Proceedings of the Eighth IASTED International Conference
INTERNET AND MULTIMEDIA SYSTEMS AND APPLICATIONS
August 16-18, 2004, Kauai, Hawaii, USA

A Practical Demonstration of Network Traffic Generation

Stefano Avallone, Donato Emma, Antonio Pescapè and Giorgio Ventre
Dipartimento di Informatica e Sistemistica, University of Napoli "Federico II" (Italy)
{stavallo, pescape, giorgio}@unina.it, demma@napoli.consorzio-cini.it

ABSTRACT
Simulation modeling of computer networks is an effective technique for evaluating the performance of network, transport, and application-level protocols. Traffic generation is one of the key challenges in modeling and simulating the Internet. In this paper we present a real simulation tool for controlled traffic generation over real networks. This work presents a platform, named *Distributed Internet Traffic Generator* (D-ITG), which allows to simulate Internet Traffic and accurately replicate appropriate stochastic processes for both IDT (*Inter Departure Time*) and PS (*Packet Size*) random variables over real IP networks. We believe that D-ITG shows interesting properties when compared to other traffic generators and that it permits to accurately analyze network performance.

KEY WORDS: Parallel and Distributed Processing, Internet Tools, Modeling and Simulation.

1. Introduction

As far as mathematical and theoretical aspects, compared with other scientific areas, network research appears significantly less mature concerning methodology. Internet research is affected by models that sometimes are poorly suited to the problem under investigation, by lack of understanding of properties and limitations of the models used, and by tools that have various limitations and that are poorly integrated [1] [2]. While each of the analysis methods (simulation, emulation, testbed experiments and Internet-wide experiments) has its own particular strengths and shortcomings, typically only one of these methods is used to investigate a particular problem. Certain weaknesses of the chosen methods can have unwanted implications on the results and deductions made from them. There appears to be insufficient comparison or adjustment between results obtained by different methods.

As computer networking has become more ubiquitous, researchers are increasingly focused on optimizing computer networks performance and improving network utilization in terms of throughput and offered delay, jitter and packet loss. This process cannot leave the study of traffic patterns and properties out of consideration. In the last twenty years researchers have been looking for the definition of stochastic processes that could be used as accurate and simple models for traffic generation in packet switched networks and in particular in IP networks. In order to be as realistic as possible, traffic models should accurately represent relevant statistical properties of the original traffic. On the other hand, traffic models should not become overly complex. Thus, a compromise is needed between accuracy of the representation and problem manageability. It is therefore of critical importance to identify which traffic characteristics are essential and which can be ignored by the modeling process.

One of the main applications of traffic models is the generation of synthetic, yet realistic traffic to be injected into a network. In the case of studies related to the Internet, simulations should not only reflect the wide scale of real scenarios, but also the rich variety of traffic sources, in terms of both protocol typologies and data generation patterns. Modeling the Internet traffic is an important and essential task and we think that traffic theory should be increasingly used to guide the design of the future multi-service and integrated Internet. It is unlikely that we will be able to understand traffic characteristics, predict network performance (Quality of Service (QoS), Service Level Agreement (SLA) definition, …), or design dimensioning tools without analytical and rigorous models. The successful evolution of the Internet is tightly coupled to the ability to design simple and accurate models with the property of reproducibility. Traffic theory suggests us the application of mathematical modeling to explain the relationship between traffic performance and network capacity, traffic demand and experimented performance [3] [4]. Deriving such relations allows us to understand what kinds of performance guarantees are feasible and what kinds of traffic control or traffic engineering techniques are necessary. The objective of traffic theory is ultimately the definition of settled network engineering procedures as, for example, the *Erlang* formula in dimensioning telephone networks. The derivation of these procedures and the proof of their general validity, however, require mathematical modeling. Unfortunately, nowadays Internet traffic is far too complicated to be modeled using the techniques developed for the telephone network or for simple computer systems and currently traffic theory plays a very minor role in the design of the Internet. Network management has so far been dominated by passive monitoring. Emerging networking technologies however force the development of active testing and performance analysis tools. As a result of the limitations of existing testing and performance analysis tools, our tool was developed in order to make available a traffic generator with a plenty of useful features.

The rest of the paper is organized as follows. Section 2 presents the motivations at the base of this work. In the Section 3 our analysis and experimentation is presented. Finally, Section 4 presents some conclusion remarks and issues for research.

2. Motivation

The purpose of our Internet Traffic Generator is to build up a suite that can be easily used to generate repeatable sets of experiments by using a reliable and realistic mixture of available traffic typologies. D-ITG enables to simulate many traffic scenarios that could be originated by a typical network test-case made of large number of users and network devices, as well as by different network topologies.
Typically other traffic generators can only generate UDP traffic with limited generation performance and offer a limited set of traffic source models. D-ITG implements both TCP and UDP traffic generation according to several probability distributions (*exponential, uniform, constant, pareto, cauchy, normal, ...*) both for *IDT (Inter Departure Times)* and *PS (Packet Size)* random variables.
Through the use of our tool, a network administrator can evaluate the performance of a network, locating possible problems and observing the effects of different solutions. It is rarely possible to actually set up enough workstations and/or PCs and users to create loads equivalent to those on a real network. We have developed D-ITG [17] to address this issue. The generation of realistic traffic patterns helps in studying protocols and applications of interest in today's Internet. D-ITG can generate UDP and TCP traffic and is designed for the generation of "layer 7" traffic (application layer traffic). D-ITG primary design goals are: (i) reproducibility of network experiments: we implemented a method that can use the same seed for different stochastic experiments. It is possible to reproduce experiments by choosing the same seed value for the *packets inter-departure* and *packets size* random processes; (ii) investigation of scaling effects: using different network loads or different network configurations is possible to study scalability problems; (iii) increase the generation performance with respect to existing Traffic Generators; (iv) increase the available traffic source models with respect to other Traffic Generators; (v) the possibility of simulating more traffic sources, repeating many times exactly the same traffic pattern (not only its mean value) and getting information not only about received packets but also about transmitted packets; (vi) measuring the round trip time and one way delay.
D-ITG can be applied for both benchmarking and measurement techniques. D-ITG provides tools for understanding network behavior and by using these properties it is possible to trace guidelines for network planning and real implementation. With our generator we can simulate (and not emulate) traffic and thanks to it, it is possible to create a physical test whereby network parameters can be collected and analyzed. In our vision traffic simulation means the reproduction of a "traffic profile" according to theoretical stochastic models. Instead, traffic emulation means the reproduction of a specific protocol (i.e. reproduction of http messages without using a browser). A detailed presentation of D-ITG with respect to traffic model validation and software architecture is reported in [5] and in [6] whereas extensive heterogeneous network measurements using D-ITG are reported in [7] and [8]. The distributed architecture and an experimental validation of D-ITG is presented in [9]. In [5] is presented a detailed related works section too. As far as this last point in the Figure 1 a detailed comparative analysis is shown. These results are related to an experimentation of 75000 UDP pkt/s with packet size equal to 1024 byte and an experiment duration equal to 60 s. The experimental testbed is made by two Linux PCs with a Gigabit back-to-back connection. PCs hardware details are: Intel Pentium 4 2,6 GHz, CPU Cache 512; Controller Ethernet: 3Com Gigabit LOM (3c940); Hard Disk: Maxtor 6Y080L0 (Fast ATA/Enhanced IDE Compatible, Ultra ATA/133 Data Transfer Speed, 2MB Cache Buffer, Quiet Drive Technology, 100% FDB - fluid dynamic bearing – motors). In this analysis we have taken into account the following traffic generators: Mtools [6], Rude/Crude [10], Mgen [11], Iperf [12] and finally UDPgenerator [13]. As far as experimental results, D-ITG shows the best performance. It is important to underline that Iperf works in a different way with respect to D-ITG. Indeed Iperf does not produce a log file: it provides only an estimation of received and transmitted date rate at the end of the experiment. For a deep analysis, a comprehensive list of traffic generators can be found at www.grid.unina.it/software/ITG/link.html.

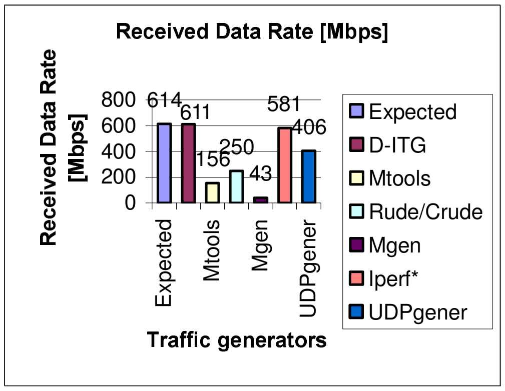


Figure 1 : Traffic Generators, a comparative analysis

Due to data rate comparative analysis, the previous results are carried out using IDT and PS equal to constant value. In the next section we show some simple examples of stochastic synthetic traffic generation.

3. Simulating Internet Traffic: analysis and experimentation

This section reports some practical usage examples of D-ITG and comments on the results. In particular we show:

- TCP traffic generation:

    - Constant IDT and Poisson distributed PS
    - Constant IDT and uniformly distributed PS
    - Pareto distributed IDT and constant PS
- UDP traffic generation
    - *Variable Bit Rate* Video Traffic: constant IDT and normally distributed PS
- Telnet traffic generation
- VoIP (*Voice over IP*) traffic generation

The former part shows TCP and UDP traffic generation whereas the latter one shows the application layer traffic generation. The first example reports the generation of TCP traffic. The payload size of all packets is constant and equal to 16 bytes. The flow lasts 60 seconds and the packet generation process is a pareto process, characterized by shape equal to 3 and scale equal to 10. We want to calculate now the expected average bit rate, in order to verify the accuracy of our D-ITG. First, we note that *mcalc* and *ez* utilities simply consider the payload size of packets (and not their full size) in determining the average bit rate, and we will do so as well [14].

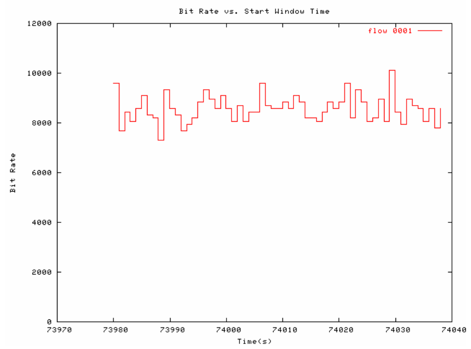
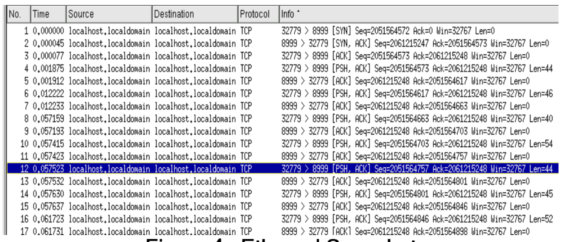


Figure 2 : Resulting Plots of TCP generation with IDT=pareto and PS= constant (window size of 1 sec)

```
Num pkts recvd   :  4010
Join delay       : 73979.945 sec
Recv pkt rate    :    66.939 pkt/sec
Recv data rate   :     8.570 kbps
Pkts dropped     :     0
```
Figure 3: Output results of TCP generation with IDT=pareto and PS= constant

In case of Pareto with parameters previously defined, the mean random variable is equal to 15, that means the average time interval between the departure of two consecutive packets is 15 msecs; the payload size is 16 bytes (=128 bits). Therefore the average bit rate is equal to the ratio between 128 bits and 15 msecs, which yields 8,533Kbps. The resulting plots and mcalc output derived from the log file of sender are shown in Figure 2 and Figure 3 and confirm our expectations. Notice that bit rate plots (Figure 2) need the specification of a window size that is the time interval on which the bit rate must be computed. Moreover, notice that all the parameters from mcalc output are related to traffic generated, even though they are addressed as "received" (this is due to the fact that MGEN does not log sent packets and therefore analyzes only receiver's log files).


```
No. Time     Source                Destination           Protocol Info
  1 0.000000 localhost.localdomain localhost.localdomain TCP      32779 > 8999 [SYN] Seq=2051564572 Ack=0 Win=32767 Len=0
  2 0.000045 localhost.localdomain localhost.localdomain TCP      8999 > 32779 [SYN, ACK] Seq=2061215247 Ack=2051564573 Win=32767 Len=0
  3 0.000077 localhost.localdomain localhost.localdomain TCP      32779 > 8999 [ACK] Seq=2051564573 Ack=2061215248 Win=32767 Len=0
  4 0.001875 localhost.localdomain localhost.localdomain TCP      32779 > 8999 [PSH, ACK] Seq=2051564573 Ack=2061215248 Win=32767 Len=44
  5 0.001912 localhost.localdomain localhost.localdomain TCP      8999 > 32779 [ACK] Seq=2061215248 Ack=2051564617 Win=32767 Len=0
  6 0.012222 localhost.localdomain localhost.localdomain TCP      32779 > 8999 [PSH, ACK] Seq=2051564617 Ack=2061215248 Win=32767 Len=46
  7 0.012233 localhost.localdomain localhost.localdomain TCP      8999 > 32779 [ACK] Seq=2061215248 Ack=2051564663 Win=32767 Len=0
  8 0.057159 localhost.localdomain localhost.localdomain TCP      32779 > 8999 [PSH, ACK] Seq=2051564663 Ack=2061215248 Win=32767 Len=40
  9 0.057193 localhost.localdomain localhost.localdomain TCP      8999 > 32779 [ACK] Seq=2061215248 Ack=2051564703 Win=32767 Len=0
 10 0.057415 localhost.localdomain localhost.localdomain TCP      32779 > 8999 [PSH, ACK] Seq=2051564703 Ack=2061215248 Win=32767 Len=54
 11 0.057423 localhost.localdomain localhost.localdomain TCP      8999 > 32779 [ACK] Seq=2061215248 Ack=2051564757 Win=32767 Len=0
 12 0.057523 localhost.localdomain localhost.localdomain TCP      32779 > 8999 [PSH, ACK] Seq=2051564757 Ack=2061215248 Win=32767 Len=44
 13 0.057532 localhost.localdomain localhost.localdomain TCP      8999 > 32779 [ACK] Seq=2061215248 Ack=2051564801 Win=32767 Len=0
 14 0.057630 localhost.localdomain localhost.localdomain TCP      32779 > 8999 [PSH, ACK] Seq=2051564801 Ack=2061215248 Win=32767 Len=45
 15 0.057637 localhost.localdomain localhost.localdomain TCP      8999 > 32779 [ACK] Seq=2061215248 Ack=2051564846 Win=32767 Len=0
 16 0.061723 localhost.localdomain localhost.localdomain TCP      32779 > 8999 [PSH, ACK] Seq=2051564846 Ack=2061215248 Win=32767 Len=52
 17 0.061731 localhost.localdomain localhost.localdomain TCP      8999 > 32779 [ACK] Seq=2061215248 Ack=2051564898 Win=32767 Len=0
```
Figure 4 : Ethereal Snapshot

```
Num pkts recvd   :  1000
Join delay       : 57989.328 sec
Recv pkt rate    :   100.017 pkt/sec
Recv data rate   :    38.254 kbps
Pkts dropped     :     0
```
Figure 5: Mcalc output

In the second example we show a TCP traffic generation with a constant IDT (which results in 100 packets per second) and a PS that follows a Poisson process with an average value of 48 bytes.
Figure 4 reports the snapshot of a traffic analyzer (Ethereal). This figure allows checking that D-ITG correctly creates TCP packets. The expected data rate is equal to 38 Kbps (48 bytes * 100 packets * 8).

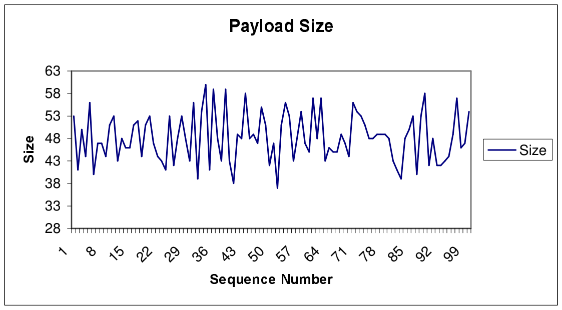


Figure 6 : Poisson packet size trend (D-ITG output)

This value is confirmed by the results shown in Figure 5, which reports the statistics of the D-ITG generation obtained using mcalc.
Furthermore, stemming from the theory, the deviation is equal to 48. Using the real generated values we have a deviation equal to 48.7 bytes and an average value equal to 47,761 bytes with a relative error respect to theoretical value equal to 1.4%. Figure 6 reports the dimension of the first 100 packets.
In the third example we show a TCP traffic generation with a constant IDT (which results in 100 packets per second) and a uniformly distributed PS between 200 and 400 bytes. The expected mean bit rate is 240 Kbps whereas the value achieved by D-ITG is 240.279 Kbps. Figure 7 reports D-ITG output whereas in the figure 8 the mcalc output is sketched.
In the fourth example we show the accuracy of D-ITG. Indeed we show how D-ITG is able to reproduce theoretical traffic model following the expected results. In particular using the results of Figure 10 we can carry out a comparison between a real traffic trace (Figure 9) and D-ITG simulated traffic.

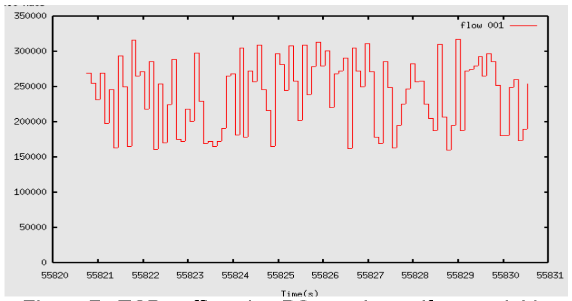
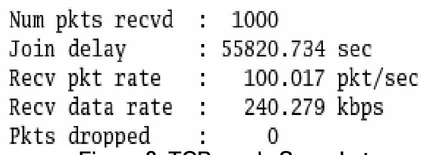


Figure 7 : TCP traffic using PS equal to uniform variable



```
Num pkts recvd   :  1000
Join delay       : 55820.734 sec
Recv pkt rate    :   100.017 pkt/sec
Recv data rate   :   240.279 kbps
Pkts dropped     :     0
```

Figure 8: TCP mcalc Snapshot

We compare a real video traffic - VBR (*Variable Bit Rate*) – having the following characteristics: the traffic trace is 2 hours long, there are 24 frames per second, frame dimensions follow a normal distribution with μ=27791 bytes e σ=6254 bytes [15]. In D-ITG we generate UDP traffic with one frame per packet, twelve minutes of traffic generation, IDT equal to 24 packets per second and finally PS equal to normal distribution with μ=27791 bytes e σ=6254 bytes. Comparing the trend in the two figures, we have the opportunity of checking the accuracy of D-ITG.

In the next example we show how D-ITG is able to generate application level traffic. In particular we show the generation of Telnet traffic and VoIP traffic. The Telnet traffic model considers a Pareto distributed IDT with shape β=0.90 and scale $\alpha \approx 1$, whereas PS follows the TCPLib [16]. In Figure 11 a two hours Telnet traffic trace is sketched and in figure 12 the cumulative probability function of Telnet packet size is shown. Furthermore in Figure 13 the TCPlib trend is reported. This figure reports the comparison among TCPLib distribution and two exponential trend lines: looking at this representation, we can show that the Telnet PS does not follow an exponential distribution. In particular there is a substantial difference especially for the first bytes.

As far as D-ITG Telnet traffic generation we consider a 10 seconds generation interval. In order to verify the PS distribution we analyze log files using the mcalc utility (Figure 14): the data rate is equal to 1.181 kbps (i.e. 147.6 byte per second).

In order to calculate the average value of packet dimension we calculate the ratio between the data rate and the packet rate (147.6 bytes per second divided per 74.248 packets per second): the result is 1.98 bytes per packet. In the theoretical model the bytes-per-packet average value is equal to 1 byte in the 62.8% of the total packets whereas it is equal to 2 bytes in the 12.1% of the same total: using D-ITG we can conclude that the theoretical value is respected. Finally, in figure 15 the D-ITG Telnet PS is reported.

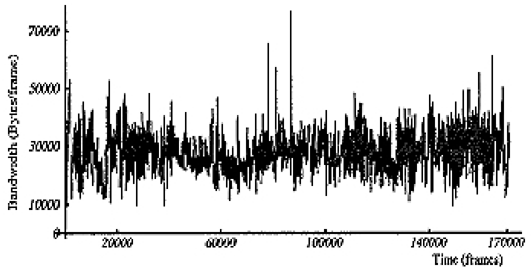


Figure 9 : UDP Video Traffic (real traffic)

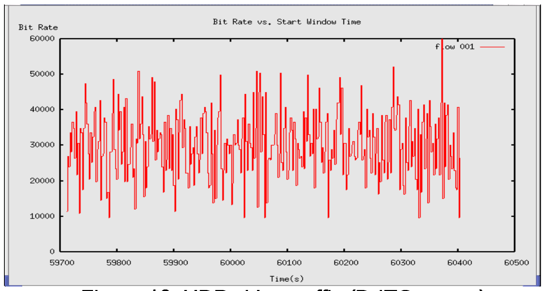


Figure 10: UDP video traffic (D-ITG output)

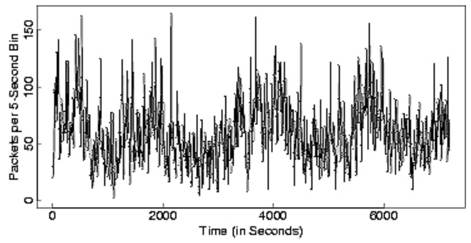


Figure 11 : Two hours of Telnet Traffic

Finally, in the last example we show how D-ITG is able to generate different sessions of VoIP traffic. These sessions are different in terms of codecs, in case of using VAD (*Voice Activity Detection*) in conjunction with header compression. Indeed, in the generation phase D-ITG is able to specify the used codec, number of sample per packet, VAD option and finally RTP (Real Time Protocol) header compression. In order to calculate the number of bytes of payload we use the relation Payload = Codec_Rate*Frame_Time*VAD where [Codec_Rate] = Kbps and [Frame_Time] = second, while VAD is a pure number. In our real implementation we use an average reduction of the payload equal to 35%.

Figure 16 reports the mcalc output related to a VoIP traffic generation with Codec G.711 with one voice sample per packet and without VAD. In this case in the theoretical model we have 100 packets per second and the payload equal to 80 bytes. Taking into the account 8 bytes of RTP header the expected data rate is 88*100*8 = 70.4 Kbps that is the same value of our traffic generation.

Figure 17 reports the mcalc output related to a VoIP traffic generation Codec G.711 with one voice sample per packet and with VAD. In this case, while in the theoretical model we have 100 packets per second and the payload equal to 80 bytes, using the VAD we observed a reduction equal to 35%. Taking again into the account 8 bytes of RTP header, the expected data rate is (80*0.65)*100*8 = 48 Kbps that is the same value of our traffic generation.

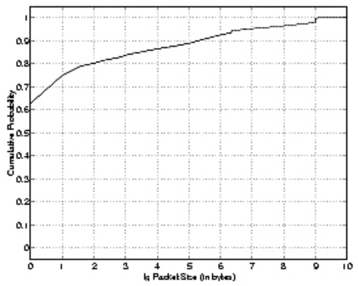


Figure 12: Telnet Packet Dimension: Cumulative Probability

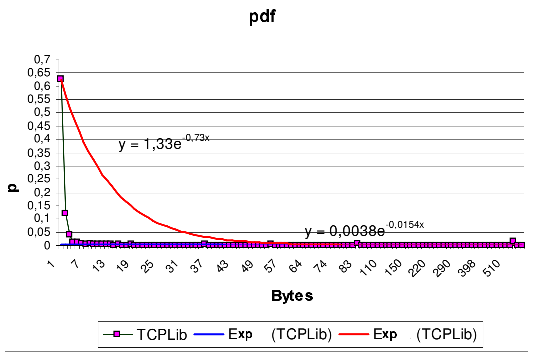


Figure 13 : TCPLib

```
Total packets received  :    740 pkts
Total recv packet rate  :     74.248 pkt/sec
Total recv data rate    :      1.181 kbps
Est. num pkts dropped   :      0 pkts
```

Figure 14: Mcalc Telnet Output

In figure 18 the mcalc output related to a simulation with Codec G.711, two voice samples per packet and without VAD is reported. In this case, the theoretical model envisions 50 packets per second and the payload equal to 80 bytes. With 8 bytes of RTP header and the two voice samples per packet, the expected data rate is (80*2+8)*50*8 = 67.2 Kbps that is the same value of our traffic generation.

In figure 19 the mcalc output related to a simulation with Codec G.729, two voice samples per packet and without VAD is reported. In this case in the theoretical model we have 50 packets per second and the payload equal to 10 bytes. Considering 8 bytes of RTP header and the two voice samples per packet, the expected data rate in this case is (10*2+8)*50*8 = 11.2 Kbps that is the same value of our traffic generation.

In figure 20 the mcalc output related to a simulation with Codec G.729, three voice samples per packet and without VAD is reported. In this case in the theoretical model we have 33 packets per second and the payload equal to 10 bytes. With 8 bytes of RTP header and the three voice samples per packet, the expected data rate is (10*3+8)*50*8 = 10.032 Kbps that is still the same value of our traffic generation.

In figure 21 the mcalc output related to a VoIP traffic generation with Codec G.723.1 with one voice sample per packet and without VAD is reported. In this case in the theoretical model we have 26 packets per second and the payload equal to 30 bytes. Taking into account 8 bytes of RTP header the expected data rate is (30+8)*26*8 = 7.9 Kbps that is the same value of our traffic generation.

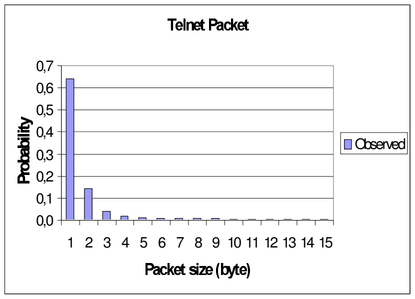


Figure 15 : D-ITG Telnet PS

```
PACKET RECEPTION STATISTICS
Total packets received   :   1000 pkts
Total recv packet rate   :     99.978 pkt/sec
Total recv data rate     :     70.455 kbps
Est. num pkts dropped    :      0 pkts
```

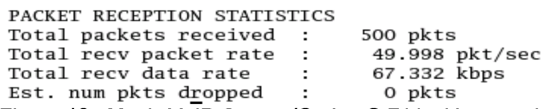
Figure 16 : Mcalc VoIP Output (Codec G.711 with one voice sample per packet and without VAD)

```
PACKET RECEPTION STATISTICS
Total packets received   :   1000 pkts
Total recv packet rate   :    100.017 pkt/sec
Total recv data rate     :     48.056 kbps
Est. num pkts dropped    :      0 pkts
```

Figure 17 : Mcalc VoIP Output (Codec G.711 with one voice sample per packet and with VAD)

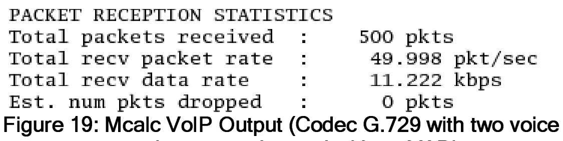
4 Conclusion and issues for research

After using other existing traffic generators in our network testing and network measurement operations, we experimented the lack of the necessary characteristics in a single traffic generator. Therefore we decided to implement our D-ITG. The basic idea of creating a new traffic generator arose from the lacks of existing ones with the possibility of simulating more complex traffic sources, repeating many times exactly the same traffic pattern (not only its mean values) and getting information not only about received packets but also about transmitted packets. With respect to other presented generators our D-ITG has been planned for generating network traffic (ICMP), transport layer traffic (TCP and UDP), several "layer 5-7" traffic (TELNET, SMTP, DNS, VoIP, …). In the case of TCP and UDP, the traffic pattern to be generated is defined by choosing a probability distribution (and its related parameters) for the IDT (*Inter Departure Times*) and PS (*Packet Size*) random processes. D-ITG makes available several statistical distributions (exponential, uniform, constant, pareto, cauchy, normal, …) both for IDT and PS random variables. In the case of ICMP and "layer 5-7" protocols, the distributions (and their related parameters) for IDT and PS are automatically defined according to theoretical models. Finally, we improved the generation performance with respect to other traffic generators. In this paper we show several examples of how to use our tool and an analysis of its performance. We focus on the generation phase in order to verify that the traffic requirements specified by the network administrator or by a generic user are met. Experimental results are compared with theoretical expected values.



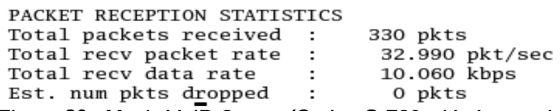
```
PACKET RECEPTION STATISTICS
Total packets received   :    500 pkts
Total recv packet rate   :     49.998 pkt/sec
Total recv data rate     :     67.332 kbps
Est. num pkts dropped    :       0 pkts
```

Figure 18 : Mcalc VoIP Output (Codec G.711 with two voice samples per packet and without VAD)



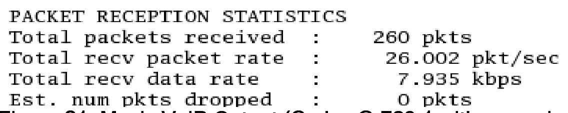
```
PACKET RECEPTION STATISTICS
Total packets received   :    500 pkts
Total recv packet rate   :     49.998 pkt/sec
Total recv data rate     :     11.222 kbps
Est. num pkts dropped    :       0 pkts
```

Figure 19: Mcalc VoIP Output (Codec G.729 with two voice samples per packet and without VAD)



```
PACKET RECEPTION STATISTICS
Total packets received   :    330 pkts
Total recv packet rate   :     32.990 pkt/sec
Total recv data rate     :     10.060 kbps
Est. num pkts dropped    :       0 pkts
```

Figure 20 : Mcalc VoIP Output (Codec G.729 with three voice samples per packet and without VAD)



```
PACKET RECEPTION STATISTICS
Total packets received   :    260 pkts
Total recv packet rate   :     26.002 pkt/sec
Total recv data rate     :      7.935 kbps
Est. num pkts dropped    :       0 pkts
```

Figure 21: Mcalc VoIP Output (Codec G.723.1 with one voice sample per packet and without VAD)

At present, we implemented a *centralized version* and we are working on *two kinds of distributed generators*. In the first distributed version there is a log server that is used by senders and receivers for data logging. In the second distributed version, processes of both senders and receivers have been implemented using MPI library. By separating generation and log processes, we eliminated the interference problem between them, which results in better overall performance. By eliminating interference problems the distributed version is able to replicate theoretical traffic figure imposed at sender side with greater accuracy [9]. Currently we are working on a "mobile version" of our D-ITG too. We are working on the porting of D-ITG under PDA platform with Linux FAMILIAR operating system. This implementation would make it possible to carry out a complete characterization of a real heterogeneous mobile network [8]. After this step we are planning to use D-ITG (with mobile extension) in a wide range of access network. Mobile Internet access using WLAN and GPRS/3G has gained good popularity. We would test D-ITG in a more large heterogeneous environment made by both heterogeneous (wired and wireless) network (WLAN, Bluetooth, UMTS, GPRS, …) and heterogeneous users' device (Laptop, PDA, Advanced Mobile Phone, PC,…). Using this implementation is possible to carry out a complete characterization of a real heterogeneous wired-wireless networks. Indeed, D-ITG enables performance evaluation of *heterogeneous devices* (Laptop, PC desktop, IPAH, …) over *heterogeneous networks* (Wired LAN, WLAN, …). D-ITG is currently downloadable and freely available at www.grid.unina.it/software/ITG.

Acknowledgements

This work has been carried out partially under the financial support of the "Ministero dell'Istruzione, dell'Università e della Ricerca (MIUR)" in the framework of the FIRB Project "Middleware for advanced services over large-scale, wired-wireless distributed systems (WEB-MINDS)".

References

[1] W. Willinger, and V. Paxson, "Where Mathematics meets the Internet", Notices of the American Mathematical Society, Vol.45, No.8, August 1998, pp. 961-970.
[2] V. Paxson. "Empirically derived analytic models of wide area TCP connections" IEEE/ACM Transactions on Networking, 2(4):316--336, August 1994.
[3] M. Zukerman, T. D. Neame and R. G. Addie, "Internet Traffic Modeling and Future Technology Implications", Infocom 2003
[4] M. E. Crovella, A. Bestavros, "Self-similarity in World Wide Web traffic: Evidence and possible causes", IEEE/ACM Trans. on Networking, Vol. 5, No. 6, Dec. 1997, pp. 835-846.
[5] A. Pescapè, S. Avallone, G. Ventre "Analysis and experimentation of Internet Traffic Generator", New2an'04, Next Generation Teletraffic and Wired/Wireless Advanced Networking, pp. 70-75 – ISBN 952-15-1132-X
[6] A. Pescapè, M. D'Arienzo, S. P. Romano, M. Esposito, S. Avallone, G. Ventre, ``Mtools" - IEEE Network, Software Tools for Networking 2002, Vol. 16 No. 5 pag. 3. ISSN 0890-80445
[7] A. Pescapè, S. Avallone, G. Ventre ``Distributed Internet Traffic Generator (D-ITG): analysis and experimentation over heterogeneous networks", accepted poster at ICNP 2003
[8] A. Pescapè, G. Iannello, G. Ventre, L. Vollero, "Experimental analysis of heterogeneous wireless networks", WWIC 2004, Wired/Wireless Internet Communications 2004, LNCS Vol. 2957 - pp. 153 - 164, ISBN: 3-540-20954-9
[9] A. Pescapè, D. Emma, G. Ventre, "Analysis and experimentation of an open distributed platform for synthetic traffic generation", 10th IEEE FTDCS 2004 - May 2004 - China
[10] http://www.atm.tut.fi/rude
[11] http://mgen.pf.itd.nrl.navy.mil
[12] http://dast.nlanr.net/Projects/Iperf/
[13] http://www.citi.umich.edu/projects/qbone/generator.html
[14] http://manimac.itd.nrl.navy.mil/MGEN/Utilities/
[15] W. Willinger and M.W.Garrett "Analysis, modeling, and generation of self similar VBR video traffic". SIGCOMM'94 Conference, pages 269--280, Sept. 1994.
[16] P. Danzig and S. Jamin, "tcplib: A Library of TCP Internetwork Traffic Characteristics", Report CS-SYS-91-01, Computer Science Department, University of Southern California, 1991.
[17] http://www.grid.unina.it/software/ITG