the performance of the proposed systems under different attack situations. Following are the attack traces used in the experiments.

1. DARPA [66]: This trace contains a randomly spoofed DDoS attack of around 5 minute duration.

2. CAIDA-2007 [65]: This attack trace contains one hour ICMP request DDOS attack traffic. This trace contains attack at different attack rate dynamics such as low rate, high rate, increasing and pulsing attacks. Figure 2.2 shows different attack rate dynamics of CAIDA-2007 DDoS trace.

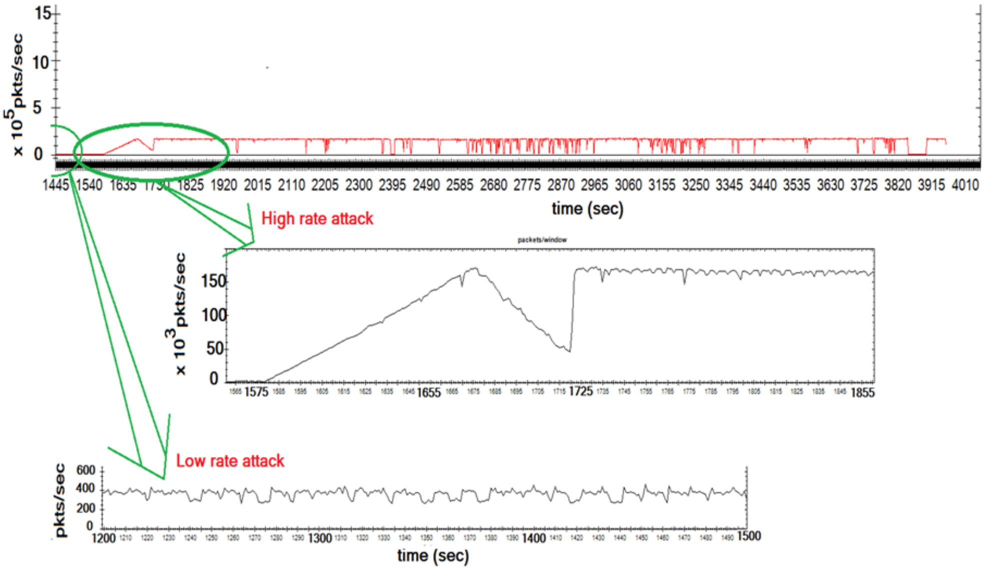

Figure 2.2: Anatomy of CAIDA-2007 DDoS attack trace

3. Test-bed generated attack traces: Other than CAIDA-2007 and DARPA, to evaluate the proposed defense mechanisms I also used DDoS attack trace generated in a test-bed environment. To generate DDoS attack traces with different specifications I developed a tool referred to as TUCANNON. I report the tool in Appendix-A .

2.4.4.3 Network Topology Datasets

Part of my research address the issue of mitigating spoofing attacks. The proposed methods against such attacks assume the participation of one or more intermediate routers in a network. To evaluate such techniques I used CAIDA's LNK0304 [64] Internet topology datasets to represent the intermediate routers.

These datasets contains trace route informations from selected monitoring points to different destinations. The datasets contain both complete as well as incomplete paths. Also for a single destination there are multiple paths in some of the datasets. For our experiments we consider only the complete paths. In case of multiple paths to the same destination from the monitoring point, we chose the longest path sequence. Thus, after preprocessing we get a tree, where the root of the tree is the destination and the leafs are the sources. Out of skirter's 25 monitoring points, we chose the topology generated from m-root, f-root and mwest monitoring points, in our experiments. However, topology related to other monitoring points also exhibit almost same characteristics.

2.5 Discussion

This chapter presents a brief discussion of different types of defense solutions proposed in the literature. This chapter also discusses different solutions based on their computational model. Some of the background concepts and the datasets used in the research are also discussed in this chapter. In the next chapter a lightweight DDoS detection mechanism has been discussed.