Defending against Denial of Service in a Self-Aware Network: A practical approach

Georgios Loukas, Gulay Oke, and Erol Gelenbe

Intelligent Systems and Networks Group
Dept. of Electrical & Electronic Engineering
Imperial College London

Abstract. In recent years, Denial of Service attacks have evolved into a predominant network security threat. Motivated by an impressive variety of reasons and directed against an equally impressive variety of targets, DoS attacks are not as difficult to launch as one would expect. Protection against them is, however, disproportionately difficult. Recognising the fact that the networks of the near future will feature self-awareness and online monitoring, we present a comprehensive system for DoS defence that is specifically designed for such self-aware networks. 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 then prioritised and rate-limited according to the measured probability. Since in a Self-Aware Network, packet routing is dynamic and depends on current network metrics, both detection and response must run individually on each network node, since the nodes through which the attack traffic will pass, may change continuously. We present the experimental results that we obtained using this DoS defence system applied on a real networking testbed that runs the Self-Aware CPN routing protocol.

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 [16], to endangerment of human life [17] and compromising of national security [19]. 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. Recognising that the networks of the near future will have autonomic capabilities, we have developed a DoS defence system that is specifically geared towards such Self-Aware Networks (SAN). A SAN should provide a clear view of its current condition and be able to quickly recognise areas of congestion. It should feature dynamic routing of the traffic according to the requests of its users

and the presence of congested areas. It should also be constantly monitoring the behaviour of all traffic, keep a history of it, and act upon anomalies. In other words, a SAN should be always fully aware of the real-time parameters that determine its performance as a service-providing medium. Naturally, a SAN will be expected to feature autonomic security, especially against Denial of Service attacks.

Denial of service attacks harm Self-Aware Networks in the same way they harm conventional networks, only to a different degree for the various network resources, with the factor of differentiation being the dynamic routing. In a conventional network, the attack paths can be several but remain constant, which results into complete overwhelming of specific nodes or specific links on those static attack paths. A legitimate flow which does not use any of these paths, would be relatively unaffected by the attack, while the rest would suffer complete outages. In a SAN, the routing protocol attempts to accommodate all traffic by dynamically changing the paths. As a result, the attack is distributed in the whole network and affects the quality of service of the legitimate flows in a more balanced manner, in which all flows are affected, but fewer suffer complete outage. For this reason, a DoS defence system for SAN must counter the attack on the paths that the attack is using at each time.

A comprehensive 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 and 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. 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 [26].
- ***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 [13] 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 :

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

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 [27].

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:

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

A more accurate approach for the fusion of the likelihood values can be found in [23], where we employed both feedforward and recursive structures of the random neural network for a variety of inputs. One can use the more sophisticated method which is based on learning, but it is not within the scope of this paper, which is to provide integrated defence and show how the two building blocks, detection and response, interact for various probabilities of successful classification.

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

Here we tried to combine a DoS response mechanism with the detection mechanism. We chose rate-limiting as the response mechanism because it has been widely used before with success [12, 14]. In the overall defence architecture that is shown in Figure 1, the detection mechanism is deployed at the first-hop neighbours of the victim. It monitors the traffic continuously and outputs a numerical value L for the average likelihood of having a developing attack in the incoming traffic. The value L is utilised by the response mechanism to turn the rate-limiter on and off.

3.1 Rate-limiting

Rate-limiting is the process of allowing traffic only up to a maximum limit to pass: traffic over 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

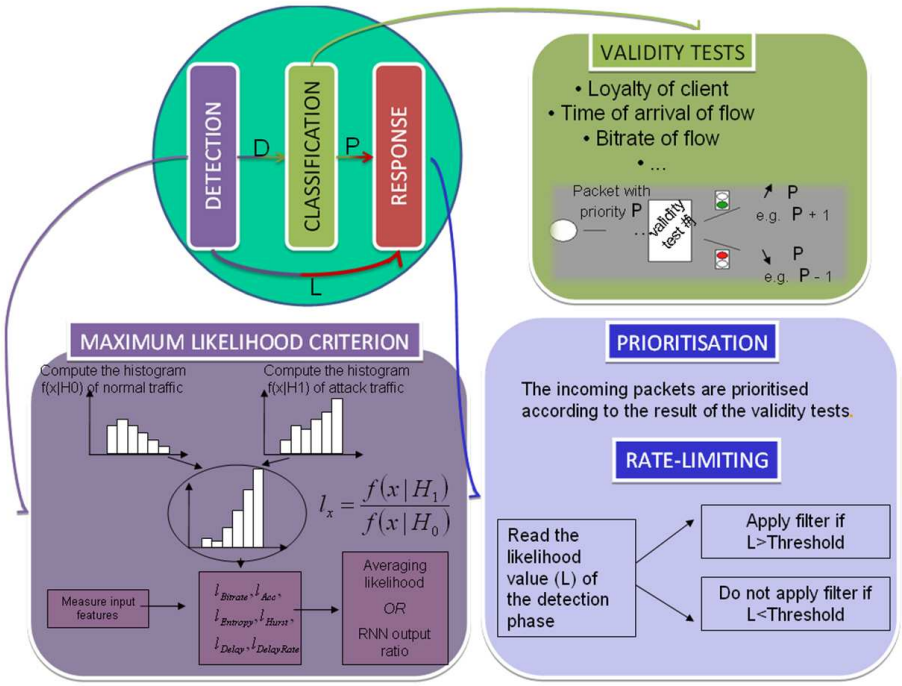

Fig. 1. A comprehensive defense architecture against DoS

to allow short bursts in excess of that rate [20]. 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:

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

The incoming packets are first queued into two priority bands, with packets coming from legitimate sources having higher priority than packets coming from nodes listed as possible attack sources. We assumed that as soon as the attack starts it is possible to trace back the true IP addresses 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 this classification to normal and DoS traffic 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 the value L computed by the detection mechanism is high, then the filter is turned on to stop the flow of the packets to the subsequent nodes, and is turned off otherwise. In the simplest case, the filter parameter burst can be determined as:

$$rate = \begin{cases} RateMin \text{ if } L \geq limit \\ RateMax \text{ if } L < limit \end{cases} \quad (3)$$

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 takes an intermediate value without allowing or stopping altogether the traffic.

3.2 Classification and Corresponding Prioritisation

Classification is a vital part of DoS defence, since the probability of correctly distinguishing normal from attack traffic is a dominant factor of the performance of the overall defence system during an attack. The existing literature provides a wide range of classification techniques with varying success for different normal and attack traffic patterns. Classification can be done with passive or active tests of the validity of incoming traffic. Passive tests include the anomaly-based criteria presented in [11], conditional legitimate probability [18], hop-count filtering [7] 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 [8]. 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. We denote P_n the probability of a normal packet to be correctly classified as normal and P_d the probability of a DoS packet to be correctly classified as DoS packet. The result of the classification is then provided 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. In practice, a packet that has been classified as normal packet should be served before any packet that has been classified as DoS packet and is in the same node's queue. This ensures that packets with higher probability of being valid, are offered preferential service. Packets which have been marginally classified as invalid may receive service if there is available bandwidth so as to minimise the collateral damage inflicted by false detection, while packets that have been identified as being harmful are delayed if there is enough capacity or dropped otherwise.

4 Experimental Results

We have tested the performance of this method with experiments in our 15-node networking testbed, the topology of which is depicted in Figure 2. The 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. In our experiments we use the CPN to ensure that the traffic arrives to their destination quickly using the optimal routes.

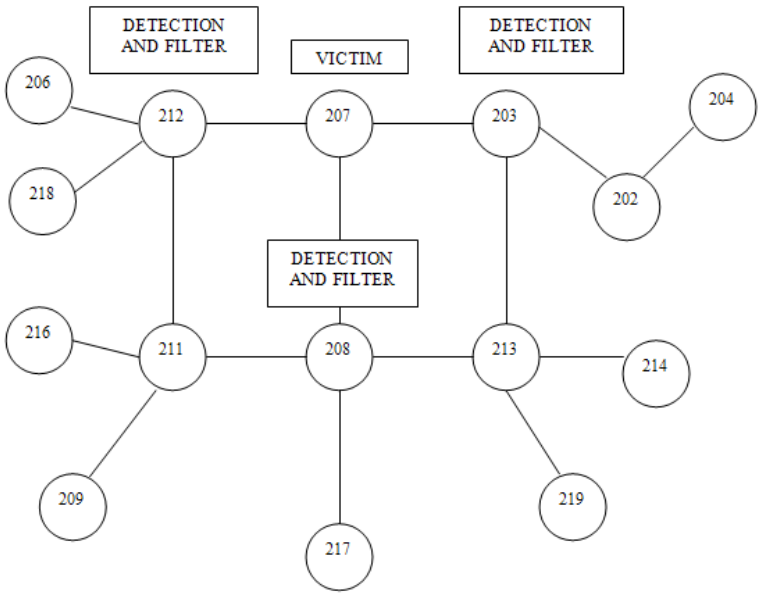

Fig. 2. The 15-node topology used in the response experiments

For the selected topology, the victim is node 207. The experiments last for $120s$. Between $t = 0$ and $t = 60s$, there is only normal traffic in the network, as depicted in Figure 3.

This traffic consists of two cycles of the same pattern, one from the beginning until $t = 60s$ and the other from $t = 60s$ to $t = 120s$. At $t = 60s$ the attack starts and lasts for $40s$. The attackers are the nodes 202, 206, 209, 214, 216, 217 and 219, which send varying attack traffic on top of the existing normal

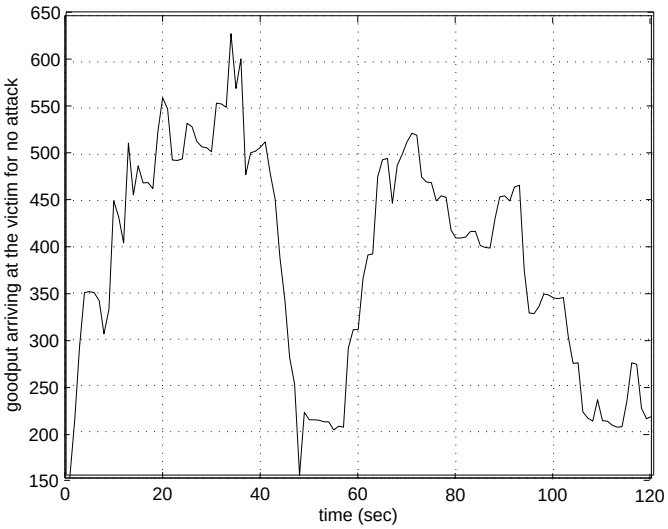

Fig. 3. Graph of goodput arriving at the victim when there is no attack

traffic. To implement the attack traffic we have used attack traffic traces found in [15]. We assume the existence of a classification mechanism that classifies the packets into normal and DoS with given probabilities of false alarm and correct detection. Packets classified as normal packets, enter the high priority band in the outbound queue of each node, while packets classified as DoS packets enter the low-priority band and are served only while the high-priority band is empty. Let us consider a probability of false alarm 0.1 and correct detection 0.9. The first-hop neighbours (203, 208 and 212) run the detection algorithm and evaluate L in real-time. To decrease the impact of false alarms and sudden bursts, we use the rolling average of L of the last $6s$. They determine whether to apply the filter or not according to the rate equations given in section 3.2. The results obtained in our experiments are illustrated in Figures 4 and 5. Figure 4 shows the likelihood of attack calculated at first hop neighbours.

It is observed that the computed likelihood of attack increases between $t = 60s$ and $t = 100s$ to correctly signal the attack in the network. We use term *goodput* for the bitrate of incoming traffic that originates from normal sources. Figure 6 depicts the average goodput at the victim when defence is applied and when there is no defence, 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 the defence system is operating, the ratio is 0.885, while when it is off it is as low as 0.64. Thus, the defence system has achieved a significant increase on the rate of normal traffic arriving at the

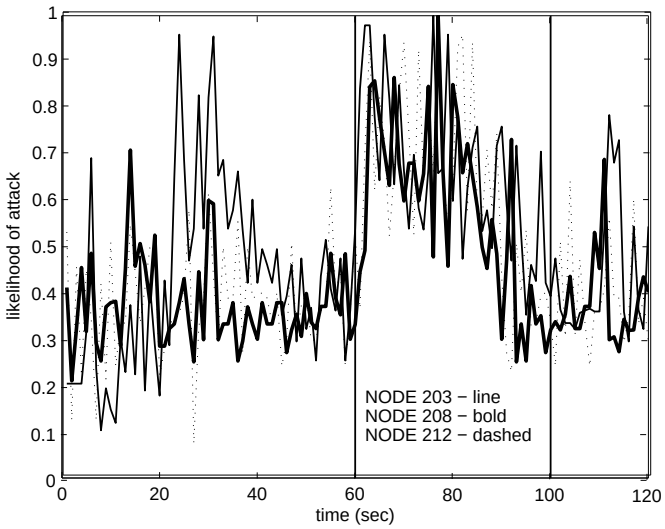

Fig. 4. Graph of likelihood of attack versus time at the nodes neighbouring to the victim

victim while it is under attack. Figure 5 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 that combines a statistical approach based on the maximum likelihood detection criterion with a machine-learning approach which uses maximum likelihood estimation, and a rate-limiting response 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. Rate-limiting filters are turned on and off to limit the traffic according to the likelihood of attack. 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 is our first attempt to build an integrated, dynamic and self-adaptive response architecture against DoS in networks with self-aware characteristics.

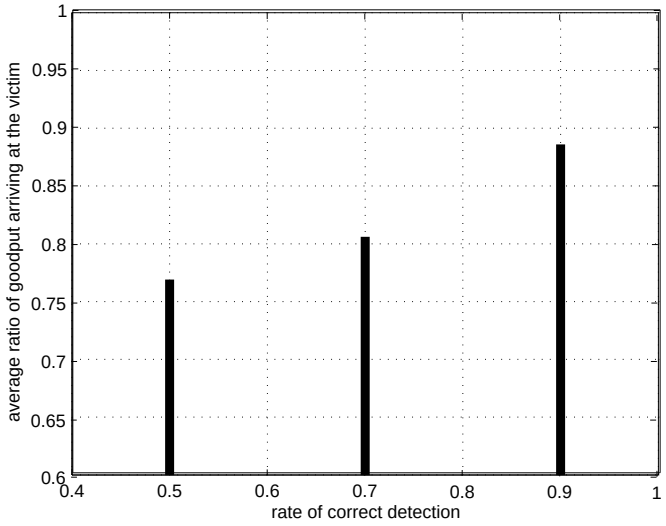

Fig. 5. Graph of average ratio of goodput (attack to non-attack cases) arriving at the victim versus correct detection rate (False alarm rate is fixed at 10 percent)

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. Noh S., Lee C., Choi K. and Jung G. (2003) Detecting Distributed Denial of Service (DDoS) Attacks through Inductive Learning. *Lecture Notes in Computer Science*, **2690**, pp. 286-295.
7. 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.

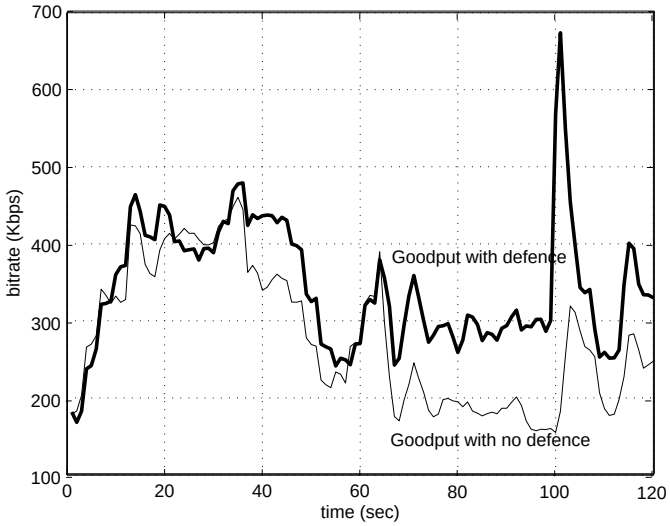

Fig. 6. Graph of goodput measured at the victum versus time for defence and no defence

8. 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.
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. Mirkovic J. and Reiher P. (2005). D-WARD: A source-end defense against flooding Denial-of-Service attacks. *IEEE Transactions on Dependable and Secure Computing*, **336(3)**, pp. 216–232.
12. 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.
13. Li M. (2006) Change Trend of Averaged Hurst Parameter of Traffic under DDOS Flood Attacks *Computers and Security*, **25**, pp. 213-220.
14. Gelenbe E. and Loukas G. (2007) Self-Aware Approach to Denial of Service Defence. *Computer Networks*, 51, pp.1299-1314.
15. UCLA CSD packet traces: http://www.lasr.cs.ucla.edu/ddos/traces /public/usc/.
16. SecurityFocus, August 2004: FBI busts alleged DDoS Mafia, http://www.securityfocus.com /news/9411.
17. BBC, September 2001: Teenager cleared of hacking, http://news.bbc.co.uk/1/hi/england /hampshire/dorset/3197446.stm.

18. 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.
19. Goth G. (2007): The Politics of DDoS Attacks. In IEEE Distributed Systems Online, **8(8)**.
20. Linux Advanced Routing and Traffic Control, http://lartc.org/.
21. 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.
22. Loukas G., Gelenbe E., Oke G. Detection and Defence against Denial of Service Attacks. Submitted to *ACM Computing Surveys*.
23. 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.
24. 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*.
25. 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)*.
26. 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.
27. R.O. Duda, P.E. Hart, and D.G. Stork: *Pattern Classification*, pp. 20-214, John-Wiley and Sons, 2001.