D-ITG Distributed Internet Traffic Generator

S. Avallone   S. Guadagno   D. Emma   A. Pescapè

G. Ventre

Università di Napoli Federico II

COMICS Lab, Dipartimento di Informatica e Sistemistica

Via Claudio 21, 80125, Napoli, Italy

{stavallo,pescape,ventre}@unina.it, {demma,salvatore.guadagno}@napoli.consorzio-cini.it

Abstract

In this paper we present our traffic generation platform, named D-ITG (Distributed Internet Traffic Generator). Its features are first described. We believe that some capabilities are in fact innovative, since no other traffic generator offers them. Then, we describe the original architecture of D-ITG, which allows our traffic generator to achieve high performance. We hint at a comparison with other traffic generators and conclude with the ongoing work to add new features.

1. Introduction

Distributed Internet Traffic Generator (D-ITG) is a platform capable to produce traffic that accurately adheres to patterns defined by the inter departure time between packets (IDT) and the packet size (PS) stochastic processes. Such processes are implemented as an i.i.d. sequence of random variables. A rich variety of probability distributions is available: constant, uniform, exponential, Pareto, Cauchy, normal, Poisson and gamma. Also, D-ITG embeds some models proposed to emulate sources of various protocols: TCP, UDP, ICMP, DNS, Telnet and VoIP (G.711, G.723, G.729, Voice Activity Detection, Compressed RTP). This means that the user simply chooses one of the supported protocols and the distribution of both IDT and PS will be automatically set.

D-ITG can perform both one-way-delay (OWD) and round-trip-time (RTT) measurement, packet loss evaluation, jitter and throughput measurement. For each generation experiment it is possible to set a seed for the random variables involved. This option gives the possibility to repeat many times exactly the same traffic pattern by using the same seed. Also, D-ITG permits the setting of TOS (DS) and TTL packet fields.

D-ITG allows to store information both on the receiver side and the sender side. It is thus possible to retrieve information on the traffic pattern generated. Additionally, D-ITG enables the sender and the receiver to delegate the logging operation to a remote log server. This option is useful when the sender or the receiver have limited storage capacity (e.g. PDAs, Palmtops, etc.). Also, it can be used to analyze log information "on-the-fly", for example, in case the sender is instructed by a controller entity to adapt the transmission rate based on channel congestion and receiver capacity.

Another innovative feature is that the sender can be remotely controlled by using ITGApi. This means that the D-ITG sender can be launched in daemon mode and wait for commands that instruct it to generate traffic flows.

D-ITG is currently available on Linux, Windows and Linux Familiar platform. It is freely available at *http://www.grid.unina.it/software/ITG/*.

2. D-ITG Software Architecture

D-ITG platform exhibits a distributed multi-component architecture. Figure 1 shows a graphical overview on the relationship among the main bricks of D-ITG platform. The communication between sender and receiver is done by using a separate signalling channel and ruled by a protocol for the configuration of the experiment (Traffic Specification Protocol - TSP). A short description of the three main components of D-ITG architecture follows.

ITGSend ITGSend is the sender component of the D-ITG traffic generation platform. ITGSend can operate in three different modes: (i) single flow mode: ITGSend generates just one flow; a single thread is responsible for the generation of the flow and the management of the signaling channel through the TSP protocol;(ii) multiple flows mode: ITGSend generates a set of flows. It operates as a multithreaded application: one of the threads implements the TSP protocol and drives
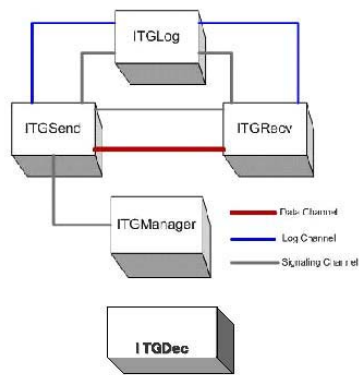
Figure 1. D-ITG Software Architecture

Figure 2. Traffic Generators, a comparative analysis

the generation process, while the others generate traffic flows; (iii) daemon mode: ITGSend is remotely controlled by ITGManager using the ITGApi. To collect statistics on the generation process ITGSend can log (either locally or remotely by using the log server ITGLog) detailed information about the generated flows.

ITGRecv ITGRecv always works as a concurrent daemon, listening for new TSP connections. When a TSP connection request arrives, ITGRecv generates a new thread that is responsible for the management of the communication with the sender. Each single flow is received by a separate thread. Like ITGSend, ITGRecv can store information either locally or remotely by using the log server ITGLog.

ITGLog ITGLog is a "log server", running on a different host than ITGSend and ITGRecv, which receives and stores the log information from multiple senders and receivers. The logging activities is handled using a signaling protocol. This protocol allows each sender/receiver to register on, and to leave, the log server. The log information can be sent using either a reliable channel (TCP) or an unreliable channel (UDP).

3. Performance Evaluation

In this section we present some experimental results. A more detailed analysis can be found in [7, 6]. We have carried out several experiments to compare D-ITG to other widely used traffic generators: Mtools [5], Rude/Crude [3], Mgen [2], Iperf [1] and UDPgenerator [4]. Figure 2 illustrates the results of the generation of an UDP flow characterized by a rate of 75000 pkt/s (each packet having a size of 1024 bytes) lasting for 60 seconds. Sender and receiver run on two Linux PCs directly connected through a Gigabit Ethernet. PCs have an Intel Pentium IV 2,6 GHz processor, 1 GB RAM and a 3Com Gigabit LOM (3c940) Ethernet con-
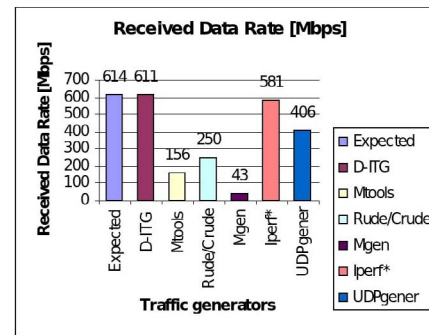
troller. D-ITG shows the best performance, as its received data rate is the closest to the expected one. It is also important to underline that Iperf works in a different way than D-ITG. Iperf does not produce a log file, it only provides an estimation of the received and transmitted date rate at the end of the experiment. A comprehensive list of traffic generators can be found at *www.grid.unina.it/software/ITG/link.html*.

4. Future and Ongoing Work

Currently our ongoing work is about IPv6 and Multicast support. We are also working on porting D-ITG on FreeBSD and Windows CE/Pocket PC platforms. Finally, we are working on a distributed implementation using the MPI (Message Passing Interface) paradigm. As far as the future work we are planning the possibility to remotely change transmission rate "on-the-fly" and the addition of models for other application layer protocols (SMTP, SNMP, HTTP, ...).

References

[1] Iperf. http://dast.nlanr.net/Projects/Iperf/.

[2] Mgen. http://mgen.pf.itd.nrl.navy.mil.

[3] Rudecrude. http://www.atm.tut.fi/rude.

[4] Udpgenerator. http://www.citi.umich.edu/projects/qbone/generator.html.

[5] S. Avallone, M. D'Arienzo, S. P. Romano, M. Esposito, A. Pescape', and G. Ventre. Mtools. *IEEE Network, Software Tools for Networking*, 16(15):3, 2002.

[6] S. Avallone, D. Emma, A. Pescape', and G. Ventre. A distributed multiplatform architecture for traffic generations. To appear in Proc. of International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS'04), July 2004.

[7] D. Emma, A. Pescape', and G. Ventre. Analysis and experimentation of an open distributed platform for synthetic traffic generation. In *Proc. of IEEE FTDCS*, May 2004.