Browsing Behavior Mimicking Attacks on Popular Web Sites for Large Botnets

Shui Yu
School of IT
Deakin University
Burwood, VIC 3125
Australia
syu@deakin.edu.au

Guofeng Zhao
Inst. of Mobile Int. Tech.
Chongqing Uni. of Posts and Tel.
Chongqing, 400065
China
zhaogf@cqupt.edu.cn

Song Guo
School of CSE
The University of Aizu
Aizuwakamatsu
Japan
sguo@u-aizu.ac.jp

Yang Xiang
School of IT
Deakin University
Burwood, VIC 3125
Australia
yang@deakin.edu.au

Athanasios V. Vasilakos
National Tech. Uni. of Athens
Heroon Polytechniou 9
15780 Zografou
Greece
vasilako@ath.forthnet.gr

***Abstract*—With the significant growth of botnets, application layer DDoS attacks are much easier to launch using large botnet, and false negative is always a problem for intrusion detection systems in real practice. In this paper, we propose a novel application layer DDoS attack tool, which mimics human browsing behavior following three statistical distributions, the Zipf-like distribution for web page popularity, the Pareto distribution for page request time interval for an individual browser, and the inverse Gaussian distribution for length of browsing path. A Markov model is established for individual bot to generate attack request traffic. Our experiments indicated that the attack traffic that generated by the proposed tool is pretty similar to the real traffic. As a result, the current statistics based detection algorithms will result high false negative rate in general. In order to counter this kind of attacks, we discussed a few preliminary solutions at the end of this paper.**

***Index Terms*—attack simulation; browsing behavior; botnet.**

I. Introduction

Botnet based network attacks, such as distributed denial of service (DDoS), virus, information phishing and anonymity attacks, are pervasive in the Internet, and they cause great financial loss [1], [2], [3], [4]. In general, majority of the current detection algorithms are based on features of known attack methods, as a result, it is hard to identify new attacks. Moreover, sophisticated hackers are trying their best to simulate legitimate behavior to fly under the radar [5]. In other words, we hardly know the truth of false negative of our detection algorithms in practice. Therefore, it is necessary for defenders to design attack methods to examine and improve our defense tools. This is also a motivation of this paper.

Web browsing is very popular nowadays, and the web becomes a major media for information dissemination for governments, companies and individuals. DDoS attacks on web sites reward attackers financially or politically. We have witnessed increasing number of this kind of attacks [6]. A few detection and mitigation strategies are in place for application layer DDoS attacks [7], however, it is hard to know the true false negative rate of the existing detection algorithms.

In this paper, we propose a novel application layer DDoS attack method, which simulates legitimate browsing behavior to fool detectors. The conditions to perform this kind of attacks is available: there are usually millions of bots for a large botnet. If every bot acts as a legitimate browser, then it is sufficient to deny the service of a target web site.

The paper makes the following contributions.

- We proposed a browsing behavior mimicking attack method. This kind of attack may exist already, however, the available detection algorithms cannot identify it, therefore, attackers are flying under the radar. Our real data experiments demonstrated that the proposed algorithm does generate traffic which is very similar to the real traffic.
- We pointed out possible solutions to counter the proposed attack method. For example, invisible special hyperlinks can be embedded into web pages to help detection algorithms.
- The proposed algorithm can be used as a DDoS attack traffic generator as we have few real DDoS attack data sets available for research communities.

The rest of this paper is organized as follows. Related work is discussed in Section II, followed by the modeling and analysis of the browsing behavior mimicking attack in Section III. In Section IV, we present the implementation details of the attack algorithm. Performance evaluation is conducted in Section V. Counter attack strategies are discussed in Section VI. Finally, we summarize this paper and present future work in Section VII.

II. Related Work

Cyber attackers are organizing botnets to carry out their malicious tasks [3], [4]. The basic story about botnet is like follows. A botmaster writes a program, called bot or agent, and he installs the bots at the compromised computers on the Internet using various network virus-like techniques. All the bots form a army or botnet which are controlled by the botmaster to commit illegal tasks, such as launching DDoS attack, sending spam emails, performing phishing activities, collecting sensitive information. The hosts running these attack tools are known as bots or zombies [1], [8], [9]. There is a command and control (*C&C*) server(s) to communicate with the bots and collect data from bots. In order to disguise himself from legal forces, botmaster changes the url of his *C&C* frequently, such as weekly. Majority of the current

DDoS attacks are performed by botnets [10]. DDoS attackers target on exhausting the victim's resources, such as network bandwidth, computing power, operating system data structures, and so on. An excellent explanation about this could be found in [3]. Research has indicated that the size of a botnet could be millions [3], [11]. Under such circumstance, it is easy for attackers to pretend as legitimate users to conduct attacks, and it is hard to be identified.

The characteristics of legitimate browsers have been explored widely since the beginning of the Internet. Breslau, et al. analyzed the web accessing behavior and found that it follows the Zipf-like distribution, and the parameter α varies from traces to trace ranging from 0.64 to 0.83 [12]. A general form of the popularity distribution is called Zipf-Mandelbrot distribution [13]. This findings are used widely for research papers, such as [14] [15]. Moreover, [16] analyzed that the page viewing time (following the Pareto distribution) causes the Zipf distribution of request traffic. Huberman et al. indicated that the probability of browsing length of a web user follows the two-parameter inverse Gaussian distribution [17].

III. Browsing Behavior Mimicking Attack Modeling

In this section, we will establish the browsing behavior model. An attack tool can be built based on the model for every bot in a botnet. The model is established for one bot, and the aggregated attack will be discussed in the following section.

For a given web site, we assume that there are $N(N > 0)$ web pages in total, and they are sorted by popularity from the most to the least as $w_1, w_2, \ldots, w_N$. Let random variable W be the requested web page, and $Pr(W = i)$ be the access probability of page w_i. It is well-known that the page popularity follows the Zipf-Mandelbrot distribution, which is as follows.

$$Pr(W = i) = \frac{\Omega}{(i+q)^{\alpha}} \tag{1}$$

where $\alpha(0 < \alpha)$ is the *skewness* factor, which dominates the skewness of the distribution, and $q(q \geq 0)$ is the *plateau* factor, which make the probability of the highest ranked pages flatted. The Zipf-Mandelbrot distribution becomes the Zipf distribution when $\alpha = 1$, and it becomes the Zipf-like distribution when $q = 0$. Since $\sum_{i=1}^{N} Pr(W = i) = 1$, $\Omega = \left(\sum_{i=1}^{N} \frac{1}{(i+q)^{\alpha}}\right)^{-1}$.

We will use $Pr(W = i)$ to decide the first page to browse, therefore, the victim will see the requests follows the Zipf-Mandelbrot distribution, rather than requests focus on a few web pages. Therefore, this strategy can disable some request statistical based detection algorithms.

Once the browsing page is decided, the bot submits the page request to the victim, and download the page. When the requested page has been downloaded, the bot starts to "read" the page. The reading duration is usually decided by the Pareto distribution. As indicated in [18], [19], the Zipf distribution and the Pareto distribution are essentially the same in different formats. Moreover, [16] analyzed that the page viewing time (following the Pareto distribution) causes the Zipf-Mandelbrot distribution of request traffic. The Pareto distribution shares the same skewness factor, α, with the Zipf-Mandelbrot distribution in equation (1). Hence, the viewing time distribution is described as follows.

Let random variable V be the page viewing time for a given web page, and v_m be the minimum viewing time for web pages. For a given web page with viewing time v, the probability density function of the viewing time distribution is defined as follows.

$$Pr(V = v) = \alpha \cdot v_m^{\alpha} \cdot v^{-(\alpha+1)} \tag{2}$$

where $v_m \leq x$, and α is also called the *Pareto index*.

Following the properties of the Pareto distribution, when $\alpha > 1$, the mean of the viewing time distribution is as follows.

$$\overline{V} = \frac{v_m \alpha}{\alpha - 1} \tag{3}$$

After "viewing" the current page, the bot needs to request another page. In this case, the bot has to simulate the legitimate users on how many pages to request for one browsing session. The legitimate browsing length is determined by the inverse Gaussian distribution [17], which is defined as follows.

Let L be the number of links that a browser surfs in a web site, then the distribution is

$$Pr(L = l) = \sqrt{\frac{\lambda}{2\pi l^3}} exp\left[\frac{-\lambda(l-\mu)^2}{2\mu^2 l}\right], l = 1, 2, \ldots \tag{4}$$

where μ is the mean, and λ is the shape parameter. The inverse Gaussian distribution approximates to the Gaussian distribution when $\lambda \to \infty$.

Let random variable T be the time a browser stays with a web site for one browsing session. Based on equation (4) and equation (2), we can obtain probability of the browsing duration for a bot for a browsing session as follows.

$$\begin{aligned} & Pr(T = t) = Pr(L = l, V = v) \\ = \quad & \alpha \cdot v_m^{\alpha} \cdot v^{-(\alpha+1)} \cdot \sqrt{\frac{\lambda}{2\pi L^3}} exp\left[\frac{-\lambda(l-\mu)^2}{2\mu^2 l}\right] \end{aligned} \tag{5}$$

Based on the Wald's Theorem [20], we know that

$$\overline{T} = E[T] = E[l] \cdot E[V] = \mu \cdot \frac{v_m \alpha}{\alpha - 1} \tag{6}$$

With the page popularity (equation (1)) and browsing duration (equation (5)), for one given bot, we obtain its frequency of generate a request of page i under the condition of viewing time v and browsing length l as follows.

$$\begin{aligned} & Fr(W = i, V = v, L = l) \\ = \quad & \frac{1}{Pr(W = i, V = v, L = l)} \\ = \quad & \frac{i^{\alpha}}{\Omega} \cdot \frac{v^{-(\alpha+1)}}{\alpha \cdot v_m^{\alpha}} \cdot \sqrt{\frac{2\pi l^3}{\lambda}} exp\left[\frac{\lambda(l-\mu)^2}{2\mu^2 l}\right] \end{aligned} \tag{7}$$

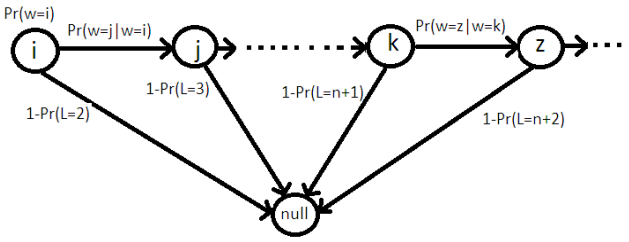

Fig. 1. The Markov chain model for browsing behavior mimicking model

We can use the Markov chain model to represent our browsing behavior mimicking. As shown in Figure 1, a bot takes a starting page (suppose to be page $i(i = 1, 2, \ldots, N)$) using the page popularity probability $Pr(W = i)$. For the next page, there are two options: either follow one of the hyperlinks in page i (denoted as $\{h_i\}$); or leave the current web site. The probability of leaving the current web site is

$$1 - Pr(L = n + 1), n = 1, 2, \ldots \tag{8}$$

where n is the number of page requests that have been issued by a bot. We use $null$ to represent a special page, which does not belong to this web site. In the Markov model, if the next state is $null$, it means the browser leaves the web site.

If the bot follows the set of hyperlinks, $\{h_i\}$, of the current page, then the bot needs to select one hyperlink from the set. Suppose the next page is page j, we denote the transition probability as follows.

$$Pr(W = j|W = i) = \frac{Pr(W = j)}{\sum_{k\in\{h_i\}} Pr(W = k)} \tag{9}$$

Because the bot either chooses one hyperlink from $\{h_i\}$ or leave the web site. Therefore,

$$\begin{aligned} 1 &= Pr(L = n + 1) \cdot \sum_{k\in\{h_i\}} Pr(W = k|W = i) \\ &+ Pr(W = null|W = i) \end{aligned} \tag{10}$$

We have analyzed the individual browsing behavior, however, it is not sufficient to simulate the legitimate behavior of browsers in terms of groups. We have noticed that the number of browsers is dynamic in different time points in a day. Figure 2 is a sample of one day period from our data set, and we noticed that this dynamics are similar for the whole data set. So far, we do not have a formula to describe it. Let random variable U be the number of browsers for a given time point t', and $Pr(U = u|t')$ is the probability that there are u browsers at time point t'.

We can calculate how many requests, R, are received for page $i(i = 1, 2, \ldots, N)$ at time t'.

$$\begin{aligned} R(W = i, t = t') &= Pr(W = i) \cdot \Pr(U = u|t') \\ &= \frac{\Omega}{(i + q)^\alpha} \cdot Pr(U = u|t') \end{aligned} \tag{11}$$

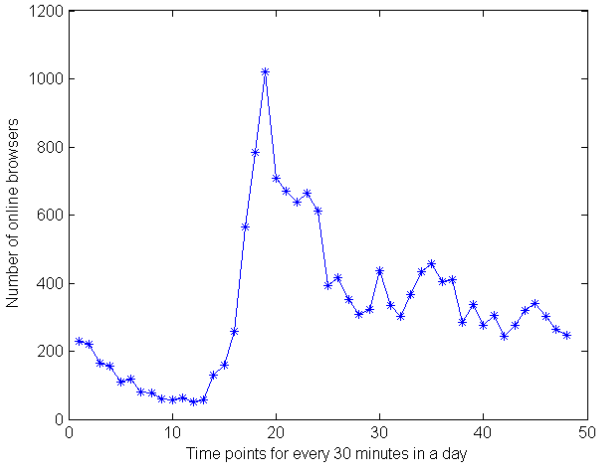

Fig. 2. The distribution of the number of online browsers for a day

Moreover, in order to examine the effectiveness of the proposed attack method, we can investigate how similar the generated traffic against a real data counterpart. There are a number of method to carry out this task, such as correlation coefficient and information distance.

Let X_i and X_j $(i \neq j)$ be two sequence with same length N. We define the correlation coefficient of the two sequences as

$$\rho_{X_i,X_j}[k] = \frac{r_{X_i,X_j}[k]}{\frac{1}{N}\left[\sum_{n=1}^{N-1} x_i^2[n] \sum_{n=1}^{N-1} x_j^2[n]\right]^{1/2}}. \tag{12}$$

where $k = 0, 1, 2, \ldots, N - 1$ is the phrase difference.

We can also use the Kullback-Leibler distance to measure the similarity of X_i and X_j as follows.

$$D(X_i, X_j) = \sum_{x\in\chi} Pr(X_i = x) \cdot \log \frac{Pr(X_i = x)}{Pr(X_j = x)} \tag{13}$$

where χ is the probability space of X_i and X_j.

In this paper, we will plot diagrams to indicate the similarity in a direct way.

IV. Mimicking Attack Algorithm

In this section, we present the implementation of the legitimate browsing behavior mimicking algorithm. The details can be found in Algorithm 1.

In case that the size of the botnet is not sufficient, a bot can sleep for a random time interval and start a new 'browsing' session in order to increase the attack strength.

V. Performance Evaluation of the Proposed Attack Method

In order to evaluate the effectiveness of the proposed attack method, we have collected web browsing dynamics data for 10 days (June 1 - 10, 2010) at a Chinese backbone network center. There are thousands of web sites in the center, we took

Algorithm 1: Browsing behavior mimicking algorithm

Input: α, l, v_m

Output: A sequence of page requests

//Initialize the parameters;

1. Extracting the IP address of the local computer;
2. Generate a random number x using the ip address;
3. Generate a browsing length l using x according to equation (4) ;
4. Determine the first request page i to request using x according to equation (1);

while *True* **do**

  //Requesting pages;

  for $i = 1$ **to** $l - 1$ **do**

    a. Extract hyperlinks $\{h_i\}$ from current page i ;

    b. Determine one hyperlink j from set $\{h_i\}$ according to equation (1);

    c. Waiting for a time interval according to equation (2);

    d. Submit request for page j ;

  end

  //sleeping for a random;

  Sleeping a random time interval ;

  Spoofing a different source IP address ;

end

one of the most popular news web site, www.sohu.com.cn, for our experiments. We arrange the data into a matrix: each web page of the web site is a row in the matrix; and every column denotes the number of requests in 30 seconds. We sampled the requests for each page every 30 seconds for the 240 hours duration.

In order to identify the specific parameters for the attack model, we use the first day of the data set to extract the parameters.

We first identify the α for the web page popularity distribution. We take the first day data from the data set, and we used the LSS method to find the most suitable α is 1.31. The result is shown in Figure 3.

Following the previous reports [16], [7], [21], [19], we take the minimum viewing time $v_m = 30$ seconds. Following equation (6), the mean of viewing one page is $\frac{30 \times 1.31}{1.31-1} = 126.77$ (seconds). Humerman et al. indicated that $\mu = 15$ in their experiments [17] for browsing length. Therefore, following equation (6), the mean of life span for a browsing session is $\overline{T} = 15 \times 126.77 = 1901.55$ (seconds), which is around 31.69 minutes.

With all these parameters in hands, we try to examine the effectiveness of the request distribution of the proposed attack tool. We first extract the parameters of one sample (6 pm - 6:30pm of day 1) and using the parameters to generate a requests traffic. We compare the generated traffic with the real data (6 pm - 6:30 pm of day 10). The result is shown in Figure 4.

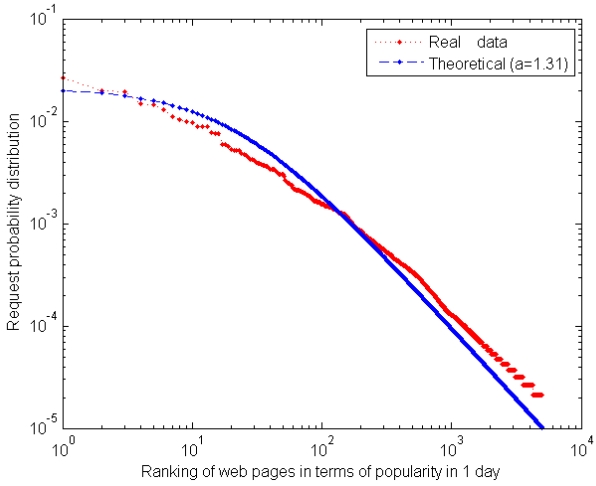

Fig. 3. Finding the skewness factor α of the target web site, there were 5,001 different pages were requested during that day

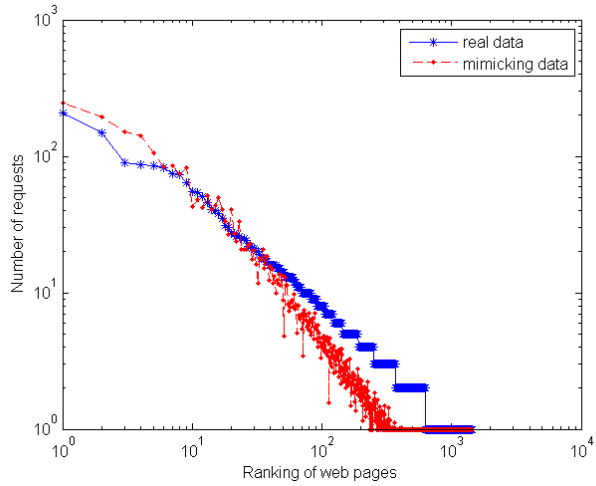

Fig. 4. The comparison between the mimicking requests ($q = 4$, users = 326, $\mu = 15$ with Gaussian noise) and the requests of the real data (requests during 6pm - 6:30 pm of day 10)

From Figure 4, we found that the generated data is similar to the real data in general. As a result, the page popularity based detection algorithms are invalid to deal with the proposed attack tool.

DDoS traffic is an aggregated traffic from many bots, we further investigate the aggregated traffic of the proposed attack tool. We use the parameters that we extracted from day 1 data to simulate the whole day request dynamics of day 10, and the result is shown in Figure 5. The result indicated that the generated aggregated requests are similar to the real request dynamics of day 10. Therefore, the user dynamics based detection algorithms are invalid against the proposed attack tool.

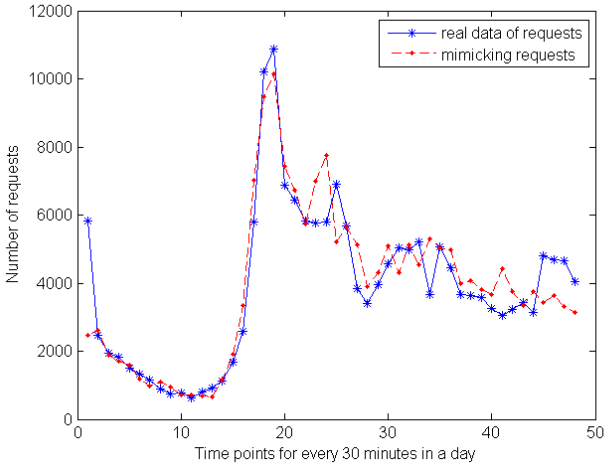

Fig. 5. The comparison on number of requests for every 30 minutes between the real data of day 10 and that generated by a botnet

VI. Discussion on Counter Attacks

As we have demonstrated, it is very difficult to discriminate legitimate request traffic from botnet generated request traffic based on statistical metrics.

In order to counter the proposed browsing behavior mimicking attack, we have to introduce new information into the system to detect this kind of attacks. One counter attack strategy is to insert a number of invisible hyperlinks into every web page, and all these hyperlinks point to a special page. As a result, for the legitimate cases, there is no or very few requests for the special page because human browsers usually do not click the invisible hyperlink. However, in attack casess, bots will submit requests for the special page as bots crawl downloaded pages and select the special hyperlink with a high probability. Once server received a given number of the special requests for a given time interval, then we know the server is under DDoS attack.

VII. Summary and Future Work

In this paper, we proposed a statistics based legitimate browsing behavior mimicking algorithm for botnets. In the proposal, every bot employs the same algorithm to generate web page requests for a given victim. As the bots use statistical methods with different seeds, e.g. local IP addresses, to generate attack traffic, therefore, the traffic is unique for every individual bot, however, the aggregated traffic follows the statistical patterns of legitimate traffic. As a result, the existing statistics based detection algorithms are invalid to the proposed attack in general. Our real data experiments confirmed that the proposed attack tool does generate traffic that is pretty similar to legitimate traffic. The proposed method can be used to generate DDoS attack traffic for network security research communities to test their defense algorithms and tools.

Our preliminary investigation indicated that statistical detection methods are invalid for the proposed attack method, however, there are may and should exist effective methods from other angle, such as information theory, which can address this detection issue although we do not know what are they and how can they achieve the goal. This remains as a future work.

References

[1] T. Peng, C. Leckie, and K. Ramamohanarao, "Survey of network-based defense mechanisms countering the dos and ddos problems," *ACM Computing Survey*, vol. 39, no. 1, 2007.

[2] M. Edman and B. Yener, "On anonymity in an electronic society: A survey of anonymous communication systems," *ACM Computing Survey*, vol. 42, no. 1, 2009.

[3] B. Stone-Gross, M. Cova, L. Cavallaro, B. Gilbert, M. Szydlowski, R. Kemmerer, C. Kruegel, and G. Vigna, "Your botnet is my botnet: Analysis of a botnet takeover," in *CCS '09: Proceedings of the 2009 ACM conference on computer communication security*, 2009.

[4] M. Bailey, E. Cooke, F. Jahanian, Y. Xu, and M. Karir, "A survey of botnet technology and defenses," in *Proceedings of the 2009 cybersecurity applications and technology conference for Homeland security*, 2009.

[5] S. Yu, W. Zhou, and R. Doss, "Information theory based detection against network behavior mimicking ddos attack," *IEEE Communications Letters*, vol. 12, no. 4, pp. 319–321, 2008.

[6] A. El-Atawy, E. Al-Shaer, T. Tran, and R. Boutaba, "Adaptive early packet filtering for protecting firewalls against dos attacks," in *Proceedings of the INFOCOM*, 2009.

[7] Y. Xie and S.-Z. Yu, "A large-scale hidden semi-markov model for anomaly detection on user browsing behaviors," *IEEE/ACM Transactions on Networking*, vol. 17, no. 1, pp. 54–65, 2009.

[8] B. Stone-Gross, M. Cova, L. Cavallaro, B. Gilbert, M. Szydlowski, R. Kemmerer, C. Kruegel, and G. Vigna, "Your botnet is my botnet: Analysis of a botnet takeover," in *Proceedings of the 2009 ACM conference on computer communication security*, 2009.

[9] C. Y. Cho, J. Caballero, C. Grier, V. Paxson, and D. Song, "Insights from the inside: A view of botnet management from infiltration," in *Proceedings of USENIX LEET*, 2010.

[10] V. L. L. Thing, M. Sloman, and N. Dulay, "A survey of bots used for distributed denial of service attacks," in *SEC*, 2007, pp. 229–240.

[11] M. A. Rajab, J. Zarfoss, F. Monrose, and A. Terzis, "My botnet is bigger than yours (maybe, better than yours): why size estimates remain challenging," in *HotBots'07: Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets*, 2007.

[12] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and zipf-like distributions: Evidence and implications," in *Proceedings of the INFOCOM*, 1999, pp. 126–134.

[13] Z. K. Silagadze, "Citations and the zipf-mandelbrot's law," *Complex Systems*, vol. 11, p. 487, 1997.

[14] A. Klemm, C. Lindemann, M. K. Vernon, and O. P. Waldhorst, "Characterizing the query behavior in peer-to-peer file sharing systems," in *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*. New York, NY, USA: ACM, 2004, pp. 55–67.

[15] M. Hefeeda and O. Saleh, "Traffic modeling and proportional partial caching for peer-to-peer systems," *IEEE/ACM Transactions on Networking*, vol. 16, no. 6, pp. 1447–1460, 2008.

[16] M. E. Crovella and A. Bestavros, "Self-similarity in world wide web traffic: evidence and possible causes," *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 835–846, 1997.

[17] B. A. Huberman, P. L. T. Pirolli, J. E. Pitkow, and R. M. Lukose, "Strong regularities in world wide web surfing," *Science*, vol. 280, no. 3, 1998.

[18] M. Mitzenmacher, "A brief history of generative models for power law and lognornal distributions," *Internet Mathematics*, vol. 1, 2004.

[19] W. J. Reed and M. Jorgensen, "The double pareto-lognormal distribution - a new parametric model for size distributions," *Communication in Statistics - Theory and Methods*, vol. 33, no. 8, pp. 1733–1753, 2003.

[20] P. V. Mieghem, *Performance Analysis of Communications Networks and Systems*. New York, NY, USA: Cambridge University Press, 2005.

[21] S. Burklen, P. J. Marron, S. Fritsch, and K. Rothermel, "User centric walk: An integrated approach for modeling the browsing behavior of users on the web," in *ANSS '05: Proceedings of the 38th annual Symposium on Simulation*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 149–159.