A New Self Similar Traffic Generator (SSTG)

Jahangir Dadkhah Chimeh, Education & Research Institute for ICT, Seyed Ali Alavian, The Faculty of Applied Science of Post and Communications

***Abstract* — Today Internet traffic is one of the most important traffics in the communication networks. We study different types of the Internet traffic in this paper. Besides, we design and develop an Internet traffic platform by the name of SSTG which generates Email, ftp, WWW, Telnet self similar traffic services. This platform also characterizes structure and timing diagram of them to the designer of the data communication systems to develop an algorithm.**

***Keywords* — Internet, Traffic Generator, Self similar.**

I. Introduction

Internet traffic is a dominant traffic in the existing communication networks. Internet traffic has more complexity relative to voice traffic. However, they have an important role in the new networks. Thus software developers have produced many kinds of Internet traffic software. The existing Internet traffic software are divided into three different groups. One is for evaluating the active Internet networks and the other is foe testing the new devices and the last is used in simulator softwares like NS and OPNET [1].

Besides, traffic generators are divided into software and hardware generators. Although hardware generators are more precise the software ones and have also better performance, they are expensive and also produced by the specific vendors. Software generators have less precision and are cheaper and generally made by the research and development centers and universities. The researchers use software generators not only for the economical reasons but also mainly for their easy deployment in many network nodes and ability to modify their codes to achieve the better performance [2].

Totally there are four groups of traffic generators as: application level generators, flow level generators, packet level generators and multilevel generators. A lot of largely used traffic generators work at packet level but multilevel generators are not made available to the users [3].

Transactions are divided into real time and non-real time processes. A real time transaction should be replied to during a definite time, otherwise it is a non-real time transaction. Real time transactions enter the network according to a Poisson process.

Besides, according to the delay sensitivity, services are divided into the following four groups[4], [5], [6]:

- Conversational services (with less than 100ms delay).
- Interactive services (with a delay of around 4 sec).
- Streaming services (with a delay of around 300ms).
- Background services (without any specific restriction).

Some of the services are error tolerable and some are non-error tolerable. Web browsing services are of interactive services and email and ftp are of background ones but all of them are error non-tolerable services. Indeed when a user issues a web page request, web browser program itself sends additional requests to the network to download the pictures (e.g., Icons, buttons, logos) in that page. Therefore, since the document entry process is a mixed process and not dependent to the user, it isn't a Poisson process [7]. Thus an ON/OFF process may be used for Internet traffic in which during the active (ON) periods the documents are transferred and during the inactive (OFF) periods no data information are transferred. OFF periods are usually longer than ON period. Thus in section II we explain the traffic model in the SSTG platform. In section III we explain the SSTG properties. In section IV we pay attention to the properties of some other Internet traffic platforms and finally we draw the conclusion.

II. Traffic Models in SSTG

In the designed platform we chose a traffic model to able the system developer to characterize the complete specifications of the requested traffic. We know that the file transfer, web browsing and email transfer are non-real time traffics. Since web browsing is one of the most popular services, to syrvey these traffics we choose it in one session. Users generate these traffics based on the possion distribution. Traffic model characterizes how the packet data enters a queue. We selected the hierarchical traffic model. The hierarchical WWW traffic model includes three layers named as session, packet call and packet (Fig. 1) [8], [9].

Aapart the traffic (session) generation behavior, a traffic model describes the packet call and packet behavior. The presented model offers analytical expressions for the number of packets in a packet call, packet length and interarrival time of packets. Indeed this model represents how packets are generated and how they are entered the buffer of the network.

Totally any connection may contain one or more sessions and also a session may contain one or more packets calls. Besides, there are one or more packets in a packet call. In other words, a packet call may be constituted from a bursty sequence of packets. Depending on the traffic types (email, ftp, …), a session may include one, two or more packet calls [8]. For example in the web browsing service, a session is constituted from some

Jahangir Dadkhah Chimeh, Ph.D., is faculty member of Education & Research Institute for ICT in Tehran, Iran (email: dadkhah@erict.ac.ir).

different pages that the user may study each one before downloading the other pages. Besides, a packet call represents the contents of a web page. After receiving a complete page in the mobile station (MS), the user studies it during the reading time. The study time is referred to the reading time. Then user issues the other requests, e.g., clicking on an Icon of that page. This indeed corresponds to request another packet call. Off course a session may only contain a packet call.

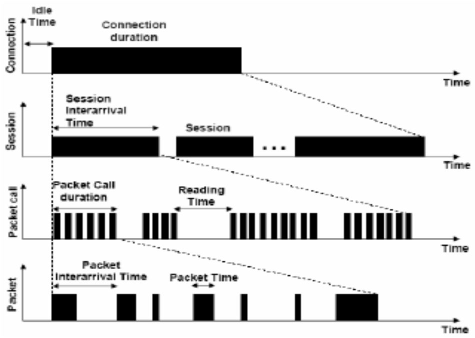

Fig. 1 A hierarchical model for www service [10], [11]

To achieve a behavior like Fig. 1 we should characterize some parameters as follow:

- Session arrival process,
- Number of packet calls in a session,
- Reading time between consecutive packets,
- The number of packets in a packet call,
- Interarrival times between the two consecutive packets in a packet call.
- And the packet size.

Besides, in order to catch the typical behavior described in Fig. 1 the following statistical processes may be applied to the web browsing service [11], [12]:

- Session arrival process

The arrival of session set-up request to the network is modeled as a Poisson process with λ rate.

- Number of packet calls per session, *Npc*

This is a geometrically distributed random variable with a mean μ_{Npc} [packet calls], i.e.,

$$N_{PC} \in Geom(\mu_{N_{PC}}) \tag{1}$$

- Reading time between packet calls, *Dpc*

This is a geometrically distributed random variable with a mean μ_{Dpc} [reading time steps], i.e.,

$$D_{PC} \in Geom(\mu_{D_{PC}}) \tag{2}$$

- Number of datagrams within a packet call, N_d can be geometrically distributed random variable with a mean μ_{N_d} [packet], i.e.,

$$N_d \in Geom(\mu_{N_d}) \tag{3}$$

- Inter arrival time between datagrams (within a packet call) D_d. This is a geometrically distributed random variable with a mean μ_{D_d} [model time steps], i.e.,

$$D_d \in Geom(\mu_{D_d}) \tag{4}$$

- Size of a datagram, S_d.

Pareto distribution with cut-off is used for packet size. Packet size is defined with the following formula:

PacketSize=min(x, m) (5)

where *x* is normal Pareto distributed random variable (α =1.1, k=81.5 bytes) and *m* is maximum allowed packet size, *m*=66666 bytes [11]. The PDF of the Packet size becomes:

$$f_x(x) = \begin{cases} \dfrac{\alpha.k^{\alpha}}{x^{\alpha+1}} & , \quad k \le x \le m \\ \\ \beta & , \quad x > m \end{cases} \tag{6}$$

where β is the probability that x>m. It can easily be calculated as:

$$\beta = \int_m^{\infty} f_x(x)dx = \left(\frac{k}{m}\right)^{\alpha} \qquad \alpha > 1 \tag{7}$$

In (8), the requisite that $\alpha > 1$ is only to ensure that the E[X] of the Pareto distribution exists. The result (7) is valid for all $\alpha > 1$.

Then mean packet size can be calculated as:

$$\mu_x = \int_{-\infty}^{\infty} x f_x(x)dx = \int_k^{m-} x \frac{\alpha.k^{\alpha}}{x^{\alpha+1}}dx + m\left(\frac{k}{m}\right)^{\alpha} \tag{8}$$

And after simplifying we have

$$\mu_x = \frac{\alpha k - m\left(\dfrac{k}{m}\right)^{\alpha}}{\alpha - 1} \tag{9}$$

With the parameters above the average packet size is $\mu_x = 480$ bytes.

Table 1 gives default mean values for the distributions of some typical www services of different rates. As a consequence, the average size of a packet call is $\mu_{N_d}.\mu_x = 25 \times 480 = 12kbytes$.

TABLE 1: CHARACTERISTICS OF A WWW TRAFFIC IN DIFFERENT RATES [11].

Packet based information types	Average number of packet calls within a session	Average reading time between packet calls [s]	Average amount of packets within a packet call []	Average interarrival time between packets [s][1]	Parameters for packet size distribution
WWW surfing UDD 8 kbit/s	5	412	25	0.5	k = 81.5 (= 1.1
WWW surfing UDD 32 kbit/s	5	412	25	0.125	k = 81.5 (= 1.1
WWW surfing UDD 64 kbit/s	5	412	25	0.0625	k = 81.5 (= 1.1
WWW surfing UDD 144 kbit/s	5	412	25	0.0277	k = 81.5 (= 1.1
WWW surfing UDD 384 kbit/s	5	412	25	0.0104	k = 81.5 (= 1.1
WWW surfing UDD 2048 kbit/s	5	412	25	0.00195	k = 81.5 α = 1.1

Fig. 2 also depicts sizes of 1000 packets according to the Pareto distribution.

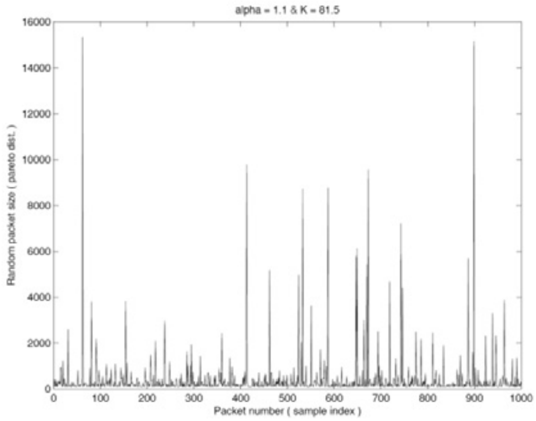

Fig. 2 1000 packets with mean of 480

III. Properties of SSTG

This software is basically based on the MATLAB platform and is written based on the hierarchical model expressed in [11] and also uses self similar and mathematical functions expressed in the previous section. In a session, traffic is generated in the form of the packet calls and packet and the traffic self similarity is also due to the Pareto function used for the packet length. In this platform the number of traffic services are entered from the input (user interface). Then in addition to the reading time, the user selects the data rate from the options 8k, 32k, 64k, 144k, 384k, 2048k. Besides, the average number of packet calls are selected based on the traffic types [8]. Then the number of the packets are chosen. Finally based on the Pareto distribution the number of bytes in each packet is selected. Here, we are capable to find the start and end times of all objects in a session which are stored in a "Tuser" traffic matrix. Since the traffic services are structurally different, to start the SSTG, the user first selects one of the four different traffics telnet, ftp, email and web browsing.

Thus the SSTG platform provides the user the following capabilities:

- Generation of desired traffic telnet, ftp, email and web browsing according to the request,
- Determination the number of the users,
- Determining/Finding the number of packet calls' and packets' start and end times,
- Determining/Finding the number of packets in a packet call, their length and start and end times,
- Finding the position of each bit,
- Plotting all the above parameters,
- All the above Internet traffic services are stored as dat extension suffix.

Figs. 3-5 depict the generated WWW traffic by this platform. Fig. 3 depicts a session of the web browsing which includes five packet calls. Fig. 4 depicts first packet call of Fig. 3 which includes 14 packets. Fig. 5 depicts second packet call of Fig. 3 which includes 9 packets.

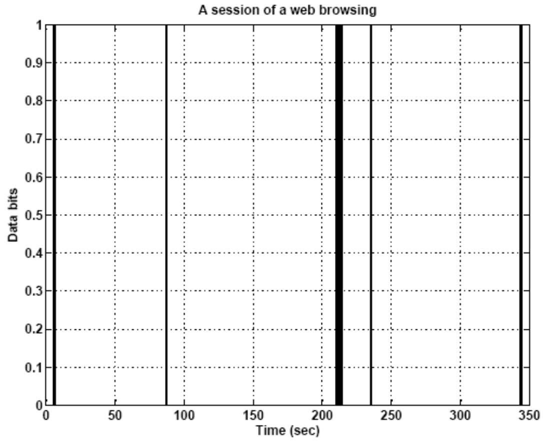

Fig. 3 Timing diagram of a WWW session

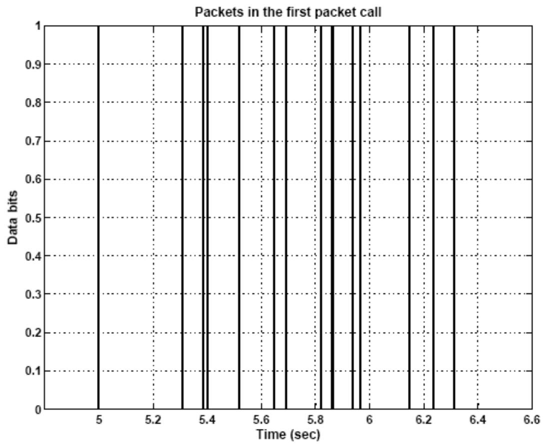

Fig. 4 Timing diagram of packets in the first packet call

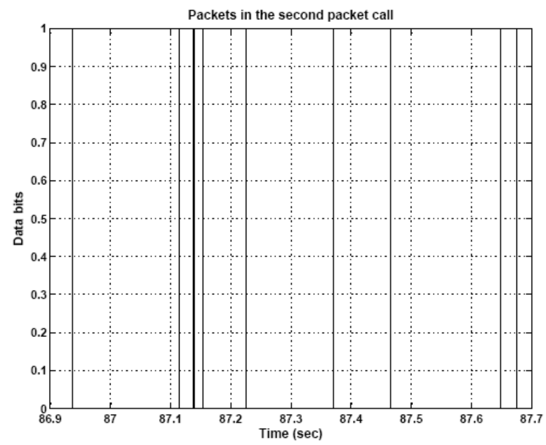

Fig. 5 Timing diagram of packets in the second packet call

For example in Fig. 3, start and end times of the first packet call are 5 sec and 6.443 sec respectively and start and end times of the second packet call in that figure are 87.114sec and 87.667 sec respectively. Besides, start and end times of the first packet in first packet call are 5 and 5.0001 sec respectively and start and end times of the first packet in the second packet call are 87.1140 sec and 87.1145 sec respectively.

IV. Some other traffic generators

We briefly pay attention to some other traffic generators in this section. There are more than 32 Internet traffic generators on [1] and [2] web. These softwares are used to measure the network performance matrices e,g., packet transfer delay, round trip time. These softwares are generally under Linux and work through its command line. Their generated traffics are generally UDP, TCP and HTTP. Mtools, Trafgen, Network Traffic Generator by rsandila and RUDE & CRUDE in [1] and [2] are some of them.

V. Conclusion

As it was shown the SSTG platform is basically different from the existing commercial platforms. This platform calculates the precise time stamps of the generated traffics and exposes them to the system developer. These quantities can be used for the research and development of the communication systems. On the other hand, generated traffic is of the Application layer type while the generated traffic in the existing softwares are of the transport layer type. Besides, this platform operates under Windows operating system while the existing softwares operate in the command line of the Linux operating system.

Acknowledgment

Here it is necessary to thank the Education & Research Institute for ICT (ERICT) for the financial and intellectual support of this research.

References

[1] http://www.icir.org/models/trafficgenerators.html.

[2] http://www.grid.unina.it/Traffic/index.php.

[3] A. Botto, et al., “Do you trust your software-based traffic generator”, IEEE Communications Magazine, pp. 158-165, Sep. 2010.

[4] ETSI TS 123 107, “Quality of Service (Qos) Concept and Architecture”, V4.6.0, 2002-12.

[5] L. Skorin_kapov, D. Huljenic, E. N. Tesla, “Analysis of End-to-End QoS for Networked Virtual Reality services in UMTS“, IEEE Communications Magazine, April 2004, pp. 49-55.

[6] D. Soldani, J. Laiho, “A Virtual Time Simulator for Studying QoS Management Functions in UTRAN”, Vehicular Technology Conference, 2003, IEEE 58th, Vol. 5, pp. 3453-3457.

[7] Tero Ojanpera and Ramjee Prasad (Editors), WCDMA: Towards IP Mobility AND Mobile Internet, Artech House Publishers, Boston, London, 2001.

[8] J. Dadkhah Chimeh et al., Internet Traffic Modeling and Capacity Evaluation in UMTS, International Journal of Hybrid Information Technology, Vol. 1, No. 3, July, 2008, , p.p. 109-120

[9] Van Nguyen, Trung, Capacity Improvement Using Adaptive Sectorization in WCDMA Cellular Systems With Non-Uniform and Packet Mode Traffic, Victoria University, Melbourne, PhD Thesis, March 2005.

[10] N. D. Tripathi, “Simulation Base analysis of the radio interface performance of an IS-2000 system for various data services”, VTC 2001 Fall, IEEE VTS 54th.

[11] 3GPP TR 101 112, "Selection procedures for the choice of radio transmission technologies of the UMTS", V3.3, 2004, www.etsi.org.

[12] D. Soldani, QoS Management in UMTS Terrestrial Radio Access FDD Networks, Helsinki University of Technology, PhD Thesis, October 2005.