Distributed Defence Against Denial of Service Attacks: A Practical View

Gulay Oke and Georgios Loukas
Dept. of Electrical and Electronic Engineering
Imperial College London
SW7 2BT
g.oke, georgios.loukas@imperial.ac.uk

Abstract

In recent years, Denial of Service attacks have evolved into a predominant network security threat. In our previous work, we identified the necessary building blocks for an effective defence mechanism and suggested ways to integrate them. Here, we present the results of this integration on the DoS-resilience of a real networking testbed which runs the Self-Aware CPN routing protocol. The incoming traffic at each node is monitored with a detection mechanism that is based on maximum likelihood estimation. In response to high probability of attack, the traffic is rate-limited proportionally to the measured probability. We illustrate the results of the experiments we have performed to demonstrate the efficiency of the distributed defence system that we propose.

Keywords: Denial of service, Attack detection, Rate-limiting, Distributed response

1. INTRODUCTION

During the last decade Denial of Service attacks (DoS) have evolved from simple acts of nuisance to a predominant network security threat with repercussions ranging from significant financial losses [15], endangerment of human life [16] and compromising of national security [18]. Due to the simplicity of the concept and the availability of the relevant attack tools, launching a DoS attack is relatively easy, while defending a network resource against it is disproportionately difficult. In the majority of DoS attacks the attacker acquires control of a large number of hosts, which are unaware that their machines are compromised, and orders them to simultaneously target a victim network node or set of nodes. In the most general sense, a complete DoS defence system should be able to detect the existence of the attack in real-time and trigger classification and response mechanisms. Classification refers to distinguishing between normal traffic (sent by legitimate users) and attack traffic (sent by nodes controlled by the attacker). Response mechanisms usually involve dropping the traffic that was identified as attack traffic during the classification phase, or redirecting it to a honeypot where it can be analysed. Classification and response are usually resource-demanding procedures that should not be running continuously, but only when an attack is suspected. For this reason, a comprehensive DoS defence system must include a mechanism that monitors the traffic and signals developing attacks with low false alarm and high correct detection rates, in a timely fashion. The faster a DoS attack is detected, the easier it is to block it before it develops in full force.

We present a DoS defence system that involves detection, nd response, and we describe how these building blocks are integrated in practice. The detection mechanism uses as input a variety of suitable metrics to capture both the instantaneous behaviour and the longer-term statistical properties of the traffic, including the incoming bitrate, entropy, delay, etc. Statistical information related to the network is collected offline by finding the probability density functions for these input features for both normal and attack traffic and calculating the likelihood ratio for each input, which are then combined by evaluating their average. The overall likelihood ratio, L provided by the detector is a numerical value that expresses the average likelihood of having a developing attack within the incoming traffic. This value is utilised by the response mechanism to turn the rate-limiter on and off. As the rate-limiting mechanism, we use Token Bucket Filtering (TBF) which is a simple light-weight queueing discipline that only allows packets up to a set rate to pass. In the following sections we explain the defence system in detail.

2. DETECTION AGAINST DENIAL OF SERVICE ATTACKS

The task of DoS detection can be formulated as a pattern classification problem, where the observed traffic is classified as normal or attack traffic. In our DoS detection mechanism, the incoming traffic is monitored in terms of various features for decision taking and we utilise the maximum likelihood detection criterion to take individual decisions for each of the input features. The collected information is then combined in a fusion phase to yield an overall decision about the traffic. The overall mechanism comprises the selection of the input features, offline statistical information gathering and information fusion for the final decision taking.

For the input feature selection step, we selected the following features that capture both the instantaneous and the longer-term statistical behaviour of the traffic, without introducing high computational costs:

- ***Bitrate***. A very high rate of incoming traffic is by far the most conspicuous indicator of a flooding DoS attack. Similar measurements, such as the number of packets per flow are often used in detection mechanisms [25].
- ***Increase rate of Bitrate***. Depending on its type, a DoS attack typically demonstrates sudden and sustained increases in the rate of the incoming traffic. For example, flooding attacks start with a long period of increasing bitrate, while in pulsing attacks, the incoming traffic undergoes consecutive periods of increasing and decreasing bitrate.
- ***Entropy***. The entropy related to a data with a probabilistic description is inherently associated with the randomness or uncertainty of information in the data. It has been reported in the technical literature that the entropy contained in normal internet traffic and traffic under DoS attack differ significantly [4]. In our work, we compute the entropy of the value of the incoming bitrate at the nodes we monitor according to [1]:

$$E = -\sum_i f_i log_2 f_i \tag{1}$$

  where f_i are the probability density functions obtained from the normalized histogram values for the bitrate. This is expected to yield a higher value when the probability distribution expands over a wider range of values, indicating an increase in uncertainty.
  It has been studied in detail in [12] that the self-similarity properties of normal and attack traffic are distinctively different. Since the Hurst parameter is an indicator of the self similarity of traffic, it can be used in DoS detection. Xiang et al [9] use the variations of the Hurst parameter of the number and the size of packets to detect attacks. In our approach we compute the actual value of the Hurst parameter for the incoming bitrate, for which we have used the (R/S) analysis, as described [10]. If x is the bitrate of the incoming traffic, n is the observation time, and N is the total number of observation points, then (R/S) is given by :

$$(R/S)_N = \frac{\max\limits_{1\leq n\leq N} \sum_{n=1}^{N}(x-\bar{x}) - \min\limits_{1\leq n\leq N} \sum_{n=1}^{N}(x-\bar{x})}{\sqrt{\frac{\sum_{n=1}^{N}(x-\bar{x})^2}{N}}}$$

  The Hurst parameter and $(R/S)_N$ are related by $(R/S)_N = cN^H$, which for $c = 1$ becomes $H = log_N((R/S)_N)$.
- ***Delay***. Although a DoS attack is also expected to increase the packet delays as congestion builds up, to our knowledge it has not been used as an attack indicator. For the fastest and least invasive way to detect changes in the delays, the node we monitor sends constantly a small number of packets to all its direct neighbours. By measuring the average round trip time (RTT) for the acknowledgments to return, we have a clear indication of the congestion near the node.

- ***Increase rate of Delay***. Depending on the type of the attack and for its whole duration, the packet delays are expected to undergo significant changes.

In the off-line statistical information gathering phase, the probabilistic description of the network is obtained First, the probability density function (pdf) values are obtained for both normal and attack traffic and then the likelihood ratios are calculated based on the pdfs. At each victim candidate of the network, the incoming traffic is analysed offline to collect this statistical information. Estimates of probability density functions for both normal and attack traffic are computed for each of the input features described above. The pdfs are denoted by $f_{feature}(x|w_N)$ and $f_{feature}(x|w_D)$, where *feature* is replaced by bitrate, increase in bitrate (bit acceleration), entropy, Hurst parameter, delay and delay rate respectively, x is the measured value of the feature from the available traffic data, w_N denotes the normal traffic and w_D the attack traffic. We have used the histogram method to compute the estimates of the probability density functions. With this method the range of observable values for a variable is divided into a number of intervals and for each interval, we compute the ratio of the number of data points that fall into it to the total number of data points available [26].

In the second step, the probability density function estimates obtained above for each input and for both traffic types are used to compute the likelihood ratios $l_{feature}$ of each feature: $l_{feature} = \frac{f_{feature}(x|w_D)}{f_{feature}(x|w_N)}$. These likelihood ratios are later used in real-time by the decision taking mechanism.

This statistical information collected about the network is utilised during the decision taking process. First, decision for each feature is given individually, and the individual decisions are then combined in an information fusion step to yield a final outcome for the state of the traffic. The numerical values of the features are measured in real-time and a likelihood ratio for each feature is computed. Then, these values are aggregated in a higher-level decision taking step, which provides a compensation for possible errors, and should decrease the rate of false alarms and missed detections. In this case we will simply take the average of the individual likelihood values:

$$l_{final} = \frac{l_{bit} + l_{acc} + l_{entr} + l_{Hurst} + l_{delay} + l_{delrate}}{\textit{total number of features}} \tag{2}$$

A more accurate approach for the fusion of the likelihood values can be found in [22], where we employed both feedforward and recursive structures of the random neural network for a variety of inputs.

A wide variety of DoS attack detection methods have been suggested in the literature, usually based on symbolic analysis of the traffic packets and in particular of IP addresses and other significant packet content. Other approaches are based on the timing characteristics of the packets streams. All of them require or assume some representation of what is a normal traffic stream as opposed to a DoS related stream. Also, many of the techniques require an on-line tuning or learning phase that is used to create patterns, data or statistics to compare with presumed attacks. For this paper, we have used a detection method that we developed with the purpose of maintaining low computational cost and with the additional requirement that the output is not a boolean value, but the probability of the existence of an attack. Any other detection mechanism that fulfills these two requirements could also be used.

3. RESPONSE AGAINST DOS

In this part of our research we tried to combine rate-limiting which we chose as response mechanism against DoS attacks [11, 13] with the detection mechanism. In the overall architecture, the detection mechanism is deployed at the first-hop neighbours of the victim monitors the traffic continuously and outputs a numerical value L expressing the average likelihood of having a developing attack in the incoming traffic. This value is utilised by the response mechanism to turn the rate-limiter on and off. The overall architecture of the defence system is shown in Figure **??**:

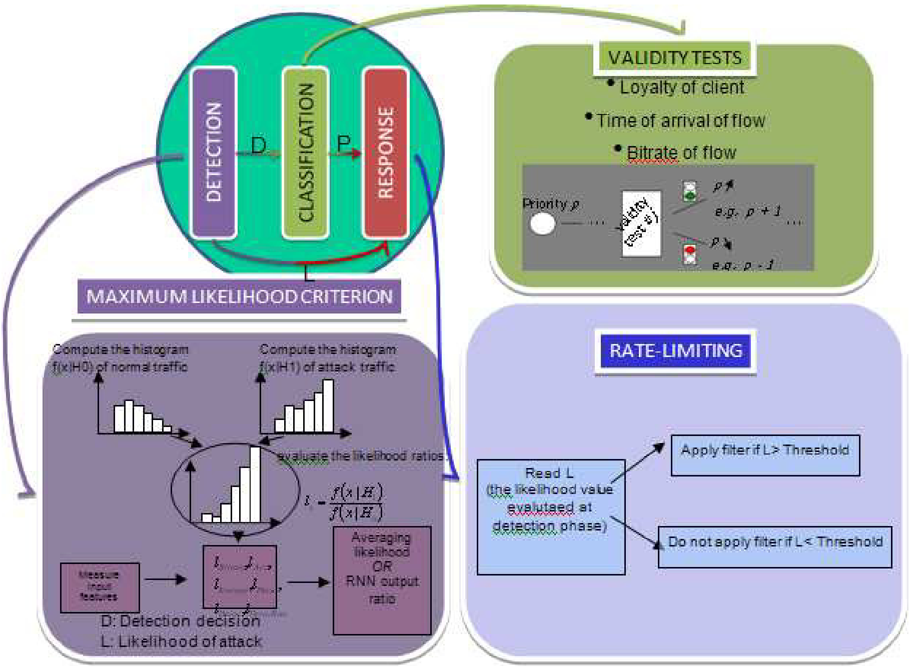
DETECTION
D
CLASSIFICATION
P
RESPONSE
L
VALIDITY TESTS
• Loyalty of client
• Time of arrival of flow
• Bitrate of flow
Priority p
validity test #1
p
e.g. p + 1
p
e.g. p - 1
MAXIMUM LIKELIHOOD CRITERION
Compute the histogram f(x|H0) of normal traffic
Compute the histogram f(x|H1) of attack traffic
evaluate the likelihood ratios
$l = \frac{f(x|H)}{f(x|H)}$
Measure input features
Averaging likelihood OR RNN output ratio
D: Detection decision
L: Likelihood of attack
RATE-LIMITING
Read L (the likelihood value evalutaed at detection phase)
Apply filter if L> Threshold
Do not apply filter if L< Threshold

FIGURE 1: A comprehensive defense architecture against DoS

3.1. Classification and Corresponding Prioritisation

Classification is a vital part of DoS defence. In fact, the probability of correctly distinguishing normal from attack traffic has great impact on the performance of the overall defence system during an attack. In the literature, one can find a wide range of classification techniques with varying success for different normal and attack traffic patterns. Classification can be done with other passive or active tests of the validity of incoming traffic. Passive tests include the anomaly-based criteria presented in [8], conditional legitimate probability [17], hop-count filtering [6] and many others. Active tests are these which in some way try to interact with suspected attack traffic sources so as to test their legitimacy. Examples include Graphical Turing Tests [5] and Netbouncer [7]. It is well-known, however, that classification methods are not easy to evaluate and there has been no such comprehensive comparison up to now. For this reason, we will not consider a specific classification mechanism, but we will assume different ”success” values for the classification process, in the form of correct detection and false alarm probabilities. Our goal is to evaluate our defence system for different such values. The result of the classification is then directly connected to the second element of our response mechanism, the prioritisation. More specifically, the incoming traffic is allocated to priority bands depending on the result of the classification. For example, assuming a 2-band priority system, normal packets should be assigned to the first band and attack packets to the second.

3.2. Rate-limiting

Rate-limiting is the process of allowing traffic only up to a maximum limit to pass. It essentially means that traffic in excess of a set limit is dropped to avoid congestion. For rate-limiting, we have used Token Bucket Filtering (TBF) which is a simple light-weight queueing discipline that only allows packets up to a set rate to pass, with the possibility to allow short bursts in excess of that rate [19].

To apply the filter, we have used commands at the application layer of Linux that determine the desired latency, bandwidth, buffer and burst limits, such as the following example:

$$tc\ qdisc\ change\ dev\ eth4\ parent\ 1:1\ handle\ 10:\ tbf\ rate\ 15Mbit\ latency\ 10ms\ burst\ 15000$$

In our implementation, we first queued the incoming packets into two, with packets coming from legitimate sources having higher priority of being served with respect to packets coming from nodes listed as possible attack sources. To do this, we assumed that as soon as the attack starts it is possible to trace back the true IP addreses of the sources to determine the legitimate and illegitimate nodes. Since there is always error rate associated with this procedure, we assigned predetermined false alarm and correct detection rates for the traceback when we are evaluating our results. Then we integrated the filter with the detection system where the numerical output of the detector L is used by the filter to turn on and off. If L, computed by the detection mechanism is high, the filter is turned on to stop the flow of the packets to the subsequent nodes, while if L is low, the filter is off to enable the flow of packets. In the simplest case, the filter parameter burst can be determined as:

$$rate = \begin{cases} RateMin & \text{if } L \geq limit \\ RateMax & \text{if } L < limit \end{cases} \tag{3}$$

In the above equations $RateMax$, $RateMin$, $MaxLimit$ and $MinLimit$ represent the maximum and minimum values of the rate parameter in the filter, the value of the likelihood after which the burst takes its maximum value(full rate-limiting) and the value of the likelihood before which burst takes its minimum value (no rate-limiting) respectively.

It is also possible to allow for intermediate values of L where the filter correspondingly takes an intermediate value as a limit to decrease the flow of packets to protect the network against a probable attack, without stopping them altogether to permit outgoing legitimate packets.

3.3. Achieving Distribution

The defence approach we have proposed is dynamic and distributed in the sense that every node runs the algorithm itself and decides whether or not to turn the filter on. The response is determined according to the severity of the attack. Only a few of the neighbouring nodes will be employed in rate-limiting if the severity of the attack, denoted by L is not too high, allowing for legitimate packets also to reach the destination, otherwise all the nodes will drop packets. The nodes which are not under destructive attack will continue sending out packets, so the directional information related to the attack will also be employed.

In distributing the response against DoS between nodes, we have been motivated by the task allocation mechanism of ants. In intelligent task distribution systems inspired by swarm intelligence, each member of the team is assigned a threshold, either low, or intermediate or high. The members join to perform the task probabilistically, by comparing the stimulus from the physical world to the threshold they have. Thus, highly demanding jobs require more members to join the team, while easier jobs can be accomplished by only a few workers [27].

In our DoS defence system, the physical stimulus is the likelihood value L evaluated by the detection mechanism at each node. If L at a particular node is not high, it will not join to the response team for filtering, but will turn the filter on if L is above some threshold. Each node gives its own decision according to the stimulus it receives. This approach is useful since each task consumes some of the resources of the node and if not needed the node should not be willing to take on additional task. Also, the aim of the defence mechanism is to drop attack packets while allowing as many legitimate packets as possible to reach their destination. Turning the filter on only at highly required instances could also serve this purpose.

4. EXPERIMENTAL RESULTS

We have tested the performance of this method by the experiments we have designed on our 15-node experimental testbed depicted in Figure 2. The networking testbed is running the CPN Self-Aware routing protocol, which provides detailed measurements of the traffic characteristics in real-time and is particularly resilient to failures and attacks thanks to its self-adaptive design [2, 3].

The Cognitive Packet Network (CPN) is an autonomic Quality of Service (QoS)-driven routing protocol. In CPN each flow specifies the QoS metric that it wishes to optimise, and data payload 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. Each flow specifies its QoS requirements in the form of a QoS "goal" and SPs associated with each flow constantly explore the network and obtain routing decisions from network routers based on observed relevant QoS information. In our experiments we use the CPN to ensure that the traffic arrives to their destination quickly using the optimal routes.

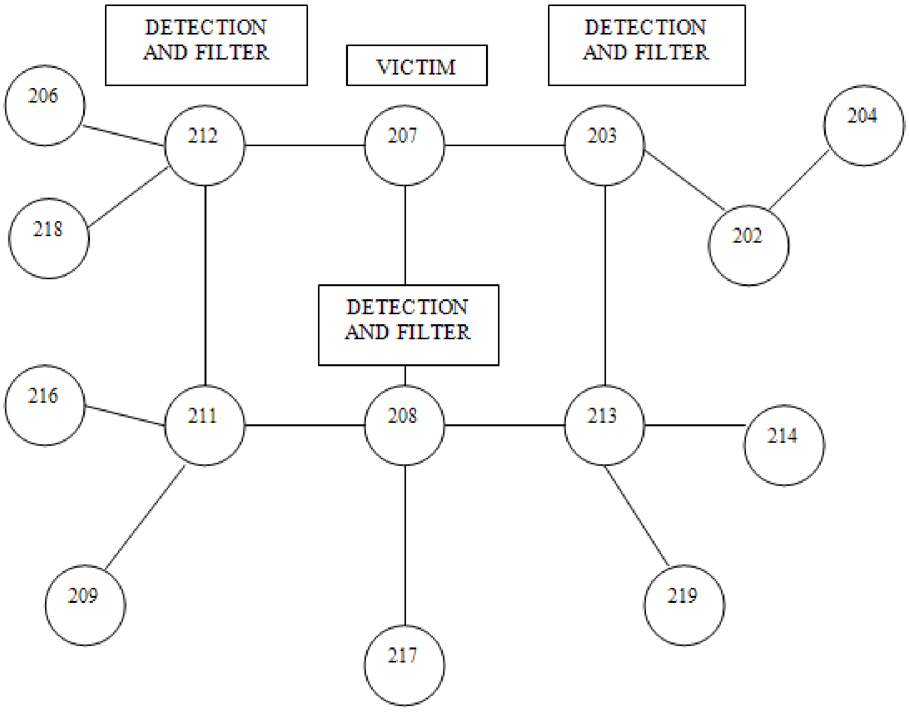

FIGURE 2: The 15-node topology used in the response experiments

In our topology, node 207 is the victim. The experiments last for 120 sec. Between 0-60 sec, there is normal traffic in the network (goodput), which is depicted in Figure **??**. This traffic is basically two cyles of the same pattern, one from 1st to 60th sec. and the other from 61st to 120th sec. At the 60th sec the attack starts and lasts for 40 seconds. Attackers, nodes 202, 206, 209, 214, 216, 217 and 219 send varying attack traffic superposed onto normal traffic existing in the network. To implement the attack traffic we have used traces we have collected from [14] We assume that it is possible to turn on a traceback mechanism to classify the packets according to whether they are coming from legitimate or illegitimate sources based on the source IP addresses. At each node of the network packets are queued according to the source address and the packets in the second queue (illegitimate) receive service after all the packets in the first queue (legitimate) are serviced. Since there can naturally be an error rate associated with the traceback mechanism, we have assigned a false alarm rate of 10 percent and correct detection rate of 90 percent for the determination of legitimate and illegitimate sources. The first hop neighbours 203, 208 and 212 continuously run the detection algorithm and evaluate L. They determine whether to apply the filter or not according to the rate equations given in section 3.2. The experimental results obtained are illustrated at in Figure 4 to Figure 6. Figure 4 shows the likelihood of attack calculated at first hop neighbours. It is observed that the computed likelihood of attack increases between the 60th and 100th seconds to correctly signal the attack in the network. Figure 5 depicts the average goodput arriving from the legitimate sources measured at the victim for the cases of defence and no defence, obtained for 10 runs of the experiment for each case. To have a more precise result for the performance of the defence system, we evaluated the ratio of the average goodput arriving at the victim for the second cycle of the input traffic (when there is attack) to the average value of the goodput for the first cycle (when there is no attack). When defence system is on, the ratio is 0.885, when it is off, the ratio is 0.64. So, the defence system has achieved a conspicious increase on the goodput arriving at the victim. Figure 6 illustrates the variation of the average value of this ratio for a fixed false alarm rate of 10 percent and varying correct detection rates of the packet classification mechanism.

5. CONCLUSIONS

In this paper, we have described our research towards the design of a comprehensive defence architecture against DoS attacks. Our defence system consists of a detection mechanism combining a statistical approach based on the maximum likelihood detection criterion with a machine-learning approach which uses maximum likelihood estimation and a rate-limiting mechanism triggered by the result of the detection. The response mechanism deployed at the first hop neighbours of the victim monitors the traffic continuously and evaluates a parameter signalling the likelihood of a developing attack. TBF filters are turned on and off to limit the packets or allow for their transmission according to the attack likelihood. Since each node collects the statistics and employs the response system itself, this is a distributed architecture which distributes the response task dynamically according to the severity of the attack. The approach we have presented here our first attempt to build an integrated, dynamic and self-adaptive response architecture against DoS.

REFERENCES

[1] Shannon C.E. and Weaver W. (1963) *The Mathematical Theory of Communication*. University of Illinois Press.

[2] Gelenbe E., Lent R. and Xu Z. (2001) Measurement and performance of a cognitive packet network. *Computer Networks (Amsterdam, Netherlands: 1999)*, **37(6)**, pp. 691-701.

[3] Gelenbe E., Lent R., Montuori A. and Xu Z. (2002) Cognitive packet networks: QoS and performance. *Proc. MASCOTS 2002, Modeling, Analysis and Simulation of Computer and Telecommunications Systems*, pp. 3-9.

[4] Feinstein L., Schnackenberg D., Balupari R. and Kindred D. (2003) Statistical Approaches to DDoS Attack Detection and Response. *Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX'03)*.

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

[6] S. Jing, H. Wang, and K. Shin. Hop-Count filtering an effective defense against spoofed traffic. *Proc. ACM Conference on Computer and Communications Security (CCS '03)*, ISBN: 1-58113-738-9, pp. 30-41, , Washington DC, USA, October 27-30, 2003.

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

[8] J. Mirkovic. D-WARD: Source-End Defense Against Distributed Denial-of- Service Attacks. *PhD dissertation*, Univ. of California Los Angeles, August 2003, http://lasr.cs.ucla.edu/ddos/dward-thesis.pdf.

[9] Xiang Y., Lin Y., Lei W.L. and Huang S.J. (2004) Detecting DDOS attack based on Network Self-Similarity. *IEE Proceedings in Communication*, **151**, pp. 292-295.

[10] Cajueiro D.O. and Tabak B.M. (2004) The Hurst Exponent over Time:Testing the Assertion That Emerging Markets Are Becoming More Efficient. *Physica A*, **336**, pp. 521-537.

[11] Gelenbe E., Gellman M. and Loukas G. (2005) An autonomic approach to denial of service defence. *In Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, pp. 537-541.

[12] Li M. (2006) Change Trend of Averaged Hurst Parameter of Traffic under DDOS Flood Attacks *Computers and Security*, **25**, pp. 213-220.

[13] Gelenbe E. and Loukas G. (2007) Self-Aware Approach to Denial of Service Defence. *Computer Networks*, 51, pp.1299-1314.

[14] UCLA CSD packet traces: http://www.lasr.cs.ucla.edu/ddos/traces /public/usc/.

[15] SecurityFocus, August 2004: FBI busts alleged DDoS Mafia, http://www.securityfocus.com /news/9411.

[16] BBC, September 2001: Teenager cleared of hacking, http://news.bbc.co.uk/1/hi/england /hampshire/dorset/3197446.stm.

[17] Y. Kim, W. Lau, M. Chuah, and H. Chao. PacketScore: A Statistics-Based Packet Filtering Scheme against Distributed Denial-of-Service Attacks. *IEEE transactions on dependable and secure computing*, Vol. 3(2), pp. 141–155, 2006.

[18] Goth G. (2007): The Politics of DDoS Attacks. In IEEE Distributed Systems Online, **8(8)**.

[19] Linux Advanced Routing and Traffic Control, http://lartc.org/.

[20] Oke G., Loukas G., Gelenbe E.(2007) Detecting Denial of Service Attacks with Bayesian Classifiers and the Random Neural Network. *Proceedings of FUZZ-IEEE 2007, London, July 23-26*, pp. 1964-1969.

[21] Loukas G., Gelenbe E., Oke G. Detection and Defence against Denial of Service Attacks. Submitted to *ACM Computing Surveys*.

[22] Oke G. , Loukas G. (2007) A Denial of Service Detector based on Maximum Likelihood Detection and the Random Neural Network. *The Computer Journal*, **50(6)**, pp. 717-727.

[23] Loukas G., Oke G. (2007) Likelihood Ratios and Recurrent Random Neural Networks in Detection of Denial of Service Attacks. *Proceedings of SPECTS 2007, San Diego, July 16-18*.

[24] Loukas G., Oke G. (2007) A biologically inspired denial of service detector using the random neural network. *Proceeding of IEEE MASS 2007, Pisa, October 8-11 (BIONETWORKS workshop)*.

[25] M. Kim, H. Na, K. Chae, H. Bang, and J. Na: “A Combined Data Mining Approach for DDoS Attack Detection”, *Lecture Notes in Computer Science*, Vol. 3090, pp. 943-950, 2004.

[26] R.O. Duda, P.E. Hart, and D.G. Stork: *Pattern Classification*, pp. 20-214, John-Wiley and Sons, 2001.

[27] E. Bonabeau, *Swarm intelligence : from natural to artificial systems*, Oxford University Press, 1999.

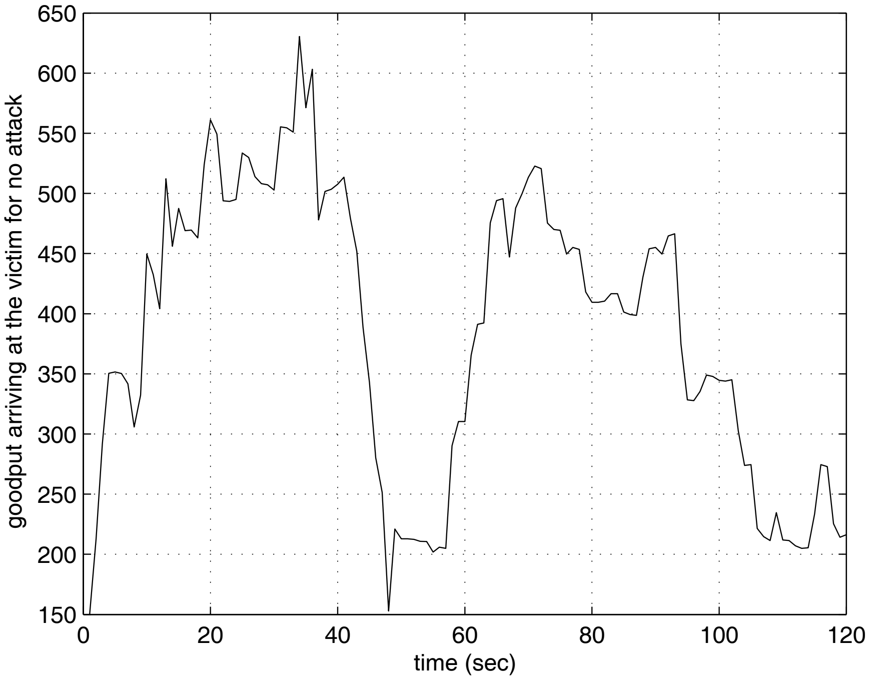

FIGURE 3: Graph of goodput arriving at the victim when there is no attack

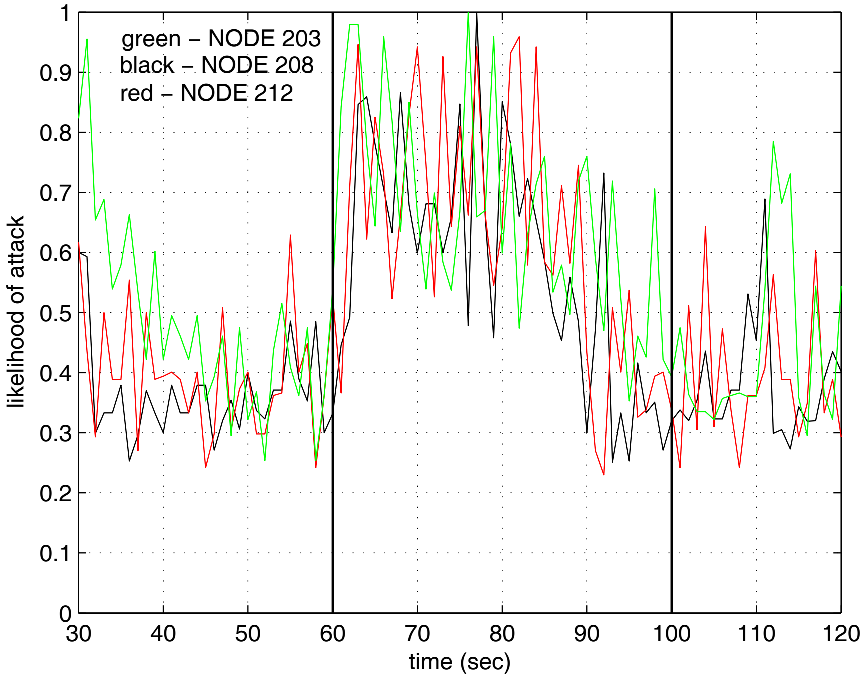

FIGURE 4: Graph of likelihood of attack versus time at the neighbouring nodes to the victim

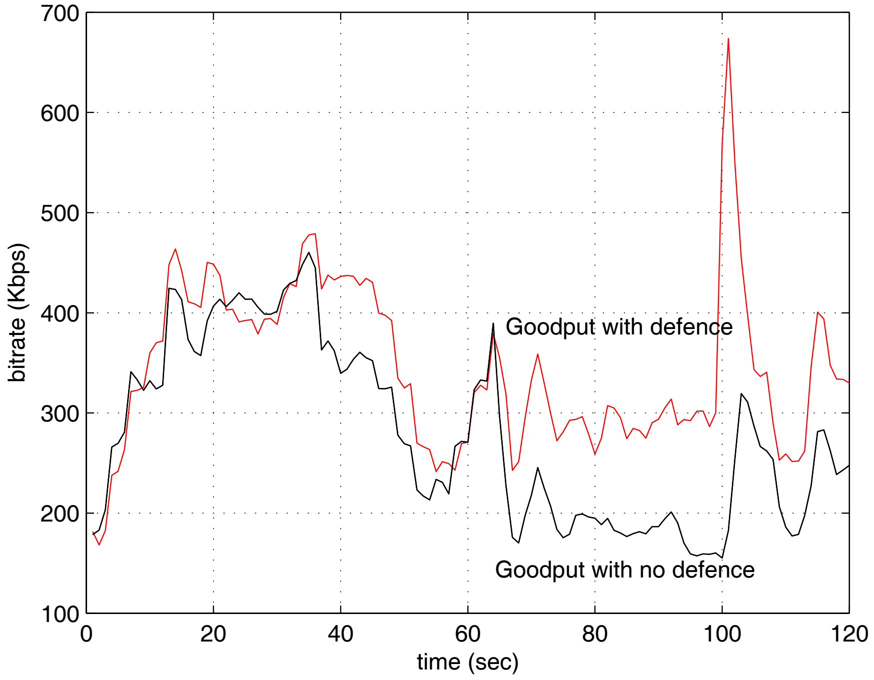

FIGURE 5: Graph of goodput measured at the victum versus time for defence and no defence

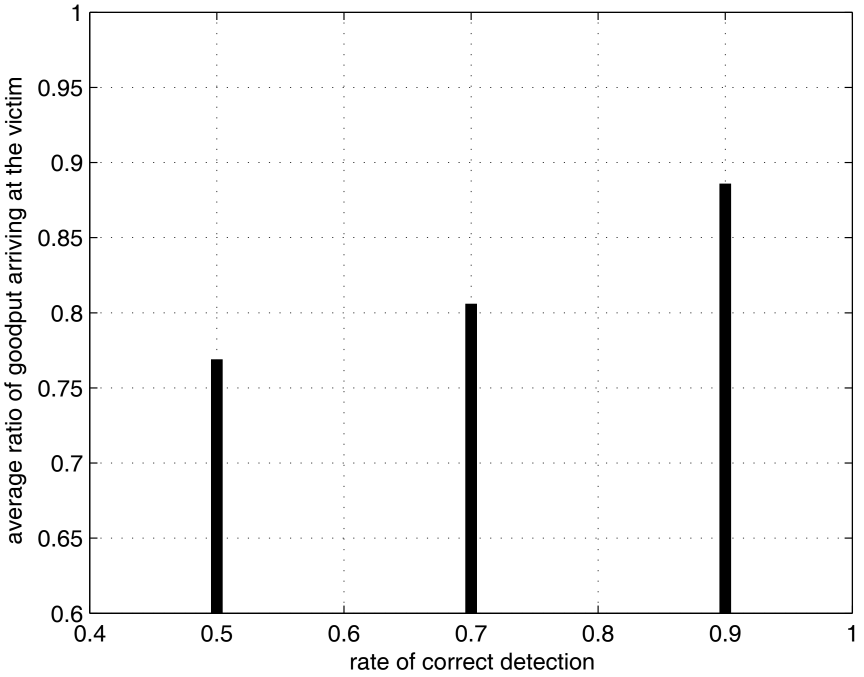

FIGURE 6: Graph of average ratio of goodput (attack to non-attack cases) arriving at the victim versus correct detection rate (False alarm rate is fixed at 10 percent)