doi:10.1093/comjnl/bxh000

Protection against Denial of Service Attacks: A Survey

Georgios Loukas and Gülay Öke

Intelligent Systems and Networks Group, Imperial College London
Email: georgios.loukas@imperial.ac.uk, g.oke@imperial.ac.uk

Denial of Service (DoS) is a prevalent threat in today's networks because DoS attacks are easy to launch, while defending a network resource against them is disproportionately difficult. Despite the extensive research in recent years, DoS attacks continue to harm, as the attackers adapt to the newer protection mechanisms. For this reason, we start our survey with a historical timeline of DoS incidents, where we illustrate the variety of types, targets and motives for such attacks and how they evolved during the last two decades. We then provide an extensive literature review on the existing research on denial of service protection with an emphasis on the research of the last years and the most demanding aspects of defence. These include traceback, detection, classification of incoming traffic, response in the presence of an attack, and mathematical modelling of attack and defence mechanisms. Our discussion aims to identify the trends in DoS attacks, the weaknesses of protection approaches and the qualities that modern ones should exhibit, so as to suggest new directions that DoS research can follow.

Received 6 May 2009

1. INTRODUCTION

A Denial of Service attack (DoS) is any intended attempt to prevent legitimate users from reaching a specific network resource. Such attacks have been known to the network research community since the early 1980s. In fact, a 1983 paper provides one of the first descriptions of DoS in operating systems [1] and in 1985 R.T. Morris comments on the fact that there is no provision in the Internet Protocol to discover the true origin of a packet [2]. A decade later it became clear that attackers would routinely exploit this weakness by faking their source address and sending large volumes of traffic to victim computers. Today, DoS attacks are usually distributed: the attacker takes control of a large number of lightly protected computers, such as those that do not have firewall or up-to-date antivirus software, and orders them to send traffic simultaneously from all machines to the victim computer (Figure 1). As a result, some routers and links in the vicinity of the target are overwhelmed, and a number of legitimate clients may not be able to connect to it. Typical victims of such attacks are the servers of e-commerce websites, news websites, corporate networks, banks, and governmental websites. Our aim with this paper is to provide a comprehensive survey of the existing research on Denial of Service attacks and discuss the directions that this research can take in the near future.

Our survey begins with a timeline of the most significant DoS attack incidents to date, with brief descriptions of the types of attack used in each case. We then discuss the elements that an ideal DoS defence framework should contain and we go through the existing proposals on the most important challenges in DoS research. These include the elimination of IP spoofing, the classification of incoming traffic between normal and attack, the detection of an attack, and the response against it. In Section 7, we present the few relevant mathematical models that have been proposed in the literature, and and in Section 8 we conclude with our observations and recommendations for future research on DoS.

2. INCIDENTS, TYPES AND MOTIVES

Although DoS attacks existed during the 1980s and early 1990s, at the beginning they were not viewed as high-profile security incidents by the general public. This perception started to change as the Internet was becoming a mainstream medium. In this section, we present a timeline of the most notable DoS incidents, followed by a brief description of each new type of attack used at each incident.

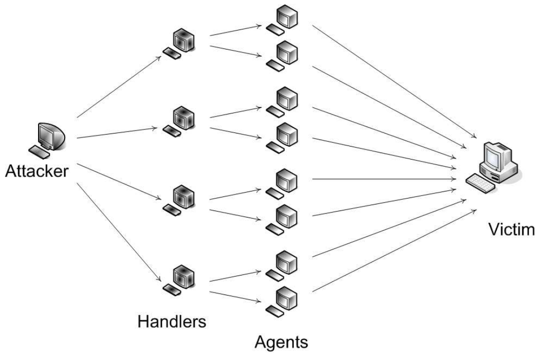

FIGURE 1. DDoS: In a Distributed DoS attack the attacker compromises a first tier of vulnerable computers and through them orders a second tier of several more computers to simultaneously attack a specific target

2.1. The early years: before the year 2000

In September 1996, a "SYN Flood" DoS attack took the New York City Internet service provider Panix off-line for a week, while subsequent attacks disabled the web servers of the Internet Chess Club and The New York Times. Two months later, the first commercial product specifically designed for DoS attacks was released [3]. It detected attacks by watching for incoming SYN packets, and responded by resetting the connections if the victim computer received traffic at rate higher than a certain threshold. However, it failed to halt an attack on Webcom's main server which knocked thousands of commercial websites off-line. The attacker had randomised the IP addresses and the attack rate was 200 packets/sec, which was very high at the time. In "**SYN Flood**" attacks, the attacking system sends SYN messages to the victim server system that appear to be legitimate but in fact reference a client system that is unable to respond to the SYN/ACK messages. This means that the final ACK message will never be sent to the victim server system and due to the many half-open connections, the victim server system becomes eventually unable to accept any new incoming connections [4].

In January 1997, a teenager attacked the IRC network Undernet and several ISPs in Norway, Romania, the United Kingdom and the United States, with a combination of "ping" and "SYN Flood" attacks. At each stop, he logged onto the server, obtained root access, then deleted files and cancelled accounts. The "**ping attack**" is one of the simplest DoS attacks, where the victim is flooded with more TCP/ICMP packets than it can handle. In "**IRC-based DDoS attacks**", an IRC communication channel is used to connect the client to the agents. The attackers can use legitimate IRC ports for sending commands to the agents, which makes their tracking more difficult, because IRC servers tend to receive large volumes of traffic. The attacker no longer needs to maintain a list of agents, since she can simply log on to the IRC server and see a list of all available agents. The agent software installed in the IRC network usually communicates with the IRC channel and notifies the attacker when the agent is up and running. IRC networks also provide for easy file sharing, which is one of the passive methods of agent code distribution and an easy way for attackers to secure secondary victims to participate in their attacks [5].

In January 1998, DALnet and other IRC networks became targets of "smurfing", where the attacker is using ICMP echo request packets directed to IP broadcast addresses from remote locations to generate DoS attacks. There are three parties in these attacks: the attacker, the intermediary, and the victim. The intermediary receives an ICMP echo request packet directed to the IP broadcast address of their network. If the intermediary does not filter ICMP traffic directed to IP broadcast addresses, many of the machines on the network will receive this ICMP echo request packet and send an ICMP echo reply packet back. When all the machines on a network respond to this ICMP echo request, the result can be severe network congestion and outages. When the attackers create these packets, they do not use the IP address of their own machine as the source address. Instead, they create forged packets that contain the source address of the attacker's intended victim. The result is that when all the machines at the intermediary's site respond to the ICMP echo requests, they send replies to the victim's machine, which is overwhelmed by the amount of traffic [6]. Similar is the "**fraggle**" attack, which uses UDP packets instead of ICMP echo packets.

During the same period, the Pentagon, NASA, several American military network systems and hundreds of universities were targeted in a series of DoS attacks launched by a single individual, a teenager from Israel. The hacker used mainly "Teardrop" and "Bonk" techniques, which exploited known vulnerabilities of the Microsoft Windows operating systems, and succeeded against those computers that were not up to date with the latest security patches. "**Teardrop**" attacks exploit the fact that the Internet Protocol requires fragmentation of the packets that are too large for the next router to handle. Each fragmented packet identifies an offset to the beginning of the first packet that enables the entire packet to be reassembled by the receiving system. In the teardrop attack, the attacker puts a confusing value in the second or later fragment, and if the receiving operating system cannot cope with such fragmentation then it may crash. Ironically, a month after the release of the relative patch by Microsoft, a new variety of Teardrop emerged, **"Bonk"**, which worked specifically on a vulnerability created by this patch.

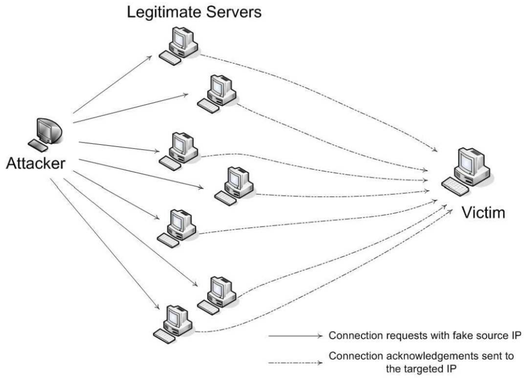
Legitimate Servers
Attacker
Victim
Connection requests with fake source IP
Connection acknowledgements sent to the targeted IP

FIGURE 2. DRDoS: In a Distributed Reflector DoS attack (DRDoS) the attacker uses a fake source IP (the target's) and sends connection requests to several legitimate servers. When these servers respond they send their acknowledgement packets to the attacker's target

2.2. The period of highest known frequency of DoS attacks: from 2000 to 2004

In February 2000, Yahoo, eBay, Amazon, Datek, Buy, CNN, ETrade, ZDNet and Dell were among the high-profile targets of a 15-year old Canadian nicknamed "Mafiaboy". The attack, which reached the rate of 1GB/sec caused unprecedented financial damage and changed the public perception regarding DoS. From then on, DoS and in general Internet crime, started moving from the IRC networks to e-commerce. "Mafiaboy" was sentenced in September 2001 for causing over $1.7 billion damages.

BT, BTInternet and Gameplay were attacked in July 2000 by a frustrated anonymous customer, as revenge for bad service. This was the first of many DoS attacks used as a form of protest.

In January 2001, a new type of attack appeared, which reached the rate of 90 Mbps against Register.com [7]. A few months later Paxson et al. published the first and definitive paper on the analysis of that new type [8], which took the name Distributed Reflector (DRDoS). The concept of the "**Distributed Reflector DoS attack**" is very similar to that of the "Smurf" attack. The attacker orders his army of compromised computers to send connection requests to several perfectly legitimate computers, but using the victim's IP as the source in their packets. When the legitimate computers reply to these requests, the receiver of all is the victim.

The most famous network security incident of 2001 was a computer worm of Chinese origin, known as "Code Red", which swept through 250,000 computers in nine hours, with the infected computers being programmed to simultaneously launch a DoS attack against the website of the White House. The White House was forced to change its numerical IP Web address and prompted the Pentagon to take its public Websites temporarily off-line. The damage inflicted by "Code Red" was estimated at about $2.6 billion.

Also in 2001, a British teenager was accused of launching a DDoS attack on the Port of Houston's IT systems, which rendered the Port's web service inaccessible for several hours. The attack was allegedly aimed at a chatroom user, and the Port's computers were only used as intermediaries. The teenager claimed that his computer could have been taken over by a hacker using a Trojan Horse program and he was found not guilty. The affected server contained crucial data on navigation, tides, water depths and weather. This was the first known occasion where part of a country's national infrastructure was disabled by an electronic attack.

In June 2002, the Website of the government of Pakistan was the victim of a politically motivated attack launched by Indian hackers that used "YAHA", a worm with Denial of Service payload. Similarly to "Code Red", "YAHA" caused an infected computer to make repeated connection attempts to the Pakistani government's website and attempted to terminate anti-virus and firewall software.

In October 2002, the DNS root servers [1] were under attack for about an hour, with several becoming unavailable for regular Internet traffic. The remaining root servers withstood the attack and ensured that the Internet's overall performance was not degraded significantly. Although it was hardly noticeable to the average end-user, this was the most serious hacker attack ever attempted on a key piece of the Internet infrastructure. A year later most of the root servers had applied a new routing technique known as Anycast, with which several operators are replicating these servers around the world. As a result, in February 2007, when hackers targeted again the DNS system of the Internet, including the Internet Corporation for Assigned Names and Numbers (ICANN) and UltraDNS, there was not a single point of failure caused by the attack.

In January 2003, yet another worm with DoS payload appeared to cause major disruptions worldwide. The worm hit South Korea particularly hard because although it had the world's highest penetration of broadband Internet services at the time, less than 40% of South Korean firms had installed any firewall. The losses in the South Korean stock market were estimated at about US$860,000. A few months later, the same worm caused a 5-hour outage to the safety monitoring system of the nuclear power plant at Ohio. The "**SQL Slammer**" worm was self-propagating malicious code that exploited a known vulnerability in Microsoft SQL Server 2000. Once the worm compromised a machine, it would try to propagate itself to other randomly chosen IP addresses [9].

[1] At the time, there were only 13 root servers, operated by U.S. government agencies, universities, nonprofit organisations and companies.

In August 2004, a corporate executive in Massachusetts was charged with hiring hackers to launch DoS attacks and cause a total of $2 billion in losses to three competitors. The attacks had begun in October 2003 and were mainly SYN and HTTP Floods. In an "**HTTP Flood**", the attacker uses a large number of compromised computers that simultaneous and request web content, such as images, from a victim website. A variation is the "**HTTP Spidering**" attack, which starts from a given HTTP link and then follows all links on the website in a recursive way [10], inspired by the way search engines gather their data.

2.3. Recent trends: from 2004 up to 2009

Since 2004, DoS incidents have been deliberately not widely publicised, as the scene has shifted to the sensitive field of economic crime and DoS incidents harm the victims' reputation in the eyes of the increasingly security-aware public. Major new trends include Cyber-extortion and bot armies.

In January 2006, the million dollar page, a British teenager's novel advertising idea to earn him $1m in 4 months, became very quickly famous around the world. This instant media attention drew the attention of cyber-extortionists, who bombarded the website with intense DoS attacks, initially asking for $5,000 and later $50,000 to avert them. The website was under attack for a whole month. DoS-related cyber-extortion has escalated recently. The general consensus is that the victims should avoid paying at all costs, since otherwise they appear as "soft targets" to attackers, information very easily spread among cyber-criminals. In reality, however, Internet downtime is so damaging for the finances and the reputation of online companies that most victims choose to pay and simply inform the authorities.

Several groups of cyber-criminals specialise in compromising large numbers of computers vulnerable to Denial of Service attacks. They build these "armies of bots", of a few thousand to allegedly up to 1.5 million computers, and rent them to potential DoS attackers. In May 2006, a 20-year old "botmaster" was sentenced to five years in prison for hijacking 500,000 computers. He was selling access to them to other hackers, who used them to launch Dos attacks and send spam emails.

More recently, during the August 2008 armed conflict between Russia and Georgia, a series of coordinated DoS attacks of unidentified origin crippled Georgia's Internet infrastructure [11]. Similar DoS attacks had been launched a year earlier against Estonia's Internet infrastructure [12].

2.4. The historical timeline

Although DoS attacks are launched since the beginning of computer networks, they were not considered a significant topic of research until relatively recently, when they started harming ISPs, governmental websites and the e-commerce. The effectiveness of these attacks and their subsequent publicity prompted the influx of newer and even more effective attacking techniques against an increasingly wide range of targets. With DoS techniques becoming distributed and powerful attacking tools being readily available on the Internet, it became quickly apparent that DoS cannot be handled in the same way as other computer security issues. For example viruses have always been countered with dedicated antivirus software running on the victim computer, but DoS attacks are aiming at overwhelming the target resource altogether, so that the victim cannot employ a defence on its own. The fact that the Internet operates on old networking protocols with limited provision for security is yet another advantage for the attackers. Of course, the increase in research interest did provide solutions, which have recently managed to halt the escalation of the DoS phenomenon. Distributed defence techniques have been designed and the majority of the DoS attacks can now be countered in networks where some sort of defence has been deployed. This can be considered as the end of an era, during which a "script-kiddie" could download a tool and launch an attack against practically any website. Today, attacks have shifted towards economic crime and cyber-warfare, and although less widespread they can be much more harmful. As a result, the new era of DoS research has to produce even more effective solutions with even less overhead in the absence of an attack and as small disruption in the presence of one. In the next section we present a comprehensive survey of the existing research, from the initial ideas that shaped today's DoS defences, to the latest that we expect to shape the DoS defences of the near future.

3. DEFENCE MECHANISMS AGAINST DENIAL OF SERVICE

The extreme diversity of DoS attacks has produced similarly diverse protection proposals from the network security research community. In most cases a complete protection architecture should include the following elements:

- ***Detection*** of the existence of an attack. The detection can be either anomaly-based or signature-based, or a hybrid of these two. In anomaly-based detection, the system recognises a deviation from the standard behaviour of its clients, while in signature-based it tries to identify the characteristics of known attack types.
- ***Classification*** of the incoming packets into valid (normal packets) and invalid (DoS packets). As in detection, one can choose between anomaly-based and signature-based classification techniques.
- ***Response***. In the most general sense, the protection system either drops the attacking packets in a timely fashion or renders them

harmless by redirecting them into a trap for further evaluation and analysis.

Detection and Classification usually overlap, since the method used to detect the existence of an attack often provides the necessary information to start responding towards probable normal and probable DoS traffic. Also, all three elements of protection may benefit by the use of an additional secondary element, which is the ***traceback*** of the real source of the traffic.

4. DETECTING THE EXISTENCE OF A DOS ATTACK

Attack detection would not be necessary in the ideal case of a defence architecture with proactive qualities that would render such attacks ineffective. However, DoS attacks against one's network do not happen as often as to justify the processing demands of a continuously operating proactive system. For this reason, it is preferable to have a detection system that initiates defence on demand. In this section, we present a summary of the existing literature on such detection methods.

4.1. The use of Learning Techniques

Learning paradigms, such as neural networks, radial basis functions and genetic algorithms are increasingly used in DoS detection because of the intelligent and automatic classification that they can offer. In [13], a statistical pre-processor is used to extract features from packets that can indicate the existence of a DoS attack. A feature vector is changed to numerical form and feeded to an unsupervised Adaptive Resonance Theory net (ART). In ART nets, learning is accomplished by updating the cluster weights according to the relative similarity between the weights and the input. The ART is first trained with normal and attack types of input vectors and then in real-time classifies the packets using the adjusted cluster weights. Features that can be used are the ratio of types of packets and the average packet header and packet sizes.

Another learning-based technique is presented in [14]. Appropriate data are collected from the incoming packets, a feature estimator evaluates the frequencies of their appearance, and a RBFNN detector classifies them as normal or DoS. Experiments with TCP traffic show that the TTL, window size and some of the TCP flags do not necessarily provide useful information about the occurrence of a DoS attack. The success of the approach varies considerably depending on the choice of time frame, set of input features, number of hidden neurons and training data. As a natural continuation of this work, the same authors have used genetic algorithms to study the optimum feature selection problem [15]. With the same detection technique, they present a method to determine the importance of each input feature in DoS detection, by first selecting an initial set of 44 statistical features and then evaluating the most relevant features with a genetic algorithm. The usefulness of selected features vary with the number of hidden neurons and with the selection and mutation probabilities. Generally, the SYN and URG flags, as well as some specific ranges of ports are the most significant for the identification of a DoS attack. Similar detection techniques can be found in [16], where flooding attacks are detected by minimisation of the generalisation error bound of the RBFNN, and in [17, 18], where RBFNNs are used as classifiers for an entropy-based feature grouping in a multiple classifier system (MCS). MCS systems are also proposed in [19] and [20].

In [21], DoS attacks are detected with traffic rate analysis (TRA) and three machine learning algorithms, namely C4.5, CN2 and a Bayesian classifier. The TRA system classifies IP packets into TCP, UDP or ICMP packets and further sums up the SYN, FIN, RST, ACK, PSH and URG flags for TCP packets. For a specific sampling period, the rate of a certain type of flag is determined by dividing the total number of this flag's occurrences to the total number of TCP packets observed while the ratio of the number of TCP, UDP or ICMP packets to the total number of IP packets gives the protocol rate. The authors provide experimental results in a simulated TCP-based network, which show that SYN and ACK flag rates for inbound traffic provide significant information for detection of SYN flooding attacks, while the best performance is obtained by the Bayesian classifier. The usefulness of Bayesian classifiers for DoS detection is also investigated in [22], where likelihood estimation is combined with a feedforward implementation of the Random Neural Network (RNN) [51]. The approach is evaluated for different traffic data in a large networking test-bed. The authors use a combination of statistical features collected in real-time from the incoming traffic, namely the bitrate, the increase in bitrate, the entropy, the Hurst parameter, the delay and the increase of the delay.

Another method to identify and select the useful features in DoS detection is presented in [23], which describes a data mining approach based on an automatic feature selection mechanism combined with a neural network classifier. The authors use a decision tree to select the best out of a set of candidate features, which are then used in the neural network classifier. In their investigation, the packet count per flow and the source port variance of TCP traffic are the most suitable.

In [24], a data fusion system is presented, which aggregates information about the Internet traffic collected by different sensors and combines it using the Dempster-Shafer Theory of Evidence. In the DS framework one can state hypotheses, define membership, belief, plausibility and doubt functions regarding these hypotheses and eventually use a rule to combine all evidence and obtain a single conclusion. The proposed system is tested on a university network where information is collected with a Snort plugin and MIB entries.

Fuzzy techniques have also been proposed for DoS detection. In [25], an Adaptive Neuro-Fuzzy Inference System is used together with the Fuzzy C-Means Clustering Algorithm to detect DoS attacks; the authors test the method by performing experiments on the DARPA/KDD99 dataset. In the detection scheme presented in [26], the contents of the incoming packets are analysed with a probe detection system of fuzzy cognitive maps and a black list of IP addresses is constructed accordingly. In another fuzzy-based technique [27], the detection process uses cross-correlation functions between incoming and outgoing traffic as inputs to a fuzzy classifier. The authors discuss the resulting tradeoff between the accuracy of the detector and the increase in the computational demands when opting for higher dimension in the fuzzy classification.

In [28], three computational intelligence techniques for DoS detection, namely support vector machines, multivariate adaptive regression splines and linear genetic programs, are compared in a multi-agent setting. The same authors address also the feature selection problem with feature ranking algorithms in [29] and continue this work in [30], where they discuss alternative approaches, including Frequent Pattern tree mining, classification and regression trees, and TreeNet.

A vital issue in using learning-based detection is the selection of the set of input features that will provide useful and significant information about the incoming traffic. Despite the existence of a number of relevant papers, there is still no consensus on a standardised set of features.

4.2. Applying Statistical Signal Analysis

Internet traffic has statistical properties which can be used to detect DoS traffic. For example, Normal Internet traffic is considered to be long-range dependent (LRD) or equivalently self-similar [31]. A time series is said to be LRD when its autocorrelation function $r_{xx}(\tau)$ is not summable, that is if $\int_0^\infty r_{xx}(\tau)d\tau = +\infty$. This LRD property can be best established by evaluating the Hurst parameter H defined from the autocorrelation function $r_{xx} \sim c\tau^{2H-2}$ as τ becomes very large, where c is a positive constant. The higher the value of H the more self-similar the traffic. In [32], the self-similarity property of Internet traffic is used to identify DoS attacks. The authors use the packet number or packet size as the input feature and evaluate the Hurst parameter H by statistical techniques. In their approach, the variance of H in consecutive time intervals is calculated and if there is a doubling of the variance, it is decided that a DoS attack is in progress.

In [33], the entropy is used instead of the Hurst parameter, to evaluate the randomness of the incoming traffic. This is combined with the chi-square statistic, which is a measure of the statistical significance and rough estimate of confidence in DoS detection.

In another statistical approach [34, 35], Internet traffic is modelled as Fractional Gaussian Noise (FGN), where it is reported that FGN can be used to approximate autocorrelation functions (ACF) of different types of traffic with the same order of modelling accuracy. Their detection scheme is based on the autocorrelation functions of the incoming traffic: the abnormal traffic is the sum of the number of bytes in packets in normal traffic and the attack traffic and the norm of the differences of the autocorrelation functions is used to signal DoS. The authors use simulation of a specific case study to verify that FGN can be used in real-time detection despite existing research showing the limitation of FGN in modelling the ACF of real-time traffic [36].

The Cumulative Sum (CUSUM) algorithm is a change point detection algorithm, which is applicable for detecting sharp changes in variables, and has been used widely in DoS detection. It is a lihelihood-ratio based method which signals abrupt changes from one valid hypothesis to another. CUSUM is first proposed for DoS detection in [37], which uses as variables the TCP flags of the incoming traffic. The performance of the CUSUM approach for SYN flood attacks is evaluated in [38], in terms of the detection probability, false alarm ratio, and detection delay. The same concept of CUSUM-based detection is explored in [39], where it is applied in a distributed framework, in autonomous and logically divided parts of a network.

In [40] an early-bird system of traffic anomaly detection scheme is proposed. This consists of a data centre, a traffic anomaly detection module and an event correlation module. The raw variables collected at the data engines, namely the packet count variable of each protocol, the packet count variables of SYN and FIN flags, ICMP unreachable messages and the packet length count variable of packet payload, are aggregated at the data centre. Then the traffic anomaly detection module uses Statistical Prediction Theory to detect anomalies based on the data provided by the data centre. The traffic is observed for a long period of time and is classified into periodic and non-periodic parts, for each using a different method to predict anomalies, which are then evaluated with predefined pattern profiles.

It is often assumed that DoS packets must be highly correlated, contrary to legitimate traffic which is is some sense random. For this reason, the authors of [41] suggest using Kolmogorov complexity metrics for DoS detection. If $K(x)$ is the Kolmogorov metric for the incoming packets and the complexity of a concatenated string XY is smaller than the sum of the complexities of each string $K(XY) \leq K(X) + K(Y)$, then this implies a high correlation in the traffic. In practice, the authors use estimates for the complexity metrics and compute the complexity differentials $[K(x_1) + K(x_2) + ... + K(x_n)] - K(x_1x_2.....x_n)$, which indicate legitimate traffic when they yield a value close to zero. An improvement upon this work is presented in [42], where a different Kolmogorov metric estimate is proposed, this time measuring the correlation between the first and second halves of the strings. The corresponding DoS detection algorithm evaluates iteratively the fluctuations of the Kolmogorov complexity differentials and signals an attack when the fluctuations are increased.

The authors of [43] have recently proposed a more proactive detection method, which concentrates on the changes that occur in the network traffic during the developing phases of the attack. They take into account the various steps of launching an attack, from the recruitment of the different tiers of compromised computers to their simultaneous attack. Since these recruitment processes generate a lot of messages in the network, the authors use entropy of source and destination addresses together with rates of packet types as their main statistical data. They separate each phase of the attack with a clustering algorithm and identify detection pre-cursors. Although it is not easy to evaluate the specific method, since the authors use a relatively outdated DARPA dataset, their suggestion that an attack can be detected at its earliest stages is promising.

Several experiments have shown that the energy distribution of normal traffic is stationary. During a DoS attack however, the traffic behaviour appears to change significantly, as well as the energy distribution variance. For this reason, detection methods may use wavelet transform analysis to extract information about the energy content of the packets [44]. Traffic is considered to be normal if the variation in the energy distribution is smaller than a predetermined threshold.

A network traffic burst detection algorithm based on the continuous wavelet transform is presented in [45]. With this method, flat bursts in the network traffic are considered to belong to three classes (long bursts, short bursts and one-point bursts), and a continuous wavelet transform algorithm is used for their real-time identification.

Recently, wavelet approximation and system identification theory have been combined for anomaly detection in [46]. However, DoS is handled as one of five general categories of intrusions, with only two input features being directly relevant. In [47], the use of correlation of destination IP addresses, port numbers and the number of flows. The authors employ wavelet transforms to study the address and port number correlation over several timescales. Since, the input features refer to the outgoing traffic at an egress router, the approach is suitable for an edge network, where the attack can be detected near its sources.

Generally, wavelet based methods are able to reach detection decisions in relatively short time, but need significant computational resources for this. They use sliding sampling windows and time step increments, the choice of which determine the algorithm's performance. A small window may be inadequate for the calculation of the energy distribution variation, while a large window increases the computational needs. As is explained in [44], an additional consideration that the wavelet methods introduce is the boundary effect that can exist in wavelet analysis.

4.3. The use of Multi-Agent systems

Another important direction of DoS detection research is distributed detection in a multi-agent framework. Such an example is the Source IP monitoring scheme [48], where new legitimate source IP addresses appearing in the traffic are collected in an IP Address Database during the off-line training phase. The database is used together with data gathered from the incoming traffic to decide about possible DoS attacks in the detection and learning phases. The CUSUM method is used to extract information about the change in the number of new IP addresses, yielding a variable that represents the change and a local threshold for each agent. All agents apply this scheme and if any of them suspects an attack, it broadcasts this information to the other agents. Since the selection of the broadcast threshold is important, a gradient-based learning technique is used to select an optimal value that minimises both the communication overhead and the confirmation delay. More multi-agent approaches can be found in [49], which uses concepts from the biological immune systems and in [50], which employs a Black Board Architecture and a firewall to detect DoS attacks and to form a black list of IP addresses. In general, the multi-agent approaches are more accurate at detecting highly distributed DoS attacks, but are inevitably slower compared to a centralised system simply because of the delay that inter-agent communication introduces.

4.4. Conclusions on DoS Detection

A wide variety of DoS attack detection methods have been suggested in the literature, usually based on symbolic analysis of the traffic packets and in particular of IP addresses and other significant packet content. Other approaches are based on the timing characteristics of the packets streams. All require or assume some representation of what is a normal traffic stream as opposed to a DoS stream. Also, many of the techniques require on-line tuning or a learning phase that is used to create patterns, data or statistics to compare with presumed attacks. Although each of these approaches offers very interesting ideas and insights into DoS attacks, there is not enough work so far that offers a comparison of them under the same or similar conditions. Comparisons become very difficult when one considers that there is a significant variation even in the types of attacks that the different detection schemes are trying to address. Thus, with the vast differences in the experimental setups and simulation datasets that researchers have used, we feel that an evaluation and accuracy comparison of the numerous contributions included in this survey is likely to be very difficult, but is worth pursuing. Furthermore, some of the studies we have surveyed concentrate solely on detection of DoS, while there has been a significant amount of work on extending intrusion detection techniques to DoS and developing general detection systems to work for any kind of attack including DoS. However, the fact that DoS attacks use traffic that in isolation is harmless and becomes harmful when concentrated against a target, differentiates them from other types of network security attacks.

This survey indicates that an overall DoS detection mechanism will have to pragmatically combine different detectors in order to provide robust and effective detection, especially as attack methods and schemes evolve and are improved. Initial steps in this direction can be found in some of the work that we have surveyed. In [20, 18], the outputs of intelligent decision units are combined to detect a DoS attack, while in [19], information about the data content of packets, and the past and present connection statistics are fused. In [24], information measured from different sensors is combined using the Dempster-Shafer Theory of evidence, and in [22] multiple features are fused using a random neural network [51] for DoS detection.

4.5. Classification of traffic by investigating the source's validity

DoS detection initiates the defence of a network and provides useful information, but does not address the core issue of classification between normal and DoS traffic. Classification is usually done with passive or active dedicated validity tests. Some *passive tests* which can be used regardless of the types of traffic include the following:

- *Is the source of the packet a known "loyal client"?* Although we take for granted that DoS attackers will spoof their addresses, this does not mean that the source IP address of a packet is not useful information. For example, if User-X, who has a static IP address, visits her favourite news web-site every day at 9am, stays for a while and consumes a reasonable amount of bandwidth, there is no reason for this web-site's detection mechanism to suspect that her packets are potentially harmful, unless her behaviour is somehow dramatically different from the usual one. Thus, in times of congestion, even if there is no service differentiation, User-X should not be blocked in favour of a User-Y, who has not been recognised as a regular client and consumes a lot of bandwidth.
- *Did the source of the packet first appear before or after the detection of the attack?*

  Since DoS attacks are almost exclusively distributed, the distributed aspect can make it easier to detect their existence. Attack flows do not all traverse the Internet through the same paths, so they reach the victim destination at different times, which results in a gradual increase of the incoming traffic. This ramp-up behaviour was initially proposed as a means to tell whether an attack is distributed or single-source [52], but can also be an indication that a DDoS attack has been initiated. A longer ramp-up time will also be associated with a greater number of spoofed source IP addresses, which means that the IP addresses which arrive after the DDoS attack and until it reaches its peak are more likely to be illegitimate.
- *Is the client honouring her QoS agreements?*

  In Self-Aware QoS-driven network environments [53] in which clients may specify that they belong to a specific type, or request a certain level of QoS, the degree with which a QoS agreement is honoured by the client is a strong indication of her validity. Both DoS attackers and misbehaving clients will fail such a test.
- *Does the time-to-live (TTL) field in an IP packet agree with the value that can be inferred from the packet's apparent source address?*

  This test refers to Hop Count Filtering (HCF), an idea described in [54] which exploits the fact that although the attacker can forge any field in the IP header, he/she cannot falsify the number of hops a packet needs to reach its destination starting from its source address. A simple algorithm infers the number of hops traversed (from the packet's TTL field) and compares it to the value that can be inferred for the source, which is stored in a relevant table. If the two values are significantly different, this is a clear indication of IP spoofing and it is a good reason to treat this source's packets as illegitimate.

These and other passive tests have the advantage of being relatively light-weight, but are not sufficient to achieve accurate classification on their own. More accuracy inevitably requires more specialisation.

In [5], a set of anomaly-based classification criteria for flows and connections are proposed. For instance, a TCP or an ICMP flow may be classified as an attack flow if its packet ratio exceeds a threshold. For the UDP protocol, a normal-flow model is proposed to be a set of thresholds based on the upper bound of allowed connections per destination, the lower bound of allowed packets per connection, and the maximum allowed sending rate per connection. The classification of connections is also done based on limits of the connections' allowed packet ratios and sending rates.

Other recent work [55] uses the Bayesian concept of the conditional legitimate probability (CLP) as the basis for a packet filtering scheme. Traffic characteristics during an attack are compared with previously measured legitimate traffic characteristics, and the CLP provides an indication of the legitimacy of suspected packets. An extension to reduce complexity and enhance performance is presented in [56]. However, as with all profile-based, and particularly Bayesian profile-based DoS approaches, the greatest challenge is not the fine-tuning of the defence mechanism but acquiring dependable traffic profiles.

Another option for validity tests is to use criteria based on the type of service. For example, a network which offers Voice-over-IP (VoIP) services should include specific classification criteria based additionally on the specifics of VoIP traffic behaviour. Then, it becomes a matter of having specialised knowledge, in this case of VoIP traffic behaviour, both for the defence system and the attacker who tries to emulate it [57].

The limits and thresholds used by these passive tests can be set by using the network administrator's experience, or with an automatic learning process using data collected from ongoing observation of the nodes.

Active tests try to interact with suspected attack traffic sources to test their legitimacy. The first question that one has to answer when an attack is suspected is whether it is a real attack or just unusually high legitimate traffic. That is because sudden ramp-up behaviour of traffic also occurs during "flash crowds", where there is a sharp increase in the number of legitimate visitors of a website due to some significant event [58]. DoS attackers have recently started exploiting this fact by abandoning full-strength bandwidth floods in favour of attacks that evade detection by imitating the signature characteristics of flash crowds. In response to this, detection mechanisms may try to distinguish between flash crowds and attacks by exploiting their fundamental difference. Flash crowd flows are generated by human users, while DDoS flows are generated by compromised computers. Thus, one can potentially use Reverse Turing tests, such as CAPTCHAs to tell the difference [59]. A human can easily tell the sequence of letters that appear in a CAPTCHA's image, while computers usually cannot (Figure 3).

FIGURE 3. Example of a graphical CAPTCHA (a Completely Automated Public Turing Test to Tell Computers and Humans Apart)

However, issuing a graphical Turing test and expecting an answer requires a connection to be established between the attacker and the web-server, thus rendering the authentication mechanism itself a potential DoS target. The authors of [60] address this issue and suggest minor modifications to the TCP protocol to overcome it. They also argue that it is not the actual answer to the test but the behaviour of the client that matters. A human would solve the puzzle either immediately or after reloading the page a few times. A computer would probably continue requesting the web page. However, as in all arms races, it is a matter of time for a countermeasure to appear. For example, dedicated applications built by Computer Vision researchers achieve up to 92% success in solving commonly used types of CAPTCHAs [61]. Advances in Artificial Intelligence along with simple craftiness limit their value, while they are also a major impediment to users whose vision impairment. Still, although CAPTCHAs cannot be the sole method of classification, we do agree with the principle that DoS attacks can be more readily detected if we can distinguish between human and computer generated traffic.

Several methods have been proposed for actively challenging the clients' legitimacy, such as the work presented in [62] and the system called Netbouncer [63], which is representative of this category. Netbouncer keeps a list of authorised users (beyond suspicion), while the rest undergo a series of tests, divided into packet-based, flow-based and application and session-oriented, such as CAPTCHAs. Various QoS techniques are utilised to assure fair sharing of the resources by the traffic of the legitimate clients, while this legitimacy expires after a certain interval and needs to be challenged again.

A similar concept is explored in puzzle-based defence solutions. In a connection-oriented test for DoS attacks, the clients are asked to solve a little cryptographic puzzle before their connection request is authorised. The puzzle may take a little time to solve, while the defending server can rapidly verify the result. This slows down the attacker, but does not guarantee that it will suffice, since overwhelming the puzzle-generation process is still possible if the attacker's rate is sufficiently high. Examples of such approaches can be found in [64], [65], [66], and most recently in [67], where a puzzle-based DoS defense architecture is proposed. It operates on multiple layers and includes puzzle auctions for end-to-end protection and congestion puzzles for IP-layer protection.

Recently, Khattab et al. have proposed the live baiting approach [68], which uses group-testing to discover the defective members in a population of users. Since the group-testing theory requires an a priori estimate of the number of defective members, the authors present a probing method for the initial estimate and an adaptive technique to correct the estimate in real-time. They also present a technique to detect users that manipulate their test results.

Although it is too early to evaluate the effectiveness of this approach, especially with the limited NS-2 simulation results presented, the live baiting approach addresses some of the limitations of earlier work on classification, especially in terms of scalability.

The active test solutions that we have presented in this section often achieve impressive levels of detection accuracy. However, they involve some form of communication with the attacker and additional processing, and as a result can become DoS vessels themselves.

4.6. Conclusions on DoS Classification

The wide variety of defence approaches that were presented in this survey exhibit several similarities. As with DoS detection, classification is also performed by measuring features of the incoming traffic and comparing them either to a normal profile in anomaly-based or to a DoS profile in signature-based methods. These features may be actual statistical features measured in real-time or simple observations acquired by actively testing the users of the network and asking them to prove their legitimacy. Anomaly-based methods are less accurate, but apply to a broader range of attacks, while signature-based methods are more dependable, but only for the attacks that they have been designed to detect and counter. Although both are used, academic research tends to prefer anomaly-based classification methods, since it is far easier to keep a signature of the legitimate users' normal traffic for one's network than the signatures of a range of known attack types, which can never be complete.

5. RESPONDING TO AN ATTACK

The raison d'etre of a DoS defence mechanism is to either perform or facilitate the response against the attack, which ideally means for the network to return to its normal operating condition in terms of delays, packet losses, legitimate user connections etc. In most cases, DoS response is largely based on a classification process, such as the ones presented in Section 4.5 and others that we describe here together with their corresponding response mechanism. We will investigate separately DoS response in conventional non-adaptive networks, such as the Internet, and in autonomic networks that self-adapt to improve their users' service [69].

5.1. DoS response in conventional networks

One of the most influential works on DoS response is presented in [70], which introduces a generic system for Aggregate-based Congestion Control (ACC) that learns a congestion signature and identifies the small number of aggregates responsible for congestion. During times of sustained high congestion, the ACC system tries to find the congestion signature using the latest packet drop history. The authors propose a destination-based identification algorithm, which first draws out a list of high-bandwidth 32-bit addresses based on the drop history or a random sample and then clusters these addresses into 24-bit prefixes. For each of the clusters, it tries to find a longer prefix that still contains all the dropped packets, since a longer prefix characterises a congestion signature better and does not punish a large category of traffic. Although this algorithm is simple to implement, it results in unfairness for the legitimate traffic to the congested destination. A more accurate and flexible identification algorithm is needed to maintain the fairness between friendly and misbehaving aggregates. Once the router knows the congestion signature, it then filters the bad traffic according to this signature. Furthermore, a *pushback* scheme [71] is given to let the router ask its adjacent routers to filter the bad traffic at an earlier stage. This is a mechanism which adds functionality to the routers to detect and preferentially drop packets that probably belong to an attack. Upstream routers are also notified to drop such packets so that their resources are used to route only legitimate traffic. This is an effective scheme for various types of DDoS attacks, but its success heavily depends on the congestion signature, which is not always sufficiently accurate.

Another of the first significant approaches for proactive defence against DoS is Secure Overlay Services (SOS) [72], which is geared towards emergency communications. The architecture of SOS is constructed using a combination of overlay tunnelling, routing via consistent hashing, and filtering. It reduces the probability of successful attacks by performing intensive filtering near protected network edges introducing randomness and anonymity. The former helps push the attack point perimeter into the core of the network, where high-speed routers can handle the volume of attack traffic, while the latter makes it difficult to target nodes along the path to a specific SOS-protected destination. The goal of SOS is to route only the authenticated users' traffic to the server and drop everything else. The clients are authenticated at the overlay entrance and they use the overlay network to reach the server. Only a small set of source addresses are approved to reach the server, while all other traffic is heavily filtered out. The main advantage of SOS is that it can be applied over existing IP infrastructure and can guarantee to some extent that in times of crisis an authenticated user will have access to the victim server. However, SOS is more difficult to deploy in a fully public network, since the clients must be aware of the overlay network and use it to access the victim. Also, it does not offer protection for the incoming links of the filtering router in front of the client, which can be quite easily overwhelmed by sheer volumes of DoS traffic. The SOS approach is generalised by Mayday [73], in which overlay networks and lightweight packet filtering are combined. The overlay nodes perform client authentication and protocol verification, and then relay the requests to a server, which is protected from the outside with simple packet filtering rules.

Pricing techniques have also been suggested for protection against DoS attacks. Dynamic Resource Pricing [74] is a distributed gateway architecture and a payment protocol that imposes dynamically changing prices on both network, server, and information resources. In this way it manages to push the cost of initiating service requests back to the requesting clients, which theoretically should at least slow down the attacker.

In [75], a defence method based on regulation of resource consumption is presented. The authors use a Linux based prototype to show that traditional QoS rate-based regulation combined with their proposed window-based regulation of aggregated resources at the network layer can mitigate the impact of DOS attacks on end servers. Traffic regulation policies are enforced across traffic classes according to the resource usage of packets or flows and is done at an aggregate level and not the individual flow level.

In [76], it is proposed that the ISPs carry the packets of the victim's "VIP" clients in a privileged class of service, protected from congestion, whether malicious or not, while all non-VIP traffic is considered as low-priority and can be dropped in the case of an attack. The approach is simple but can prove very useful for transaction-based websites, such as e-commerce ones. Quite similar, in the sense that its users are also considered as "VIP" or not, proactive server roaming is a novel proposal where an active server changes its location within a pool of physical servers to defend itself against unpredictable or untraceable attacks [77]. Again, only known legitimate users are explicitly informed by the roamer of its IP address and are able to follow the active server as it roams. The same authors have incorporated this idea of roaming to the concept of honeypots. Honeypots are nodes which attempt to appear as attractive targets for attackers, but provide no service to legitimate users; they only exist to capture and analyse attack traffic, and they do not receive any legitimate traffic. Since honeypots can be avoided by sophisticated attacks, in [78] they propose roaming honeypots, a mechanism that allows the locations of honeypots to be unpredictable, continuously changing, and disguised within a server pool. So, in this scheme a continuously changing subset of the servers remain active and provide service, while the rest are acting as honeypots.

Another contribution to DoS response is DEFCOM [79], which suggests that the current paradigm of designing isolated defence systems should be abandoned. DEFCOM is a distributed framework that enables the exchange of information and services between existing defence nodes. For example, since attack detection is best done near the victim, while response is most effective at the source of the attack, each node should be specialised in a different aspect of the defence. Defence nodes must be able to communicate and must support at least the following messages: *Attack alerts* (generated from the Alert Generators to the rest of the network), *Rate-Limit requests* (the rate-limit requests should be sent upstream), *Resource requests* (each node should be able to issue a resource request to its downstream neighbours), and *Traffic Classification* (classifier nodes must communicate with their downstream neighbours to ensure that the bulk of legitimate traffic will not be dropped). A similar distributed approach to DoS response is explored in [80], with an architecture that coordinates the countermeasures using multi-cast, annotated topology information and blind detection techniques. The main element of the architecture is the cossack watchdog, which is software located at edge networks, able to monitor traffic, detect attacks and communicate with other watchdogs to filter the traffic collectively.

A particularly interesting branch of DoS research is that of low-rate attacks, also known as shrew attacks, where a well orchestrated periodic burst can achieve disruption for which a flood-based attack would need volumes of traffic. These shrew attacks, first presented in [81], exploit the fixed minimum TCP RTO property. Some related work has appeared since, including detection of such attacks at the edge routers [82] and a study for the related effect of buffer sizes [83]. Detection is also achieved with a hypothesis testing scheme, when the autocorrelation functions and power spectrum densities of the traffic are obtained [84].

Also dealing with advanced DoS attacks, DDoS Shield is a recently developed system that combines a suspicion assignment mechanism with a scheduler [85]. It is designed specifically for application layer attacks that evade normal defences by being non-intrusive and protocol-compliant. The paper explores the earlier concepts of attack probability and proportional defence in a more rigourous manner. In practice, each session is tagged with a continuous value of suspicion, and there are two scheduling policies that use suspicion as their control primitive.

5.2. DoS response in Self-Aware Networks

Denial of service attacks harm Self-Aware Networks (SAN) [53] in the same way they harm conventional networks, only to a different degree for the various network resources, with the factor of differentiation being the dynamic routing [86]. In a conventional network, the attack paths can be several but remain constant, which results into complete overwhelming of specific nodes or specific links on those static attack paths. A legitimate flow which does not use any of these paths would be relatively unaffected by the attack, while the rest would suffer complete outages. In a SAN, such as the Cognitive Packet Network (CPN) [87], the routing protocol attempts to accommodate all traffic by dynamically changing the paths. As a result, the attack is distributed in the whole network and affects the quality of service of the legitimate flows in a more balanced manner. All flows may be affected, but fewer suffer complete outage. In CPN, this is achieved with the use of "smart packets" that constantly explore the network, "dumb packets" that carry the data and acknowledgement packets that help monitor the condition of the network. Each user specifies its QoS requirements in the form of a QoS "goal" and at each CPN node a local reinforcement learning (RL) algorithm based on online measurements elicit a decision from the node as to the next hop to travel to. The arrival of a smart packet triggers the execution of the RL algorithm. Each router stores a specific Random Neural Network (RNN) [51] for each QoS class and for each active source-destination pair. Each RNN node represents the decision to choose a given output link for a smart packet and has as many neurons as the possible outgoing links. Decisions are taken by selecting the output link for which the corresponding neuron is the most excited [88]. The performance of the CPN protocol has been extensively evaluated both in normal operating conditions [89] and under failures [90]. An admission control system makes sure that the quality of service for the traffic of new users will satisfy their requirements and that existing users will not be affected [91, 92].

The natural adaptability and resilience of CPN in adverse network conditions has been further improved with the DoS-specific defence system presented in [93, 94] and later extended in [95]. Each network node informed of a DoS attack contributes to the defence by examining incoming packets for deviations from normal behaviour. Packets undergo a collection of anomaly-based tests which may differ for each type of traffic. Nodes that take part in the defence prioritise traffic according to the results of the tests. Additionally, the upstream routers are instructed to rate-limit any traffic directed towards the victim node. With this two-fold protection framework, packets with higher probability of being both valid and harmless are offered preferential service, while packets that have been marginally classified as invalid will still receive service if there is available bandwidth, so as to minimise the collateral damage inflicted by false detection. Packets that have been identified as harmful are either dropped or delayed by being assigned low priority. The authors also discuss various simplifications of the scheme.

5.3. Conclusions on DoS Response

Most of the existing literature uses different forms of redirection or dropping of the packets that the classification methods identify as illegitimate. Redirecting the offending packets to a controlled part of the network, not only decreases the congestion in the victim network, but also provides the opportunity to analyse the attack. However, such a safe and controlled environment usually does not exist and is difficult to build within a network. The family of response methods based on dropping packets is much more common, with packet filtering that depends on classification rules being the predominant technique. In practice, the large variety of DoS attack and response methods present the need for a mapping between the two. The authors of [6] propose such a framework that chooses a defence mechanism for a given attack mechanism, but results of their method have not been presented yet. Self-aware networks are generally more resilient to network threats, such as worms [90] and Denial of Service attacks, but may still need threat-specific defence systems to maintain the quality of service of their users.

6. IDENTIFYING THE TRUE SOURCE OF AN ATTACK

Although identifying the source of an attack is not enough to respond against it, it can be part of a broader repertoire of countermeasures. One of the first measures proposed for limiting attacks from spoofed source addresses is Ingress Filtering, a technique based on configuring routers to drop arriving packets with IP addresses deemed to be outside a predetermined "acceptable" range [96]. In essence, packets are not allowed to leave a border network if their source address does not belong to this border network. The more widely deployed the better the protection that Ingress Filtering will offer. Of course, it requires that the receiving router has sufficient power and sufficient knowledge to examine the source address of each packet and distinguish between valid and invalid ones, while attackers can still overcome it by choosing legitimate addresses at random. Variations of Ingress Filtering are often used as a lightweight first step to identify part of the attacking traffic.

It has been suggested that the actual IP address of an attacking agent can in fact be inferred by "IP traceback" [97, 98, 99, 100, 101, 102], where it is proposed either to monitor the flows traversing the network and mark packets probabilistically or to attempt to infer the attacking flows' paths based on their impact on the network. Traceback needs to be done with minimal cost in time and storage and minimal false alarms, while respecting the privacy of the contents of the audited packets and not requiring interactive operational support from Internet Service Providers. A fairly accurate and widely deployed traceback mechanism can also act as a deterrent for attackers that are aware of its existence [103]. However, despite the elegant ideas and interesting implementations described in the literature, current traceback mechanisms are not effective deterrents. In most cases, not only they cannot be deployed widely enough, but they also suffer from a common inherent weakness, which is that fake traceback messages created by attackers and included into the attack flows can help them hide their origin. However, even a rough estimate of the sources of attack packets usually does help in DDoS defence. IP traceback can also be applied "post-mortem", after the attack is over, to help protect against future attacks. The following is a more detailed description of the main traceback proposals in the literature.

The first important work on determining the path that a packet traverses over the Internet in the context of a DoS attack [104] proposes to systematically flood links which potentially belong to attack paths, observe the variations in the received packet rates, and create a map of the routes from the victim to every network, then start with the closest router and apply a brief burst of load to each link attached to it. If the loaded link belongs to the attack path, the induced load will perturb the attacking stream. As the authors admit they assume that they can map the paths from the victim to all possible networks, that routes are all symmetric, can be discovered and are fairly consistent, and that the attacking packet stream arrives from a single source. However, this interesting approach to the traceback problem has very limited use in today's networks; not to mention that flooding networks with traffic to protect them against flooding may be inappropriate. It is usually referred to as controlled flooding or more generally as link-testing.

In [97], a Probabilistic Packet Marking (PPM) scheme is proposed for the first time. The scheme encodes partial attack path information and includes it in IP packets as they arrive in routers. The authors describe some basic marking algorithms, such as appending each node's address to the end of the packet, node sampling, where each router chooses to write its address in the node field with some low probability, and edge sampling of participating routers. The victim receives all marking information and reconstructs probabilistically the attack path. In [105], it is shown that the reconstruction of the attack paths based on the original Probabilistic Packet Marking techniques is computationally extremely intensive even for only a few attack sources. Despite its weaknesses, PPM is the basis of several related proposals.

A similar in concept Internet Draft [106] suggests the use of a new type of ICMP messages specifically generated by routers for traceback purposes. For one in every 20,000 packets, the router copies the contents of the packet and information about adjacent routers into these customised ICMP messages. Again the victim should potentially be able to reconstruct the approximate attack path. In [107], however, a DoS technique is presented, which can render such ICMP traceback not only useless, but possibly harmful for the network.

Some of the limitations of PPM are addressed in [98], where two marking schemes for IP traceback, the advanced and the authenticated one, are proposed. Since almost all Internet routes have less than 32 hops, an 11-bit hash of the IP address and a 5-bit hop count can be encoded in the 16 bits of a packet's fragmentation field. The authors describe two different hashing functions with which the order of the routers can be inferred when the attack path is reconstructed at the victim node.

Others follow a different approach to probabilistic packet marking [100]; they reframe the IP traceback problem as a polynomial reconstruction problem and use algebraic techniques originally developed for coding theory and machine learning, and present a series of schemes, all based on the principle of reconstructing a polynomial in a prime field, to encode the probabilistic path information inside the IP header with the very few bits that are available or non-crucial. A strength of this approach is that it is expected to benefit automatically from the parallel ongoing research in its underlying mathematical techniques. However, there have not been reported any relevant improvements yet.

The majority of the traceback solutions are targeted at high-speed attacks, where a large number of packets are sent during the detection phase. In [99, 101], on the other hand, a scheme is proposed to trace even single-packet attacks, such as those which exploit vulnerabilities in the packet processing of TCP/IP stack implementations ("Teardrop" is one of them, as presented in Section 2). This "Source Path Isolation Engine" (SPIE) system, supports tracing by storing a few bits of unique information, essentially packet digests, for a period of time as the packets traverse the Internet. To minimise the storage needs for the digest tables, they use Bloom filters, which are space-efficient data structures with independent uniform hash functions. The software implementations of SPIE perform adequately for slow to medium speed routers. To achieve better results for faster routers, in [108] a hardware implementation as a processing unit inserted into the router, or as stand-alone connected to the router through an external interface, is described. In our opinion, the implementation cost and computation overhead of this approach appear to be very high for a technique oriented against an uncommon family of DoS attacks. In [109], a new approach is proposed, which makes use of packet marking to further improves scalability, while still being able to trace a single IP packet. Both storage time and access time are reduced in comparison to SPIE.

Single-packet traceback has also been recently explored in [110], where again packet marking is used to avoid storing information in the routers, while a generalised bloom filter provides the basis of the traceback mechanism. The authors of [111] follow a similar packet marking technique, but also introduce an approach that is based on an AS-level overlay network, so as to allow incremental deployment. Their proposal does not require a priori knowledge of the network topology and they show how it combines with the BGP protocol for practical deployment.

7. MATHEMATICAL MODELLING OF DENIAL OF SERVICE

In most of the existing literature on DoS defence systems, the classification and response mechanisms consist of checking each incoming packet or flow for their legitimacy according to some predefined rules and then dropping those that do not abide by them. There are important differences between the various proposals, but most do belong to this same family of mechanisms. For this reason, the authors of [95] have developed a queuing network model to predict the impact that both attack and defence have on the performance of a network. With this approach, the total traffic rate λ_i arriving externally to node i is composed of two parts: $\lambda_i = \sum_{\mathbf{n}} \lambda^n_{i,\mathbf{n}} + \sum_{\mathbf{d}} \lambda^d_{i,\mathbf{d}}$, where $\lambda^n_{i,\mathbf{n}}$ is the "normal" or benign incoming traffic rate which belongs to normal flow $\mathbf{n}$, and $\lambda^d_{i,\mathbf{d}}$ is the arrival rate of DoS packets belonging to attack flow $\mathbf{d}$. Each node is modelled by a single server queue with service time s_i representing both the time it takes to process the packet in the node and the actual transmission time. The traffic intensity parameter ρ_i is then: $\rho_i = s_i(\sum_{\mathbf{n}} I^n_{i,\mathbf{n}}(1 - f_{i,\mathbf{n}}) + \sum_{\mathbf{d}} I^d_{i,\mathbf{d}}(1 - d_{i,\mathbf{d}}))$, where for node i, $I^n_{i,\mathbf{n}}$ is the arriving traffic rate of the normal flow $\mathbf{n}$, and $I^d_{i,\mathbf{d}}$ is the arriving traffic rate of a DoS flow $\mathbf{d}$. Assuming that any traffic that is correctly or mistakenly thought to be DoS traffic is dropped at the input of the node, and since the traffic which effectively enters a node has been filtered in this manner, the traffic equations for the system become:

$$I^n_{n_j,\mathbf{n}} = \lambda^n_{n_1,\mathbf{n}} \prod_{l=0}^{j-1} ((1 - L_{n_l})(1 - f_{n_l,\mathbf{n}}))$$

$$I^d_{d_j,\mathbf{d}} = \lambda^d_{d_1,\mathbf{d}} \prod_{l=0}^{j-1} ((1 - L_{d_l})(1 - d_{d_l,\mathbf{d}})),$$

where f_{nl} is the probability that a normal packet is mistakenly dropped by the defence mechanism at the l-th node on the route of normal flow $\mathbf{n}$, and d_{dl} is the probability that an attack packet will be correctly dropped at the l-th node on the route of attack flow $\mathbf{d}$.

The specific mathematical model is particularly suitable for flood attacks, as it considers the disruption in the victim network to be the result of congestion in the nodes and links. In [95], it is validated by comparison with the results of simulation in NS-2 and experiments in a networking testbed. Recently, a new queuing model has been proposed for the analysis of networks under DoS attacks [112]. The model assumes normal packets and attack packets arriving according to Poisson processes and that the service times are general but different for the two types of requests, while there is a maximum number of connections that can be served at each time.

In [113], a specialised mathematical model for router throttling is formulated. The system parameters include feedback delays, the hysteresis control limits, and the traffic source rates, and each router is modelled as a source of traffic for a server S which is to be protected. The server generates a throttle signal $\Phi(t)$ as a function of the aggregate traffic workload and the hysteresis control limits L_S and U_S:

$$\Phi(t) = \begin{cases} -1 & \text{if } \sum_{i=1}^{N} T_i^*(t) \geq U_S \\ 1 & \text{if } \sum_{i=1}^{N} T_i^*(t) \leq L_S \\ 0 & \text{otherwise} \end{cases}$$

A differential equation is used for the throttle rate:

$$\frac{dr_i(t)}{dt} = \min(1 - r_i(t), \beta_i)\mathbf{1}(\Phi(t - r_i) = 1)) \\ - \frac{r_i(t)}{2}\mathbf{1}(\Phi(t - r_i) = 1))$$

where $\beta_i > 0$ is the incremental step size for router i. The forwarding traffic route at router i is then given by: $T_i^*(t) = T_i(t)r_i(t)$.

This control-theoretic model is designed to explore how specific system parameters can affect the performance and stability of router throttling as a DoS response strategy, and also to study various multi-source flow control problems.

Mathematical models such as the two summarised in this section can be particularly useful not only to understand the dynamics of DoS attacks, but also to predict the performance of various defence methods before implementing them. Mathematical modelling can also be used to fine-tune existing defence methods and suggest optimal configurations that can respond to specific threats. For the time being this practice is rarely followed among DoS researchers, but we expect it to become more widespread as accuracy and optimality in defence are needed to respond to the latest attack trends.

8. CONCLUSIONS AND SUGGESTIONS FOR FUTURE DOS RESEARCH

Since most networks that employ some sort of DoS protection are now in a position to defend against the less sophisticated types of attack, researchers can direct their efforts specifically towards the newer attack trends. Important recent trends to address include the attacks of extreme scale generated by bot armies and the pinpoint attacks that try to maximise the victim's financial losses for reasons of cyber-extortion or illegal competition. However, even if today's types of attack are to be addressed at some point, researchers should not to be "one step behind" but should look for ways to end this arms race. For example, since the attractiveness of DoS lies in the convenience with which an attacker can inflict disproportionately significant damage, research should be directed not only towards decreasing the damage of an attack, as is done today, but also towards increasing the difficulty of launching one.

Another consideration that the inherent elusiveness of DoS attacks raises is how particularly difficult it is to prosecute offenders. This could be facilitated by introducing the concept of collection of evidence as part of a DoS defence mechanism. In fact, a first attempt for such a system can be seen in [114], which aims to aid the prosecution of attackers by assigning fingerprints that identify repeated attacks by the same hosts. Along the same lines of legal prosecution, however, there is an even more important issue that remains. In the definition of DoS that we propose at the very beginning of this paper, what distinguishes DoS from simple network congestion or flash crowds is the intention of the offender. This question of intention behind a network outage would be of interest for both legal prosecution and actual DoS defence, but has not yet been addressed in technical terms.

Of course, before such ambitious goals are to be achieved, current research needs to address its two major weaknesses, namely the lack of standards of evaluation for the defence methods and the scarce information on modern types of attacks. Launching real attacks against real networks with real legitimate users is impractical, which leaves the researchers with less dependable data sets, such as outdated traffic traces and simulated traffic. A pragmatic solution to these problems consists of organising a close cooperation of the research community with organisations that are frequently under attack, such as e-commerce and betting websites. Accurate and up-to-date data sets will help distinguish the best defence approaches in an unbiased manner and will prompt further research and improvements in existing mechanisms.

Our literature survey indicates that there have not been many completely new ideas in recent years. Recent work has mainly improved upon the the first-wave concepts produced right after the 2000 “Mafiaboy” incident. Many of these pioneering suggestions have worked particularly well, but usually for limited parts of what a complete DoS defence architecture needs to contain. Thus, future research will have to face the major challenge of effectively combining the various proposals for DoS detection, classification and response methods to complement each other’s strengths and weaknesses in a common architecture.

REFERENCES

[1] Gilgor, V. (1983) A note on the Denial-of-Service Problem. *Proceedings of Symposium on Security and Privacy (SP’83)*, Oakland, CA, USA, 25-27 April, pp. 139–149. IEEE Computer Society, Washington, DC, USA.

[2] Morris, R. T. (1985) A weakness in the 4.2BSD Unix TCP/IP software. Computer Science Technical Report 117. AT&T Bell Labs, Murray Hills, NJ, USA.

[3] Shipley, G. (1999) ISS RealSecure pushes past newer IDS players. *Network Computing*, **10**, 95–111.

[4] Tuncer, T. and Tatar, Y. (2008) Detection SYN flooding attacks using fuzzy logic. *Proceedings of International Conference on Information Security and Assurance (ISA’08)*, Washington, DC, USA, 24-26 April, pp. 321–325. IEEE Computer Society, New York, NY, USA.

[5] Mirkovic, J. and Reiher, P. (2005) D-WARD: A source-end defense against flooding denial-of-service attacks. *IEEE Transactions on Dependable and Secure Computing*, **2**, 216–232.

[6] Asosheh, A., Dr. and Ramezani, N. (2008) A comprehensive taxonomy of DDoS attacks and defense mechanism applying in a smart classification. *WSEAS Transactions on Computers*, **7**, 281–290.

[7] Mirkovic, J., Dietrich, S., Dittrich, D., and Reiher, P. (2004) *Internet Denial of Service Attack and Defense Mechanisms*, 1st edition. Prentice Hall PTR, New Jersey, USA.

[8] Paxson, V. (2001) An analysis of using reflectors for distributed denial-of-service attacks. *ACM SIGCOMM Computer Communication Review*, **31**, 38–47.

[9] Moore, D., Paxson, V., Savage, S., Shannon, C., Staniford, S., and Weaver, N. (2003) Inside the slammer worm. *IEEE Security and Privacy*, **1**, 33–39.

[10] Hemenway, K. and Calishain, T. (2003) *Spidering Hacks*. O’Reilly & Associates, Inc., Sebastopol, CA, USA.

[11] Goucher, W. (2009) The tipping point. *Computer Fraud & Security*, **1**, 11–13.

[12] Lesk, M. (2007) The new front line: Estonia under cyberassault. *IEEE Security and Privacy*, **5**, 76–79.

[13] Jalili, R., Imani-Mehr, F., Amini, M., and Shahriari, H.-R. (2005) Detection of distributed denial of service attacks using statistical pre-processor and unsupervised neural networks. *Lecture Notes in Computer Science*, **3439**, 192–203.

[14] Gavrilis, D. and Dermatas, E. (2005) Real-time detection of distributed denial-of-service attacks using RBF networks and statistical features. *Computer Networks and ISDN Systems*, **48**, 235–245.

[15] Gavrilis, D., Tsoulos, I., and Dermatas, E. (2004) Feature selection for robust detection of distributed denial-of-service attacks using genetic algorithms. *Lecture Notes in Artificial Intelligence*, **3025**, 276–281.

[16] Ng, W., Chan, A., Yeung, D., and Tsang, E. (2006) Construction of high precision RBFNN with low false alarm for detecting flooding based denial of service attacks using stochastic sensitivity measure. *Lecture Notes in Artificial Intelligence*, **3930**, 851–860.

[17] Chan, A., Ng, W., D.S, Yeung, and Tsang, E. (2005) Multiple classifier system with feature grouping for intrusion detection: Mutual information approach. *Lecture Notes in Artificial Intelligence*, **3683**, 141–148.

[18] Chan, A., Yeung, D., Tsang, E., and Ng, W. (2006) Empirical study on fusion methods using ensemble of RBFNN for network intrusion detection. *Lecture Notes in Artificial Intelligence*, **3930**, 682–690.

[19] Giacinto, G., Roli, F., and Didaci, L. (2003) Fusion of multiple classifiers for intrusion detection in computer networks. *Pattern Recognition Letters*, **24**, 1795–1803.

[20] Mukkamala, S., Sung, A. H., and Abraham, A. (2005) Intrusion detection using an ensemble of intelligent paradigms. *Journal of Network and Computer Applications*, **28**, 167–182.

[21] Noh, S., Lee, C., Choi, K., and Jung, G. (2003) Detecting distributed denial of service (DDoS) attacks through inductive learning. *Lecture Notes in Computer Science*, **2690**, 286–295.

[22] Öke, G. and Loukas, G. (2007) A denial of service detector based on maximum likelihood detection and the random neural network. *Computer Journal*, **50**, 717–727.

[23] Kim, M., Na, H., Chae, K., Bang, H., and Na, J. (2004) A combined data mining approach for DDoS attack detection. *Lecture Notes in Computer Science*, **3090**, 943–950.

[24] Siaterlis, C. and Maglaris, B. (2004) Towards multisensor data fusion for DoS detection. *Proceedings of symposium on Applied computing (SAC'04)*, Nicosia, Cyprus, 14-17 March, pp. 439–446. ACM, New York, NY, USA.

[25] He, H., Luo, X., and Liu, B. (2005) Detecting anomalous network traffic with combined fuzzy-based approaches. *Lecture Notes in Computer Science*, **3645**, 433–442.

[26] Lee, S., Kim, Y., Lee, B., Kang, S., and Youn, C. (2005) A probe detection model using the analysis of the fuzzy cognitive maps. *Lecture Notes in Computer Science*, **3480**, 320–328.

[27] Wei, W., Dong, Y., Lu, D., and Jin, G. (2006) Combining cross-correlation and fuzzy classification to detect distributed denial-of-service attacks. *Lecture Notes in Computer Science*, **3994**, 57–64.

[28] Mukkamala, S. and Sung, A. H. (2004) Computational intelligent techniques for detecting denial of service attacks. *Proceedings of conference on Innovations in Applied Artificial Intelligence (IEA/AIE'04)*, Ottawa, Canada, 17-20 May, pp. 616–624. Springer Verlag, Berlin, Germany.

[29] Sung, A. and Mukkamala, S. (2004) The feature selection and intrusion detection problems. *Lecture Notes in Computer Science*, **3321**, 468–482.

[30] Mukkamala, S., Xu, D., and Sung, A. (2006) Intrusion detection based on behaviour mining and machine learning techniques. *Lecture Notes in Artificial Intelligence*, **4031**, 619–628.

[31] Uhlig, S. and Bonaventure, O. (2001) Understanding the long-term self-similarity of internet traffic. *Lecture Notes in Computer Science*, **2156**, 286–298.

[32] Xiang, Y., Lin, Y., Lei, W., and Huang, S. (2004) Detecting DDOS attack based on network self-similarity. *IEE Proceedings Communications*, **151**, 292–295.

[33] Feinstein, L., Schnackenberg, D., Balupari, R., and Kindred, D. (2003) Statistical approaches to DDoS attack detection and response. *Proceedings of Information Survivability Conference and Exposition (DISCEX-III)*, Washington, DC, 22-24 April, pp. 303–314. DARPA, Arlington, VA, USA.

[34] Li, M., Chi, C.-H., and Long, D. (2004) Fractional gaussian noise: A tool for characterizing traffic for detection purpose. *Lecture Notes in Computer Science*, **3309**, 94–103.

[35] Li, M. (2004) An approach to reliably identifying signs of DDOS flood attacks based on LRD traffic pattern recognition. *Computers and Security*, **23**, 549–558.

[36] Tsybakov, B. and Georganas, N. (1998) Self-similar processes in communications networks. *IEEE Transactions on Information Theory*, **44**, 1713–1725.

[37] Wang, H., Zhang, D., and Shin, K. (2002) Detecting SYN flooding attacks. *Proceedings of INFOCOM'02*, New York, NY, USA, 23-27 June, pp. 1530–1539. IEEE Communications Society, New York, NY, USA.

[38] Siris, V. and Papagalou, F. (2004) Application of anomaly detection algorithms for detecting syn flooding attacks. *Proceedings of GLOBECOM'04*, Dallas, TX, USA, 29 November - 3 December, pp. 2050–2054.

[39] Leu, F. and Yang, W. (2005) Intrusion detection with CUSUM for TCP-based DDoS. *Lecture Notes in Computer Science*, **3823**, 1255–1264.

[40] Gu, R., Yan, P., Zou, T., and Guo, C. (2005) An automatic and generic early-bird system for internet backbone based on traffic anomaly detection. *Lecture Notes in Computer Science*, **3420**, 740–748.

[41] Kulkarni, A. and Bush, S. (2006) Detecting distributed denial of service attacks using kolmogorov complexity metrics. *Journal of Network and Systems Management*, **14(1)**, 69–80.

[42] Furuya, F., Matsuzaki, T., and Matsuura, K. (2005) Detection of unknown DoS attacks by Kolmogorov-complexity fluctuation. *Lecture Notes in Computer Science*, **3822**, 395–406.

[43] Lee, K., Kim, J., Kwon, K. H., Han, Y., and Kim, S. (2008) DDoS attack detection method using cluster analysis. *Expert Systems with Applications*, **34**, 1659–1665.

[44] Li, L. and Lee, G. (2005) DDoS attack detection and wavelets. *Telecommunication Systems*, **28(3)**, 435–451.

[45] Yang, X., Liu, Y., Zeng, M., and Shi, Y. (2004) A novel DDoS attack detecting algorithm based on the continuous wavelet transform. *Proceedings of Advanced Workshop on Content Computing (AWCC'04)*, ZhenJiang, JiangSu, China, 15-17 November, pp. 173–181.

[46] Lu, W. and Ghorbani, A. A. (2009) Network anomaly detection based on wavelet analysis. *EURASIP Journal On Advances In Signal Processing*, **2009**, 1–16.

[47] Kim, S. S. and Reddy, A. L. N. (2008) Statistical techniques for detecting traffic anomalies through packet header data. *IEEE/ACM Transactions on Networking*, **16**, 562–575.

[48] Peng, T., Leckie, C., and Ramamohanarao, K. (2003) Detecting distributed denial of service attacks by sharing distributed belief. *Lecture Notes in Computer Science*, **2727**, 214–225.

[49] Cetnarowicz, K. and Rojek, G. (2004) Behavior based detection of unfavourable resource. *Lecture Notes in Computer Science*, **3038**, 607–614.

[50] Seo, H. S. and Cho, T. H. (2002) Modeling and simulation for detecting a distributed denial of service attack. *Proceedings of Australian Joint Conference on Artificial Intelligence (AI'02)*, 16-21 September.

[51] Gelenbe, E. (1993) Learning in the recurrent random neural network. *Neural Computation*, **5**, 154–164.

[52] Hussain, A., Heidemann, J., and Papadopoulos, C. (2003) A framework for classifying denial of service attacks. *Proceedings of conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM'03)*, Karlsruhe, Germany, 25-29 August, pp. 99–110. ACM, New York, NY, USA.

[53] Gelenbe, E., Lent, R., and Nunez, A. (2004) Self-aware networks and QoS. *Proceedings of the IEEE*, **92**, 1478–1489.

[54] Jing, S., Wang, H., and Shin, K. (2003) Hop-count filtering an effective defense against spoofed traffic. *Proceedings of International Conference on Computer and Communications Security (CCS'03)*, Washington, DC, USA, 27-30 October, pp. 30–41. ACM, New York, NY, USA.

[55] Kim, Y., Lau, W., Chuah, M., and Chao, H. (2006) PacketScore: A statistics-based packet filtering scheme against distributed denial-of-service attacks. *IEEE Transactions on Dependable and Secure Computing*, **3(2)**, 141–155.

[56] Ayres, P., Sun, H., Chao, H., and Lau, W. (2006) ALPi: a DDoS defence system for high-speed networks. *IEEE Journal of Selected Areas in Communications*, **24(10)**, 1864–1876.

[57] Sisalem, D., Kuthan, J., and Ehlert, S. (2006) Denial of service attacks targeting a sip voip infrastructure: attack scenarios and prevention mechanisms. *IEEE Network*, **20**, 26–31.

[58] Jung, J., Krishnamurthy, B., and Rabinovich, M. (2002) Flash crowds and denial of service attacks: characterization and implications for CDNs and web sites. *Proceedings of conference on World Wide Web (WWW'02)*, Honolulu, Hawaii, USA, 7-11 May, pp. 553–561. ACM, New York, NY, USA.

[59] Morein, W. G., Stavrou, A., Cook, D. L., Keromytis, A. D., Misra, V., and Rubenstein, D. (2003) Using graphic turing tests to counter automated DDoS attacks against web servers. *Proceedings of Conference on Computer and Communications Security (CCS'03)*, Washington, DC, USA, 27-30 October, pp. 8–19. ACM, New York, NY, USA.

[60] Kandula, S., Katabi, D., Jacob, M., and Berger, A. (2005) Botz-4-sale: surviving organized DDoS attacks that mimic flash crowds. *Proceedings of Symposium on Networked Systems Design & Implementation (NDSI'05)*, Houston, TX, USA, 4-6 April, pp. 287–300. USENIX Association, Berkeley, CA, USA.

[61] Mori, G. and Malik, J. (2003) Recognizing objects in adversarial clutter - breaking a visual captcha. *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR'03)*, Madison, Wisconsin, USA, 16-22 June, pp. 134–141.

[62] Gao, Z. and Ansari, N. (2006) Differentiating malicious DDoS attack traffic from normal TCP flows by proactive tests. *Communication Letters*, **10(11)**, 793–795.

[63] Thomas, R., Mark, B., Johnson, T., and Croall, J. (2003) Netbouncer: client-legitimacy-based high-performance DDoS filtering. *Proceedings of Information Survivability Conference and Exposition (DISCEX-III)*, Washington, DC, USA, 22-24 April, pp. 14–25. DARPA, Arlington, VA, USA.

[64] Juels, A. and Brainard, J. (1999) Client puzzles: A cryptographic countermeasure against connection depletion attacks. *Proceedings of Network and Distributed System Security Symposium (NDSS'99)*, San Diego, CA, USA, February, pp. 151–165. The Internet Society, Washington, DC, USA.

[65] Aura, T., Nikander, P., and Leiwo, J. (2001) DOS-resistant authentication with client puzzles. *Lecture Notes in Computer Science*, **2133**, 170–177.

[66] Wang, X. and Reiter, M. K. (2003) Defending against denial-of-service attacks with puzzle auctions. *Proceedings of Symposium on Security and Privacy (SP'03)*, Washington, DC, USA, pp. 78–92. IEEE Computer Society, New York, NY, USA.

[67] Wang, X. and Reiter, M. K. (2008) A multi-layer framework for puzzle-based denial-of-service defense. *International Journal of Information Security*, **7**, 243–263.

[68] Khattab, S., Gobriel, S., Melhem, R., and Mosse, D. (2008) Live baiting for service-level dos attackers. *Proceedings of INFOCOM'08*, Dublin, Ireland, 13-19 April, pp. 171–175. IEEE Communications Society, New York, NY, USA.

[69] Dobson, S., Denazis, S., Fernandez, A., Gaiti, D., Gelenbe, E., Massacri, F., Nixon, P., Saffre, F., Schmidt, N., and Zambonelli, F. (2006) A survey of autonomic communications. *ACM Trans. Auton. Adapt. Syst.*, **1**, 223–259.

[70] Mahajan, R., Bellovin, S. M., Floyd, S., Ioannidis, J., Paxson, V., and Shenker, S. (2002) Controlling high bandwidth aggregates in the network. *ACM SIGCOMM Computer Communication Review*, **32**, 62–73.

[71] Ioannidis, J. and Bellovin, S. (2002) Implementing pushback: Router-based defense against DDoS attacks. *Proceedings of Network and Distributed System Security Symposium (NDSS'02)*, San Diego, California, USA, 6-8 February, pp. 1–12. The Internet Society, Washington, DC, USA.

[72] Keromytis, A. D., Misra, V., and Rubenstein, D. (2002) SOS: secure overlay services. *Proceedings of conference on Applications, technologies, architectures, and protocols for computer communications SIGCOMM'02*, Pittsburgh, Pennsylvania, USA, 19-23 August, pp. 61–72. ACM, New York, NY, USA.

[73] Andersen, D. (2004) Mayday: Distributed DoS filtering for internet services. *ACM SIGCOMM Computer Communication Review*, **34**, 39–44.

[74] Mankins, D., Krishnan, R., Boyd, C., Zao, J., and Frentz, M. (2001) Mitigating distributed denial of service attacks with dynamic resource pricing. *Proceedings of Annual Computer Security Applications Conference (ACSAC'01)*, Washington, DC, USA, 10-14 December, pp. 411–421. IEEE Computer Society, New York, NY, USA.

[75] Garg, A. and Reddy, A. N. (2004) Mitigation of DoS attacks through qos regulation. *Microprocessors and Microsystems*, **28**, 521 – 530.

[76] Brustoloni, J. (2002) Protecting electronic commerce from distributed denial-of-service attacks. *Proceedings of conference on World Wide Web (WWW'02)*, Honolulu, Hawaii, USA, 7-11 May, pp. 553–561. ACM, New York, NY, USA.

[77] Khattab, S., Sangpachatanaruk, C., Znati, T., Melhern, R., and Mosse, D. (2003) Proactive server roaming for mitigating denial-of-service attacks. *Proceedings of Conference on Information Technology Research and Education (ITRE'03)*, 10-13 August, pp. 1–5.

[78] Khattab, S., Sangpachatanaruk, C., Moss, D., Melhem, R., and Znati, T. (2004) Roaming honeypots for mitigating service-level denial-of-service attacks. *Proceedings of International Conference on Distributed Computing Systems (ICDCS'04)*, Washington, DC, USA, pp. 328–337. IEEE Computer Society, New York, NY, USA.

[79] Mirkovic, J., Reiher, P., and Robinson, M. (2003) Forming alliance for DDoS defense. *Proceedings of New Security Paradigms Workshop*, Centro Stefano Francini, Ascona, Switzerland.

[80] Papadopoulos, C., Lindell, R., Mehringer, J., Hussain, A., and Govindan, R. (2003) COSSACK: Coordinated suppression of simultaneous attacks. *Proceedings of Information Survivability Conference and Exposition (DISCEX-III)*, Washington, DC, USA, 22-24 April, pp. 2–13. DARPA, Arlington, VA, USA.

[81] Kuzmanovic, A. and Knightly, E. W. (2006) Low-rate tcp-targeted denial of service attacks and counter strategies. *IEEE/ACM Transactions on Networking*, **14**, 683–696.

[82] Shevtekar, A., Anantharam, K., and Ansari, N. (2005) Low-rate TCP denial-of-service attack detection at edge routers. *IEEE Communications Letters*, **9**, 363–365.

[83] Sarat, S. and Terzis, A. (2005) On the effect of router buffer sizes on low-rate denial of service attacks. *Proceedings of International Conference on Computer Communications and Networks (ICCCN'05)*, San Diego, California, 17-19 October. IEEE Computer Society, Washington, DC, USA.

[84] Chen, Y. and Hwang, K. (2006) Collaborative detection and filtering of shrew DDoS attacks using spectral analysis. *Journal of Parallel and Distributed Computing*, **66**, 1137–1151.

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

[86] Gelenbe, E. (2005) Users and services in intelligent networks. *Lecture Notes in Computer Science*, **3837**, 30–45.

[87] Gelenbe, E., Lent, R., and Xu, Z. (2001) Measurement and performance of a cognitive packet network. *Journal of Computer Networks*, **37**, 691–701.

[88] Sakellari, G. (2009) The cognitive packet network: A survey. *The Computer Journal: Special Issue on Random Neural Networks*, to appear.

[89] Gelenbe, E., Lent, R., Montuori, A., and Xu, Z. (2002) Cognitive packet networks: Qos and performance. *Proceedings of the IEEE International Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS'02)*, Ft. Worth, TX, USA, October, pp. 3–12. IEEE, New York, NY, USA.

[90] Sakellari, G. and Gelenbe, E. (2009) Adaptive resilience of the cognitive packet network the presence of network worms. *Proceedings of the NATO Symposium on C3I for Crisis, Emergency and Consequence Management*, Bucharest, Romania, 11-12 May. NATO Research & Technology Organisation.

[91] Sakellari, G., Gelenbe, E., and D' Arienzo, M. (2008) Admission of QoS aware users a smart network. *ACM Transactions on Autonomous and Adaptive Systems,*, **3**, 4: 1–28.

[92] Sakellari, G. and Gelenbe, E. (2008) A multiple criteria, measurement-based admission control mechanism for self-aware networks. *Proceedings of the third International Conference on Communications and Networking China (CHINACOM'08)*, Hangzhou, China, 25-27 August, pp. 1060–1064. IEEE, New York, NY, USA.

[93] Gelenbe, E., Gellman, M. and Loukas, G. (2004) Defending networks against denial of service attacks. *Proceedings of Conference on Optics/Photonics in Security and Defence (SPIE'04)*, London, UK, 25 October, vol. 5611, pp. 233–243. SPIE, Bellingham WA, USA.

[94] Gelenbe, E., Gellman, M. and Loukas, G. (2005) An Autonomic Approach to Denial of Service Defence. *Proceedings of WoWMoM Workshop on Autonomic Communications and Computing (WoWMoM/ACC'05)*, Taormina, Italy, 13-16 June, pp. 537–541. IEEE Computer Society, New York, NY, USA.

[95] Gelenbe, E. and Loukas, G. (2007) Self-aware approach to denial of service defence. *Computer Networks*, **51(5)**, 1299–1314.

[96] Ferguson, P. and Senie, D. (2000) Network ingress filtering: Defeating denial of service attacks which employ IP source address spoofing. RFC 2827. Internet Engineering Task Force, USA.

[97] Savage, S., Wetherall, D., Karlin, A., and Anderson, T. (2000) Practical network support for IP traceback. *Proceedings of conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM'00)*, Stockholm, Sweden, 28 August - 1 September, pp. 295–306. ACM, New York, NY, USA.

[98] Song, D. and Perrig, A. (2001) Advanced and authenticated marking schemes for IP traceback. *Proceedings of INFOCOM'01*, Anchorage, Alaska, USA, 22-26 April, pp. 878–886. IEEE Communications Society, New York, NY, USA.

[99] Snoeren, A. C. (2001) Hash-based IP traceback. *Proceedings of conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM'01)*, San Diego, California, United States, 27-31 August, pp. 3–14. ACM, New York, NY, USA.

[100] Dean, D., Franklin, M., and Stubblefield, A. (2002) An algebraic approach to IP traceback. *ACM Transactions on Information and System Security*, **5**, 119–137.

[101] Snoeren, A. C., Partridge, C., Sanchez, L. A., Jones, C. E., Tchakountio, F., Schwartz, B., Kent, S. T., and Strayer, W. T. (2002) Single-packet IP traceback. *IEEE/ACM Transactions on Networking*, **10**, 721–734.

[102] Sung, M. and Xu, J. (2002) IP traceback-based intelligent packet filtering: A novel technique for defending against internet DDoS attacks. *Proceedings of International Conference on Network Protocols (ICNP'02)*, Paris, France, 12-15 November, pp. 302–311. IEEE Computer Society, Washington, DC, USA.

[103] Aljifri, H. (2003) IP traceback: A new denial-of-service deterrent? *IEEE Security and Privacy*, **1**, 24–31.

[104] Burch, H. (2000) Tracing anonymous packets to their approximate source. *Proceedings of conference on System administration (LISA'00)*, New Orleans, Louisiana, USA, 3-8 December, pp. 319–328. USENIX Association, Berkeley, CA, USA.

[105] Song, D. and Perrig, A. (2000) Advanced and authenticated marking schemes for IP traceback. Technical Report UCB/CSD-00-1107. University of California, Berkeley, USA.

[106] Bellovin, S. (2000). ICMP traceback messages. Internet Draft: draft-bellovin-itrace-00.txt.

[107] Tupakula, U. and Varadharajan, V. (2006) Analysis of traceback techniques. In Safavi-Naini, R., Steketee, C., and Susilo, W. (eds.), *Proceedings of Australasian Information Security Workshop AISW-NetSec'06*, Hobart, Australia, pp. 115–124.

[108] Sanchez, L., Milliken, W., Snoeren, A., Tchakountio, F., Jones, C., Kent, S., Partridge, C., and Strayer, W. (2001) Hardware support for a hash-based IP traceback. *Proceedings of Information Survivability Conference & Exposition II (DISCEX-II)*, Anaheim, California, 12-14 June, pp. 146–152. DARPA, Arlington, VA, USA.

[109] Gong, C. and Sarac, K. (2008) A more practical approach for single-packet IP traceback using packet logging and marking. *IEEE Transactions on Parallel and Distributed Systems*, **19**, 1310–1324.

[110] Laufer, R. P., Velloso, P. B., Cunha, D., Moraes, I. M., Bicudo, M. D., Moreira, M. D., and Duarte, O. (2007) Towards stateless single-packet IP traceback. *Proceedings of Conference on Local Computer Networks (LCN'07)*, Washington, DC, USA, pp. 548–555. IEEE Computer Society, New York, NY, USA.

[111] Castelucio, A., Salles, R., and Ziviani, A. (2009) An AS-level overlay network for IP traceback. *Network: Special Issue on Recent Developments in Network Intrusion Detection*, **23**, 36–41.

[112] Aissani, A. (2008) Queueing analysis for networks under DoS attack. *Proceedings of international conference on Computational Science and its Applications (ICCSA'08), Part II*, Perugia, Italy, 30 June - 3 July, pp. 500–513. Springer-Verlag, Berlin/Heidelberg, Germany.

[113] Yau, D., Lui, J., Liang, F., and Yam, Y. (2005) Defending against distributed denial-of-service attacks with max-min fair server-centric router throttles. *IEEE/ACM Transactions on Networking*, **13**, 29–42.

[114] Hussain, A., Heidemann, J., and Papadopoulos, C. (2006) Identification of repeated denial of service attacks. *Proceedings of INFOCOM'06*, Barcelona, Spain, 23-29 April, pp. 1–15. IEEE Communications Society, New York, NY, USA.