CHAPTER 14

Design and Performance Evaluation of a Packet-Switching Satellite Emulator

Tomaso de Cola [1], Mario Marchese [2], Giancarlo Portomauro[1]

[1] CNIT - Italian National Consortium for TelecommunicationsGenoa Research Unit University of Genoa, Via Opera Pia 13, 16145, Genoa, Italy
[2] DIST - Department of Communication, Computer and System Science University of Genoa Via Opera Pia 13, 16145, Genoa, Italy

Abstract. *The increasing interest for the satellite technology as suitable means for transporting multimedia applications has fostered several research investigations aimed at identifying proper algorithm tunings and exploring the performance of communication protocols in general. To this end, the support of analysis and research tools is of primary importance to assess performance of data communication and, hence, to provide the satellite system designers with proper indications suited to develop effective satellite networks.*

From this standpoint, a satellite emulation tool is presented, showing the features it provides and a large set of applications in which it might be employed to asses the performance of data communication.

1. INTRODUCTION

Satellite technology has many advantages with respect to terrestrial one in terms of area coverage, high availability of bandwidth and intrinsic multicast capability. On the other hand, it presents critical aspects such as high delay of propagation and non-negligible packet loss, which have to be taken into account in the design of satellite architecture. As a consequence, within this framework, there is the need of developing new instruments and schemes to improve the efficiency of the communications along with the testing of the proposed solutions.

A satellite system may be hardly studied on the field. It is expensive and it often concerns only software components for earth stations. Alternatives are necessary to investigate new systems and to evaluate the performance. The first one is the analytical study. It is very attractive but complex; moreover it often requires simplifications and approximations. The second alternative is the simulation of the system behaviour via software, in order to by-pass the complexity of real systems and to test solutions not yet technically feasible, in view of future evolutions. In this case, there is the need of modelling effort, even if a model, often, is not accurate enough to consider all the aspects of a real system.

A third alternative, which seems to summarize most of the advantages of the solutions mentioned, is emulation, which is composed of hardware and software components that behave as a real system. An emulator allows using real traffic and it is similar to the real system also from the point of view of the physical architecture.

The paper presents the design and implementation of an emulator concerning on-board satellite packet switching. This proposal takes as reference the emulation system developed within a wider Project called "Emulation of on-board satellite switching systems" (ACE - ASI CNIT Emulator) [1], and extends its main functionalities, in order to have a powerful instrument for performance analysis of data communication achieved in satellite networks [2].

The paper is structured as follows. The need for a real-time simulator is discussed in Section 2 together with a presentation of the related works that can be found in the literature. Section 3 shows the proposal concerning the architecture and specifies the modules, embedded within the emulation core, needed to perform the satellite channel emulation together with the real-time issue and the transport of information. In Section 4, the performance analysis and the response of such system in presence of bulk transfer and multimedia traffic are given, in order to show the robustness and the capabilities of the proposed emulation system. Finally, Section 5 contains the conclusions.

2. BACKGROUND AND RELATED WORKS

Due to network complexity, simulation plays a vital role in attempting to characterize both the behaviours of the current Internet and the possible effects of proposed changes to its operations [3]. However, simulation and analysis are restricted to exploring a constructed, abstract model of the world while measurement and experimentation provide an effective instrument for exploring the real world. In this perspective emulator allows creating network topologies, conditions and to perform real-time tests with various prototype protocols and products.

NIST Net is implemented [4] as a kernel module extension for the Linux operating system and an X Window System-based user interface application. It provides parameters such as delay, packet drop probability, fraction of duplicated packets and link bandwidth. Dummynet [5], implemented as a FreeBSD UNIX kernel patch in the IP stack, works by intercepting packets on their way through the protocol stack. The NRL Mobile Network Emulator (MNE) performs real-time simulations in wireless environment by extending network capabilities implemented on the IP stack in Linux kernel and exploiting IPTABLES capabilities [6]. EGPRS system [7] allows emulation of GPRS environments by means of a virtual network device, which allows the definition of a proper radio link control operating at the MAC level. Finally, EMPOWER system [8] distributes the emulation core in several nodes and performs the real-time simulation by means of virtual devices modules linked to the network stack.

Likewise EGPRS and EMPOWER approaches, our proposal (ACE) is based on the employment of network virtual devices, which allow implementing specific MAC functionalities running on the top of the network device driver. In the following, the description of the proposed emulation architecture is presented together with the presentation of the software modules necessary to interface the emulated environment with the rest of the architecture.

3. EMULATION SYSTEM

A satellite system is constituted by a certain number of ground stations (each composed of a satellite modem that acts both at the physical and at the data link layer) and a satellite that communicates with the ground station over the satellite channel. The emulator should allow testing various kinds of protocols of TCP/IP suite and switching systems, in order to evaluate suitable solutions to be adopted. In the following subsections, particular attention is devoted to the description of the ACE architecture and then to the implementation of the emulation core in terms of software modules and interaction between "emulated" and "real" world.

3.1 Overall Architecture

In a real satellite system, it is possible to identify the following main parts. The satellite channel, characterized by its own peculiarities (e.g. bandwidth, propagation delay, bit error rate, etc.) connects the ground stations with the satellite platforms, supporting on-board switching capabilities [9]. On the other hand, it is worth noting the role played by the satellite modem, mounted within the ground stations, which provides an interface towards the upper layers (namely the network layer) and performs operation of framing, access to the medium, and channel coding for example, through the data link protocol.

In the ACE system, two main units are required to reproduce effectively the behaviour of a real system, as shown in Figure 1:

- Emulator Unit (EU): it has a powerful elaboration capacity, carries out most of the emulation, as the decisions about each PDU (i.e. the loss, delay and any statistics of each PDU regards the EU).
- Gateway (GTW): it operates as interface among the Emulator Unit (EU) and the external PCs, and it is responsible of managing the specific functionalities of a satellite modem in the real system. For this purpose, the interface between the modem at the ground station and the protocols of the upper network layers has been implemented in the ACE system by creating a virtual device.

The aforementioned units communicate by means of control packets, carrying notifications about the emulation. The data transportation is achieved exploiting the capabilities of TAP device [10], provided by Linux kernel, which allows the emulator to behave exactly as a broadcast link that connects more stations. Concerning the traffic traversing the emulation system and managed by its modules, we have:

- the real traffic, (the information coming from the external PCs) transported from a GTW (input GTW) to the other (output GTW) and then forwarded to the upper layer (or discarded) through notification from the EU.

- The control traffic, exchanged among the EU and the GTWs, carrying the notifications generated by the emulation core, i.e. delivery instant, packet corruption and so on.

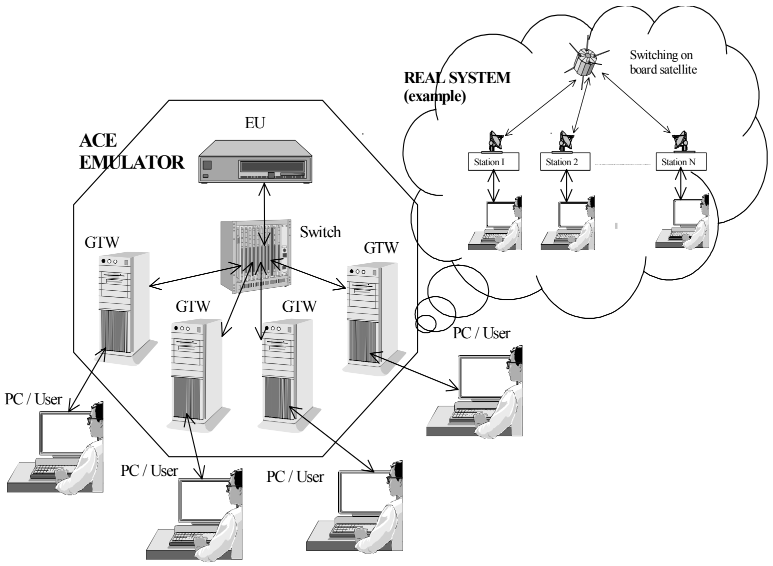


Fig. 1 Overall Emulator Architecture and Real System

3.2 Software Modules

The software implementation of the emulation core is composed of several modules, which are responsible of the real-time simulation as well as the generation of the control traffic and the data communication among the different GTWs. The emulation core is organized in two levels, as depicted in Figure 2:

- Low Level (LL): it is responsible for providing the functions needed to the communication and the interaction among the physical units (GTWs and EU).
- High Level (HL): it is responsible for the emulation processing. It allows the emulation of the main transmission characteristics (e.g. bit error ratio, channel fading, loss and packet delay), the implementation of various media access control protocols and the on-board switching architecture.

In order to make the architecture modular and flexible to further evolutions, an Object Oriented Programming paradigm has been considered. A particular attention has to be reserved to the main operations performed by the High Level, namely the emulation of the data communication and the behaviour of the satellite channel.

Concerning the former aspect, as indicated in Figure 2 and Figure 3, two objects play a fundamental role, the Station Provider and the Station. When a new station needs to register itself to the emulation system, a notification is communicated (by *setStationProvider()*) to the Station Provider, which will be responsible of creating a new Station object. Once the operation has completed, the Station defines an L3interface in order to have an interface towards the network layer and starts receiving/transmitting PDUs by means of the methods *put_pdu()* and *fetch_pdu()*.

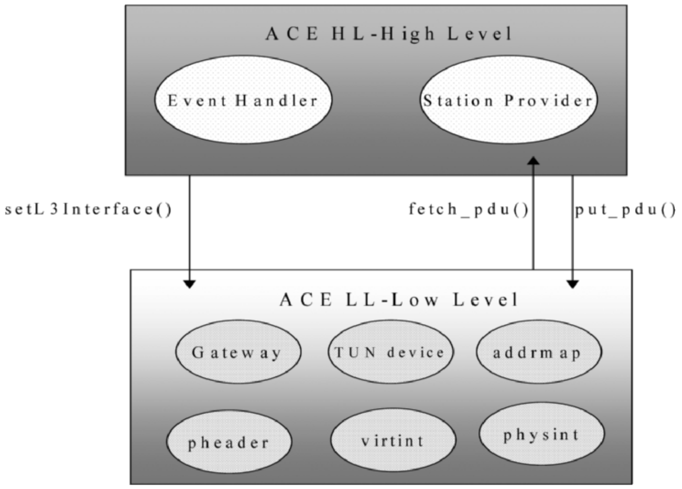


Fig. 2 Software architecture of the ACE system

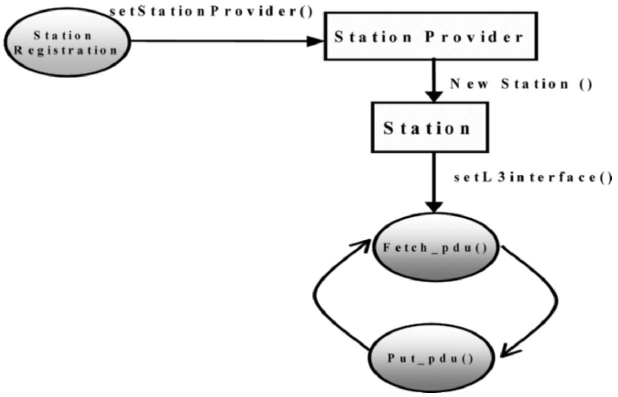


Fig. 3 Objects involved in the data communication

Concerning the emulation of the satellite channel, a coupled interaction between the objects characterising the station and the satellite channel itself is required.

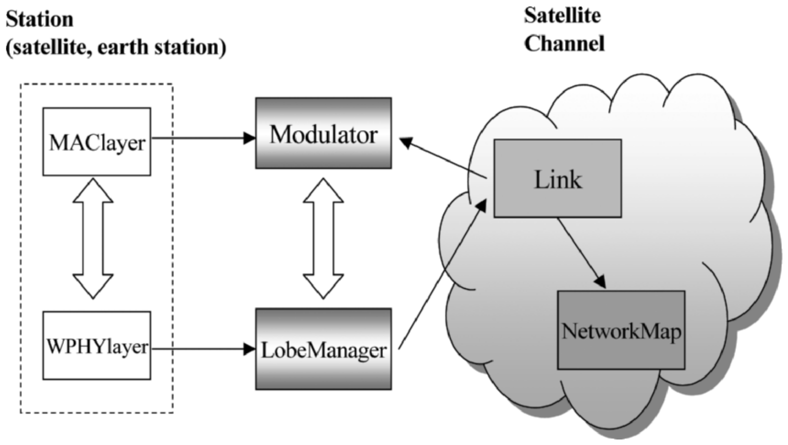


Fig. 4 Objects involved in the channel emulation

On one hand, as indicated in Figure 4, the station components involved in the emulation are the *MAClayer* (MAC Layer) and the *WPHYlayer* (Physical Layer) and on the other hand, the elements necessary to the characterisation of the channel are the objects *Link* and *NetworkMap*, which maintain the information about the network topology and the physical link peculiarities. The communication between the channel and the station is realised by means of the *Modulator* object, which performs also operations of access to the medium, encoding and digital modulation, and of the *LobeManager* that effectively manages the satellite channel.

4. VALIDATION TESTS

4.1 Measure of Gateway Delivery Accuracy

This measure is dedicated to test the accuracy of the developed code in delivering the PDUs to the upper network layer at a given time instant. The difference between the instant of the real delivery and the scheduled instant is the performance index in order to evaluate the delay in PDU delivery operation to the upper layers. The tests have been composed as follows:

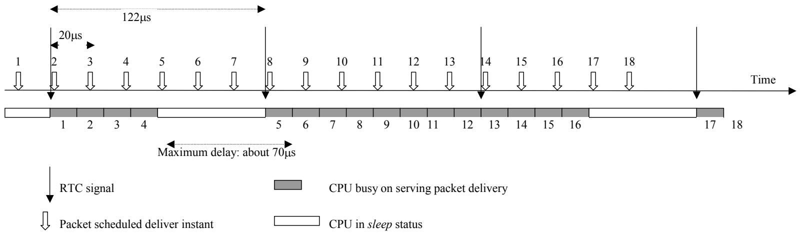
- The overall number of packets delivered in each test is 1000;
- The dimension of the packets delivered to the upper network layer is fixed at 1500 bytes (the Ethernet MTU);

- The scheduled time interval between the delivery of two consecutive packets is not time variant within the same test, especially, the following values have been tested: 20ms, 200 ms, 2 ms, 20 ms;
- Two different values of RTC have been used: a frequency of 4096 Hz and a frequency of 8192 Hz (the maximum frequency settable on the RTC).

A frequency of 4096 Hz corresponds to an intervention of the CPU (Central Processor Unit), which performs the real delivery, approximately each 244 ms, while a frequency of 8192 Hz each 122 ms. So, if the system works as expected, the delay (the difference) between the instant of the real delivery operation and the scheduled instant should range from 0 to 122 ms (or 244ms). The mean value of the delay should be about 60 ms. A delay constantly much higher than the upper bound means that the system is saturated and the real time behaviour cannot be controlled. Table I contains the mean value of the mentioned delay for the different scheduled delivery interval and RTC. Actually, the results obtained respect the expected values. Only the case apparently more critical (20 ms) behaves better than expected. The behaviour is explained in Figure 5 and in Figure 6, where the events occurring in the emulator for the 20 ms and the 200 ms case, respectively, are reported over the time. In the first case, when the CPU time needed to process and deliver a packet is over, it is probable there is another packet scheduled to be served before the CPU returns to the sleep status, waiting for the next RTC signal. On the contrary, in the second case, when the service time is over, the process returns immediately to the sleep status because there is no packet scheduled to be sent. The overall effect, in the first case, is a drastic reduction of the delay value. Figure 7 contains the mentioned delay versus time in the case of a RTC frequency of 8192 Hz and a scheduled delivery of 20 ms. The figure allows checking the real behaviour over time and to verify that not only the mean value is under the upper bound but that each single packet has been delivered with a low delay. Similar considerations may be done concerning the other tests (200 ms, 2 ms and 20 ms), not reported. It is important to observe that the results provided by adopting a RTC frequency of 4096 Hz are also satisfying when they are compared with the mean delay of satellite systems. The advantage of a lower RTC frequency is a CPU less loaded. Anyway, the measures about the CPU load performed during the tests have shown an acceptable load level even in the case of a RTC frequency of 8192 Hz.

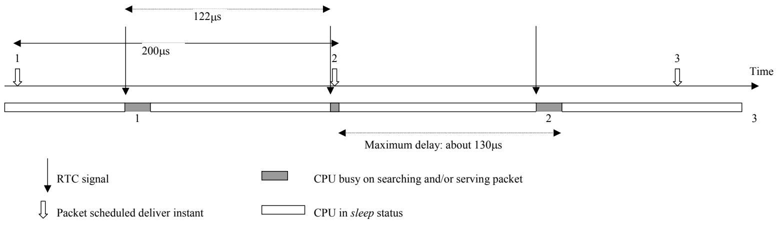
Table 1 Mean value of the delay

Delivery interval	Mean delivery delay [μs]	
	4096 Hz	**8192 Hz**
20 μs	57.656	25.559
200 μs	112.840	70.024
2 ms	130.460	69.850
20 ms	129.870	70.284



122μs
20μs
Time
Maximum delay: about 70μs
RTC signal
Packet scheduled deliver instant
CPU busy on serving packet delivery
CPU in *sleep* status

Fig. 5 Emulator events - 20 ms



122μs
200μs
Time
Maximum delay: about 130μs
RTC signal
Packet scheduled deliver instant
CPU busy on searching and/or serving packet
CPU in *sleep* status

Fig. 6 Emulator events - 200 ms

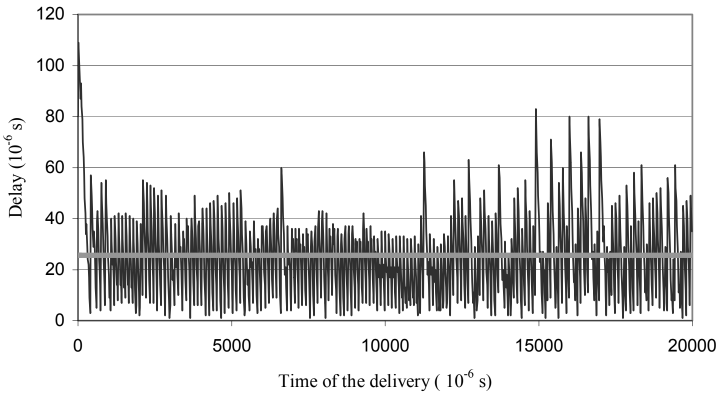


Fig. 7 Delay vs. time, 8192 Hz, 20 ms

4.2 Performance Limits of the Emulation System

In order to identify the possible scenarios and applications, in which it would be possible to employ such a tool as a research instrument, the system effectiveness has been checked also by varying the emulated link bandwidth and propagation delay. In practice, the aim of this validation has been to see which are the emulation limits of the system and hence to draw the edges of this emulation framework. In more detail, different propagation values have been considered in order to reproduce common satellite scenarios including Low Earth Orbit (LEO), Medium Earth Orbit (MEO) and Geostationary Earth Orbiting (GEO) satellite platform, each giving rise to delays of 5, 100 and 260 ms, respectively. Moreover, for the sake of the completeness, also the case of propagation delay set to 0 s has been taken under consideration in order to evaluate the responsivity of the system when the link propagation delay can be assumed negligible as in cabled networks or in some wireless configurations (e.g. wireless sensor networks and short-range WLANs). In correspondence to these propagation delay settings, also several emulated bandwidth values have been considered. In particular, tests have been performed when the link bandwidth ranged from 64 kbit/s up to 26 Mbit/s. The interest has been mainly directed to this performance interval because, when a satellite connection has to be shared among different users its use is effective from a financial point view if the available bandwidth is comparable with that of ISDN and other cabled systems. moreover, the limit of 26 Mbit/s has been set during the trial campaign. Actually, the collected measures have shown that the system robustness is completely independent of the specific propagation delay set. On the contrary, the impact of the emulated link bandwidth plays a more important role and can negatively affect the overall data communication. In practice, when the emulated bandwidth ranges between 64 kbit/s and 24 Mbit/s, the system is able to reproduce the given scenario with a percentage deviation lower than 0.5%. Once the bandwidth gets over 24 Mbit/s (e.g. 26 Mbit/s) the emulation system is no longer able to match the physical channel settings, most likely because of the physical hardware limits and the time constraints imposed by the emulation procedures during the phases of PDU fetching and delivering to the upper layers (namely IP). In more detail, this behaviour can be explained and clarified by taking account the inner network architecture and the fetching and delivering operations performed at the ACE layer. In particular, the delay introduced by these operations may affect negatively the whole emulation system. Finally, it is expected that using Real Time systems should improve the effectiveness of the system, allowing to extend the bandwidth interval (i.e. over 26 Mbit/s).

In Table 2, the most significant results are summarised. For the sake of clarity, the first column represents the reference bandwidth values, while the second one shows the values actually set on the emulation system.

Table 2 Emulated Link Bandwidth Performance

	Emulation System		Measurements
Reference Bandwidth [Kbit/s]	Bandwidth [Kbit/s]	Propagation Delay. [ms]	Bandwidth [Kbit/s]
64	65.28	0	64
		5	64
		100	64
		260	64
512	522.24	0	512
		5	512
		100	512
		260	512
1000	1020.00	0	1000
		5	1000
		100	1000
		260	1000
4000	4080.00	0	4000
		5	4000
		100	4000
		260	4000
10000	10200.00	0	9970
		5	9960
		100	9990
		260	9990
16000	16320.00	0	15900
		5	15900
		100	15700
		260	15900
20000	20400.00	0	19800
		5	19800
		100	19800
		260	19800
24000	24480.00	0	24000
		5	24000
		100	23900
		260	23900
26000	26520.00	0	Failed Test
		5	Failed Test
		100	Failed Test
		260	Failed Test

The slight difference between the values reported in the two columns is due to the fact that the measures have been performed at the application layer, while bandwidth setting refer to the datalink layer; consequently, the overhead introduced by layers operating between application and datalink layers has to be accounted for.

4.3 Comparison with a Real System

In this section, a particular emphasis is given to the comparison of the ACE behaviour with a real satellite system.

The real system taken as reference employs the satellite ITALSAT II (13° EST). It provides coverage in the single spot-beam on Ka-band (20-30 GHz), providing an overall bandwidth of 36 MHz. The satellite station has a full-duplex channel with a bit-rate of 2 Mbit/s.

In order to test the effectiveness and the response of the proposed emulation system, the performance analysis has been accomplished considering two different patterns of data traffic.

On one hand, we have taken as reference bulk transfers, generated by FTP-like applications, when different versions of TCP are mounted on the PCs, in order to see the emulator response, in presence of aggressive TCP flows, with respect to the real system behaviour. On the other hand, we have considered the case in which multimedia traffic mixed with FTP transfers traverses the system, in order to show the effectiveness of ACE when TCP and UDP connections are multiplexed together on the same satellite channel.

The application used to get the result is a simple ftp-like one, which allows transferring data between the two remote sites. In order to extend the effectiveness of our analysis we have considered modified [11] NewReno TCP versions [12]. In more detail, the robustness of the system in presence of a slow-start phase more aggressive (as indicated in Table 3) has been tested to evaluate the emulator response. In this perspective, we have taken as main parameter the transmission instant for each data packet released to the network and the tests are achieved by performing a data transfer of about 3 Mbytes.

The testbed employed in this case is composed of two PCs communicating through the ACE system configured with two gateways (GTW1 and GTW2 respectively in Figure 8).

Table 3 Different TCP Implementations Tested

TCP version	Initial Window (IW)	TCP buffer (bytes)	Congestion Window Increase (k)
IW2-64Kbytes	2	64K	1
IW4-256Kbytes	4	256K	1
IW6-k2-256Kbytes	6	256K	2
IW6-k8-64Kbytes	6	64K	8
IW6-k4-320Kbytes	6	320K	4

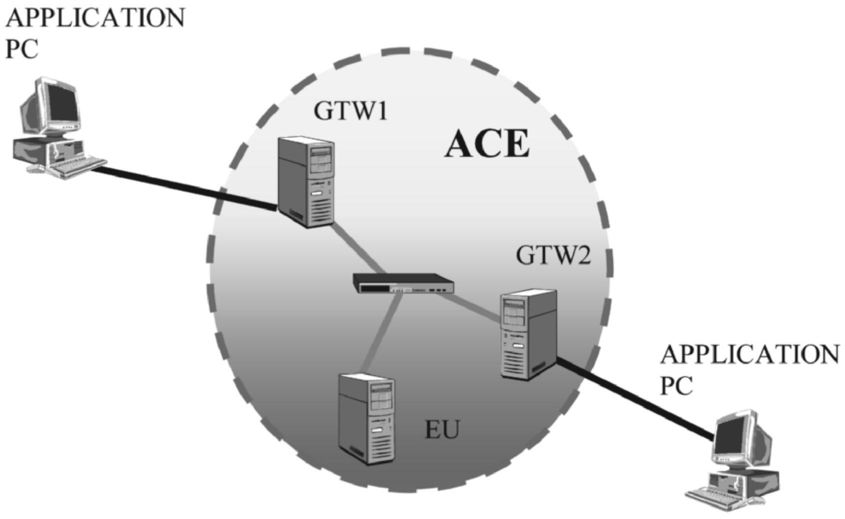


Fig. 8 Configuration of ACE emulation system

In the first instance, we may consider the case of standard TCP employment, when an initial transmission window (IW) equal to 2 segments and a TCP Receiver Buffer of 64 Kbytes are assumed (IW2-64Kbytes in Table I). The comparison between ACE and a real system is depicted in Figure 6 where, for the sake of simplicity, only the first 144 packets are reported. It can be seen that the difference between the two cases is really minimal: the curves, describing how the packet transmission instant changes during the data transfer, just overlap.

As a second instance, a more aggressive implementation of TCP is considered in order to investigate the response of proposed emulation system in presence of high burst of data. For this purpose a TCP buffer of 320Kbytes with IW equal to 6 packets and congestion window increase k equal to 4 is assumed (IW6-k4-320Kbytes in Table 3).

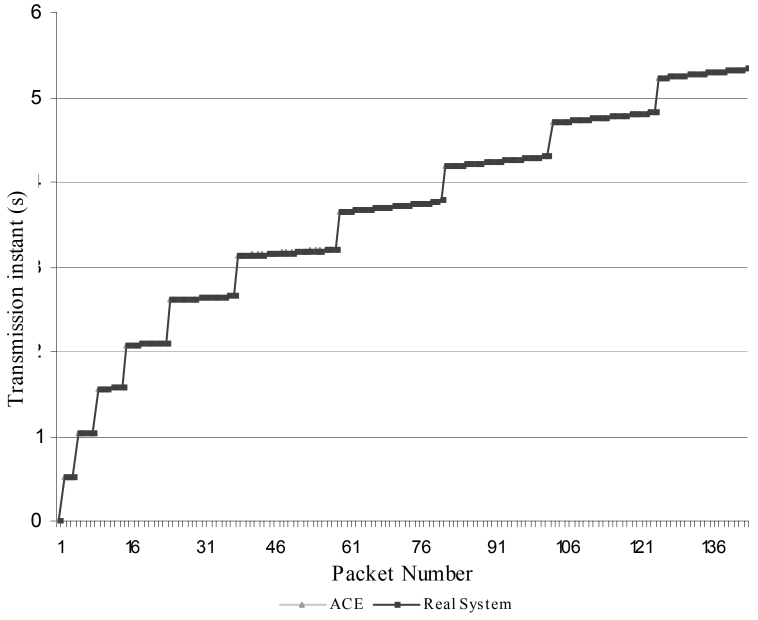


Fig. 9 ACE vs. real satellite system: TCP with IW=2 and receive buffer=64Kbytes

Also in this test (shown in Figure 10), we can point out that the emulator behaviour simply overlap the one registered in a real satellite system. This further test thus verifies emulator capabilities of reflecting the overall performance of real systems and following the behaviour of different TCP implementations. It is also important to highlight that ACE system respects real time constraints and then the time dedicated to the emulation elaboration does not affect the overall performances since it does not involve a hard delay in the data packets processing.

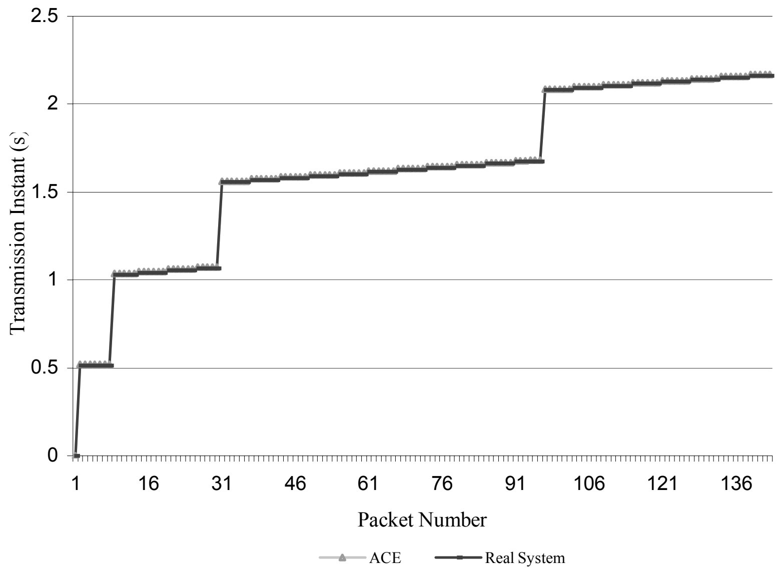


Fig. 10 ACE vs. real satellite system: TCP with IW=6, k=4 and receive buffer=320Kbytes

For the sake of simplicity, we omit the analysis of other TCP versions (i.e. IW4-256Kbytes, IW6-k2-256Kbytes and IW6-k8-64Kbytes in Table 1), which lead to similar conclusions.

The next step is represented by the comparison of the performed tests in order to show the accuracy of the emulator with respect to the real behaviour. In particular, we consider as a main parameter the average value of the difference ξ between the transmission instants in the emulation and in the real cases, defined as:

$$\xi = \frac{T_{real} - T_{emul}}{T_{real}}$$

where T_{real} and T_{emul} represent the transmission instants respectively in the cases of the emulation and of the real behaviour (Table 4).

Table 4 Difference Between Emulated And Real Behaviours

TCP implementations	ξ (%)
IW2-64Kbytes	$9.7 \cdot 10^{-3}$
IW4-256Kbytes	0.01
IW6-k2-256Kbytes	0.03
IW6-k8-64Kbytes	0.05
IW6-k4-320Kbytes	0.09

Table 4 shows that, even in presence of more aggressive TCP implementations (e.g. IW6-k8-64Kbytes and IW6-k4-320Kbytes), the percentage difference between the real and the emulated behaviours is very low and it is numerically below 0.1%.

Finally, in order to evaluate the full operability of the system, a more complex scenario has been considered. The aim was to check the effectiveness of ACE when several applications shared the emulator. Furthermore, this test allowed studying the response of emulation system when different protocols were operating at transport and network layers. In more detail, we have taken as reference UDP/TCP based applications such as videoconference and transfer file processes and an ICMP packet traffic generated by a PING application. Summarizing, the following applications have been employed:

- FTP (File Transfer Protocol);
- SDR (Session Directory Tool) for multicast conferences on the Mbone;
- Ping.

The general architecture employed is depicted in Figure 11.

Four PCs are connected to gateways belonging to the emulation system; a further PC within the emulator is responsible of the emulation functionalities. The processes considered in the test-bed are resident on the PCs shown in the figure. It is also important to say some more words about the application employed. The three processes act at the same time and are considered to analyse how the whole system behaves when different real data communications are performed. In more detail, FTP is considered in order to evaluate the emulator performance when a big transfer of data ruled by the TCP protocol is accomplished. SDR allows showing the ACE behaviour when a real – time service is required. PING is employed to verify that a correct RTT value, proper of geostationary link, is experienced.

A snapshot of the overall behaviour is presented in Figure 12, where different windows show the applications considered.

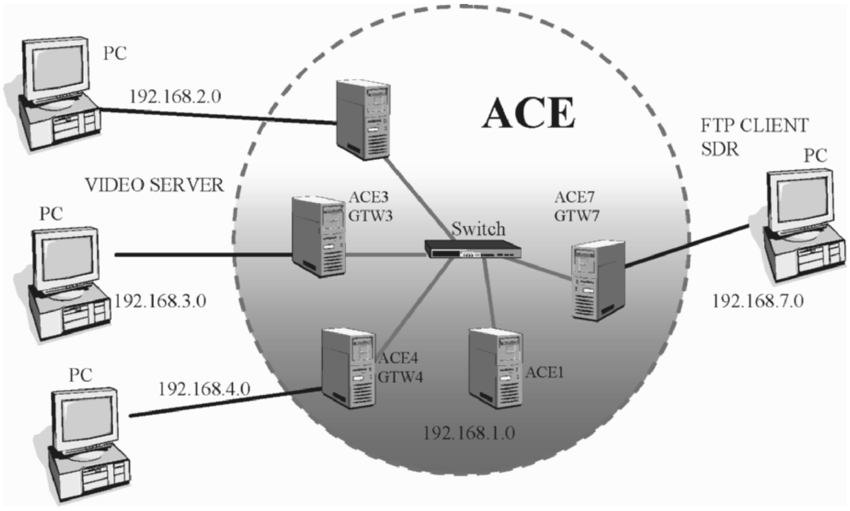


Fig. 11 Test-bed general architecture

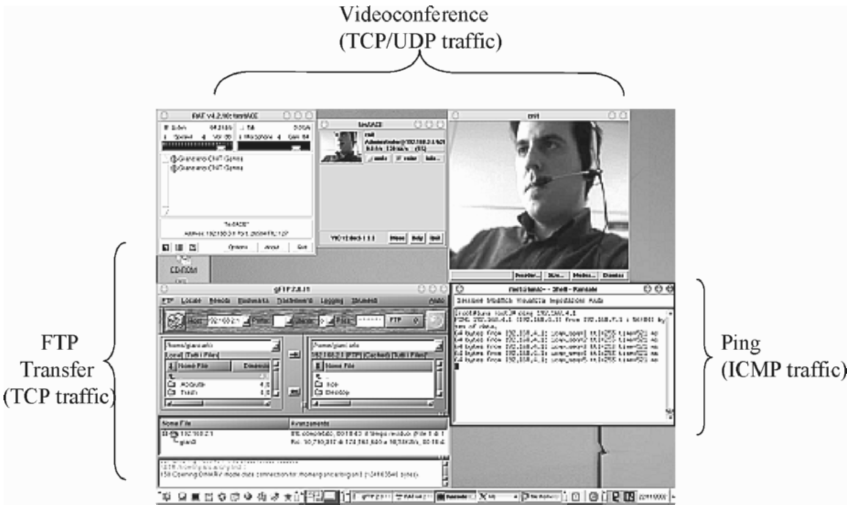


Fig. 12 Traffic patterns involved in the test

The windows on the left side describe the FTP session, while the ones on the top reproduce a webcam application (employed by the videoconference tool). The RTP–based real–time service (videoconference application) and the TCP-based communication are accomplished at the same time. The tests show that emulation system is able to deal with different traffic patterns without introducing extra delays during the processing operations. As a further validation, the PING window, on the right side, shows that the system performs as in the correct Geostationary satellite RTT, set to about 520 ms, even if multiple flows enter in the emulation system.

5. CONCLUSIONS

The analysed framework puts emphasis on the necessity of emulation instruments for the study of data communication performed in satellite networks. For this purpose, the paper proposes an emulation system for packet-switching satellite systems, able to reproduce fairly well the conditions in which real communications are usually performed. The ACE emulator is composed of units called Gateways (GTW) and the Elaboration Unit (EU), which has a powerful elaboration capacity and carries out most of the emulation. A particular benefit has also been optained by the adoption of the Object Oriented paradigm in the implementation of the emulation core. It has allowed a flexible definition of a real-time environment that may be extended to different satellite scenarios, simply by acting on the objects resident in the High Level implementation, without care of how the transmission is physically performed.

The results have been used to test the accuracy of the emulation in order to show the effectiveness of such an instrument when traffic of different flavours, i.e. bulk transfer and multimedia communication is analysed. In the former case (i.e. bulk transfer), reported results have highlighted that the emulation system is able to reflect very fairly the real behaviour even when aggressive TCP flows are entering in the system. In particular, we have registered a maximum relative error with respect to real behaviour of about 0.1%. In the latter case, emulation tool has shown its capability of supporting different traffic flows (TCP mixed with UDP), ratifying the robustness and the flexibility of the proposed system, even if a hard load is driven throughout the satellite network under investigation.

REFERENCES

[1] G. Albertengo, T. Pecorella, M. Marchese, "The ACE project: a Real Time Simulator for Satellite Telecommunication Systems," Proc. Sixth Ka-Band Utilization Conference, Cleveland, OH, Jun. 2000, pp. 571-576.

[2] M. Allman, D. Glover, L. Sanchez, "Enhancing TCP Over Satellite Channels using Standard Mechanism," IETF, RFC 2488, Jan. 1999.

[3] Sally Floyd, Vern Paxson, "Difficulties in simulating the internet," IEEE/ACM Transactions on Networking, no. 4, pp. 392-403, Aug. 2001.

[4] NIST Interworking Technology Group. NIST Net network emulation package. http://snad.ncsl.nist.gov/ itg/nistnet/

[5] L. Rizzo, "Dummynet: a simple approach to the evaluation of network protocols," ACM Computer Communication Review, no. 1, vol. 27, pp. 31-41, Jan. 1997.

[6] Naval Research Laboratory January 24, 2003 NRL/FR/5523–03-10,054 NRL Mobile Network Emulator, Washington, DC 20375-5320.

[7] X. Qiu, K. Chawla, L. F. Chang, J. Chuang, N. Sollenberger, J. Whitehead, "An enhanced RLC/MAC design for supporting integrated services over EGPRS", IEEE WCNC 2000, no. 1, Chicago, IL, Sep. 2000, pp. 907-912.

[8] P. Zheng, L. Ni , "EMPOWER: A Network Emulator for Wireless and Wireline Networks," IEEE INFOCOM 2003, no.1, San Francisco, CA, Mar. 2003, pp.1933-1942.

[9] M. Marchese, M. Perrando, "A packet-switching satellite emulator: A proposal about architecture and implementation", IEEE ICC 2002, no. 1, New York City, NY, Apr. 2002, pp. 3033-3037.

[10] Universal TUN/TAP Device Driver http://vtun.sourceforge.net/tun/ index.html.

[11] M. Marchese, "TCP Modification over Satellite Channels: Study and Performance Evaluation", International Journal of Satellite Communications, Special Issue on IP, Vol. 19, Issue 1, pp. 93-110, Jan./Feb. 2001.

[12] S. Floyd, T. Henderson, A. Gurtov, "The NewReno Modification to TCP's Fast Recovery Algorithm", IETF RFC 3782, Apr. 2004.