

Experiment-Driven Modeling of Open-Source Internet Traffic Generators

Leopoldo Angrisani, *Senior Member, IEEE*, Alessio Botta, *Member, IEEE*, Gianfranco Miele, *Member, IEEE*, Antonio Pescapé, *Senior Member, IEEE*, and Michele Vadursi, *Senior Member, IEEE*

Abstract—This paper is a first step toward the goal of evaluating the measurement accuracy of the interdeparture times (IDTs) provided by software traffic generators. To provide evidences of the proposed methodology, in this paper, we focus our attention on the experimental characterization of the internal generation cycle of a well-known, open source generator, namely, distributed internet traffic generator. We tested the traffic generator for different systems, and under the best possible conditions, i.e., with the minimum system load. The resulting performance may be seen as the ideal limit the generators can tend to. In this paper, we present the rationale for the activity, the underlying methodology and a set of tests aimed at highlighting the relevance of the clock resolution in the accuracy of IDT. In addition, we present a statistical model for the IDT of packets. The model allows to capture the effects of the various phenomena that affect this important generation parameter. We have also performed a set of experiments with another traffic generator to show the generalizability of the results. These results show that the contributions proposed in this paper are general and valid for a wide set of software-based traffic generators.

Index Terms—Interdeparture time (IDT), traffic generator, uncertainty.

I. INTRODUCTION

NETWORK traffic generators are widely employed in computer network by both university and industry for performance testing, analysis, and so on. They are able to inject packets following a particular traffic pattern into a network, to test the performance, to investigate if it can reliably support a particular application in controlled environments, and to investigate other important aspects. These goals can only be reached if the traffic generator is able to accurately replicate the traffic generated by real network applications

Manuscript received December 9, 2013; revised August 3, 2014; accepted August 4, 2014. This work was supported by the Ministry of Education, Universities and Research through PLATINO Project under Grant PON01 01007, SMART HEALTH Project under Grant PON04a2 C, and SIRIO Project under Grant PON01 02425. The Associate Editor coordinating the review process was Dr. Dario Petri.

L. Angrisani, A. Botta, and A. Pescapé are with the Department of Electrical Engineering and Information Technology, University of Naples Federico II, Naples 80125, Italy (e-mail: angrisan@unina.it; botta@unina.it; pescape@unina.it).

G. Miele is with the Department of Electrical and Information Engineering, University of Cassino and Southern Lazio, Cassino 03043, Italy (e-mail: g.miele@unicas.it).

M. Vadursi is with the Department of Engineering, University of Naples Parthenope, Naples 80143, Italy (e-mail: vadursi@uniparthenope.it).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TIM.2014.2348633

or, in more general terms, to generate traffic with specific statistical characteristics [1].

As a consequence, results obtained by tests that involve network traffic generators are strictly dependent on the ability of the generators to accurately emulate or replicate the desired traffic shape or statistical pattern [2], [3].

Traffic generators are implemented over both hardware and software platforms. The former are especially designed by instrument manufacturers and implemented on dedicated high-performance hardware. As a consequence, they are typically more precise and reach very high performance, but they are expensive. In addition, those solutions are preconfigured to carry out a certain type of tests, and they are difficult to customize, being tied to an hardware implementation. On the contrary, software traffic generators are cheaper, often open-source or free of charge and more flexible, but it is expected that they have lower performance in terms of accuracy and precision [4], [5].

In spite of these characteristics that seem to endorse hardware-based traffic generators, the use of software-based traffic generator is very widespread in networking research and in network performance testing. There are several reasons that justify this choice and most of them are strictly connected to their flexibility. As an example, they can be easily installed on several nodes, to emulate a network with distributed traffic sources, or they can be updated for specific purposes adding, for example, new traffic patterns.

Certified information about the imposed values of the characteristics of the traffic generated by software-based traffic generator, such as bit rate, interdeparture time (IDT), packet rate, and so on, is extremely needed. It should be provided as the manufacturers of hardware-based traffic generators already do with their products.

Unfortunately, certifying this information is very difficult because the metrological properties (i.e., accuracy of the traffic generation process) depend on the commercial off-the-shelf hardware used, the operating system (OS) adopted, and the status of the host used for traffic generation [6]. Therefore, without that information, the reference is uncertain and consequently obtained results could be useless. In the computer science community this problem is known and investigated by considering several approaches [6]–[10], but it is not fully analyzed according to the guide for the expression of uncertainty in measurements (GUM) [11]. At the same time, the interest of part of the instrumentation and measurement community has been recently attracted by performance

and accuracy issues related to networks and distributed computing [12]–[17].

Stemming from the previous experiences of the authors on accuracy evaluation of software-based IP traffic generators [6], [18]–[20], in this paper, we analyze the factors that influence the IDT accuracy of software-based traffic generators and characterize them from a metrological point of view, as the basis for the evaluation of their instrumental uncertainty. We focus on IDT because it is the most sensitive to poor accuracy. Previous papers existing in the literature have already characterized some generators, in terms of average and experimental standard deviation of traffic rate (or of IDT) [5]–[9]. The step ahead in this paper is a model of IDT, based on the analysis of the empirical distribution of observed IDTs. In particular, we are interested in matching the modifications observed in the distribution of IDTs with the way the IDT-adaptation mechanism operates. To this aim, a well-known software-based traffic generator, distributed internet traffic generator (D-ITG) [4] has been considered for the experiments and for providing a use case for the application of the proposed methodology. Using D-ITG, we analyze the interfering factors that affect the IDT by performing a set of experiments in a controlled scenario. Even if we perform this paper using D-ITG, the methodology, the results obtained, and the model proposed are general and can be applied to a large set of software traffic generators [21]–[24]. Experiments with another traffic generator have also been performed, showing the generalizability of the results obtained.

In Section II, brief notes on D-ITG are reported. The methodology adopted for the characterization is described in Section III along with the presentation of experimental results. Finally, conclusions are drawn in Section IV.

II. BRIEF NOTES ON D-ITG

The D-ITG [4] is a well-known tool that is able to generate IPv4 and IPv6 traffic, as well as traffic at network, transport, and application layer. The D-ITG uses stochastic processes to emulate the IDT and packet size (PS) of real applications, supporting several statistical distributions for IDT and PS random variables (exponential, uniform, cauchy, normal, pareto, etc.). This approach is actually followed by a large set of traffic generators [6]. Among the two random variables, the IDT is the most sensitive to poor accuracy, being tightly dependent on the way the host running the traffic generator manages the time (process scheduling, time function resolution, etc.). A simplified version of the generation loop of D-ITG (repeated for every packet generated) is shown in Fig. 1. The figure evidences the main operations performed for the generation of the packet with the names inside the left-side boxes referring to the operation performed and the names in the right-side boxes referring to the kind of such operations (i.e., system call, memory access, etc.). The boxes in light blue are related to optional activities. It is worth underlining that the (mnemonic) name of the operation is not necessarily the name of the system call possibly used. The actual system call depends on the OS used, and D-ITG runs on different OSs. We used the Linux version for the tests as reported in Section III-A,

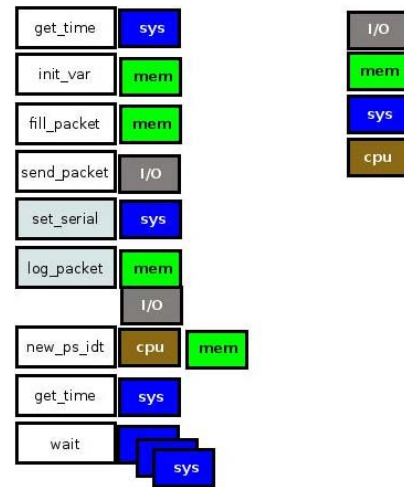


Fig. 1. D-ITG simplified generation cycle.

in which more details on how these operations are performed on this OS are also reported. As shown, the generation loop contains different memory accesses (mem), system calls (sys), computations (cpu), and I/O requests. In particular, for every packet, D-ITG:

- 1) fetches the current time from the OS;
- 2) initializes some variables;
- 3) fills the packet payload with the timestamp that was taken at the first step and with other information;
- 4) pushes the packet into the outgoing socket buffer;
- 5) raises a signal on the serial port, if required (this is used in case there is necessity to synchronize external equipments, such as packet or signal analyzers, etc.);
- 6) stores the log information for this packet, if required (this information is actually buffered in RAM for a number of packets, and dumped on disk periodically);
- 7) draws the new IDT and PS using a random number generator;
- 8) fetches again the current time from the OS to know how long it has passed since the beginning of the loop, and computes the time to be actually waited (here, the IDT recovery mechanism possibly takes place, as explained below);
- 9) waits for the remaining time before sending the new packet.

These operations are necessary for packet generation, and this generation loop is very similar to one of other packet-level traffic generators.

The comparison between the expected IDT and the elapsed time is also used by the generator for a mechanism called IDT recovery. This mechanism is used in case the generation of the current packet is in late with respect to the expected time. In case the sending of packet happens in late, the following packets are sent in advance with respect to the expected time to recover such delay. This is implemented using a variable that accounts for the delays accumulated during the generation loops, which is considered, when deciding if and how much to wait for the generation of the next packet. Therefore, the IDT are continuously compensated until the delay accumulated is not recovered. This mechanism can cause a number of

packets to be sent back-to-back in case the generation loops accumulate some delay. This has a bad effect on the accuracy of the IDT, but it allows to maintain the average packet rate requested by the user.

III. EXPERIMENTAL RESULTS

A. Characterization Methodology

Considering there are several sources of uncertainty in the internal generation loop of D-ITG, which are connected to a number of internal (i.e., strictly depending on the process of packet generation) and external (i.e., connected with the other ongoing processes managed by the OS) operating conditions, we decided to start by characterizing each operation, or small group of operations. The goal of the experimental activity is twofold: 1) understanding the contribution of each operation in the generation loop shown in Fig. 1 in the accumulation of delays that constitute the IDT, along with its variability and 2) paving the way to the expression of the uncertainty associated to IDTs in the generation loop of D-ITG.

The time needed for the completion of operations, such as memory accesses, computations, and I/O requests is basically deterministic and therefore has little impact on the uncertainty of IDT, except for a possible systematic effect. The most critical issues, in terms of variability and predictability, are OS system calls. This is the reason why the initial experiments have been focused on them. In particular, in Fig. 1 three operations requiring system calls can be singled out: 1) the `get_time` (using the actual `gettimeofday()` on Linux OS); 2) the `set_serial()`; and 3) the `wait()` (using a `select()` on Linux OSs). The `set_serial()` is used to emit a signal on the serial ports, which can be used to synchronize external hardware, such as signal or traffic analyzer. This operation is optional and not used for our tests to reduce the complexity of the analysis. Therefore, the attention has been focused on the remaining two, with specific attention to their implementation in Linux OS, being the OS most used by the users of these tools. As the analysis methodology is general, results may be easily generalized for other OSs.

Theoretically speaking, one could be interested to evaluate the variability of the completion time of each of the two functions. However, from an operational point of view, it is difficult to characterize the functions separately. We need to timestamp packets as they go through the steps of the generation cycle, but timestamping implies further involving the OS with a new call to the `gettimeofday()`, which brings on additional time contribution (and uncertainty) to the process. In other words, to characterize the `select()` function, we need to execute a `select()`, and a `gettimeofday()`, anyway.

The approach consists in the iterated execution of the functions for a time of 120 s, and is similar to the approach followed in [7]. The first-order difference of the timestamping results is then calculated, to achieve a vector containing the execution time of each iteration. Please note that we are making differential time measurements on the same host, so the uncertainty on IDT measurements due to the clock instability is negligible compared with the uncertainty due to OS calls.

Finally, the timestamping results of the series of the sole `gettimeofday()` and `select()` functions have been compared with those obtained when the full generation loop of D-ITG is executed. The goal is to experimentally verify, which is the relative weight of the OS function calls in the generation loop. To achieve this, the loop in Fig. 1 (with the exclusion of the optional operations) has been executed to generate constant bitrate (CBR) traffic¹ stream with a packet rate that is consistent with the waiting time given as input to the `select()` function in the previous tests. The IDT have been evaluated as the difference of two successive `gettimeofday()` results. In particular, when referring to D-ITG, our results are based on the timestamps collected by such tool during each cycle within the `get_time` block in Fig. 1. Such timestamps are dumped in a file after the generation by D-ITG. We analyzed this file to obtain the timestamps of the packets sent.

B. Experimental Results

The tests have been repeated for different imposed values of IDT, from 0.2 ms to 0.1 s. In addition, they have been performed on three different hosts, with the following characteristics.

- 1) CPU 4-core Intel(R) Xeon(R) CPU E5-1620 @ 3.60 GHz.
- 2) 8-GB RAM.
- 3) 1-TB SATA HD.
- 4) Network card: Intel 82574L gigabit network.
- 5) OS: Ubuntu 12.04.2 LTS (GNU/Linux 3.2.0-38-generic-pae i686).

Experiments were conducted in situations, where the sending and receiving processes were far from the saturation of the resources (i.e., all the cores were at least 50% idle). Generated packets are average sized (carrying 512 bytes of transport-layer payload each). In general, however, serialization and propagation delays do not have a big impact on our characterization, being the measurements taken at the source host. In detail, we generated packets from and to the same host, so that packets were not traversing any network interface. In addition, we used the timestamps reported in the log file of the sending process. The IDT of packets have then been calculated as the difference between consecutive timestamps (i.e., timestamps related to consecutive packets). In summary, all the time samples are referred to the same clock, and we take the difference between these timestamps. In this way, we overcome issues related to clock synchronization. In addition, three different runs were done for each experiment.

It is worth noting that the hosts as well as D-ITG have been configured to minimize the CPU resources consumption, i.e., with graphical interface inactivated, single user, local traffic generation (i.e., destination host = source host, and traffic flowing through the loopback interface, configured with an MTU of 16436 bytes) and generation of average sized packets (i.e., 512 bytes). In addition, the hosts, their OSs and

¹We performed the experiments with CBR to reduce the causes of variability of the IDT. Using random PSs introduces another possible cause of interference, which we wanted to avoid in this current stage. We left studies on the effect of variable bitrate traffic as a future work.

TABLE I

EXPERIMENTAL RESULTS OF THE ACCURACY TEST PERFORMED ON THE SIMPLIFIED GENERATION LOOP COMPOSED BY THE OS CALLS SELECT() AND GETTIMEOFDAY()

IDT [ms]	μ_{IDT} [ms]	Δ_{IDT} [%]	σ_{IDT} [ms]	$\sigma_{IDT_{rel}}$ [%]	max [ms]	min [ms]
100	100.125	0.13	0.017	0.01	100.277	100.047
10	10.066	0.66	0.015	0.15	10.666	10.006
4	4.061	1.52	0.012	0.29	4.188	4.003
2	2.061	3.03	0.013	0.6	2.873	2.003
1	1.059	5.94	0.014	1.4	3.403	1.003
0.5	0.5631	12.61	0.0082	1.5	2.807	0.504
0.3	0.360	19.86	0.010	2.8	3.071	0.304
0.2	0.2596	29.78	0.0059	2.3	2.307	0.207

TABLE II

EXPERIMENTAL RESULTS OF THE ACCURACY TEST PERFORMED ON THE CONSIDERED SOFTWARE TRAFFIC GENERATOR

IDT [ms]	μ_{IDT} [ms]	Δ_{IDT} [%]	σ_{IDT} [ms]	$\sigma_{IDT_{rel}}$ [%]	max [ms]	min [ms]
100	100.000	0.0001	0.022	0.022	100.128	99.897
10	10.0000	0.00006	0.0072	0.072	10.068	9.946
4	4.0000	0.00005	0.0044	0.11	4.066	3.968
2	2.000	0.00005	0.033	1.67	7.297	0.005
1	1.0000	0.00006	0.0024	0.24	1.068	0.952
0.5	0.50	0.00005	0.13	26.9	42.429	0.003
0.30	0.30	0.00005	0.49	164	241.925	0.003
0.2	0.20	0.00005	0.59	295	292.639	0.003
0.1	0.10	0.00005	0.51	511	374.123	0.003
0.05	0.05	0.00001	0.69	1377	633.646	0.002
0.02	0.02	0.000008	0.51	2547	711.248	0.002
0.01	0.01	0.00003	0.65	6487	757.708	0.002

the generation tools are run in typical conditions, such as no special hardware, no kernel modifications, tools running in user-space, with no particular privileges or priority, and so on. This is because we wanted to experiment the same operating conditions of typical users.

Tables I and II account for the experimental results of accuracy tests performed on the simplified generation loop, and the software traffic generator, respectively. Results are expressed in terms of average and experimental standard deviation of IDT, as well as relative difference between imposed and measured average IDT, namely Δ_{IDT} .

The experimental results show that for any packet rate, the measured IDT is greater than expected, for both experiments involving the simplified generation loop and those involving the traffic generator. The difference Δ_{IDT} becomes relatively high for packet rates higher than 1000 pkt/s, and comparable values of Δ_{IDT} have been experienced for the same imposed packet rate.

The experimental standard deviation is of the same order of magnitude for packet rates up to 1000 pkt/s. On the contrary, for higher packet rates, the variability of the IDTs observed for the software traffic generator becomes much higher, as demonstrated by the experimental standard deviation, which is one or even two orders of magnitude higher than that exhibited by the IDT of the simplified generation loop. It seems that the set of operations performed by the traffic generator starts to

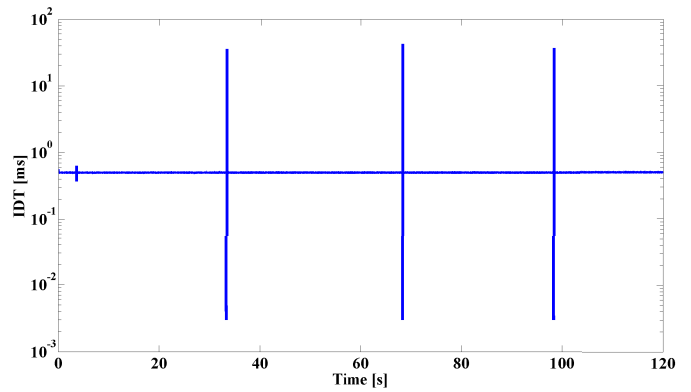


Fig. 2. Measured IDT versus time (nominal IDT = 0.5 ms).

have a non-negligible role in the reduction of repeatability of IDTs when the imposed IDT goes under 1 ms.

A closer look to the results summarized in Table II permits to formulate some hypotheses on the origin reasons behind such a relevant increase in the relative standard deviation. By plotting the experienced IDT versus time, the presence of some spikes can be noted, especially when the generator output is a CBR traffic with a fast packet rate (Fig. 2). Such spikes could be responsible for the high values of standard deviation. We will come back to this issue at the end of this section.

C. Traffic Generator Characterization

The analysis of the experimental results has highlighted that the behavior of the traffic generator when used to produce CBR traffic can be described with reference to three different regimes of operation, depending on the packet rate: 1) slow; 2) medium; and 3) fast. The observed evolution of IDT versus time, as well as the histogram of IDT occurrences [empirical probability density function (pdf)] vary in the three cases. In almost all the cases, some spikes in the IDT sequences are observable. These range from some milliseconds up to fractions of second and their amplitude is a function of the generation rate. The characterization is presented hereinafter according to such taxonomy.

1) *Slow Rate Generation*: Analyzing the plot shown in Fig. 3, showing the measured IDT versus time, when an imposed IDT value equal to 10 ms [Fig. 3(a)] and an 100 ms [Fig. 3(b)] has been, respectively, considered, it is possible to observe that the measured IDT shows a constant base trend around the nominal value. This reasonably results in a Gaussian-like histogram with mean equal to the expected IDT (Fig. 4). However, it is interesting to notice that the histograms includes some Gaussian-like shifted at multiples of $b = 10 \mu s$, whose peaks are much lower than the main one. We are now investigating whether the amount of such offset ($10 \mu s$) depends on the particular host on which the tests are conducted or rather on the generator loop.

It is worth noting that the Gaussian-like shapes farther from the main one are more evident when slower generation rates are used.

Spikes in the IDT sequence are absent or rare. When present, they are of the order of few milliseconds.

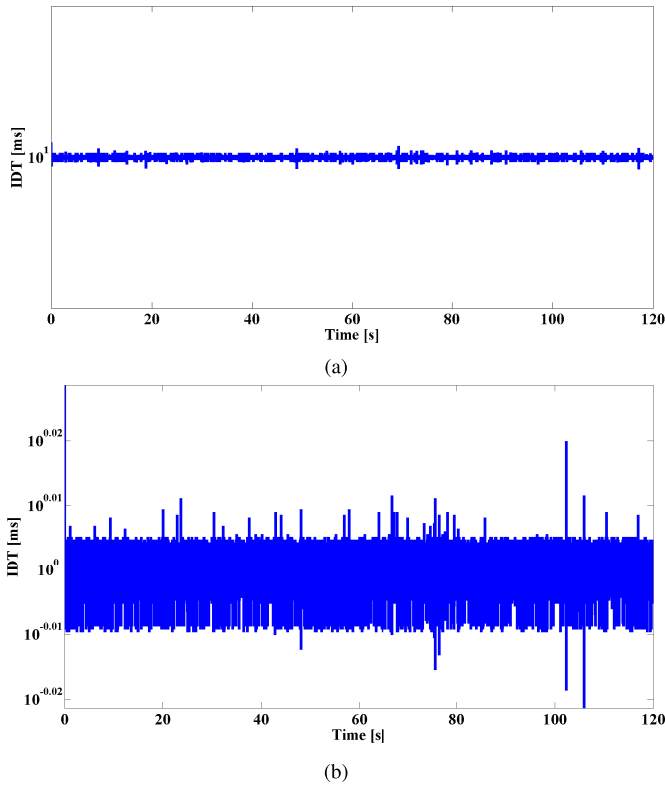


Fig. 3. Measured IDT versus time when a slow CBR has been imposed. (a) Imposed IDT = 10 ms. (b) Imposed IDT = 1 ms.

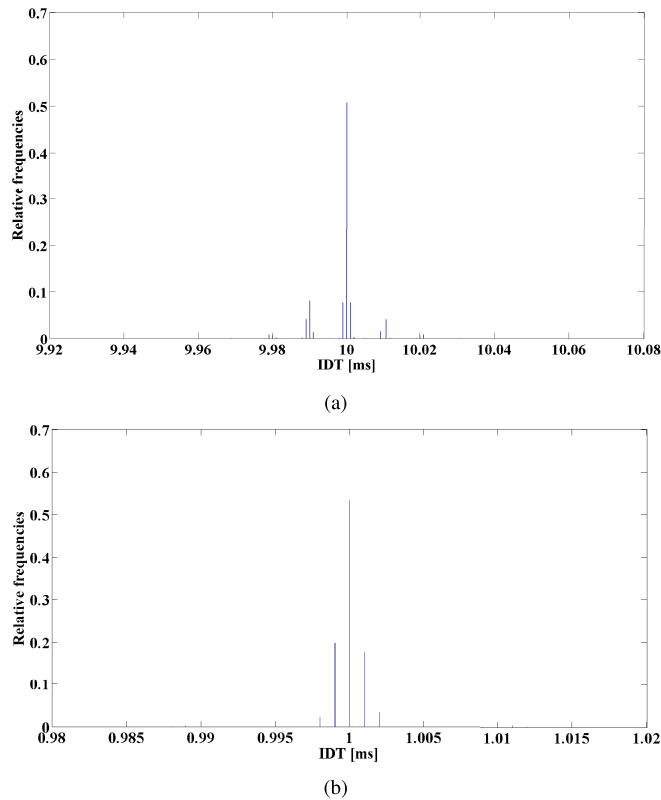


Fig. 4. Measured IDT histograms when a slow CBR has been imposed. (a) Imposed IDT = 10 ms. (b) Imposed IDT = 1 ms.

2) *Medium Rate Generation*: The evolution of measured IDT versus time and related histograms are shown in Figs. 5 and 6. Spikes are always present and of the order of

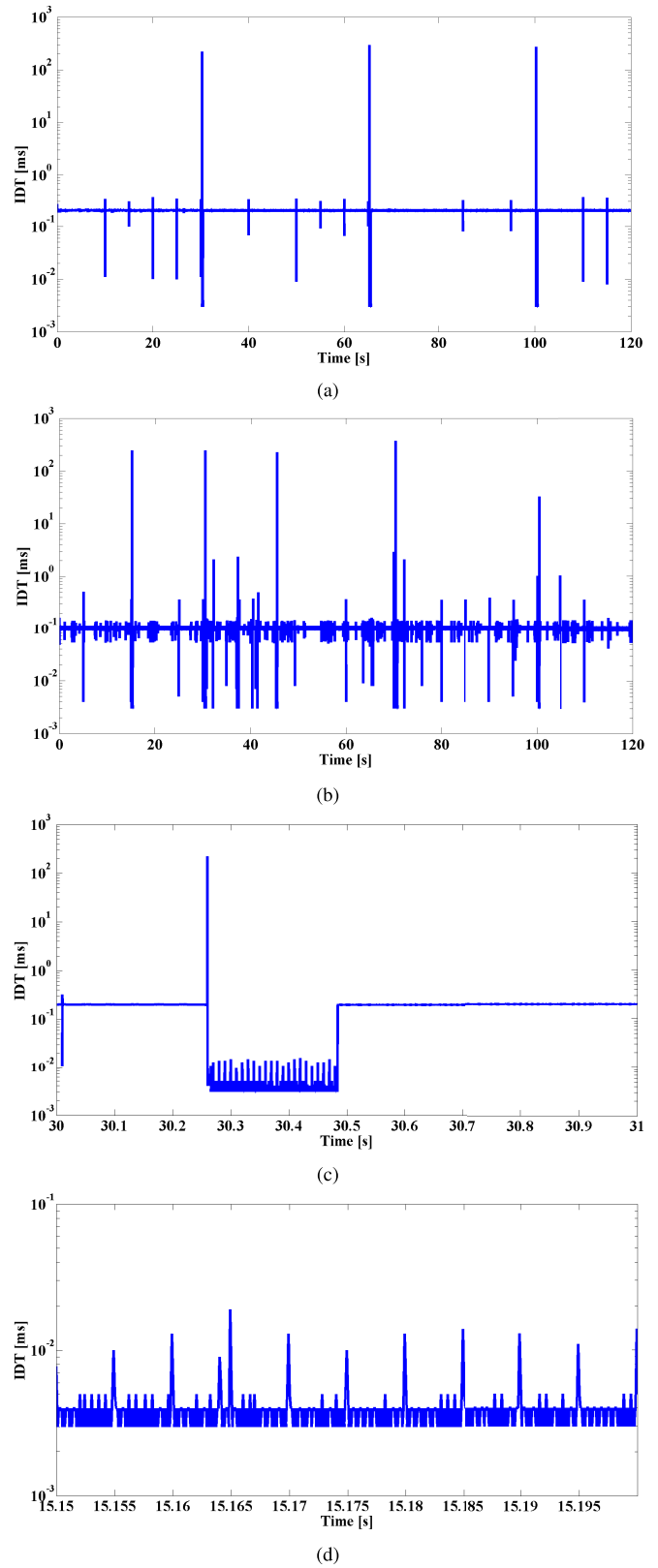


Fig. 5. Measured IDT versus time when a medium CBR has been imposed. (a) Imposed IDT = 0.2 ms. (b) Imposed IDT = 0.1 ms. (c) Particular of (a). (d) Particular of (b).

tens of milliseconds. Their number and amplitude are a direct function of the generation rate. The IDT recovery mechanism of the traffic generator triggers the quick generation of a

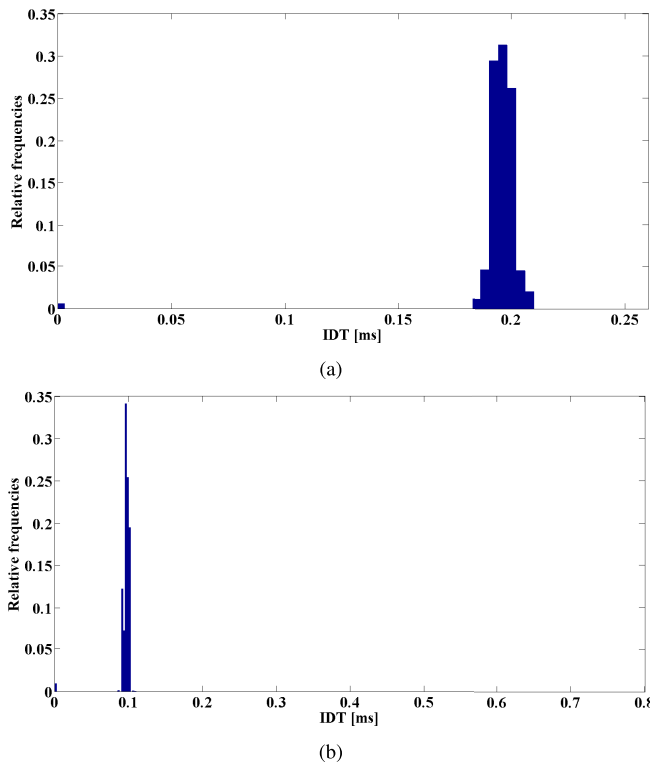


Fig. 6. Measured IDT histograms when a medium CBR has been imposed. (a) Imposed IDT = 0.2 ms. (b) Imposed IDT = 0.1 ms.

set of consecutive packets immediately after each spike, to compensate for such imbalance in the packet rate. Such packets are sent at the minimum IDT, m , permitted by the host (equal to $3 \mu\text{s}$ in our tests) and their number consequently depends on the amplitude of the spike. Within such train of recovery packets, some periodic mini-spikes can be observed every 50 packets, no matter the packet rate is [a zoomed example is shown in Fig. 5(d)]. Their amplitude is of the order of $10\text{--}25 \mu\text{s}$, and increases with the nominal packet rate.

Compared with the slow rate generation, in the histogram of IDTs, a part of the distribution is now shifted around m , with a negligible number of occurrences on the far right of the x -axis, representing the spikes. The histogram tends to assume a bimodal shape. When the generation rate is increased, the occurrences in the intervals around m become predominant, as more spikes are observed, whose amplitude increases too.

3) *Fast Rate Generation*: The generator can be said to be generating at fast rate, when the nominal IDT is lower than a threshold Θ ($\sim 60 \mu\text{s}$ in our tests). Although the machine can theoretically generate without problems at such rate, every now and then, the observed IDT is around Θ , which involves the activation of the IDT recovery mechanism, with few packets sent with the minimum IDT, as clearly shown in Fig. 7(c) and (d). Such phenomenon is independent from the so-called spikes, which are observed in this case, being of the order of hundreds of milliseconds. The two phenomena are thus superimposed, and both cause the activation of the recovery mechanism, though for different duration. It is the Θ -IDTs with the consequent compensation packets (and not the spikes) that are responsible for the relative standard deviation greater than 1.

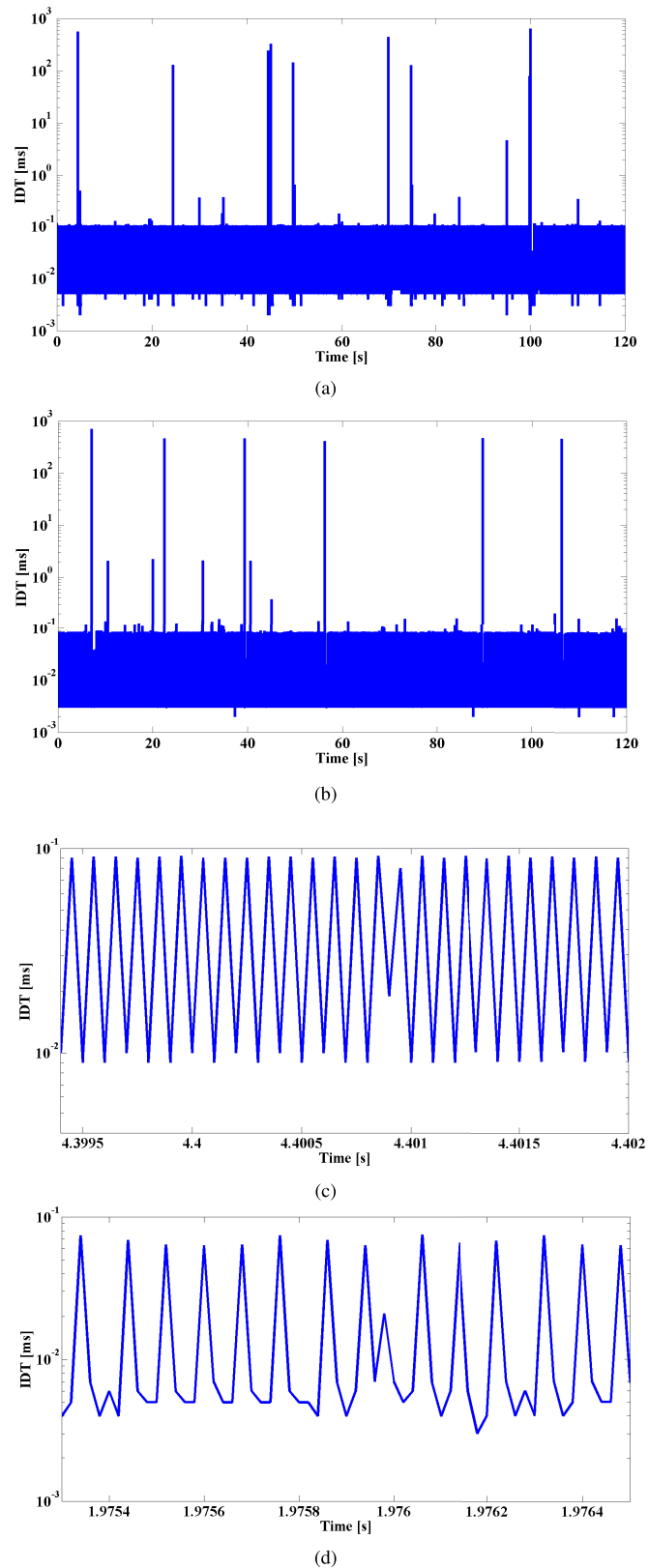


Fig. 7. Measured IDT versus time when a fast CBR has been imposed. (a) Imposed IDT = 0.05 ms. (b) Imposed IDT = 0.02 ms. (c) Particular of (a). (d) Particular of (b).

The histogram is clearly bimodal, with values concentrated around m and Θ , and a notable reduction of occurrences around the nominal IDT, as the rate is increased. At very

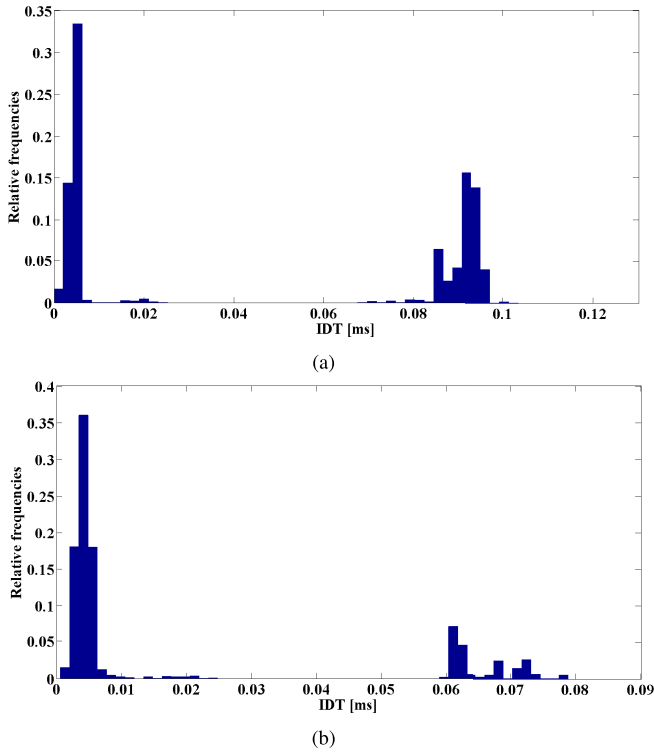


Fig. 8. Measured IDT histograms when a fast CBR has been imposed. (a) Imposed IDT = 0.05 ms. (b) Imposed IDT = 0.02 ms.

fast rates, this may result in having no packets with IDT equal or similar to the nominal IDT. All the packets would be concentrated around the threshold Θ and around the recovery IDT value, m , as shown in Fig. 8.

D. Traffic Generator Modeling

On the basis of the results discussed in Section III-C, a first modeling of the pdf of the IDT of the generated packets can be attempted as follows. At slow rate, we have a Gaussian-like distribution centered at the nominal value of IDT (μ), plus some tail Gaussian-like distributions shifted at multiples of b

$$f_X(x) = \alpha \cdot \mathcal{N}(\mu, \sigma^2) + \sum_i \beta_i \cdot \mathcal{N}(\mu \pm i \cdot b, \sigma_i^2) \quad (1)$$

where $\beta_i < 1$ are decreasing with μ and i , and the factor $\alpha < 1$ grants the pdf normalization.

At medium rate generation, the presence of spikes in the IDT evolution versus time triggers the IDT recovery mechanism, which determines the appearance of nonzero pdf values at m . The pdf model is thus refined to yield

$$f_X(x) = \alpha \cdot \mathcal{N}(\mu, \sigma^2) + \sum_i \beta_i \cdot \mathcal{N}(\mu \pm i \cdot b, \sigma_i^2) + \sum_{i=1}^{n_s} \frac{1}{N} \cdot \delta(x - s_i) + \frac{n_r}{N} \cdot \delta(x - m) \quad (2)$$

where δ is the Dirac's delta, n_s is the number of spikes, each of which has an amplitude of s_i and is thus represented in the pdf by a delta located at s_i , and n_r is the number of recovery

TABLE III
EXPERIMENTAL RESULTS OF THE ACCURACY TEST PERFORMED ON A DIFFERENT SOFTWARE TRAFFIC GENERATOR (MGEN)

IDT [ms]	μ_{IDT} [ms]	Δ_{IDT} [%]	σ_{IDT} [ms]	$\sigma_{IDT_{rel}}$ [%]	max [ms]	min [ms]
100	100.00	0.00006	0.20	0.20	103.05	96.93
10	10.00	0.00008	0.71	7.1	68.50	0.008
2	2.00	0.00003	0.50	25	74.14	0.008
1	1.00	0.00004	0.50	50	140.58	0.008
0.1	0.10	0.00004	0.70	700	582.05	0.007
0.05	0.05	1.3	1.2	2500	1582.36	0.007

packets with IDT equal to m . Actually, the amplitude of the so-called tail distributions is negligible at such rates, and thus the term can be neglected, yielding

$$f_X(x) = \alpha \cdot \mathcal{N}(\mu, \sigma^2) + \sum_{i=1}^{n_s} \frac{1}{N} \cdot \delta(x - s_i) + \frac{n_r}{N} \cdot \delta(x - m). \quad (3)$$

Finally, at fast rate, a part of pdf is concentrated around the threshold Θ , with distribution $g(x, \theta)$, and a large number of packets are generated with IDT in a neighborhood of m

$$f_X(x) = \alpha \cdot \mathcal{N}(\mu, \sigma^2) + g(x, \theta) + \gamma \cdot \mathcal{N}(m, \sigma_m^2) + \sum_{i=1}^{n_s} \frac{1}{N} \cdot \delta(x - s_i) \quad (4)$$

with $\gamma < 1$. Please note that coefficients α , β_i , and γ may vary when moving from slow to medium to fast rate distributions. In addition, α may be zero, which means that the pdf around the nominal IDT is nonzero only for slower generation rates (Fig. 8).

E. Further Experiments and Remarks

To verify the validity of the model, we have performed some additional tests with an alternative traffic generator, MGEN [24]. For these experiments we used the same conditions of the previous ones, carried out with D-ITG. In particular, we used the same hosts, we generated traffic locally (i.e., through the loopback interface), with constant IDT and constant PS (512 bytes of transport-layer payload), and we used the difference between consecutive timestamps (reported by the sending process) as IDT samples. Table III shows the obtained results with MGEN. By comparing the results in Tables II and III, we can see that the behavior of the two generators is similar in terms of difference between imposed and mean IDT, but MGEN shows a larger experimental standard deviation.

It is interesting to observe that the pdf model derived for D-ITG holds for MGEN too. Fig. 9(a) and (b), which shows the measured IDT histograms for MGEN, when imposed IDT are equal to, respectively, 0.1 and 0.05 ms, are very similar to Figs. 6(b) and 8(a), which are related to D-ITG. The only difference is a wider distribution around the modal values for MGEN, with respect to D-ITG, which is coherent with the higher standard deviation it exhibits for 10000 and 20000 pkt/s rates.

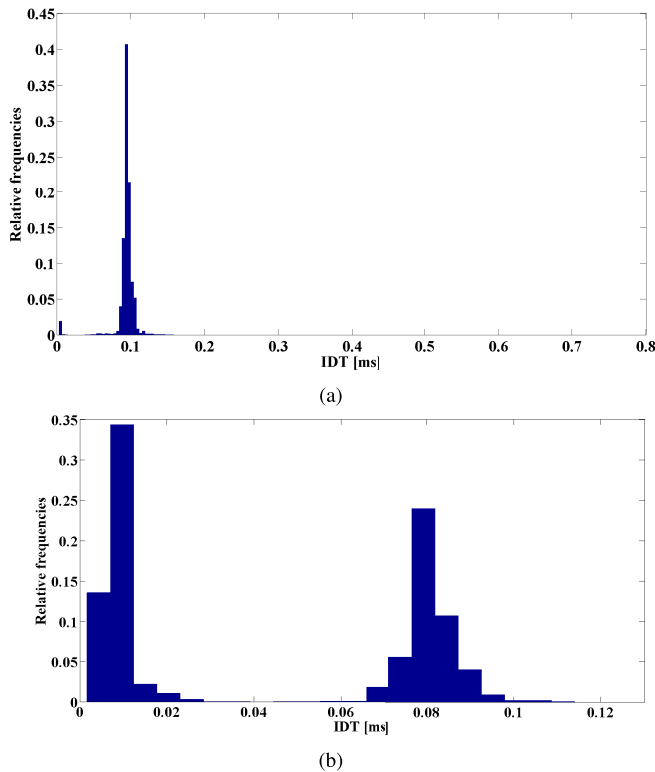


Fig. 9. Measured IDT histograms for MGEN for medium and fast imposed CBR. (a) Imposed IDT = 0.1 ms. (b) Imposed IDT = 0.05 ms.

TABLE IV
EXPERIMENTAL RESULTS OF THE ACCURACY TESTS
INVOLVING AN ACTUAL NETWORK INTERFACE

IDT [ms]	μ_{IDT} [ms]	Δ_{IDT} [%]	σ_{IDT} [ms]	$\sigma_{IDT,rel}$ [%]
100	100.00	0.00008	0.024	0.0024
10	10.00	0.00007	0.0088	0.088
2	2.00	0.00005	0.026	1.3
1	1.00	0.00005	0.0022	50
0.1	0.10	0.00005	0.69	693

It is worth giving a final remark about the effects of the use of an actual interface on the results. We have therefore performed the same tests described before related to D-ITG, but using real network interfaces. For these tests, we used two hosts with the same characteristics reported before and directly connected through a crossover cable using their gigabit Ethernet interfaces. Related results are shown in Table IV. A comparison with the results given in Table II, which are related to the loopback tests, permits to highlight some minor differences to the relative experimental standard deviation in the two cases. On the contrary, no significant differences are observables with respect to the Δ_{IDT} . The reason for such outcomes is that, while it is true that a larger number of kernel pieces are involved in the transmission process, in this paper timestamps are taken inside the process sending the packets (i.e., inside the loop shown in Fig. 1), so before the packets leave the sending host and go through the network.

Therefore, the underlying network is not expected to have relevant influence on the distribution of IDTs, in this case.

IV. CONCLUSION

This paper is the first step toward the goal of evaluating the measurement uncertainty of the IDTs provided by software traffic generators. It has presented a methodology and experimental results aimed to characterize the internal generation cycle of a well-known, open source generator, namely D-ITG, for different systems, and under the best possible conditions, i.e., with the minimum system loads. Having similar characteristics of other software traffic generators, D-ITG has been used in this paper as a playground for measuring the uncertainty of the IDT provided by software traffic generators.

The results have shown that for slower packet rates, the experimental standard deviations experienced for the execution of a simplified loop select()-gettimeofday() and for the execution of the internal loop of the software traffic generator used for the tests, are of the same order of magnitude. On the contrary, for packet rates faster than 1000 pkt/s, the IDTs measured in the tests with the software traffic generator exhibit a much higher experimental standard deviation, compared with those measured when the simplified loop is executed. This suggests that the other functionalities and operations performed by the traffic generator are responsible for a larger variability of IDTs, when the latter are lower than 1 ms. Of course, a much wider experimental campaign is needed to assess this behavior. As regards the difference between imposed and average measured IDT, no relevant gap is observed in the two cases.

This paper has also presented a model for the IDT. The model considers the various factors that affect the accuracy of the IDT and shows how these factors impact the IDT. Experiments, that have been carried out also considering a different traffic generator, show the generalizability of the results.

Ongoing research activities are focused on extending the set of test cases at different (faster) packet rates, with different (non-CBR) traffic profiles, and analyzing the results, to infer the roles of the different sources of measurement uncertainty and ultimately evaluate the uncertainty of the IDT of the traffic generator, in compliance with the GUM [11]. The final goal is to define a procedure, based on the presented model, for predicting the behavior of a software generator with IDT-adaptation mechanism after a given set of tests on streams of packets with different nominal IDTs are launched to estimate the parameters appearing in (1)–(4).

ACKNOWLEDGMENT

The authors would like to thank P. D'Ambrosio for his help in the execution of the experiments.

REFERENCES

- [1] L. Angrisani and C. Narduzzi, "Testing communication and computer networks: An overview," *IEEE Instrum. Meas. Mag.*, vol. 11, no. 5, pp. 12–24, Oct. 2008.

- [2] F. Dressler, "Policy-based traffic generation for IP-based networks," in *Proc. 25th Poster Session IEEE Conf. Comput. Commun. (INFOCOM)*, Barcelona, Spain, Apr. 2006, pp. 1–3.
- [3] K. V. Vishwanath and A. Vahdat, "Realistic and responsive network traffic generation," *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 4, pp. 111–122, Aug. 2006. [Online]. Available: <http://doi.acm.org/10.1145/1151659.1159928>
- [4] A. Botta, A. Dainotti, and A. Pescapé, "A tool for the generation of realistic network workload for emerging networking scenarios," *Comput. Netw.*, vol. 56, no. 15, pp. 3531–3547, Oct. 2012. [Online]. Available: <http://dx.doi.org/10.1016/j.comnet.2012.02.019>
- [5] A. Santos, S. Fernandes, R. Antonello, G. Szabo, P. Lopes, and D. Sadok, "High-performance traffic workload architecture for testing DPI systems," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Dec. 2011, pp. 1–5.
- [6] A. Botta, A. Dainotti, and A. Pescapé, "Do you trust your software-based traffic generator?" *IEEE Commun. Mag.*, vol. 48, no. 9, pp. 158–165, Sep. 2010.
- [7] M. Paredes-Farrera, M. Fleury, and M. Ghanbari, "Precision and accuracy of network traffic generators for packet-by-packet traffic analysis," in *Proc. 2nd Int. Conf. Testbeds Res. Infrastruct. Develop. Netw. Commun. (TRIDENTCOM)*, Mar. 2006, pp. 31–37.
- [8] P. Arlos, "On the quality of computer network measurements," Ph.D. dissertation, Dept. Telecommun. Syst.—School Eng., Blekinge Inst. Technol., Karlskrona, Sweden, 2005.
- [9] K. Wac, P. Arlos, M. Fiedler, S. Chevul, L. Isaksson, and R. Bults, "Accuracy evaluation of application-level performance measurements," in *Proc. 3rd EuroNGI Conf. Next Generat. Internet Netw.*, May 2007, pp. 1–5.
- [10] A. Dainotti, A. Botta, and A. Pescapé, "Do you know what you are generating?" in *Proc. ACM CoNEXT Conf.*, 2007, pp. 32:1–32:2. [Online]. Available: <http://doi.acm.org/10.1145/1364654.1364695>
- [11] *Guide 100 Evaluation of Measurement Data—Guide to the Expression of Uncertainty in Measurement*, JCGM, Sèvres, France, 2008.
- [12] E. Nunzi, "Uncertainties analysis in RTT network measurements: The GUM and RFC approaches," in *Proc. IEEE Int. Workshop Adv. Methods Uncertainty Estimation Meas. (AMUEM)*, Apr. 2006, pp. 87–91.
- [13] L. Angrisani, S. D'Antonio, M. Esposito, and M. Vadursi, "Techniques for available bandwidth measurement in IP networks: A performance comparison," *Comput. Netw.*, vol. 50, no. 3, pp. 332–349, Feb. 2006. [Online]. Available: <http://dx.doi.org/10.1016/j.comnet.2005.06.003>
- [14] L. De Vito, S. Rapuano, and L. Tomaciello, "One-way delay measurement: State of the art," *IEEE Trans. Instrum. Meas.*, vol. 57, no. 12, pp. 2742–2750, Dec. 2008.
- [15] A. Bondavalli, A. Ceccarelli, L. Falai, and M. Vadursi, "A new approach and a related tool for dependability measurements on distributed systems," *IEEE Trans. Instrum. Meas.*, vol. 59, no. 4, pp. 820–831, Apr. 2010.
- [16] C. M. De Dominicis, P. Pivato, P. Ferrari, D. Macii, E. Sisinni, and A. Flammini, "Timestamping of IEEE 802.15.4a CSS signals for wireless ranging and time synchronization," *IEEE Trans. Instrum. Meas.*, vol. 62, no. 8, pp. 2286–2296, Aug. 2013.
- [17] D. Fontanelli, D. Macii, S. Rinaldi, P. Ferrari, and A. Flammini, "A servo-clock model for chains of transparent clocks affected by synchronization period jitter," *IEEE Trans. Instrum. Meas.*, vol. 63, no. 5, pp. 1085–1095, May 2014.
- [18] L. Angrisani, A. Botta, G. Miele, and M. Vadursi, "An experimental characterization of the internal generation cycle of an open-source software traffic generator," in *Proc. IEEE Int. Workshop Meas. Netw. (M&N)*, Oct. 2013, pp. 74–78.
- [19] L. Angrisani, D. Capriglione, L. Ferrigno, and G. Miele, "A methodological approach for estimating protocol analyzer instrumental measurement uncertainty in packet jitter evaluation," *IEEE Trans. Instrum. Meas.*, vol. 61, no. 5, pp. 1405–1416, May 2012.
- [20] L. Angrisani, D. Capriglione, L. Ferrigno, and G. Miele, "Internet protocol packet delay variation measurements in communication networks: How to evaluate measurement uncertainty?" *Measurement*, vol. 46, no. 7, pp. 2099–2109, Aug. 2013.
- [21] K. V. Vishwanath and A. Vahdat, "Swing: Realistic and responsive network traffic generation," *IEEE/ACM Trans. Netw.*, vol. 17, no. 3, pp. 712–725, Jun. 2009.
- [22] A. W. Kolesnikov and B. E. Wolfinger, "Web workload generation according to the unilog approach," in *Proc. KIVS*, 2011, pp. 49–60.
- [23] TG. SRI and University of Southern California. [Online]. Available: <http://www.postel.org/tg/tg.html>, accessed Nov. 15, 2013.
- [24] Naval Research Laboratory (NRL) PROTOcol Engineering Advanced Networking (PROTEAN) Research Group. Multi Generator (MGEN). [Online]. Available: <http://cs.itd.nrl.navy.mil/work/mgen/>, accessed Mar. 10, 2014.



Leopoldo Angrisani (M'09–SM'12) was born in 1969. He received the M.S. (*cum laude*) degree in electronic engineering from the University of Salerno, Salerno, Italy, in 1993, and the Ph.D. degree in electrical engineering with the curriculum in electrical and electronic measurements from the University of Naples Federico II, Naples, Italy, in 1997.

He has been a Full Professor of Electrical and Electronic Measurements with the Department of Control Systems and Information Engineering, University of Naples Federico II, since 2010, where he is currently with the Department of Information Technologies and Electrical Engineering. He is also a member of the Board of the Ph.D. Program on Information Technologies and Electrical Engineering with the University of Naples Federico II. He has been involved in several scientific research programs, both local and of relevant national interest, and in many industrial research projects, in cooperation with small, medium, and great enterprises, for some of which he served and currently serves as the Scientific Coordinator. He has authored or co-authored over 250 scientific articles, one-third of which have been published in relevant international journals. His current research interests include electrical and electronic measurements, and, in particular, performance assessment of telecommunication systems and apparatuses, communication networks test and measurement, compressive sampling-based measurements, measurements for homeland security applications, detection, measurement, and classification of nonstationary signals, automatic measurement instruments based on digital signal processors, and measurement uncertainty.

Prof. Angrisani is a member of the Italian Association "Electrical and Electronic Measurements Group," the National Inter-University Consortium for Telecommunications, the Technical Committee CT 1/25 Terminology, Quantities and Units of the Italian Electrotechnical Committee, the Administrative Committee of the National Inter-University Research Consortium on Metrics and Measurement Technology on Electrical Systems, and the Administrative Committee of the Center of Advanced Metrological Services, University of Naples Federico II. He is the Representative of Italy with the Validation Team-VT 60050 for maintenance and management of the International Electrotechnical Vocabulary. He is a member, as an expert, of the Technical Board of the Italian project concerning the measurement of Internet fixed access performance, and the Technical Board of the Italian project concerning the measurement of the quality perceived by the final user of mobile personal communications offered through terrestrial wireless networks, both promoted by the Italian Communications Regulatory Authority. He is a Senior Member of the IEEE Instrumentation and Measurement Society and the IEEE Communications Society. He is the Vice President of the Italy Chapter of the IEEE Instrumentation and Measurement Society. He was the General Chairman of the first and second edition of IEEE International Workshop on Measurements and Networking in 2011 and 2013. He was one of the Promoters of the TC-37 Measurements and Networking Technical Committee of the IEEE Instrumentation and Measurement Society. He was a recipient of the IET Communications Premium Award for the paper entitled Performance Measurement of IEEE 802.11b-Based Networks Affected by Narrowband Interference Through Cross-Layer Measurements (published in IET Communications, Vol. 2, No. 1, January 2008) in 2009, and the IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT Outstanding Reviewer of 2013 Award from the IEEE Instrumentation and Measurement Society in 2013.



Alessio Botta (M'12) received the M.S. degree in telecommunications engineering and the Ph.D. degree in computer engineering and systems from the University of Naples Federico II, Naples, Italy.

He currently holds a post-doctoral position with the Department of Computer Engineering and Systems, University of Naples Federico II. He has co-authored over 50 international journal (the IEEE COMMUNICATIONS MAGAZINE, the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, and Elsevier's *Computer Networks*) and conference [the IEEE Global Communications (Globecom), the IEEE International Conference on Communications (ICC), and the IEEE Symposium on Computers and Communications (ISCC)] publications. His current research interests include networking, and, in particular, network performance measurement and improvement, with a focus on wireless and heterogeneous systems.

Dr. Botta has served and serves as an independent reviewer of research and implementation project proposals for the Romanian government. He was a recipient of the Best Local Paper Award at the IEEE ISCC 2010. In the research area of networking, he has chaired international conferences and workshops, served and serves several technical program committees of international conferences (IEEE Globecom and IEEE ICC), and acted as a reviewer for different international conferences (the IEEE Conference on Computer Communications) and journals (the IEEE TRANSACTIONS ON MOBILE COMPUTING, the IEEE NETWORK MAGAZINE, and the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY).



Gianfranco Miele (S'06–M'08) received the M.S. (*cum laude*) degree in telecommunication engineering and the Ph.D. degree in electrical and information engineering from the University of Cassino, Cassino, Italy, in 2004 and 2008, respectively.

He has been an Assistant Professor with the Department of Electrical and Information Engineering, University of Cassino and Southern Lazio, Cassino, since 2009. He has authored or co-authored about 40 papers in instrumentation and measurements. His current research interests include electrical and electronic measurements, and, in particular, design and implementation of innovative methods for performance assessment of RF telecommunication systems and communication networks, image-based measurement systems, measurement of electromagnetic compatibility, and DSP-based measurement systems.

Dr. Miele is a member of the Italian Association "Electrical and Electronic Measurements Group" and the IEEE Instrumentation and Measurement Society. He was a recipient of the Carlo Offelli Prize for the Best Ph.D. Dissertation in electronic measurement entitled Design and Implementation of an Apparatus for Reliable and Repeatable Power Measurement in DVB-T Systems in 2008.



Antonio Pescapé (SM'09) received the M.S. Laurea degree in computer engineering and the Ph.D. degree in computer engineering and systems from the University of Naples Federico II, Naples, Italy.

He is currently an Assistant Professor with the Department of Electrical Engineering and Information Technology, University of Naples Federico II. He teaches courses in computer networks, computer architectures, programming, and multimedia, and has also supervised and graduated over 100 B.S., M.S., and Ph.D. students. He has co-authored over 130 journal (the IEEE/ACM TRANSACTION ON NETWORKING, *Communications of the ACM*, the IEEE COMMUNICATIONS MAGAZINE, the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, the IEEE WIRELESS COMMUNICATIONS MAGAZINE, and the IEEE NETWORKS) and conference (SIGCOMM, Conext, IMC, PAM, Globecom, and ICC) publications, and has co-authored a patent. His current research interests include Internet monitoring, measurements and management, and network security.

Prof. Pescapé has served and serves as the Workshops and Conferences Chair [including the IEEE ICC (NGN symposium)], and on over 90 Technical Program Committees of IEEE and ACM conferences. He serves as an Editorial Board Member of the *Journal of Network and Computer Applications*, served as an Editorial Board Member of the IEEE SURVEY AND TUTORIALS from 2008 to 2011, and was a Guest Editor of the Special Issue of Computer Networks on Traffic classification and its applications to modern networks. He has served and serves as an independent reviewer/evaluator of research and implementation projects and project proposals co-funded by the EU Commission, Swedish government, several Italian local governments, the Italian Ministry for University and Research, and the Italian Ministry of Economic Development. For his research activities, he has received several awards.



Michele Vadursi (M'08–SM'12) received the M.S. (Hons.) degree in telecommunication engineering and the Ph.D. degree in electrical engineering from the University of Naples Federico II, Naples, Italy.

He has been an Assistant Professor with the Department of Engineering, University of Naples Parthenope, Naples, since 2006. He was awarded the Italian scientific qualification of Associate Professor in Electrical and Electronic Measurements in 2012. He has authored or co-authored about 80 journal and conference papers in measurement systems and metrology. His current research interests include the design, implementation, and characterization of innovative measurement methods for testing communication systems and networks, and measurements on distributed systems and systems of systems.

Dr. Vadursi is a member of the of the IEEE Instrumentation and Measurement Society and the Italian Association "Electrical and Electronic Measurements Group." He was a recipient of the *IET Communications* Premium Award in 2009 as a co-author of the paper entitled Performance Measurement of IEEE 802.11b-Based Networks Affected by Narrowband Interference Through Cross-Layer Measurements (IET Communications, 2008).