IJICTR International Journal of Information & Communication Technology Research

Volume 8- Number 2- Spring 2016 (15-21)

DDoS Traffic Modeling, Identification and Attack Mitigation to Cloud Hosted servers by Drift plus Penalty methods

Ahmad Salahi
Iran Telecommunication Research Center
Tehran, Iran
ahmadsalahi@yahoo.com ; salahi@itrc.ac.ir

Received: September 23, 2015- Accepted: January 27, 2016

Abstract — **During DDoS attack to a cloud hosted server, to counter the attack, more resources should be assigned to it. In this paper we first develop a mathematical formula for input packet rate during DDoS attack, and propose a method to identify the botnet that created the attack. We introduce two algorithms for resource assignments to protect cloud hosted servers. The drift plus penalty algorithm minimizes the average cost of resource assignment, and stabilizes the queue size. The modified version of this algorithm is drift plus extended penalty, which minimizes the average cost and compensate penalty function by considering delay.**

Keywords. DDoS generated traffic modeling. Cloud computing. Drift plus penalty. Drift plus extended penalty. Dynamic resource allocation

I. INTRODUCTION

A denial of services attack is an attempt to make a server or network resource unavailable to its users. If this attack launched by more than one source, then it is called distributed denial of service or DDoS attack. Today, most companies provide their services through their web sites, and DDoS attack can make their web sites unavailable, leading to financial and reputation damage. Cloud computing is defined as a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources(e.g., networks, servers, storage, application and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [6].

Because of attractive business models provided by cloud service providers (C.S.P), cloud computing is growing fast in I.T. industry. Some of the benefits of cloud computing are: reduction of cost, rapid deployment, easy administration, no new hardware to buy, no software updates or annual maintenance, pay-as-you-grow subscription pricing, built-in scalability , redundancy, and anywhere access via an Internet connection [13].

Challenges in cloud computing are reliability, security, cost, complexity, regulation, legal issues, performance, migration, lack of standards , reversion and privacy issues [4],[14].

Since cloud service providers, possess huge amount of resources, it is very difficult to beat them by DDoS attacks on their customers. We can divide related works into three different areas. First DDoS attack mitigation in cloud, in [1], [2], it is suggested that DDoS attack and defense against it in cloud is resource competition, if cloud service providers, provide sufficient resources to counter a DDoS attack, it will not be successful, also it is assumed, input traffic during DDoS attack have Poisson distribution and derived formulas to assign security modules to protect host servers in cloud. In [22] a survey of mitigation against EDoS attacks in cloud has been discussed. Second DDoS traffic modeling, in [19] a non-Gausian and long memory statistical characterization of internet traffic proposed, this empirical model relevantly

describes a large variety of internet traffic, both legitimate or illegitimate (DDoS traffic). In [20] a survey of various methods to identify legitimate and illegitimate traffic is provided, the main focus is to distinguish DDoS traffic from normal traffic.

In [16] it is assumed internet traffic is better modelled by Pareto Probability distribution function which has a heavy tail that is not shown in Poisson.

Third area is botnet identification. In [21] a comparative study analysis of botnet identification methods was proposed.

In normal condition when there is no DDoS attack ongoing, there exists a constant average rate input packets to server. Sources of traffic are independent, inter arrival times are independent and identically distributed, so it is reasonable to assume probability distribution of input traffic is Poisson.

During DDoS attack, sources of attack traffic are bots of a botnet, with the same DDoS attack code, inter arrival times of packets received by victim are highly correlated and are not independent, identically distributed, so the assumption of any definite probability distribution function is very difficult [3]. Some attempts to simulate DDoS traffic already [7] have been done. To develop new algorithms to defend against DDoS attacks, we should not consider any specific probability distribution for DDoS attack traffic. We have used queue length and number of arrivals as measures for any control actions to assign resources to the servers.

Since using more resources will cost more, we have used a penalty function and tried to minimize it. In this paper two algorithms developed for resource assignment, to counter DDoS attack traffic, neither needs knowledge of DDoS traffic probability distribution, these algorithms are: Drift plus penalty [9], [10], [11], and [12] which minimize average cost and stabilize queue length. Drift plus extended penalty, which minimizes an extended penalty function by considering cost and delay.

Remainder of this paper is organized as follows: In chapter II, we develop a formula for the rate of packets generated by a DDoS botnet, and in chapter III a method to identify the botnet that created the attack. In chapter IV two different types of drift plus penalty algorithms were introduced. In chapter V, we provide the performance evaluation of these methods and finally in chapter VI conclusions will be provided

II. DDoS GENERATED TRAFFIC MODELING

In normal condition, when there is no DDoS attack ongoing, input traffic can be assumed to have Poisson distribution with constant average rate λ packets per second.
If we use discrete time slots $t \in \{0, 1, \dots\}$ and the duration of each time slot is T seconds, then the probability of arrival of k packets in one period T is:

$$Pr\{X(t) = k\} = e^{-\lambda T}\frac{(\lambda T)^k}{k!} \quad k = 0,1,.. \qquad (1)$$

During DDoS attack, number of packet arrivals in one time slot $t \in \{0, 1, \dots\}$ is the addition of two components:

- Normal Poisson traffic X(t)
- DDoS traffic A(t)

Generation of random numbers with Poisson distribution is simple [5], here, we try to develop a formula for A(t). We assume only one botnet attacked the victim during DDoS attack, so all attacking bots (live bots) belong to one botnet and have the same DDoS attack code. Because the same code generates attack packets, with high probability the rate of attack packets generated by each bot is the same. Since bot master sends the attack message in which the time to start the attack and address of victim is mentioned all participating bots start sending their attack packets almost synchronously.

-$N(\ t)$ is the number of packets received by victim at each time slot t, where all live bots send just one packet synchronously.

-$N'_{total}(t)$ is the total rate of packet received by victim at time t.

Fig. 1 shows location of victim and those bots, that their attack packets receive at the same time to victim. We have used the terminology listed in Table 1.

If each of live bots sends one packet synchronously, then the rate of packets N'_τ received by victim can be computed as follows, Figuer.1.

$$d^2N = \rho(\varphi, \emptyset) \times R\ sin\varphi d\emptyset \times Rd\varphi, \qquad 0 \le \varphi \le \pi$$
$$0 \le \emptyset \le 2\pi$$

$$K(\varphi) = \int_0^{2\pi} \rho(\varphi, \emptyset) \times d\emptyset$$

$$dN = R\ sin\varphi \times Rd\varphi \times K(\varphi) \qquad 0 \le \varphi \le \pi$$

$$R\varphi = c\tau$$

$$dN = c\ sin\frac{C}{R}\tau \times K(\frac{C}{R}\tau)\ \times Rd\tau \qquad 0 \le \tau \le \frac{\pi R}{C}$$

Table 1. Parameters and their values used for DDoS attack

Symbol	Description	Value(unit)
L	Average number of bots	1000
r	Rate of packet generation by a bot	100 pps
TA	Duration of attack	300 s
λ	Rate of non- attack packets	1000 pps
TN	Duration of non-attack	20 s
R	Radius of earth	6000 km
c	Velocity of movement of packets	12000π km/s
N_a (.)	Number of alive bots	poissrnd(1000,1,1)
μ	Service rate of each security module	14000 pps
φ	Polar angle	Degree
∅	Azimuthal angle	Degree
$\rho(\varphi,\emptyset)$	Density of live bots	bot/km^2
NB	Total number of bots	bot
a(t)	Rate of total packets received by victim	Packet per second
τ	The delay between sending a packet from a bot to receiving it by victim.	

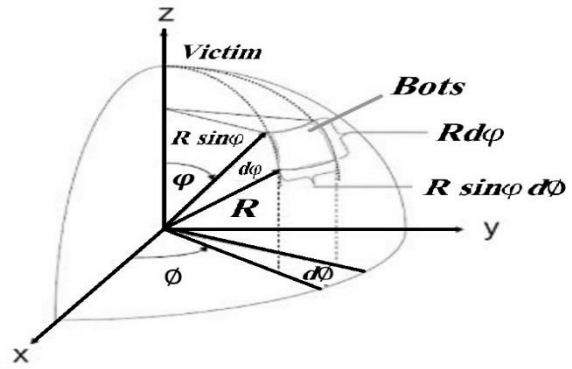

Fig. 1. Location of victim and some of bots that their attack packets receive at the same time to victim.

$$N'_\tau = \begin{cases} Rc\ sin\frac{C}{R}\tau \times K(\frac{C}{R}\tau) & 0 \le \tau \le \frac{\pi R}{C} \\ 0 & else\ where \end{cases} \quad (2)$$

If botnet considered as a linear system, then N'_τ is similar to its impulse response, so it can be used as a kind of identity for the botnet.

If the same botnet attacks another victim at polar coordinate $(\varphi_1, \emptyset_1)$,

and we make a linear transformation $\varphi' = \varphi - \varphi_1$ and $\emptyset' = \emptyset - \emptyset_1$.

Then $K(\varphi')$ will become $K(\varphi - \varphi_1)$, which is a linear shift of $K(\varphi)$ and

$$N'_\tau = \begin{cases} Rc\ sin\frac{c}{R}(\tau - \tau_1) \times K\left(\frac{c}{R}(\tau - \tau_1)\right) & \tau_1 \le \tau \le \frac{\pi R}{c} + \tau_1 \\ 0 & else\ where \end{cases} \quad (3)$$

Where $\tau_1 = \frac{\varphi_1 R}{c}$.

Duration of attack is *TA* seconds and each bot sends packets with the rate *r* packets per second. Therefore at time *t*

$$\tau_i = t - \frac{i}{r}\ , i = 0, 1 \ldots TA * r - 1.$$

Then the total rate of reception of packets $N'_{total}(t)$ by victim at time t is:

$$N'_{total}(t) = \sum_{i=0}^{r \times TA - 1} N'_{\tau_i}$$

$$= \sum_{i=0}^{r \times TA - 1} N'(t - \frac{i}{r}) \qquad 0 \le t - \frac{i}{r} \le \frac{\pi R}{c}$$

(4) *r* is usually greater than one hundred.

If we use $v = \frac{i}{r}$ $\quad \Delta v = \frac{i+1}{r} - \frac{i}{r} = \frac{1}{r}$:

$$N'_{total} \cong r \int N'(t - v)dv \quad 0 \le t - v \le \frac{\pi R}{c} \quad (5)$$

To find the range of integral, we can divide the range of t into three parts as follows:

$$0 \le t \le \frac{\pi R}{c} \Rightarrow 0 \le v \le t$$

$$\frac{\pi R}{c} \le t \le TA \Rightarrow t - \frac{\pi R}{c} \le v \le t \quad (6)$$

By substituting the above ranges for v into integral in (5):

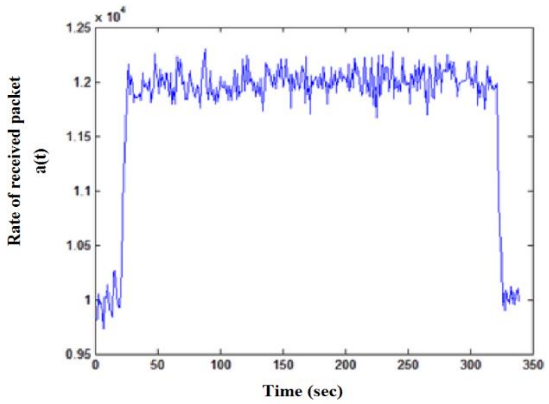

Fig.2. The total rate of packets received by victim.

Note: $N_a(.)$ is the returned number when calling poissrnd (L, 1, 1)

$$N'_{total}(t) \cong \begin{cases} rN(\ t) & 0 \le t \le \frac{\pi R}{c} \\ rN\left(\frac{\pi R}{c}\right) & \frac{\pi R}{c} \le t \le TA \\ rN\left(\frac{\pi R}{c}\right) - rN(t - TA) & TA \le t \le TA + \frac{\pi R}{c} \end{cases} \quad (7)$$

Where $N(0\ ) = 0$, and $N\left(\frac{\pi R}{c}\ \right) = N_a(.)$.

In simplest case, $\rho(\varphi, \emptyset)$ is uniformly distributed all over the world and $\rho(\varphi, \emptyset) = \rho = N_a(.)/4\pi R^2$ where $N_a\ (.)$ is the number of live bots in botnet and by integrating from (2), then:

$N(\tau) = \frac{1}{2} N_a\ (.)\left(1 - cos\frac{C}{R}\tau\right)$, and by substituting it in (7),

$$N'_{total}(t) \cong \frac{1}{2} r N_a\ (.) \begin{cases} 1 - cos\frac{c}{R}t & 0 \le t \le \frac{\pi R}{c} \\ 2 & \frac{\pi R}{c} \le t \le TA \\ 1 + cos\frac{c}{R}(t - TA) & TA \le t \le TA + \frac{\pi R}{c} \end{cases} \quad (8)$$

The total number of packets received by victim at time slot t is:

$$a(t) = X(t) + A(t)$$

$A(t) = N'_{total}(t).T$ and T, is the duration of one time slot.

If botnet has *NB* members, and *p* be the probability that a bot be alive and participates in DDoS attack, then

$$P\{N_a(.) = k\} = \binom{NB}{k}(p)^k(1 - p)^{NB - k} \quad (9)$$

$L = NB \times p$ is the average number of alive bots. The number of bots (NB) in important botnets is well above millions [17]. For the following reasons a small part of these bots are participating in attack:

- Compromised computers (bots) are few hours in a day are online and can participate in an attack.

-A large percent of bots detected and removed

-To prevent detection of bots only a small part of them invited to participate in an attack. For example in [18] the size of medium large DDoS attacks are around 30Mpps, if we assume average rate of attack by a bot is 1000pps then the number of participating bots are around 30000. If *p* be very small and *NB* be very high,

then Binomial distribution can be approximated by Poisson distribution [5], and

$$Pr\{N_a(.) = k\} = e^{-L}\frac{(L)^k}{k!} \qquad k = 0, 1, 2, \ldots \quad (10)$$

Fig. 2 shows $a(t)$ versus time, parameters used are listed in Table 1.

III. IDENTIFICATION OF BOTNET

We can distinguish three independent parameters of DDoS attack, $N_a(.)$, r and $K(\varphi)$, which are number of alive bots, rate of packets send to victim by each alive bot and φ_ density of bots in the world. Victims of DDoS attack can record the rate of incoming packets $a(t)$ and draw a graph like Fig. 2. If they can derive N'_τ (or $K(\varphi)$) from $a(t)$ then perhaps they can identify the botnet from existing library of previous DDoS attacks. If we look at (7) we see $N'_{total}(t) = rN_a(.)$, when $\frac{\pi R}{c} \le t \le TA$ which means it does not depend on density of bots, so we focus on the start of the attack, namely when $N'_{total}(t) = rN(\ t), \quad 0 \le t \le \frac{\pi R}{c}$.

We divide the duration $\frac{\pi R}{c}$ into k time slots, each of them with period $\frac{1}{r}$, therefore $k = \frac{\pi R r}{c}$.
Because $a(t) = X(t) + rN(\ t).T$, if we take $t = \frac{j}{r}, j = 1, \ldots, k$, and $T = \frac{1}{r}$, then

$$a(j) = X(j) + N(j),\ j = 1, 2 \ldots, k \qquad (11)$$

$$a(j) - a(j-1) = X(j) - X(j-1) + N(j) - N(j-1) \quad j = 2, \ldots, k \qquad (12)$$

Since $X(j) \sim$ Poisson$\left(\frac{\lambda}{r}\right)$,

$X(j) - X(j-1) \sim$ Skellam $(0, \frac{2\lambda}{r})$, namely

$$\Pr\{(X(j) - X(j-1)) = k\} = e^{-\frac{2\lambda}{r}} I_{|k|}(\tfrac{2\lambda}{r}),$$

where $I_{|k|}(\frac{2\lambda}{r})$ is the modified Bessel function of the first kind. Therefore mean and variance of $X(j) - X(j-1)$ is 0 and $\frac{2\lambda}{r}$ respectively.

We can design a linear estimator for $N(j) - N(j-1)$ respect to $a(j) - a(j-1)$.

$$\tilde{N}(j) - \tilde{N}(j-1) = c1(a(j) - a(j-1)) + c2 \qquad (13)$$

By taking mean and variance of both sides of (12), we have :

$$E(a(j) - a(j-1)) = E(N(j) - N(j-1)) = m \qquad (14)$$

$$Var(a(j) - a(j-1)) = \frac{2\lambda}{r} + Var(N(j) - N(j-1)) \qquad (15)$$

$$m = \frac{1}{k-1}\sum_{j=2}^{j=k}(a(j) - a(j-1))$$

$$\sigma^2 = \frac{1}{k-1}\sum_{j=2}^{j=k}(a(j) - a(j-1))^2 - m^2$$

By taking mean and variance of (13) and substituting (14) and (15) in (13)

$$m = c1.m + c2 \quad \text{and} \quad \sigma^2 - \frac{2\lambda}{r} = c1^2.\sigma^2$$

$$c1 = (1 - \tfrac{2\lambda}{r\sigma^2})^{0.5} \quad \text{and} \quad c2 = (1 - (1 - \tfrac{2\lambda}{r\sigma^2})^{0.5})m$$

$$\widetilde{N'_\tau}(j) = \frac{\tilde{N}(j) - \tilde{N}(j-1)}{\frac{1}{r}} = r\left(1 - \tfrac{2\lambda}{r\sigma^2}\right)^{0.5}(a(j) - a(j-1)) + r(1 - (1 - \tfrac{2\lambda}{r\sigma^2})^{0.5})m \qquad (16)$$

$\widetilde{N'_\tau}(j)$ is the estimator for N'_τ (j). Fig. 3 shows $\widetilde{N'_\tau}(j)$ and N'_τ(j), when $K(\varphi) = \frac{3}{4R^2}N_a(.)|\sin 2\varphi|$ and parameters used are listed in Table 1

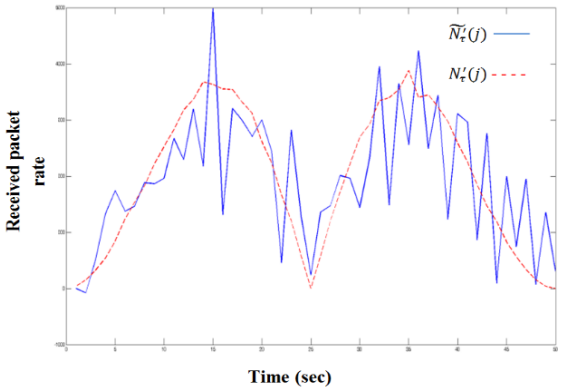

Fig.3. Comparison of N'_τ and $\widetilde{N'_\tau}$.

IV. DDoS ATTACK MITIGATION IN CLOUD

To remove attack packets from entering server in cloud, packet filtering methods for DDoS Attack Defense were pro-posed [8],[1]. We should use a security module like firewall or IDPS or UTM in front of servers to detect attack packets and prevent them from entering server. The cloning of these security modules in cloud can be done very fast.

When traffic increases or decreases the number of these security modules can be changed properly. Figuer. 4 shows the place of security modules in front of server to prevent attack packets from entering server.

Addition of each new security module increases the cost. We have a penalty function, which is linearly related to the number of security modules

A. DRIFT PLUS PENALTY METHOD

The Drift plus Penalty method applies to queuing systems that operate in discrete time slots $t \in \{0, 1, \ldots\}$ [9], [10], [11], [12]. This algorithm stabilizes the queue length and minimizes average cost [11]. The symbols used are listed in Table 2.

A nonnegative function $L(t) = \frac{1}{2}Q^2(t)$ is defined, where $Q(t)$ is the length of queue at the beginning of time slot t, this function is called Lyapunov function.

The Lyapunov drift is defined as:

$$\Delta(t) = \tfrac{1}{2}Q^2(t+1) - \tfrac{1}{2}Q^2(t) \qquad (17)$$

$$Q(t+1) = max\{0, Q(t) + a(t) - b(t)\} \qquad (18)$$

Where $a(t)$ and $b(t)$ are arrivals and departures from queue in time slot t. If we substitute (18) in (17) then:

$$\Delta(t) \le \frac{1}{2}\big(a(t)-b(t)\big)^2 + Q(t)\big(a(t)-b(t)\big) \quad (19)$$

If the queue is stable then $\frac{1}{2}\big(a(t)-b(t)\big)^2 \le B$

where B is an upper bound for $\frac{1}{2}\big(a(t)-b(t)\big)^2$ so:

$$\Delta(t) \le B + Q(t)\big(a(t)-b(t)\big) \quad (20)$$

At each time slot, the value of $Q(t)$ and arrivals are monitored and a control action (like increase or decrease of security modules) is decided to control the queue size. If $p(t)$ is the cost of resources in time slot *t*, then Drift plus Penalty algorithm, at each time slot, minimizes the function

$$DP(t) = \Delta(t) + Vp(t)$$

Where *V* is a weighting constant. The *V* parameter can be chosen to ensure the time average of $p(t)$ is arbitrarily close to optimum. This algorithm does not require any knowledge of input probability distribution function.

Table 2. Symbols in Drift plus penalty algorithm

Symbol	Description	Value(unit)
Q(t)	Length of queue at the beginning of time slot t	Number of packets
t	Time slot number	
Δ (t)	Lyapunov Drift	
p(t)	Penalty function	

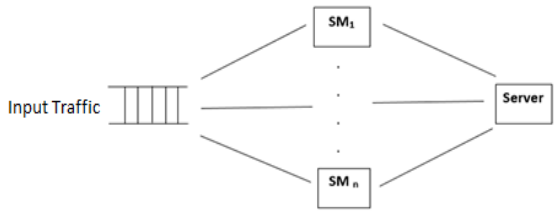

Fig. 4. Server protections in cloud

B. DRIFT PLUS PENALTY ALGORITHM

The cost of resources used in each time slot is proportional to the number of security modules used (Fig. 4), so:

$$p(t) = VC\ i(t) \quad (21)$$

Where *C* is the cost of each security module in one time slot and *V* is a weight constant. $i(t)$ is the number of security modules in time slot *t*.

The length of queue at the beginning of time slot *t* is $Q(t)$, the service rate of each security module is μ and the number of departed packets is $b(t)$ and the number of security modules in time slot t is $i(t)$,since we want to minimize the cost , each security module should be used at its highest capacity so $b(t) = \mu i(t)$, $i(t) \ge 1$. and define Drift plus Penalty function as:

$$DP(t) = VCi(t) + Q(t)[a(t) - \mu i(t)] \quad (22)$$

We want to choose $i(t)$ such that average of

$p(t)$, $t \in \{0, 1, \dots\}$ is minimized. We assume:

$$Q(t) + a(t) = k1(t)\,\mu + r(t)$$

Where $0 \le r(t) < \mu$

and $k(t) = \max\{1, k1(t)\}$.

If in each time slot we choose $i(t) > k(t)$, then because $Q(t+1) = max\{0, Q(t) + a(t) - \mu i(t)\}$,

$Q(t+1) = 0$, and $VCi(t) > VCk(t)$ so average of $p(t)$ is not minimized.

If otherwise we choose $i(t) = k(t) - l \ < k(t), l \ge 1$, then $Q(t+1) = Q(t) + a(t) - \mu(k(t) - l) = r(t) + \mu l$ and μl of the packets received in time slot *t* not serviced, and passed to time slot *t+1*, which adds *l* new security modules in time slot *t+1*, so there is no benefits in these choices for $i(t)$ and only delays are increased, so the optimum choice is $i(t) = k(t)$.

Note: in real situations because we don't know the value of $a(t)$ until the end of time slot t, we can use $a(t-1)$ as an estimator for $a(t)$.

C. DRIFT PLUS EXTENDED PENALTY ALGPRITHM

In this algorithm we try to define a new penalty function which considers delay as a new variable in its definition. We define $k(t)$ and $r(t)$ by the following:

$Q(t) + a(t) = k1(t)\mu + r$, where $0 \le r(t) < \mu$,

And $k(t) = \max\{1, k1(t)\}$, and $i(t) \ge 1$ be the number of security modules in time slot t, we define a function $Imax(t)$ as follows:

$$\text{Imax(t)} = \begin{cases} 0 & i(t) > k(t) \\ k(t) & i(t) \le k(t) \end{cases} \quad (23)$$

Definition of extended penalty function which considers delay is:

$$p(t) = VC\left(i(t) + \frac{Imax(t)^2}{i(t)}\right) \quad i(t) \ge 1 \quad (24)$$

If $i(t) > k(t)$ then $Q(t+1) = 0$ and delay is zero, otherwise delay is proportional to $\frac{Imax(t)^2}{i(t)}$ and penalty will be compensated by this extra term in penalty function.

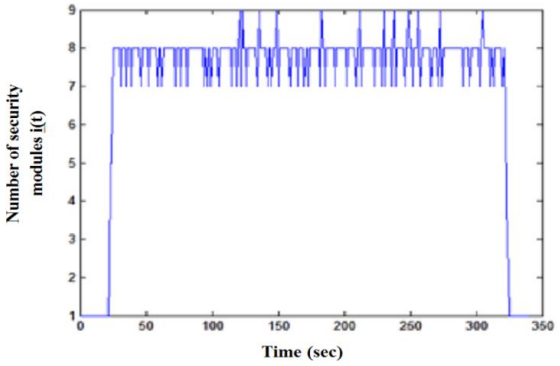

Fig.5. (a) Number of security modules versus time in the first method

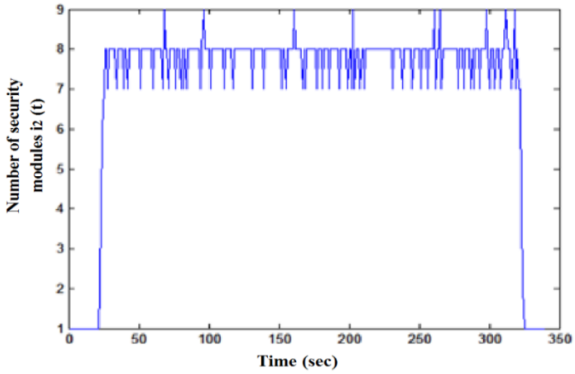

Fig.5. (b) Number of security modules versus time in the second method

V. PERFORMANCE EVALUATION

In this section we define a metric for average total cost of security modules used for the above Drift plus penalty methods. First we define the following metric:

$$\text{Sum} = \sum_{t=1}^{TT} i(t)$$

Where TT is total time of test and Sum is the total number of security modules per time slot used during test, and is normalized form of total cost (if the cost of one security module in one time slot be c, then total cost is c*Sum).

The drift plus penalty methods can be formulated as follows: $Q(t) + a(t) = k1(t)\mu + r(t)$

Where $0 \le r(t) < \mu$

$k(t) = \max\{1, k1(t)\}$.

$Q(t+1) = max\{0, Q(t) + a(t) - \mu i(t)\}$.

First method:

$$DP(t) = VCi(t) + Q(t)[a(t) - \mu i(t)],$$
$$i(t) = k(t) \qquad (25)$$

Second method:

$$\text{DP2(t)} = \min_{1 \le i2(t) \le k(t)} \left\{ VC\left(i2(t) + \frac{Imax(t)^2}{i2(t)}\right) + Q(t)[a(t) - \mu i2(t)] \right\}, \quad 1 \le i2(t) \le k(t) \qquad (26)$$

$$\text{max(t)} = \begin{cases} 0 & \text{i2(t)} > \text{k(t)} \\ \text{k(t)} & \text{i2(t)} \le \text{k(t)} \end{cases} \qquad (27)$$

Duration of test was *TT=340* second. In the first and last *TN= 20* seconds, there was no DDoS attack, in the middle TA= 300 second, there was a DDoS attack. Figure 2 shows the rate of packets generated in this period and the value of parameters are listed in Table 1. The service rate of each security module is $\mu =$ 14000 pps.

Result shows:

$$sum = \sum_{1}^{340} i(t) = 2396$$
$$sum2 = \sum_{1}^{340} i2(t) = 2389$$

Because both methods are drift penalty methods , therefore the average cost for both of them should be minimum and the results show the cost is approximately the same and it conforms with results of drift plus penalty method [11].

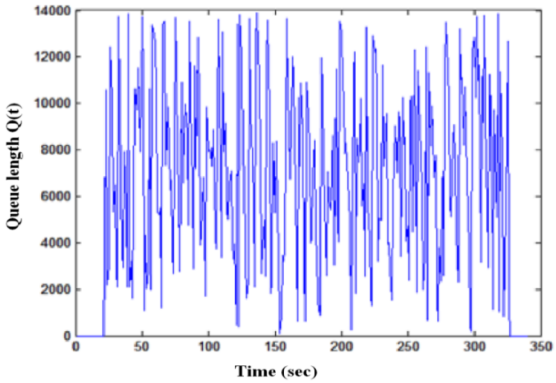

Fig. 6. (a) Queue length versus time in the first method

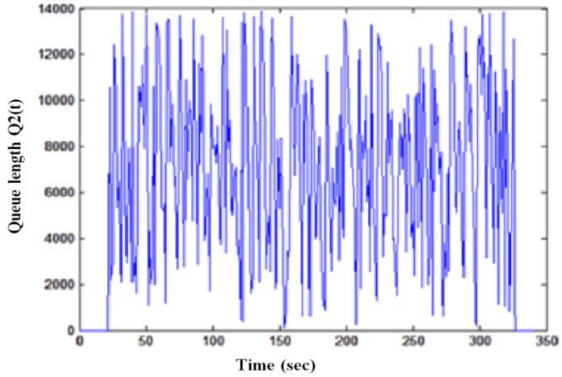

Fig. 6. (b) Queue length versus time in the second method

In both methods $0 \le Q(t), Q2(t) < \mu$ for t=1, 2… 340, which shows the stability of queues. Figures 6a, and 6b, show the $Q(t), Q2(t)$ respectively. The number of security modules $i(t), i2(t)$ in each time slot t is shown in Figures 5a, 5b, respectively.

VI. CONCLUSION

In this paper, we proposed an analytical model for DDoS traffic, the number of participating bots in a DDoS attack in one time slot is approximated by Poisson distribution. We showed geographical distribution of bots can be used as an identifier for a botnet and derived a linear estimator for this identifier.

We used Lyapunov optimization and Drift plus penalty method to offer algorithms for dynamic assignment of security modules in discrete time slots to protect host servers in cloud. Two different penalty functions proposed, in first penalty function only the number of security modules in that time slot is considered and in second one, number of security modules plus delay have been used. The results of simulation show cost is minimized and the size of queue is less than μ. Further work for more accurate estimator of N'_τ to locate concentration of attacking bots is suggested.

REFERENCES

[1] Shui Yu, and Song Guo, "Can we beat DDoS attack in Cloud?" IEEE transaction on parallel and distributed system, Vol. 25, No.9, September 2014.

[2] S. Yu, "Distributed Denial of Sservice Attack and Defense, spring briefs in computer science", DOI 10.1007/978-1-4614-9491-1_1, © the author(s) 2014

[3] Donald Gross, John F. Shortie, James M. Thompson, Carl M. Harris, "Fundamentals of Queuing theory" fourth edition, Wiley, 2008.

[4] Cem Gurkok, "Security Cloud computing systems" chapter 6 of "Computer and information security handbook".

[5] Alberto Leon Garcia, "Probability, statistics and random processes for electrical engineering". Chapter 12, third edition, 2009, Pearson prentice Hall.

[6] Peter Mell, Timothy Grance. NIST Special Publication 800-145,' The NIST Definition of Cloud Computing'.

[7] Jie Wang, Raphael C.-W. Phan, John N. Whitley and David J. Parish, "Advanced DDoS attack traffic simulation with a test center platform" international journal for information security research (IJISR), volume 1, issue 4, December 2011.

[8] J. Rameshbabu , B.Sam Balaji, R.Wesley Daniel, K.Mala, "A prevention of DDoS attacks in Cloud using NEIF techniques", international journal of scientific and research publications, volume 4, issue 4, April 2014, ISSN 2250-3153.

[9] M. J. Neely, "Distributed and secure computation of convex programs over a network of connected processors," DCDIS conf., Guelph, Ontario, July2005.

[10] M. J. Neely, "Stochastic Network Optimization with Application to Communication and Queueing Systems", Morgan & Claypool, 2010.

[11] https://en.wikipedia.org/wiki/Drift_plus_penalty.

[12] https://en.wikipedia.org/wiki/Lyapunov_optimization.

[13] https://en.wikipedia.org/wiki/Cloud-based_networking.

[14] https://en.wikipedia.org/wiki/Cloud_computing_issues.

[15] Jiejun Kong, Mansur Mirza, James Shue,"Random Flow Network Modeling and Simulation For DDoS Attack Mitigation", ICC,03.

[16] V. Paxon , S. Floyd, "Wide Area Traffic : The Failure of Poisson Modeling,", IEEE/ACM Transaction on Networking, 3(3): 226-244,1995.

[17] https://en.wikipedia.org/wiki/Botnet.

[18] https://www.stateoftheinternet.com/downloads/pdfs/2015-cloud-security-report-q3.pdf

[19] Antoine Scherrer; Nicolas Larrieu; Philippe Owezarski; Pierre Borgnat; Patrice Abry," **Non-Gaussian and Long Memory Statistical Characterizations for Internet Traffic with Anomalies**",IEEE Transactions on Dependable and Secure Computing ,Year: 2007, Volume: 4, Issue: 1 ,Pages: 56 - 70, DOI: 10.1109/TDSC.2007.12.

[20] V. Akilandeswari; S. Mercy Shalinie," **Probabilistic Neural Network based attack traffic classification**",2012 Fourth International Conference on Advanced Computing (ICoAC) Year: 2012,Pages: 1 - 8, DOI: 10.1109/ICoAC.2012.6416848

[21] E. Ilavarasan; K. Muthumanickam," **A Survey on host-based Botnet identification**",Radar, Communication and Computing (ICRCC), 2012 International Conference on ,Year: 2012 ,Pages: 166 - 170, DOI: 10.1109/ICRCC.2012.6450569.

[22] Shafiq Ul Rehman; Selvakumar Manickam; Singh," A survey of mitigation techniques against Economic Denial of Sustainability (EDoS) attack on cloud computing architectur",

Year: 2014,Pages:1 - 4, DOI: 10.1109/ICRITO.2014.7014767

A. Salahi was born in Tehran, Iran, on Feb.10.1947. He received his B.Sc. degree in electrical engineering from Tehran University, Iran, 1970, his M.Sc. degree from Kansas University, Lawrence, Kansas, USA, 1974, and Ph.D. degree from Purdue University, West Lafayette, Indiana, USA, 1979, all in electrical engineering. At present he is associate professor and senior project manager in Network Security Department in Iran Telecommunications Research Center. His research interests are network security, cloud security and security protocols. He was awarded Khwarazmi international twice in 1993 and 2000.

This Page intentionally left blank.