Interdisciplinary Information Sciences Vol. 19, No. 2 (2013) 173–200
©Graduate School of Information Sciences, Tohoku University
ISSN 1340-9050 print/1347-6157 online
DOI 10.4036/iis.2013.173

A Survey on DDoS Attack and Defense Strategies: From Traditional Schemes to Current Techniques

Muhammad AAMIR* and Mustafa Ali ZAIDI

Shaheed Zulfikar Ali Bhutto Institute of Science & Technology (SZABIST), Karachi, Pakistan

Received July 10, 2013; final version accepted October 8, 2013

Distributed Denial of Service (DDoS) attacks exhaust victim's bandwidth or services. Traditional architecture of Internet is vulnerable to DDoS attacks and an ongoing cycle of attack & defense is observed. A recent attack report of year 2013 — 'Quarter 1' from Prolexic Technologies identifies that 1.75 percent increase in total number of DDoS attacks has been recorded as compared to similar attacks of previous year's last quarter. In this paper, different types and techniques of DDoS attacks and their countermeasures are surveyed. The significance of this paper is the coverage of many aspects of countering DDoS attacks including new research on the topic. We survey different papers describing methods of defense against DDoS attacks based on entropy variations, traffic anomaly parameters, neural networks, device level defense, botnet flux identifications, application layer DDoS defense and countermeasures in wireless networks, CCN & cloud computing environments. We also discuss some traditional methods of defense such as traceback and packet filtering techniques, so that readers can identify major differences between traditional and current techniques of defense against DDoS attacks. We identify that application layer DDoS attacks possess the ability to produce greater impact on the victim as they are driven by legitimate-like traffic, making it quite difficult to identify and distinguish from legitimate requests. The need of improved defense against such attacks is therefore more demanding in research. The study conducted in this paper can be helpful for readers and researchers to recognize better techniques of defense in current times against DDoS attacks and contribute with more research on this topic in the light of future challenges identified in this paper.

KEYWORDS: application layer, attack, DDoS, network, security

1. Introduction

Denial of Service (DoS) attacks are very common in the world of internet today [1]. Increasing pace of such attacks has made servers and network devices on the internet at greater risk than ever before. Due to the same reason, organizations and people carrying large servers and data on the internet are now making greater plans and investments to be secure and defend themselves against a number of cyber attacks including Denial of Service.

The traditional architecture of World Wide Web is vulnerable to serious kinds of threats including DoS attacks. The attackers are now quicker in launching such attacks because they have sophisticated and automated DoS attack tools available which require minimal human effort. The attack aims to deny or degrade normal services for legitimate users by sending huge traffic to the victim (machines or networks) to exhaust services, connection capacity or the bandwidth. A few common types of DoS attacks (specified as remote DoS attacks in [1]) are given in Table 1.

2. Distributed Denial of Service Attacks

In a Distributed Denial of Service (DDoS) attack, the attacker makes a huge impact on the victim by having multiplied power of attack derived by a large number of computer agents. It becomes possible for an attacker because he takes large number of computer machines under his control over the internet before applying an attack. In fact, these computers are vulnerable machines in the public network and attacker can exploit their weaknesses by inserting malicious code or some other hacking technique so that they become under his control. These compromised machines can be hundreds or thousands in numbers. They behave as agents of the attacker and are commonly termed as 'zombies.' The entire group of zombies is usually named as a 'botnet.' The size of botnet decides the magnitude of attack. For larger botnet (increased number of zombies in a botnet), attack is more severe and disastrous.

Within a botnet, the attacker chooses 'handlers' which perform command and control functions and pass the instructions of attacker to zombies. The zombies directly attack on the victim. There is a group of zombies or agents under each handler. These handlers also pass the information received from zombies to attacker about the victim [2]. Therefore, handlers are the machines which directly communicate with attacker and zombies. As handlers and zombies are also compromised machines in the public network under attacker's control, the users of such machines are mostly

*Corresponding author. E-mail: aamir.nbpit@yahoo.com

Table 1. Some common DoS attacks [1].

Type of DoS Attack	Target	Exploited Vulnerability
Network Device Attack	Hardware Device (such as a Router)	Software bug in device's software.
Operating System Level Attack	Operating System (OS Services)	Bug in OS software.
Application Level Attack	Application Layer (Software Services)	Bug in victim software (usually identified through Port Scanning technique).
Data Flood Attack	Bandwidth or connection capacity of network	Limited bandwidth and server capacity to process requests (heavy traffic is sent towards victim to exhaust services).
Protocol Feature Attack	Protocol Services (mainly at network layer)	Limitation of a protocol such as IP address spoofing (Internet Protocol is a part of TCP/IP stack).

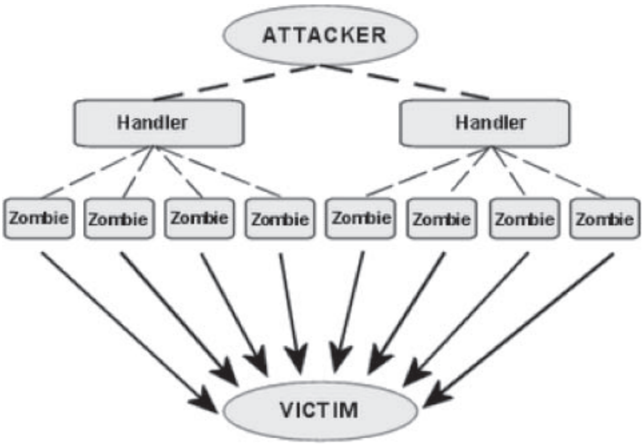


Fig. 1. Architecture of DDoS attack.

unaware of the fact that there machines are being used as a part of some botnet. A typical architecture of DDoS attack is mentioned in Fig. 1. The attack employs client server technology and a stream of data packets is sent to the victim for exhausting its services, connections, bandwidth etc. The data flood attack type of DoS is mostly used in DDoS attacks.

With the evolution of internet, cyber attacks have also increased manifold. Earlier DDoS attacks were manual where attacker had to perform many steps before the launch of final attack, such as port scanning, identifying available machines in the public network to create botnet, inserting malware etc. With the passage of time, sophisticated attack tools have been developed to assist attackers in performing all or some steps automatically to reduce human effort. The attackers can just configure desired attack parameters and the rest is done by automated tools.

Some common automated attack tools available are *Trinoo*, *TFN* (Tribe Flood Network), *TFN2K*, *Stacheldraht*, *Shaft*, *Knight* and *Trinity*. Some of them work on IRC (Internet Relay Chat) where handlers and zombies do not know identities of each other and the communication among them is done indirectly. The others are agent based in which communication is direct and handlers and zombies know each other's identity [3]. Therefore, when DDoS attacks are classified by the degree of automation, they are mentioned as *Manual*, *Semi-automatic* and *Automatic* attacks [1].

DDoS attacks are further classified by attack rate dynamics i.e., the way how rate of attack varies with respect to the passage of time. The classes are *Continuous Rate* and *Variable Rate* attacks [1]. In continuous rate, the attack has constant flow after it is executed. On the other hand, variable rate attack changes its impact and flow with time, making it more difficult to detect and respond. Within variable rate, the attack rate dynamics can further be implemented as *Fluctuating* or *Increasing* [1]. Moreover, based on the data rate of attack traffic in a given network, the attacks are also categorized as *high rate* and *low rate* DDoS attacks [4].

DDoS attacks are also classified in literature as 'by impact' i.e., it can be *Disruptive* in which the normal service is completely unavailable to users, or it can be *Degrading* in which the service is not completely unavailable but experiences considerable decrease in the productivity [1].

The major classification of DDoS attacks is 'by exploited vulnerability' [1] through which an adversary launches attack on the victim. The classification is given in Fig. 2 (as specified in [1]). In the said classification, *flood attack* is used to bring down the victim's machine or network's bandwidth. It has a few major sub-classes like *UDP flood*, *ICMP flood* and *TCP flood*. In fact, all flooding attacks generated through DDoS can be of two types; *direct attacks* and *reflector attacks* [5]. In direct attacks, zombie machines directly attack the victim as shown in the attack architecture in Fig. 1. On the other hand, in reflector attacks, zombies send request packets with spoofed IP (IP of the victim) in source address field to a number of other compromised machines (PCs, routers etc.) and the reply generated from such

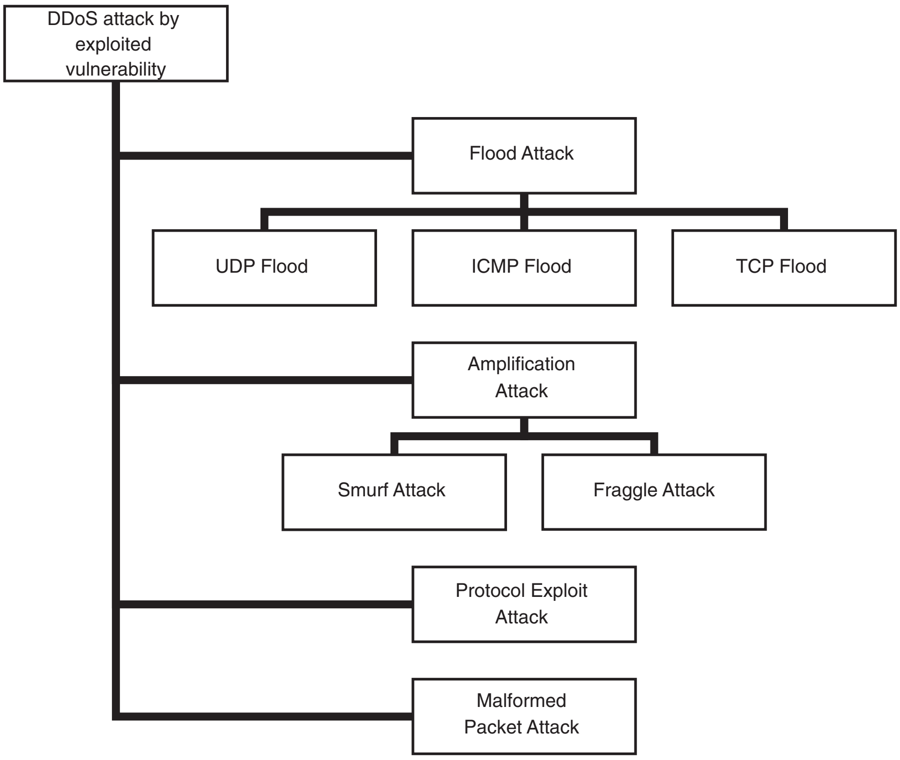


Fig. 2. DDoS attack by exploited vulnerability — Classification [1].

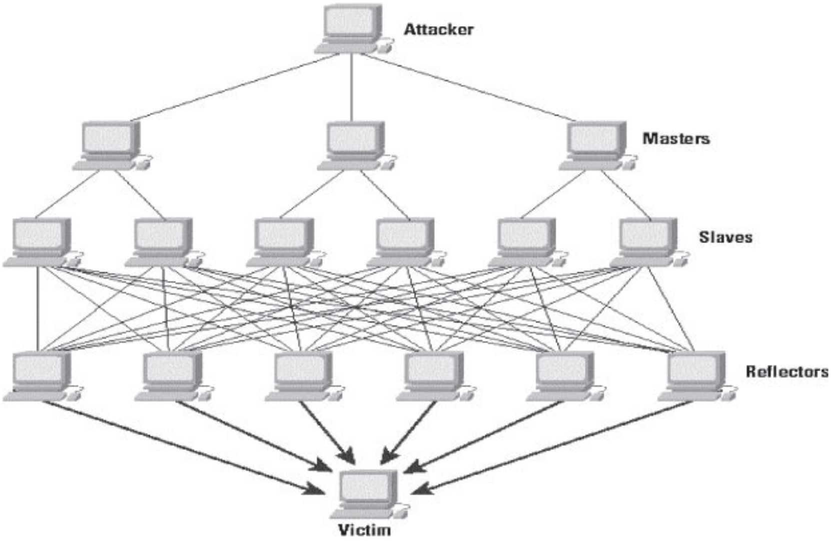


Fig. 3. Architecture of DDoS reflector attack ('Masters' represent Handlers and 'Slaves' represent Zombies) [5, 12].

machines is targeted towards the victim for an impact desired by the attacker. In such a way, reflection of the traffic is observed in these kinds of attacks. A classic example is sending 'ping' requests with spoofed source IP, and replies are targeted towards the victim. The goal of attacker launching such attacks is to saturate bandwidth of the victim with huge amount of traffic. The architecture of reflector attack is shown in Fig. 3. A brief description of attacks mentioned in Fig. 2 (obtained from [1]) is provided in Table 2.

2.1 Application layer DDoS attacks

In network layer or infrastructure layer (Layer 3) attacks, the malicious part resides in packet header or payload to compromise victim's CPU cycles, processing, bandwidth etc. However, with the introduction of sophisticated DDoS detection & mitigation tools, attackers have also started changing their strategies to avoid detection and mitigation by increasing their focus towards *application layer (Layer 7) attacks*. These attacks mimic legitimate traffic in

Table 2. Classification of DDoS attacks — By exploited vulnerability [1].

Type of DDoS Attack	Target	Exploited Vulnerability	Method of Attack	Impact
UDP Flood Attack	Server or Network (Software Services/ Connection Capacity or Bandwidth)	Limited bandwidth and server capacity to process requests	Heavy traffic (UDP Packets) is sent towards victim with randomly selected destination port.	Victim replies with 'Destination Host Unreachable' packets. When it is kept busy continuously beyond processing capacity, it crashes. Network's bandwidth is also exhausted.
ICMP Flood Attack	Network (Bandwidth)	Limited bandwidth	Heavy traffic (ICMP Packets) of 'Ping' requests is sent towards a machine on target network. They are sent directly or through agents for larger impact.	Massive traffic is generated leading in bandwidth saturation.
TCP Flood Attack	Server or Network (Connection Capacity or Bandwidth)	Limited bandwidth and server capacity to process requests	Heavy TCP traffic (packets) of legitimate-like headers and random payload is sent towards victim.	Massive traffic is generated leading in bandwidth saturation and degradation of server's CPU consumption.
Smurf Attack	Server or Network (Connection Capacity or Bandwidth)	Limited bandwidth and server capacity to process requests	Heavy traffic (ICMP Echo Packets) is sent towards victim with randomly selected destination port.	Massive traffic is generated leading in bandwidth saturation and degradation of server's CPU consumption (or it can even crash).
Fraggle Attack	Server or Network (Connection Capacity or Bandwidth)	Limited bandwidth and server capacity to process requests	Heavy traffic (UDP Echo Packets) is sent towards victim with randomly selected destination port.	Massive traffic is generated leading in bandwidth saturation and degradation of server's CPU consumption (or it can even crash).
Protocol Exploit Attack	Server (Connection Capacity)	Protocol feature (e.g., three-way handshake for TCP SYN attack)	Heavy traffic of spoofed SYN signals is sent towards victim. The ACK signals is not acknowledged by attacker.	Server waits for final acknowlegment for certain time. Buffer capacity is limited, hence it results in full queue buffer. New requests can not be processed.
Malformed Packet Attack	Server (Processing Capacity)	Limited processing capability of server	Malicious packets are sent towards victim with manipulated entries in IP address fields.	Server can not process malicious packets and can completely crash if traffic is too heavy.

communications to disturb or destroy victim's resources. Therefore, traditional DDoS detection techniques are unable to identify such attacks. In these attacks, complete communication with the victim is established just like legitimate users and numerous connections are generated aiming to deny or degrade the service or bandwidth for legitimate clients.

Application layer attacks are subject to the establishment of complete TCP connections with the victim. Therefore, the attacker has to disclose real IPs of zombie machines to the victim. Otherwise, it is not possible to make such connections. However, due to large number of zombies, the attacker does not worry about this attack limitation [5]. If such machines are identified and filtered at some stage, the attacker uses other group or pool of zombies to process the continuity of attack. After establishing TCP connections with the victim in a large number, the attacker starts communication through sending requests for relatively large processing such as downloading heavy image files or making database queries. In this way, resources are reserved against such attack traffic to deny or degrade the services for legitimate users. Effectively, application layer attacks are also flooding attacks and categorized as *HTTP flood*, *HTTPS flood*, *FTP flood* etc. Sometimes, they are collectively mentioned as *GET floods*.

2.2 Motivation behind DDoS attacks

People behind DDoS attacks may be motivated by personal, social or financial benefits. Attackers may do so due to personal revenge, getting publicity or some political motivation. Hoever, most DDoS attacks are launched by organized criminal groups targeting financial websites such as banks or stock exchanges. They also focus on targeting other finance related businesses such as e-commerce and gambling sites.

The financial impact of DDoS attacks on victims can be disastrous. In recent past, criminal groups have launched a number of attacks on stock exchange websites throughout the world. A few DDoS attacks reported in years 2011 and 2012 were on NASDAQ & BATS stock exchanges along with Chicago Board Options Exchange (CBOE), New York stock exchange and Hong Kong stock exchange [6–8]. As a consequence, severe incidents have been reported such as disruption of business activities of some major trading companies for some duration, resulting in financial losses to them. A recent attack report of year 2013 — 'Quarter 1' from Prolexic Technologies [9] indicates that 1.75 percent increase in total number of DDoS attacks has been recorded as compared to attacks of previous year's last quarter.

3. DDoS Detection & Mitigation

Distributed Denial of Service is a huge threat to the Internet today [10]. Attackers are now quicker to launch DDoS attacks with sophisticated attack tools, aiming to get financial benefits and other advantages by denying or degrading victim's resources for legitimate users. Numerous research papers have been published which review DDoS attacks and propose their detection & mitigation techniques. We find some comprehensive papers & articles to enrich our literature review on the topic [1–5, 11–29]. Some reviewed methods have also been referred in tables of first author's previous work [30]; however this paper contains major changes as compared to the previous study. Discussion on Prolexic attack report of year 2012 quarter 1 and performance evaluation in OPNET simulator constitute a significant part of previous publication whereas they have not been covered in this paper. The focus here is to discuss more DDoS mitigation methods with some description and cover a wide range of techniques of DDoS defense proposed and tested mainly in the near past. There are many papers & articles published on the topic providing survey on DDoS attacks and defense in detailed or limited scopes and proposing new methods of defense. Although we survey several new mitigation techniques against DDoS attacks in this paper, it is a fact that accurate detection and mitigation of DDoS attacks is still a difficult task as the traffic is so aggregated at network hops that it is not easy to identify attack packets within a mix of normal and attack traffic. Therefore, the authors believe that there is a potential scope of adding more survey papers on the topic mainly to cover recent research efforts and proposed techniques in order to add more knowledge in the domain and enable readers & researchers to get benefits in identifying better defense techniques of current times against DDoS attacks. In this way, they can be able to contribute with more research against future challenges in this area. In the following sections of this paper, we review several detection and mitigation mechanisms against DDoS attacks which are quite common and more promising in recent times such as statistical analysis of network traffic to estimate attack strength in real time, role of neural networks in real time attack analysis and research attempts to mitigate application layer DDoS attacks which are drawing more attention of attackers today. In addition to those, traditional methods of traceback such as packet marking, packet logging and pushback etc. are also discussed.

The ability of a DDoS detection and mitigation technique lies on its accuracy and reliability so that false positives and false negatives in a system can effectively be reduced i.e., it should not allow packets to pass through the mitigation mechanism that belong to the attack traffic (false negatives) and reach the victim, and it should also not drop packets that belong to the legitimate traffic (false positives). As far as the countermeasures against DDoS are concerned, they are usually categorized in three types of techniques mentioned below [24]:

- Survival techniques
- Proactive techniques
- Reactive techniques

In survival techniques, devices and systems which may be a victim of some DDoS attack are equipped with sufficient resources so that services may still be available for legitimate users in case of occurrence of a DDoS attack. The resources such as CPU power, bandwidth, memory etc. are made sufficient and redundancy of resources is also maintained wherever applicable.

In proactive techniques, the aim is to detect an attack earlier than it can reach the victim. After detection, a mitigation procedure can be called immediately to filter or rate-limit the attack traffic. In reactive techniques, the victim actually encounters a DDoS attack on its services and then a detection & mitigation procedure is called to trace the attacking origin and filter the traffic coming from identified sources.

The above mentioned defense mechanisms can be applied by the control centers that may be located at different points [24] such as:

- Source-end
- Core-end
- Victim-end
- Distributed ends

At source-end defense point, source devices identify malicious packets in outgoing traffic and filter or rate-limit the traffic. It is the best point of defense as minimum damage is done on legitimate traffic. Moreover, another advantage is the minimum amount of traffic to be checked at this point for which fewer resources are required by the detection & mitigation mechanism.

In core-end defense, any core router in the network can independently attempt to identify the malicious traffic and filter or rate-limit the same. However, at this point of defense, the traffic is aggregated i.e., both attack and legitimate packets arrive at the router. In case of a filtering technique, it is a possibility that legitimate packets would also be dropped. On the other hand, it is a better place to rate-limit all the traffic.

In the victim-end defense technique, the victim detects malicious incoming traffic and filter or rate-limit the same. It is a place where a legitimate and attack traffic can clearly be distinguished. However, attack traffic reaching the victim may have severe effects such as denied or degraded services and bandwidth saturation.

Attack detection and mitigation at distributed ends can be the most promising strategy against DDoS attacks [24]. As mentioned before, source-end is a better place for both filtering and rate-limiting the attacks. The core-end is good to rate-limit all kinds of traffic whereas the victim-end can clearly identify the attack traffic in a mix of legitimate and attack packets. Therefore, distribution of methods of detection and mitigation at different ends can be more advantageous. For example, an attack can be identified at the victim-end for which attack signature can be generated. Based on this signature, the victim can send requests to upstream routers to rate-limit such attack traffic. There are various intrusion detection systems available to detect attacks and prevent systems at device or network level such as Host-based Intrusion Detection System (HIDS), Network-based Intrusion Detection System (NIDS), Host-based Intrusion Prevention System (HIPS), Network-based Intrusion Prevention System (NIPS), and Wireless Intrusion Prevention System (WIPS) etc.

4. Traditional Schemes of DDoS Detection & Mitigation

4.1 Some common countermeasures against DDoS attacks

In this section, we study some well known countermeasures against DDoS attacks. They are quite common today in various DDoS defense implementations. Two proactive and two reactive techniques are discussed:

- Ingress/Egress Filtering
- D-WARD
- Hop Count Filtering (HCF)
- SYN Cookies

In ingress/egress filtering technique [31], edge routers are programmed by network administrators to filter packets coming into the network (ingress filtering) and going out (egress filtering). The packet filtering is commonly based on source IP addresses beyond the allocated address space to a network from which a packet is received at router's interface. The source address beyond the allocated space is deemed to be spoofed and hence the packet is discarded. However, the filtering can also be based on some other criteria such as port number, protocol type etc. This method is a source-end, proactive technique capable of protecting against both direct and reflector types of DDoS attacks [24].

The ingress/egress filtering is easy to deploy as ISPs and network administrators have the knowledge of assigned IP address spaces allocated to different customer networks. Therefore, IP spoofing can be prevented. However, it has some limitations such as:

1. Sophisticated attackers can spoof IP addresses from the subnet range. For such an attack, the ingress/egress filtering cannot detect IP address spoofing.
2. Attackers are now more focused towards application layer attacks in which spoofing is not used and actual addresses of zombies are revealed, such as HTTP flood attacks to download images from a website. The ingress/egress filtering cannot identify such attacks.
3. Implementation of filtering policies and rules increases the administrative overhead.

D-WARD [32] refers to a firewall installed at source-end networks. It detects DDoS attacks originated from such networks by collecting traffic statistics of outgoing packets from border routers and comparing them with given models of network traffic based on transport and application protocol specifications. In this way, it can differentiate the legitimate, suspicious and attack traffic. It further rate-limits all traffic for a destination identified to be under attack and prefers the legitimate traffic to pass for other destinations and connections. This method is also a source-end, proactive technique capable of protecting against both direct and reflector DDoS attacks [24].

The D-WARD defense technique is capable of quickly detecting the attacks based on traffic anomalies with reference to given protocol specifications. It can identify heavy floods and accordingly rate-limit the traffic to prevent the victim from severe damage. It is a source-end defense; therefore impact of DDoS attack on a victim is limited. However, it still has a few major limitations such as:

1. Network performance is highly degraded due to the computation of traffic anomalies at edge router.
2. Sufficiently large overhead is imposed on router, for which the router requires high processing power.
3. Since the accuracy of discriminating attack traffic from legitimate traffic at source-end may not be very high, there is a chance of high false positives and false negatives in this technique.

Hop Count Filtering (HCF) [33] is a packet filtering technique at victim-end which observes TTL (Time-To-Live) values of incoming packets. The TTL value of a packet is observed and a guess is made about the same which should be inserted in the packet at sender. The difference between the initial and observed values provides the hop count. In fact, the victim-end server maintains a table of frequently communicating legitimate clients with their source IP addresses and corresponding hop counts. In a DDoS attack scenario, packets with spoofed source addresses are dropped (having no entry in the table or their source addresses do not match with relevant hop counts). For such requests, the victim does not offer its resources such as TCP buffer etc. This method is a victim-end, reactive technique capable of protecting against direct DDoS attacks [24]. However, the technique has also some major shortcomings such as:

1. Legitimate traffic of clients working under a Dynamic Host Configuration Protocol (DHCP) pool suffers from the denial of service.
2. The technique does not explain the availability of services to legitimate users behind Network Address Translation (NAT) since all users behind a NAT usually communicate over the internet with same public IP address. Such legitimate clients also suffer from the denial of service problems.

3. Users with legitimate requests having their IP addresses not in the table at victim-end also suffer from rejections of requests.

The SYN cookies technique [34] is considered to be the most promising defense against SYN flood attacks. In this method, instead of storing Initial Sequence Number (ISN) of SYN packets, the server stores authentication information of SYN/ACK packets. This authentication code is also a sequence number (authentication cookie) generated and stored by the server upon replying with a SYN/ACK packet to the requesting party. In order to calculate this sequence code (the cookie value), server uses hash function (MD5 is normally used) on some packet parameters i.e., source address, source port, destination address, destination port, and Maximum Segment Size (MSS) values. In addition, a counter is used which is a different value approximately after every minute. Further, a secret value is also used which is changed on every boot of the server. The server, upon receiving a packet with ACK flag set (last signal of TCP three-way handshake), verifies the cookie. If the value is found correct, it establishes the connection. This method is a victim-end, reactive technique (filtering method) capable of protecting against SYN flood attacks [24]. However, the method has a few major shortcomings such as:

1. Server exercising SYN cookies method does not offer robustness against SYN flood attacks overwhelming the bandwidth.
2. Server is unable to resend any lost SYN/ACK packet since the relevant information is not available any more.
3. Computational power and resources of the server may exhaust against large SYN flood attacks due to the need of calculating cookie values through hash function against each SYN packet.

4.2 Statistical analysis of network traffic

Researchers have so far made good contributions to make use of statistical features of network traffic for detection of DDoS attacks. They are also used for traceback schemes i.e., identifying the attack source and applying mitigation techniques such as filtering or rate-limiting [5, 24]. The use of Regression Analysis is proposed in [35] and [36] where strength of DDoS attack is estimated and compared with actual strength. The comparison results are promising, indicating that the method is applicable for DDoS strength evaluation in router or a separate unit communicating with the router. Two forms i.e., multiple and polynomial regressions are discussed. The multiple regression method is described as:

$$Y_i = \dot{Y}_i + \varepsilon_i, \tag{1}$$

$$\dot{Y}_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \cdots + \beta_p X_{pi}. \tag{2}$$

Here, 'Y' is the dependent variable. X_1, X_2 upto X_{p} are 'p' independent variables and β_0 is the intercept. β_1, β_2 upto β_{p} are coefficients of 'p' independent variables and ε is the regression residual. 'i' represents a particular flow count for which 'Y' is determined.

Using the above description and applying it on the network traffic monitored at a router, the strength of DDoS attack can be estimated. A fact to be considered here is that the network traffic consists of packets, and packets cannot be input to mathematical models themselves. Instead, their flow mechanics or aggregated volume may be used as inputs to derive mathematical relationships. Hence, a flow-volume based approach is applied here in the process to construct the traffic profile under normal traffic scenario. When total traffic arriving at a router in a designed time window 'Δt' is deviated from the constructed profile based on flow-volume relationship, an attack is detected and its strength is calculated that can be used to estimate the risk and level of compromise against the attack. The 'multiple regression' is applied when more than one independent variables are studied to be linked with one dependent variable or the output. In this case, independent deviations in flow and volume (inputs) of the traffic are studied in specific time intervals and the strength of DDoS attack (output) is calculated. Several more statistical parameters contribute towards changing the traffic flow and volume, hence the overall aggregation in the network. Such parameters are also considered and carefully calibrated to make an effective detection and strength estimation of DDoS attacks.

In polynomial regression, relationship between one independent variable and one dependent variable is expressed as an i_{th} order polynomial. Equation (1) is the same whereas $\dot{Y}_i$ in polynomial regression is described as:

$$\dot{Y}_i = \beta_0 + \beta_1 X + \beta_2 X^2 + \cdots + \beta_n X^n. \tag{3}$$

Again, 'Y' is the dependent variable as expressed in Eq. (1). 'X' is an independent variable appearing upto n_{th} order of the polynomial and β_0 is the intercept on XY-plane. β_1, β_2 upto β_n are coefficients of X in the n_{th} order.

In this DDoS attack estimation technique; a relationship is established between the deviation in sample entropy (input) of the traffic in specific time interval and the strength of DDoS attack (output). This scheme is based on the assumption that the attack traffic is seen different in the network from normal traffic. The deviation in entropy i.e., 'X' is represented as:

$$X = H_c - H_n. \tag{4}$$

Here, H_c is the calculated entropy in a time interval 'Δt' and H_n is normal entropy (entropy value under normal traffic scenario). When deviation is observed in the value of entropy in a specific time interval, it is detected that DDoS attack has occurred and the strength of DDoS attack is thus calculated by applying polynomial regression model [36].

Sample entropy ‘H’ [36, 37] is defined as the degree of concentration of a distribution. It is given as:

$$H = -\sum_{i=1}^{N} p_i \log_2 p_i. \tag{5}$$

In Eq. (5), p_{i} is equal to n_i/S where n_i represents number of bytes arriving in i_{th} flow of traffic in a specified time interval and ‘S’ is the summation of total number of bytes in ‘N’ flows. It is represented as:

$$S = \sum_{i=1}^{N} n_i. \tag{6}$$

In order to detect an attack and estimate its strength, sample entropy is calculated in time intervals ‘Δt’ continuously. When the calculated entropy is different from the normal entropy H_n, an attack is detected and the difference between entropy values i.e., X is used to estimate the attack strength through polynomial regression. The value of sample entropy indicated in Eq. (5) lies in the range from 0 to $\log_2 N$.

In [38], focus is on surveying three different detection techniques of DDoS flood attacks on target networks. These techniques are:

- Activity Profiling
- Change-Point Detection
- Wavelet Analysis

Authors investigate how above mentioned techniques are successful in detecting network-based flood attacks with different test-bed environments as presented in earlier research papers. The aim is to identify better technique to discriminate DDoS attacks from sudden boost in legitimate activities. They find that each detector gives promising results in limited testing scenarios and none is able to completely detect occurrences of DDoS attacks. Therefore, they believe that a combination of various approaches with seasoned network operators can increase a chance of getting improved results.

Activity Profiling is achieved by monitoring header information of packets in a network. It is determined by average packet rate in a flow of packets with similar fields such as address or port fields in IP packets. Change-Point Detection [39] refers to statistical analysis where network traffic is initially filtered for unique fields of IP packets such as address or protocol fields and the resultant is stored as time series, which is a cluster’s activity represented in time domain. In case of a DDoS attack, this time series shows statistical changes which can be monitored for detection of attacks. Wavelet analysis [40, 41] for DDoS attack detection is the observation of network traffic as spectral components. When DDoS attack occurs, anomalous signals are caught; which are separate from background noise. However, the input signal contains both noise and anomalous component at a time.

In Table 3, a summary of testing results is provided which authors of [38] surveyed in [42–47] to detect network-based flood attacks and identify better technique(s) of discriminating DDoS attacks from legitimate activities or flash events.

Table 3. Summary of testing results [38].

Method of Detection	Test Data	Detection Results	Complexity (1 = Lowest; 6 = Highest)
Activity Profiling	Private network data (03 weeks)	12,000 DoS attacks detected on 5,000 distinct victims.	6
	Publicly available data sets (06)	02 out of 02 attacks detected.	3
Change-Point Detection	NS-2 simulation (100 nodes)	UDP and ICMP flood detection with 1–36 seconds of detection delay.	1
	Private Network data sets (03)	70% detection at 33 SYNs per second.	1
Wavelet Analysis	University data (03 weeks)	47% detection rate over 119 time series.	4
	University data with 109 anomalies (03 weeks)	38 out of 39 anomalies detected.	5

4.3 Traceback schemes

Traceback in DDoS defense refers to identifying attack source(s) through some mechanism so that the attack may be blocked or mitigated at origin. However, effectively implementing the traceback to identify DDoS source is difficult due to some well known reasons such as easy spoofing of source IP addresses by the attacker, stateless nature of IP

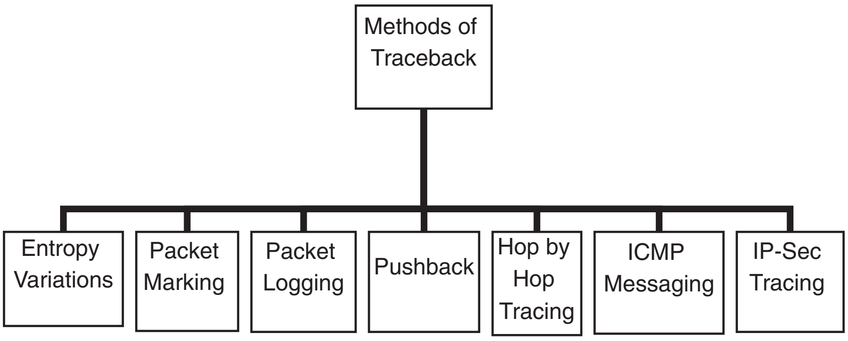


Fig. 4. Traceback schemes — Classification.

routing where complete path is not known (only next hop is usually inserted and updated in router's routing table), link layer spoofing (MAC address spoofing) and intelligent attack techniques provided by modern attack tools [48].

In a research attempt in [49], authors use entropy variations of network traffic 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 present experimental analysis to claim that the method is able to implement accurate traceback in a large-scale DDoS attack scenario (attack with thousands of zombies) within a few seconds.

In [50], authors focus on detection and traceback of low-rate DDoS attacks as they are very much like normal traffic and have more ability to conceal their attack related identities in the aggregate traffic. Two new information metrics are proposed (generalized entropy metric and information distance metric) to detect low-rate DDoS attacks. In said approach, they measure difference between legitimate and attack traffic through their newly proposed information metrics and are able to detect the attack a number of hops earlier than counts mentioned in previously proposed schemes. Their information metrics can increase detection sensitivity of the system and thus the scheme is capable of identifying low-rate DDoS attacks reducing false positive rate effectively. Moreover, this traceback mechanism can efficiently trace all attacks generated at an attacker's own LAN i.e., zombies.

In addition to entropy variation scheme, a few other traditional methods also exist to traceback DDoS sources [48]. They are the schemes of reactive nature. The classification is shown in Fig. 4.

In packet marking schemes, the idea is to trace the path through upstream routers upto the attack sources i.e., zombies. It is a common method employed in traceback implementations but contains some inherent drawbacks. There are two types of packet marking i.e., *probabilistic* and *deterministic* packet marking. In probabilistic packet marking (PPM), each router embeds its IP address probabilistically into the packets travelling from source to destination. The method is based on the assumption that attack packets are much more frequent than legitimate packets. Once the attack is identified, the victim needs sufficient number of packets to reconstruct the path upto the source through embedded information inside the packets. There is no specific field in an IP packet for such markings. Therefore, it utilizes rarely used 16-bit fragment ID in IP packets for the markings [14]. However, this technique has some major drawbacks with it. For example, it is valid only for direct attacks. It cannot detect the true location of attack source in case of reflector attacks as the traced location will be of reflector machines and not of zombies. Moreover, in a well distributed attack with a fairly large number of zombies, the chance of wrong construction of the path increases. It is also a known fact that today, due to large number of zombies, the attackers disclose real IPs of zombie machines (as in application layer attacks) and hence the sources are already revealed. In such cases, packet marking schemes and other traceback methods are useless. The packet marking scheme also places significant computational overhead on intermediate routers when traceback is initiated. It also assumes that victim remains available during the process of traceback (which requires some minutes) as the victim has to send control messages to upstream routers. However, in real scenarios, the bandwidth is saturated due to attack impacts and therefore the control messages are dropped, resulting in wrong construction or misconstruction of attack path. In addition to these drawbacks, packet marking scheme can also be easily paralyzed. That is, if the attacker sends packets with larger than MTU (Maximum Transmission Unit) size of packets, the packet marking is not possible as fragment ID field is used in such cases for packet identification. The routers do not mark packets and according to [51], routers will then be sending the marking information through ICMP packets which is even more complicated and contains some additional drawbacks. For example, due to bandwidth saturation after DDoS attack, several such ICMP packets may be dropped in the network path and the victim would not be able to construct the path. Moreover, some networks do not allow passing ICMP packets through their border routers; therefore the attack tree would not be accurately constructed [24].

In deterministic packet marking (DPM), the router embeds its IP address deterministically into the IP packets. The scheme was introduced to overcome some drawbacks of probabilistic packet marking as it has simple implementation and requires less computational overhead on intermediate routers. However, it has its own limitations. In this scheme, packets are marked with the information of only first ingress edge router i.e., the complete path is not stored as in PPM. Therefore, it requires even more packets to reconstruct the attack path [48]. Moreover, it also has some inherent shortcomings just like PPM scheme discussed above. The scheme is nonetheless more efficient due to deterministic marking of packets as an attempt by the attacker to spoof the mark is overwritten with correct mark by the first router through which the packet traverses.

In packet logging scheme [48] which is also referred as Source Path Isolation Engine (SPIE), the information of each packet is stored or logged at routers through which the packet is passed. The routers under this scheme are termed as Data Generation Agents (DGAs). The stored information of the packet contains constant header fields and first 8 bytes of the payload which are hashed through many hash functions to produce *digests*. These digests are stored by DGAs using *bloom filter*, a space-efficient data structure. This structure is capable of reducing storage requirements by large magnitude. When about 70% of a bloom filter is filled, it is archived for later information processing and a new bloom filter is used. The duration of using a single bloom filter is called *time period*. Hash functions are changed during different time periods and the data necessary to reconstruct the attack path is stored in a table called Transform Lookup Table (TLT). When an attack is detected under packet logging scheme, the central management unit called SPIE Traceback Manager (STM) sends requests to units allocated for region wise management of DGAs known as SPIE Collection and Reduction Agents (SCARs). Each SCAR obtains copies of digests and TLTs from DGAs of its own region for an appropriate time period. It can identify which packets were forwarded by which router and reconstruct the path based on the obtained information. All SCARs report the calculated information to STM. The STM is finally able to reconstruct attack path through the whole network based on the information provided by SCARs. The main drawback of this scheme has been identified as the requirement of enormous computational power and storage capacity due to hash processing and bloom filter usage.

In pushback scheme [52], the router under congestion sends rate-limit request to upstream routers. In fact, it determines from which routes the stream of packets is arrived and devises an attack signature for such traffic. The signature belongs to the aggregate traffic having some common property such as same destination address [24]. A local mechanism called Aggregate Congestion Control (ACC) is responsible to determine congestion on the router and create attack signature. Based on this signature, the router sends requests to adjacent neighbors (upstream routers) to rate-limit such aggregate traffic. Then, the neighbors recursively send requests (propagate pushback) to further upstream routers. In hop by hop tracing scheme, the debugging idea is used where the source of attack traffic is identified on the router closed to victim considering the incoming aggregate traffic flow by adjacent routers. The process is repeated iteratively to upstream routers until the attack source is revealed [48]. In ICMP messaging scheme [53], routers are programmed to send ICMP messages along with the network traffic. Such ICMP packets contain some path information in them such as source address, destination address and authentication parameters etc. A typical router programmed under such scheme normally sends one ICMP messaging packet for every 20,000 packets passing through it i.e., a traceback message is sent with the proportion of 0.005 percent of the network traffic [48].

The IP-Sec [54] refers to per packet authentication in IP networks through shared secret keys. It is based on the idea that per packet authentication provides more secure communication of IP terminals through the network. It is also assumed that per packet authentication is enough to prevent DDoS attacks as bogus packets are identified during the authentication process and accordingly discarded [55]. However, a major shortcoming of IP-Sec is the requirement of high computational power during the process of authentication. In such cases, a large volume of incoming packet streams may shift the DDoS impact from victim to authentication module. Before authentication, the IP-Sec mechanism checks Security Parameter Index (SPI) value which resides in the packet header in addition to authentication information. The SPI value is unique for each flow and only those packets are forwarded to authentication phase which have a valid SPI whereas packets with invalid SPI are discarded. In real cases, attackers are able to discover a session's SPI through intercepting messages in traffic flow pertaining to that particular session on internet, or by observing the impacts as a result of their own actions such as hit & try methods. The successful discovery of SPI leads to the success in denial of service attack [13].

Table 4. Overview of traceback schemes.

Traceback Scheme	Method
Probabilistic Packet Marking	Packets are marked *probabilistically* to traceback attacking source.
Deterministic Packet Marking	Packets are marked *deterministically* to traceback attacking source.
Packet Logging	Packet information is stored (logged) at intermediate devices (routers).
Pushback	Congested routers send *rate-limit* requests to upstream nodes and traceback attacking source(s) through *attack signature*.
Hop by Hop Tracing	Attacking traffic source is identified on the router closed to victim considering the incoming aggregate traffic flow by adjacent routers.
ICMP Messaging	Routers send ICMP messages containing path information, along with the network traffic.
IP-Sec	Per packet authentication is followed through shared secret keys.

4.4 Analysis & future scope

Our survey analysis of traditional schemes to counter DDoS attacks depicts that even at the level of network layer attacks; some enriched schemes have been developed by attackers such as reflector attacks. The detection of such attacks needs huge security investment as well as overhead on intermediate routers and devices. The reduction of such investment cost and overhead is still a major challenge for future research. Moreover, there is a need of strong research cooperation among various ISPs to share protocols and records for an effective defense against DDoS attacks. The source of attack is located through upstream routers which may belong to other ISPs. Therefore, more collaborative efforts would be required to design criteria of blocking traffic for servers which belong to other ISPs.

5. Current Evolutionary Techniques of DDoS Detection & Mitigation

5.1 Application of neural networks in DDoS detection

Artificial Neural Networks (ANNs) are famous learning models for their ability to cope with demands of a changing environment [56]. They are self-learning and self-organizing models which make them a suitable choice for processes which seek advantages like robustness, fault tolerance and parallelism. Moreover, due to self-learning characteristic, they are good enough to identify and resist unknown disturbances in a system. This property of neural networks has been utilized in DDoS attack detections in some research attempts, as they are capable of identifying unknown attack patterns that may exist in DDoS attacks.

In [57], authors use Linear Vector Quantization (LVQ) model of ANN. In this model, input layers accept input vectors called neurons with specified weights which are adjustable according to ANN's self-learning mechanism. The middle layers process the information and pass it on to output layers. In fact, input and middle layers exhibit same kind of functionality in all ANN models. However, the transfer function used for information processing at middle layers is unique for each kind of neural network and the appropriate result is consequently forwarded to output layers. In the case of LVQ model, the information in middle layers is processed in such a way that the winner neuron takes the entire output share and accordingly passes it on to output layers. It is similar to self-organizing maps and applied in techniques of pattern recognition, multi-layer classification and data compression. Under supervised learning, it knows the target output against different forms of various input patterns [57, 58].

Dataset pertaining to a typical DDoS attack flow is simulated in five steps in [57] as shown in Fig. 5. After testing the system with LVQ model, authors use the same dataset with Backpropagation (BP) model of ANN (to be discussed ahead) for comparative study. On the basis of comparison results, they claim that LVQ is more accurate in determining DDoS attacks than BP. They show that LVQ is 99.723% accurate on average against tested dataset whereas the average accuracy of BP is 89.9259% for the same dataset. Accuracies are computed on the basis of percentages of obtained false positives and false negatives against each sample of testing data. There are 10 samples used to test the systems for each of the LVQ and BP models.

In other research attempts found in [59] and [60], authors use BP model of neural networks to estimate the strength of DDoS attack in real time and predict the number of zombies respectively. Backpropagation neural network is a multilayer feed forward network with backpropagation (feedback) of an error function [61]. A simple feed forward

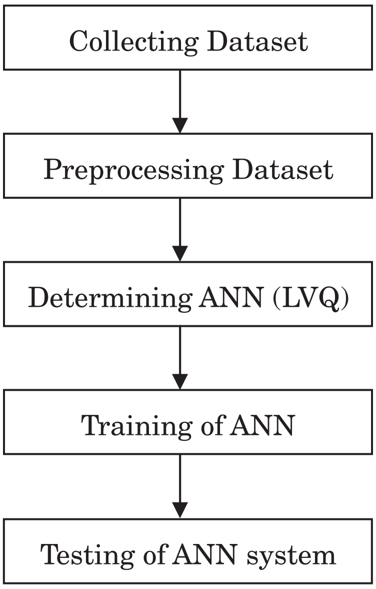


Fig. 5. Implementation phase — Analyzing DDoS with LVQ [57].

neural network has only three layers i.e., input, output and middle layers. Input layer passes on certain weights to middle layer which processes them and sends calculated weights to the output. Each weight is revised according to gradient descent of the error through output layer, back propagated to hidden layer and then to the input layer. Again the information is fed forward and error is fed backward. In this way, weights are adjusted to reduce error and execute learning and training of the neural network. This process is continued until network's output error is brought down to an acceptable level or the preset time of learning is achieved [62].

In [59], authors train the BP neural network with a dataset of variations in traffic entropy as inputs and the corresponding actual DDoS strengths as outputs. 20 different samples in the dataset are used for training with 10 Mbps attack strength as the lowest and 100 Mbps being the highest in the dataset. The entropy variations are calculated as discussed before. Therefore, the scheme is based on an assumption that the attack traffic is seen different in the network from normal traffic. The model is tested with four random inputs of entropy variations for which calculated attack strengths are 20, 50, 70 and 95 Mbps. The BP neural network's output is seen promising with little errors. False positives and false negatives are also very less. Moreover, authors also test the system with variations in network size i.e., number of neurons in processing layer. They use two layer feed forward network with BP algorithm and find that with the increase in network size, errors are further reduced and more accuracy is achieved. However, in real cases, increasing the network size also increases both training time and implementation cost.

In [60], authors train the BP neural network to predict number of zombies behind a DDoS attack. They train the system with a dataset of variations in traffic entropy as inputs and the corresponding actual number of zombies behind DDoS attack as outputs. The dataset is used for training from 10 to 100 zombies with an increment of 5. The attack strength is a constant rate of 25 Mbps. It effectively changes the attack rate per zombie in each data sample ranging from 0.25 to 2.5 Mbps. The model is tested with different random inputs of entropy variations and the BP neural network's output was seen promising with little errors. Moreover, they also test the system with variations in network size and find that with the increase in network size, errors are further reduced and more accuracy is achieved.

In [63], authors test Time Delay Neural Network (TDNN) to produce early warning system against DDoS attacks. TDNN is a type of neural networks in which time delay factor is incorporated or hidden inside the representative signal. In their work, a Demilitarized Zone (DMZ) is created and TDNN is implemented in two-layer pattern. The node activity is monitored by neighboring nodes and attack information is passed on to the expert module for integrated analysis. The layered structure enables the system to take some appropriate actions as a proactive strategy against DDoS attacks such as initiating the deployed Intrusion Prevention System (IPS). Their detection results on deployed architecture show that proposed scheme is able to give 82.7% correct detection rate as compared to 46.3% with general Intrusion Detection System (IDS).

5.2 Botnet fluxing and defense

In recent times, DDoS attackers use sophisticated attack tools to hide necessary traffic information for successful attacks and prevention from any traceback. Many schemes have been deployed to detect botnets behind a DDoS attack based on the attack signature. However, new attack techniques employing botnets (handlers & zombies) are clever enough not to be detected by such schemes as they have unknown signatures or are polymorphic (in many forms) in existence [64]. Two advanced botnet mechanisms surveyed in [2] are:

- Fast Flux (FF)
- Domain Flux (DF)

These two mechanisms behind botnets may not necessarily be used for DDoS. They can also be employed by attackers for other kinds of attacks such as cross-site scripting and e-mail spamming etc. However, as they can be the sources behind DDoS attacks as well, we discuss these techniques and the possible defense against them in this section.

In FF [65], frequent change in a set of IP addresses occurs that belong to a particular domain name. In DF [66], frequent change in a set of domain names occurs that belong to a particular IP address. Behind the fast flux technique, the idea is to compromise a Domain Name Service (DNS) with spoofed IP addresses of short TTLs and from a large IP pool against a single domain name. DNS query is sent to a compromised server by the victim to access a domain name. Due to short TTLs of IP addresses, the victim has to resend the query to DNS server when an assigned IP is expired. In the response of each query, DNS gives a different IP (spoofed address) to the victim which connects it to a fluxing agent (botnet agent). In this way, different agents connect to the victim at different times. Each time, the agent redirects the request to actual server and the response is relayed back to victim. The DNS server in this technique is a compromised machine but not a fluxing agent. The botnet agents are controlled by a Command and Control (C&C) server. The C&C (under attacker's instructions) is responsible to manage the IP pool and the corresponding domain. This process makes the detection of botnet and identification of attack source quite complicated and difficult which is beyond the reach of traditional traceback schemes. However, it has a single point of failure due to one domain name i.e., once the fluxing behind a domain name is identified and it is taken down, the botnet is lost from the attacker's point of view [2].

In DF, DNS is also a part of fluxing where malicious botnet agents (acting as DNS servers) generate domain names through a Domain Generation Algorithm. The domain names are obtained by agents from the C&C server and other servers under the control of a botnet master [2, 67]. The domain names are dynamically generated through the domain generation algorithm and remain consistent at a point of time. The C&C server and agents are seeded with same values to make sure the consistency of domain names. For this purpose, C&C server and agents follow the same algorithm. Agents try to obtain the domain name from a maintained domain list by communicating with C&C server and other servers. The names are obtained repeatedly until a DNS query is fulfilled. In such cases where a current domain name is not accessible or blocked by authorities, botnet agents try to calculate the other one through same algorithm [2]. It is mentioned in [68] that the algorithm in Torpig (a DF based botnet) uses current week and current year values to calculate Top Level Domain (TLD). In case of failure in resolving a domain name, it uses other information (such as current day value) or some hard-coded information from a configuration file.

Some FF and DF detection methods are provided in [69–71]. In [69], authors develop an empirical metric to detect fast fluxing in networks commonly known as Fast-Flux Service Networks (FFSN). Their metric is based on three possible parameters which can be used to identify the difference between normal traffic and FFSN behavior. The parameters are:

- Number of IP domain mappings in all DNS lookups.
- Number of name server records in a single domain lookup.
- Number of autonomous systems in all IP domain pairs.

They develop a metric called flux-score based on above mentioned parameters and use a linear decision function to identify the existence of an FFSN. The results of their two-month long experimental observations show that their metric can differentiate the normal traffic and FFSN behavior with very low false positives.

In [70], authors develop a real time FFSN prediction model to analyze a website's DNS with distributed architecture through a mix of active and passive methods. The model is based on three major components mentioned below:

- Sensors
- Fast Flux Monitor Database
- Fast Flux Monitor

The sensors are further categorized into active and passive sensors. They are used to monitor different IP traffic parameters such as TTL, IP address validity, activity and footprint index etc. The FF monitor database is used to record the parameters obtained by sensors. The analysis of this stored data is a source to establish some analytical knowledge about different parameters of FFSN such as footprints, IP sharing statistics, country of origin and the Internet Service Provider (ISP) etc. The third component is used to classify FFSN through a Bayesian network and calculate a prediction confidence with the help of parameters obtained through sensors. They show that the report generated by this model can assist security analysts in analyzing a website's security with fair accuracies.

In [71], authors use a supervised machine learning method to prevent users from accessing malicious websites. They classify automated URLs based on statistical analysis. The model is designed to make use of lexical features as well as host based properties of malicious domain names. The training of model is achieved through three classification techniques mentioned below:

- Naive Bayes
- Support Vector Machine (SVM)
- Logistic Regression

With the help of these techniques, four different data sets are presented to the model (two malicious and two benign). The analyzed lexical features are entire URL length and dots & words in a domain name etc. The selected host-based features include registrar properties (WHOIS analysis) and properties of domain name such as geographical properties (physical location) etc. The results of this analysis appear to be fair enough to distinguish malicious domains and benign ones with a modest rate of false positives. They find that lexical features along with WHOIS analysis provide rich information whereas the overall analysis is used to extract full classification for accurate detection. They further improve their model in [72] where the same set of lexical and host-based features is used but additionally the model is given a live feed of labeled domain names over the time to make it capable of identifying suspicious URLs with enhanced accuracy.

5.3 Device level defense features in switches and routers

In addition to covering various DDoS detection and mitigation techniques that focus on traffic parameters and anomalies, there exist some authentication based security schemes at device level such as routers and switches to prevent networks and devices from a wide range of attacks. They also provide an effective first line of defense against DDoS attacks. Therefore, we provide a discussion on some of them in this section.

In [11], some schemes are studied that belong to new device level capabilities of routers and switches against various attacks including DDoS. The schemes are:

1. Defense against DDoS attacks using Router's packet forwarding mechanism in more effective way.
2. Defense against SYN flooding attacks using TCP blocking in CISCO Routers.
3. Employing Trusted Platform Module (TPM) hardware incorporated Switches.

Defense against DoS or DDoS using a Cisco router can be accomplished by setting effective packet forwarding mechanism through Unicast RPF (Unicast Reverse Path Forwarding) function which checks CEF (Cisco Express Forwarding) table after receiving a packet. If the route is defined in the table for particular IP scheme of which a packet is received, it forwards the packet. If the route is not found in table, it discards the packet.

Defense against SYN flooding attacks using *TCP blocking* in CISCO Routers can be accomplished by working in an Internetwork Operating System (IOS) environment for which Cisco has introduced the feature after version 11.3. In this feature, a router can be programmed for any of the two available modes i.e., intercept mode and monitoring mode. In intercept mode, router makes TCP connections with clients on behalf of the server. It sends the acknowledgement to client (second signal of three-way handshake) and waits for final acknowledgement from the client. When acknowledgement is received, it shifts the connection transparently to the server. In such cases where final acknowledgement is not received, the connection is closed without transferring the impact to the server. The time-out limits are made very strict to prevent connections from illegitimate users and save router's own resources. In the monitoring mode, router just observes the connection establishment phase between the client and the server. If the final acknowledgement is not received from client within a preset time limit, the connection is closed by the router. The TCP intercept feature in Cisco routers is enabled after creating an extended access list to define source and destination IP addresses used for the intercept in order to prevent internal host or the network [11]. It is analyzed in [73] that Access List (ACL) rules can be defined in routers to prevent networks from potential intrusions. These rules are normally based on the alerts generated by some Intrusion Detection System (IDS) such as Snort (an open source IDS) [74].

Trusted Platform Module (TPM) [75] is the name of a published specification and its implementation to ensure information security in a given system. It is provided by Trusted Computing Group (TCG), an industrial organization developing standards for TPM [76]. It is implemented through a *TPM chip* or *TPM security device*. The idea behind this implementation is to provide a security mechanism to a given system by establishing a chain of trust from root to the entire system through an authentication process. The authentication is based on cryptographic keys stored inside the hardware of TPM chip which is capable of providing a range of passwords through different security algorithms such as random number generator, RSA algorithm and SHA-1 algorithm etc. All cryptographic functions are executed inside the TPM chip. A network switch incorporated with a TPM chip can trigger TPM authentication process upon detecting a DDoS attack through a detection mechanism. The flow of executions in such a case can be as mentioned below:

Flow of TPM authentication process

1. Send request (through switch) to the server for obtaining Public Key of Server (PKS) and Client Certification Authentication Table (CCAT).
2. Receive client's request (through switch) to access the server.
3. Generate a random number 'Ri' and send it to the client.
4. Get 'Ri' signed by authentication server and send it to the switch.
5. Get the Public Key of Client (PKC) from CCAT, decrypt it (say 'Rc') and match with 'Ri.'
6. In case Rc = Ri, mark the client as authenticated and ask the switch whether this client is sending legitimate traffic or it is a DDoS source according to applied detection mechanism. (Switch verifies it by further communication with the server. The switch and server maintain a Client Permission Table in dynamic mode for this purpose).

A virtual connection is first established between server and client under monitoring mode. After specific time with positive client response, the connection is made direct. The Client Permission Table (CPT) is signed by the server. When a client is identified as a malicious user, CPT is updated and access is denied. The reason of generating a random number is to create a different challenge for each client or multiple connection attempts of the same client so that an effective measure can be applied against replay attacks. When the detection mechanism identifies that an earlier detected attack has now been stopped, it notifies the switch to stop authentication and validation process through TPM chip [11].

Table 5. Some common DDoS mitigation methods (Evolutionary techniques).

Basis of Defense	Method
Neural Networks [57, 59, 60, 63]	Magnitude of attack (number of zombies and attack rate) identification through back propagation neural network.
	Magnitude of attack (number of zombies and attack rate) identification through LVQ model of neural network.
	Early warning signals of DDoS attack using Time Delay Neural Network (TDNN). The time delay factor is incorporated or hidden inside the representative signal.
Botnet Fluxing [2, 69–71]	Fast Flux (IP addresses of same domain are frequently changed).
	Domain Flux (Domain names of same IP address are frequently changed).
Defense Mechanism in Switching/ Routing Devices [11]	Packet Forwarding/TCP Blocking in routers.
	TPM hardware chip in switches.

5.4 Defense against application layer DDoS attacks

Application layer DDoS attacks are now very popular in the networking world. They establish complete TCP connections with victim and then start flooding with several *GET* requests to bring down the victim or saturate bandwidth through outbound traffic such as downloading heavy images from a website. In this way, they conceal their identity in a more sophisticated way to trick the detection schemes. In fact, most of the detection and mitigation mechanisms can identify network layer attacks through packet inspection techniques. Therefore, application layer attacks are more successful tools for attackers to harm victims in current times.

Researchers have made some good contributions towards identifying DDoS attacks through the inspection of traffic anomalies that arise due to attack based traffic flow and connection attempts. The most important challenge in this perspective is to differentiate between an attack and a flash crowd. The flash crowd refers to sudden increase in legitimate connections on a server or website occurring at the same time or within a short period [77]. Some of our surveyed attempts in this paper to examine traffic anomalies for detection of application layer DDoS attacks can identify both network layer and application layer attacks, whereas others are focused towards shielding against application layer DDoS attacks only. In this section, we review both types of proposed schemes to provide a better insight of defense against application layer DDoS attacks.

In [78], authors propose an early discovery of DDoS flooding attacks through network-wide monitoring effects. They find that such macroscopic effects reveal a shift in the spatial-temporal patterns of network traffic when a DDoS attack strikes. They test the effects with different modes of attack such as pulsing attack, increasing rate attack and constant rate attack etc. The simulation results show that the shift in spatial-temporal patterns can be captured effectively with a few observation points. Moreover, the time and location of an attack can also be revealed without observing changes at the victim side.

In [79], authors devise a mechanism of parametric methods to detect anomalies in network traffic using aggregate traffic properties without any need of flow separation. The mechanism developed is called bivariate Parametric Detection Mechanism (bPDM). It uses packet size and traffic rate statistics to make a probability ratio test and is able to highly reduce the false positive rate. The metric used to detect network traffic anomalies through their mechanism is bit-rate Signal to Noise Ratio (SNR). They claim that it is an effective metric to detect anomalies and validate their claim by evaluating bPDM with bit-rate SNR in three different scenarios, including a real-time DoS attack. They find that the method is able to detect different attacks in a few seconds. It is also identified that bit-rate SNR is more effective to detect network traffic anomalies as compared to earlier proposed packet SNR in [80]. They evaluate both metrics through bPDM and conclude that bit-rate SNR is better in terms of detection time. They also evaluate when bit-rate SNR is used as detection metric, the detection time decreases with increase in bit-rate SNR value. Moreover, detection time also decreases with increase in the attack rate.

In [81], an attempt is made to distinguish DDoS attacks from flash crowds through hybrid probability metric. Application layer DDoS attacks are similar to flash crowds; however, they still have some differences like traffic rate, access dynamics and source distributions of IP addresses. Using such differences, authors devise an algorithm to distinguish DDoS traffic from flash crowds and evaluate it in simulations as well as on a small experimental test-bed. In their algorithm, they basically work on traffic flows and examine anomalies by setting two grouping thresholds for 'variation' and 'similarity index.' Based on calculated variations of any two distributions and comparing them with given threshold values, they are able to distinguish DDoS attacks from flash crowds within a normal network flow with reduced false positives and false negatives. Hence, the algorithm also increases system's sensitivity. A simple flow of their work is given in Fig. 6. The decision device stops DDoS flow and allows legitimate traffic to pass.

In [82], authors make an attempt to detect application layer DDoS attacks in real Web traffic under the event of flash crowd. They introduce a scheme based on document popularity [83] and devise a multidimensional Access Matrix to obtain spatial-temporal patterns of a flash crowd in normal flow. The matrix is abstracted by component analysis of the flow [84] and document popularity of a certain website is obtained from the server log. The anomaly in network traffic is then detected through a detector based on hidden semi-Markov model, proposed in their previous work [85]. This detector is used to explain dynamics of the matrix and detect DDoS attacks. The authors examine different types of application layer DDoS attacks (constant rate attack, pulsing attack etc.) during a real-time flash crowd event, then obtained data is fitted in their proposed detector. The results show that their model can detect potential application layer DDoS attacks using entropy of document popularity.

In [86], authors propose a mechanism to counter application layer DDoS attacks called *DDoS Shield.* It has two components, one is suspicion assignment mechanism and the other, which they proposed earlier as a foundation of their work [87], is called *DDoS Resilient Scheduler.* They choose some specific properties of attack based sessions such as asymmetric workload and request flooding to identify application layer attacks. Based on these properties, the suspicion assignment mechanism issues a continuous value (not a binary value) to a session according to its variation from the reference behavior (legitimate behavior) and employs DDoS resilient scheduler to determine whether and when a session is to be processed. They use an experimental test-bed with a hosted web application to determine the efficiency of their proposed mechanism. The results describe that DDoS Shield significantly improves victim's performance when an attack is applied with asymmetric workload with an aim to overwhelm server's resources.

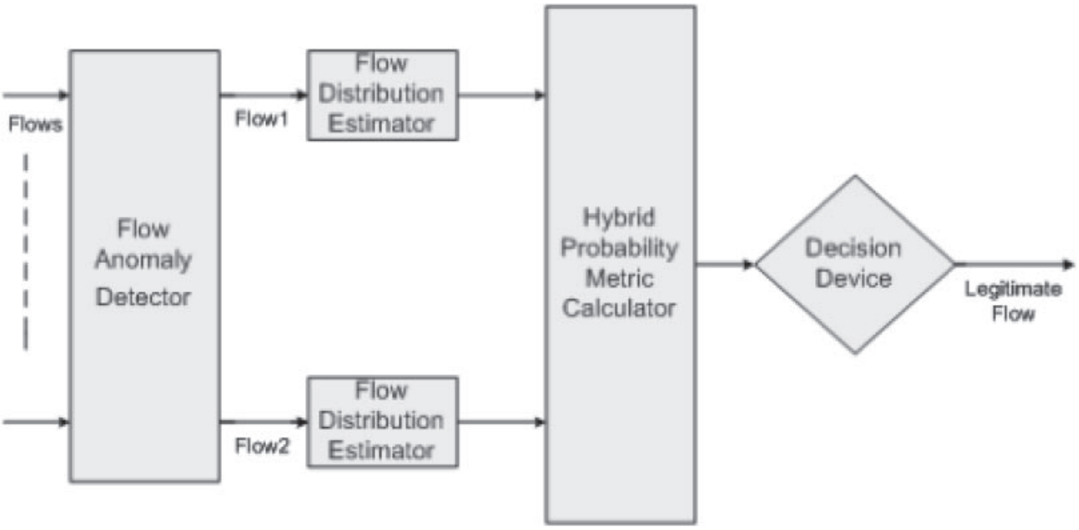
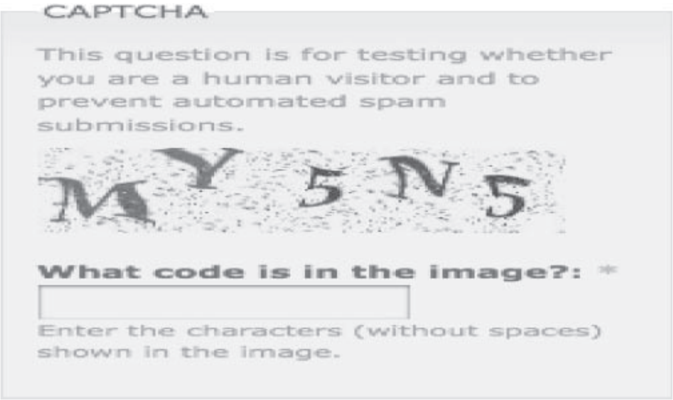

Flows
Flow Anomaly Detector
Flow1
Flow2
Flow Distribution Estimator
Flow Distribution Estimator
Hybrid Probability Metric Calculator
Decision Device
Legitimate Flow

Fig. 6. DDoS detection through hybrid probability metric to differentiate between DDoS attacks and flash crowds [81].



CAPTCHA

This question is for testing whether you are a human visitor and to prevent automated spam submissions.

What code is in the image?: *

Enter the characters (without spaces) shown in the image.

Fig. 7. An example of CAPTCHA test.

Another well known and widely used defense against application layer DDoS attacks is CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) puzzle [88], which is considered to be the most promising technique against application layer DDoS attacks in current times [24]. In this scheme, a challenge-response test is presented to a prospective client requesting to establish a connection with a server. The purpose is to make sure that the response is generated by a human and not an automated machine targeting the server against some kind of attack. It is a good defense against e-mail spam and automated posting to forums and blogs etc. Today, many websites use CAPTCHA at initial login and registration phases to protect servers against application layer DDoS attacks such as HTTP flood etc. In Fig. 7, an example of CAPTCHA test is shown.

CAPTCHA test is an effective technique against HTTP flood and SYN flood attacks. It is a victim-end, filtering technique with threshold-based mechanism [24]. However, it has some limitations as mentioned below:

1. This technique is not effective against bandwidth flooding attacks such as TCP flood and UDP flood. Moreover, it does not counter reflector attacks.
2. This technique prevents any legitimate automated client (if non-human users are required in the system) to establish a connection with the server.
3. CAPTCHA codes are predictable when small pools of fixed images are used.
4. CAPTCHA is annoying for users as they have to solve the test and wait for response before accessing the server. It is not a user-friendly technique and thus legitimate users may be lost for a given server, especially when images are not clear [89].
5. CAPTCHA codes are broken by attackers using image recognition techniques [90]. In such schemes, background noise is removed from CAPTCHA image and then it is segmented to pass through recognition algorithms. In order to improve the defense against such schemes, modern CAPTCHA images include background noise and animations [91] which make an image harder to be recognized by machine based recognition. However, inclusion of such contents often makes images very difficult to be easily read by a human. As a result, legitimate human users become annoyed and use of offered services may be found limited.

Table 6. Some evolutionary mitigation methods against application layer DDoS attacks.

Basis of Defense	Method
Network-wide monitoring [78]	Observing shift in spatial-temporal patterns of network traffic on occurrence of DDoS attack.
Changes in network's aggregate traffic anomalies [79]	Observing packet size and traffic rate parameters through proposed bPDM mechanism to calculate probability ratio test.
Observing changes in network's traffic anomalies through proposed metric [81]	Observing traffic rate, access changes and IP address distribution parameters through proposed hybrid probability metric to analyze proposed grouping thresholds i.e., 'similarity index' and 'variation.'
Observing document popularity in real time web traffic [82]	Observing spatial-temporal patterns of real time web traffic in flash crowd events and analyzing changes on occurrence of DDoS attack through document popularity.
DDoS Shield [86]	Observing some specific properties of attack-based sessions such as *asymmetric workload* and *request flooding* to identify application layer DDoS attacks.
Automated client puzzle [88]	Presenting CAPTHA puzzle images to clients to avoid machine based automated DDoS attacks.

5.5 Some latest research efforts against DDoS attacks

In this section, we review some latest research on countermeasures against DDoS attacks. The efforts discussed here are not published before year 2012.

In a recent research attempt in [92], authors address the issue of group synchronization required by a server while maintaining multiple clients through port-hopping mechanism [93]. In such cases where clock-rate drifts are present among different communicating parties, there are chances that control signals might be lost, keeping the server port open for long time and thus becoming vulnerable to application layer DDoS attacks. They propose an algorithm called BIGWHEEL that offers port-hopping mechanism for servers in multiparty communications without any need of group synchronization. Moreover, an adaptive algorithm called HOPERAA is also proposed to execute port-hopping in the presence of clock-rate drifts. In fact, the need of group synchronization raises scalability issues in port-hopping; whereas research in [92] mentions that port-hopping can be achieved in a scalable way (through proposed algorithm, without the need of group synchronization). The proposed algorithm is implemented on a server and maintains a simple interface with each client. The protocol's port-hopping period is fixed; therefore it creates minimal chances for an adversary to launch application attack on server's port after eavesdropping [94]. However, the method is only tested for fixed clock drifts and hopping frequencies. Further investigations are required for the same parameters in variable mode.

In [95], authors try to address the issues of slow reconstruction of attack path and impact of spoofed packet markings generated by sources of attack in traceback methods of probabilistic packet marking (PPM). They propose a fast traceback scheme called Adaptive Probabilistic Marking (APM) in which TTL field of each packet is set to a uniform value when it enters into the first hop router. This value decreases by one when packet is forwarded by a router in the network. Each intermediate router on the path of packet determines router-level hop number (number of hops packet has traveled) and marks the packet in probabilistic manner with a value that is inversely proportional to this number. The scheme can also be used along with other probabilistic marking schemes. Their NS-2 simulations indicate that APM can reduce the time of effective reconstruction of attack path by more than 20% as compared to existing schemes of probabilistic marking. Moreover, spoofed marks cannot affect the victim or influence the process of traceback. They compare APM with PPM, and two of its improvements called TTL-based packet marking (TPM) and dynamic probabilistic packet marking (DPPM) [96, 97].

In [98], authors use an adaptive method of clients' requests to overcome denial of service attacks and they name it *Adaptive Selective Verification* (ASV). They show that clients can adapt to an attack in effective manner by increasing request rate in consecutive time windows for which the attack rate can be estimated. They use *bandwidth as currency* concept and clients do not have to assume a server state or network congestion. In fact, their aim is to occupy larger proportion of bandwidth by increasing clients' request rate and thus limit the share of attack traffic on bandwidth. In ASV, clients increase their number of requests in consecutive time windows. This increase is exponential and goes upto a threshold level. The server uses sampling method in bounded space to sample incoming packets from input stream of packets. Their NS-2 simulations indicate that ASV is highly adjustable as compared to its non-adaptive counterparts in highly variable attack rate scenarios.

In [99], authors use traffic pattern to propose an analysis which is based on the behavior of traffic flow to differentiate DDoS attacks from legitimate activities. They present a method which can detect DDoS attacks regardless of types of attack packets (DDoS attacks can be generated through different forms of sources such as worms, botnets etc.). The idea behind their work is based on an assumption that DDoS attack traffic has repetitive patterns and specific features which can differentiate it from flash events (legitimate traffic). Their mathematical model is based on *Pearson's correlation coefficient* defines as:

$$\rho_{X,Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y}. \tag{7}$$

It measures dependency of two variables X and Y on each other with expected values μ_X and μ_Y and standard deviations σ_X and σ_Y respectively. Both standard deviations (σ_X and σ_Y) are finite and greater than zero. For the purpose of getting a method relatively independent from types of attack packets, their approach contains two types of algorithms and two methods under each type. One is Methodology Algorithms under which 'Correlation between arrival rate and sequence number' and 'Correlation of self arrival rate' are determined. Second is Threshold Algorithms under which 'Predictable attack rate' and 'Non-predictable attack rate' are discussed. Extensive simulations are executed to optimize the proposed method. Then, experiments with several data sets are performed to show that proposed techniques can discriminate DDoS attacks from legitimate traffic.

In [100], authors present a protocol called LOT, a lightweight tunneling protocol with an aim to prevent network traffic against IP spoofing and flooding attacks. It is deployed at network's communication gateways. Two gateways with LOT implemented between them can detect each other and create the tunnel between them to secure communication. The protocol allows a gateway to discard spoofed IP packets which specify source addresses in other gateway and vice versa. Hence it can protect the communication from many types of attacks including DDoS attacks. For mitigation of DDoS attacks, their work suggests the use of per-flow quotas to identify flooding of packets from different networks. In addition to the efficiency of LOT protocol that it does not pass spoofed packets to destination, it can also filter packets based on filtering rules determined by destination gateway. LOT protocol is also able to detect inter-gateway congestion and can drop the congesting traffic to increase communication efficiency. It is lightweight, easy to detect and a plug & play mechanism implemented at edge gateways (not needed at hosts or core routers). Their experimental evaluations show that LOT has negligible overhead on bandwidth and processing.

In [101], authors achieve DDoS detection with improved time limits through non-asymptotic fuzzy estimators. The estimator is deployed on mean packet inter-arrival times. The problem is distributed into two subsections; one is actual DDoS detection and the other is identification of victim IP addresses. The first part i.e., DDoS detection is achieved using strict real time limits. The second part i.e., identification of victim IP addresses is made through comparatively relaxed constraints. The aim of authors is to identify victim IP addresses in a timely manner to launch added anti-intrusion applications on offended hosts. The proposed method uses packet arrival time as the main statistic of DDoS attack determination. The empirical evaluations show that the proposed method is a way to identify malicious and offended hosts in a timely manner before victims get fully exhausted due to DDoS attack.

A game theoretic approach is used in [102] to offer defense against DoS/DDoS cyber attacks. The DDoS attack is modeled as a one-shot & zero-sum game with non-cooperation. To perform an attack, authors investigate multiple features in terms of cost with malicious traffic distribution and number of attackers. It is demonstrated in analytical terms that a single optimal strategy of defense is available to defender in which upper boundaries are set to attacker payoff depending upon the rational or irrational attackers. The proposed approach is also validated through NS-2 simulations.

In [103], authors propose an approach to detect botnet activities through traffic behavior analysis. They classify traffic behavior using machine learning strategy. It is shown experimentally that botnet activities can possibly be identified in smaller time windows with high accuracy. The experimental data set is also tested to compare the performance of proposed approach with BotHunter [104]. It is mentioned that the proposed method is better to detect botnet activities.

In [105], an anomaly based approach is used to detect low-rate DDoS attacks. Low-rate DDoS attacks are intelligent methods through which attackers can send malicious traffic at sufficiently lower transmission rate to deceive traditional anomaly based DDoS detection techniques. Hence authors propose two new information metrics, *generalized entropy metric* and *information distance metric*. Through such metrics, they measure difference between legitimate traffic and attacking traffic to detect DDoS attacks. The proposed metrics are further combined with the algorithm of IP traceback to produce a collaborative technique of defense against low-rate DDoS attacks.

In [106], authors propose a mechanism to detect application layer DDoS attacks in heavy backbone traffic. It is a fact that many traditional application layer DDoS mitigation techniques are not much capable to handle attacks within heavy backbone traffic such as core web server applications. Hence special considerations are required for devising methods to detect DDoS attacks on such type of applications. The proposed method examines traffic entropy and characterizes the traffic through a set of models. Through such models, it builds a Real-time Frequency Vector (RFV) to detect application layer DDoS attacks. The detection principles are further integrated with a modular architecture of defense to produce a collaborative filter against application layer DDoS attacks at backbones. The experimental analysis shows that the filter is capable to detect and stop attack based requests while allowing legitimate requests to pass at backbones. In addition to the mentioned techniques, some good work has also been published in [107–109] (see Table 7).

5.6 Analysis & future scope

Our analysis of current evolutionary techniques to counter DDoS attacks indicates that application layer attacks are now getting more popular in attackers due to their unique properties of legitimate-like behavior. It is a fact that network layer attacks which contain packet manipulations are now relatively easier to detect with modern detection and mitigation tools. However, application layer DDoS defense needs more research for development of highly

Table 7. Some recent research efforts on DDoS mitigation.

Scheme	Method
BIGWHEEL and HOPERAA algorithms [92]	Avoiding group synchronization between communicating servers for port-hopping mechanism through proposed BIGWHEEL algorithm and providing adaptive port-hopping to servers where clock-rate drifts are present through proposed HOPERAA algorithm.
Adaptive Probabilistic Marking (APM) [95]	Observing TTL fields of packets to initiate proposed traceback scheme and reconstruction of attacking path on occurrence of DDoS attack.
Adaptive Selective Verification (ASV) [98]	Increasing legitimate request rate (in adaptive manner upto a threshold level) in consecutive time windows by legitimate clients on occurrence of DDoS attack.
Traffic Pattern Analysis [99]	Observing changes in traffic flow and analyzing patterns using Pearson's correlation coefficient to calculate standard deviations of observed parameters. The analysis to distinguish DDoS attacks from legitimate activities is made through proposed algorithms.
LOT Defense [100]	Establishing tunnel between two communicating gateways through proposed lightweight protocol to prevent traffic against IP spoofing and flooding attacks.
Non-asymptotic Fuzzy Estimator [101]	Deploying proposed fuzzy estimator on mean packet inter-arrival times to detect DDoS attacks and identify victims' IP addresses.
Game Theoretic Approach [102]	Observing malicious traffic distribution & number of attackers, and modeling DoS/DDoS cyber attack as one-shot & zero-sum game with non-cooperation.
Traffic Behavior Analysis [103]	Detecting botnet activities by classifying the traffic behavior using machine learning strategy.
Anomaly based Approach through New Information Metrics [105]	Detecting low-rate DDoS attacks through two (02) new information metrics: *generalized entropy metric* and *information distance metric*.
Real-time Frequency Vector (RFV) [106]	Detecting application layer DDoS attacks in heavy backbone traffic by examining traffic entropy and characterizing the traffic through a set of models. The models are used to construct the proposed Real-time Frequency Vector (RFV) to detect application layer DDoS attacks at backbones.
Application Layer DDoS Attack Blockage [107]	Detecting and blocking DDoS attack at application layer through proposed comprising of a novel set of algorithms. The framework is capable to block application layer DDoS attack while passing legitimate requests including the flash traffic.
Internet Threat Monitor (ITM) and Honeypots [108]	Observing DDoS attacks on Internet Threat Monitor (ITM), which is used to detect, measure and track attacks including DDoS attacks. DDoS attacks are made on ITM using Botnet and those attacks are modeled through proposed information-theoretic framework. Using this model, attacks are generalized and an effective detection is proposed using honeypots.
TCP SYN Flooding Attack Detection through New Scheme [109]	Detecting TCP SYN flooding attack through newly proposed router based scheme using Counting Bloom Filter algorithm and CUSUM algorithm.

effective defense tools. Although some papers have been presented on the topic which we reviewed [78, 79, 81, 82, 85–87, 92, 95, 98–103, 105–109], but their practical implementation has not been checked at a widespread level. As mentioned before, CAPTCHA is considered to be the most promising technique against application layer DDoS attacks but it also has some major shortcomings which we pointed out. Therefore, application layer DDoS detection and mitigation would require more research with the challenge of distinguishing attack events from flash crowds.

6. DDoS Detection & Mitigation in Wireless Networks

In this section of the paper, we focus on some DDoS attacks and their detection mechanisms in common forms of wireless networks, such as sensor networks, mobile ad hoc networks and wireless local area networks.

6.1 Defense against DDoS attacks in WSNs and MANETs

Sensors are promising devices and their wireless networks are highly used for facilitating real-time data on large scale implementations with complex environments. They are employed for industrial automation, various military applications, home land security and medical examinations etc. They have limited processing capabilities with small memories. The benefit of their deployment is to get increased capabilities with lesser cost. However, as we find in almost every wireless network, their major limitation is the security of wireless channel.

Any attacker in wireless network's radio range can jam the network, eavesdrop on the traffic or send malicious data to network nodes. This leads to a major vulnerability to DDoS attacks in such networks. In this survey, we find some good studies on various types of denial of service attacks in wireless sensor networks (WSN) along with relevant defenses [110, 111].

Authors in [111] discuss many forms of attacks already mentioned in [110]; however, some more discussions are also included for additional information on the topic. In Table 8, attacks and defenses discussed in [110] & [111] are consolidated.

Table 8. WSN layers and DoS attacks/defense [110, 111].

Network Layer	Attacks	Defenses
Physical	Jamming	Spread spectrum, lower duty cycle, detect & sleep, region mapping, mode change.
	Node tempering	Hiding of nodes, tamper proofing.
Link/MAC (Medium Access Control)	Collision	Error-correcting code.
	Exhaustion	Rate limitation.
	Unfairness	Small frames.
	Interrogation	Authentication and antireplay protection.
	Denial of sleep	Authentication and antireplay protection, detect & sleep, broadcast attack protection.
Network/Routing	Neglect and greed	Redundancy, probing.
	Misdirection	Authorization, monitoring, egress filtering.
	Black hole	Authorization, monitoring, redundancy.
	Homing	Header encryption, dummy packets.
	Spoofing/Clustering messages	Authentication and antireplay protection, secure cluster formation.
	Hello floods	Pairwise authentication, geographic routing.
Transport	Flooding/SYN flooding	Client puzzles, SYN cookies.
	Desynchronization	Packet authentication.
Application	Path-based DoS	Authentication and antireplay protection.
	Deluge/Reprogramming attack	Authentication and antireplay protection, authentication streams.
	Overwhelming sensors	Sensor tuning, data aggregation.

As shown in Table 8, jamming attack in wireless networks is a physical layer attack. It is considered as a major problem in wireless networks because of shared medium. It is like choking the communication channel in a wired network for bandwidth congestion. In WSNs or other wireless networks, a malicious node can continuously transmit radio signals in shared wireless channel to block legitimate access of network nodes. This is known to be a jamming attack and adversaries are termed as jammers. There are many jamming techniques that vary from simple methods of jamming such as a continuous transmission of interference signals to block communication, to advanced methods which exploit protocol vulnerabilities of victim(s). In [112], a comprehensive survey is presented to cover various jamming attacks and methods of defense. In Table 9, jamming models discussed in [112] are mentioned with relevant information and specified references.

Mobile Ad hoc Networks (MANET) have no pre-determined infrastructure or topology. The mobility of nodes is very high and they form the shape of a MANET dynamically. There is no central control and the communication is done in ad hoc mode i.e., all nodes also behave as routers where they assume the responsibility of forwarding packets generated by other sources and destined for other nodes. These unique characteristics of MANET have made it vulnerable to many security attacks including DDoS attacks [119]. As in other wireless networks, nodes in MANET share the same wireless channel and hence they can be compromised by attackers through various DoS attacks mentioned in Table 8.

Many research papers have been published on efforts to secure MANETs against different attacks. In [120], authors propose a distributed Intrusion Detection System (IDS) for mobile nodes in MANET. In their scheme, an IDS agent functions on each mobile node in the network. Authors in [121] extend this work and focus on securing AODV routing protocol of MANETs through their IDS scheme. In [122], authors propose Support Vector Machine (SVM) based IDS. In this system, SVM classifies the traffic in normal and abnormal classes to detect attacks. It is capable to detect events that are previously unseen, thus it is a powerful technique of intrusion detection. In [123], authors propose a local IDS scheme. In this system, mobile agents are configured to collect data from Management Information Base (MIB) through Simple Network Management Protocol (SNMP). A mobile agent running on SNMP does not incur much cost to collect information from MIB in this scheme. All these papers have mainly focused on protecting MANET nodes from jamming attacks.

In [124], authors propose an IDS engine that combines neural networks and a protection scheme based on watermarking techniques. Each node in MANET creates its own Self Organizing Map (SOM) and a global map of neighbors. Thus, a node can select secure path(s) to forward packets and avoid compromised neighboring nodes. The proposed scheme is also evaluated for different mobility patterns and traffic conditions to show its efficiency. In [125], authors give an idea of *protection nodes* to protect high priority and important MANET nodes from DDoS attacks. Nodes with less priority are designated as protection nodes. When DDoS attack on some important node ‘N’ is detected, the traffic is redirected to protection node(s) whereas ‘N’ works normally. It is assumed that adversary would stop attacking after a length of time when goal would not be achieved. Their NS-2 simulations show that it is an effective scheme in limited scenarios with small overhead when tested on AODV routing protocol. However, a limitation exists when attacker floods the network overall (through broadcast etc.) to congest the entire MANET. In [126], authors propose a lightweight and distributed IDS (based on neural network) against security attacks in MANET.

Table 9. Jamming models in wireless networks [112].

Jamming Model	Complexity (1=Lowest; 4=Highest)	DoS Level (1=Lowest; 4=Highest)	Anti-Jamming Resistance (1=Lowest; 4=Highest)
Constant [113]	1	4	3
Deceptive [113]	1	4	3
Random [113]	1	2	3
Reactive [113]	4	4	1
Packet Corruption [114, 115]	2	4	1
Narrow-band [116]	4	4	2
DIFS Waiting [114, 115]	3	4	1
Identity Attacks [117]	3	4	4
Layered Attacks [118]	4	4	3

Since MANET nodes work with limited resources, SOM architectures may not fit as far as computational power and energy requirements are concerned. Hence, they come up with a lightweight architecture called Distributed Hierarchical Graph Neuron (DHGN) that works with input patterns organized in hierarchical graphs. It works according to principles of SOM but has lower accuracy than SOM. However, their experiments show that this scheme is still comparable to SOM when limited resource utilizations of MANET nodes are considered. In a recent research effort in [127], authors propose a cluster analysis based hybrid defense against DDoS attacks in MANET. In this work, they investigate packets and employ XOR markings on attack traffic. It helps in identifying source node from which malicious data is sent. Such a node is isolated from MANET and its further communication is blocked. They analyze the proposed scheme on a specific data set for packet acceptance rate along with attack detection. The experimental results show that it is an efficient scheme to detect and mitigate DDoS attacks in MANETs.

6.2 DoS attacks in 802.11 based wireless networks

Most of the current implementations of wireless networks use IEEE 802.11 MAC (Medium Access Control) layer standard. It is a data link layer standard primarily developed for Wireless Local Area Networks (WLAN). It has been evaluated in research that MAC layer in wireless networks is vulnerable to many forms of attacks including different variations of DDoS attack. Some common forms are transmission of malicious packets, active attacks (e.g., replay attacks), passive attacks (e.g., eavesdropping), session hijacking etc. In WLAN, nodes communicate via central control devices known as Access Points (APs). A form of denial of service attack in WLAN scenario is shown in Fig. 8. It can be seen that an attacker is able to snoop communications between other stations and their respective APs.

Authors in [128] focus on the problem of unauthorized access gained by attackers due to weaknesses of existing security protocols such as WEP. They propose architecture for mobile nodes against DoS attacks. Different scenarios are discussed, taking into account the mobility of attackers and victims. In [129], authors evaluate some existing wireless protocols against attacks on WLAN. They also develop a formal method to model semantic DoS attacks in wireless networks and use it to find protocol vulnerabilities. They also find new deadlock vulnerability in 802.11 and validate it experimentally. In [130], authors discuss in detail the existing denial of service attacks and countermeasures in 802.11 based wireless networks. In Table 10, a few common attacks and defenses discussed in [130] are mentioned.

6.3 Analysis & future scope

While surveying DDoS attack and defense techniques in different wireless forms of networks, we analyze that a repetitive cycle of attack and defense goes on with the inclusion of more automated, enhanced and sophisticated tools like we observe in traditional World Wide Web. In addition to the World Wide Web, DDoS attacks are also common in specific protocols, services and infrastructure such as WSN, WLAN and MANET. Some research studies also point out that multimedia communications on wireless platforms also face DDoS challenges like wired networks; such as SIP (Session Initiation Protocol) flood attacks in VoIP (Voice over IP) [26, 131]. Therefore, mitigating DDoS problem against these specific services and networks would also require significant research and implementation attempts in future.

7. An Overview of DDoS Detection & Mitigation in Future Generation Networking: CCN and Cloud Computing

In this section, we focus on some DDoS attacks and their detection mechanisms in Content Centric Networking (CCN) and Cloud Computing environments which are considered as forms of the future internet.

7.1 DDoS protection in CCN

CCN [132] is considered as a form of future internet. In TCP/IP model, the information is obtained from hosts

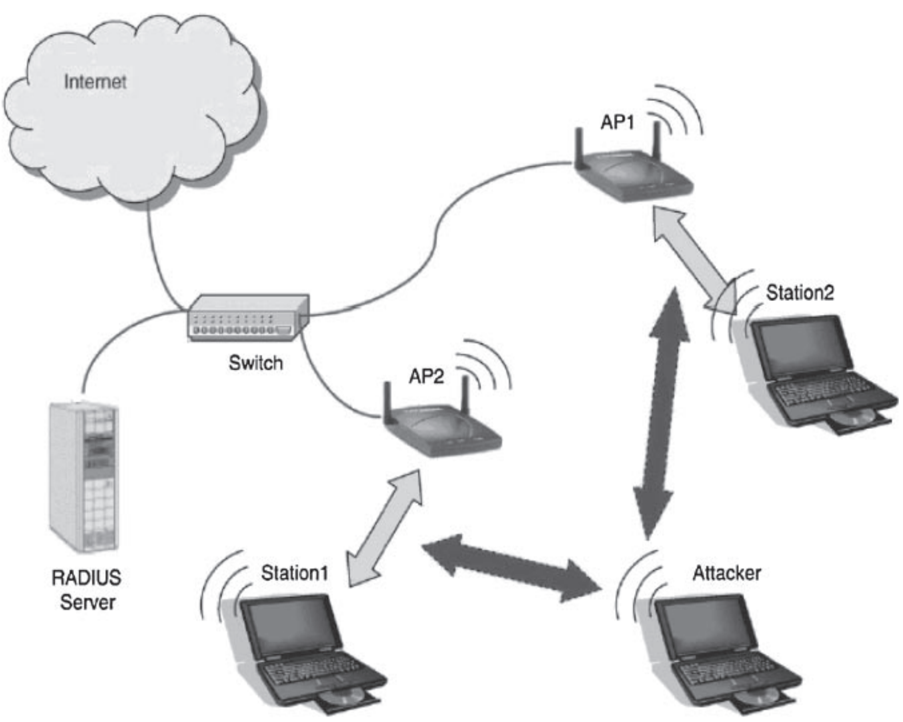


Fig. 8. A form of DoS attack in WLAN [130].

Table 10. DoS attacks/defense in 802.11 networks [130].

Attacks	Defenses
ICMP ping flood	Filtering, IDS.
ARP poisoning	Cryptographic protection, Filtering, IDS.
Authentication/Association request flood	Cryptographic protection, protocol repair, client puzzle, IDS, decresing retry limit, signal strength info identification, RF fingerprint identification.
Probe request flood	Cryptographic protection, client puzzle, IDS, decresing retry limit, signal strength info identification, RF fingerprint identification.
Deauthentication/Deassociation	Cryptographic protection, MAC address spoof detection, IDS, delaying effects, signal strength info identification, RF fingerprint identification.
Monopolizing attack	Rapid frequency hopping, multi-hop forwarding, spatial retreat.
Reactive attack	Rapid frequency hopping, multi-hop forwarding, spatial retreat.
Preamble attack	Rapid frequency hopping, multi-hop forwarding, spatial retreat.
Symbol attack	Rapid frequency hopping, multi-hop forwarding, spatial retreat, forward error correction.

through IP addresses (net, subnet etc.) where the required content (data) is available. On the other hand, CCN addresses ‘content’ (not host) in packets through content ‘names’ (not location). Hence, it is discussed under umbrella of networking named content. When CCN is evaluated against TCP/IP stack, it is found that IP addresses in layer 3 are replaced with ‘content chunk.’ In layer 4, transport protocol represents an agreement between consumer and publisher (or producer). Two packet types in CCN are *Interest* and *Data* packets. A consumer node can ask for some content through Interest packet by broadcasting it over all available links. A producer node hearing the Interest packet can reply with Data packet of required content chunk if it satisfies consumer’s interest. It is determined by ‘Content Name’ fields in Interest and Data packets. A Data packet satisfies consumer’s interest packet if content name in Interest packet is a prefix of content name in Data packet. At the same time, a balance is maintained between interest and data during communication sessions.

There are many built-in security features in communication strategy of CCN. In normal IP communications of current times, security is a concern to be dealt at levels of hosts (mostly servers) and communication channels/links. On the other hand, CCN has a security in the content itself. The content is digitally signed by the publisher, and consumer determines whether received content is trustworthy and safe. Moreover, the encryption of content or content names is also an additional layer of security which usually remains transparent to the network. In normal IP communications, IP address spoofing is an easy task for attacker. On the other hand, hiding content is much difficult for him since any node on network which can satisfy an interest through available links with the content or its copy is able to reply the consumer.

The interest propagation technique in CCN also makes it capable to fight against DDoS attacks. An adversary would find it more difficult to launch DDoS attack in CCN environment than TCP/IP. The balance between interest and data makes flooding attacks much difficult for attacker. He can only try flooding the network through interest packets. If zombies behind an attack generate large amount of interest packets with same content name, an aggregation of interest is observed and only one pending interest would ever be forwarded on available link, hence the flooding will not occur. If zombies use different content names with same targeted prefix, the cached copies of data satisfying interest would be available on various places in the network. As soon as an interest packet would find a content copy that satisfies it at the nearest possible location of attacking source, the interest would be cleared. Hence, it would still make flooding very difficult to occur in CCN environments. More comprehensive discussion on routing, security and flow control features of CCN is available in [133]. Researchers are also making efforts to explore new CCN dimensions by evaluating routing algorithms, data modeling, memory management and developing monitoring tools [134–138].

7.2 DDoS defense in cloud computing environment

Cloud computing [139] is also a form of resource sharing towards future internet implementation. An entity with limited resources can be the part of a larger cloud to increase its capabilities and processing capacity without investing on new infrastructure or software licensing. Cloud computing is an increasing trend in last few years because of its effective benefits. There are many cloud infrastructures that exist [140–142]. However, as more entities have become the part of clouds, concerns have been raised about the safety and security of new inclusions. A comprehensive survey on security issues in service delivery paradigm of cloud computing has been presented in [143].

Security is a major concern of cloud computing [144]. A new entity may become the part of a cloud with an aim to degrade its services. Therefore, many security parameters have been evaluated and tested on various cloud scenarios. DDoS attacks are also a threat to cloud computing. Traditional methods of detecting and mitigating DDoS may not highly be successful in clouds due to their relatively low efficiency and large storage etc. Many research efforts have been presented for securing cloud environments. A confidence based filtering (CBF) method for packets in clouds is proposed in [145]. Packets are first collected during non-attack period to generate profile. During attack period, packets are again collected to calculate the score of a packet with reference to the nominal profile of non-attack period to determine whether the packet should be discarded. In [146], authors present DDoS defense as a network service in which they use cloud infrastructure called *CLAD* to protect Web servers against denial of service attacks. In fact, CLAD protection is a form of cloud which faces attack traffic and works as an application running web proxy or a virtual machine. The protected server is behind the cloud and only accepts requests from CLAD. Authors of [147] propose a combined approach with Service Oriented Architecture (SOA) based traceback technique called SBTA (SOA based traceback approach) and Cloud-filter to mitigate DDoS attacks in cloud computing. The cloud-filter filters attack packets whereas SBTA provides path reconstruction to determine attacking source(s). The use of neural networks have also been evaluated upto some extent to detect DDoS attack incidents in cloud environment. Authors of [148] test a cloud traceback model using back propagation neural network called Cloud Protector, which is trained to detect and filter DDoS attack traffic in cloud environment. More studies on DDoS related issues in cloud computing environments can be found in [149–153].

7.3 Analysis & future scope

A survey on DDoS attack and defense techniques in cloud computing environments reveals that with inclusions of more entities in clouds, they not only get bigger but face greater security challenges including DDoS. Some research studies also depict that research on security aspects of future internet and next generation mobile computing is a huge subject in itself [154, 155]. DDoS is now considered to be a scalability problem in networks [156]. The current architecture of World Wide Web is not fundamentally scalable, thus susceptible to DDoS attacks. A network which is fundamentally and dynamically scalable in all aspects may not have DDoS problems associated with it. Normally, communications with the world are made through networks built upon fundamental internet architecture which is vulnerable to DDoS attacks. Therefore, such networks are also the part of ongoing offense and defense of DDoS problem [157]. On the other hand, networks created upon a separate and clean infrastructure are immune to DDoS. However, such networks are not found in existence due to the need of heavy investments and resources behind them. The creation of such networks and increasing the scalability of underlying internet architecture to improve defense against DDoS attacks is a huge challenge for future research.

8. Concluding Remarks

We present in this paper, a review on Distributed Denial of Service attack and defense techniques with an emphasis on current DDoS defense schemes based on entropy variations and other traffic anomalies, neural networks and application layer DDoS defense. Further, DDoS defense strategies in wireless networks, CCN and cloud computing environments are also covered. In addition, some traditional techniques such as traceback and packet filtering are also discussed in the paper. The study reveals that new attack techniques have been introduced with sophisticated DDoS attack tools such as botnet fluxing, GET floods and reflector attacks. With such enriched attacks, the defense is even more challenging especially in case of application layer DDoS attacks where attack packets are in the form of legitimate-like traffic mimicking in the events of flash crowds. The major challenge identified in research is to distinguish application layer DDoS attacks from flash crowds with an acceptable rate of false positives and false negatives. Although some good research attempts have been presented for defense against application layer DDoS attacks, their practical implementation across a wide range of networks has not been verified i.e., only test-bed cases in limited scope are evaluated and discussed. Some common defense techniques mentioned in this paper have also been reviewed in a critical manner to identify their inherent shortcomings. Even the most promising technique against application layer DDoS attacks in current times i.e., CAPTCHA has also some major drawbacks. Therefore, the future research in this domain is even more challenging. DDoS is now considered to be a scalability problem for networks built upon the current internet architecture. It may not be a problem of the same magnitude for fully scalable networks designed upon separate and clean infrastructure.

REFERENCES

[1] Mitrokotsa, A., and Douligeris, C., Denial-of-Service Attacks. Network Security: Current Status and Future Directions, Wiley Online Library, Chapter 8, 117–134 (2006).

[2] Zhang, L., Yu, S., Wu, D., and Watters, P., A Survey on Latest Botnet Attack and Defense, Proceedings of IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), 53–60 (2011).

[3] Mishra, A., Gupta, B. B., and Joshi, R. C., A Comparative Study of Distributed Denial of Service Attacks, Intrusion Tolerance and Mitigation Techniques, Proceedings of IEEE European Intelligence and Security Informatics Conference (EISIC), 286–289 (2011).

[4] Ghazali, K. W. M., and Hassan, R., "Flooding distributed denial of service attacks — A review," *J. Comput. Sci.*, **7**: 1218–1223 (2011).

[5] Beitollahi, H., and Deconinck, G., Denial of Service Attacks: A Tutorial, Electrical Engineering Department (ESAT), University of Leuven, Technical Report: 08-2011-0115 (2011).

[6] Information WeekSecurity: ⟨http://www.informationweek.com⟩ (2012).

[7] Business Insider: ⟨http://articles.businessinsider.com⟩ (2011).

[8] SecureList: ⟨http://www.securelist.com⟩ (2012).

[9] Prolexic Technologies: "Prolexic Attack Report Q1 2013," ⟨http://www.prolexic.com⟩ (2013).

[10] Conti, M., Chong, S., Fdida, S., Jia, W., Karl, H., Lin, Y. D., Mahonen, P., Maier, M., Molva, R., Uhlig, S., and Zukerman, M., "Research challenges towards the future internet," *Comput. Commun.*, **34**: 2115–2134 (2011).

[11] Chao-yang, Z., DoS Attack Analysis and Study of New Measures to Prevent, Proceedings of IEEE International Conference On Intelligence Science and Information Engineering (ISIE), 426–429 (2011).

[12] Ahlawat, N., and Sharma, C., "Classification and prevention of distributed denial of service attacks," *Int. J. Adv. Eng. Sci. Technol.*, **3**: 52–60 (2011).

[13] Badishi, G., Herzberg, A., Keidar, I., Romanov, O., and Yachin, A., An Empirical Study of Denial of Service Mitigation Techniques. IEEE Symposium on Reliable Distributed Systems (SRDS), 115–124 (2008).

[14] Subhashini, K., and Subbalakshmi, G., "Tracing sources of DDoS attacks in IP networks using machine learning automatic defence system," *Int. J. Electron. Commun. Comput. Eng.*, **3**: 164–169 (2012).

[15] Lin, S. C., and Tseng, S. S., "Constructing detection knowledge for DDoS intrusion tolerance," *Expert Syst. Appl.*, **27**: 379–390 (2004).

[16] Wang, Y., Lin, C., Li, Q. L., and Fang, Y., "A queuing analysis for the denial of service (DoS) attacks in computer networks," *Comput. Network*, **51**: 3564–3573 (2007).

[17] Douligeris, C., and Mitrokotsa, A., "DDoS attacks and defense mechanisms: Classification and state-of-the-art," *Comput. Network*, **44**: 643–666 (2004).

[18] Aissani, A., and Achour, M. Y., Evaluation of the Severity of DoS Attacks on Computer Networks, Proceedings of IARIA 2nd International Conference on Performance, Safety and Robustness in Complex Systems and Applications (PESARO), 8–13 (2012).

[19] Mirkovic, J., and Reiher, P., "A taxonomy of DDoS attack and DDoS defense mechanisms," *ACM SIGCOMM Comput. Commun. Rev.*, **34**: 39–53 (2004).

[20] Abliz, M., and Znati, T., New Approach to Mitigating Distributed Service Flooding Attacks, Proceedings of IARIA 7th International Conference on Systems (ICONS), 13–19 (2012).

[21] Sen, J., Chowdhury, P. R., and Sengupta, I., A Mechanism for Detection and Prevention of Distributed Denial of Service Attacks, Distributed Computing and Networking, Lecture Notes in Computer Science (Springer-Verlag), **4308**: 139–144 (2006).

[22] Kang, S. H., Park, K. Y., Yoo, S. G., and Kim, J., "DDoS avoidance strategy for service availability," *Cluster Comput.*, **16**: 241–248 (2013).

[23] Lee, Y., and Lee, Y., Detecting DDoS Attacks with Hadoop, ACM CoNEXT Student Workshop (2011).

[24] Beitollahi, H., and Deconinck, G., "Analyzing well-known countermeasures against distributed denial of service attacks," *Comput. Commun.*, **35**: 1312–1332 (2012).

[25] Tariq, U., Hong, M., and Lhee, K., A Comprehensive Categorization of DDoS Attack and DDoS Defense Techniques, Advanced Data Mining and Applications, Lecture Notes in Computer Science (Springer-Verlag), **4093**: 1025–1036 (2006).

[26] Peng, T., Leckie, C., and Ramamohanarao, K., "Survey of network-based defense mechanisms countering the DoS and DDoS problems," *ACM Comput. Surv.*, **39**: 1–42 (2007).

[27] Ying, Z., "Distributed denial of service attack principles and defense mechanisms," *Advances in Natural Science (CS*

Canada), **4**: 15–17 (2011).

[28] Loukas, G., and Oke, G., “Protection against denial of service attacks: A survey,” *Comput. J.*, **53**: 1020–1037 (2010).

[29] Kumar, A. R., Selvakumar, P., and Selvakumar, S., Distributed Denial-of-Service (DDoS) Threat in Collaborative Environment — A Survey on DDoS Attack Tools and Traceback Mechanisms, Proceedings of IEEE International Advance Computing Conference (IACC), 1275–1280 (2009).

[30] Aamir, M., and Arif, M., “Study and performance evaluation on recent DDoS trends of attack & defense,” *Int. J. Inf. Technol. Comput. Sci., MECS Publisher*, **5**: 54–65 (2013).

[31] Ferguson, P., and Senie, D., Network Ingress Filtering: Defeating Denial of Service Attacks which Employ IP Source Address Spoofing, RFC 2827 (2000).

[32] Mirkovic, J., and Reiher, P., “D-WARD: A source-end defense against flooding denial-of-service attacks,” *IEEE T Depend Secure*, **2**: 216–232 (2005).

[33] Wang, H., Jin, C., and Shin, K. G., “Defense against spoofed IP traffic using hop-count filtering,” *IEEE ACM T Network*, **15**: 40–53 (2007).

[34] Eddy, W., TCP SYN Flooding Attacks and Common Mitigations, RFC 4987 (2007).

[35] Gupta, B. B., Agrawal, P. K., Joshi, R. C., and Misra, M., Estimating Strength of a DDoS Attack Using Multiple Regression Analysis, Communications in Computer and Information Science (Springer), **133**: 280–289 (2011).

[36] Gupta B. B., Agrawal, P. K., Mishra, M., and Pattanshetti, M. K., On Estimating Strength of a DDoS Attack Using Polynomial Regression Model, Communications in Computer and Information Science (Springer), **193**: 244–249 (2011).

[37] Shannon, C. E., “A mathematical theory of communication,” *ACM Mob. Comput. Commun. Rev.*, **5**: 3–55 (2001).

[38] Carl, C., Kesidis, G., Brooks, R. R., and Rai, S., “Denial-of-service attack-detection techniques,” *IEEE Internet Computing*, **10**: 82–89 (2006).

[39] Lung-Yut-Fong, A., Levy-Leduc, C., and Cappe, O., “Distributed detection/localization of change-points in high-dimensional network traffic data,” *Stat. Comput.*, **22**: 485–496 (2012).

[40] Li, L., and Lee, G., “DDoS attack detection and wavelets,” *Telecommun. Syst.*, **28**: 435–451 (2005).

[41] Lu, W., and Ghorbani, A. A., “Network anomaly detection based on wavelet analysis,” *EURASIP J. Adv. Sig. Pr.* (2009).

[42] Moore, D., Voelker, G. M., and Savage, S., Inferring Internet Denial-of-Service Activity, Proceedings of Usenix Security Symposium (2001).

[43] Feinstein, L., Schnackenberg, D., Balupari, R., and Kindred, D., Statistical Approaches to DDoS Attack Detection and Response, Proceedings of IEEE DAPRA Information Survivability Conference and Exposition, 303–314 (2003).

[44] Blazek, R. B., Kim, H., Rozovskii, B., and Tartakovsky, A., A Novel Approach to Detection of Denial-of-Service Attacks via Adaptive Sequential and Batch-Sequential Change-Point Detection Methods, Proceedings of IEEE Workshop on Systems, Man, and Cybernetics Information Assurance, 220–226 (2001).

[45] Wang, H., Zhang, D., and Shin, K. G., Detecting SYN Flooding Attacks, Proceedings of IEEE 21st Annual Joint Conference on Computer and Communication Societies (INFOCOM), 1530–1539 (2002).

[46] Barford, P., Kline, J., Plonka, D., and Ron, A., A Signal Analysis of Network Traffic Anomalies, Proceedings of ACM SIGCOMM Internet Measurement Workshop, 71–82 (2002).

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

[48] Kumar, K., Sangal, A. L., and Bhandari, A., Traceback Techniques Against DDoS Attacks: A Comprehensive Review, Proceedings of IEEE 2nd International Conference on Computer and Communication Technology (ICCCT), 491–498 (2011).

[49] Yu, S., Zhou, W., Doss, R., and Jia, W., “Traceback of DDoS Attacks Using Entropy Variations,” *IEEE T Parall. Distr.*, **22**: 412–425 (2011).

[50] Xiang, Y., Li, K., and Zhou, W., “Low-rate DDoS attacks detection and traceback by using new information metrics,” *IEEE T Inf. Foren. Sec.*, **6**: 426–437 (2011).

[51] Savage, S., Wetherall, D., Karlin, A., and Anderson, T., “Network support for IP traceback,” *IEEE ACM T Network*, **9**: 226–237 (2001).

[52] Mahajan, R., Bellovin, S. M., Floyd, S., Ioannidis, J., Paxson, V., and Shenker, S., “Controlling high bandwidth aggregates in the network,” *ACM Comput. Commun. Rev.*, **32**: 62–73 (2002).

[53] Lipson, H. F., Tracking and Tracing Cyber-Attacks: Technical Challenges and Global Policy Issues, CERT Coordination Center, Special Report: CMU/SEI-2002-SR-009 (2002).

[54] Kent, S., and Atkinson, R., Security Architecture for the Internet Protocol, RFC 2401 (1998).

[55] Garber, L., “Denial-of-service attacks rip the internet,” *IEEE Comput.*, **33**: 12–17 (2000).

[56] Liu, Y., Cukic, B., and Gururajan, S., “Validating neural network-based online adaptive systems: A case study,” *Software Qual. J.*, **15**: 309–326 (2007).

[57] Li, J., Liu, Y., and Gu, L., DDoS Attack Detection Based on Neural Network, Proceedings of IEEE 2nd International Symposium on Aware Computing (ISAC), 196–199 (2010).

[58] Biehl, M., Ghosh, A., and Hammer, B., “Dynamics and generalization ability of LVQ algorithms,” *J. Mach. Learn Res.*, **8**: 323–360 (2007).

[59] Agarwal, P. K., Gupta, B. B., Jain, S., and Pattanshetti, M. K., Estimating Strength of a DDoS Attack in Real Time Using ANN Based Scheme, Communications in Computer and Information Science (Springer), **157**: 301–310 (2011).

[60] Gupta, B. B., Joshi, R. C., Misra, M., Jain, A., Juyal, S., Prabhakar, R., and Singh, A. K., Predicting Number of Zombies in a DDoS Attack Using ANN Based Scheme, Communications in Computer and Information Science (Springer), **147**: 117–122 (2011).

[61] Xu, Z. H., Chen, W. B., Yang, W. F., and Liu, F., Fast Algorithm of Evolutional Learning Neural Network, Proceedings of IEEE International Conference on Intelligent Systems Design and Engineering Application (ISDEA), 262–265 (2012).

[62] Zhao, Z., Xin, H., Ren, Y., and Guo, X., Application and Comparison of BP Neural Network Algorithm in MATLAB,

Proceedings of IEEE International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), 590–593 (2010).

[63] Chang-Lung, T., Chang, A. Y., and Ming-Szu, H., Early Warning System for DDoS Attacking Based on Multilayer Deployment of Time Delay Neural Network, Proceedings of IEEE 6th International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 704–707 (2010).

[64] Li, C., Jiang, W., and Zou, X., Botnet: Survey and Case Study, Proceedings of IEEE 4th International Conference on Innovative Computing, Information and Control (ICICIC), 1184–1187 (2009).

[65] Hu, X., Knysz, M., and Shin, K. G., Measurement and Analysis of Global IP-Usage Patterns of Fast-Flux Botnets, Proceedings of IEEE INFOCOM, 2633–2641 (2011).

[66] Bilge, L., Kirda, E., Kruegel, C., and Balduzzi, M., EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis, Proceedings of Internet Society 18th Annual Network & Distributed System Security Symposium (2011).

[67] Lee, J., Kwon, J., Shin, H., and Lee, H., Tracking Multiple C&C Botnets by Analyzing DNS Traffic, Proceedings of IEEE 6th Workshop on Secure Network Protocols (NPSec), 67–72 (2010).

[68] Stone-Gross, B., Cova, M., Cavallaro, L., Gilbert, B., Szydlowski, M., Kemmerer, R., Kruegel, C., and Vigna, G., Your Botnet is My Botnet: Analysis of a Botnet Takeover, Proceedings of 16th ACM Conference on Computer and Communications Security (CCS), 635–647 (2009).

[69] Holz, T., Gorecki, C., Rieck, K., and Freiling, F. C., Measuring and Detecting Fast-Flux Service Networks, Proceedings of Internet Society 16th Annual Network & Distributed System Security Symposium (2008).

[70] Caglayan, A., Toothaker, M., Drapeau, D., Burke, D., and Eaton, G., Real-Time Detection of Fast Flux Service Networks, Proceedings of IEEE Cybersecurity Applications & Technology Conference for Homeland Security (CATCH), 285–292 (2009).

[71] Ma, J., Saul, L. K., Savage, S., and Voelker, G. M., Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs, Proceedings of 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 1245–1254 (2009).

[72] Ma, J., Saul, L. K., Savage, S., and Voelker, G. M., Identifying Suspicious URLs: An Application of Large-Scale Online Learning, Proceedings of ACM 26th Annual International Conference on Machine Learning (ICML), 681–688 (2009).

[73] Naveed, M., Un Nihar, S., and Babar, M. I., Network Intrusion Prevention by Configuring ACLs on the Routers Based on Snort IDS Alerts, Proceedings of IEEE 6th International Conference on Emerging Technologies (ICET), 234–239 (2010).

[74] SNORT: Open Source Network Intrusion, Prevention and Detection System (IDS/IPS), ⟨http://www.snort.org⟩ (2012).

[75] Sidheeq, M., Dehghantanha, A., and Kananparan, G., Utilizing Trusted Platform Module to Mitigate Botnet Attacks, Proceedings of IEEE International Conference on Computer Applications and Industrial Electronics (ICCAIE), 245–249 (2010).

[76] Trusted Computing Group (TCG), ⟨http://www.trustedcomputinggroup.org⟩ (2012).

[77] Ari, I., Hong, B., Miller, E. L., Brandt, S. A., and Long, D. D. E., Managing Flash Crowds on the Internet, Proceedings of IEEE/ACM 11th International Symposium on Modeling, Analysis and Simulation of Computer Telecommunications Systems (MASCOTS), 246–249 (2003).

[78] Yuan, J., and Mills, K., "Monitoring the macroscopic effect of DDoS flooding attacks," *IEEE T Depend Secure*, **2**: 324–335 (2005).

[79] Thatte, G., Mitra, U., and Heidemann, J., "Parametric methods for anomaly detection in aggregate traffic," *IEEE ACM T Network*, **19**: 512–525 (2011).

[80] He, X., Papadopoulos, C., Heidemann, J., Mitra, U., and Riaz, U., "Remote detection of bottleneck links using spectral and statistical methods," *Comput. Network*, **53**: 279–298 (2009).

[81] Li, K., Zhou, W., Li, P., Hai, J., and Liu, J., Distinguishing DDoS Attacks from Flash Crowds Using Probability Metrics, Proceedings of IEEE 3rd International Conference on Network and System Security (NSS), 9–17 (2009).

[82] Xie, Y., and Yu, S. Z., "Monitoring the application-layer DDoS attacks for popular websites," *IEEE ACM T Network*, **17**: 15–25 (2009).

[83] Krashakov, S. A., Teslyuk, A. B., and Shchur, L. N., "On the universality of rank distributions of website popularity," *Comput. Network*, **50**: 1769–1780 (2006).

[84] Shlens, J., A Tutorial on Principal Component Analysis, ⟨http://sloan-swartz.salk.edu/˜shlens/pca.pdf⟩ (2009).

[85] Xie, Y., and Yu, S. Z., A Novel Model for Detecting Application Layer DDoS Attacks, Proceedings of IEEE 1st International Multi-Symposiums on Computer and Computational Sciences (IMSCCS), 56–63 (2006).

[86] Ranjan, S., Swaminathan, R., Uysal, M., Nucci, A., and Knightly, E., "DDoS-shield: DDoS-resilient scheduling to counter application layer attacks," *IEEE ACM T Network*, **17**: 26–39 (2009).

[87] Ranjan, S., Swaminathan, R., Uysal, M., and Knightly, E., DDoS-Resilient Scheduling to Counter Application Layer Attacks under Imperfect Detection, Proceedings of IEEE 25th International Conference on Computer Communications (INFOCOM), 1–13 (2006).

[88] Ahn, L. V., Blum, M., and Langford, J., "Telling humans and computers apart automatically," *Commun. ACM*, **47**: 56–60 (2004).

[89] Caum, L. O., Why is CAPTCHA so annoying?, ⟨http://lorenzocaum.com/blog/why-is-captcha-so-fing-annoying/⟩ (2011).

[90] Mori, G., and Malik, J., Recognizing Objects in Adversarial Clutter: Breaking a Visual CAPTCHA, Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, I-134–I-141 (2003).

[91] Athanasopoulos, E., and Antonatos, S., Enhanced CAPTCHAs: Using Animation to Tell Humans and Computers Apart, Communications and Multimedia Security (Springer), **4237**: 97–108 (2006).

[92] Fu, Z., Papatriantafilou, M., and Tsigas, P., "Mitigating distributed denial of service attacks in multiparty applications in the presence of clock drifts," *IEEE T Depend Secure*, **9**: 401–413 (2012).

[93] Hari, K., and Dohi, T., Sensitivity Analysis of Random Port Hopping, Proceedings of IEEE 7th International Conference on Ubiquitous Intelligence & Computing and 7th International Conference on Autonomic & Trusted Computing (UIC/ATC), 316–321 (2010).

[94] Badishi, G., Herzberg, A., and Keidar, I., "Keeping denial-of-service attackers in the dark," *IEEE T Depend Secure*, **4**: 191–204 (2007).

[95] Tian, H., Bi, J., and Jiang, X., "An adaptive probabilistic marking scheme for fast and secure traceback," *Networking Science*, **2**: 42–51 (2013).

[96] Paruchuri, V., Durresi, A., and Chellappan, S., TTL Based Packet Marking for IP Traceback, Proceedings of IEEE Global Telecommunications Conference (GLOBECOM), 1–5 (2008).

[97] Liu, J., Lee, Z. J., and Chung, Y. C., "Dynamic probabilistic packet marking for efficient IP traceback," *Comput. Network*, **51**: 866–882 (2007).

[98] Khanna, S., Venkatesh, S. S., Fatemieh, O., Khan, F., and Gunter, C. A., "Adaptive selective verification: An efficient adaptive countermeasure to thwart DoS attacks," *IEEE ACM T Network*, **20**: 715–728 (2012).

[99] Thapngam, T., Yu, S., Zhou, W., and Makki, S. K., "Distributed denial of service (DDoS) detection by traffic pattern analysis," *Peer Peer Netw. Appl.*, DOI: 10.1007/s12083-012-0173-3 (2012).

[100] Gilad, Y., and Herzberg, A., "LOT: A defense against IP spoofing and flooding attacks," *ACM T Inform. Syst. Se*, **15**: (2012).

[101] Shiaeles, S. N., Katos, V., Karakos, A. S., and Papadopoulos, B. K., "Real time DDoS detection using fuzzy estimators," *Comput. Secur.*, **31**: 782–790 (2012).

[102] Spyridopoulos, T., Karanikas, G., Tryfonas, T., and Oikonomou, G., "A game theoretic defence framework against DoS/DDoS cyber attacks," *Comput. Secur.*, DOI: 10.1016/j.cose.2013.03.014 (2013).

[103] Zhao, D., Traore, I., Sayed, B., Lu, W., Saad, S., Ghorbani, A., and Garant, D., "Botnet detection based on traffic behavior analysis and flow intervals," *Comput. Secur.*, DOI: 10.1016/j.cose.2013.04.007 (2013).

[104] Gu, G., Porras, P., Yegneswaran, V., Fong, M., and Lee, W., BotHunter: Detecting Malware Infection through ids-Driven Dialog Correlation, Proceedings of 16th USENIX Security Symposium (2007).

[105] Senthilmahesh, P. C., Hemalatha, S., Rodrigues, P., and Shanthakumari, A., DDoS Attacks Defense System Using Information Metrics, Proceedings of 3rd International Conference on Trends in Information, Telecommunication and Computing, Lecture Notes in Electrical Engineering (Springer, New York), 25–30 (2012).

[106] Zhou, W., Jia, W., Wen, S., Xiang, Y., and Zhou, W., "Detection and defense of application-layer DDoS attacks in backbone web traffic," *Future Gen. Comput. Syst.*, DOI: 10.1016/j.future.2013.08.002 (2013).

[107] Sujatha, S., and Radcliffe, P. J., A Novel Framework to Detect and Block DDoS Attack at the Application Layer, Proceedings of 2013 IEEE TENCON Spring Conference, 578–582 (2013).

[108] Prasad, K. M., Karthik, M. G., and Krishna, E. S. P., An Efficient Flash Crowd Attack Detection to Internet Threat Monitors (ITM) Using Honeypots, Proceedings of 2nd International Conference on Advances in Computing and Information Technology, Advances in Intelligent Systems and Computing (Springer-Verlag, Berlin, Heidelberg), 595–610 (2012).

[109] Manoj, R., and Tripti, C., An effective approach to detect DDoS attack, Proceedings of 2nd International Conference on Advances in Computing and Information Technology, Advances in Intelligent Systems and Computing (Springer-Verlag, Berlin, Heidelberg), 339–345 (2012).

[110] Wood, A. D., and Stankovic, J. A., "Denial of service in sensor networks," *IEEE Comput.*, **35**: 54–62 (2002).

[111] Raymond, D. R., and Midkiff, S. F., "Denial-of-service in wireless sensor networks: Attacks and defenses," *IEEE Pervas Comput.*, **7**: 74–81 (2008).

[112] Pelechrinis, K., Iliofotou, M., and Krishnamurthy, S. V., "Denial of service attacks in wireless networks: The case of jammers," *IEEE Commun. Surv. Tut.*, **13**: 245–257 (2011).

[113] Xu, W., Trappe, W., Zhang, Y., and Wood, T., The Feasibility of Launching and Detecting Jamming Attacks in Wireless Networks, Proceedings of ACM MobiHoc Conference, 46–57 (2005).

[114] Acharya, M., and Thuente, D., Intelligent Jamming Attacks, Counterattacks and (Counter)2 Attacks in 802.11b Wireless Networks, Proceedings of OPNETWORK-2005 Conference (2005).

[115] Acharya, M., Sharma, T., Thuente, D., and Sizemore, D., Intelligent Jamming Attacks in 802.11b Wireless Networks, Proceedings of OPNETWORK-2004 Conference (2004).

[116] Gummadi, R., Wetheral, D., Greenstein, B., and Seshan, S., Understanding and Mitigating the Impact of RF Interference on 802.11 Networks, Proceedings of 2007 ACM Conference on Applications, Technologies, Architectures and Protocols for Computer Communications (SIGCOMM), 385–396 (2007).

[117] Bellardo, J., and Savage, S., 802.11 Denial-of-Service Attacks: Real Vulnerabilities and Practical Solutions, Proceedings of USENIX Security Symposium (2003).

[118] Brown, T. X., James, J. E., and Sethi, A., Jamming and Sensing of Encrypted Wireless Ad hoc Networks, Proceedings of ACM MobiHoc Conference (2006).

[119] Kaur, G., Chaba, Y., and Jain, V. K., "Distributed denial of service attacks in mobile ad hoc networks," *World Academy of Science, Engineering and Technology*, **73**: 725–727 (2011).

[120] Zhang, Y., and Lee, W., Intrusion Detection in Wireless Ad-hoc Networks, Proceedings of ACM 6th International Annual Conference on Mobile Computing and Networking (MobiCom), 491–498 (2000).

[121] Bhargava, S., and Agrawal, D. P., Security Enhancements in AODV Protocol for Wireless Ad hoc Networks, Proceedings of IEEE VTS 54th Vehicular Technology Conference (VTC), 2143–2147 (2001).

[122] Deng, H., Zeng, Q. A., and Agrawal, D. P., SVM-Based Intrusion Detection System for Wireless Ad hoc Networks, Proceedings of IEEE 58th Vehicular Technology Conference (VTC), 2147–2151 (2003).

[123] Albers, P., Camp, O., and Percher, J. M., Security in Ad hoc Networks: A General Intrusion Detection Architecture Enhancing Trust Based Approaches, Proceedings of 1st International Workshop on Wireless Information Systems (2002).

[124] Mitrokotsa, A., Komninos, N., and Douligeris, C., Intrusion Detection with Neural Networks and Watermarking Techniques for MANET, Proceedings of IEEE International Conference on Pervasive Services, 118–127 (2007).

[125] Xiang, M., Chen, Y., Ku, W. S., and Su, Z., Mitigating DDoS Attacks Using Protection Nodes in Mobile Ad hoc Networks, Proceedings of IEEE Global Telecommunications Conference (GLOBECOM), 1–6 (2011).

[126] Mahmood, R. A. R., Amin, A. H. M., Amir, A., and Khan, A. I., Lightweight and Distributed Attack Detection Scheme in Mobile Ad hoc Networks, Proceedings of ACM 7th International Conference on Advances in Mobile Computing and Multimedia (MoMM), 162–169 (2009).

[127] Devi, P., and Kannammal, A., A Hybrid Defense Mechanism for DDoS Attacks Using Cluster Analysis in MANET, Proceedings of ACM International Conference on Advances in Computing, Communications and Informatics (ICACCI), 287–291 (2012).

[128] Tupakula, U., Varadharajan, V., and Vuppala, S. K., Countering DDoS Attacks in WLAN, Proceedings of ACM 4th International Conference on Security of Information and Networks (SIN), 119–126 (2011).

[129] Eian, M., and Mjolsnes, S. F., The Modeling and Comparison of Wireless Network Denial of Service Attacks, Proceedings of ACM 3rd SOSP Workshop on Networking, Systems and Applications on Mobile Handhelds (2011).

[130] Bicakci, K., and Tavli, B., "Denial-of-service attacks and countermeasures in IEEE 802.11 wireless networks," *Comput. Stand. Inter.*, **31**: 931–941 (2009).

[131] Keromytis, A. D., "Voice-over-IP security: Research and practice," *IEEE Secur. Priv.*, **8**: 76–78 (2010).

[132] Jacobson, V., Thornton, J. D., Smetters, D. K., Briggs, N., Plass, M., Braynard, R., Shi, E., Barber, S., Solis, I., Mosko, M., and Garcia-Luna, J. J., Content-Centric Networking. Whitepaper; Palo Alto Research Center, 2–4 (2007).

[133] Jacobson, V., Smetters, D. K., Thornton, J. D., Plass, M., Briggs, N., and Braynard, R., "Networking named content," *Commun. ACM*, **55**: 117–124 (2012).

[134] Arianfar, S., Nikander, P., and Ott, J., On Content-Centric Router Design and Implications, Proceedings of ACM Workshop on Re-Architecting the Internet (2010).

[135] Ul Haque, M., Pawlikowski, K., Willig, A., and Bischofs, L., Performance Analysis of Blind Routing Algorithms Over Content Centric Networking Architecture, Proceedings of IEEE International Conference on Computer and Communication Engineering (ICCCE), 922–927 (2012).

[136] Carofiglio, G., Gallo, M., Muscariello, L., and Perino, D., Modeling Data Transfer in Content-Centric Networking, Proceedings of ACM 23rd International Teletraffic Congress (ITC), 111–118 (2011).

[137] Carofiglio, G., Gehlen, V., and Perino, D., Experimental Evaluation of Memory Management in Content-Centric Networking, Proceedings of IEEE International Conference on Communications (ICC), 1–6 (2011).

[138] Kang, W., Sim, B., Kim, J., Paik, E., and Lee, Y., A Network Monitoring Tool for CCN, Proceedings of IEEE World Telecommunications Congress (WTC), 1–3 (2012).

[139] Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., and Zaharia, M., "A view of cloud computing," *Commun. ACM*, **53**: 50–58 (2010).

[140] Amazon EC2: ⟨http://aws.amazon.com/ec2/⟩ (2013).

[141] Google App Engine: ⟨https://developers.google.com/appengine/⟩ (2013).

[142] Windows Azure: ⟨http://www.windowsazure.com/en-us/⟩ (2013).

[143] Subashini, S., and Kavitha, V., "A survey on security issues in service delivery models of cloud computing," *J. Netw. Comput. Appl.*, **34**: 1–11 (2011).

[144] Anthes, G., "Security in the cloud," *Commun. ACM*, **53**: 16–18 (2010).

[145] Chen, Q., Lin, W., Dou, W., and Yu, S., CBF: A Packet Filtering Method for DDoS Attack Defense in Cloud Environment, Proceedings of IEEE 9th International Conference on Dependable, Autonomic and Secure Computing (DASC), 427–434 (2011).

[146] Du, P., and Nakao, A., DDoS Defense as a Network Service, Proceedings of IEEE Network Operations and Management Symposium (NOMS), 894–897 (2010).

[147] Yang, L., Zhang, T., Song, J., Wang, J. S., and Chen, P., Defense of DDoS Attack for Cloud Computing, Proceedings of IEEE International Conference on Computer Science and Automation Engineering (CSAE), 626–629 (2012).

[148] Joshi, B., Vijayan, A. S., and Joshi, B. K., Securing Cloud Computing Environment Against DDoS Attacks, Proceedings of IEEE International Conference on Computer Communication and Informatics (ICCCI), 1–5 (2012).

[149] Idziorek, J., Tannian, M., and Jacobson, D., "Insecurity of cloud utility models," *IT Professional*, **15**: 22–27 (2013).

[150] Chonka, A., Xiang, Y., Zhou, W., and Bonti, A., "Cloud security defense to protect cloud computing against HTTP-DoS and XML-DoS attacks," *J. Network Comput. Appl.*, **34**: 1097–1107 (2011).

[151] Zissis, D., and Lekkas, D., "Addressing cloud computing security issues," *Future Gen. Comput. Syst.*, **28**: 583–592 (2012).

[152] Khorshed, M. T., Ali, A. B. M. S., and Wasimi, S. A., "A survey on gaps, threat remediation challenges and some thoughts for proactive attack detection in cloud computing," *Future Gen. Comput. Syst.*, **28**: 833–851 (2012).

[153] Idziorek, J., and Tannian, M., "Security analysis of public cloud computing," *Int. J. Commun. Network Distrib. Syst.*, **9**: 4–20 (2012).

[154] Karrer, R. P., Kuehn, U., and Huehn, T., Joint Application and Network Defense Against DDoS Flooding Attacks in the Future Internet, Proceedings of IEEE 2nd International Conference on Future Generation Communication and Networking (FGCN), 11–16 (2008).

[155] Hashim, F., Kibria, M. R., and Jamalipour, A., Detection of DoS and DDoS Attacks in NGMN Using Frequency Domain Analysis, Proceedings of IEEE 14th Asia-Pacific Conference on Communications (APCC), 1–5 (2008).

[156] Chung, Y., "Distributed denial of service is a scalability problem," *ACM Comput. Commun. Rev.*, **42**: 69–71 (2012).

[157] Walfish, M., Vutukuru, M., Balakrishnan, H., Karger, D., and Shenker, S., "DDoS defense by offense," *ACM T Comput. Syst.*, **28**: (2010).