2012-09-01

Real time DDoS detection using fuzzy estimators

Shiaeles, SN

10.1016/j.cose.2012.06.002
Computers and Security

Available online at www.sciencedirect.com

SciVerse ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

Real time DDoS detection using fuzzy estimators

Stavros N. Shiaeles [a], Vasilios Katos [a,*], Alexandros S. Karakos [a], Basil K. Papadopoulos [b]

[a] Democritus University of Thrace, Department of Electrical and Computer Engineering, Section of Software, Building A, ECE, Kimmeria Campus, Xanthi 67100, Greece

[b] Democritus University of Thrace, Department of Civil Engineering, Section of Mathematics, Xanthi 67100, Greece

ARTICLE INFO

Article history:
Received 28 November 2011
Received in revised form
12 April 2012
Accepted 4 June 2012

Keywords:
Distributed denial-of-service attacks
Anomaly based intrusion detection
Fuzzy estimators
α-cuts
Poisson arrival

ABSTRACT

We propose a method for DDoS detection by constructing a fuzzy estimator on the mean packet inter arrival times. We divided the problem into two challenges, the first being the actual detection of the DDoS event taking place and the second being the identification of the offending IP addresses. We have imposed strict real time constraints for the first challenge and more relaxed constraints for the identification of addresses. Through empirical evaluation we confirmed that the detection can be completed within improved real time limits and that by using fuzzy estimators instead of crisp statistical descriptors we can avoid the shortcomings posed by assumptions on the model distribution of the traffic. In addition we managed to obtain results under a 3 sec detection window.

1. Introduction

A Distributed Denial of Service (DDoS) attack is a relatively simple, yet very powerful technique to attack Internet resources (Douligeris and Mitrokotsa, 2004). Perhaps the most representative DDoS attack in terms of social, political and national impact was the 2007 attack on Estonia which literally "unplugged" the Internet from the country (Goth, 2007; Jenik, 2009). DDoS attacks are recognized to be part of cyber warfare tactics but are often employed for blackmail and extortion, primarily for financial gain purposes.

In principle a posteriori DDoS detection is trivial, in the sense that it is noticed once it is successful. However a DDoS maintains a manifestation phase where the attack develops and reaches a threshold which compromises the availability of a legitimate service. Depending on both the attacker and victim resources, the DDoS manifestation phase may range from a few seconds to minutes. As such, in order to thwart a DDoS attack, not only the detection of the event must be completed during the manifestation phase, but the offending hosts need to be identified in order for an incident response control to be effective. In terms of incident response effectiveness, the underlying control must be able to block network traffic belonging to the DDoS attack vector.

In this paper we explore the use of fuzzy estimators on network traffic in order to establish whether a DDoS takes place and to identify the suspect, participating hosts. The rest of the paper is organized as follows. In Section 2 we present a review of the current state of the art DDoS detection techniques. In Section 3 we develop the theoretical underpinning of the proposed method. Sections 4 and 5 contain the empirical results and conclusions respectively.

* Corresponding author.
E-mail addresses: shiaeles@ee.duth.gr (S.N. Shiaeles), vkatos@ee.duth.gr (V. Katos), karakos@ee.duth.gr (A.S. Karakos), papadob@civil.duth.gr (B.K. Papadopoulos).
0167-4048/$ – see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cose.2012.06.002

2. Background and motivation

Detection of security breach attempts such as network intrusion and DoS attacks fall into two main categories, namely pattern (Mirkovic and Reiher, 2004) or misuse detection and anomaly detection (Katos, 2007; Patcha and Park, 2007). In the former we explicitly define patterns of behavior that are classified as malicious and should these are observed within the network traffic, we assume that the underlying system is under attack. In anomaly detection we model normal or benign behavior is and if any outliers emerge outside the prescribed envelope we conclude that the system is under attack. Our proposed method falls into the anomaly detection category.

From a practical perspective, a DDoS attack is associated with bursting traffic (Li et al., 2003). As such, DDoS detection focuses on distinguishing DDoS traffic bursts with benign type of bursts, such as flash crowds for example. In anomaly detection terms, we need to define what normal behavior is. On the network level, this is typically done by adopting a packet arrival model. However, choosing a suitable model is problematic. Although the most prevalent theoretic model in networking is Poisson (Park et al., 2006) which has been used for many years, the modern Internet has triggered a heated discussion and dispute in the literature. In their landmark paper, Paxson and Floyd (1995) explicitly argue that Internet traffic cannot be expressed by Poisson arrival. Although this position has many followers, their claim is directly disputed by Gribble and Brewer (1997). As it seems that no consensus can be reached in the selection of the model, we are lead to the conclusion that the model must depend upon a particular number of parameters (such as type of protocol, whether it is human generated or not, temporal scope) and context (see for example Arlitt and Williamson, 1997).. In Wang's et al. (2002) words, "it may not be possible to model the total number of TCP connections at all times by a simple parametric model". For example, flash crowds are assumed to be Poisson (Li et al., 2008a,b; Ari et al., 2003), whereas HTTP traffic as a whole may or may not be display Poissonity; the work by Guerin et al. (2003) captures these contradictions.

However there seems to be a slight precedence of Poissonity in the literature when it comes to modeling human generated HTTP traffic. This is true when the temporal window of analysis is relatively small, as in the opposite case the arrivals may be non-stationary and will in effect depart from a Poisson model. A small window is desirable in DDoS attack detection, and therefore deviations from the Poisson model may reveal that the packet arrival times may not be human generated (i.e. botnet driven DDoS attacks).

This paper was motivated against the above and we argue that Poisson can be considered for DDoS detection, but only in conjunction with fuzzy estimators. A fuzzy estimator will in essence capture all statistical information within a fuzzy number (in our particular case we use alpha-cuts, α-cuts). By doing this, any error introduced due to the adoption of inappropriate model tends to zero, as the fuzzy estimator allows for this uncertainty. The limitation though of using such an approach is the dependency upon historical data and therefore lack of such data do not allow the application of the approach. However, lack of historical data is rather uncommon in real life, production systems.

Another constraint set out in this paper is the real time requirement. We argue that any DDoS method in order to be effective and offer added value to the infrastructure it protects should be able to perform in real time. We consider the upper limit for detection delay to be equal to the capacity of the server which is being protected. In a recent paper (Wang and Yang, 2008) a "real time" detection of DDoS was achieved by using fuzzy rules on the Hurst parameter. The time needed for the attack to be detected successfully was 13 s which can be classified as realtime in a certain context. The Hurst parameter was also considered (Xia et al., 2010) which in this case it was calculated through statistical traffic analysis and more particularly through the discrete wavelet transform (DWT) and the Schwarz information criterion (SIC). Wei et al. (2006) augment fuzzy classification approaches with cross correlation in order to improve the accuracy of DDoS detection. Although combination of methods is expected to produce improved accuracy results, the realtime requirement is not met due to the increased computational costs.

The nature of the DoS attack has encouraged the employment of many statistical tools (Feinstein et al., 2003). Apart from their appropriateness, statistical tools are also preferred in DDoS detection because of their high responsive potential (Oshima et al., 2010; Lee et al., 2006). In (Sengar et al., 2008; Tang et al., 2009) the authors make use of the Hellinger Distance which is a metric used to measure the distance between two probability distributions. The detection method is applied in the domain of VoIP communications. Covariance analysis (Jin and Yeung, 2004; Yeung et al., 2007) is also used to statistically distinguish normal traffic behavior from flooding.

Other categories of DDoS detection tools include the use of entropy (Lakhina et al., 2005; Feinstein et al., 2003; Yu et al., 2008), neural networks (Arun Raj Kumar and Selvakumar, 2011), fractals and wavelets (Li and Lee, 2003; Li, 2004; Rincón and Sallent, 2005), as well as Support Vector Machines (Ramamoorthi et al., 2011; Shon et al., 2005), Genetic Algorithms (Lee et al., 2011; Li et al., 2008a,b) and FCMs (Siraj et al., 2004).

3. Description of the proposed method

We initially provide a qualitative description of the proposed method. Consider a web site with varying, benign hits throughout a period of time (say a day). Since the number of hits varies, the corresponding time series will be non-stationary; in our case this will be the TCP packet arrival times related to the HTTP traffic. The period needs to be broken into smaller time windows where the length of each time window would be small enough so that it is comparable to the real time detection DDoS limits and to fit to a Poisson model. For each period we calculate the average packet arrival time. If we were to guarantee that the underlying model is Poisson, then during an attack we could statistically compare the recorded, historical mean with the current, observed one. In the case of an attack we would test whether the new mean is statistically smaller than the historical one. However, since an attack – being non-human – may not fit a Poisson

description, the statistical comparison is not appropriate. Therefore we would need to relax the model assumption. In this paper this is achieved by the introduction of fuzzy estimators and more specifically with the so called α-cuts which are formally described in the next section. The method adopted in this research was originally developed and published in recent work by Tsironis et al. (2010) and Chrysafis and Papadopoulos (2009).

Upon detection of a DDoS attack, the next step would be to identify the offending hosts. This is a challenging phase for two reasons. First, the accuracy of the method needs to be high in terms of false negatives and positives. Second, in order to the method to be practical and offer added value, it needs to be able to detect the hosts in real time, that is within certain tight limits. Since the mean would already be expressed by a fuzzy estimator, we have all the information necessary to perform a computationally inexpensive comparison. Detection is done by measuring the mean packet arrival for each IP against the fuzzy estimator.

3.1. *Non-asymptotic fuzzy estimators: our approach*

In this section, we present a more natural way of constructing fuzzy estimators. The network parameter we have selected to monitor is the packet arrival interval and the fuzzy estimator we attempt to construct is the mean packet arrival time. As stated earlier, the fuzzy estimator is capable of capturing all the statistical information generated from the historical data in a single (fuzzy) number. In a DDoS event the observed packet arrival time will be less than the mean packet arrival time. We move on to describe how to derive this fuzzy estimator of the mean.

Proposition 1. Let $X_1,X_2,\ldots,X_n$ be a random sample and let$x_1,x_2,\ldots x_n$ be sample values assumed by the sample. Let also$\beta\in[0,1)$. If the sample size is large enough and Φ denotes the standard normal distribution function, then

$$M(x)=\begin{cases}\dfrac{2}{1-\beta}\Phi\left(\dfrac{x-\bar{x}}{\sigma/\sqrt{n}}\right)-\dfrac{\beta}{1-\beta} \text{ if } \bar{x}-\dfrac{\sigma}{\sqrt{n}}\Phi^{-1}\left(1-\dfrac{\beta}{2}\right)\le x\le\bar{x}\\ \dfrac{2}{1-\beta}\Phi\left(\dfrac{\bar{x}-x}{\sigma/\sqrt{n}}\right)-\dfrac{\beta}{1-\beta} \text{ if } \bar{x}\le x\le\bar{x}+\dfrac{\sigma}{\sqrt{n}}\Phi^{-1}\left(1-\dfrac{\beta}{2}\right)\end{cases} \quad (1)$$

the base of which is exactly the 1-β confidence interval for μ and the α-cuts of this fuzzy number are the closed intervals:

$${}^{\alpha}M=\left[\bar{x}-z_{g(\alpha)}\frac{\sigma}{\sqrt{n}},\bar{x}+z_{g(\alpha)}\frac{\sigma}{\sqrt{n}}\right] \quad (2)$$

which are exactly the $(1-a)(1-\beta)$ confidence intervals for μ, where

$$g(\alpha)=\left(\frac{1}{2}-\frac{\beta}{2}\right)\alpha+\frac{\beta}{2},\quad\left(g:[0,1]\to\left[\frac{\beta}{2},0.5\right]\right)$$

and

$$z_{g(\alpha)}=\Phi^{-1}(1-g(\alpha))$$

The graph of this fuzzy number is presented in Fig. 1.

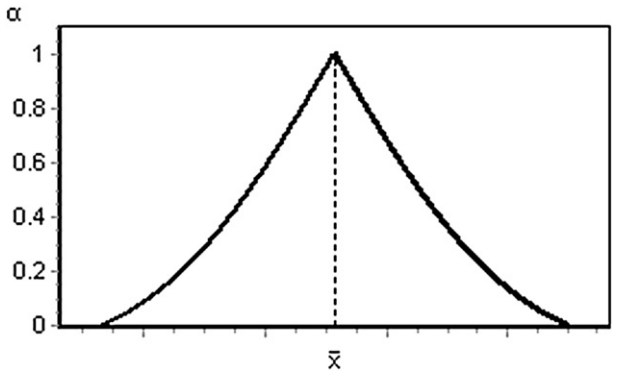

Fig. 1 – Non-asymptotic fuzzy mean estimator.

The fuzzy estimator consists of the triangle-shaped lines which are constructed by the discrete observations obtained from the empirical network data. Let us assume that the above graph is a fuzzy estimator from historical, non-DDoS data (say from the previous day). The mean value $\bar{x}$ specifying the peak of the graph divides the triangle into a left and a right side. From this point onward we need to estimate the mean arrival times of packets of the current, present traffic, t_c. We then place this value on the fuzzy estimator and if it is on the left-hand side (that is $t_c<\bar{x}$) then we record a DDoS attack. For $t_c>\bar{x}$ we can either consider a normal network operation, or alternatively we can assign a possibility of DDoS value, where this possibility increases the closer t_c is to $\bar{x}$ and depending on the security policy we can set a further threshold or actions, such as increase the logging or alert levels.

We now show how to calculate t_c. As a starting hypothesis, we consider that the traffic fits the Poisson density function

$$f(x)=P(X_t=x)=\frac{(qt)^x}{x!}e^{-qt}$$

which has distribution function $F(t)=1-e^{-qt}$

In this case q equals to the number of packet arrivals/second.

$$P(T<t)=1-e^{-qt}$$

We have to find t_c, such that

$$F(t_c)=1-e^{-qt_c}\le p,$$

where p is a given probability.

Solving this inequality, we take: $t_c\ge\ln(1-p)/-q$

We take in to account that$E(T)=qt$ and firstly we do the estimation $E(T)=\bar{t}=qt$.

Then, we take the confidence intervals for mean and we form the fuzzy estimator for t_c using equation (2).

Let $[l_\alpha,r_\alpha]$ be the α -cut for the fuzzy number $E(T)$.

Then, $[E(T)]_\alpha=[l_\alpha,r_\alpha]$ and hence, we find the $\alpha-cut$ for the fuzzy number t_c as follows:

$$[t_c]_\alpha=\left[\ln\left(\frac{1}{1-p}\right)\frac{1}{r_\alpha},\quad\ln\left(\frac{1}{1-p}\right)\frac{1}{l_\alpha}\right]$$

Upon detecting a DDoS attack, we move on to the second challenge, which is about identifying the offending IP

Daily Statistics for October 2011

Day	Hits		Files		Pages		Visits		Sites		kB F		kB In		kB Out	
1	381513	2.13%	312404	2.20%	88348	2.41%	6991	2.32%	5634	3.12%	5784608	1.95%	0	0.00%	0	0.00%
2	431619	2.41%	357051	2.51%	101510	2.77%	7420	2.46%	6050	3.35%	6837816	2.31%	0	0.00%	0	0.00%
3	806944	4.51%	638318	4.49%	139058	3.79%	12746	4.23%	10681	5.91%	12160045	4.11%	1	14.29%	23	14.29%
4	634967	3.55%	508776	3.58%	111268	3.04%	10622	3.52%	9402	5.20%	9330245	3.15%	0	7.14%	11	7.14%
5	754617	4.22%	595507	4.19%	142774	3.90%	11752	3.90%	10162	5.62%	11935422	4.03%	0	7.14%	11	7.14%
6	678318	3.79%	539276	3.79%	122348	3.34%	11642	3.86%	9819	5.43%	9804036	3.31%	0	7.14%	11	7.14%
7	620516	3.47%	499423	3.51%	130972	3.57%	11085	3.68%	8898	4.93%	9617851	3.25%	0	0.00%	0	0.00%
8	398848	2.23%	330758	2.32%	90487	2.47%	7547	2.50%	5933	3.28%	6537778	2.21%	0	0.00%	0	0.00%
9	461903	2.58%	379004	2.66%	96250	2.63%	8055	2.67%	6619	3.66%	7609757	2.57%	0	0.00%	0	0.00%
10	803530	4.49%	623215	4.38%	151999	4.15%	12837	4.26%	10510	5.82%	12447377	4.20%	1	14.29%	23	14.29%
11	736608	4.12%	584861	4.11%	153359	4.18%	12076	4.01%	10077	5.58%	11355650	3.83%	0	0.00%	0	0.00%
12	700520	3.92%	552844	3.89%	136271	3.72%	11832	3.93%	9668	5.35%	10770933	3.64%	0	7.14%	11	7.14%
13	688105	3.85%	521297	3.66%	119783	3.27%	11371	3.77%	10363	5.74%	12676426	4.28%	1	14.29%	23	14.29%
14	635013	3.55%	502331	3.53%	129285	3.53%	10405	3.45%	9403	5.20%	11434388	3.86%	0	7.14%	11	7.14%
15	397465	2.22%	317076	2.23%	97350	2.66%	6954	2.31%	5832	3.23%	7226049	2.44%	0	0.00%	0	0.00%
16	420596	2.35%	342945	2.41%	96444	2.63%	7110	2.36%	6205	3.43%	7090468	2.39%	0	0.00%	0	0.00%
17	726789	4.06%	571680	4.02%	139154	3.80%	11672	3.87%	10119	5.60%	11257399	3.80%	0	0.00%	0	0.00%
18	654189	3.66%	508592	3.58%	134089	3.66%	10764	3.57%	9299	5.15%	11135492	3.76%	0	0.00%	0	0.00%

Fig. 2 – Job seeking site statistics.

addresses as follows. In a specific time window (typically this is in the region of 1 s in order to satisfy the real time requirement) we calculate the density of the each unique IP address (that is the number of packets generated by unique IP) and from that we can recalculate the mean inter-arrival time t_c as described above, but for this time on a per-IP basis. In a similar manner, if t_c is below the mean of the fuzzy estimator, we classify the corresponding IP address as part of the DDoS. Naturally, this approach is expected to perform better in the case of botnets sending requests on a high rate.

4. Empirical evaluation

4.1. *Data sets*

We used the publicly available LLS_DDOS_1.0 DARPA Intrusion Detection Evaluation datasets (MIT DARPA, 2000) and also generated our own datasets. The primary data were generated by attacking a popular job seeking site residing on the university campus (Fig. 2). The site has around 8000 visits per day and is

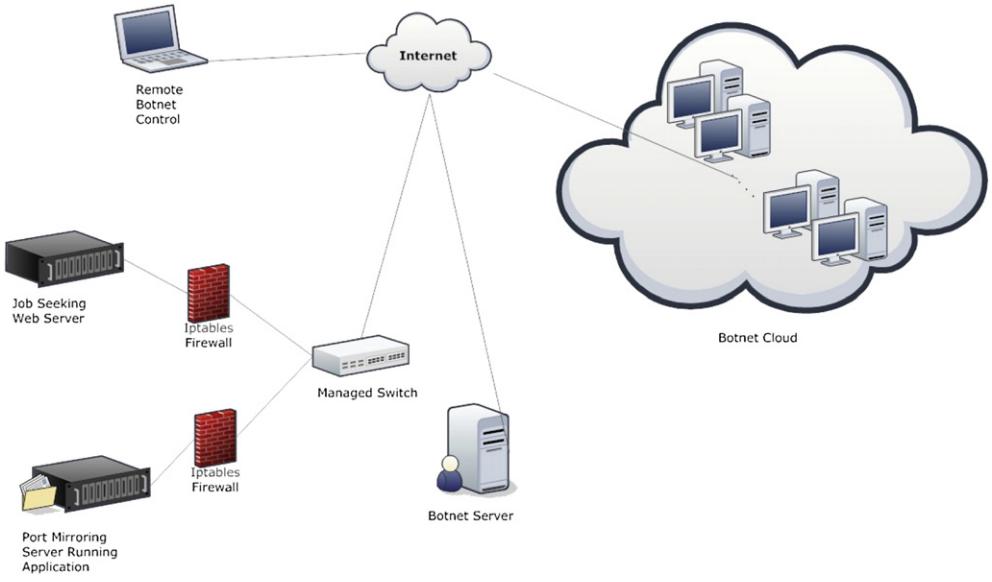

Fig. 3 – The testbed.

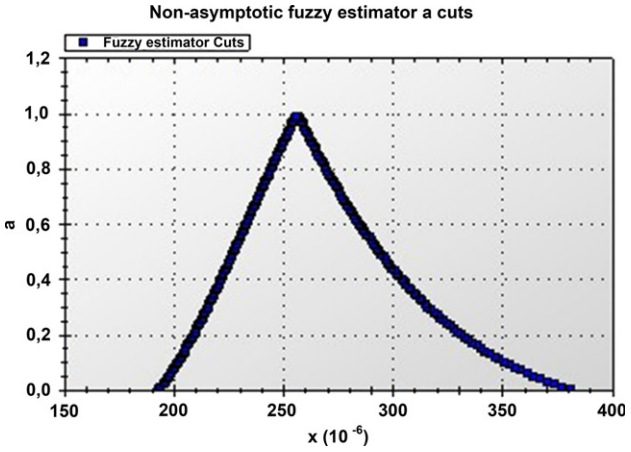

Fig. 4 – 4 sec of normal traffic t_c α-cuts.

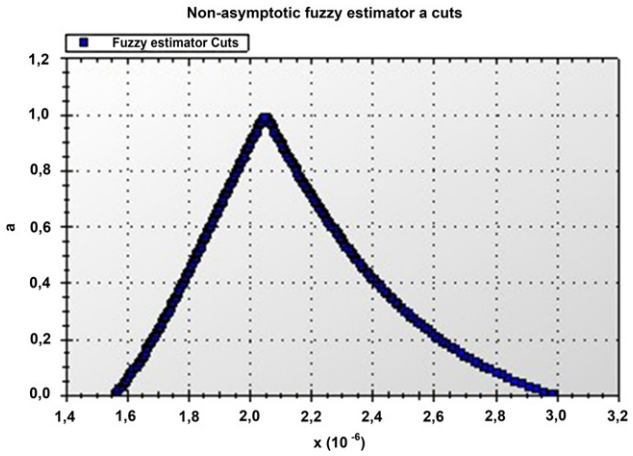

Fig. 6 – 4 sec DDoS traffic t_c α-cuts.

considered to be the most commercially successful graduate job seeking site on a national scale. The fact that the site is hosted on a university campus network was particularly suitable as we could emulate DDoS activity without causing any network bottlenecks and we were able to assess the effectiveness of the proposed method and more particularly its real time aspects.

The data was collected by mirroring the server's ethernet port and by capturing the inbound traffic on ports 80 and 443. This was considered to be the most appropriate approach as all other traffic was blocked at the firewall level.

We executed two attacks in different days and conditions, generating two datasets. The first day we attacked the server during a low visit period, whereas the second day we attacked the server during a high peak visit period. For our experiment we used hping and BlackEnergy Bot which is an HTTP-based botnet used primarily for DDoS attacks. Unlike most common bots, this bot does not communicate with the botnet master using IRC but the widely used web services. It also has the ability to encrypt the communication data with the server. The bot was setup in a fully controlled environment. The total number of bots we utilized was 6, communicating with the C&C Server (Fig. 3). For more information on the attack refer to Shaeles and Psaroudakis (2011).

4.2. *Empirical results*

t_c and α-cuts were calculated according to the approach described in Section 3. We calculated t_c for normal traffic during the busiest hours of the server. Then this attribute was converted to a fuzzy estimator and then the values were used to identify the IPs involved in the DDoS in the imported data as follows. Firstly we calculate the α-cut boundaries in line with Fig. 1 presented above. The peak of the curves denotes the expected mean value of t_c. This value essentially splits the graph into two areas. Values of t_c residing on the left side of $\overline{t_c}$. are considered to be DDoS attacks. Values of t_c residing on the right side of $\overline{t_c}$.ve a degree of possibility for a DDoS attack.

More analytically the α-cuts were empirically obtained as follows. We split normal traffic data into files with 500, 1000, 5000, 10,000, 20,000, 30,000, 40,000, 50,000, 100,000, 150,000, 200,000 network packets – with each packet denoting a network event – and we produced t_c graphs for each of the files; the split allows us to consider the differences of the traffic as we can get a finer granularity of the $\overline{t_c}$. In the Figures below we present graphs that show in our sample of 4 s of

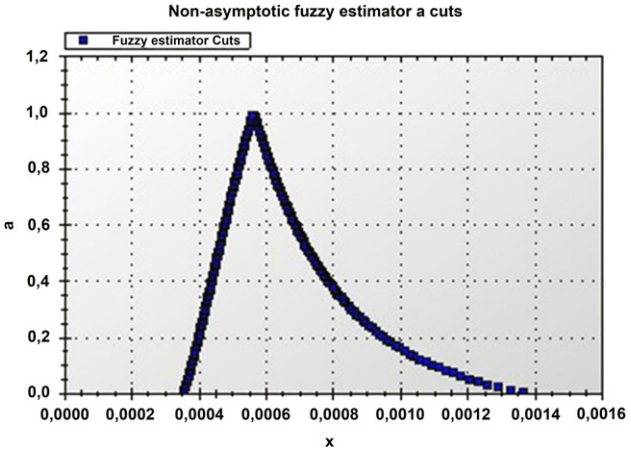

Fig. 5 – 12 sec normal traffic t_c α-cuts.

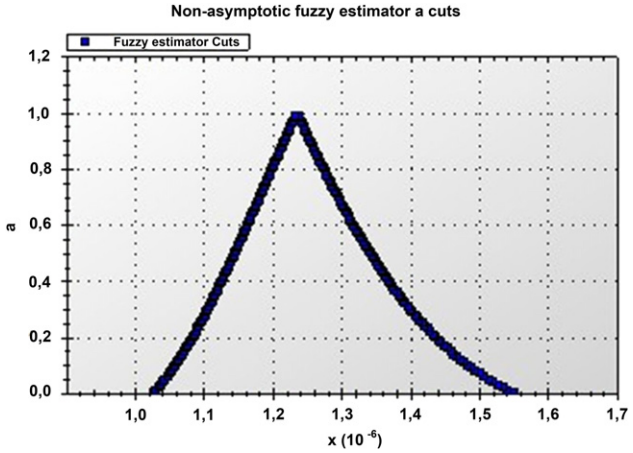

Fig. 7 – 12 sec DDoS traffic t_c α-cuts.

normal traffic corresponding to approximately 1000 packets (Fig. 4) and 12 s normal traffic on a lesser busy period, corresponding to the same number of packets (Fig. 5). It should be noted that the orders of $\overline{t_c}$.e comparable, as they are shown in a different scale of the *x*-axis.

In contrast, the 4-s DDoS traffic contains more than 100,000 packets in the csv file and the 12 s of DDoS traffic is in the area of 610,000 packets in the file. The graphs or DDoS traffic are shown in Fig. 6 and Fig. 7.

From visually inspecting the above graphs we can establish that for up to a period of 2 s, the curve forms for DDoS and normal traffic are not particularly distinguishable; however, in the case of a DDoS we obtain considerably smaller values. If we increase the sample size we obtain the results as shown in Fig. 4 which is expected as all our traffic is closer to $\overline{t_c}$. .e obtained similar results with the DARPA dataset. We split the dataset (LLS_DDOS_1.0-inside.dump) into chunks of 5000, 10,000, 20,000–100,000, 150,000 and 200,000 packets which corresponded to approximately 2 min to 1.5 h periods. The import time for each chunk ranged from less than half a second to 23 s. We established that 5000 packets for this dataset were sufficient to perform successful detection. The detection time was performed in 2 s.

4.3. *Performance, accuracy and limitations*

The execution of the implemented algorithm for our datasets took around 1 min to import 610,000 packets and 40 s to analyze them and return potential IPs that participate in the DDoS attack (Shaeles and Psaroudakis, 2011) (Fig. 8). The system used was Intel Core Quad Q9950 with 8 GB of RAM. Both in terms of performance and accuracy, the proposed approach provided significant results as it could identify successfully 3/5, 5/5 or 5/6 IPs (depending on the dataset chunk) involved in the DDoS in 1.5–5.9 s respectively. The corresponding packet count ranges from 5000 to 20,000.

Following our tests we can see that successful DDoS detection is possible after collecting about 5000 network events but best results occur after 20,000 packets. With 20,000 packets the computation was completed in 1.8 s. With respect to training, the detection requires a minimum of 5000 packets or 2 s worth of traffic. During a DDoS flood, 2 s of traffic may correspond to up to 100,000 packets. This means that 20,000 packets will be captured in 400 ms. As such, the total time for detection is expected to be in the region of 2.4 s.

With respect to the DARPA dataset the proposed method detected successfully the 2 attacking IPs and 4 spoofed IPs as false positives. According to the dataset description there were three attacking IPs, but the third one did not have any traffic to the victim server in the scenario we investigated and therefore it was non-surprisingly not detected. Another point was that with the DARPA dataset the attacks were on various ports apart from port 80. Since the proposed method depends only on the arrival time, the attack was detected. As other ports (such as telnet and ftp) definitely do not follow a Poisson model, our results confirm the independence from the Poissonity requirement. It should also be noted that the historical data of the DARPA dataset were limited. We used 4 s worth of packets for the training which was sufficient to yield fairly accurate results. According to the DARPA dataset specifications, there were three offending IPs in total. Our method

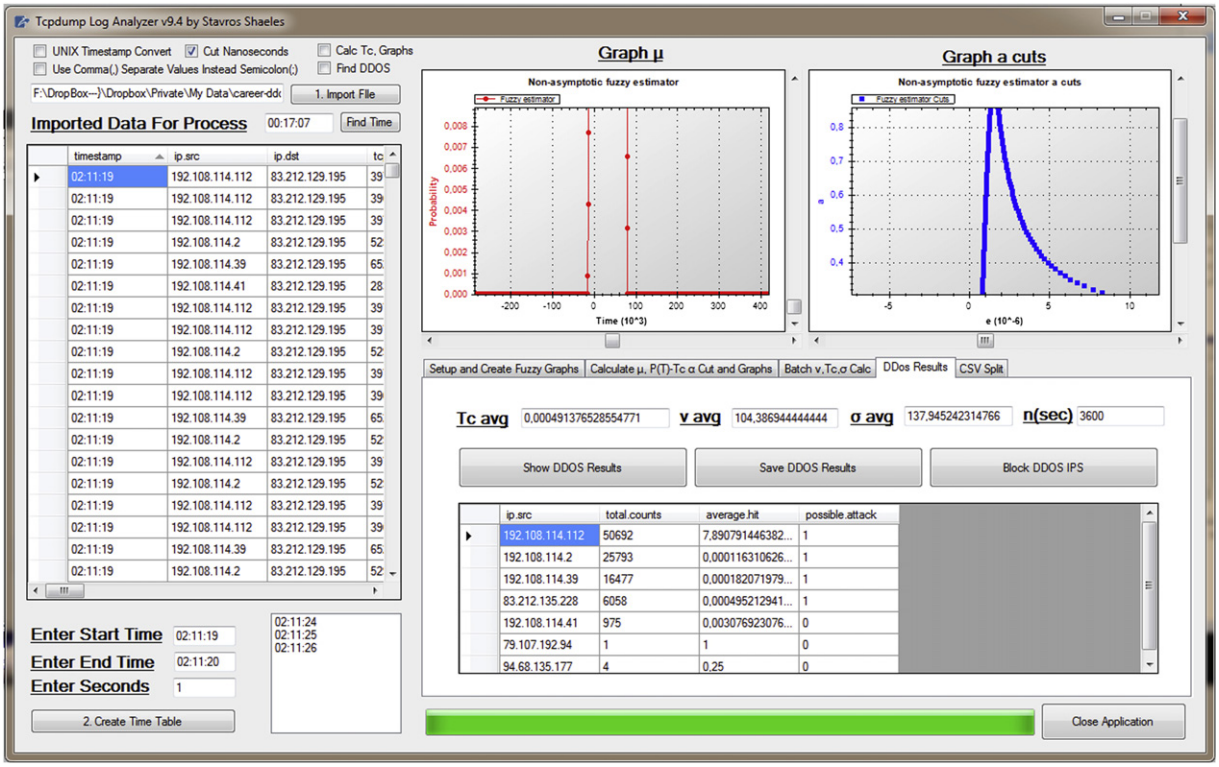

Fig. 8 – Results from 4 s (100,000 packets).

Table 1 – Dataset summary and findings.

Dataset	Time window (range)	Number of packets (range)	Analysis time (range)	Number of IPs found	Analysis time vs. no. of packets correlation coefficient	r^2
Low traffic period (botnet)	1–4 sec	5K–100K	1–6 ms	5/6 with 40 K packets, 2sec training 5/6 with 20K packets, 5K packets training	0.994625245	0.9892
High traffic period (hping)	38–95 sec	5K–100K	79–131 ms	2/2 for 10,000 packets and over.	0.995133989	0.9902
MIT-DARPA LLS_DDOS_1.0	228–1933 sec	5K–70K	122–10K ms	2/3 with 5000 packets	0.983643161	0.9675

detected successfully the two IPs, but after inspecting the dataset we observed that the third IP communicated only with the attack host rather than the victim server. As such, the effective success rate was 100%.

Table 1 presents a summary of the datasets and some quantitative attributes. There is a strong linear relation between the number of packets and analysis time. The total response time is proportional to the total number of unique IPs. In Figs. 9 and 10 we show the representative relationships for our two datasets respectively.

Comparing this method with other published research, we need to highlight that all papers that we consulted on real time DDoS detection display their time performance abilities, but most of them do not explicitly state the data import delays. Naturally, data import delays are expected to be independent of the actual detection algorithm performance, but we argue that when proposing a practical real time solution, the total time (or computational complexity) needs to be included, as the data import and preparation needs may be different for each detection algorithm. For instance, our implementation requires that the data are sorted by IP numbers. Although we use an efficient sorting algorithm, the overheads due to the sorting complexity are present and cannot be avoided. As such, the total response times presented above include also data import delays. For example, Gavrilis and Dermatas (2005) who develop an efficient and effective neural network classifier, claim DDoS detection within a 6 s window, but there is no information on the total time. If we assume that this 6 s window is the best case scenario, then our proposed approach is about 2.5 times fold more efficient. Such significant difference is anticipated as our approach uses only one feature (arrival time).

In general the proposed method is prone to false positives for spoofed IPs or NAT arrangements. This is expected because of the limited granularity of attributes the proposed method has. We prefer real time detection methods to be susceptible to false positives which can later be corrected by other means (such as packet inspection), rather than the opposite. As there is no silver bullet for DDoS detection, in production environments we need integrated threat management systems including a component which focuses on the real timeliness of DDoS detection. IP spoofing would therefore need to be addressed by augmenting or integrating the proposed methods with other ones (see for example MIT's spoofer project, Beverly and Bauer, 2005) as well as network and firewall configurations (for example, block the 10.0.x.x

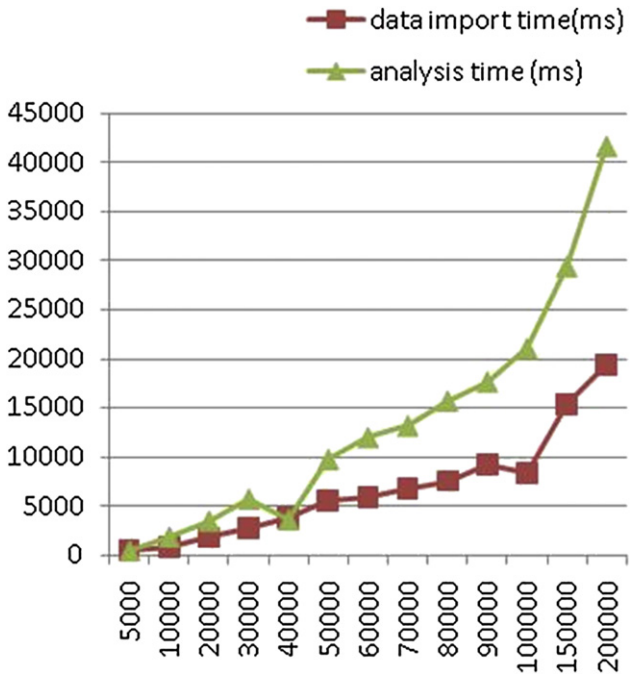

Fig. 9 – Processing overheads for botnet dataset (time vs. number of packets).

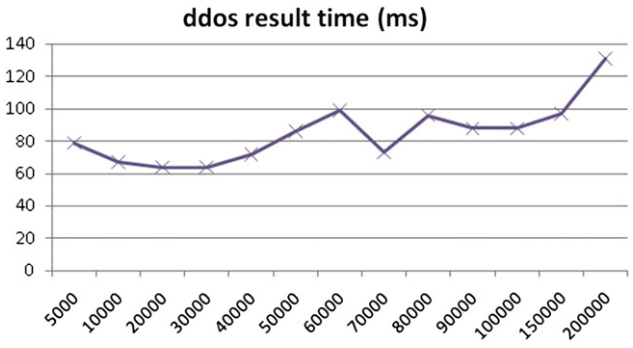

Fig. 10 – Total DDoS response time for syn flood attack using hping dataset (time vs. number of packets).

and 192.168.x.x spoofed packets, or through packet inspection).

5. Conclusions and future work

We proposed an approach for detecting a DDoS attack using a fuzzy estimator on the mean time between network events. DDoS detection is particularly challenging in sites with a large average number of hits, as the detection methods typically generate false positives and are not practical. Yet, when a DDoS attack is detected, it is imperative to identify the offending hosts in a timely manner in order to offer added value intrusion response services. The proposed method is capable of detecting a DDoS and identifying the malicious IPs before the victim service suffers from exhaustion of resources due to the attack. The empirical evaluation showed that the proposed method can have an over 80% success rate (which corresponds to 20% Type-II errors).

The method can run on a mid-range PC and can provide near-real time DDoS detection. However, its full potential would be appreciated if run on a higher end PC or by employing the parallel architecture of graphics cards. Currently we are implementing a version of the algorithm which will be compatible to NVidia's CUDA framework and we are also considering a non-preemptive OS kernel. The non-preemptive kernel is required in order to improve the import and analysis times.

Although the proposed method uses the arrival time as the main metric for discriminating benign from DDoS traffic, it is expected that additional features will substantially improve the accuracy and possibly speed of the proposed method, as it will require a smaller amount of data. In general as this method is very accurate in detecting the DDoS attack and fairly accurate for identifying the offending IP addresses within strict time limits that allow the system to respond in real time, the identification challenge can be further refined by the application of other methods. The proposed method depends upon the time parameter (and more specifically on packet inter-arrival times) so a finer granularity by introducing other aspects (such as packet parameters, protocols and so forth) is expected to improve the identification accuracy. A short term, ongoing research activity is the evaluation of the identification ability of the method in large botnets and establish the thresholds where false negatives become significant.

In the case of flash crowds we expect that the method will detect a DDoS but will not be able to classify any IP as an offending one. Flash crowds typically involve many IPs and do not make many requests per second per IP. Therefore the method can explicitly detect flash crowd activity if it will detect a DDoS but no IPs. Such analysis deserves a future research line.

In this paper we attempt to relax the strict requirements of a model as this is problematic, instead of trying to find a better model which we conjecture that it would be a futile exercise. Nevertheless, we need to assume some model as a point of reference, and the most obvious and popular one was Poisson. Possibly our proposed method will also work with other models, which is an area of future research.

Acknowledgments

The authors are indebted to Ioannis Psaroudakis for his help on setting up the test environment and collecting the primary data. Research supported in part by the Greek GSRT/CO-OPERATION/SPHINX Project (09SYN-72-419).

REFERENCES

Ari, I., Hong, B., Miller, E., Brandt, S. Long, D. Managing Flash Crowds on the Internet. In: Proceedings of the 11TH IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer Telecommunications Systems (MASCOTS'03); 2003.

Arlitt M, Williamson C. Internet Web servers: workload characterization and performance implications. IEEE/ACM Transactions on Networking 1997;5(5):631–45.

Arun Raj Kumar P, Selvakumar S. Distributed denial of service attack detection using an ensemble of neural classifier. Computer Communications 2011;34(11):1328–41.

Beverly R, Bauer S. The spoofer project: inferring the extent of source address filtering on the internet. USENIX SRUTI; 2005.

Chrysafis KA, Papadopoulos BK. Cost–volume–profit analysis under uncertainty: a model with fuzzy estimators based on confidence intervals. International Journal of Production Research 2009;47(21).

Douligeris C, Mitrokotsa A. DDoS attacks and defense mechanisms: classification and state-of-the-art. Computer Networks 2004;44(5):643–66.

Feinstein L, Schnackenberg D, Balupari R, Kindred D. Statistical approaches to DDoS attack detection and response. In: Proceedings of DARPA Information Survivability Conference and Exposition, vol. 1; 2003. p. 303–14.

Gavrilis D, Dermatas E. Real-time detection of distributed denial-of-service attacks using RBF networks and statistical features. Computer Networks 2005;48(2):235–45.

Goth G. The Politics of DDoS attacks. IEEE Distributed Systems Online 2007;8. art. no. 0708–o8003.

Gribble, S. and Brewer, E. System design issues for internet middleware services: deductions from a large client trace. In: Proceedings of the USENIX Symposium on Internet Technologies and systems, California; 1997, pp. 207–218.

Guerin CA, Nyberg H, Perrin O, Resnick S, Rootzen H, Starica C. Empirical testing of the infinite source Poisson data traffic model. Stochastic Models 2003;19(2):151–200.

Jenik A. Cyberwar in Estonia and the Middle East. Network Security 2009;2009(4):4–6.

Jin S, Yeung D. A covariance analysis model for DDoS attack detection. In: Proc. IEEE International Conference on Communications; 2004.

Katos V. Network intrusion detection: evaluating cluster, discriminant, and logit analysis. Information Sciences 2007; 177(15):3060–73.

Lakhina A, Crovella M, Diot C. Mining anomalies using traffic feature distributions. In: Proceedings of ACM SIGCOMM; 2005.

Lee S, Chung B, Kim H, Lee Y, Park C, Yoon H. Real-time analysis of intrusion detection alerts via correlation. Computers & Security 2006;25(3):169–83.

Lee SM, Kim DS, Lee JH, Parka JS. Detection of DDoS attacks using optimized traffic matrix. Computers & Mathematics with Applications 2011.

Li M, Chi CH, Jia W, Zhao W, Zhou W, Cao J, et al. Decision analysis of statistically detecting distributed denial-of-service

flooding attacks. International Journal of Information Technology and Decision Making 2003;2(3):397–405.
Li L, Lee G. DDoS attack detection and wavelets. Computer Communications and Networks 2003:421–7.
Li B, Xie S, Qu Y, Keung G, Lin D, Liu J, et al. Inside the new coolstreaming: principles, measurements and performance implications. IEEE Infocom 2008a.
Li M. An approach to reliably identifying signs of DDoS flood attacks based on LRD traffic pattern recognition. Computers and Security 2004;23(7):549–58.
Li Y, Guo L, Tian Z, Lu T. A lightweight web server anomaly detection method based on transductive scheme and genetic algorithms. Computer Communications 2008b;31:4018–25.
Mirkovic J, Reiher P. A taxonomy of DDoS attack and DDoS defense mechanisms. ACM SIGCOMM Computer Communications Review 2004;34(2):39–54.
MIT DARPA. Intrusion detection evaluation dataset. Available from: http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/data/2000data.html; 2000.
Oshima S, Nakashima T, Sueyoshi T. DDoS detection technique using statistical analysis to generate Quick response time, International Conference on Broadband. Wireless Computing, Communication and Applications 2010:672–7.
Park, C., Shen, H., Marron, J.S., Hernandez-Campos, F., Veitch, D. Capturing the elusive poissonity in web traffic. In: 14th IEEE International Symposium on modeling, analysis, and Simulation of Computer and Telecommunication systems, 2006. pp. 189–196.
Patcha A, Park JM. An overview of anomaly detection techniques: existing solutions and latest technological trends. Computer Networks 2007;51(12):3448–70.
Paxson V, Floyd S. Wide area traffic: the failure of Poisson modeling. IEEE/ACM Trans.. on Networking 1995;3(3):226–44.
Ramamoorthi A, Subbulakshmi T, Mercy Shalinie S. Real time detection and classification of DDoS attacks using Enhanced SVM with string kernels, In: IEEE-International Conference on Recent Trends in information Technology; 2011. pp. 91–96.
Rincón D, Sallent S. On-line segmentation of non-stationary fractal network traffic with wavelet transforms and Log-likelihood-based statistics. Lecture Notes in Computer Science 2005;3375:110–23.
Sengar H, Wang H, Wijesekera D, Jajodia S. Detecting VoIP floods using the Hellinger distance. IEEE Transactions on Parallel and Distributed Systems 2008;19(6):794–805.
Shaeles SN, Psaroudakis ID. A study of a botnet creation process and the impact of a DDoS attack against a web server. Hakin9 Extra – Botnets 2011:8–12. issue 5.
Shon T, Kim Y, Lee C, Jongsub M. A machine learning framework for network anomaly detection using SVM and GA, In: Proc. Of Systems, Man and Cybernetics (SMC) Information Assurance Workshop; 2005. pp. 176–183.
Siraj A, Vaughn RB, Bridges SM. Decision making for network health assessment in an intelligent intrusion detection system architecture. International Journal of Information Technology and Decision Making 2004;3(2):281–306.
Tang J, Cheng Y, Zhou C. Sketch-based SIP flooding detection using Hellinger distance. Global Telecommunications Conference GLOBECOM; 2009. pp. 1–6.
Tsironis LC, Sfiris DS, Papadopoulos BK. Fuzzy performance evaluation of workflow stochastic petri nets by means of block reduction. IEEE Transactions on Systems Man and Cybernetics Part A – Systems and Humans 2010;40(2):352–62.
Wang J, Yang G. An intelligent method for real-time detection of DDoS attack based on fuzzy logic. Journal of Electronics (China) 2008;25(4):511–8.
Wang H, Zhang D, Shin K. Detecting SYN flooding attacks. In: Proceedings of the IEEE Infocom, New York, NY; 2002.
Wei W, Dong Y, Lu D, Jin G. Combining cross-correlation and fuzzy classification to detect distributed denial-of-service attacks. Lecture Notes in Computer Science, LNCS 2006;3994: 57–64.
Xia Z, Lu S, Li J. Enhancing DDoS flood attack detection via intelligent fuzzy logic. Informatica 2010;34:497–507.
Yeung DS, Jin A, Wang X. Covariance-matrix modeling and detecting various flooding attacks. IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans 2007;37(2).
Yu S, Zhou W, Doss R. Information theory based detection against network behavior mimicking DDoS attacks. IEEE Communications Letters 2008;12(4):318–24.

Stavros N. Shaeles is a member of the IEEE and the IEEE Computer Society. He received his diploma in Electrical and Computer Engineering in Democritus University of Thrace in 2007. Currently he is a PhD candidate in the research area of computer security and has many years of industrial experience in programming and computer security administration.

Vasilios Katos is Assistant Professor of Information and Communications Systems Security at the Department of Electrical and Computer Engineering of Democritus University of Thrace in Greece. Prior to his current post he was Principal Lecturer at the School of Computing at the University of Portsmouth where he participated in the development of the interdisciplinary Masters course MSc in Forensic IT. He has worked in the industry as a security consultant and expert witness in information systems security. His research interests are in information security, privacy, digital forensics and incident response.

Alexandros S. Karakos received the Degree of Mathematician fromthe Department of Mathematics from Aristoteles University of Thessaloniki, Greece and the Maitrise d' Informatique from the university PIERRE ET MARIE CURIE, Paris. He completed his PhD studies atuniversity PIERRE ET MARIE. He is Assistant Professor at the Dept. of Electrical and Computer Eng., Democritus University of Thrace,Greece. His research interests are in the areas of Distributed systems, data mining, data analysis and programming languages.

Basil K. Papadopoulos is Professor of Mathematics and Statistics at Democritus University of Thrace, Section of Mathematics, Department of Civil Engineering. Interest topics: Topology, Mathematical Modeling, Fuzzy logic with applications in Engineering. He has over eighty papers published in various journals, in all the above topics both as a sole author or a co-author. He has received the degree of Mathematics from the Aristotele University of Thessaloniki and a PhD from Democritus University of Thrace through scholarship State Scholarship Foundation.