A FLEXIBLE TRAFFIC GENERATOR FOR TESTING PABX AND CALL CENTER PERFORMANCE

Georg Rößler
Tenovis GmbH & Co. KG,
Kleyerstraße 94, D-60326 Frankfurt am Main, Germany, Georg.Roessler@tenovis.com

Thomas Steinert
Univ. of Stuttgart, Institute of Communication Networks and Computer Engineering,
Pfaffenwaldring 47, D-70569 Stuttgart, Germany, steinert@ind.uni-stuttgart.de

Abstract Testing PABXs and call centers under high load is of paramount importance because customers rely on these systems for safety and business critical applications. A traffic generator for high traffic volumes is presented in this paper. The two key characteristics of the traffic generator are the generation of traffic with configurable statistical properties and its flexibility in many aspects, ranging from the number and kind of emulated users to the user behaviour models and signalling protocols. This extraordinary flexibility could only be achieved by a pure software approach. The architecture clearly separates the basic runtime system and support functions which are implemented in the traffic generator core from user models and signalling protocols which the traffic generator reads and executes during tests.

Keywords: Environment simulation, ISDN-PABX, call center, performance testing

1. Motivation

Todays enterprise telecommunications infrastructure consists of interconnected systems ranging from networked PABXs (Private Automatic Branch Exchange) to server computers and client PCs. The applications running on these systems provide a large range of services to users, for example computer-supported telephony for increased productivity and convenience; call centers with sophisticated rule-based call distribution schemes; location tracking of medical personnel using mobile phones in hospitals for optimal dispatching of emergency calls. These examples illustrate the enormous increase of the system complexity compared to the stand-alone PABX system which once was the backbone of the enterprise telecommunications infrastructure.

Because of the safety and business critical applications, the systems must exhibit outstanding reliability. This reliability must also be guaranteed during high load or overload situations. Careful system design and performance analysis give confidence that a system can handle the traffic volume it has been designed for and that it performs well in overload phases. However, it is absolutely necessary to verify the expected performance through tests before it is deployed on customer sites.

In our context, the focus for load tests is on signalling traffic in the PABX as well as on the interaction between PABXs and servers. The main traffic sources are still in the telephone part of the system, hence the load generator aims at the emulation of telephone users attached via terminals or trunk lines. This paper describes a load generator which we call LGen. It is designed to fulfill two main requirements. The first requirement is that LGen can generate high traffic volumes with configurable statistical properties. The second requirement is flexibility such that it can easily be adapted and extended to varying and newly emerging test situations.

The next section explains the requirements to LGen in more detail. Then the system concept and its structure are described. The subsequent section presents example tests which have been carried out.

2. Requirements to the load generator LGen

The first requirement is that the generated traffic has to be statistically correct which means that call interarrival times, call holding times, etc. have to follow given distributions. Additionally, the generated traffic has to react correctly on varying system behaviour, e.g. repeat call attempts after call blocking due to overload. Further, LGen has to provide burst call arrivals as they appear in call centers when phone numbers are presented in TV commercials. LGen has to be parameterizable in order to simulate different load conditions. Both the distributions and their parameters must be configurable. Finally, the generated traffic has to reach an offered load of at least 30000 BHCA. This load is high enough to saturate a single module of the Tenovis Integral 33 PABX [1].

The desired flexibility implies that user behaviour and signalling protocols must not be hard-coded in LGen. Instead, this behaviour has to be provided by separate description files which LGen reads and executes.

Call patterns and timing behaviour have to be easily configurable. Otherwise, testers find it difficult to use the test system. Also, number and kind of emulated telephone users and trunk lines should be easily configurable. This permits LGen to adapt to different PABX configurations with little effort.

In a first step, fixed digital terminals for standard subscribers and for call center agents as well as digital trunk lines have to be supported. Later, mobile DECT (Digital Enhanced Cordless Telephone) subscribers have to be added. Both telephony and user movement which causes roaming and handover for DECT phones have to be emulated.

Statistics on events and durations have to be collected. These statistics are particularly important for long tests and tests sequences. Tracing of events and protocol data units (PDUs) with configurable filter conditions is required. The primary application area of LGen is in the laboratory during development and integration testing. It is an advantage, however, if LGen can also be applied for tests on customer systems, in particular for call centers (during off-peak hours).

3. Load test system

LGen is implemented in software running on standard workstations. The following section presents the general principle of performing load tests with LGen. Section 3.2 presents the realization of LGen.

3.1 General principle

Figure 1 shows a typical test configuration for testing the Integral 33 system from Tenovis. An Integral 33 module comprises a Generic Control Unit (GCU) on which runs the switching software, a switching fabric, and several line cards called Interface Control Units (ICUs).

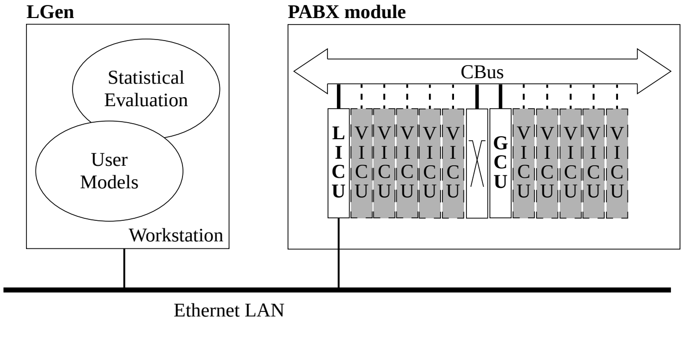

Figure 1. Typical test configuration of the load test system

In the PABX, the test focus is on the software running on the GCU. Usually, subscriber and trunk lines are connected to ICUs. They exchange signalling messages with the GCU via the CBus which is a proprietary communication system for message transfer.

For load tests with LGen, the real ICUs are removed and replaced by virtual ICUs (VICUs). The key characteristic of a VICU is that the GCU software receives the same messages as if a real ICU was present, although the ICU slot is empty. Note that it is possible to replace only some real ICUs by virtual ones and to keep some real ICUs in the module and to include them in the tests.

The so called LAN ICU (LICU) plays an important role for the representation of VICUs. The main task of the LICU is to forward signalling messages from the LAN to the CBus and vice versa. The LGen software generates the signalling messages - as if real users were present - and sends them through a TCP connection to the LICU. The LICU simply transmits the messages on the CBus. For CBus messages from the GCU, the LICU uses the same TCP connection to send them to LGen.

An LICU is capable to represent multiple VICUs at the same time. This is possible because the LICU can be configured to receive CBus messages for multiple destination addresses simultaneously. The source addresses for CBus messages are provided by the sender of messages.

The second task of the LICU is to relieve LGen from handling the commissioning phase for VICUs. A set of simple control messages between LGen and LICU has been defined for that purpose.

While in Figure 1 only one PABX module is depicted, it is possible to run tests with multiple modules of a PABX (or even multiple PABX systems) from a single LGen. It is sufficient to put an LICU in each module where VICUs are simulated to generate traffic.

3.2 Realization

This section presents the realization of LGen and, furthermore, shows the possibilities of this approach. For a detailed description of the structure and especially the traffic generation principle we refer to [2].

Figure 2 shows the building blocks of the software structure of LGen. The communication between them is based on LGen internal messages. This allows to decouple the execution of the different signal processing entities and, thus, to process the messages concurrently in multiple threads. The model applied is very similar to the “peer model” (also known as “workcrew model”). This principle allows to distribute the load to different threads, so that LGen adapts to the underlying hardware platform, e.g. a multiprocessor workstation, in an optimized way. Furthermore, all software modules of LGen are designed in that way, that no searching routine for e.g. the determination of the destination of a message takes place during a test. Only a calendar module performs a search routine, but its algorithms do not depend on the number of entries. Therefore, LGen scales not only with the underlying hardware platform but also with the number of emulated users.

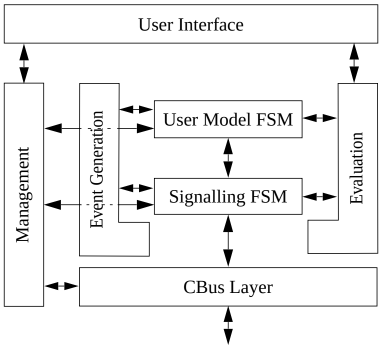

Figure 2. Software structure of LGen

The traffic generation principle of LGen is based on the emulation of the behaviour of single users by so called "User Models". The superposition of the signalling traffic of a multitude of these User Models represents a realistic traffic mixture. For describing a User Model an FSM approach was chosen, where the functional behaviour of an emulated user is separated from its temporal behaviour. The functional behaviour is defined by the **User Model FSM** and determines what an emulated user does, e.g. how he reacts on a refused connection establishment. The temporal behaviour is defined by means of **Event Generators** and determines, when a user does something, e.g. when he will try again to establish a connection. This allows to use one User Model FSM for different temporal behaviours e.g. to emulate a standard user and a business user with the same User Model FSM only with different behaviour concerning the time. There are quite a number of publications about augmented FSMs describing both functional and temporal behaviour, e.g. [3] gives a nice overview. Except a paper by Lemppenau and Tran-Gia [4] describing also a system for traffic generation most of the publications focus on performance analysis and not on the emulation of behaviour in real-time.

A User Model FSM controls a **Signalling FSM** which realizes layer 3 of a signalling system by means of Service Data Units (SDUs). SDUs sent by the User Model FSM trigger actions of the Signalling FSM, those received by the User Model FSM indicate the current state of the Signalling FSM. By this approach the User Model FSM creates an abstract view of the underlying signalling protocol.

To generate traffic a User Model has to perform spontaneous actions. Therefore the principle of Event Generators was introduced: Event Generators generate input signals for User Model FSMs which we call Events. The interarrival times of these Events are statistically distributed according to adjustable distribution density functions. Upon reception of an Event a User Model FSM performs a state transition where further signalling actions like sending an SDU to trigger the setup of a signalling connection take place. Figure 3 shows the relation between User Model FSM, Event Generator and Signalling FSM.

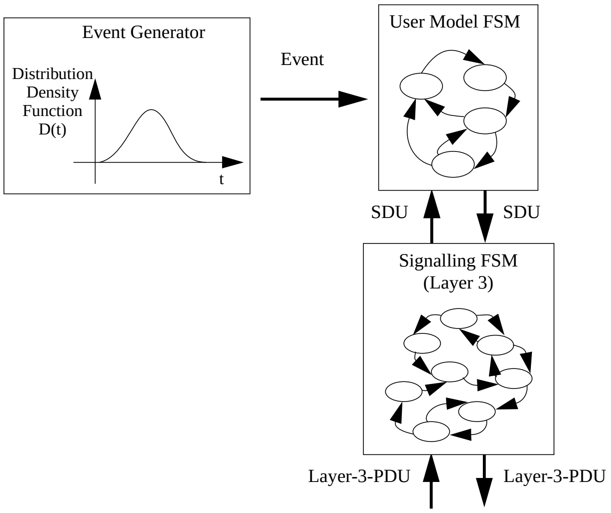

Figure 3. Relation between User Model FSM, Event Generator and Signalling FSM

For the **Evaluation** of a load test measurement variables concerning signalling connections, like e.g. “Setup Answer Delay” or “Number of Accepted Setup”, are collected. Furthermore, it is possible to record a message trace of the exchanged signalling messages, which allows to examine them in detail.

For the specification of a User Model FSM a subset of the Specification and Description Language (SDL) defined by the ITU-T is applied. In the configuration phase LGen creates a data structure corresponding to the specified User Model FSM relating its SDL statements to implemented and compiled C functions. During a load test the data structure is interpreted and the corresponding functions are executed. This principle combines the flexible method for specifying a communication process by means of SDL with the capabilities of a programming language and the performance of a speed optimized program.

The same principle is also applied for Signalling FSMs which are realized according to the respective standards. Only the interface to the upper layer, the User Model FSM, is adapted for the needs of LGen by communicating via protocol independent SDUs. Signalling FSMs send layer 3 PDUs as a reaction on a time out, an SDU or a PDU. At the **CBus Layer** these PDUs are integrated into CBus Messages and sent to an LICU of a PABX module. In the other direction layer 3 PDUs which are extracted from CBus Messages received by the CBus Layer are sent to the corresponding Signalling FSM.

This approach allows to integrate several signalling protocols into LGen. So far the ISDN-D-channel protocols according to ITU-T Q.931 and to TN-1R6 which is a Tenovis specific proprietary advancement of the german ISDN-D-channel protocol 1TR6 and a signalling protocol for DECT are realized. Therefore, LGen is capable to emulate various user types as standard users at ISDN terminals, ACD (Automatic Call Distribution) users at call centers, digital trunk lines and DECT mobile users. Furthermore, for each user type multiple temporal configurations can be applied in a load test.

An important requirement of LGen is the generation of traffic bursts. This is already integrated in the standard way of defining the traffic characteristics: A group of emulated users which will be activated at the same time has to be configured in that way that it generates a call within a short time frame after activation. This results in a signalling traffic peak after the activation of this group of users.

4. Example Tests

A first series of load tests concentrated on the PABX. The expected switching performance and the system stability under high load and overload could be confirmed. A selection of these tests is repeated with each new software release. The fact that LGen can produce high load and keep it constant gave the opportunity to validate the system dimensioning rules as they are implemented in a configuration tool.

In an ideal system, each call contributes a fixed amount of CPU occupation. In real systems, however, operations like searching for a particular call context to which a signalling message belongs may take longer when many calls are active at the same time. A series of tests demonstrated a linear relation between the offered traffic and the observed CPU load of the GCU. Thus the real system can be regarded as an ideal system in this respect.

The next step was towards call centers and other CTI-based systems. Testing these systems is the main application area of LGen. The focus for these tests is on the interaction between servers and the PABX and on the applications running on dedicated servers. The call distribution application is a key element of a call center. The call distribution schemes are implemented in customer specific call flow scripts which the application interprets. In the more sophisticated call flow scripts, the resulting system load usually increases faster than the offered traffic. This phenomenon is mainly caused by additional activities, for example automatic announcements about possibilities to leave a voice message or about the expected waiting time, for callers who have to wait before they can be connected to an agent. An important question in this case is which load a call center can handle, in particular if the offered load exceeds the customer expectations such that many callers have to wait.

In a further step, LGen was extended for emulation of DECT mobile users. Emulating DECT users for load tests is interesting because the high user density together with the narrow cells results in potentially high roaming and handover rates. Inside buildings, base stations are usually installed only a few tens of meters apart on each floor. Moreover, in DECT the phones send roaming requests when they detect that another base station is more suitable for them, whereas in public cellular mobile networks like GSM it is usually the network that triggers the roaming. With LGen scenarios like many DECT users leaving a meeting at the same time and thus creating many roaming events could be tested. The special caching mechanisms which have been implemented to cope with mass roaming situations have been found to work well.

5. Conclusion

LGen has proven to be a valuable tool for testing enterprise telecommunications systems. Call rates of more than 100000 BHCA can be generated on a Sun UltraSPARC 1 workstation. The high performance is due to the system design for efficient execution of user and signalling models. This illustrates that the flexibility has been achieved without negative impact on the performance.

References

[1] http://www.tenovis.com/tenovis_international/Details/Integral33.htm

[2] T. Steinert, G. Rößler: Generation of realistic signalling traffic in an ISDN load test system using SDL User Models, Proceedings of FORTE/PSTV 2000, Pisa

[3] Mitschele-Thiel, Müller-Clostermann: Performance engineering of SDL/MSC systems, Computer Networks 31 (1999)

[4] Lemppenau, Tran-Gia: A Universal Environment Simulator for SPC Switching System Testing, Proc. 11th International Teletraffic Congress (ITC), Kyoto 1985