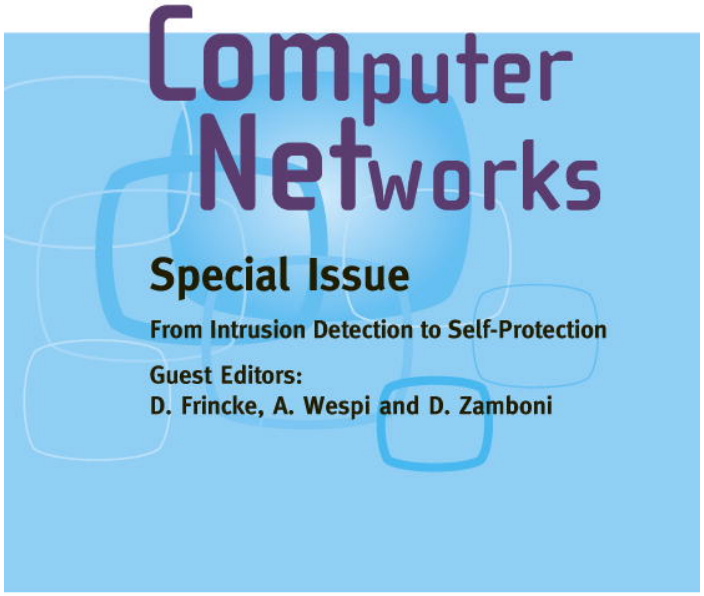
Volume 51 Issue 5 11 April 2007 ISSN 1389-1286



Available online at www.sciencedirect.com
ScienceDirect
Computer Networks 51 (2007) 1299–1314
Computer Networks
www.elsevier.com/locate/comnet

A self-aware approach to denial of service defence ☆

Erol Gelenbe *, George Loukas

Intelligent Systems and Networks Group, Department of Electrical and Electronic Engineering, Imperial College, London SW7 2BT, United Kingdom

Available online 18 October 2006

Abstract

Denial of service (DoS) attacks are a serious security threat for Internet based organisations, and effective methods are needed to detect an attack and defend the nodes being attacked in real time. We propose an autonomic approach to DoS defence based on detecting DoS flows, and adaptively dropping attacking packets upstream from the node being attacked using trace-back of the attacking flows. Our approach is based on the Cognitive Packet Network infrastructure which uses smart packets to select paths based on Quality of Service. This approach allows paths being used by a flow (including an attacking flow) to be identified, and also helps legitimate flows to find robust paths during an attack. We evaluate the proposed approach using a mathematical model, as well as using experiments in a laboratory test-bed. We then suggest a more sophisticated defence framework based on authenticity tests as part of the detection mechanism, and on assigning priorities to incoming traffic and rate-limiting it on the basis of the outcome of these tests.

Keywords: Denial of service; Self-aware; Cognitive packet network; Internet

1. Introduction

DoS attacks are known to the network research community since the early 1980s; indeed, in his 1985 paper on TCP/IP weaknesses, Morris writes [1] "The weakness in this scheme (the IP protocol) is that the source host itself fills in the IP source host id, and there is no provision to discover the true origin of the packet". A typical generic DoS attack is practically always distributed (DDoS): the attacker takes control of a large number of lightly protected computers (e.g., without firewall and up-to-date antivirus software) and orders them to send simultaneously a large number of packets to a specific target. The attacker exploits the weakness of IP by faking their source IP address ("IP spoofing"). As a result, some routers and links in the vicinity of the target are overwhelmed, and a number of legitimate clients cannot connect to it anymore. Typical targets are the servers of e-commerce web-sites, which can suffer significant financial loss. Other targets may be news web-sites, corporate networks, banks, etc. To cite just two examples, (1) the Arabic and English language web sites of the satellite television network Al-Jazeera suffered two days of Internet disruptions caused by American hackers as

☆ This work is supported by the UK Engineering and Physical Sciences Research Council under Grant EPSRC GR/S52360/01 for the project on "Self-Aware Networks and Quality of Service" at Imperial College London.
* Corresponding author. Tel.: +44 2075946342; fax: +44 2075946274.
E-mail addresses: e.gelenbe@imperial.ac.uk (E. Gelenbe), georgios.loukas@imperial.ac.uk (G. Loukas).

1389-1286/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2006.09.009

revenge for showing pictures of dead and captive American soldiers in Iraq (*Source: BBC, "Hackers cripple Al-Jazeera sites"*, http://news.bbc.co.uk/2/hi/technology/2893993.stm), and (2) a corporate executive in Massachussets was charged with using DoS attacks to cause a total of $2 billion in losses to three of his main competitors (*Source: SecurityFocus – "FBI busts alleged DDoS Mafia"*, http://www.securityfocus.com/news/9411).

After a landmark attack occurred against several major online organisations in one week of February 2000 [7], DoS became an important topic of research. One of the first defensive measures proposed was Ingress Filtering, which is an approach to thwart IP address spoofing by configuring routers to drop arriving packets that arrive with IP addresses which are deemed to be outside a predetermined "acceptable" range [2]. 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. Although not enough on its own, variations of Ingress Filtering can be used as a good lightweight first step to identify part of the attacking traffic. Later, it was suggested that the real IP address could in fact be inferred with a technique called "IP traceback" [6,9,18]. The traceback mechanism uses probabilistic packet marking to allow the victim to identify the network path traversed by attack traffic without requiring interactive operational support from Internet Service Providers (ISPs). However, injecting false traceback messages into the packet stream can help attackers hide their origin, even a rough estimate of the sources of attack packets can help the DDoS defence. IP traceback can also be applied after the attack is over to protect against future attacks; such "post-mortem" techniques [8] can be used to analyse the characteristics of attacking traffic (such as rates and more detailed statistical characteristics) and the slave nodes or entry points that it uses.

Although many intelligent defence techniques and architectures have been proposed, the evolution of the DoS phenomenon implies that new means of attack will arise. Until a few years ago, DoS attacks were used by hackers to knock web pages off-line, usually for revenge, or as a token of power and to demonstrate their programming skills. Now they are considered to be an important weapon in the hands of cyber-criminals and for cyber-warfare. DoS attacks have reportedly been used against Business competitors, for extortion purposes, for political reasons, and even as a form of "legitimate" protest. It is this variety of targets and types of attack that dictate the need for flexible defence systems which can react according to both the attacker's aims and the defender's needs. Thus in this paper we propose and evaluate an autonomic approach, based on network self-observation and adaptive reaction, to defend network nodes against DoS attacks. The techniques that we propose, implement and evaluate, exploit the Cognitive Packet Network architecture introduced in [3] and discussed in [19,21].

1.1. A generic framework for denial of service protection

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 those two. In anomaly-based detection, the system recognises a deviation from the standard behaviour of its clients, while in the latter 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 or it redirects them into a trap for further evaluation and analysis.

Therefore we start with the postulate that we will consider a generic DDoS defence scheme that is based on the following principles:

- The node which is targeted by a DDoS attack (the victim node) has the ability to detect or to be informed about the attack, based either on a local or distributed detection scheme. All nodes upstream, from the victim up to the source(s) of the attack, will also be informed of the ongoing threat.
- The victim node and the informed nodes will react by *dropping packets* which are thought to be part of the attack.
- The attack itself can produce buffer overflows and saturation of network resources such as CPU capacity, due to the inability of the nodes

or routers to handle the resulting heavy packet traffic.
- The detection scheme is always imperfect, so that both false alarms and detection failures are possible. Imperfections are possible both with regard to the detection of the attack as a whole, and the identification of the packets that belong to this attack. Thus, for any packet that flows in the network we need to consider a probability of correct identification as being an attacking packet, and a probability of false alarm, which means that some attacking packets will be missed and some non-attacking packets may be incorrectly dropped.

The classification of packets or flows as being a DoS attack or being valid is probably the weak point of DoS defence techniques. Many classification methods that have been suggested tend to use a set of specialised rules for specific types of traffic. The classification can then be carried out either by passive observation or by actively asking the network's apparent clients to demonstrate their validity.

1.1.1. Passive tests

Some passive tests which can be used for all types of traffic include the following:

- *Is the source of the packet a known "loyal client"?*
  In our approach, we take for granted that DoS attackers will spoof their addresses. However, that 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 his favorite news web-site every day at 9 am, 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 his packets are potentially harmful, except if his behaviour is somehow dramatically different from the usual one. Thus, in times of congestion, even if there is no official service differentiation, User-X should not be blocked in favor of a User-Y, who has not been recognised as a regular client and consumes a lot of bandwidth. To our knowledge current DoS detection mechanisms do not treat preferentially users who may be recognised as being "loyal clients".
- *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. Since attack flows do not all traverse the Internet through the same paths, they reach the victim destination at different times, resulting in a gradual increase of the incoming traffic. This ramp-up behaviour [17] can be a good indication that a DDoS attack is being initiated. A longer ramp-up time will also be associated with a greater number of spoofed source IP addresses. Consequently, it also means that the IP addresses which arrive after the DDoS starts and before it ends are more likely to be illegitimate. Again, early detection is obviously a great advantage to identifying the spoofed set of IP addresses.
- *Is the client honouring his/her QoS agreements?*
  We are particularly interested in QoS-driven network environments in which clients may specify that they are a specific type of customer, or request a certain level of QoS. In case the QoS request is accepted, then the degree with which the agreement will be honoured by the client is a strong indication of his/her validity. DoS attackers and misbehaving clients would both fail this 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 the ingenious idea described in [12] 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. Their very simple algorithm infers the number of hops traversed until then (from the TTL field) and compares them to the value that can be inferred for this source, which is stored in a relevant table. If those two values are significantly different, that is a clear indication of IP spoofing, and is a good reason to treat this source's packets as being illegitimate. In the CPN protocol, the header field *r_len* (route length) in a packet can be used instead of TTL.

These different passive tests have the advantage of being relatively light-weight, but are by themselves not sufficient to achieve accurate classification. More accuracy inevitably requires more specialisation.

In [25] a set of anomaly-based classification criteria for flows and connections is proposed. For instance, a TCP flow may be classified as an attack flow if its packet ratio (TCP_{rto}) is above a set threshold. Similarly, for the ICMP protocol they propose to

use $ICMP_{rto}$ as a detector. 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.

One more option for validity tests is to use criteria based on the type of service. This approach is close to the QoS-driven approach of our ongoing work on CPN. For example, a network which offers Voice-over-IP (VoIP) services should include VoIP-specific classification criteria.

The limits and thresholds used by all these types of tests can be set by using the network administrator's experience, or with an automatic learning process in all or some of the nodes using data collected from on-going observation.

1.1.2. Active tests

The ramp-up behaviour of traffic also occurs during "flash crowds" when there is a sharp increase in the number of legitimate visitors to a web-site based on some significant event, and DoS attackers have recently started exploiting this fact by abandoning full strength bandwidth floods in favor of attacks which escape detection by imitating the signature characteristics of flash crowds. In response to this, detection mechanisms should try to distinguish between flash crowds and attacks after a ramp-up has been detected. Fortunately, flash crowd flows are generated by human users, while DDoS flows are generated by compromised computers, and one can potentially use Reverse Turing tests to tell the difference. Over the last three years, Graphical Turing tests or visual CAPTCHAs, or simply CAPTCHAs (Completely Automated Public Turing Test to Tell Computers and Humans Apart), have been commonly used to block automated requests to web-sites, such as the automated email account registration which has plagued Hotmail and Yahoo in the past. A human can easily tell the sequence of letters that appear in a CAPTCHA's image, while computers usually cannot (Fig. 1), so that CAPTCHAs have been suggested to counter DDoS attacks against web-servers [14].

Fig. 1. This captcha of "smwm" obscures its message from computer interpretation by twisting the letters [taken from http://en.wikipedia.org/wiki/Captcha].

However, issuing a graphical Turing test and expecting an answer requires that a connection be established between the attacker and the web-server, thus rendering the authentication mechanism a potential DoS target. The authors of [24] 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. We agree with these ideas but believe that despite its value, a DoS detection mechanism cannot depend solely on CAPTCHAs. In all arms races it is only a matter of time for a countermeasure to arrive on the scene. Dedicated applications built by Computer Vision researchers achieve up to 92% success in solving commonly used types of CAPTCHAs [16]. Advances in Artificial Intelligence along with simple craftiness limit the value of CAPTCHAs, while visual CAPTCHAs are also a major impediment to computer users whose vision is impaired. Thus, although we are not sure that CAPTCHAs can be the sole method of classification, it is reasonable to suggest that DoS attacks can be more readily detected if we can distinguish between humans and computer generated traffic, along with other techniques that recognise legitimate and DoS flows. Other methods have also been proposed for actively challenging the clients' legitimacy [15]. Some may achieve impressive levels of detection accuracy, but all suffer from the common weakness of being exploitable as DoS vessels themselves, in the same way as simple ACKs are used as DoS vessels in the reflector DoS attack [5]. It is also obvious that the benefit of a greater number of validity tests lies in the accuracy of the classification, while the resulting cost may include greater delay in the decision and more processing overhead.

2. A mathematical evaluation of denial of service protection

Before we address the issue of how we propose to implement a DDoS detection and protection mechanism, let us discuss an approach that we have developed [20] to analyse the impact of DDoS protection on overall network performance based on the probabilities of detection and of false alarm. We assume an abstract network model in which DoS packets are identified with a certain probability

and dropped, while some other valid (i.e., non-DoS) packets are mistakenly identified as DoS flows and also dropped.

The packet network consists of N nodes $1,\ldots,i,\ldots,N$. At any node i, the arriving traffic is the aggregate of several normal (valid or non-DoS) flows, and possibly of several invalid (DoS) flows, where $\mathbf{n}=(n_1,n_2,\ldots,n_j,\ldots,n_{L(\mathbf{n})})$ and $\mathbf{d}=(d_1,d_2,\ldots,d_j,\ldots,d_{L(\mathbf{d})})$ are the paths in a normal and a DoS flow, respectively. $L(\mathbf{n})$ is the path length of flow $\mathbf{n}$, and j is used to denote the position of a generic node inside the path. The total traffic rate λ_i arriving externally to node i is composed of two parts:

$$\lambda_i=\sum_{\mathbf{n}}\lambda^n_{i,\mathbf{n}}+\sum_{\mathbf{d}}\lambda^d_{i,\mathbf{d}}, \tag{1}$$

where $\lambda^n_{i,\mathbf{n}}$ is the "normal" or benign incoming traffic rate which belongs to normal flow n, and $\lambda^d_{i,\mathbf{d}}$ is the arrival rate of DoS packets belonging to flow d.

Any traffic that node i takes to be DoS traffic is dropped at the entrance of the node. Thus, a fraction $f_{i,\mathbf{n}}$ of normal traffic (the probability of false alarms) and a fraction of DoS traffic $d_{i,\mathbf{d}}$ (the probability of correct detection) will be dropped as it arrives to the node. If the node's DoS detection mechanism were perfect we would have $f_{i,\mathbf{n}}=0$ and $d_{i,\mathbf{d}}=1$. Once a packet is admitted into a node, it is queued and then forwarded based on its destination address. We model each node 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

$$\rho_i=s_i\left(\sum_{\mathbf{n}}I^n_{i,\mathbf{n}}(1-f_{i,\mathbf{n}})+\sum_{\mathbf{d}}I^d_{i,\mathbf{d}}(1-d_{i,\mathbf{d}})\right), \tag{2}$$

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 $\mathbf{d}$.

Since DoS attacks will tend to overwhelm the node's packet processing and transmission capability, packets will be lost by the node with probability L_i. We could use different formulas to relate traffic intensity to this probability based on modelling congestion of various type that may occur at the node. We take a simplistic view that this loss probability is due to buffer overflow, and we use loss probability expressions for a finite capacity queueing model [4].

Since 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

$$\begin{aligned} I^n_{n_j,\mathbf{n}}&=\lambda^n_{n_1,\mathbf{n}}\prod_{l=0}^{j-1}((1-L_{n_l})(1-f_{n_l,\mathbf{n}})),\\ I^d_{d_j,\mathbf{d}}&=\lambda^d_{d_1,\mathbf{d}}\prod_{l=0}^{j-1}((1-L_{d_l})(1-d_{d_l,\mathbf{d}})),\end{aligned} \tag{3}$$

where we set $L_{n_0}=L_{d_0}=f_{n_0,\mathbf{n}}=d_{d_0,\mathbf{d}}=0$. These equations express the fact that, at any node, an incoming packet may be dropped due to correct or mistaken identification as a DoS packet, or due to buffer overflow because the node is overloaded, while all packets which enter the buffer queue and are not dropped are eventually routed to the next node on their path or absorbed at the current node if it is itself the destination node. Eq. (3) relate input rates to the nodes to the buffer overflow or loss probabilities, while ρ_i and consequently the buffer overflow probabilities L_i in turn depend on the traffic rates. The solution of (3) is obtained numerically via a non-linear iteration:

- Step 0

$$I^{n,(k=0)}_{i,\mathbf{n}}=\lambda^n_{i,\mathbf{n}}, \tag{4}$$

$$I^{d,(k=0)}_{i,\mathbf{d}}=\lambda^d_{i,\mathbf{d}}. \tag{5}$$

- Step $k>0$

$$\rho^{(k)}_i=s_iI^{(k-1)}_i, \tag{6}$$

$$L^{(k)}_i=\rho_i^{B_i,(k)}\frac{1-\rho^{(k)}_i}{1-\rho_i^{B_i+1,(k)}}, \tag{7}$$

$$I^{n,(k)}_{n_j,\mathbf{n}}=\lambda^n_{n_1,\mathbf{n}}\prod_{l=0}^{j-1}((1-L^{(k)}_{n_l})(1-f_{n_l,\mathbf{n}})), \tag{8}$$

$$I^{d,(k)}_{d_j,\mathbf{d}}=\lambda^d_{d_1,\mathbf{d}}\prod_{l=0}^{j-1}((1-L^{(k)}_{d_l})(1-d_{d_l,\mathbf{d}})), \tag{9}$$

$$I^{(k)}_i=\sum_{\mathbf{n}}I^{n,(k)}_{i,\mathbf{n}}+\sum_{\mathbf{d}}I^{d,(k)}_{i,\mathbf{d}}. \tag{10}$$

The "goodput" or aggregate of the packets that either have reached their destination, or are ready to be forwarded to the next node in their route, is used as a measure of the effectiveness of the DoS

protection scheme, and also of how successful or unsuccessful the DoS attack has been. Thus after the algorithm converges we obtain the goodput $G(i)$ at each node using

$$G(i) = \sum_{\mathbf{n}} I^n_{i,\mathbf{n}}(1 - L_i)(1 - f_{i,\mathbf{n}}). \quad (11)$$

2.1. Numerical example

To illustrate the use of the mathematical model we evaluate the impact of a DoS attack on the network topology shown in Fig. 2. In this example, web-server 0 is being attacked by three DoS flows of 2500 packets/s each, entering the network through nodes 3, 4 and 5, so that the model represents a DDoS attack. Both web-servers receive normal packets by all valid clients at a rate of 100–500 packets/s per client (100 corresponds to very low and 500 to very high load level). We evaluate the impact of the attack and the defence mechanism by considering the goodput or rate of "valid" packets which make it safely to their destination nodes. We investigate the impact of the attack on the goodput at each node under varying load levels and different detection probabilities. We choose an average service time per packet of $s_i = 0.4$ ms and a buffer size of $B_i = 40$ packets at each node. The results presented in Figs. 3 and 4 show that a moderate attack

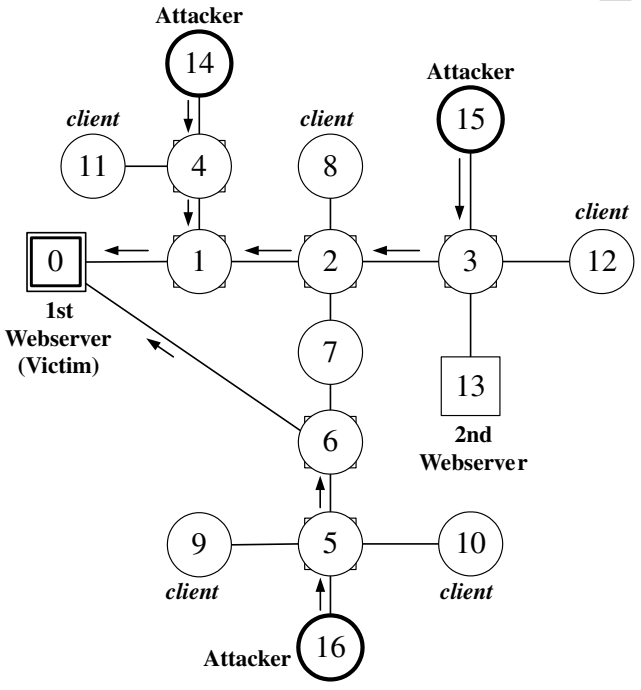


Fig. 2. Attack scenario: 3 attack flows through 3–5 towards web-server 0.

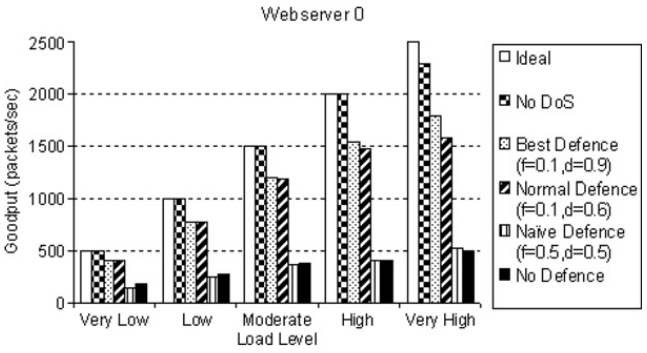


Fig. 3. Mathematical analysis results for web-server 0.

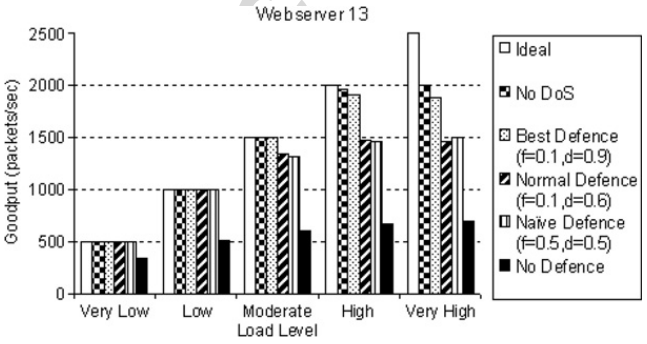


Fig. 4. Mathematical analysis results for web-server 13.

can cause an undefended network's performance to degrade dramatically. For example, at high load level, the victim web-server (0) operates at less than 22% of its ideal capacity, compared to 99% without the attack. Applying a simplistic defence in which we drop half of the packets which are destined to it (Fig. 3, naïve defence, $f = 0.5$, $d = 0.5$), the results do not improve, at least from the victim's perspective. They do improve though for web-server 13 (Fig. 4). So, if web-server 0 were not a crucial part of the infrastructure, but only a decoy or a "honeypot" whose role is to attract the attacking traffic, then that very lightweight naïve defence choice would prove useful. We represent a more sophisticated defence approach as normal defence, for which we arbitrarily choose ($f = 0.1$, $d = 0.6$) as the set of dropping probabilities. The results show significant improvement in both web-servers for all load levels. An even more accurate defence ($f = 0.1$, $d = 0.9$) would of course yield even better results.

2.2. Simulation of a DoS attack

In this section we pursue the preceding discussion using simulations in order to illustrate the behav-

iour of the small network in Fig. 2 operating in the presence of a DoS attack. We carried out the simulations using the NS-2 network simulator [13]. For the sake of comparison we first left the network undefended and then applied a simple defence mechanism with packet drops based on fixed probabilities along the lines of our earlier discussion.

In the simulation each normal flow is composed of UDP traffic at constant rates of 100–500 packets/s depending on the desired load level, while the DoS flows are again UDP traffic at a rate of 2500 packets/s. The packet size we chose is 500 bytes for both normal and DoS flows. The links are full duplex and have a bandwidth of 100 Mbits/s. For all nodes the service time is 0.4 ms and the buffer size is 40 packets. The queues are simulated with the "Drop-Tail" mechanism, which implements simple FIFO scheduling and drop-on-overflow buffer management. Later, in conjunction with the discussion in Fig. 5, the results are presented in comparison to the other three methods, including the experimental analysis presented in the next section.

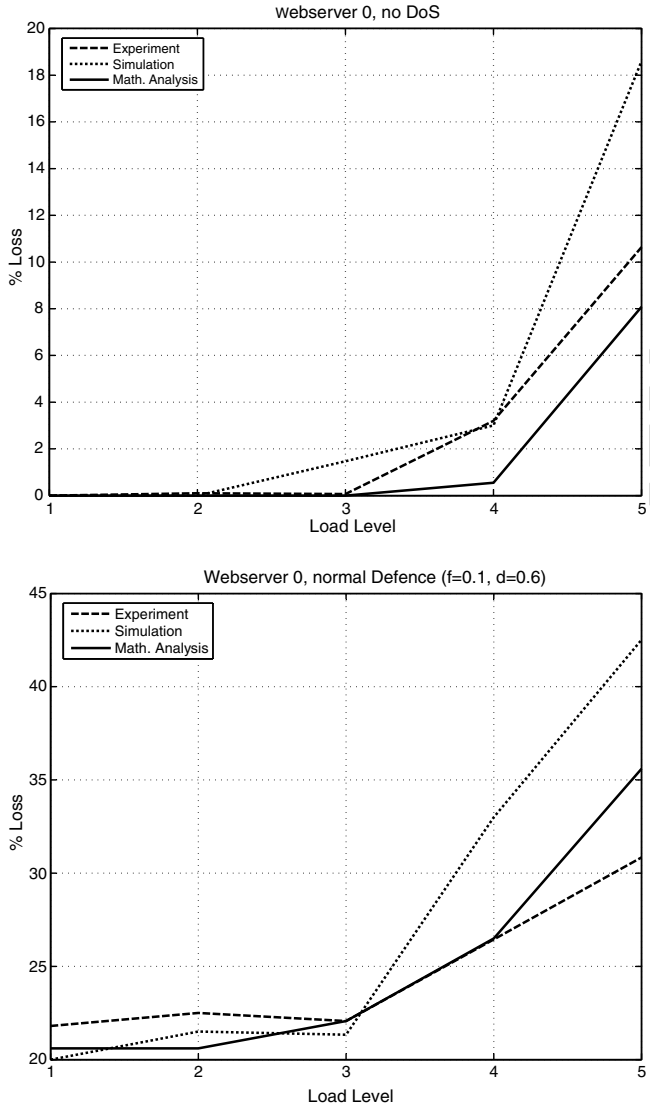


Fig. 5. Loss percentage for different load levels. Apart from a few exceptions the results found by solving the mathematical model numerically, simulating in NS-2, and experimenting on a test-bed, were from very close to identical.

3. Experimental approach using CPN for defence against DDoS attacks

The Cognitive Packet Network (CPN) is a Quality of Service (QoS)-driven routing protocol, in which each flow specifies the QoS metric (e.g., delay, loss, jitter, or other composite metrics) that it wishes to optimise [3,11]. Payload in CPN is carried by source routed "dumb packets" (DPs), while "smart packets" (SPs) and "acknowledgment packets" (ACKs) gather and carry control information which is used for decision making.

In CPN, each flow specifies its QoS requirements in the form of a QoS "goal". SPs associated with each flow constantly explore the network, and obtain routing decisions from network routers based on observed relevant QoS information. SPs store the identities of the nodes they visit, and collect local measurements such as times and loss rates. At each CPN node, the SP uses a local reinforcement learning algorithm based on measurements collected by previous SPs and ACKs, to elicit a decision from the node as to the next hop to travel to. When a SP reaches the destination node of the flow, an ACK packet is generated and returned to the source according to the opposite (destination to source) path traversed by the SP, but from which all node repetitions have been removed by using a right-to-left deletion algorithm to delete the subpaths between identical nodes. When the ACK reaches the source, the forward route, which is the reverse of the route that it used, is stored for subsequent payload or dumb packets (DPs) which will be source-routed to the destination.

Our CPN-based DDoS defence technique exploits the ability of CPN to trace traffic going both down- and up-stream thanks to SPs and ACK packets, so as to facilitate the stifling of malicious traffic. When a CPN node detects a DoS attack, it will use the ACKs to ask all intermediate nodes upstream to drop packets of the incoming flow. Detection can be achieved by allowing any

node to determine for itself two parameters governing bandwidth allocation: the maximum that it is able to receive (BTOT), and the maximum that it is willing to allocate to any particular flow that traverses it (BClient); both are dynamic parameters that may change over time as a function of the conditions at the node, and on the identity and QoS needs of the flows, and they may also vary during the life of a particular flow or connection. This idea can be extended to allowing a node to specify different bandwidth restrictions for flows of different QoS classes.

When a CPN router receives a SP or DP from a flow that it has not already seen before (e.g., with a new source-destination pair, accompanied possibly by a new QoS class), it will send a specific Flow-ACK packet back to the source along the reverse path, and inform the source of its (BClient) allocation. This may occur periodically for each ongoing flow. The node will monitor all of the flows that traverse it, and drop some or all of the packets of any flow that exceeds this allocation. When the allocation is exceeded, the node informs (using ACKs) upstream nodes that packets of this flow should be dropped. Other possible actions could include diverting the flow into a "honeypot", or into a special overlay network used for protection, or it may simply alert a network administrator.

We implemented this approach on a CPN test-bed consisting of 2.4 GHz Pentium-4 PCs configured as shown in Fig. 2. In order to compare the experimental results with those of the simulation and the mathematical analysis, we tried to use model parameter values which mimic the parameters used in the test-bed. Both SPs and Dumb Packet Acknowledgments were disabled and the routes in the network were manually configured to remain static during the entire experiment. To distinguish between valid and DoS traffic we used different QoS protocols. The various defence mechanisms were implemented through the use of configurable drop probabilities which were allowed to be different for different classes of traffic. In order to emulate the value of 0.4 ms service time for all nodes, we used a delay-based FIFO queueing mechanism imposed on each outgoing interface. To mimic the simulator's use of a single buffer per node, for forwarding nodes with two output links (nodes 1–3, and 6) the size of the FIFO buffer was divided in half (20 packets instead of 40 in the test-bed). Each experiment lasted 60 s. The results for both the simulations and the measurement experiments are very similar to those obtained with the mathematical model, and some representative results are summarised in Fig. 5.

3.1. Evaluating autonomic CPN-based defence

The experiments we have described which were conducted on a network test-bed, as well as the NS-2 based simulations, validate the predictions of the mathematical model. However they do not implement the dynamic response that is needed to defend against a DoS or DDoS attack. Thus, in a second set of experiments we tested the CPN-based defence approach for a network application (video streaming) with specific QoS requirements in terms of bandwidth.

The application which is used to illustrate the attack and defence mechanism is a UDP based MPEG1 video stream that is transferred from node 3 to node 30 in the CPN test-bed shown in Fig. 6 (top left). The video received at 30 in the un-attacked network is uncorrupted, as shown in Fig. 6 (top right). Then a DDoS attack which is meant to overwhelm node 30, by saturating it with incoming packets, is launched from nodes 1 and 2. When there is no defence, the attack corrupts the video stream, making it unintelligible as shown in Fig. 6 (bottom left).

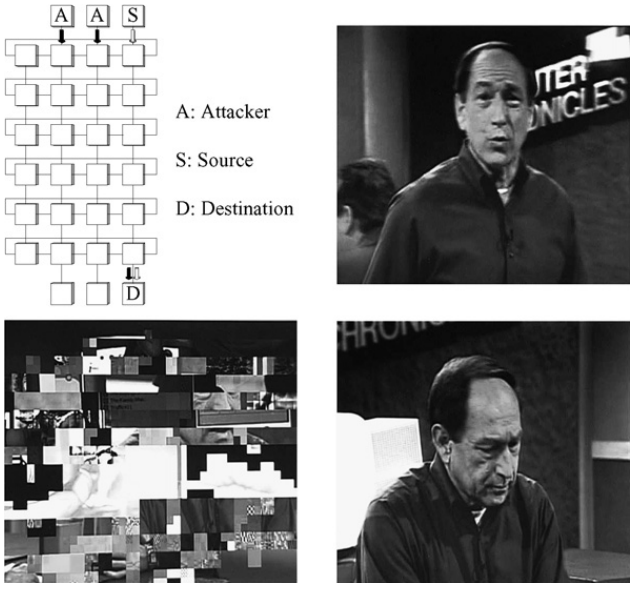


Fig. 6. Experimental evaluation of our defence scheme. The top-left figure shows the CPN test-bed used to conduct the experiments. Top-right shows a frame of video in the un-attacked network. Bottom-left demonstrates the corruption in the video stream due to the attack. Bottom-right shows the restored video sequence after defence is enabled.

We then apply the naïve defence algorithm, based on node 30 detecting a high incoming traffic rate on certain incoming paths, using the CPN trace-back mechanism to order that packets be dropped upstream on the paths, and also dropping traffic coming from those paths into node 30 itself. This alleviates the impact of the attack, resulting in the clear video stream shown in Fig. 6 (bottom right). The experiment shows that if CPN or a similar network routing mechanism is used, when a node is able, even imperfectly, to recognise that it is being attacked, a sensitive real-time data stream can be protected. Of course, the protection is not instantaneous and requires some time to become effective after the attack is detected.

3.2. Discussion

There are many different ways to improve QoS in networks, including admission control which is not discussed at all in this paper. CPN is a specific autonomic technique that offers adaptive routing as a way to offer better QoS to users, and this paper examines how the mechanisms in CPN and specifically SPs and ACKs can also be exploited to improve defence against DoS attacks. The discussion about QoS improvements available through CPN, supported by experimental results, are reported elsewhere [19,21] and are also the object of a US patent.

The technique we have proposed and evaluated is specifically oriented towards the use of self-awareness in the network and is based on a strictly automatic (non-manual) defence without direct human intervention. The event being detected is a significant degradation of QoS for existing traffic flows; admittedly this may not be just due to an attack but could also be due to network overload. However the very fast rise, over some milliseconds, of performance degradation and the fact that it does not rapidly recede is a fairly solid indication of an attack. Indeed, performance degradation and rapid traffic rises due to flash crowds or sudden popularity will give rise to packet losses, and the TCP congestion control mechanism will quickly respond at the user end to significantly reduce the incoming traffic flow. On the other hand, a DDOS attack will ignore its own packet losses and continue maintaining or even increasing the traffic flow.

The CPN mechanism we have described is based on smart packets (SPs) and ACK packets which precede or accompany the incoming traffic towards the node being protected and back to the sources of the flow. Furthermore the rate of SPs and ACKs is proportional to the rate of the flow which they are accompanying. Thus the SPs and ACKs allow all traffic flows to be monitored and traced back to their sources, and dropped along their paths if the flows are considered to be attack flows.

As any defence mechanism, this approach can have a negative effect on benign traffic flows which are mistaken to be attacks. However in this case they are still benign traffic flows which are causing excessive congestion and degradation, and as such if they are TCP compliant they will react by reducing their instantaneous traffic rates.

If we just rely on "natural" packet losses and congestion control as is available through TCP/IP, or use QoS based techniques such as pre-assigned fair queueing, we would negatively affect *all* traffic flows equally without the ability to trace-back attacking traffic towards their source, and directly drop packets in the traffic flows which are part of the attack, or the specific flows which cause performance degradation.

Although we have only discussed how one may deal with simple DDOS models, since our approach exploits the fact that CPN always uses trace-back as an ongoing technique to do QoS routing, if and when the attacking flows modify their sources and paths they will still be accompanied, just like any flow, by SPs and ACKs; thus as the attacks adapt and use different additional sources, so do the countermeasures that we have described.

Defence can obviously be conducted only at the nodes through which the attacking traffic flows: if we could do this perfectly at each source of the attack this would obviously be optimal since it would cut out from the network *all* the attacking traffic. If we just do it at the destination of an attack this is the worst that can happen since the network bandwidth will be uselessly occupied by attacking traffic. Since it is unrealistic to be able to cut off all the attacking traffic at its sources, it is best to be able to drop attacking traffic at each node that it traverses; in CPN this means dropping it on *all the nodes* in *each of the paths* that the attacking traffic uses.

Note that one can also optimise detection, just as is done for instance in signal processing and target recognition. It is possible to construct optimal Bayesian likelihood detectors that would minimise the probability of false alarms and maximise the probability of correct detection. However, all such

techniques assume that the probability distributions (for instance of traffic rates, or of the rise in traffic) are available both for normal and for normal plus attacking traffic, which is currently not the case. Classification is a more refined step going beyond detection: once an attack is detected the observed traffic is a sum of normal traffic and of attacking traffic. Classification is the task of separating these two kinds of traffic which requires more refined statistical tools.

Finally let us point out that our approach is based on constructing a CPN-like environment around nodes or infrastructures that one would like to protect. This may be carried out either via actual addition of hardware nodes, or via an overlay structure that includes the node that we wish to protect. Thus our philosophy is to create protected islands rather than a "completely protected Internet". However, as the density of protected islands increases we would also expect that the network as a whole would also be better protected since many DoS attacks will be nipped in the bud before they can spread over long distances. Furthermore, we would expect that important nodes that may have a large degree of connectivity should be highly protected.

4. DoS protection based on prioritisation and throttling

In the first part of this paper we saw that the performance of a defence scheme based on dropping DoS packets depends on the accuracy of the detection/classification methods. We will now present an improved version of this generic scheme which attempts to decrease the impact of false alarms on the normal packet flows. In the extended defence approach that we propose in this section, instead of just dropping traffic we also suggest the use of traffic prioritising and throttling as an additional response mechanism. Applying packet filtering for defending against DoS attacks is reported in [22] (though the full paper is not available).

We use a simple mechanism to detect the possibility of an attack. If a node (either a recipient or a transit node) receives packets towards a destination (1) at a rate higher than the current *rate threshold* (packets/s or bytes/s) and (2) with a rate increase which is higher than the current *increase threshold* (packets/s^2 or bytes/s^2), then it announces the existence of an ongoing distributed denial of service attack and sends this information to all upstream nodes and to the victim. From then on, the protection mechanism is set into motion along the informed path. In case of disagreement, it is the alleged victim node's responsibility to inform those nodes that there was a false alarm and that they should return to normal operation.

When there is a detected attack in progress, each informed node contributes to the defence by examining every incoming packet for deviations from normal behaviour. Packets undergo a collection of anomaly-based validity tests which may differ for each type of traffic (see Section 1.1). Nodes which have a role in the defence will prioritise traffic with priority levels which are related to the tests. Each time a packet fails a validity test, its priority level may drop accordingly. Additionally, the upstream routers are instructed to throttle down their traffic directed towards the victim node to a level which it can handle. This two-fold protection framework ensures that packets with higher probability of being both valid and harmless, are offered preferential service. Packets which have been marginally classified as invalid may now receive service if there is available bandwidth so as to minimise the collateral damage inflicted by false detection. Packets which have been identified as being harmful are either delayed by being assigned low priority, or dropped. Various simplifications of this scheme, based for instance on grouping all traffic that has been identified as being invalid, can also be considered.

To our knowledge, changing the priority level instead of dropping or redirecting possible DoS packets according to validity tests has not been previously proposed in the literature. On the other hand, throttling as a means of controlling the attack traffic has been extensively investigated. In [10], as mentioned earlier, a general framework to identify and control high-bandwidth aggregates in a network using max–min fair rate limits recursively from the victim to the upstream routers, was proposed, and in [23] the concept of router throttling with a *level-k max–min fairness* algorithm has been suggested to regulate the experienced server load to below its design limit so that it remains operational during a DDoS attack. Their analysis with a mathematical model, simulation and experiments, prove the usefulness of smart throttling during denial of service attacks. However, smartly orchestrated throttling cannot be used on its own, as a stand-alone defence mechanism, since without classification, there can be extensive collateral damage especially if the attack coincides with a flash-crowd.

4.1. Experimental evaluation of prioritisation and throttling

In this section we use our CPN test-bed to evaluate a detection and defence scheme that makes use of prioritisation and throttling in order to defend against a DoS attack, while reducing the impact of collateral damage to valid traffic. For throttling, we will use Token Bucket Filtering (TBF) which is a simple light-weight queuing discipline that only allows packets up to a set rate to pass, with the possibility to allow short bursts in excess of that rate. For prioritisation we will use the PRIO queuing discipline, part of the iproute2 infrastructure provided with Linux 2.2 which serves packets of a certain priority level only if all higher-priority queues are empty.

The topology and size of the attack "experiment" are shown in Fig. 7. The attack experiment lasts 25 s, and at time $t = 0$ valid traffic is flowing into the network from nodes cpn107 to cpn110 and headed towards cpn115 with four constant bit rate (CBR) flows of 2 Mbits/s each. Network link capacities are 10 Mbits so that these existing flows can be carried easily by the network. At $t = 2$ s DoS traffic starts arriving through cpn104 at constantly increasing rate towards cpn115. At $t = 5$ s and $t = 8$ s, cpn105 and cpn106, respectively, join the attack with traffic of similarly increasing rate towards cpn115. In order to emulate the fact that some normal flows may appear after the attack starts and some others may disappear, at $t = 7.5$ s we stop the flow of normal packets from one of the nodes (cpn110) and replace it with another normal flow from cpn113 at the same rate of 2 Mbits/s. As a result of the attack most of the links get increasingly saturated and the rate of valid packets received at cpn115 drops dramatically. It reaches a minimum

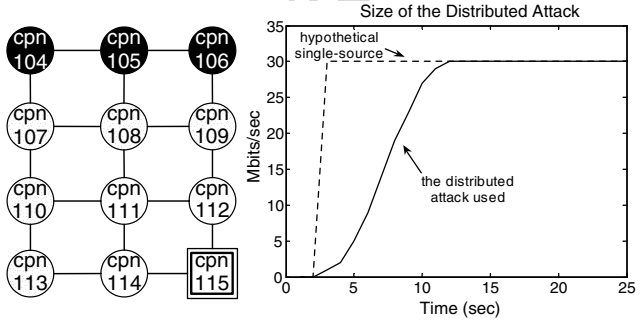


Fig. 7. The topology of the attack scenario (left) and the total attack rate (right); cpn104–106 send up to 10 Mbits/s each and cpn107–110 and cpn113 send up to 2 Mbits/s each.

when the attack stabilises at its peak rate (30 Mbits/s in total) at about $t = 14$ s.

We then repeat the same experiment after applying the following defence scheme. When the total incoming rate of cpn115 is over 15 Mbits/s (arbitrarily chosen, but quite sensible since cpn115 has only two incoming links of 10 Mbits capacity each), this is considered an attack situation and all nodes which are at most 5 hops from the victim (essentially all in this scenario) are requested to act as defenders. We use a marking system of 1–7, in which each newly appearing packet headed towards cpn115 receives an initial mark of 3. The defenders use three simple classification tests to adjust this mark before they forward the packet:

(a) *Is the packet's source among the recognised ones?* If yes then increase the mark by 1. We have arbitrarily chosen three nodes (cpn107, cpn110, cpn113) to belong to the list of recognised customers. The remaining two nodes with valid traffic and the three nodes with DoS traffic do not get a change in their mark.
(b) *Did the packet's source appear after the attack was detected?* If yes then decrease the mark by 1. Cpn106 and cpn113 appear after the attack is detected and receive that penalty (the attack is detected around $t = 7$ s when the total incoming rate at cpn115 is over 15 Mbits/s).
(c) *Is the current bitrate from the packet's source over 7 Mbits/s?* If yes then decrease the mark by 1. In this scenario cpn104, cpn105 and cpn106 all exceed that limit after a while.

The defenders use a priority scheme for their outgoing packets, with priority bands of 1–7. Each band allows up to a maximum rate of 7 Mbits/s and the packets are allocated to them according to their mark (packets with mark 1 go to the worst band, 7). As a result the 2 Mbits/s CBR flows from cpn107, cpn108 and cpn109 receive preferential treatment throughout the whole experiment and at every node, while the remaining flows compete with each other for the low to middle priority bands. The experimental results show that prioritisation according to these simple classification tests allow for practically all valid packets to reach their destination as shown in Figs. 8–12, while the DoS traffic reaching cpn115 is decreased to 50% of each maximum value as seen in Figs. 13–15.

We also repeat the same scenario with fewer nodes participating in the defence, varying the

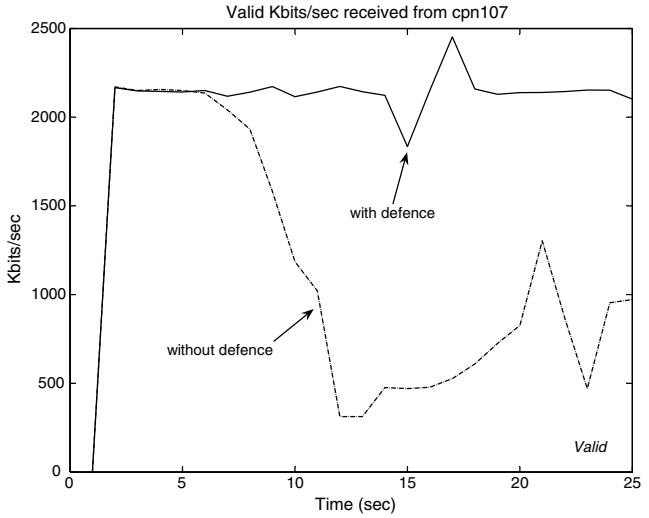


Fig. 8. The Kbits/s from cpn107 that made it safely to cpn115.

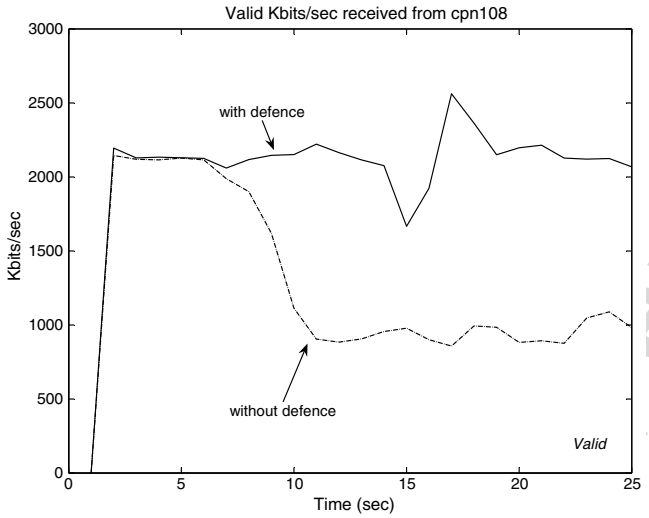


Fig. 9. The Kbits/s from cpn108 that made it safely to cpn115.

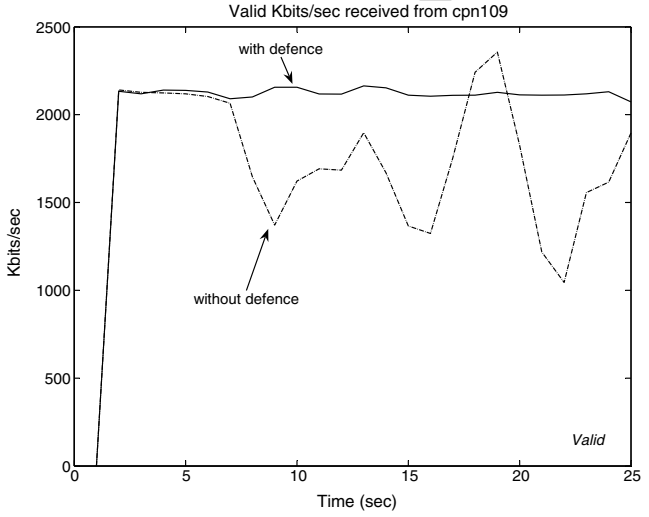


Fig. 10. The Kbits/s from cpn109 that made it safely to cpn115.

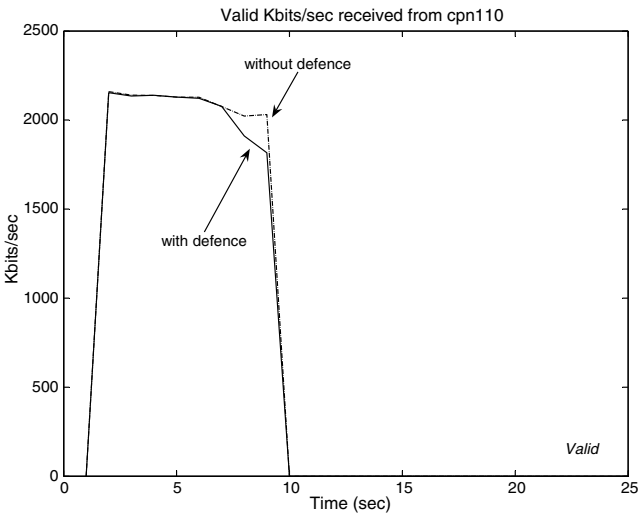


Fig. 11. The Kbits/s from cpn110 that made it safely to cpn115.

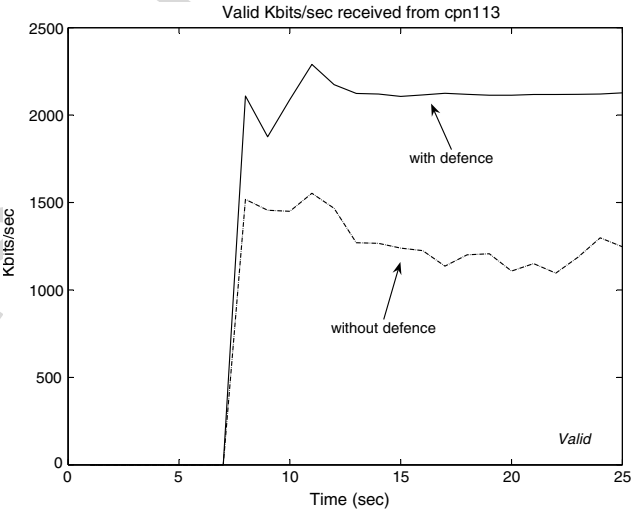


Fig. 12. The Kbits/s from cpn113 that made it safely to cpn115.

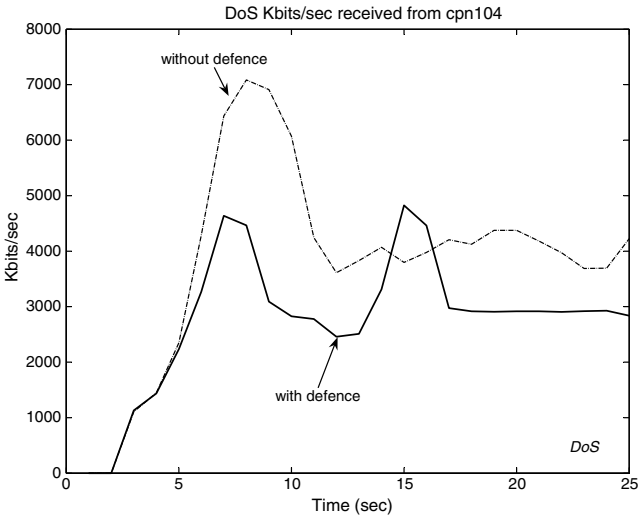


Fig. 13. The Kbits/s of DoS traffic from cpn104 that reached their target.

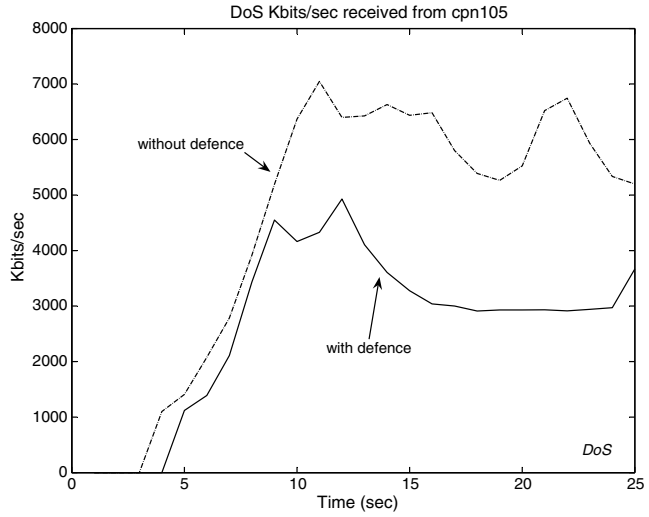


Fig. 14. The Kbits/s of DoS traffic from cpn105 that reached their target.

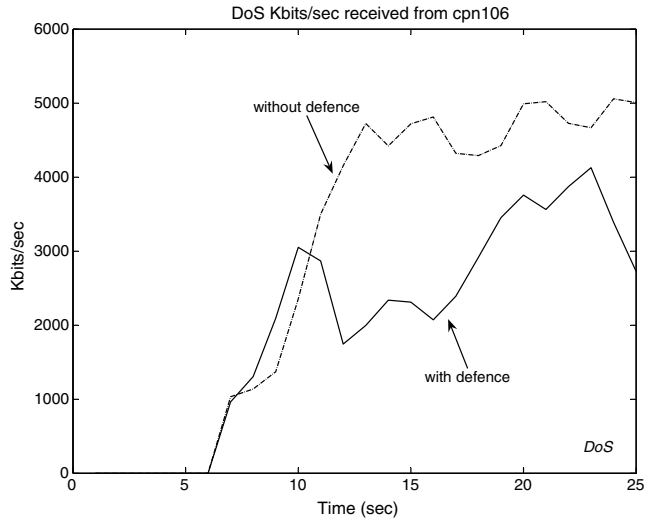


Fig. 15. The Kbits/s of DoS traffic from cpn106 that reached their target.

distance of defending nodes from 1 to 4 from the attacked node as shown in Fig. 16. The results show that a radius of defence in excess of 1 can achieve a level of success comparable to the case when all nodes are used as defenders (Fig. 17). Note that all numerical values shown are the averages of five independent repetitions of a given experiment.

5. Distributing tasks

Extensive filtering and complicated defence systems will introduce significant overhead. Thus different parts of the defense mechanism may be

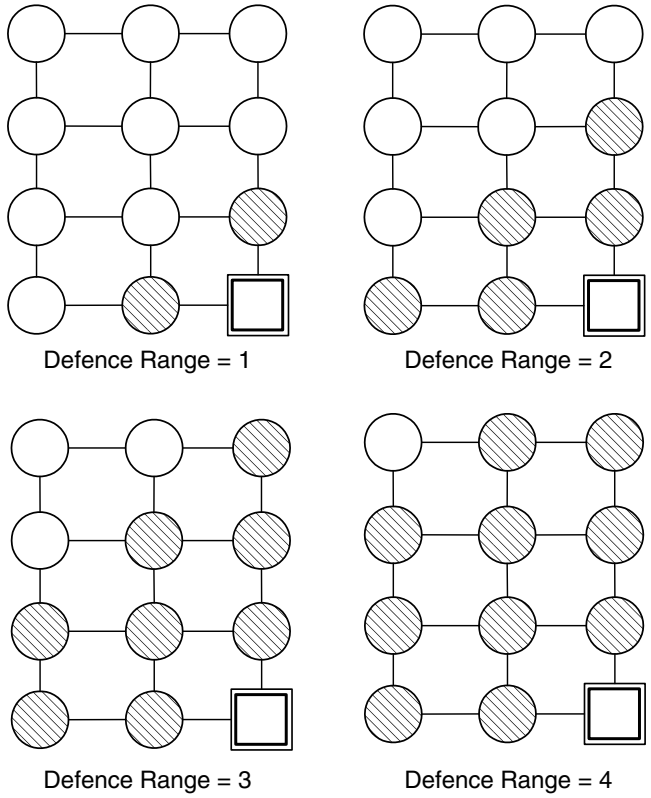


Fig. 16. The defence ranges used in our experiments.

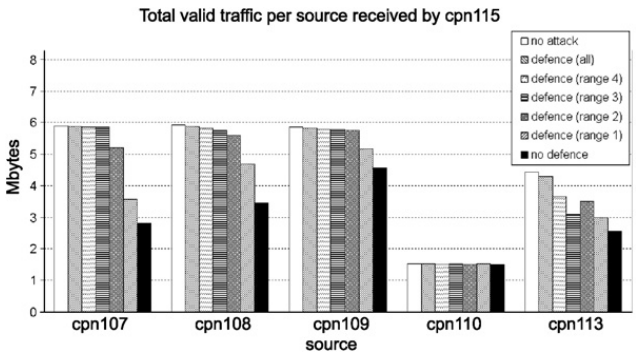


Fig. 17. The total normal traffic that makes it to cpn115 when there is no attack, defence from range 5 to 1, and no defence.

allocated to different nodes, with some specialising in detection and others in traffic classification or defence. The corresponding design may be performed with techniques similar to our mathematical model of Section 2. It is often said that defence can be carried out more effectively closer to the attackers, while detection is more accurate at the victim or at its close neighbours. We will therefore consider this issue with the help of our mathematical model.

Consider the scenario of Section 2.1 as shown in Fig. 2. However instead of using all upstream nodes

for defence, we only use the three nodes that are closest to the attackers (3–5), and then as a contrasting alternative only the ones closer to the victim (1, 2, 6) as shown in Fig. 18. Call these Cases 1 and 2, respectively. For a modest detection performance of $f_{i,\mathbf{n}} = 0.1$ and $d_{i,\mathbf{d}} = 0.6$, the results are contrary to our expectations (Fig. 19) and we see that Case 1 is worse than Case 2. By this, we do not wish to imply that the commonly held view is always wrong, but simply that a mathematical model can help make the best choices as a function of a set of measurable parameters.

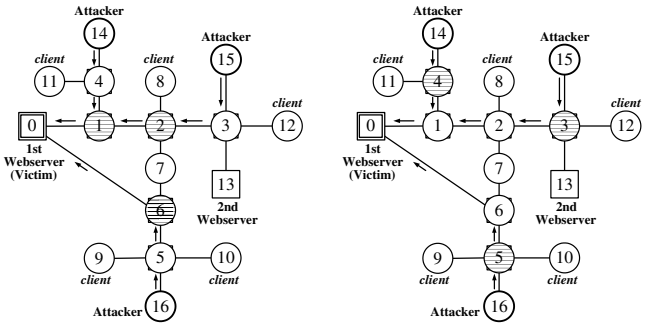


Fig. 18. Partial defence distribution. Left: nodes 1, 2 and 6. Right: nodes 3, 4 and 5.

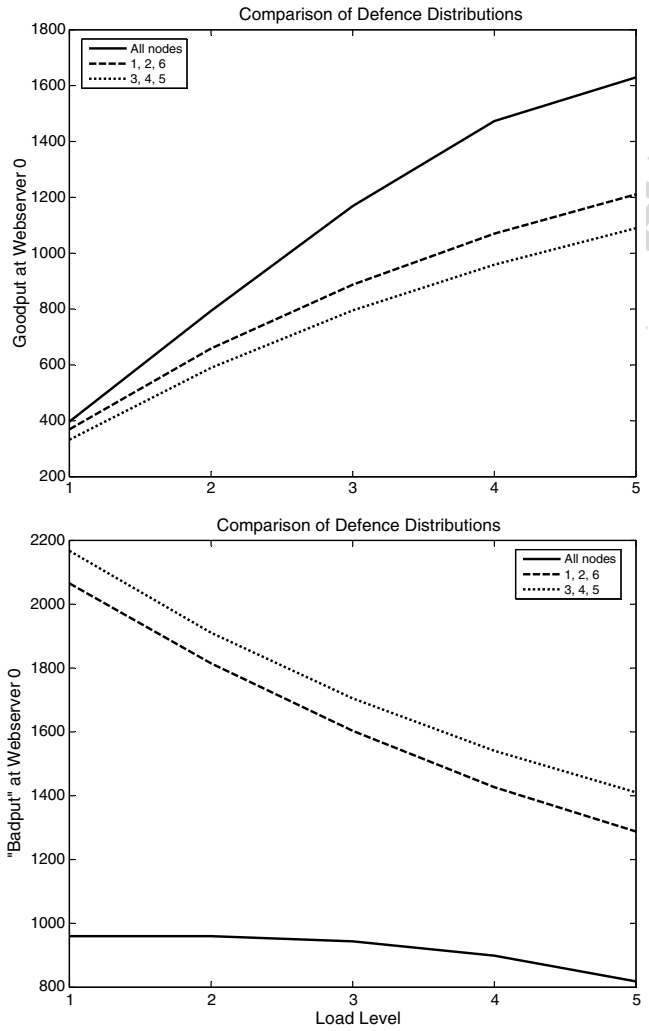


Fig. 19. Comparison of distributions of the defenders.

6. Conclusions and further work

In this paper we have presented an overall architecture for protection against DoS attacks, and we have surveyed much of the literature devoted to both classification and defence mechanisms. We have considered both passive methods of classification based on observing traffic and its characteristics, and active methods which may go as far as trying to interrogate the user to determine whether it is a "live" user or some automated user which has compromised a number of machines in order to conduct an attack.

We have proposed a mathematical model that can provide a gross evaluation of the benefits of DDoS defence based on dropping the attacking traffic, and have included the effect of the probability of correct detection, and the undesirable effect of false alarms which lead to dropping valid traffic. The mathematical model has then been validated against simulation results and experimental measurements on a test-bed.

Then we have considered an autonomic defence mechanism based on the CPN network protocol which is automatically able to trace back flows which are coming in to a node. This approach based on dropping packets at the node, and upstream from a node, has been implemented on our CPN test-bed and evaluated with a video streaming application.

We have found that the most significant drawback of using packet dropping for DoS defence is the "collateral damage" of loss of valid packets due to false alarms. Therefore we have considered a more sophisticated defence approach based on prioritisation and throttling, in which the probability that a packet is "valid" (i.e., that it is not an attacking packet) automatically determines the QoS that it receives. Experiments with the new defence scheme on our CPN test-bed showed that the resulting collateral damage is much lower.

We also suggest that the distribution of tasks in DoS defence should be assisted by a mathematical model such as the one we developed, rather than

relying on intuitive but sometimes wrong decisions. Thus our future work will include a tool to automate the analysis and design of effective DoS defence techniques. Our future work will also investigate classification techniques with improved accuracy, based on Bayesian decision making. Our ultimate aim is to apply this defence architecture in clusters of CPN-based routers around the Internet which will act as islands of defence.

References

[1] R.T. Morris, A Weakness in the 4.2BSD Unix TCP/IP software. Technical Report Computer Science #117, AT&T Bell Labs, February 1985.

[2] P. Ferguson, D. Senie, Network ingress filtering: defeating Denial of Service attacks which employ IP source address spoofing, Tech. Rep. RFC 2267, January 1998.

[3] E. Gelenbe, Z. Xu, E. Şeref, Cognitive packet networks, in: Proc. 11th Int. Conf. on Tools for AI (TAI99), IEEE Computer Society, 1999, ISBN 0-7695-0458-2, pp. 47–54.

[4] E. Gelenbe, G. Pujolle, Introduction to Queueing Networks, second ed., Wiley, London and New York, 1999.

[5] V. Paxson, An analysis of using reflectors for distributed Denial-of-Service attacks, ACM Computer Communications Review 31 (3) (2001).

[6] D. Song, A. Perrig, Advanced and authenticated marking schemes for IP traceback, in: Proc. Infocom 2001, Anchorage, Alaska, USA, 22–26 April 2001, vol. 2, pp. 878–886, ISBN: 0-7803-7016-3.

[7] BBC News, Mafiaboy hacker jailed, September 13, 2001. <http://news.bbc.co.uk/1/hi/sci/tech/1541252.stm>.

[8] G. Rice, J. Davis, A genealogical approach to analyzing post-mortem denial of service attacks, in: Secure and Dependable System Forensics Workshop, University of Idaho, September 23–25, 2002.

[9] A. Snoeren, C. Partridge, L.A. Sanchez, C.E. Jones, F. Tchakountio, B. Schwartz, S. Kent, W.T. Strayer, Single-packet IP traceback, in: IEEE/ACM Transactions on Networking, 10 (6) (2002) 721–734, ISSN: 1063-6692.

[10] R. Mahajan, S. Bellovin, S. Floyd, J. Ioannidis, V. Paxson, S. Shenker, Controlling high bandwidth aggregates in the network. ACM SIGCOMM Computer Communication Review, 32(3) (2002) 62–73, ISSN: 0146-4833.

[11] E. Gelenbe, R. Lent, Z. Xu, Cognitive packet networks: QoS and performance, in: Proc. IEEE MASCOTS Conference, Fort Worth, TX, October 2002, pp. 3–12, ISBN 0-7695-0728-X.

[12] S. Jing, H. Wang, K. Shin, Hop-count filtering an effective defense against spoofed traffic, in: Proc. ACM Conference on Computer and Communications Security, Washington DC, October 2003, pp. 30–41, ISBN 1-58113-738-9.

[13] The Network Simulator NS-2, <http://www.isi.edu/nsnam/ns>.

[14] W.G. Morein, A. Stavrou, D.L Cook, A.D. Keromytis, V. Mishra, D. Rubenstein, Using graphic turing tests to counter automated DDoS attacks against Web servers, in: Proc. 10th ACM International Conference on Computer and Communications Security (CCS'03), Washington DC, USA, October 27–30, 2003, pp. 8–19, ISBN: 1-58113-738-9.

[15] R. Thomas, B. Mark, T. Johnson, J. Croall, NetBouncer: client-legitimacy-based high-performance DDoS filtering, in: Proc. DARPA Information Survivability Conference and Exposition, vol. 1, April 22–24, 2003, pp. 14–25.

[16] G. Mori, J. Malik, Recognizing objects in adversarial clutter – breaking a visual CAPTCHA, in: Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2003 (CVPR'03), vol. 1, Madison, WI, USA, June 18–20, 2003, pp. 134–141, ISSN: 1063-6919, ISBN: 0-7695-1900-8.

[17] A. Hussain, J. Heidermann, C. Papadopoulos, A framework for classifying denial of service attacks, in: Proc. ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication 2003, Karlsruhe, Germany, August 25–29, 2003, pp. 99–110, ISBN: 1-58113-735-4.

[18] M. Sung, J. Xu, IP traceback-based intelligent packet filtering: a novel technique for defending against Internet DDoS attacks, IEEE Transactions on Parallel and Distributed Systems 14 (September) (2003) 861–872.

[19] E. Gelenbe, M. Gellman, R. Lent, P. Liu, Pu Su, Autonomous smart routing for network QoS, in: Proc. First Int. Conf. on Autonomic Computing, New York, NY, May 2004, pp. 232–239.

[20] E. Gelenbe, M. Gellman, G. Loukas, Defending networks against denial of service attacks, in: Proc. Conf. on Optics/Photonics in Security and Defence (SPIE), vol. 5611, London, UK, October 2004, pp. 233–243.

[21] E. Gelenbe, R. Lent, A. Nunez, Self-aware networks and QoS, in: Proc. of the IEEE, 92 (9) (2004) 1478–1489.

[22] Abraham Yaar, Adrian Perrig, Dawn Song, SIFF: A stateless internet flow filter to mitigate DDoS flooding attacks (Short Paper), in: 2004 IEEE Symposium on Security and Privacy, 2004, p. 130.

[23] D.K.Y. Yau, J.C. S Lui, F. Liang, Y. Yam, Defending against distributed Denial-of-Service attacks with max-min fair server-centric router throttles, IEEE/ACM Transactions on Networking 13 (1) (2005) 29–42.

[24] S. Kandula, D. Katabi, M. Jacob, A. Berger, Botz-4-Sale: surviving organized DDoS attacks that mimic flash crowds, in: Proc. 2nd USENIX Symposium on Networked Systems Design and Implementation (NSDI'05), Boston, MA, USA, May 2–4, 2005.

[25] J. Mirkovic, P. Reiher, D-WARD: a source-end defense against flooding Denial-of-Service attacks, IEEE Transactions on Dependable and Secure Computing 2 (3) (2005) 216–232.

Erol Gelenbe is the Professor in the Dennis Gabor Chair and Head of the Intelligent Systems and Networks Group at Imperial College London. His current research interests include novel packet network protocols, the performance of computer systems and networks, and probabilistic models of computations and systems. Recent research results include autonomic Quality of Service based routing and Denial of Service Defense in packet networks. A native of Istanbul and graduate of the Middle East Technical University

in Ankara, Turkey, he is a member of Academia Europaea and a Fellow of ACM, IEEE and IEE. He has some 120 papers in the major journals of computer science and electrical engineering, as well as in journals of applied probability and operations research. He has received honorary doctorates from University of Rome Tor Vergata (Italy), Bogazici University (Istanbul) and Liege University (Belgium), and scientific prizes both in Turkey and France. He is a Commander of the Order of Merit of Italy, and an Officer in the Order of Merit of France.

George Loukas received his B.S. degree from the National Technical University of Athens in 2002 in electrical and electronic engineering. Since 2003 he has been working towards a PhD in the Intelligent Systems and Networks group of the electrical engineering department of Imperial College, where he is also a research assistant since 2005. His research interests include network security, quality of service in networks and disaster management.