Modelling self-similar traffic in networks

S. Sharkovsky, E. Grab

It is widely known, that in our days traffic in computer networks mostly is self-similar (fractal).Unfortunately, this self-similar traffic, unlike standard Markov traffic, does not have adequate model. In such conditions, in order to ensure efficient processing in networks with self-similar traffic, it is necessary to allocate significantly greater resources, compared to ones in case of the standard Markov traffic model used before. Such difference demands to be very careful when estimating parameters of flows in traffic control and management systems. In this article it is proposed to use for simulation purposes MATLAB software pack and its Simulink environment.

Keywords: Self-similar traffic, Pareto Distribution, Weibull Distribution, Simulink.

I. INTRODUCTION.

It is widely common for network traffic to be self-similar (fractal). This has been confirmed long time ago by many researchers, for example in [1], [2]. In such situation the memory amount required for the traffic increases by power or by Weibull law, unlike it has been in classic Markov service systems, where memory amount was increasing exponentially. For example, the standard Markov traffic with utilization $\rho = 0.8$ and buffer overflow probability $P_{Loss} = 10^{-6}$ would require memory capacity of $K = 55$ data packets. Compared to this, the self-similar traffic with same utilization level and buffer overflow probability would require memory capacity of $K = 5 \cdot 10^8$ packets estimated by formula from [2], for Hurst parameter of $H = 0.9$ (very high self-similarity) in order to provide efficient service. When designing traffic control and admission systems, such difference requires caution when estimating parameters of the data flows.

When examining a system it is very practical to create a model. In this article we propose to use MATLAB software pack with Simulink environment. Our choice is based on following:

- Simulink allows the use of powerful SimEvents blockset [3], which gives option to create models of queuing systems;
- Simulink allows to use any MATLAB libraries for traffic estimation and control blocks, as well as any custom blocks made by user.

Everything of that beneficially excels MATLAB amongst other modeling environments. Simevents blockset includes models for:

- Traffic sources,
- Queues (including prioritized),
- Servers and other blocks.

Note, that SimEvents toolbox is not limited in any case by traffic modeling for networks. It provides powerful tools for other applications as well. A very good example of this – the article, which describes simulation of the railway operation [4] in Simulink, using SimEvents blockset.

II. MODELING G/M/1/K QUEUING SYSTEMS

As it has been mentioned before, the self-similar traffic prevails in computer networks. In order to create such traffic, it is necessary for probability distribution function (PDF) of the packet interarrival time to have a “long tail”. Two of the most often used distributions with such property are Weibull distribution and Pareto distribution. Unfortunately, SimEvents blockset doesn’t include neither one of these distributions for source entity generators (it does include Weibull distribution for source number generators, however). It is possible, however, to use any external random-number generator to specify time intervals between two events. Although there is random-numbers generators for Weibull distribution, we have to estimate distribution parameters first, since parameters of our system (intensity of requests, Hurst parameter) can’t be used directly. There was no Pareto random number generator in Simulink at the present moment at all.

First, we will create G/M/1/K model in Simulink with Weibull-distributed interarrival time of requests. Unlike Pareto distribution, which also provides means of changing self-similarity of generated data flows, Weibull distribution can lead to analytical expressions. Weibull distributed random numbers PDF is as follows:

$$f(x) = \alpha\beta^{-\alpha}x^{\alpha-1}e^{-\left(\frac{x}{\beta}\right)^{\alpha}}, \qquad (1)$$

where α and β are distribution parameters. These parameters not necessarily are the same as system’s parameters. In our case the system can be described with following parameters:

- λ – average arrival rate of requests;
- μ – service facility rate;
- K – buffer-memory capacity (queue length);
- H – self-similarity (Hurst) parameter, which varies in $0.5 \le H \le 1$.

These parameters can’t be used directly in (1), we have to estimate them first. Weibull distribution α parameter can be estimated by H parameter [2]:

$$\alpha = 2 - 2H. \qquad (2)$$

Weibull distribution β parameter can be estimated by α parameter and average inter-arrival time ($\mathrm{E}[x] = 1/\lambda$), i.e.:

$$E[x] = \beta\left[\Gamma\left(1 + \frac{1}{\alpha}\right)\right]. \qquad (3)$$

where $\Gamma(\ )$ is gamma function, which can be numerically estimated, see [5] for details. In such case we can express β parameter from (3) as follows:

$$\beta = \frac{1}{\lambda\left[\Gamma\left(1+\frac{1}{\alpha}\right)\right]}. \qquad (4)$$

In G/M/1/K model (and in P/M/1/K as well), serving time is distributed exponentially with mean service time $1/\mu$. The maximal length of queue is specified by memory capacity K. When distribution parameters have been estimated by calculating (2) and (4), Weibull distributed random number x can be generated by calculating following expression:

$$x = \beta[-\ln(1-R)]^{1/\alpha}, \qquad (5)$$

where $R \in [0;1]$ is uniformly distributed random number.

Fig. 1. illustrates full G/M/1/K model. The Time-Based Entity Generator generates events with time intervals specified by Weibull distributed numbers generator. The Weibull numbers generator has been made as a subsystem with mask, which calculates distribution parameters according to (2) and (4). Subsystem itself is illustrated in Fig. 2.

The subsystem includes uniform distribution number generator, two constant parameters (α and β) and Embedded Function block, which implements Matlab written function for generation law (5).

The model includes "Simout" block as well for verification purposes. It writes all generated random numbers in Matlab workspace, which allows us to operate with these numbers. Our goal here is to make sure, that generated numbers PDF is indeed Weibull law's PDF.

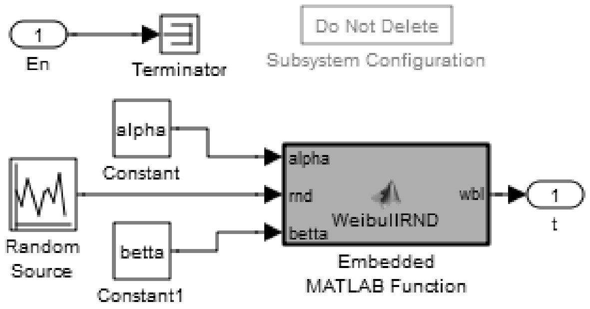

Fig. 2. Discrete event subsystem. Event-based random number generator with Weibull distribution.

The data from simulation can be used to make a histogram. This histogram can be compared to actual PDF specified by (1). Fig. 3. illustrates such comparison for 2 hours simulation of the model shown in Fig. 1. on average computer (approximately $5.6 \cdot 10^7$ random numbers). The histogram in Fig. 3. shows evaluated PDF from numbers used in simulation with following parameters:

- $\lambda = 100$ s^{-1} – average arrival rate of requests, average interarrival time $E[T_a] = 0.01$ s;
- $\mu = 125$ s^{-1} – service facility rate, average service time $E[T_s] = 0.008$ s;
- $K = 24100$ – buffer-memory capacity (queue length);
- $H = 0.9$ – self-similarity (Hurst) parameter, $0.5 \le H \le 1$.

The actual PDF was calculated according to (1) by using built-in Matlab function for PDF estimation. As Fig. 3. shows that created discrete event subsystem can be used to generate Weibull distributed traffic with specified self-similarity parameter.

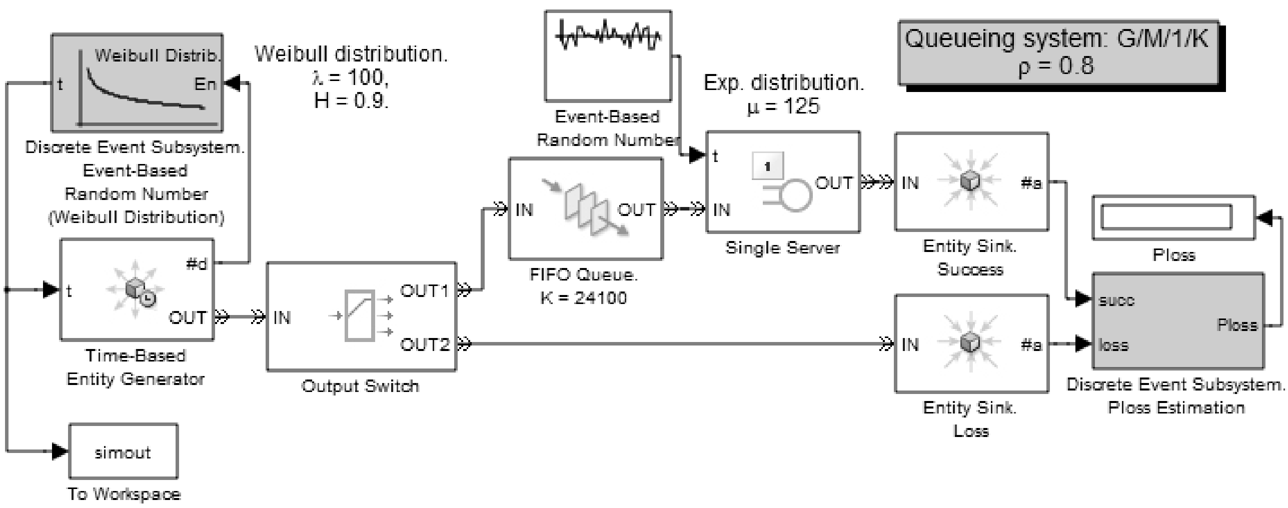

Fig. 1. Queuing system with Weibull distributed request interarrival time (G/M/1/K model).

III. MODELING P/M/1/K QUEUING SYSTEMS

The P/M/1/K model is very similar to previous model. In this model the source is described as more commonly used Pareto distributed random event generator.

The model itself remains the same, therefore we only need to use another random number generator to specify time intervals between events for the event-based generator. Pareto distribution function, in general has 3 parameters, however for network traffic generation the most commonly used form of Pareto distribution function [2] is:

$$f(x) = \frac{\alpha}{\beta}\left(\frac{\beta}{x}\right)^{\alpha+1} \quad (6)$$

where α an β d are distribution parameters. As it has been for Weibull distribution, these parameters are not the same as system's parameters. In order to generate Pareto distributed random numbers, we need to estimate these two parameters first. Pareto distribution α parameter can be estimated by H parameter [2]:

$$\alpha = 3 - 2H. \quad (7)$$

Pareto distribution β parameter can be estimated by α parameter and average inter-arrival time ($\mathrm{E}[x] = 1/\lambda$), i.e.:

$$E[x] = \frac{\alpha\beta}{\alpha - 1}, \quad \alpha \geq 1. \quad (8)$$

In such case we can express β parameter from (8) as follows:

$$\beta = \frac{\alpha - 1}{\alpha\lambda}. \quad (9)$$

When distribution parameters have been estimated by calculating (7) and (9), Pareto distributed random number x can be generated by calculating following expression:

$$x = \frac{\beta}{R^{1/\alpha}}, \quad (10)$$

where $R \in [0; 1]$ is uniformly distributed random number,.

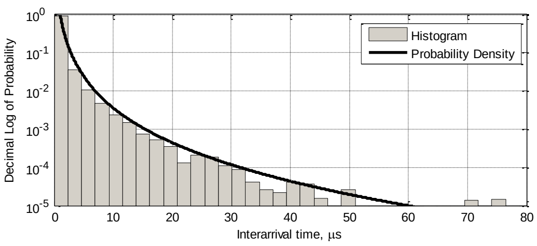

Fig. 3. Experimental histogram for generator used in G/M/1/K model and PDF of Weibull distribution law.

The model used for simulation remains the same (Fig. 1.), including all system parameters. The mask of subsystem in Fig. 2. has been modified according to (7) and (9), and Embedded Matlab Function block has been modified according to (10) as well.

The same way as it has been done for previous simulation results, we have calculated histogram of the experimental data and actual Pareto PDF, showing them on same plot, as it is shown in Fig. 4. Note, however, that Pareto PDF "long tail" is much longer, compared to Weibull PDF "long tail", so we had to cut it at value of 200 (with maximum value of approximately 4000).

The actual PDF was calculated according to (6) by using built-in Matlab function for PDF estimation (without specifying the 3-rd parameter of Generalized Pareto distribution). The Fig. 4. shows that created discrete event subsystem can be used to generate Pareto distributed traffic with specified self-similarity parameter.

IV. MODELING P/M/1/K QUEUING SYSTEMS WITH ON-OFF TRAFFIC SOURCE

The previous two simulations can be improved to describe modern traffic in networks even more precisely. The traffic in networks is not only self-similar, it also usually comes in groups of packets followed by intervals of "silence". So the traffic has ON-OFF type structure. This effect can be added in Simulink, as is shown in Fig. 5. for Pareto distributed self-similar traffic. To create such model for any other type of source events distribution law it is sufficient to change that law in respective discrete event subsystem (in this article: Weibull number generator and Pareto number generator).

The ON-OFF traffic structure is simulated by Enabled Gate, which is operated by a service unit (i.e. Server). Service time intervals specify duration of the ON phase and for this purpose it is possible to use Event-Based Random Number block, which supports most common distributions.

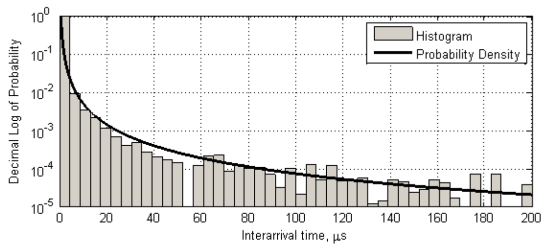

Fig. 4. Experimental histogram for generator used in P/M/1/K model and PDF of Pareto distribution law.

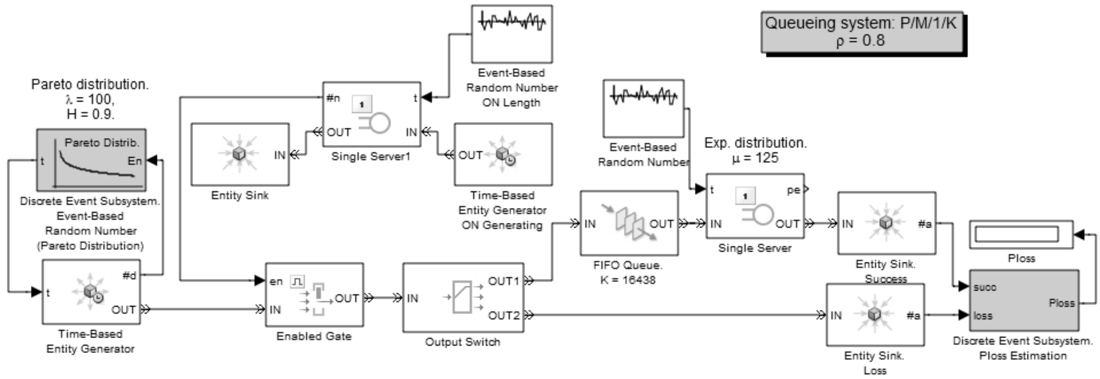

Fig. 5. Queuing system with Pareto distributed request interarrival time (P/M/1/K model) and ON-OFF traffic structure.

V. CONCLUSIONS

This article shows possibility to use Simulink environment with SimEvents blockset for network traffic modeling purposes. Although this blockset has many available blocks, it is necessary to create custom blocks, for example to create ON-OFF structure self-similar traffic with Pareto distribution, as it is described in this article.

Comparison of generated random numbers histograms with respective PDF of Pareto and Weibull law shows, that these generators can be used for generating self-similar traffic, considering it can be described by these distributions. At the moment, this is considered to be true.

REFERENCES

[1] Will E. Leland, Murad S. Taqqu, Walter Willinger, Daniel V. Wilson, "On the self-similar nature of Ethernet traffic," IEEE/ACM Transactions on Networking, Volume 2, Issue 1, Feb. 1994, pp. 203-213.1955.

[2] Лосев Ю.И., Руккас К.М., "Анализ моделей вероятности потери пакетов в буфере маршрутизатора с учетом фрактальности трафика," Вестник Харьковского Национального университета. Серия «Математическое моделирование. Информационные технологии. Автоматизированные системы управления.», Выпуск 10, 2008, c.163-169.

[3] Mathworks Matlab Documentation, "SimEvents Model and simulate discrete-event systems", Internet: http://www.mathworks.se/help/toolbox/simevents/index.html

[4] VALIGURA, K., FOLTIN, M., BLAHO, M. "Transport system realization in SimEvents tool", Slovak University of Technology in Bratislava, Faculty of electrical engineering and information technology, unpublished.

[5] Cody, J., An Overview of Software Development for Special Functions, Lecture Notes in Mathematics, 506, Numerical Analysis Dundee, G. A. Watson (ed.), Springer Verlag, Berlin, 1976.

[6] Devroye, L. Non-Uniform Random Variate Generation. New York: Springer-Verlag, 1986.

Sergey Sharkovsy, education place - Riga Technical University, Faculty of Electronics and Telecommunications, Transport Electronics and Telematics Department. Doctor's degree in transport electronics and telematics earned in 2009.

Current assignments – docent (since 2009), participating in department's research projects and lector. Field of interests: digital signal processing, network traffic. Current work place: Lomonosova 1, V-107 cabinet, Riga, Latvia.

Author's e-mail address: sergejs.sarkovskis@rtu.lv

Elan Grab, education place - Riga Technical University, Faculty of Electronics and Telecommunications, Transport Electronics and Telematics Department. Bachelor's degree in transport electronics and telematics earned during 2004 - 2007. Professional master's degree in transport electronics and telematics earned during 2007 - 2010. Currently undertaking doctoral studies at the same place and program.

Formerly worked at Hansa Electronics factory as a manufacturing process operator in 2006 – 2007. Currently working at the university simultaneously with studies at the same place since 2007. Current assignments – science assistant, participating in department's research projects, and lector – reading following courses: "Transmission of Information and Transport Digital Communication Systems" and "Principles of Communication Systems" to undergraduate students.

Current work place: Lomonosova 1, V-308 cabinet, Riga, Latvia. Author has been awarded the following rewards:

1. The 2-nd level Samsung Electronics Annual Grant for good results in studies.
2. Diploma of the 49-th RTU Student Conference of Science and Technics

Author's e-mail address: konnektor@gmail.com

Sergejs Šarkovskis, Elans Grabs. Sevlīdzīgā trafika modelēšana datortīklos.

Jau plaši zināms, ka mūsdienu datortīklu trafikam piemīt sevlīdzīgā (fraktālā) struktūra. Diemžēl, tādam sevlīdzīgajam trafikam, atšķirībā no standarta Markova trafika, nepastāv adekvātais analītiskais modelis. Tādos nosacījumos, lai nodrošinātu efektīvo datortīklu darbību ar sevlīdzīgo trafiku, nepieciešams izdalīt ievērojami vairāk resursu, nekā standarta Markova trafika modeļa gadījuma. Tāda atšķirība pieprasa lielu uzmanību aprēķinot plūsmu parametrus trafika vadības un kontroles sistēmās. Šajā rakstā tiek piedāvāts izmantot simulācijas nolūkiem datorpaketi MATLAB un tajā iekļauto vidi Simulink.

Сергей Шарковский, Элан Граб. Моделирование самоподобного трафика в компьютерных сетях.

Уже давно известно, что трафик в современных компьютерных сетях по большей части имеет самоподобный (фрактальный) характер. К сожалению, для самоподобного трафика, в отличие от стандартного Марковского трафика, не существует адекватной аналитической модели. При таких условиях, для того чтобы обеспечить эффективную работу компьютерных сетей с самоподобным трафиком, необходимо выделять значительно больше ресурсов, в сравнении с их количеством для случая модели стандартного Марковского трафика. Такое различие требует большой осторожности при расчете параметров потока в системах контроля и управления трафика. В данной статье для симуляции работы таких систем предлагается использовать программный пакет MATLAB и входящую в него среду Simulink.