Generating Web Traffic based on User Behavioral Model

Guo-feng Zhao
Institute of Future Internet Technology
Chongqing University of Posts and Telecommunications
Chongqing, P. R. China
zhaogf@cqupt.edu.cn

Min-chang Yu,
Institute of Future Internet Technology
Chongqing University of Posts and Telecommunications
Chongqing, P. R. China
615176114@qq.com

Chuan Xu,
Institute of Future Internet Technology
Chongqing University of Posts and Telecommunications
Chongqing, P. R. China
xuchuan@cqupt.edu.cn

Hong Tang
Institute of Future Internet Technology
Chongqing University of Posts and Telecommunications
Chongqing, P. R. China
tanghong@cqupt.edu.cn

***Abstract*— Generating Web traffic is of great importance to analyse performance of new designed network, test new equipment, and verify new protocols, etc. Most existing traffic generation systems tend to simulate the overall characteristics of network traffic, while neglecting of the behavior of the individual users. However, in principle, the emerged characteristics of overall traffic originate from the aggregation of individual users' access behavior. In this paper, we propose an innovative web traffic generating method based on user browsing behavior. Our method simulates the real users' accessing behavior, and visits the real web servers. Then, we design and develop a web traffic generating system. Because our system accesses the real web, it can produce almost the real network traffic. The test results show that the traffic generated by our system has characteristics of burstiness and self-similarity, which are widely exposed characteristics in real networks; meanwhile, our system better reflects real user's web browsing behavior.**

Keywords-Traffic generation; Pareto Distribution; Markov Model;

I. INTRODUCTION

The Web traffic generating system are widely used in many aspects, such as network performance test, new network protocol test, and site security assessment, etc. The traffic generated by such systems will directly determine the accuracy of experimental test results. So, how to generate the similar traffic as the real network is of great importance.

Currently, the methods of generating Web traffic can broadly be classified into two kinds: 1) traffic playback, and 2) traffic model simulation. 1) Traffic playback uses the network tools, e.g., sniffer, to capture packets and record them in a log file, then new simulating traffic can be generated based on the log file. This method can generate real network traffic captured by network tools. However, the result is time and scope limited, and cannot reflect the changing characteristics of network traffic. 2) Based on mathematical traffic models, many tools can generate network traffic. Leland [1] analyzed the real network traffic and pointed out that it had self-similarity and burstiness, and it proved to be true in many different networks. Using this method, we can produce changing network traffic similar to real network, but the traffic cannot reflect individual users' browsing behavior, such as the law of users' jump relation among different Uniform Resource Locators (URLs), users' preferences on different pages. However, many research tasks hunt for network traffic with users' behavior revealed, to test the particular network technology, such as service migration in Service-Oriented Future Internet [2].

The main work of this paper is as follows. 1) We propose a web traffic generating method based on web users' access behavior model. According to the method, first, we choose the first webpage of a real Web for a web user to visit; next, we calculate a page viewing time for the user; and then, we forecast the next page to access. 2) We present the design of the web traffic generating system, which consists of management module, preprocessing module and traffic generating module. 3) We develop a prototype using Python [3] and test the system. Results show that traffic generated by our system has similar characteristics as the real web traffic, such as self-similarity and burstiness characteristics; however, it takes users' behavior into consideration.

The remaining of the paper is organized as follows. After discussing related work in Section II, we describe our traffic generating method based on user behavior in Section III. We present the details of system design in Section IV and show test results in Section V. In Section VI, we conclude and outline future work.

II. RELATED WORK

The experimental verification of network testbed is critical for Future Internet research. Traffic generator being a key part of the network testbed is widely used in the evaluation of website and network performance. With the development of Future Internet research, traffic generator based on user behavior characteristic validates and evaluates the performance of the key technologies more effectively.

A number of successful application-specific traffic generators have been developed. Tcpreplay is a typical flow playback tool, which can replay directly packets captured by a 3rd part network data catching software such as Tcpdump [4]. Tcpreplay also supports replay packets

 ISBN: 978-1-61208-285-1

with appropriate modifications in the headers of link layer, network layer and transport layer, but such tools only mechanically replay the captured data packets at a regular rate. SPECweb, a tool of evaluating the performance of web servers, generate network traffic by sending HTTP Get requests to web server [5]. User can send requests separated by a constant interval. As a result of neglecting the real web user's behavior, traffic flow generated by SPECweb is deviant from the real network.

Alessio Botta et al. present a tool for the generation of realistic network workload that can be used for the study of emerging networking scenarios. However, it also did not take the user behavior into consideration [6].

Above analyses show that current traffic generators are developed based on part of traffic characteristics. However, they did not consider individual user browsing behavior, such as the law of users' jump relation between different URLs, users' preferences on different pages.

III. TRAFFIC GENERATING METHOD BASED ON USER BEHAVIORAL MODEL

Our web traffic generator can simulate normal access behavior of web users, and generate accurate and real network traffic. The procedure on how it works can be divided into the following steps:

Step 1: choose the first page of a real Web to visit. For example, when web users first browse a comprehensive portal website, they will choose the first page to visit, whatever it belongs to news section, sports section, or entertainment section, etc.

Step 2: spend a period of time on reading the content of the webpage. It is an interval between the accesses of two consecutive pages (viewing time).

Step 3: choose a new page for the next visit. Briefly, we extract and log the hyperlink URLs embedded in pages, and choose the next page URL based on Markov model.

In a web user's session, the Step 2 and Step 3 are running repeatedly until it logs out. In short, we need determine the first page, viewing time and the next page.

A. How to choose the first page

For a given real website, the web pages can be sorted to different ranks based on their popularity. All the ranks can be defined as follows: w_1, $w_2 \ldots$ w_n. The bigger n is, the more popular the web page is. We use random variable W to represent a web page, and w_i represent the probability of the web page accessed. It is well-known that the page popularity has the law of the Zipf-Mandelbrot distribution as in Equation (1) [7].

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

We denote α ($\alpha > 0$) as the skewness coefficient, which determines the skewness of the Zipf-Mandelbrot distribution, and q ($q \geq 0$) as the plateau coefficient. The plateau coefficient means the most popular web page is more likely. When $\alpha = 1$, the Zipf-Mandelbrot distribution becomes the Zipf distribution. When q=0, the distribution becomes Zipf-like distribution.

$$\sum_{i=1}^{N} P(W = i) = 1 \tag{2}$$

$$\Omega = \sum_{i=1}^{N} \left(\frac{1}{(i+q)^{\alpha}} \right)^{-1} \tag{3}$$

According to [7], Equation (3) can be inferred from Equations (1) and (2). We denote P(W=i) as the probability of accessing page i, and P_{max}=Max{P(W=1), P(W=2), …, P(W=n)} as the most popular web page, which means this page is more likely to become first page selected by web users. In other words, the popularity of web pages shows high degree of asymmetry. Most of the requests access a few hot pages.

B. How to calculate the viewing time

Viewing time refers to the interval bewtween two consecutive Web page requests and shows how long a user spends on a given Web page. We use the traditional ON/OFF model to describe the users' viewing behavior, as shown in Figure 1.

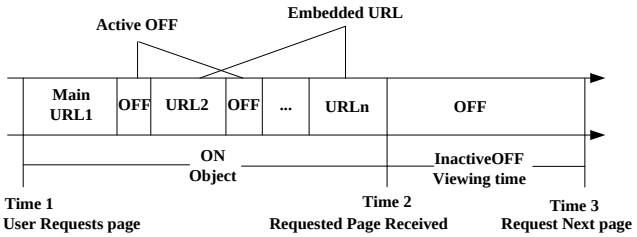

Figure 1. ON/OFF model of user browsing behavior

Figure 1 illustrates the behavior of web users. The horizontal axis represents time. On the first time slot, the client sends HTTP Get to URL1. The response message contains n embedded URL. Then, the client sends HTTP Get every OFF time, until the web browser receives all the data on the second time pot. Subsequently, the inactive OFF time means viewing time. On the third time pot, the client sends HTTP Get to request for next page.

Since the active OFF time is very short (less than 1 second) in actual development, we can ignore it relative to the web user's viewing time. According to all kinds of web browsers, we send HTTP Get requests for embedded URLs as soon as possible. Therefore, the active OFF time is influenced by the performance of client machine and network latency, and the inactive OFF time follows Pareto Distribution [8].

We denote W as the page viewing time and k=Min{wi}($1 \leqslant i \leqslant n$) as the minimum viewing time and

then the probability density function of the viewing time distribution is denoted as in Equation (4) [7].

$$P(W = w_i) = \alpha k^{\alpha} w_i^{-(\alpha+1)} \quad (4)$$

From the Equation (4), we can get cumulative distribution function, as in Equation (5).

$$F(w_i) = 1 - (k / w_i)^{\alpha} \quad (5)$$

We can get the random variable α following Pareto Distribution with the inverse function, as in Equation (6).

$$w_i = k / U^{1/\alpha} \quad (6)$$

The random parameter U follows Uniform Distribution within the range of (0, 1]. We figure out the viewing time of different Web requests, and it follows Pareto Distribution. Therefore, we can use Kolmogorov-Smirnov (KS) to compute parameter α [9].

Let X_1, X_2, …, X_n be independent identically distributed observation samples, Kolmogorov-Smirnov(KS) test of this distribution is based on D_n which is the absolute value of the maximum vertical distance between the assumed distribution function F_n (x) and the empirical distribution function F(x).

$$D_n = \sup_x | F_n(x) - F(x) | \quad (7)$$

We assume that the null hypothesis H_0 represents the hypothesis distribution function $F_n(x)$ follows the empirical distribution F(x). If the inequality (8) is satisfied, the nul hypothesis H_0 is invalid; otherwise, the distribution follows empirical distribution F(x).

$$(\sqrt{n} + 0.12 + 0.11/\sqrt{n}) D_n > c(\alpha) \quad (8)$$

where the critical value of c(α) depends on the significant level a. Since we assume a=0.05, the value of c (α) is 1.358 [10]. In (8), the smaller D_n is, the more the distribution anatomized with empirical distribution. We use a smaller D_n to figure out the parameters of hypothesis distribution, so that we can get the parameter of Pareto Distribution.

C. How to forecast next page

We forecast the browsing pattern of web users using Markov model. Markov model can be represented as a triplet MK = {W, A, π } as in Equations (9) and (10), where we denote W as a discrete random variable, its range is [w_1, w_2,…, w_n], where w_i represents one web page as model's state. Matrix A is denoted as the transition probability. Let p_{ij}=P {W_t=w_j|W_{t-1}=w_i} indicate the probability of requesting for page W_j at time t, when the web user request web page W_i at t-1 time. π represents the initial state.

$$A = (p_{ij}) = \begin{bmatrix} p_{11}\ p_{12} \cdots p_{1n} \\ p_{21}\ p_{22} \cdots p_{2n} \\ \vdots \ddots \\ p_{n1}\ p_{n2} \cdots p_{nn} \end{bmatrix} \quad (9)$$

$$\pi = (p_i) = (p_1, p_2, p_3, \cdots p_n) \quad (10)$$

The transition matrix A and the initial state matrix π can be predefined by user or computed by web log. The method can be explained as follows. First, aggregate the web log based on IP address. Second, random extract N users' web log to constitute a learning data set U= {u_1, u_2… u_n}. Taking advantage of the learning data, we can estimate all the parameters of Markov model with maximum likelihood estimation.

$$p_{ij} = \frac{S_{ij}}{\sum_{j=1}^{n} S_{ij}}, \quad p_i = \frac{\sum_{j=1}^{n} S_{ij}}{\sum_{i=1}^{n}\sum_{j=1}^{n} S_{ij}} \quad (11)$$

According to current page and the transition matrix of Markov model, we can predict the next web page user will browse. Let vector V(T)= (0, 0, 1, …, 0, 0) represent the page k at time t, and the next page location is $\max\left(\mathrm{V(t)} \cdot \mathrm{A}\right)$ at time t+1.

IV. SYSTEM DESIGN

A. Design Considerations

When designing the traffic generating system, we take the following characteristics into consideration.

Accuracy. It means the system should produce traffic which fits well in two aspects: (1) The authenticity of the network traffic, such as burstiness and self-similar. (2) the authenticity of web user browsing behavior, such as page popularity and jumping.

Concurrency. The system should simulate many web users' browsing behavior simultaneously. However, the scale of the traffic generated can be controlled and adjusted by client for different test goals.

Platform-independence. The system can be applied to as many platforms, such as Linux and Windows, to satisfy client's demand.

B. System Components

The system is composed of four key modules (management module, web log preprocessing module,

database module and traffic generating module). This modular combination of system can build new generator to meet multiple requirements. Different functional model can be assigned to several develop team, which can simplify the software development, shorten the development cycle and enhance its scalability.

The system design is shown in Figure 2, and the four key functional modules are as follows:

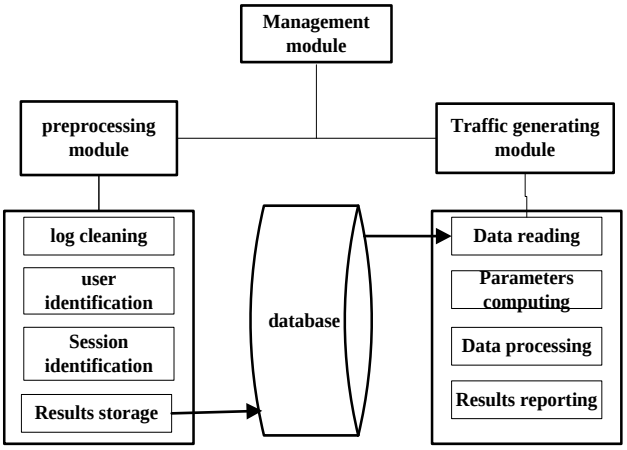

Figure 2. System design of the traffic generator

Management module. As the intermediate layer between the user and system, it takes charge of the system, distributes task，gather results, and handle bugs.

Preprocessing module. When web users access a real Web server, it returns corresponding web pages. Since a web page may contain a number of hyperlinks, we extract the hyperlink URLs embedded in such pages, and log them in a pool for next page candidate selection.

Database module. It stores the results of the web log process module, which can be used in the following traffic generating module.

Traffic generating module. As the core module in the system, its functions are as follows: (1) read the results of the log cleaning in Web log process model; (2) calculate the system's parameters, such as the Pareto Distribution parameter and Markov transition matrix; (3) interact with remote web server based on the access model, such as sending HTTP Get requests and receiving the responses.

V. TEST RESULTS

A prototype system has been developed and programmed in Python, which owns a number of complied and portable function modules. The system was implemented in a server with AMD Sempron 3800+ CPU, 2GB RAM, and running Fedora8 operating system. This server is a key part of Ocean, which is a network testbed used to evaluate research results of new protocols in Future Internet study, mainly address lacking of network background traffic generated by real user. Sixty threads were implemented concurrently, and each thread is corresponding to a Web user. We make the viewing time following Pareto Distribution (α=1.5) and extract about 100 pages to build a Markova transition matrix. We choose three time units, one second, 10 seconds and 30 seconds as sampling period for statistical analyses. This way, the test lasts about ten hours, and the results show that the traffic generated by the system has good burstiness and self-similarity.

A. *Burstiness test*

For self-similar traffic, burstiness remains regardless of the level of the aggregation because of the infinite variance of the source [11]. One way to observe this effect is by visually inspecting the time series plot vs such traffic with varying levels of aggregation [12]. In Figure 3, we show the traffic variations collected under different statistical period, respectively as 1 second (shown in Figure 3.a), 10 seconds (shown in Figure 3.b), and 30 seconds (shown in Figure 3.c). As in Figure 3, where the red line represents mean number of transmitted bytes under different statistical period, the traffic generated by our presented traffic generating system shows obvious burstiness. Moreover, the traffic burst does not significantly decrease as the time scale increased, which is consistent with the intrinsic characteristic of self-similar traffic flow.

B. *Self-similarity test*

Mathematically, a process X is called exactly (second-order) self-similar, if the aggregate process of X has the same correlation structure as X. The degree of self-similarity can be measured with Hurst parameter (H) [1]. Process X is self-similar when the value of H ranges from 0.5 to 1, and the more the Hurst parameter close to 1, the more self-similar the process is. We have applied two methods to compute the Hurst parameter, and the results are shown respectively in Figure 4.a and 4.b. When using R/S plot method, it is 0.73, and 0.72 when using variance-time plot method. The two Hurst parameters both reveal the self-similar nature of traffic generated by the system.

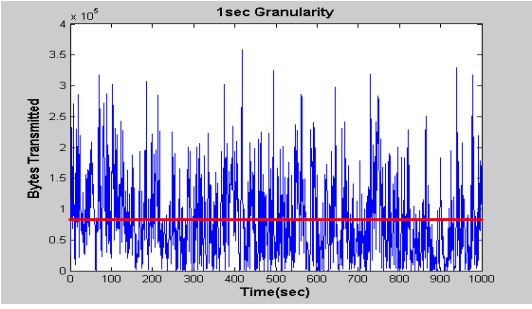

Figure 3.a. Traffic collected vs Time.
(Statistical period=1 Second)

 ISBN: 978-1-61208-285-1

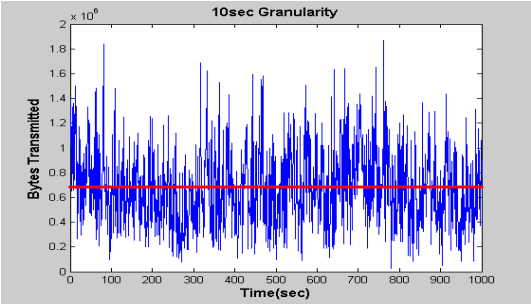

Figure 3.b. Traffic collected vs Time.
(Statistical period=10 Seconds)

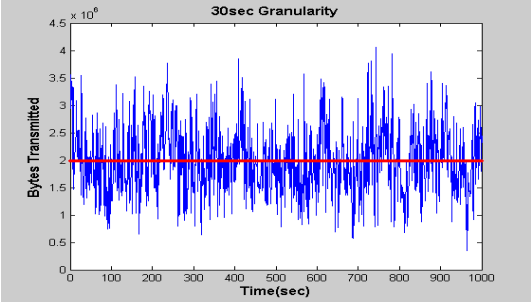

Figure 3.c. Traffic collected vs Time.
(Statistical period=30Second)

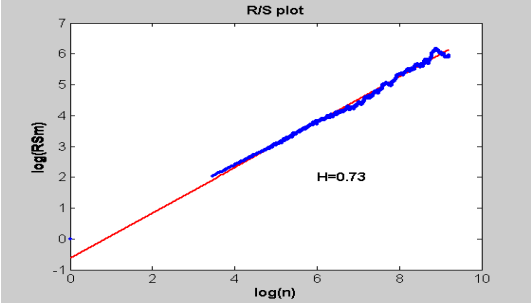

Figure 4.a. R/S plot of the traffic.
(H=0.73)

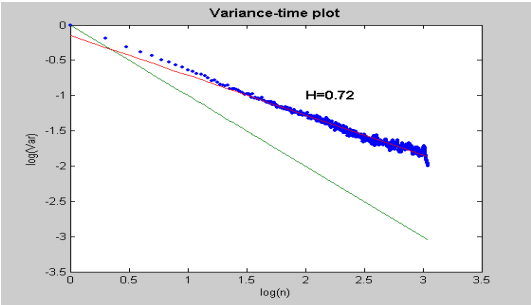

Figure 4.b. Variance-time plot of the traffic.
(H=0.72)

C. *Analysis of traffic self-similarity*

In our approach, On/Off model is used as a key part of user behavior model; we send page request to Web server according to the interval between ON state and OFF state. Threads in the system simulate ON or OFF sources, and the aggregation of them derives self-similar network traffic. However, why such approach makes our system generate self-similar traffic?

ON/OFF model has clear physical meaning. The data source is divided into two states, ON time and OFF time. Data source sends data in ON time, rather than OFF time. Take web user browsing as an example, a user sends Get requests and receives response from web server in the ON state, but there is no data transmit between the user and the server in the OFF state, which is regard as the user's thinking time.

It is assumed that the ON time and OFF time are independent and identically distribution. Suppose the duration of the ON state for the N (T), the duration of the OFF time for F(t), and the random variable N(t), F(t) for independent and identically Pareto distribution, then when aggregating enough ON/OFF sources, the generated network traffic is self-similar [13].

VI. CONCLUSION AND FUTURE WORK

In this paper, we proposed a web traffic generating method based on web users' access behavior model, and develop a traffic generating system. Compared to current traffic playback and traffic model based methods, our approach can simulate the real web users' accessing to real web servers, which are not pre-defined but determined by system clients, and generate almost the real network traffic. Test results show that traffic generated by our system has similar characteristics as the real web traffic, such as self-similarity and burstiness characteristics.

In future work, in order to improve the accuracy in simulating web users' behavior, we will polish the user behavior model, and make it more adapt the self-similarity nature of traffic. Moreover, we will extend this system so that it can generate other kinds of traffic, such as FTP, P2P, to meet different requirements.

ACKNOWLEDGMENT

This work is supported by the National Basic Research Program of China (2012CB315806) and the Natural Science Foundation of Chongqing (CSTC.2012JJB40008).

REFERENCES

[1] W. E. Leland, M. S. Taqqu, W. Willinger, and D.V. Wilson. "On the self-similar Nature of ethernet traffic(extended version)", Networking, IEEE/ACM Transactions on, vol. 2, no. 1, Feb 1994, pp. 1-15, doi: 10.1109/90.282603

[2] G. Xie, Y. Sun, Y. Zhang, Z. Li, H. Zheng, and X. Zheng. "Demo Abstract: Service-Oriented Future Internet Architecture (SOFIA)", IEEE Infocom/Poster, Shanghai, China, April 2011.

[3] Python, http://www.python.org/download/releases/2.6.8/, [retrieved:08.2012].

[4] Tcpreplay, http://tcpreplay.synfin.net/, [retrieved: 01.2013].

 ISBN: 978-1-61208-285-1

[5] SPEC, http://www.spec.org/osg/web99/, [retrieved: 01.2013].

[6] A Botta, A Dainotti, A Pescap é A tool for the generation of realistic network workload for emerging networking scenarios, Computer Networks, vol. 56, iss. 15, October 2012, pp. 3531-3547, ISSN 1389-1286, 10.1016/j.comnet.2012.02.019.

[7] S. Yu, G. Zhao, S. Guo, Y. Xiang, and A.V. Vasilakos. "Browsing behavior mimicking attacks on popular web sites for large botnets". Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on, April 2011, pp. 947-951, doi: 10.1109/INFCOMW.2011.5928949.

[8] P. Barford and M. Crovella. "Generating representative Web workloads for network and sever performance evaluation"[C]. ACM SIGMETRICS Performance Evaluation Review, pp. 151-160.

[9] P. Stuckmann, H. Finck, and T. Bahls. "A WAP Traffic Model and its Appliance for the Performance Analysis of WAP over GPRS". In Proc. of the IEEE International Conference on Third Generation Wireless and Beyond(3Gwireless '01) USA: San Francisco, 2001

[10] R. B. D' Agostino and M. A. Stephens. "Goodness-of-Fit Techniques"[M], Marcel Dekker, 1986.

[11] L.D. Catledge and J. E. Pitkow. "Characterizing browsing strategies in the World-Wide web"(J). Computer Networks and ISDN Systems. vol. 27, iss. 6, April 1995, pp. 1065–1073.

[12] H. Choi and J. Limb. "A behavioral model of Web traffic. Network Protocols", 1999. (ICNP '99) Proceedings. Seventh International Conference on, 31 Oct.-3 Nov. 1999, pp. 327-334.

[13] W. Willinger, M. S. Taqqu, R. Sherman, and D. V. Wilson. 1995. Self-similarity through high-variability: statistical analysis of ethernet LAN traffic at the source level. SIGCOMM Comput. Commun. Rev. vol. 25, no. 4, October 1995, pp. 100-113.

 ISBN: 978-1-61208-285-1