Defending Networks against Denial of Service Attacks *

Erol Gelenbe, Michael Gellman, and George Loukas

Imperial College London

ABSTRACT

Denial of service attacks, viruses and worms are common tools for malicious adversarial behaviour in networks. Experience shows that over the last few years several of these techniques have probably been used by governments to impair the Internet communications of various entities, and we can expect that these and other information warfare tools will be used increasingly as part of hostile behaviour either independently, or in conjunction with other forms of attack in conventional or asymmetric warfare, as well as in other forms of malicious behaviour. In this paper we concentrate on Distributed Denial of Service Attacks (DDoS) where one or more attackers generate flooding traffic and direct it from multiple sources towards a set of selected nodes or IP addresses in the Internet. We first briefly survey the literature on the subject, and discuss some examples of DDoS incidents. We then present a technique that can be used for DDoS protection based on creating islands of protection around a critical information infrastructure. This technique, that we call the CPN-DoS-DT (Cognitive Packet Networks DoS Defence Technique), creates a self-monitoring sub-network surrounding each critical infrastructure node. CPN-DoS-DT is triggered by a DDoS detection scheme, and generates control traffic from the objects of the DDoS attack to the islands of protection where DDOS packet flows are destroyed before they reach the critical infrastructure. We use mathematical modelling, simulation and experiments on our test-bed to show the positive and negative outcomes that may result from both the attack, and the CPN-DoS-DT protection mechanism, due to imperfect detection and false alarms.

Keywords: Information Warfare, Internet, Packet Networks, Denial of Service, Cognitive Packet Networks

1. INTRODUCTION

The fact that computer networks have become an essential part of the world's infrastructure has created a new open field for information hindrance and warfare. Network security is essential to the proper functioning of industry, business, social services and social activities. Yet a network security attack may be launched at any time by a teenager, an insider, a criminal, an industrial spy, or even a foreign government, and the motives for such an attack may vary very widely.[1]

Many network attacks are *Denial of Service (DoS)* attacks, which are sometimes combined with worms and viruses. A DoS attack can be briefly characterised as an attack with the purpose of rendering a network resource unavailable to legitimate users. With the first appearance of DoS attacks in the 1980s, there have been a plethora of variants of the same simple concept: to overwhelm the target computer with a huge rate of useless incoming packets. The aim of the attacker then is to exhaust the resources available to legitimate users. While, previously, such attacks were launched by a single source, today they are typically distributed (DDoS), where the attacker uses a large number of compromised computers to attack one or more targets simultaneously.

Although DoS attacks are not new, they were not considered to be an important topic for research until the late 1990s. In February 2000 a 15-year old individual caused billions of dollars in damage to some of the leading Internet organisations, including Yahoo.com, eBay.com, Amazon.com, Buy.com and CNN.com.[2] That incident served to illustrate the power of DoS attacks and was a catalyst for an increase of research into the area.

* This work was supported by the UK Engineering and Physical Sciences Research Council under Grant GR/S52360/01.

Further author information: (Send correspondence to E.G.)

E.G.: E-mail: e.gelenbe@imperial.ac.uk, Address: Dept. of Electrical & Electronic Eng., Imperial College London, London SW7 2AZ, UK

M.G.: E-mail: m.gellman@imperial.ac.uk

G.L.: E-mail: georgios.loukas@imperial.ac.uk

IP address spoofing is one common means of concealing the identity of DoS attack perpetrators. Practically, it refers to the creation of TCP/IP packets using fake IP addresses. Ingress Filtering[3] is an approach to thwart IP spoofing by configuring routers to drop arriving packets that arrive with illegitimate source addresses (i.e., IP addresses outside an "acceptable" range). This requires that a router have sufficient power to examine the source address of each packet, and sufficient knowledge to distinguish between valid and invalid addresses. While ingress filtering has the ability to nearly eliminate IP spoofing, the necessary level of adoption has not been reached in today's Internet, and IP spoofing still remains a problem.

Later, it was suggested that the real IP address could in fact be inferred with a technique called IP traceback.[4] 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). Moreover, the technique can be applied post-mortem, after the attack has completed. Still, attackers can inject false traceback messages into their packet stream to mask their origin.

The latest scheme proposed as a means to defend against IP spoofing is hop-count filtering (HCF).[5] The main idea is 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. This hop-count information can be inferred from the Time-to-Live (TTL) value in the IP header. HCF is simple and practical, but does not work well for bandwidth-based attacks.

One of the leading approaches in the proactive defence against DoS is Secure Overlay Services (SOS),[6] which is geared toward supporting Emergency Services or similar types of communication. This architecture is constructed using a combination of secure overlay tunnelling, routing via consistent hashing, and filtering. It reduces the probability of successful attacks by (i) performing filtering near protected network edges, pushing the attack point perimeter into the core of the network, where high-speed routers can handle the volume of the attack traffic, and (ii) introducing randomness and anonymity into the architecture, thus making it difficult for an attacker to target nodes along the path to a specific SOS-protected destination. The goal of SOS is to route only the "confirmed" users' traffic to the server and drop all the rest. However, SOS does not work for public services (the clients must be aware of the overlay network) and it still allows brute-force attacks on links entering the filtering router in front of the client.

One of the latest efforts in the field of DoS defence is DEFCOM,[7] which suggests that the current paradigm of designing defence systems which operate in isolation should be abandoned. DEFCOM is a distributed framework that enables the exchange of information and services between existing defence nodes. Since, for example, attack detection is best done near the victim, while response is most effective and collateral damage is minimal at the source of the attack, each node should be specialised in a different aspect of the defence. DEFCOM is still in a very early stage, but an obvious shortcoming is the fact that an overlay node could also be compromised and lead to damage the operation of a large part of the network.

Although many intelligent defence techniques and architectures have been proposed, the DoS phenomenon remains and evolves. While until recently, DoS attacks were used by hackers only to take web pages offline, now they are considered to be an important weapon in the hands of cyber-criminals. DoS attacks are reportedly used against business competitors,[8] for extortion purposes,[9] for political reasons,[10] and even as a form of "legitimate protest".[11] It is this variety of types and targets of DoS attacks that dictate the need for flexible defence systems which react both according to the aims of the attacker, and the needs of the defender.

In this paper, we investigate a distributed approach to defence against DDoS attacks using mathematical modelling, simulation and experiments using a networking test-bed that we have developed at Imperial College.

2. A FRAMEWORK FOR DENIAL OF SERVICE PROTECTION

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

- The node which is targeted by a DDoS attack has the ability to detect or to be informed about an ongoing DDoS. This may either be based on a local or a distributed detection scheme. The node under attack, and all nodes upstream from that node, up to the source(s) of the attack, will be informed of the ongoing attack.

- The reaction of the targeted node, and the nodes upstream from it to the attack(s) will be to drop packets which are thought to be part of that attack.
- The attack itself will produce congestion both at the targeted node and at nodes upstream, and indeed downstream, from the attacked node. This can result both in buffer overflows and in the inability of nodes or routers to handle the resulting heavy packet traffic.
- However, the detection scheme is imperfect so that both false alarms and detection failures are possible. Such imperfections are possible both at the level of an attack as a whole, and also with regard to the identification of the individual packets which may or may not play a role in an attack. Thus, we will consider the effect of the probability of correct detection, and the probability of false alarms on individual packets flowing in the network.

Before we address the issue of how in practice we propose to implement a DDoS detection mechanism, we suggest an approach to analyse the impact of DDoS protection on overall network performance. This approach is based on the mathematical model that is presented in Section 2.1. Because of its complexity, the model's solution will be based on an iterative numerical approximation. Therefore, results from a discrete event simulation are also provided in Section 2.3 to validate the analytical approximations. An actual system implementation and resulting experimental measurements are presented in section 3.

2.1. A Mathematical Model of Denial of Service Protection

To evaluate how well such a scheme might work, consider a packet network with N nodes $\{1, \ ... \ , i, \ ... \ , N\}$. In general, the traffic flowing into node i will be the aggregate of several "normal" (benign) flows, and possibly several DoS flows, where for example $\mathbf{n} = (n_1, n_2, \ ... \ , n_j, \ ... \ , n_{L(\mathbf{n})})$ and $\mathbf{d} = (d_1, d_2, \ ... \ , d_j, \ ... \ , d_{L(\mathbf{d})})$ represent 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 which arrives externally at a node 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 rate of arrival to node i of DoS packets belonging to dos flow d. They both correspond to the external traffic that enters the network through the specific node.

Node i will have capabilities to recognise the DoS traffic, though only in an imperfect manner; some DoS traffic will be mistakenly taken to be normal traffic while the remaining DoS traffic will be correctly recognised by the node for what it is. Similarly some normal traffic will be mistakenly thought to be DoS traffic. Any traffic that node i takes to be DoS traffic will be dropped at the entrance to 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, then 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 IP address.

We model each node by a single server queue so that the arrival rate of traffic to the queue is the aggregate of the traffic arriving to the queue, which was not detected as belonging to a DoS flow, while the service time s_i represents both the time it takes to process the packet in the node and the actual transmission time. Let us denote the traffic intensity parameter by ρ_i

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

where $I^n_{i,\mathbf{n}}$ represents the normal traffic that belongs to a normal flow $\mathbf{n}$ and arrives at node i, and $I^d_{i,\mathbf{d}}$ is the traffic rate of a DoS flow $\mathbf{d}$ that arrives at node i.

Since DoS attacks will tend to overwhelm the node's packet processing and transmission capability, packets may be lost by the node with some probability L_i, for instance due to buffer overflow. We can use different formulas to relate traffic intensity to the buffer overflow probability, such as expressions based on large deviation calculations or based on empirical observations. In the sequel, for the sake of simplicity, we will use a finite capacity M/M/1 queue model to compute the Loss probabilities[12]:

$$L_i = \rho_i^{B_i} \frac{1-\rho_i}{1-\rho_i^{B_i+1}}, \tag{3}$$

where B_i is the size of the buffer at node i.

Since any traffic that is correctly or mistakenly thought to be DoS traffic is dropped at the input of the node, and the traffic which effectively enters a node has been filtered in this manner, the traffic equations for the system we are considering which express the rates of normal and of DoS packets entering a node as a function of the other parameters we have indicated, become:

$$\begin{aligned} I^n_{n_j,\mathbf{n}} &= \lambda^n_{n_1,\mathbf{n}} \prod_{l=0}^{j-1} ((1-L_{n_l})(1-f_{n_l,\mathbf{n}})) \\ I^d_{d_j,\mathbf{d}} &= \lambda^d_{d_1,\mathbf{d}} \prod_{l=0}^{j-1} ((1-L_{d_l})(1-d_{d_l,\mathbf{d}})), \end{aligned} \tag{4}$$

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:

- An incoming packet will be dropped due to correct or mistaken identification as a DoS packet, and
- Within each node a packet may be dropped due to buffer overflow or because the node is overloaded due to too many incoming packets, while
- All packets which enter the buffer queue and are not dropped are eventually routed forward to the next node on their path, or absorbed at the current node if it is itself the destination node.

The equations (4) show the dependency of the traffic rates 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 (4) will be obtained via a non-linear iteration which is solved numerically. Note from (4) that it is assumed that all nodes may participate in the detection and discarding of DoS traffic, each with a different role.

The above equations can be solved numerically via the following algorithm:

- Step 0

$$I^{n,(k=0)}_{i,\mathbf{n}} = \lambda^n_{i,\mathbf{n}} \tag{5}$$
$$I^{d,(k=0)}_{i,\mathbf{d}} = \lambda^d_{i,\mathbf{d}} \tag{6}$$
$$\tag{7}$$

- Step $k > 0$

$$\rho_i^{(k)} = s_i I_i^{(k-1)} \tag{8}$$
$$L_i^{(k)} = \rho_i^{B_i,(k)} \frac{1-\rho_i^{(k)}}{1-\rho_i^{B_i+1,(k)}} \tag{9}$$

$$I^{n,(k)}_{n_j,\mathbf{n}} = \lambda^{n}_{n_1,\mathbf{n}} \prod_{l=0}^{j-1}((1 - L^{(k)}_{n_l})(1 - f_{n_l,\mathbf{n}})) \tag{10}$$

$$I^{d,(k)}_{d_j,\mathbf{d}} = \lambda^{d}_{d_1,\mathbf{d}} \prod_{l=0}^{j-1}((1 - L^{(k)}_{d_l})(1 - d_{d_l,\mathbf{d}})) \tag{11}$$

$$I^{(k)}_{i} = \sum_{\mathbf{n}} I^{n,(k)}_{i,\mathbf{n}} + \sum_{\mathbf{d}} I^{d,(k)}_{i,\mathbf{d}} \tag{12}$$

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

$$G(i) = \sum_{\mathbf{n}} I^{n}_{i,\mathbf{n}}(1 - L_i)(1 - f_{i,\mathbf{n}}) \tag{13}$$

2.2. Numerical Example

In this section we will illustrate the use of the model we have developed to evaluate the impact of an attack in the network shown in Figure 1, where a DDoS attack is taking place against the Webserver node 0.

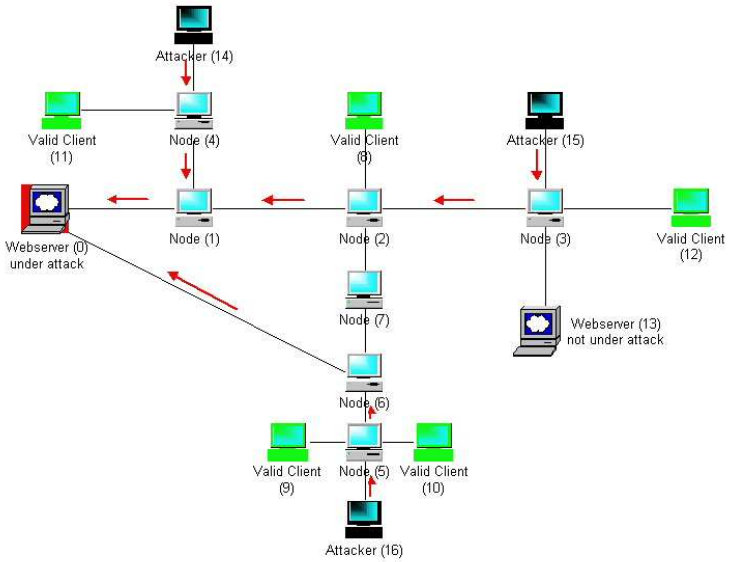

Figure 1. Distributed DoS attack against Webserver 0

In this example, Webserver 0 is being attacked by three DoS flows of 2500 packets/sec (pps) each, entering the network through nodes 3, 4 and 5. Both webservers receive normal packets by all valid clients. We will 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 will 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 $s_i = 0.4ms$ and a buffer size of $B_i = 40$ packets at each node.

Figure 2 shows that if we do not apply any kind of defence, a moderate attack could cause the network's performance to degrade impressively. For example, in high load level, the victim webserver (0) works at less

than 22% capacity, comparing to 99% without the attack. If we apply a simplistic defence in which we drop half of the packets which are destined to 0 (Figure 2, naïve defence), then the results do not improve, at least in the victim webserver's point of view.

A more sophisticated defencive approach is represented by *normal defence*. It could involve a variant of *ingress filtering* in which the nodes drop packets probabilistically according to previous knowledge. If, in the past, the vast majority of legitimate packets arriving to a node belong to a specific range of IPs, then these should be the only acceptable IPs. Such a simple policy would greatly benefit networks with confined target groups, such as a sport's website about a sport which is only played in a few countries, or a small bank's network with clients from a specific geographical area. We arbitrarily choose $(f = 0.1, d = 0.6)$ as the set of dropping probabilities to represent this *normal defence*. The results of the mathematical analysis show a significant improvement in both webservers for all load levels. An even more sophisticated defence $(f = 0.1, d = 0.9)$ would of course yield even better results.

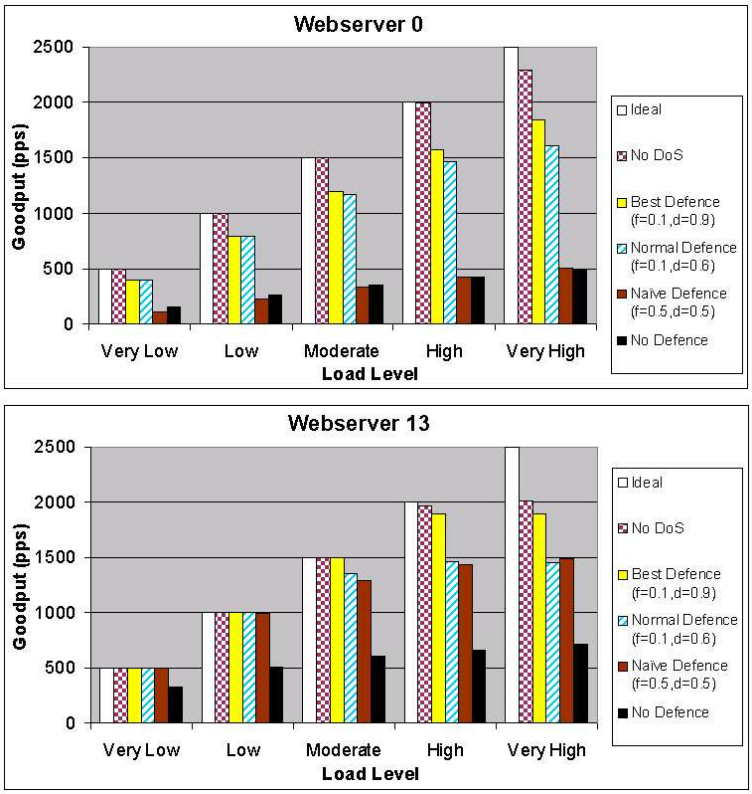

Figure 2. Impact of DoS on Webservers 0 and 13

2.3. Simulation of a DoS Attack

In this section we pursue the preceding discussion using simulations, in order to illustrate the behaviour of the small network in Figure 1 operating in the presence of a DoS attack. We consider the cases of section 2 to be applied. Simulations were carried out with the NS-2 network simulator.[13] For the sake of comparison, we first left the network undefended and then applied a simple defence mechanism based on packet drops along the lines of our earlier discussion.

In the simulation, each normal flow is composed of "normal" UDP traffic at constant rates of 100 to 500*pps* depending on the desired load level, while the DoS flows are also UDP traffic at a rate of 2500*pps*. The packet size is 500 bytes for both the normal and the DoS flows. The links between the nodes are all full duplex and have

a bandwidth of $100Mb/s$. For all nodes the service time is $0.4ms$ and the buffer size is 40 packets. The queues are simulated with the "DropTail" mechanism, which implements simple FIFO scheduling and drop-on-overflow buffer management.

In order to evaluate the effects of the attack on the network, we measure the loss percentage of the legitimate packet rate which is processed at each node. The results are shown in Figure 3, where the x-axis is the network's load level and the y-axis shows the loss percentage of legitimate packets processed by each node for each load level.

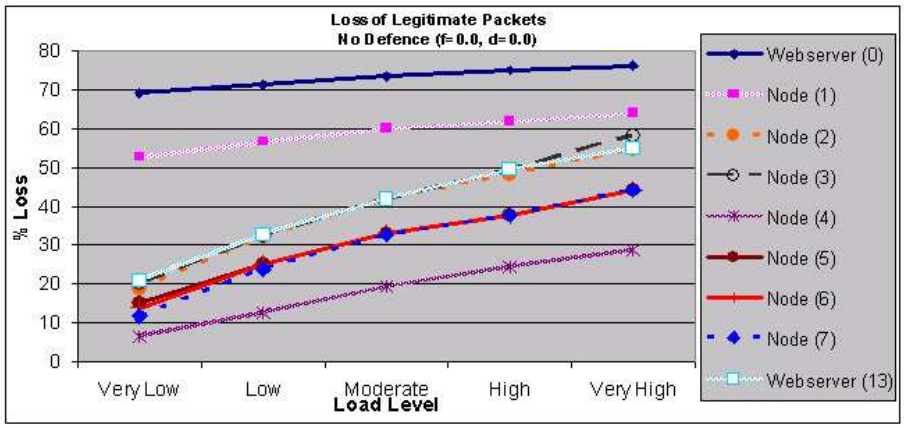

Figure 3. Impact of DoS on each node's goodput

In separate simulations we consider the network's performance under attack, when it is protected with a defence mechanism such as the one described in Section 2. We use the same sets of detection probability values as in the Mathematical Analysis section. The simulation supposes that nodes 1, 2, 3, 4, 5 and 6 are aware of the DoS attack against webserver 0 and apply their defence mechanism, while node 7 is not in any of the DoS paths and will not participate in the defence. By comparing these results (Figure 4) to the ones without defence (Figure 3) we notice different levels of improvement for each node and for each load level. The knowledge of these differences is what should dictate our defence strategy. For example, if our goal is to minimise the impact of the Denial of Service attack on webserver 0 in a network with very high load of legitimate traffic, then we should apply a Defence with detection probability values similar to the *BEST* defence ($f = 0.1, d = 0.9$). However, if webserver 0 is nothing but a decoy and the critical webserver that we want to protect is 13, then for the lower load levels it will be enough to apply the *naïve* defence, in which the nodes simply drop 50% of the packets towards webserver 0.

3. USING CPN FOR DEFENCE AGAINST DOS ATTACKS

In this section we discuss a system implementation that allows us to experiment with some of these ideas. The Cognitive Packet Network (CPN)[14–16] 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 optimize. Payload in CPN is carried by dumb packets (DPs), while smart packets (SPs) and acknowledgement packets (ACKs) gather and carry control information which is used for decision making.

Each CPN flow specifies its QoS requirements in the form of a QoS "goal" that it specifies for its own connection. SPs associated with each flow constantly explore the network, and obtain routing decisions from network routers based on observed QoS information that is relevant to the flow to which the SP belongs. SPs store the identities of the nodes they visit, and the local times at which they visit them. SPs typically constitute 20% of the packets in each flow or connection. At each node, the SP uses a reinforcement learning algorithm based on the experiences of previous smart packets to elicit a decision from the node as to the next hop to travel to. When an SP reaches the destination node of the flow, an ACK packet is generated and returned to the source

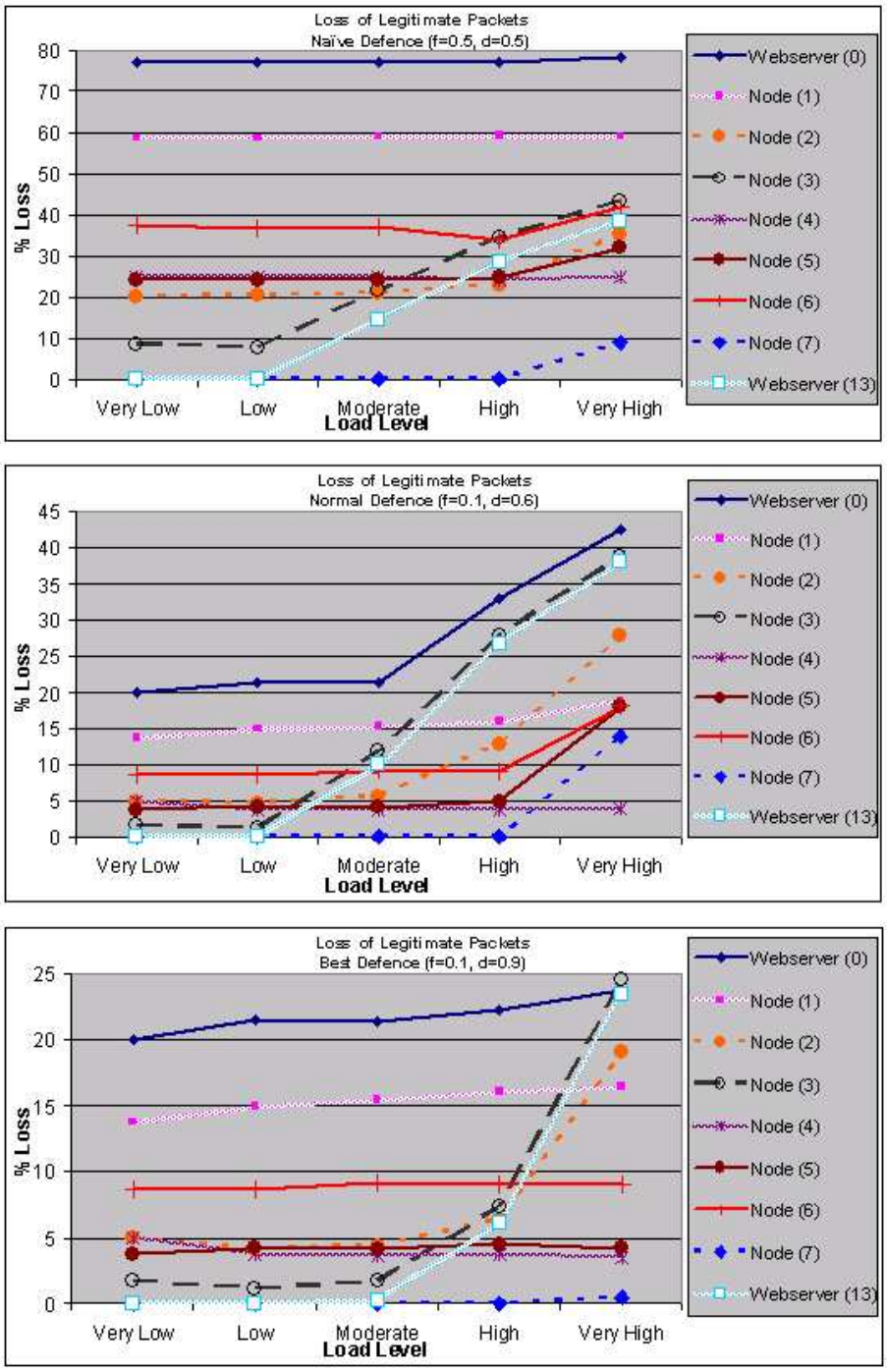

Figure 4. Impact of DoS on each node's goodput, with defence applied

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 sub-paths between identical nodes. Thus ACKs are source routed. At each node it visits, the date about the time it took the SP to go from that particular node to the destination is stored in the node's mailbox. When the ACK reaches the source, the forward

route, which is the reverse of the route that it used, is stored for usage by subsequent DPs belonging to the flow so that DPs are themselves also source-routed.

We propose to use the CPN framework to help protect network nodes and users from Denial of Service (DoS) attacks. In normal conditions, it is desirable for the network to do its best to satisfy its clients' QoS needs. However, in the instance of a DoS attack, attackers can exploit this tendency to damage both the target node's operations, and significantly impair the QoS of users whose flows traverse that node. Thus, in parallel with the user specified QoS goals which relate to individual flows, we propose that each node or IP address be able to specify a bandwidth related objective.

Thus, in the CPN-based DoS defence technique (CPN-DoS-DT) we propose that a node should determine for itself two parameters: the maximum bandwidth that it is able to receive (B_{TOT}), and the maximum allocation of bandwidth that it is willing to allocate to any one flow that traverses it B_{Client}; both are dynamic parameters that may change over time as a function of the conditions at a node, and on the identity and QoS needs of the flow, and they may 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 an 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 B_{Client} allocation. This may occur periodically for each ongoing flow. The router will monitor all of the flows that traverse it and drop all the packets of any flow that exceeds this allocation. Other possible actions could include diverting the flow that ignores the limitation into a special "honeypot", or into a special overlay network used for protection, or it may simply alert a network administrator.

3.1. Experiments with the CPN-based DoS Defence Technique (CPN-DoS-DT)

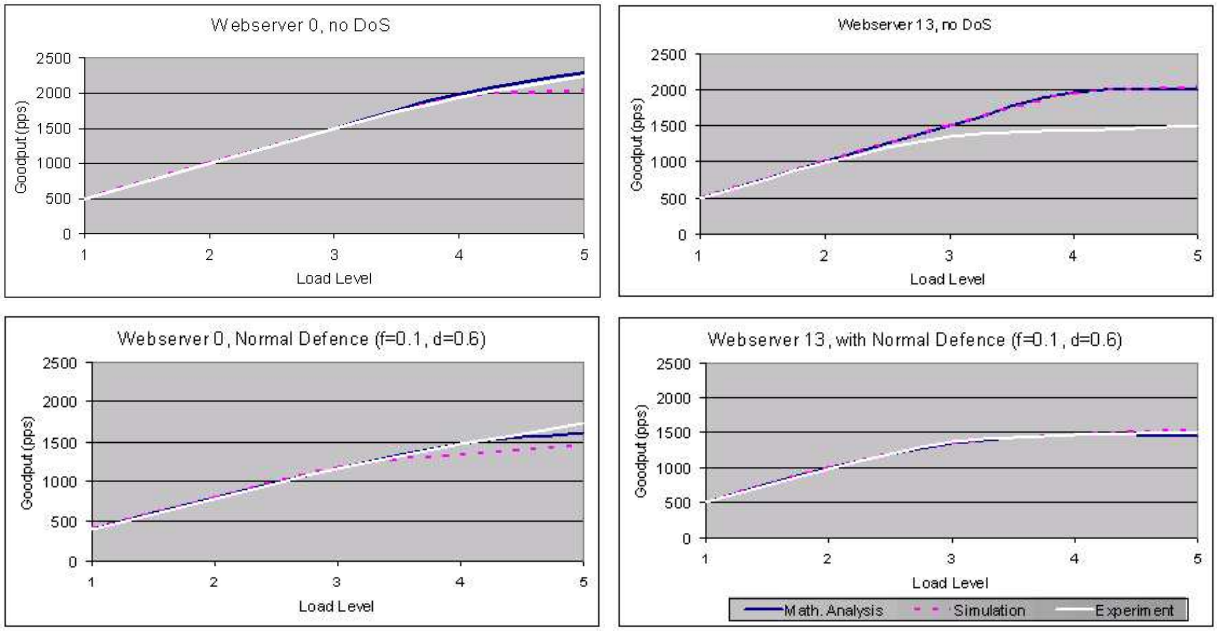

Figure 5. Goodput Comparison of the Three Analytical Methods

To illustrate our ideas concerning the practical DDoS attack protection mechanism we have discussed, experiments were conducted on a testbed of 2.4GHz Pentium 4 PCs configured as shown in Figure 1. Each PC ran the CPN networking code under the 2.4.26 version of the Linux kernel.

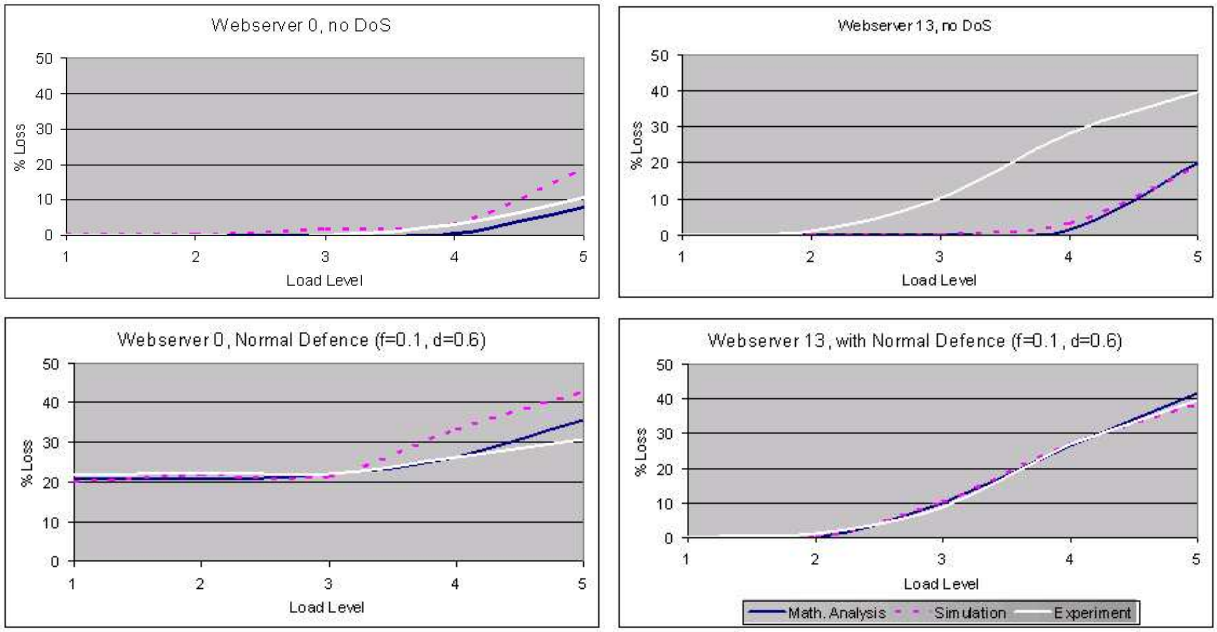

Figure 6. Loss percentage Comparison of the Three Analytical Methods

In order to best compare the results of the analysis and the simulation, care was taken to mimic the simulation parameters described above. For this reason, both Smart Packets and Dumb Packet Acknowledgements were disabled. In addition, routes in the network were manually configured. This allowed us to guarantee that routes remained static during the entire experiment. Different QoS protocols were used to distinguish attack traffic from valid client traffic.

The various defence mechanisms were implemented through the use of configurable drop probabilities which were allowed to be different for different classes of traffic. Thus, we used the same probabilities as given above in the simulations. In order to simulate the 0.4ms service time, a delay-based FIFO queueing mechanism was imposed on each outgoing interface. For forwarding nodes with two output links (i.e. Nodes 1, 2, 3, and 6), the size of the FIFO buffer was divided in half (i.e. 20 packets) to mimic the simulations use of a single buffer per node. Each experiment had a duration of 60 seconds.

The experiments showed similar trends to those found in the mathematical analysis and in the simulations as shown in Figures 5 and 6.

4. CONCLUSION

In this paper, we have discussed a general DDoS defence scheme in which the defending nodes are informed about an ongoing attack and respond by dropping packets which are correctly or mistakenly thought to be part of that attack. We present a mathematical model to investigate the impact of this approach on the "goodput", or useful traffic rate that the network is able to deliver. The goodput depends on the level of congestion that is experienced by the network during the attack, as well as on the probability of correct detection of attacking packets, and the false alarm probability. The predictions of the mathematical model are then validated by comparison with results from a series of simulations. Finally, we present a practical mechanism for implementing such a protection scheme, and describe an experimental system that offers this type of defence based on our existing CPN test-bed at Imperial College. Experimental measurements from the test-bed are also provided, and appear to confirm the predictions of the mathematical and simulation models that we discussed earlier.

REFERENCES

1. CERT Coordination Center, "CERT Overview - Incident and Vulnerability Trends", http://www.cert.org/present/cert-overview-trends, May 15, 2003.
2. BBC News, "Mafiaboy hacker jailed", Sept. 13, 2001, http://news.bbc.co.uk/1/hi/sci/tech/1541252.stm.
3. P. Ferguson and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which Employ IP Source Address Spoofing", Tech. Rep. RFC 2267, Jan. 1998.
4. S. Savage, D. Wetherall, A. Karlin, and T. Anderson, "Practical Network Support for IP Traceback", *Proc. ACM SIGCOMM*, pp. 295-306, Stockholm, Sweden, Aug. 2000.
5. S. Jing, H. Wang, and K. Shin, "Hop-Count Filtering An Effective Defense Against Spoofed Traffic", *Proc. ACM Conference on Computer and Communications Security*, pp. 30-41, ISBN 1-58113-738-9, Washington DC, Oct. 2003.
6. A. Keromytis, V. Misra, and D. Rubenstein, "SOS: Secure Overlay Services", *Proc. ACM SIGCOMM*, pp. 61-72, ISBN 1-58113-570-X, Pittsburgh, PA, Aug. 2002.
7. J. Mirkovic, P. Reiher, and M. Robinson, "Forming Alliance for DDoS Defense", *New Security Paradigms Workshop*, Centro Stefano Francini, Ascona, Switzerland, Aug. 2003.
8. SecurityFocus News, "FBI busts alleged DDoS Mafia", Aug. 2004, http://www.securityfocus.com/news/9411.
9. TechNews World, "Online Extortion Bust Highlights Profit, Problem", July 22, 2004, http://www.technewsworld.com/story/35288.html.
10. SecurityFocus News, "Yaha Worm Takes Out Pakistan Government's Site", June 26, 2002, http://www.securityfocus.com/news/501.
11. ZDNet UK, "Denial of Service attacks loom in GM food protest", Mar. 16, 2000, http://news.zdnet.co.uk/internet/0,39020369,2077753,00.htm
12. E. Gelenbe and G. Pujolle, "Introduction to Queueing Networks", 2nd Edition, 2nd Printing, J. Wiley and Sons Ltd., London and New York, 1999.
13. The Network Simulator NS-2, http://www.isi.edu/nsnam/ns.
14. E. Gelenbe, R. Lent, and Z. Xu, "Measurement and performance of Cognitive Packet Networks", *J. Comp. Networks*, Vol. 37, pp. 691-701, 2001.
15. E. Gelenbe, R. Lent, and Z. Xu, "Towards networks with cognitive packets," (Keynote Paper), *Proc. IEEE MASCOTS Conference*, ISBN 0-7695-0728-X, pp. 3-12, San Francisco, CA, Aug. 2000.
16. E. Gelenbe, M. Gellman, and P. Su "Using Loss and Delay as QoS Goals", *SPECTS Conference*, Montreal, QC, July 2003.