A network traffic sources modeling method based on measured data defragmentation

Matjaž Fras
Margento R&D
Margento d.d.
2000 Maribor, Slovenia
matjaz.fras@margento. com

Jože Mohorko, Žarko Čučej
Faculty of Electrical Engineering and Computer Science
University of Maribor
2000 Maribor, Slovenia
zarko.cucej@uni-mb.si

Abstract— Over recent years, the need for simulating complex communication networks, in order to assist evaluation, construction, and the upgrading of communication networks has become a key element in optimizing the exploitation of particular networks. In this regard traffic modeling has a key impact on network simulation reliability and, consequently, usability. For these reasons, we have developed and compared two algorithms of traffic source modeling. Both algorithms are based on mimicking a defragmentation process. They come from the captured packet estimated parameters of the probability density function regarding data files sizes and files inter-arrival times processes. The first one uses in-depth analysis of packet headers for estimating of data file lengths, and the second one does this by measuring packets' lengths and identifying the source and destination IP addresses. Both algorithms consider a TCP/IP encapsulation process.

Keywords- network traffic, traffic modeling, traffic simulation, statistic parameters' estimation.

I. Introduction

Self similar network traffic [1, 2, 4, 5] is usually modeled from an application point of view. It is usually supposed that the statistics of file sizes and file inter-arrival times are a known [3]. Such kinds of traffic models are supported by most commercial telecommunication simulation tools such as the OPNET Modeler [14, 16, 17, 19, 20, 21], as used in our simulations and experiments. Since packets are generated from files by an encapsulation process, in those cases where files are larger than a packet payload, they are fragmented so that packets are no bigger than the Maximal Transmission Unit (MTU) size. Consequently, for using the measured data of packet traffic when modeling file statistics, it is necessary to transform packet statistics into file statistics [9, 10]. This transformation contains opposite operations to the fragmentation and encapsulation process. The file distribution parameters can be estimated from histograms of transformed statistics by probability density function (pdf) fitting tools [11, 12, 13, 22] or methods, such as CCDF [5, 6] or Hill's estimator [15].

II. Packet Trafic Statistic

The self-similar network packet traffic [5, 6, 18] $Z_p(t)$, which usually poses long-range dependence [7, 8], can be generally described using a combination of two stochastic processes:

$$Z_p(t) = \psi\left\{X_p(t), Y_p(t)\right\} \tag{1}$$

where ψ is the function which depends on the packet-size process $X_p(t)$ and inter-arrival time process $Y_p(t)$. These processes are generated in a higher layer of the TCP/IP reference model from equivalent processes $X_f(t)$ and $Y_f(t)$ performed in data sources (Fig. 1). These layers at data transmission, perform fragmentation and encapsulation of data files into packets and, on reception, the defragmentation and decapsulation of packets into data files:

$$X_m(t) \underset{\substack{decapsulation\\ defragmentation}}{\overset{\substack{fragmentation\\ encapsulation}}{\rightleftarrows}} X_f(t) \tag{2}$$

$$Y_m(t) \underset{\substack{decapsulation\\ defragmentation}}{\overset{\substack{fragmentation\\ encapsulation}}{\rightleftarrows}} Y_f(t) \tag{3}$$

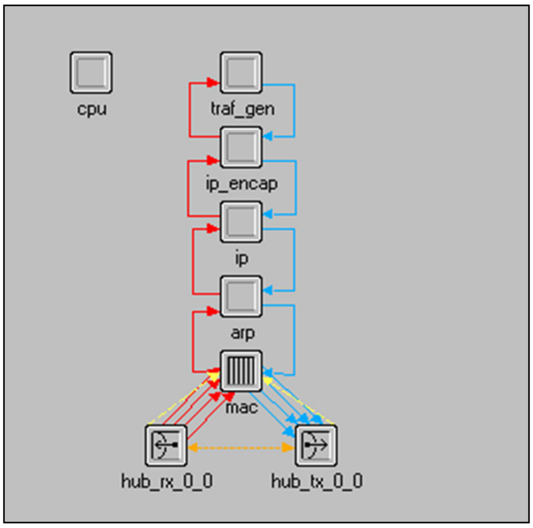

Figure 1: Node model for used IP station in simulation

Let us suppose, that the measured $Z_m(t)$, and modeled $Z_s(t)$ packet traffics are statistically equal, i.e.:

$$Z_m(t) \approx Z_s(t)\,, \qquad (4)$$

where symbol ≈ is used for statistical equivalence, then also constituent of the Z process, i.e. X and Y of measured and modeled packet traffic had to be statistically equal:

$$X_m(t) \approx X_s(t) \quad \text{and} \quad Y_m(t) \approx Y_s(t) \qquad (6)$$

Since $Z_s(t) = \psi\{X_s(t), Y_s(t)\}$ is generated from $Z_f(t)$ by a fragmentation process, then the defragmentation of $Z_m(t)$ gives $Z_f(t)$, which is the searched for result.

There are many possibilities how estimating files' lengths and their inter-arrival times at the application layer of the communication model. We investigated and compared the results of two algorithms:

1. algorithm with in-depth analysis of all packet headers,
2. algorithm with coarse inspection of IP header only.

III. PACKETS' STATISTICS TRANSFORMATION

Both algorithms calculate histograms of file source statistics. The main differences between them are complexity and the needed execution time. The first algorithm mimics a complete decapsulation process, and defragmentation in higher layers of the communication model; the second skips decapsulation by considering the average lengths of packet headers and then uses only packet lengths and inter-arrival times. These differences cause differently obtained results, but the differences are negligible, as analyzed later.

A. *First algorithm*

Each packed IP header has four so called fragmentation fields containing information about data fragmentation (Fig. 2).

0	4	8	16		32
V	IHL	ToS	TL		
ID			F	FO	
TtL		protocol	header check sum		
source address					
destination address					
options + padding					

Figure 2: IP header. Shadowed fields are used in the defragmentation process. ***Legend***: V: protocol version; IHL: Internet Header Length; ToS: Type of Service; TL: Total Length; ID: Identification Data; F: Flags; FO: Fragment Offset; TtL: Time to Live.

Any sniffers are able to extract these data from the IP header. Knowing them, it is then simple to calculate a length of IP PDU (Protocol Data Unit) which also contains a header of higher layer protocols. Using in-depth header analysis, it is possible, in the similar way to the IP header, calculate the lengths of all these headers.

The first algorithm (Fig. 4) completely respects RFC 793 [26]. It calculates exactly the lengths of the source files, and sorts these lengths within a histogram of the process $X_f(t)$.

From the time stampings of the first and last packets originated from the same file, it calculates the inter-arrival times and sorts the results within the histogram of the process $Y_f(t)$.

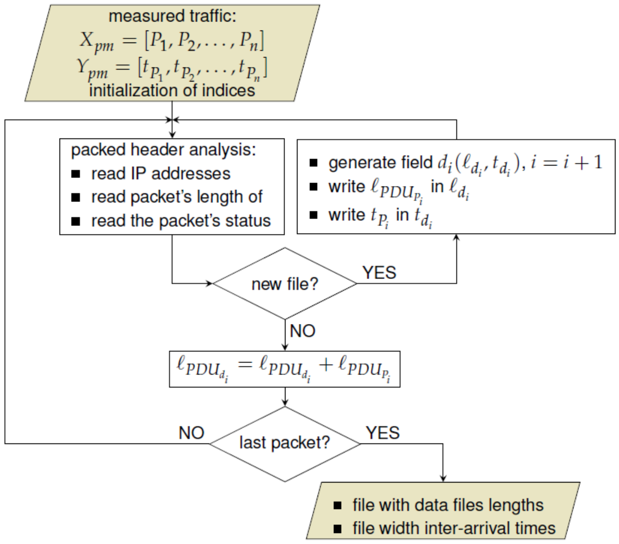

Figure 4: Simplified flowchart of the first algorithm [24].

B. *Second algorithm*

The second algorithm (Fig. 5), from all the data contained in the packets' headers, uses only source and destination IP addresses. In addition, it uses the time stamping and packets' length data provided by the sniffer. The step from histograms of $X_f(t)$ and $Y_f(t)$ to the parameters of the pdf of these processes is made by EasyFit tool [26].

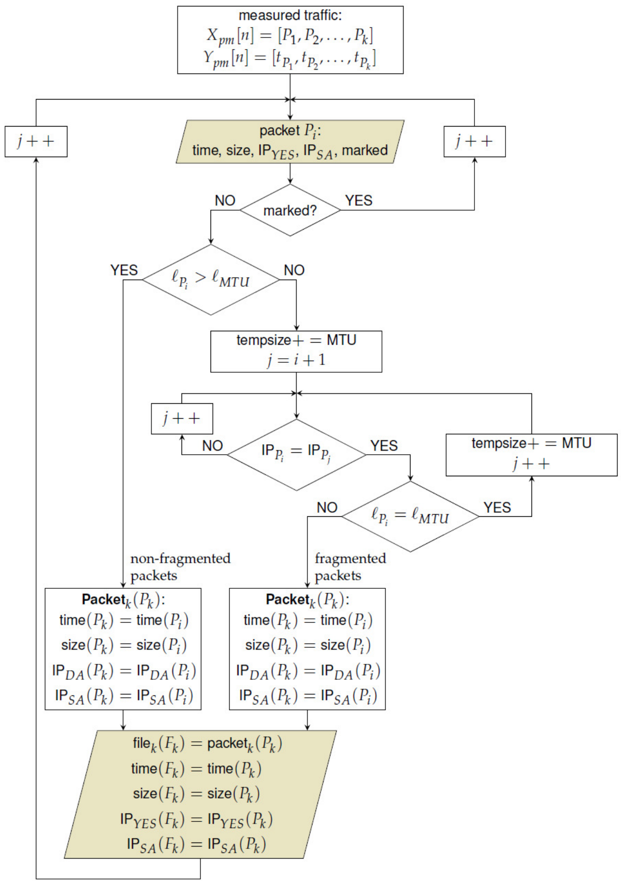

Figure 5: Flowchart of the second algorithm [25].

The second algorithm work as follows. If the packet with the new IP source and/or destination address is the result of fragmentation, i.e. $\ell_{PDU_i} = \ell_{MTU}$, then an algorithm investigates all packets with the same destination and source IP address, in sequence, to those first packets shorter than ℓ_{MTU} . This algorithm considers packets that belong to the same file and sums up their lengths and the length of the originating file, and from them subtracts the average lengths of the IP and TCP headers of each packet (Fig. 6).

Captured traffic $Z_m(t)$

No.	Time	Packet size	Source	Destination
1	0.0000	70	192.168.1.1	192.168.1.2
2	0.1000	1502	192.168.1.1	192.168.1.2
3	0.1030	1502	192.168.1.1	192.168.1.2
4	0.1050	1502	192.168.1.1	192.168.1.2
5	0.1090	540	192.168.1.1	192.168.1.2
6	0.3000	65	192.168.1.1	192.168.1.2
7	0.5000	78	192.168.1.1	192.168.1.2
8	0.6050	1502	192.168.1.1	192.168.1.2
9	0.6150	78	192.168.1.3	192.168.1.2
10	0.6200	58	192.168.1.1	192.168.1.2

Tranformed traffic $Z_s(t)$ by de-fragmentation method

No.	Time	Packet size	Source	Destination
1	0.0000	70	192.168.1.1	192.168.1.2
2	0.1000	5046	192.168.1.1	192.168.1.2
6	0.3000	65	192.168.1.1	192.168.1.2
7	0.5000	78	192.168.1.1	192.168.1.2
8	0.6050	1560	192.168.1.1	192.168.1.2
9	0.6150	78	192.168.1.3	192.168.1.2

Figure 6: Simple example of captured traffic transformed by the defragmentation method

These results are sorted in the histogram of $X_f(t)$. Similarly, as at the first algorithm, a histogram is obtained for $Y_f(t)$ and the identified pdf, as well as estimating their parameters using EasyFIT [23] tool (Fig. 7).

IV. Algorithm Validation

Validation of the developed method was performed using an OPNET Modeler simulation tool [14, 16, 17, 19, 20, 21]. An IP station was used for a simulation model (Fig. 1) with Pareto distribution (α = 1, k = 26) as the file size process and Weibull distribution (α = 0.5, k = 0.02) as the file inter-arrival time process for the file generation processes. This simulated network traffic represents the referenced traffic, on which the distribution parameters were estimated. With some modifications to the OPNET process model of the IP station (to perform packet logging functionality), the simulated packets' data of the modeled network traffic was captured, during the simulation run. The captured packet's data includes information about packet time stamps, and the sources and destinations of IP addresses. These data were sufficient for transformation using the de-fragmentation method. We calculate histograms and estimate distribution parameters on the transformed captured traffic. Any discrepancies between histograms and the chosen distribution are evaluated, using different goodness of fit tests, such as Kolmogorov-Smirnov, Anderson-Darling and Chi-Square [11-13].

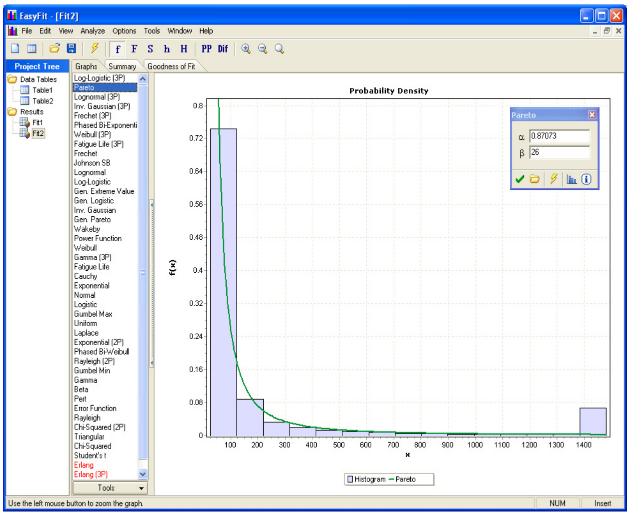

Figure 7: Histogram of a packet size process regarding captured packets, and suitable distribution with estimated parameters in EasyFit fitting tool.

V. Conclusion

This paper presents a comparison between two developed algorithms for the calculation of histograms regarding the processes which generate files, presenting a source of packet traffic. Continuous pdf's parameters, using some of the fitting tools, are later identified from these histograms. These parameters can, in-turn, be used as the parameters of a traffic generator in a simulation program.

The results of both algorithms have negligible statistical differences, so on this level the primacy has given way to a second algorithm, due to its simplicity and far shorter execution time. In those cases, where it is desirable to identify those applications that contribute to the packets' traffic, then the first algorithm is easier to expand, in such way that enables simultaneous statistical calculation of any application involved in packet traffic. The second algorithm has no such ability. It can calculate a statistic of any application only if a measurement tool, i.e. sniffer, from all the traffic filtered out only traffic originated in the selected application.

Acknowledgement

This work was partly financed by the Slovenian Ministry of Defense within the frame of the target research program "Science for Peace and Security": M2-0140 - Modeling of Command and Control information systems, and partly by the Slovenian Ministry for High Education and Science, research program P2-0065 "Telematics".

References

[1] W. E. Leland, M. S. Taqqu, W. Willinger in D. V. Wilson, “On the self-similar nature of Ethernet traffic (Extended version),” IEEE/ACM Transactions on Networking, Vol.2, pp.1-15, 1994.

[2] W. Willinger in V. Paxson, Where mathematics meets the Internet, Notices of the American Mathematical Society 45(8): 961–970, 1998.

[3] K. Park, G. Kim in M. E. Crovella, “On the Relationship Between File Sizes Transport Protocols, and Self-Similar Network Traffic,” International Conference on Network Protocols, 171–180, Oct 1996.

[4] M. E. Crovella in A. Bestavros, “Self-Similarity in World Wide Web Traffic Evidence and Possible Causes,” IEEE/ACM Transactions on Networking, 1997

[5] O. Sheluhin, S. Smolskiy and A. Osin, Self-Similar Processes in Telecommunications, John Wiley & Sons, 2007.

[6] K. Park in W. Willinger, Self-Similar Network Traffic and Performance Evaluation, John Wiley & Sons, 2000.

[7] T. Karagiannis, M. Molle in M. Faloutos, Understanding the limitations of estimation methods for long-range dependence, University of California.

[8] T. Karagiannis in M. Faloutos, Selfis: A tool for self-similarity and long range dependence analysis, University of California.

[9] M. Fras, J. Mohorko and Z. Cucej, Estimating the parameters of measured self similar traffic for modeling in OPNET, IWSSIP Conference, 27.-30 June 2007, Maribor, Slovenia.

[10] M. Fras, J. Mohorko and Z. Cucej, Packet size process modeling of measured self-similar network traffic with defragmentation method, IWSSIP Conference, 25.-28 June 2008, Bratislava, Slovakia.

[11] M. Chakravarti, R. G. Laha and J. Roy, Handbook of Methods of Applied Statistics, Volume I, John Wiley and Sons, pp. 392-394, 1967.

[12] W.T. Eadie, D. Drijard, F.E. James, M. Roos and B. Sadoulet, Statistical Methods in Experimental Physics, Amsterdam, North-Holland, 269-271, 1971.

[13] R. L. Plackett, Karl Pearson and the Chi-Squared Test, International Statistical Review 51 (1): 59-72, 1983.

[14] B. Vujičić, N. Cackov, S. Vujičić in L. Trajković, Modeling and Characterization of Traffic in Public Safety Wireless Networks, Simon Fraser University, Vancouver, Canada, SPECTS 2005

[15] B. Hill, “A Simple Approach to Inference About tbc Tail of a Distribution”, Annals of Statistics, Vol. 3, No. 5, 1975, pp.1163-1174

[16] J. Judge, H. W. Beadle and J. Chicharo, Sampling HTTP response packets for prediction of web traffic volume statistics, IEEE Global Communications Conference (GLOBECOM'98), Sydney, Australia, Nov. 8-12, 1998

[17] V. Paxon in S. Floyd, Wide area traffic: the failure of Poisson modeling, IEEE/ACM Transactions on Networking, 3(3): 226–244, 1995.

[18] H. Yõlmaz, IP over DVB: Managment of self-similarity, Master of Science, Boğaziçi University 2002

[19] B. Vujičić, Modeling and Characterization of Traffic in Public Safety Wireless Networks, Master of Applied science, Simon Fraser University, Vancouver, 2006.

[20] M. Jiang, S. Hardy in Lj. Trajkovic, Simulating CDPD networks using OPNET, OPNETWORK 2000, Washington D.C., August 2000.

[21] J. Mohorko, M Fras and Ž. Čučej, Modeling methods in OPNET simulations of tactical command and control information systems, IWSSIP Conference, 27.-30 June 2007, Maribor, Slovenia.

[22] Averill M. Law in Michael G. McComas, How the Expertfit distribution fitting software can make simulation models more valid, Proceedings of the 2001 Winter Simulation Conference.

[23] Free (demo) fitting tool EasyFit software [Online]. Available: www.mathwave.com/.

[24] M. Fras, Methods for the statistical modeling of measured network traffic for simulation purposes, Ph.D. thesis, University of Maribor, 2009, Maribor, Slovenia.

[25] M. Fras, J. Mohorko and Ž. Čučej, Modeling of captured network traffic by mimic defragmentation process, Simulation: Transactions of the Society for Modeling and Simulation International (in progress).

[26] RFC 793 - Transmission Control Protocol. Available: http://www.faqs.org/rfcs/rfc793.html