IEEE International Conference on Communications 2013: IEEE ICC'13 - 1st IEEE Workshop on Traffic Identification and Classification for Advanced Network Services and Scenarios (TRICANS)

How to Validate Traffic Generators?

Sándor Molnár[1], Péter Megyesi
High Speed Networks Lab., Dept. of Telecomm. and Mediainformatics,
Budapest Univ. of Technology and Economics, Budapest, Hungary
Email: {molnar, megyesi}@tmit.bme.hu

Géza Szabó
TrafficLab
Ericsson Research, Budapest, Hungary
Email: geza.szabo@ericsson.com

***Abstract*—Network traffic generators are widely used in networking research and they are validated by a very broad range of metrics (mainly traffic characteristics). In this paper we overview the state of the art of these metrics and unveil that there is no consensus in the research community how to validate these traffic generators and which metric to choose for validation purpose. This situation makes it extremely difficult to evaluate validation results and compare different traffic generators. We advocate the research for finding a common set of metrics for the validation and comparative evaluation of traffic generators.**

I. Introduction

Network traffic generators are vital in the design, development and management of our networks. Its importance became even more pronounced as the complexity of our networks is increased resulting in the use of simulation methodologies (e.g. ns2 or ns3) less accurate. On the other hand, the network data is the property of the operator and it results in a number of privacy issues limiting the use of the replay of measured traces. As a result a huge number of traffic generators have been developed in the last decades based on different methodologies and they were always adapted to the current need of network environments, application sets and purpose of use. See Table I and its reference list for an overview. The main function of these traffic generators is that these tools can inject packets into the network in a controlled fashion generating a synthetic traffic. The crucial requirement is that the characteristics of the synthetic traffic must capture the characteristics of actual traffic in the network. In spite of the fact that there is a long history of traffic generators and a large number of traffic generators have been proposed so far it seems that there is no consensus in the research community how to validate these traffic generators and which metric is used to evaluate the accuracy of the generator under investigation.

In this paper we address the issue of finding appropriate and common metric for the validation of traffic generators. We overview the most recent metrics researchers use for their traffic generators and categorize them. The main motivation of the paper is to unveil the current situation and show that there is no common metric being used in the state of the art of traffic generators literature and it makes the evaluation of the validation results and also the comparison of different traffic generators very difficult if not impossible. Therefore the motivation of finding a common set of metrics for this purpose is a key factor for categorize the recent and future traffic generators from the most important point of view: how accurately they can generate traffic which is reliable and can be used for the design, development and management of our networks and devices. We are raising an alert here, while the solution to the problem is not easy and deserves a deep study as a future work.

This paper is organized as follows. In Section II we present the state of the art traffic generation tools along with the validation techniques used in their introduction. Then, in Section III a categorization is given for the most frequent validation metrics. Finally, Section IV concludes the paper with a discussion on a possible set of metrics that could be the basis for establishing an agreed set of metrics by the research community for the future as the common validation measure for newly developed traffic generation tools.

II. Traffic Generators and Validation Techniques

We have investigated sufficient amount of traffic generators found in the recent literature and classified them into five categories according to the metrics used in validation perspective. Table I contains a brief overview about the presented traffic generators.

A. Replay Engines

Replay engines take previously captured traffic to send the packets out on the network interface the same timing that it was recorded. Given their purpose the only question that rises during their operation is whether the packets follow each other the same way as they were captured. This phenomenon could appear in both Inter Packet Timing (IPT) skewness (usually due to inaccurate software interrupts) and throughput saturation (due to bandwidth limitation).

The most common open-source replay application is TCPreplay [1] which can use *libpcap* files as input. It is also capable to rewrite Layer 2, 3 and 4 header information for various testing purposes. Since TCPreplay is a general, user-level software working on any UNIX platform it's performance may highly dependent on the installed environment. In [2] authors present TCPivo an open-source, high-speed packet replay engine on commodity hardware. This paper shows example of the IPT errors using different execution approaches.

As a solution for bandwidth limitation Ye et al. [3] presents a technique to replay a captured OC-48 trace on multiple commodity PCs with Gigabit Ethernet interface. The authors

[1]The research was supported by OTKA-KTIA grant CNK77802.

TABLE I: Validation Techniques for Traffic Generators

Generator Category	Traffic Generator	Description	Validation Techniques
Replay Engines	TCPreplay [1]	User-level application for replaying *libpcap* files	No validation since TCPreplay is a user-level software
	TCPivo [2]	High-speed kernel-level replay engine	Inter Packet Timing error
	Divide and Conquer [3]	Replay technique for OC-48 traces using multiple Gigabit Ethernet PCs	Wavelet scaling analysis and IPT error
Maximum Throughput Generators	Iperf [4]	User-level application for bandwidth, packet loss ratio and jitter testing	No validation since Iperf is a user-level software
	BRUTE [5]	Kernel-level packet generator	Throughput deviation compared to tuned values
	BRUNO [6]	Hardware implemented packet generator	Throughput deviation and IPT error
	KUTE [7]	Kernel-level packet generator	Throughput and IPT accuracy
	Ostinato [8]	User-level packet generator with friendly GUI	No validation since Ostinato is a user-level software
Model Based Generators	TG [9]	Packet-level generator supporting various distributions for IPT and PS values	IPT and PS values compared to tuned parameters presented in [10]
	MGEN [11]	Packet-level generator supporting various distributions for IPT and PS values	IPT and PS values compared to tuned parameters presented in [10]
High-Level and Auto-Configurable Generators	HARPOON [12]	Flow-based traffic generator that can mimic netflow based measurements	Comparison of original and synthetic traffic by throughput, byte, packet and flow volumes, PS distribution and wavelet scaling
	SWING [13]	Closed-loop, network responsive traffic generator which is able to extracts distributions for user, application, and network behavior of real measurements	Comparison of original and synthetic traffic by quantitative statistics values, wavelet scaling of different applications and distribution of various QoS metrics
	TMIX [14]	Traffic emulator for ns-2 based on source-level characterization of TCP connections	Comparison of original and synthetic traffic by throughput, flow size, RTT and application data unit distributions and wavelet scaling.
	LiTGen [15]	Open-loop, packet-level traffic generator based on realistic IP traffic modeling	Comparison of original and synthetic traffic by QoS parameters based on queuing models and wavelet scaling
	D-ITG [16]	Extensive workload generation framework that can produce traffic for wide range of network scenarios	Comparison of original and synthetic traffic by throughput and distributions of IPT and PS values
Special Scenario Generators	EAR [17]	Traffic replay technique for mimic IEEE 802.11 protocol behavior	Unique metric for measuring wireless traffic replay called *Event Reproduction Ratio*
	ParaSynTG [18]	Web traffic generator	Web specific metrics such as document size and popularity distributions
	YouTube Workload Generator [19]	Workload generation methods for mimic YouTube video traffic	Online video specific metrics like video length, size and rating distributions or cache performance
	Graph-Based Traffic Generator [20]	Flow trace generator based on *Traffic Dispersion Graphs* templates	Graph related metrics such as distribution of degrees or connected edges and verticals

validate their methodology by presenting wavelet based analysis [21] of the original and the replayed trace. Even though explanation for visual differences in various time scales are given in the paper, there is no qualitative metric presented to evaluate how close the corresponding curves are.

B. Maximum Throughput Generators

Maximum throughput generators are usually used to test end-to-end network performances. Although the application of these tools differs from the previous category, validation techniques also use throughput and IPT values.

Iperf [4] is widely used in network engineering for testing bandwidth, delay jitter and loss ratio characteristics since it's available on various platforms. Like TCPreplay, given it's generality the tool's accuracy may vary in different conditions.

BRUTE [5] is a packet-level traffic generator working in Linux kernel-level thus it guarantees more controllable behavior. The authors evaluate their tool by presenting throughput measurements in various circumstances.

BRUNO [6] is an extension of the same methodology to a specific hardware platform (Intel IXP2400). This solution is evaluated in the paper by showing that this implementation provides more precise values in both throughput and IPT level.

KUTE [7] is another Linux kernel-level packet generator tool. The tool can be set up for any given packet sanding rate then it calculates the corresponding IPT value and it sends out packet to the network interface using active sleep between consecutive packets. In KUTE's performance evaluation authors show that both packet rate and IPT properties approximates the expected value better the other traffic generator tools.

Ostinato [8] is a very recent user-level traffic generator tool available for many platforms. Users can define various traffic streams via a friendly GUI and easily transmit them to the network interface. However, we did not find any reference in the scientific literature that would validate its performance.

C. Model Based Generators

Model based traffic generators use different stochastic models for creating packet-level traces. This procedure raises the question whether the generated traffic follows the same statistics being set up by the stochastic model. This issue was addressed in [10] where authors showed that the output statistics of model based traffic generators (like [9] and [11]) do not follow what analytical result would implicate. Thus these generators should be tested for various statistics that the stochastic model would implicate (including IPT, packet size distribution and correlation).

D. High-Level and Auto-configurable Generators

These kind of traffic generators are based on higher-level model of network traffic and they are also able to automatically configure their parameters based on live measurements therefore creating an output which is statistically similar to the original traffic. From our point of view, papers promoting these generators are the most relevant examples as they all contain comparison of real and synthetic traffic traces using wide range of metrics.

HARPOON [12] is a traffic generator which is able to produce synthetic traffic based on various flow characteristics. Moreover, the tool can analyze real measurements to extract such values thus it can create artificial traffic with characteristics that are close to the original live measurement. For validation of their tool authors use metrics of byte throughput, interconnection time distribution, file size distribution, IP frequency distribution, and byte, packet and flow volume distributions. Although the presented figures show close match qualitatively, there is no quantitative comparison of the used metrics. In the limitation section authors state that other metrics such as scaling characteristics, queue length distribution for the first-hop router, packet loss process or flow durations may not match with the original due to architecture of the traffic generator.

SWING [13] is another high-level traffic generator which is able to generate traffic based on characteristics of a real trace. Validating the aggregate traffic characteristics the authors use quantitative comparison of average, median and inter-quartile range values of base statistical attributes. For comparing the generator's application and user model the paper presents the distributions of two-way link delay for hosts, loss rates for feeding links and upstream/downstream capacities. Determining whether the generated trace catches the burstiness characteristics of the original trace the author present wavelet-based scaling analysis [21]. Although the curves in the energy plots are fairly close to each other there is no quantitative comparison between them, the authors satisfied by the visual proximity.

In [14] authors presents a procedure to convert a TCP flows to *connection vectors* and a traffic generator called TMIX which is able to produce synthetic traffic based on these vectors. Firstly, the paper validates the model behind the generator by comparing the Application Data Unit (ADU) distribution of two live measurements to the corresponding synthetic. Secondly, the output of TMIX is validated by the following metrics: throughput, Round Trip Time (RTT) and flow size distributions, time series of active connections and scaling properties [21]. However, there is no quantitative comparison between the presented statistics, authors only state that there are good match between the values.

LiTGen [15] is also generator which can reproduce aggregated application (web, mail and P2P) traffic based on real measurement by extracting parameters such as session and object characteristics. Authors present the necessity of setting correlating between the number of packets and their inter-arrival time in an object for catching traffic burstiness in a more precise way. LiTGen is validated by two kind of metrics: wavelet based analysis for scaling behavior of the packet arrival process and queuing model fitting for performance characteristics. Although the presented plots show close visual match between the original and synthetic traces there is no further interpretation given.

In [16] Botta et al. presents a comprehensive study on the requirements of a suitable network workload generator. In addition, authors present the functions of the D-ITG traffic generator which is able to satisfy the conditions given above. The synthetic packet generation mode of the tool uses Hidden Markov Model approach for modeling the Inter Packet Time (IPT) and Packet Size (PS) sequence. Thus, in the paper the distribution of the IPT and PS values along with the throughput are compared between multiple real measurements and the corresponding generated trace. As in the previous examples, the validation of D-ITG lacks of quantitative comparison of the presented metrics.

E. Special Scenario Generators

These traffic generators usually represents specific type of network conditions thus they often offer unique metric techniques for the given scenario. For example, EAR [17] describes a method for transferring a packet-level capture into a sequence of events that follows the IEEE 802.11 protocol. Authors propose a quantitative metric, so called the *event reproduction ratio* to evaluate their methodology. This metrics is specific for WLAN environment thus it can not be used in any other network scenario for metric purposes.

Similar cases can be found in [18] and [19] where generation methods are presented for only WWW and YouTube traffic, respectively. In [18] authors validate their tool, called LiTGen by web specific metrics such as request frequency or document size distributions. In case of YouTube traffic in [19] authors focus on proxy cache based characteristics.

Another unique example is presented in [20] where a method is described for transforming network protocols into so called *Traffic Dispersion Graphs*. Authors give a technique for generation traffic by flow graph templates. Then the actual evaluation process is based on graph based metrics like degree distribution, connected edges and verticals.

TABLE II: Metrics Used for Validating Traffic Generators

Metric Category	Metric	Used in Validation of Traffic Generator
Packet-Level Metrics	Byte Throughput	BRUTE, BRUNO, KUTE, HARPOON, TMIX, D-ITG
	Packet Size Distribution	HARPOON, D-ITG
	Inter Packet Time Distribution	TCPreplay, Divide and Conquer, BRUNO, KUTE, D-ITG
Flow-Level Metrics	Flow Size Distribution	HARPOON, TMIX
	Flow Volume	HARPOON
Scaling Characteristics	Logscale diagram	Divide and Conquer, HARPOON, SWING, TMIX, LiTGen
	Multiscale diagram	-
QoS/QoE Related Metrics	Queueing Behavior	LiTGen
	Round Trip Time Distribution	HARPOON, TMIX

III. Metric Categories

In this section we present a categorization of the most widely used network traffic metrics found in the literature of traffic generators along with the importance and virtue of their usage. Table II contains a brief overview of this section.

A. Packet Based Metrics

Packet based metrics are the most basic representations of traffic streams thus they are widely used in the validation process of traffic generators. The most common gauge is the time series of traffic amount which we refer as throughput. Throughput can be measured in both byte and packet basis and difference is between the two methods can be expressed by Packet Sizes (PS). Due to the complex structure of network traffic, throughput usually presented in several time scales (minutes to hours). For instance in [12] authors show that HARPOON can follow the diurnal nature of real traffic. As Table II shows byte throughput is the most frequent metric used during the validation process of traffic generators.

The other basic, packet-level attribute of network traffic are time difference between successive packets, commonly referred as Inter Packet Times[2] (IPT). IPT values can reveal more about the short term behavior of a traffic stream than throughput, though a simple empirical distribution is unable to unveil second order characteristics like burstiness or correlation (such methods are presented in Subsection III-C). However, calculating the distribution of IPT is considered as a basic validation technique and used in many cases in the literature (see Table II for examples).

Although there are other packet-level metrics we consider the following three methods essential in validating a traffic generator: (1) byte throughput, (2) packet size distribution, (3) inter packet time distribution.

B. Flow Based Metrics

Flow based metrics are higher level characteristics about a traffic stream since they consider Layer 3 and 4 information. Today many network elements are capable of flow based operation, for instance open-flow devices or traffic classification tools. Even though many traffic generators are unable to represent realistic flow statistics, a high-level emulator (those presented in Subsection II-D) should be validated by such aspects.

As on operational point of view we consider two metrics relevant in this category. The volume of flows are in direct correlation with the number of instances a flow based device should run simultaneously, while the sizes of flows determine the length of a given instance. Such metrics are only accounted in the validation of HARPOON and TMIX.

C. Scaling Characteristics

Second order characteristics are responsible for the complex nature of network traffic like burstiness or long-range dependence. Due to the non-stationary property of live traffic classic second order metrics (for instance auto-correlation function or indexes of dispersion) are hardly useable for this purpose. However, over the past decades wavelet based analysis became an efficient way of unveiling the inherent correlation, burst and scaling structure of network traffic even in the presence of non-stationary effects [21]. As confirmation, we have found many examples in the literature where authors used this technique to validate their traffic generator tool (see Table II).

Recent studies showed that in some cases simple mono-fractal methods (e.g. logscale diagram) are unable to capture the scaling characteristics of network traffic so the use of multi-fractal models is needed (e.g. multiscale diagram) [21]. To investigate this phenomenon of live traffic multi-scale analysis was proposed [21]. However, we found that the literature lacks using muli-scaling analysis during the validation of traffic generators.

D. QoS/QoE Related Metrics

Quality of Service (QoS) or Quality of Experience (QoE) are important quality metrics often used and investigated. There is a need that a realistic workload generation should result in similar QoS/QoE based characteristics as live measurement results. An extensive study about QoS metrics can be found in [22]. However, as a traffic generator point of view end-to-end metrics such as one-way-delay, one-way-delay variation or route can not be measured. Thus we found two relevant metrics which appeared in validation methodologies. RTT values were presented in both HARPOON and TMIX, and queuing model fitting was presented in LiTGen for estimation of average waiting time and queue size parameters. These

[2] In the literature Inter Departure Time (IDT) and Inter Arrival Time (IAT) are also common denominations depending on the prospective of sender or receiver side, respectively.

metrics show an important practical view of the generated traffic therefore we consider them very relevant in a validation process. However, as the recent literature shows this aspect is seldom used and applied for validation goals.

IV. Discussion and Conclusion

In this paper we highlighted that there is no consensus in the research community how to validate traffic generators. A wide range of metrics are used in the state of the art solutions making it nearly impossible to compare different results and generators. We also found that validation processes lack of any kind of quantitative comparisons of different statistics, usually only the statement of visual proximity of different curves is presented. Moreover, we also found that some relevant aspects of traffic characteristics (e.g. multi-scaling properties) are completely excluded from the evaluation and validation of recent traffic generators. To overcome this problem we advocate research to find a common set of metrics that can be used for comparison and validation purposes of traffic generators and emulators. As a starting point we presented in Table II those metrics that are most frequently used and contain the most relevant information for validation. We think that a common set of these metrics or metrics derived from these characteristics could be good candidates for comparisons.

We also note that traffic generation is not a final goal but rather a tool to use for device testings (e.g., DPI), designing network devices (e.g., router memory) or performance evaluation and management of networks.

For example, DPI engines can be tested in terms of recognition accuracy, completeness and performance as well. To make it possible to test the completeness of recognition a generated traffic has to contain a wide range of protocols. This is the most difficult to provide as it means that plenty of applications should be incorporated into the test trace. To test DPI accuracy plenty of traffic of a certain protocol has to be fed to the DPI system to check the traffic coverage of the DPI signature. Moreover, the false positive ratio has to be checked if the DPI signature of a certain application is tested against other applications. During performance testing the DPI system needs a high data load, with high number of users, flows and various packet sizes.

Taking another example of designing router memory the queueing-related performance metrics are the most relevant characteristics that must be captured. Different traffic characteristics have different effect on such queueing behavior. In recent network measurements the complex scaling nature with strong correlation of the traffic including fractal characteristics can be dominant. These characteristics on the time-scale relevant for packet buffering in router memories should be studied to find a good metric for this purpose. In addition, the effect of Active Queue Management (AQM) methods with the interaction of different protocols are also among the factors that have significant effect on such metrics.

Therefore, the final goal must always determine which metrics are really important and relevant for a given purpose. This aspect must be also deeply investigated and included in the common set of metrics which can be used by the whole research community as an agreed and accepted metric set.

Our future research will address the comprehensive evaluation of these metrics from the above point of views.

References

[1] "tcpreplay," retrieved: Jan, 2013. [Online]. Available: http://tcpreplay.synfin.net/

[2] W.-c. Feng, A. Goel, A. Bezzaz, W.-c. Feng, and J. Walpole, "Tcpivo: a high-performance packet replay engine," in *Proceedings of the ACM SIGCOMM workshop on Models, methods and tools for reproducible network research*, ser. MoMeTools '03, 2003, pp. 57–64.

[3] T. Ye, D. Veitch, G. Iannaccone, and S. Bhattacharya, "Divide and conquer: Pc-based packet trace replay at oc-48 speeds," in *Testbeds and Research Infrastructures for the Development of Networks and Communities, 2005. Tridentcom 2005. First International Conference on*, feb. 2005, pp. 262 – 271.

[4] "Iperf," retrieved: Jan, 2013. [Online]. Available: http://sourceforge.net/projects/iperf/

[5] N. Secchi, R. Bonelli, S. Giordano, and G. Procissi, "Brute: A high performance and extensible traffic generator," in *Proc. SPECTS '05*, Philadelphia, PA, USA, July 2005, pp. 839–845.

[6] G. Antichi, A. Di Pietro, D. Ficara, S. Giordano, G. Procissi, and F. Vitucci, "Bruno: A high performance traffic generator for network processor," in *Proc. SPECTS '08*, Edinburgh, UK, June 2008, pp. 526–533.

[7] G. A. Sebastian Zander, David Kennedy, "KUTE - A High Performance Kernel-based UDP Traffic Engine," CAIDA, Tech. Rep., 2005. [Online]. Available: http://caia.swin.edu.au/reports/050118A/CAIA-TR-050118A.pdf

[8] "Ostinato," retrieved: Jan, 2013. [Online]. Available: http://ostinato.org/

[9] "TG," retrieved: Jan, 2013. [Online]. Available: http://www.postel.org/tg/tg.html

[10] A. Botta, A. Dainotti, and A. Pescape, "Do you trust your software-based traffic generator?" *Communications Magazine, IEEE*, vol. 48, no. 9, pp. 158 –165, sept. 2010.

[11] "MGEN," retrieved: Jan, 2013. [Online]. Available: http://cs.itd.nrl.navy.mil/work/mgen/

[12] J. Sommers and P. Barford, "Self-configuring network traffic generation," in *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, ser. IMC '04, 2004, pp. 68–81.

[13] K. Vishwanath and A. Vahdat, "Swing: Realistic and responsive network traffic generation," *Networking, IEEE/ACM Transactions on*, vol. 17, no. 3, pp. 712 –725, June 2009.

[14] M. C. Weigle, P. Adurthi, F. Hernández-Campos, K. Jeffay, and F. D. Smith, "Tmix: a tool for generating realistic tcp application workloads in ns-2," *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 3, pp. 65–76, Jul. 2006.

[15] C. Rolland, J. Ridoux, B. Baynat, and V. Borrel, "Using litgen, a realistic ip traffic model, to evaluate the impact of burstiness on performance," in *Proc. Simutools '08*, Marseille, France, March 2008.

[16] A. Botta, A. Dainotti, and A. Pescap, "A tool for the generation of realistic network workload for emerging networking scenarios," *Computer Networks*, vol. 56, no. 15, pp. 3531 – 3547, 2012.

[17] C.-Y. Ku, Y.-D. Lin, Y.-C. Lai, P.-H. Li, and K.-J. Lin, "Real traffic replay over wlan with environment emulation," in *Wireless Communications and Networking Conference (WCNC), 2012*, April 2012, pp. 2406 –2411.

[18] R. Khayari, M. Rucker, A. Lehmann, and A. Musovic, "Parasyntg: A parameterized synthetic trace generator for representation of www traffic," in *Proc. SPECTS '08*, Edinburgh, UK, June 2008, pp. 317–323.

[19] A. Abhari and M. Soraya, "Workload generation for youtube," *Multimedia Tools Appl.*, vol. 46, no. 1, pp. 91–118, Jan. 2010.

[20] P. Siska, M. P. Stoecklin, A. Kind, and T. Braun, "A flow trace generator using graph-based traffic classification techniques," in *Proc. IWCMC '10*, Caen, France, June 2010, pp. 457–462.

[21] P. Abry, P. Flandrin, M. S. Taqqu, and D. Veitch, *Wavelets for the Analysis, Estimation, and Synthesis of Scaling Data*. John Wiley & Sons, Inc., 2002, pp. 39–88.

[22] F. Michaut and F. Lepage, "Application-oriented network metrology: metrics and active measurement tools," *Communications Surveys Tutorials, IEEE*, vol. 7, no. 2, pp. 2–24, 2005.