LaasNetExp: a generic polymorphic platform for network emulation and experiments

Philippe Owezarski, Pascal Berthou, Yann Labit, David Gauchard

LAAS-CNRS
University of Toulouse
7 Avenue du Colonel Roche - 31077 Toulouse cedex 4, France
{owe, berthou, ylabit, gauchard}@laas.fr

Abstract—**Network experiments are essential for assessing and validating new networking technologies, architectures and protocols. These assessments have long been performed using network simulators. But it clearly appeared that the results got in simulations cannot be reproduced in real environment. Emulators can hardly integrate accurate models of all networking components, end host operating systems and applications what leads to unrealistic simulations very often. Therefore, some work has been issued for developing real experiment platform and network emulators. This paper addresses the motivations that raised the design and development of such an experimental platform at LAAS – laasnetexp – and describes its constituting features and components. It is in particular detailed how experimental conditions can be fully controlled for reproducible and easy to analyze experiments. Last, this paper describes how realistic conditions can be set-up in experiments by using the results of actual Internet and Internet traffic characterization, analysis and modeling. Such information helps to realistically configure emulators as well as define realistic traffic generators. The realism of such experiments is illustrated as a demonstration of the interest of laasnetexp for networking research.**

Keywords- experiments in real environment, emulation, realistic experiments, reproducible and controlled experiments, traffic generator

I. INTRODUCTION

Research in networking cannot be dissociated from experiments, be they performed by simulation, by emulating the network or by mean of experiments in real environments. Such experiments are essential to provide a priori evaluation of a network under design. They help evaluating whether such a network can work, ie. if it is free of bugs, checking its liveness, its integrity, and also making some evaluations of its performances.

With simulations, all parts of the distributed system (i.e. the interconnected networks and all their components as well as the end hosts) are based on models [2] whose realism can always be discussed. Then, when the Internet is considered, its simulation is not an easy task [3], especially because of its size (number of users and equipments), its complexity (number of protocols for instance), the behavior of its users, and of course, its fast evolution in terms of technology and usages. Because of all these features, it is easy to understand why it is so

difficult to get some realistic simulations of the Internet, or at least of some small parts of it. One of the limits of simulators then comes from their inability to be scalable, in particular because it would require workstations with extremely high computing performances and memory capacities that can very hardly be reached nowadays.

Even if simulation is still being the most used network assessment tool in network research and engineering (in particular because of its low price in terms of equipment and human investment – see Figure 1), it is commonly observed that the results are largely inaccurate. The differences between simulations results and the ones got in real environments are generally significant. Thus, experimental platforms in real environments (as Planetlab) are now raising a lot of efforts (Figure 1 & 2). In such platform, of course, all components, all operating systems, all protocols, etc. are real. The main problem with such real experiments comes from the lack of control researchers have on the experimental conditions: it is impossible to control the background traffic flowing in the networks which then makes difficult to analyze the results gained. For instance, it is almost impossible in a large experimental network as Planetlab (which is not fully monitored) to analyze the causes of a performance decrease for a given new network protocol under evaluation. It can be due to the protocol, but also a significant event which changed the experimental conditions punctually, but which is not logged. This main lack of the experiments in real environment is related to the non reproducible experimental conditions. Therefore, even if real experiments are a fantastic tool for finalizing the assessment of a network right before its setting-up, researchers needs a more easily configurable tool (and cheaper) for being able to change artificially network experimental conditions.

That is why the solution which seems to be the most suited for performing realistic and exploitable experiments is certainly network emulation. In network emulation, all end host operating systems, end to end protocols and applications are real. But the network, which is the main concerned topic in this community, is simulated. This, in particular, allows researchers to control the experimental conditions from realistic to extreme ones, thus allowing the identification of network limits. Several emulation levels can exist: an emulator can emulate a full network (coarse aggregation level) or simply one router or network device (fine aggregation level).

It is then possible to address scalability issues with emulators, as well as mixing fine and coarse emulation: for instance, when assessing the performances of a particular AS, a fine emulation can be used for close ASes, while a coarse emulation can be applied for ASes far from the AS under evaluation.

Figures 1 and 2 represent the different tool families for assessing network protocols, architectures and mechanisms. In particular, they represent the cost of the experiment tools according to the level of realism expected. Figure 1 more specifically presents the four kinds of experimental tools (this first section listed already three; the fourth is the one based on mathematical model which is the most theoretical and is often the first step of any research). Figure 2 gives examples of each of these experimental tool kinds.
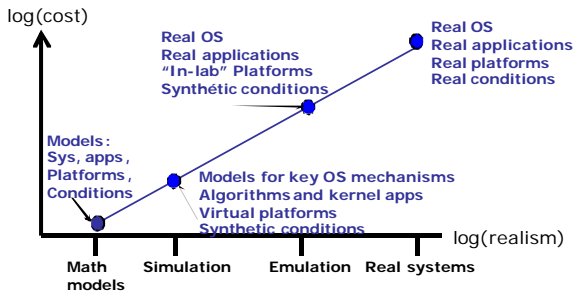

Figure1.　Virtualization, realism and cost of the different assessment tools in network research

LAAS is a French laboratory which is leading researches in networking. It then has to perform experiments for demonstrating, assessing and evaluating the performances of the network architectures, protocols and mechanisms it proposes. In particular, the main research domain in networking under consideration is related to multi-domain QoS. The main example of such multi-domains network is of course the Internet. In such framework, two main objectives related with QoS are lead: the first deals with enforcing QoS guarantees. For this purpose, solutions based on network virtualization, horizontal and vertical signaling protocols as well as strict traffic control are under progress. On the other side, LAAS also addresses researches aiming at optimizing the quality of service in the current best effort Internet. It involves essentially the design of auto-adaptative protocols and architectures based on real time traffic and QoS monitoring. In particular, it has been demonstrated that the most difficult cases for guaranteeing as well as optimizing QoS are related to the presence of traffic anomalies which then have to be detected and classified in real time. This implies that our architectures and protocols proposals need to be assessed in the presence of anomalies. The generation of controlled anomalies in parallel of controlled background traffic is then one of the requirements of our experimental platform. In addition, we are also spending a lot of efforts on access networks which appear to be one of the main sources of QoS degradation. According to the large variety of different access technologies available in the Internet, we then lead researches in low network layers (layer 2 essentially). In particular, we are spending a lot of efforts on satellite access networks.
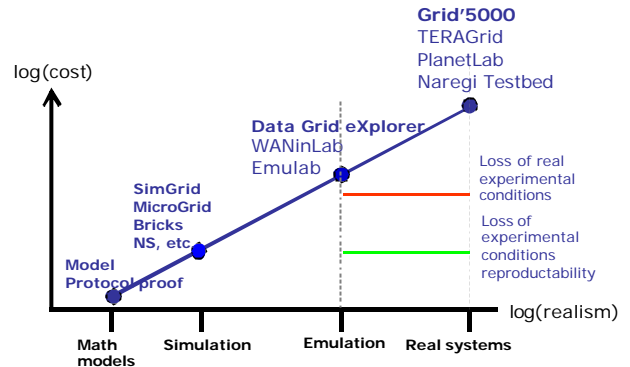

Figure 2.　Examples of experimental tools and platform for the four different kinds

In this framework, an experimental platform has been designed and installed at LAAS for allowing researchers and engineers to achieve their experiments. The objective of this platform is to be as generic as possible and to be able to take several forms (polymorphism). This platform has then been designed in order to be suited for network emulation, as well as experiments in real environments[1]. Of course, in both cases, the experimental conditions are as reproducible as possible (even in real experiments). One of the requirements for this platform was to be usable simultaneously for several different experiments, be they emulated or real. This platform is called laasnetexp and stands for "LAAS Network Experiments".

The remainder of the paper is as follows: section 2 expresses into details the experimental platform requirements. Section 3 describes how the platform has been designed and built, and how it is managed. Section 4 presents how it is used when performing experiments. It also presents some results validating the results obtained with this platform. A particular example of how the interconnection of a satellite network with terrestrial access networks can be emulated on laasnetexp is shown. Finally, section 5 concludes the paper.

II. EXPERIMENTATION NEEDS

The preceding section, which introduces the different existing experimentation tools, also introduces as a side effect some of the expected features of a platform made for running emulated or real network experiments on top of it. This section aims at listing and describing exhaustively all necessary features of the experimentation platform for leading our researches on the enforcement of multi-domains QOS. Four

---

[1] Of course, this platform contains machines on which simulators can be run. This platform then integrates the full range of experimentation tools.

different kinds of such properties required by the experiment platform are described in the following:

- EXPERIMENTAL END TO END CONTROL

  The first requirement for leading research experiments is to have a full control on the experimental conditions. This implies:

  - To control the background traffic load and properties, as well as the load of all machines involved in the experiments, be they playing the role of an end host or emulating a network part. Such total control on the experimental conditions makes experiments fully *reproducible*, and the results can then be analyzed with a full knowledge of all events arising during the experiment.

  - To be able to (re-)configure dynamically and automatically the experimental platform. This will help to pass from one experiment to another seamlessly. This is also especially true when several experiments are run together on the same platform. They must not interfere with each other, despite they share some resources. As they have no reason to be synchronized, a change on the experimental conditions for one experiment must not change the experimental conditions of the other experiment.

- MEASURED AND MONITORED PLATFORM

  The platform must also provide measurement and monitoring information on the network and its traffic all along the experiments. Of course, all assessment, validation, estimation and analysis can only be made based on this information. However, the measurement and monitoring facilities must be:

  - Fully transparent for the network, its traffic and the load of the machines involved in the experiments. Indeed, the measurement and monitoring facilities must have a zero impact on the experimentation running.

  - Very reliable, i.e. it has to provide measurement and log files or traces without missing any event or packet.

  - Very accurate, i.e. it has to provide measurement and log files or traces with very accurate timestamps, the timestamps being essential for all performances related evaluations and analysis, for instance.

  - Able to store and analyze large files, and this without disturbing the running experiments.

- INTEGRATION WITH OTHER PLATFORMS

  One of the major issues for emulating the multi-domain Internet – or at least part of it – is related to the scale of the experiments. Emulators allows the modeling of a whole domain (eventually several domains also) by just one machine. This helps for emulating large networks. However, for experiments in real environments, the only way of having scaling experiments deals with involving more machines. For this reason, it is necessary for this platform to be able to interconnect with other experimental platforms in the world, but keeping in mind that the experimental conditions must be controlled and monitored to make experiments reproducible and analysis possible.

- ISOLATION FROM REAL WORLD OR BETWEEN RUNNING EXPERIMENTS

  One of the requirements for controlling experimental conditions forces us to isolate our platform from the outside network, its traffic and more specifically its anomalies which are not of our own. In addition, our platform represents a large amount of computing and networking resources, and then could be the target of corruption attempts from hackers. Isolating it enforces a certain security level, which of course has to be reinforced by other security mechanisms (filtering essentially as we know in advance what the different features of the accepted incoming and outgoing flows are).

III. PLATFORM DESCRIPTION, SETTINGS AND MANAGEMENT

Figure 3 depicts the current laasnetexp platform. For fulfilling all these requirements, laasnetexp has been completely separated from the LAAS' operational network in order our experiments not to be disturbed by the uncontrolled traffic of our colleagues, and not to disturb their work with our stressing experiments. In addition, laasnetexp is one level higher than the LAAS' operational network in the Internet domains hierarchy. Indeed, LAAS' operational network accesses to the REMIP'2000 network which is a level 2 switched Ethernet network. In order to take advantage of the IP services provided by most of the European NRN (National Research Network) and GEANT, we then connected laasnetexp to RENATER, the French NRN. We will see later what the IP services of fundamental importance are for laasnetexp.

Nowadays, laasnetexp consists of a server and 38 experiment machines[2] (this number should be increased shortly) running different operating systems and having four Ethernet interfaces. Indeed, for being able to serve for emulations and real network experiments, two networks have been created in laasnetexp: a 3-domains real network (suited for multi-domain experiments) with public IP addresses belonging to three different networks, and an emulation network. Each machine then has two Ethernet interfaces associated with public addresses (in two different domains – see Figure 3) and two Ethernet interfaces associated with two private addresses in the emulation network.

A. The real experiment network

Each domain in this experiment platform is supported by a CISCO Catalyst 4948-10GE Ethernet switch. Domains are interconnected by dual ports Juniper M7i routers: aneto, posets and montperdu. Aneto, which is our front router, has a particular role: it cannot be disconnected from the RENATER network to keep routes from and towards our platform always

---

[2] These machines are racked Dell PowerEdge860 PC, with dual core Xeon 2,13 GHz processor, 2 GBytes of RAM and 600GBytes of hard drives.

alive. It is then not possible to set-up risky experimental conditions on this router. Posets and montperdu are more likely to be used for such kinds of risky experiments.

Getting into details for the real experimental network, this section will justify the choice made for laasnetexp in order to fulfill the experimentation requirements.

REPRODUCIBLE EXPERIMENTS BY MEANS OF CONTROLLED TRAFFIC AND MACHINE LOADS

First, laasnetexp is completely separated from the LAAS' operational network. It then does not contain any other traffic than the one specifically generated for the experiments. In addition, this property can also be reinforced by making Aneto – the first laasnetexp router in contact with the Internet – filter all incoming flows which are not involved in the experiment. The traffic is then completely controlled, and with a strict planning of experiments running at the same time, it is possible to know which machines are involved. In case of performance sensitive experiments, we then avoid having a machine involved in two experiments, for keeping a control on the machine loads, and then making possible to reproduce any of the experiments.

MEASUREMENT AND MONITORING

The monitoring and measurement system deployed in laasnetexp relies on the DAG card from ENDACE [1]. The first advantage of this card is that it is installed right after an optical or electrical splitter which lets 80 % of the signal power on the original path, and gets 20 % of this power for the DAG card. Then, the traffic cannot be impacted by the monitoring equipment, no extra delay is introduced by the splitter, and the traffic keeps its characteristics and profile. The measurement system is then completely *transparent*.

The DAG card then extracts in real time packets headers or the full packets passing on the link. For each packet recorded, the DAG card adds a 64 bits GPS timestamp. The packet (or its header) and the timestamp are then stored in a file on the local hard drive. Because of the speed of actual networks, the traffic between the DAG card and the hard drive is very high. On a Gigabit links, it imposes to use extended PCI buses, for instance the 64 bits wide and running at a frequency of at least 66 MHz. With such configuration of the DAG system[3], it is ensured that no packets will be missed: the system is then highly *reliable*. In addition, the GPS allows global clock synchronization with accuracy less than 2 μs. The timestamping system is then very *accurate*. It also represents a universal time and all DAG machines synchronized on the same GPS principle are then all synchronized between each other.

Finally, a SAN (Storage Area Network) of 4.2 TBytes and five computing servers[4] (have been installed on the LAAS' operational network for the analysis of the traces collected. Of course, the traces are downloaded from the DAG machine to the analysis servers when no experiment is running.

Note also that a DAG system has been installed on the LAAS' operational network. It will provide us with realistic traffic traces. These traces will then allow us to extract realistic network QoS and traffic parameters values which will be useful for setting-up realistic network behaviors and realistic traffics for our experiments.

INTEGRATION WITH OTHER PLATFORMS

As already said, integrating several local experimental platforms would allow larger scales experiments. However, it is very important to keep the previous properties apply to an interconnection of platforms, i.e. the background traffic and experimental conditions must be controlled for allowing reproducible experiments. For this reason, interconnections between laasnetexp and other equivalent platforms (in the EuQoS project for instance) take advantage of the PIP service (Premium IP) supported by GEANT and most of the European NRN. PIP allows the creation of a tunnel between two platforms, and it is guaranteed that the packets sent on this tunnel will be routed with the highest priority. Background traffic then cannot compete with the traffic in the tunnel. The experiment platform can then always take advantage of the "reserved" bandwidth in the tunnel, and the jitter on those packets will then be quite limited. Therefore, PIP makes users believe that there is never any traffic on the tunnel, and makes the traffic conditions of a dedicated experiment perfectly reproducible.

In such framework, all complexity of NRNs and GEANT interconnections is hidden and possible QoS disturbance (almost) impossible. All this complex interconnection of network can then be seen as a single domain. The multi-domain structure of the experimental network is then the one of laasnetexp, which is completely under control, for fully controlled and reproducible experiments.

ISOLATION FROM REAL WORLD OR BETWEEN RUNNING EXPERIMENTS

It has been seen above that for isolating laasnetexp from the uncontrolled Internet traffic, its experimental network is completely separated from the LAAS' operational network. All traffic on laasnetexp is then related to the running experiment. For preventing any risk of intrusion, the Aneto router filters all incoming connections which are not due to the running experiment.

If several experiments are run at the same time, but must not compete for getting the requested shared resources two strategies apply together: first, no machine can be involved in several experiments. Each machine is completely assigned to a single experiment. Second, dealing with the network resources, VLAN are used for creating static different routes for the flows of the different experiments. This is also possible because of the largely overprovisioned capacities of the laasnetexp routers and switches.

MANAGEMENT

All management functionalities for the laasnetexp public network are centered on a single server (aran) in charge of

---

[3] DAG cards are installed in Dell PowerEdge 1950, with a single Xeon 1,6 GHz processor, 2 GBytes of RAM, and 1 TBytes of hard drives.
4 Dell PowerEdge 6850 with two Dual Core Xeon 3,2 GHz processor, 32 GBytes of RAM and 1,5 TBytes of hard drives.

providing all necessary services: DNS, FTP, Web, accounts of the developers, etc.

More importantly, aran runs a PXE server. PXE is centered on a database containing images of specific operating systems (OS) with particular configurations, and clients (i.e. the experimentation machines of the platform) can download and automatically install them when required. The PXE server then contains the images of all OS with the required configurations for all machines in all particular experiments. And each time a new experiment requires us to deploy a new OS version with its special configuration it is saved on the PXE server. Then, it is easy to dynamically reconfigure all machines for a specific experiment. Thanks to this server, it is then possible for the laasnetexp platform to be used by several users without spending much time for reconfigurations of machines jointly used in several experiments.

B. The emulation network

The emulation network is designed and set-up as a private network with private non routable addresses. It is supported by a Catalyst 6504 CISCO switch having 96 Gigabit Ethernet ports, and a non blocking switch fabric. As for the real experiment network, this section motivates the different choices made in the design of the emulation platform.

EXPERIMENTAL END TO END CONTROL

By construction, the emulation network is private. Its traffic is then fully controlled. We also use the reservation planning of the different machines for avoiding a machine to be involved in more than one experiment. The machine load is then also easily controllable. Therefore, all experimental conditions are under control, and experiments are then easily reproducible.

MEASURED AND MONITORED PLATFORM

The same DAG solution is used for the emulation platform as the one designed for the real experimental network. The same good performances apply for the emulation network.

INTEGRATION WITH OTHER PLATFORMS

For the moment, this aspect does not apply for the emulation platform which is completely private. However, it would be very easy to use one of the machines as a router between the private and public addresses spaces, and use a PIP tunnel for interconnecting the emulation platform with other similar platforms;

ISOLATION FROM REAL WORLD OR BETWEEN RUNNING EXPERIMENTS

By construction, this emulation network is isolated from the Internet. For isolating different experiments (which use separate sets of machines) on this emulation network, the use of appropriate VLAN and the overprovisioned capacities of the switch ensure that the two experiments will not compete for the same resources.

SPECIFIC NETWORK TECHNOLOGIES EMULATION

This emulation network currently integrates a satellite network emulator. An example of how the laasnetexp emulation platform is used for emulating a particular interconnection of satellite and terrestrial networks is presented in section IV.C.

The platform also includes some wireless devices (mini-PC, PDA) and WIFI access. They allow experiments with wireless access networks which are more and more frequent in current Internet and its new usages;

MANAGEMENT

As for the public real experimental platform, it is possible to dynamically configure the involved machines and the network for a particular experiment. For the OS, the PXE server is used as already described. For the network configuration, it relies on the use of VLAN. Each experiment defines a set of VLAN for artificially creating the topology to emulate on the CISCO 6504 switch. This topology, or more precisely the set of VLAN, are saved and can be easily and quickly restored for reconfiguring the network for a specific experiment. As for the laasnetexp real experiment platform, this allows the platform to be used in several experiments during the same periods without spending much time for network and machines reconfigurations.

IV. LAASNETEXP USAGE FOR REALISTIC AND REPRODUCIBLE EXPERIMENTS

As it has been designed and set-up, the real experiment and emulation networks will provide an experimental framework whose experimental conditions are completely controlled and reproducible. However, it remains important to be able to make these experimental conditions realistic. Indeed, it is very important to test any protocol or architecture with realistic running hypothesis. It is also important to test them with extreme conditions for evaluating their limits.

Based on the real traffic traces captured on the LAAS' operational network, as well as on publicly available traffic traces, it was possible to issue realistic characteristics for the delays, loss processes, etc. in the Internet. It is then possible to make the emulators (when emulating a network) enforce such loss or delays distributions.

A. Realistic traffic generation

It then remains to generate realistic traffic on an experimental network, in particular for background traffic. Realistic traffic in this case means traffic having all the statistical characteristics of real traffic, i.e. a large variability, long range dependence (LRD) properties, etc. [5] [8] [9] [10]. It is well known that these properties have a negative impact on the quality of service [4]: it is not easy for actual protocols or network mechanisms to handle big peaks and large variations. For a solid validation of tested architecture and/or protocols, it is also recommended to exaggerate the level of such bad properties, just to detect their limits.

Monitoring: DAG GE

Moving monitoring: DAG GE

Monitoring: DAG 100 Mbps

Active measurements

Juniper M7i

CISCO router

REMIP 2000

RENATER 3

GEANT

LAAS

CISCO 7204

Ethernet 100Mb/s

DMZ

Ethernet 100Mb/s

Ethernet 100Mb/s

LAAS operationnal network

SAN + traces analysis

DELL

Eth2   Eth3
Double connection
to the private network

Eth0   Eth1
(Domain 1)   (Domain 2 or 3)

Laasnetexp.fr

WIFI

Private emulation network

Satellite emulation

Experiment machines

Multi-domains network

GigE

Aneto

193.49.97.0-63/26
Domain 1

Aran server

Posets

193.49.97.64-79/28
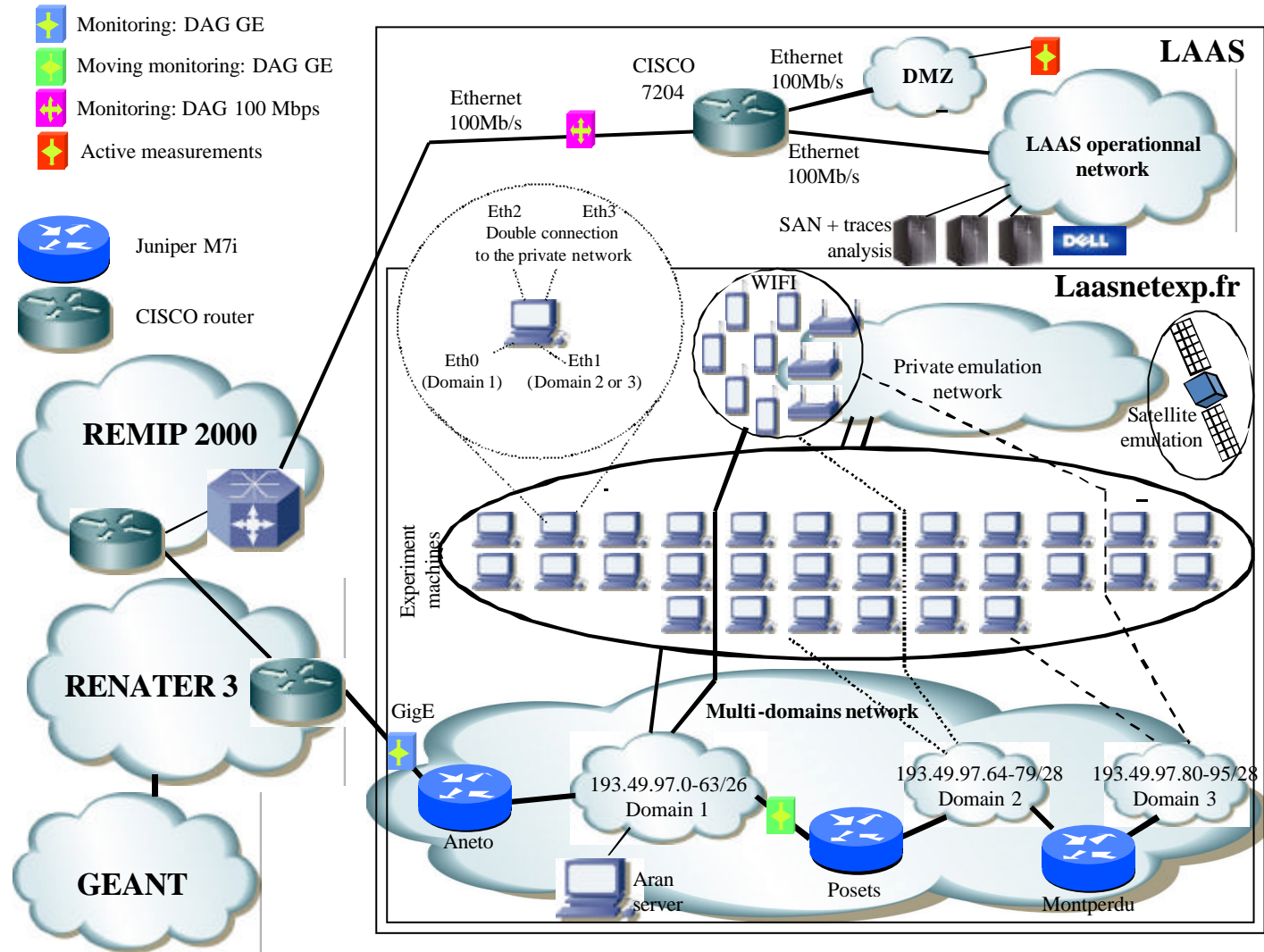Domain 2

193.49.97.80-95/28
Domain 3

Montperdu

Figure 3.   The laasnetexp platform

For generating such traffic, we issued the Gamma-Farima model for Internet traffic which is capable of describing the actual traffic variability and LRD, for both normal and anomalous traffic. Of course, many models have been issued since the last decade, but many of them rely on Gaussian assumption on the traffic (what is not true), and the other are far too complex to be useful. Our Gamma-Farima model has been presented in [6]. Its originality is its ability to very easily model the real traffic which is non Gaussian and long range dependent. This model is based on a mixture of Gamma law for modeling traffic marginal distributions, and a Farima model for modeling the LRD property. Finally, all the complexity of the traffic can be described by only 5 parameters:

- α is the (Gamma) shape parameter of the traffic

- β is the (Gamma) amplitude parameter of the traffic

- d expresses (Farima) Long Range Dependence

- φ and θ express (Farima) Short Range Dependence

For interested readers, the Gamma (α, β)-Farima(φ, d, θ) model and its validation with many real traces are presented in [6].

The related Gamma-Farima Traffic Generator (GFtg) consists of a set of 2 tools:

- The Gamma-Farima counts generator which, based on the 5 parameters of the Gamma-Farima model, generates a time series indicating how many bytes or packets have to be sent per time unit Δ.

- The Gamma-Farima injector which injects in the network the traffic indicated by the time series produced by the Gamma-Farima count generator. For respecting the time intervals between packets at the sender side without interaction from the networking environment, the traffic is generated using UDP.

B. Gamma-Farima Traffic Generator validation

METHODOLOGY

GFtg aims at generating realistic point to point background traffic. As a consequence, GFtg has only to be locally validated, as the important aspect to validate is that the sent traffic, right out of the network card of the generator machine, respects the wanted traffic characteristics.

The validation of GFtg has then been achieved on lasnetexp. A standard PC for generating the traffic has been used as well as a DAG system for collecting traces just on the link connecting the generating PC to the Ethernet switch.

GFtg has been validated on several traffic traces: public traces (as Auckland, NLANR, etc.) and traces collected in the framework of the French MetroSec project (these traces contain traces of normal traffic as well as traces containing anomalous traffic, both legitimate – as flash crowds – and illegitimate – as DoS attacks). Table 1 lists all public traces used for validating GFtg. It also indicates the kinds of

anomalies which have been generated in the framework of the MetroSec project [7]:

Flash crowds anomalies: We created anomalies that are considered legitimate under the guise of flash crowds (FC) on a Web server. Our goal was to generate realistic FCs. This is why we chose not to use automatic programs or robots, but to involve human volunteers. To do so, we asked to a large number of people (mostly French academics but not only) to browse the LAAS Web site (http://www.laas.fr). The LAAS Web site contains a large variety of files of all sizes, from simple html pages to movies, big reports, high definition pictures (of nano-devices, etc.), movies (of autonomous robots, etc.), etc. There is every indication of heavy-tailed file sizes on this Web site as is largely expected. Participants were instructed to browse the Web server on their own, as they would do in the real world when visiting a Web site publishing a new set of information they would be interested in. Precise starting and stopping times were given. FC lasted 30 minutes or so. A detailed analysis of the IP addresses present in the LAAS incoming traffic enabled us to find out that more than 150 people participated.

DoS attacks anomalies: We performed UDP flooding DDoS attacks using either IPERF or Trinoo (on computers with Linux distribution) to generate UDP flows with different throughputs. Compared to IPERF, Trinoo uses a "daemon" installed on each attacking site (four French research laboratories located in Mont-de-Marsan, Lyon, Nice, and Paris) and enabled us to create more complex and realistic attacks. The single computer target was located at LAAS in Toulouse. The traffic related to these attacks was transported via the French national network for education and research (RENATER). DDoS attacks were performed so as to be able to reproduce and modify their characteristics (duration, DoS flow intensity, packets length, and sending rate). In each case, traffic was collected by us (for durations of 60 or 90 minutes, the attack mostly occurred during the second third) before, during, and after the DDoS so that regular traffic can be analyzed before and after each attack

For validating GFtg, and for each trace:

a. the 5 Gamma-Farima parameters have been computed from the original traffic trace;

b. GFtg has been used to generate a traffic based on these 5 parameters, and this traffic has been captured by the DAG system so as to create a replayed traffic trace;

c. The 5 Gamma-Farima parameters have been computed from the replayed trace and compared to the ones of the original trace.

If both 5-tuple are close, this validates our GFtg.

Data	Date (start time)	Duration (s)	Network link	#Pkts (x10⁶)	Inter Arrival Time (ms)	Repository
PAUG	1989-08-29(11 :25)	2620	LAN(100BaseT)	1	2.6	ita.ee.lbl.gov/index.html
LBL-TCP-3	1994-01-20(14 :10)	7200	WAN(100BaseT)	1.7	4	ita.ee.lbl.gov/index.html
AUCK-IV	2001-04-02(13 :00)	10800	WAN(OC3)	9	1.2	wand.cs.xaikato.ac.nz/wand/wits
CAIDA	2002-08-14(10 :00)	600	Backbone(OC48)	65	0.01	www.caida.org/analysis/workload/oc48/
UNC	2003-04-06(16 :00)	3600	WAN(100BaseT)	4.6	0.8	www.dirt.cs.unc.edu/ts
METROSEC -ref1	2004-12-09(18 :30)	5000	LAN(100BaseT)	3.9	1.5	www.laas.fr/METROSEC
METROSEC -ref2	2004-12-10(02 :00)	9000	LAN(100BaseT)	2.1	4.3	www.laas.fr/METROSEC
METROSEC -DdoS	2004-12-09(20 :00)	9000	LAN(100BaseT)	6.9	1.3	www.laas.fr/METROSEC
METROSEC -FC	2005-04-14(14 :30)	1800	LAN(100BaseT)	3.7	0.48	www.laas.fr/METROSEC

Table 1.   Examples of traces used for validating GFtg

VALIDATION RESULTS

The following tables show for two examples taken from the preceding described traces (these two traces have been selected because they have very different average packet rates and are then a good illustration of the GFtg capabilities), the differences between the 5 parameters value of the traffic we wanted to reproduce and the 5 Gamma-Farima values measured on the replayed trace.

	a	ß	d	f	?
Original trace	2,56	2,40	0,222	0,407	0,172
Replayed trace	2,60	2,36	0,225	0,405	0,160

Table 2.   Gamma-Farima comparison of the profiles between the real and replayed traces. Average rate = 6 packets/ms

	a	ß	d	f	?
Original trace	26,69	1,36	0,263	0,015	0,274
Replayed trace	29,52	1,23	0,283	0,072	0,343

Table 3. Gamma-Farima comparison of the profiles between the real and replayed traces. Average rate = 26 packets/ms

It then appears on these two tables for low and high packet rates that the accuracy of the injector is quite good as the values of the 5 parameters of the Gamma -Farima model are very close in both cases. This has also been confirmed on all traces we have been replaying in the past with the GFtg tool.

C.   Usage example: Broadband satellite network emulation

A good example that shows the flexibility and the performances of our experimentation platform is the emulation of a complete DVB -S/RCS broadband satellite network.

The main motivation for this experimentation was to demonstrate the network and application services integration over next generation IPv6 satellite systems and the possibility to interoperate with terrestrial networks.

NETWORK ARCHITECTURE

The testbed we have built is able to emulate a complex scenario for next generation satellite network, fully compliant with the architecture adopted within the ETSI BSM group and the DVB-RCS standards. Figure 4 presents the network configuration. Each network element involved in a real DVB-S/RCS satellite network is emulated in our platform on a dedicated node. In fact, 3 users LAN of two nodes (standard Linux systems) are connected to the emulated satellite network with 3 Satellite terminals (ST) that implement an almost complete DVB -S/RCS stack. The satellite core network is emulated thanks to the Satellite Emulator (SE) as link emulator and the Network Control Center (NCC) for bandwidth management (DAMA). 11 computers, 4 VLANs and a connection to the Internet are used as described in Figure 4.
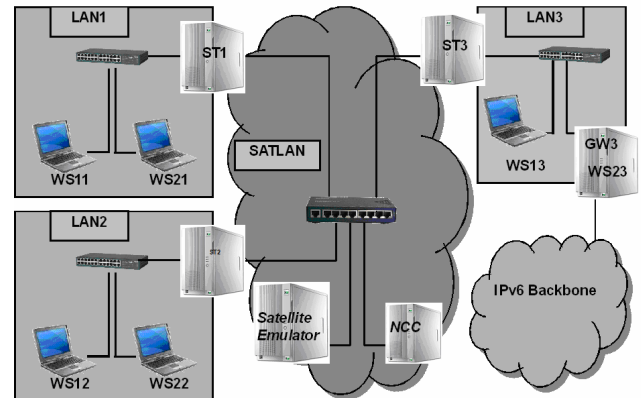

Figure 4. Full satellite emulation testbed

PHYSICAL LAYERS EMULATION

The physical layers emulation is based on two principles: The satellite carrier emulation that emulates the natural broadcasting of satellite systems, and the satellite link emulator that provides delays and errors.

The satellite carrier emulation is designed to operate on top of Ethernet frames and for each satellite channel corresponds an Ethernet multicast address. Ethernet was chosen for its native broadcast abilities (like a spot) and also for its high bandwidth capacities.

For each spot, we distinguish at least 4 channels:

- A dedicated channel for each data DVB-S flow spot descending from the satellite (one channel per DVB-S flow per spot)

- A dedicated channel for each data DVB-RCS flow on each spot ascending to the satellite

- Two dedicated channels for control frames (connection request, connection confirm, TBTP, …) one for the ascending flow and one for the descending flow

The satellite emulator can both act as a transparent or a regenerative satellite and it is able to simulate spot changes and signal format conversion. Indeed, the regenerative satellite with an onboard switching matrix processes DVB-RCS frames, switches ATM cells received from these frames and rebuilds them into DVB-S frames. This switching table could be updated by appropriated control message. However the main functionality of the SE is its satellite link emulation module which simulates in real time the moduling/coding part thanks to precalculated BER files. In addition to inject bit errors, the satellite emulator introduces delay and jitter that can be also tuned.

RESULTS

Thanks to the flexibility of the platform (programmable switch, multiple Ethernet interfaces), it was easy to set up and make a complex network infrastructure evolve. The Ethernet switch performances (720 gigabits switching matrix) are enough to go through the performance limitations of the satellite carrier emulation. In fact, with the multi-protocols (DVB-S and DVB-RCS) and the multi-spots emulation, the throughput generated by the emulation is exponentially linked to the user's traffic and the scenario complexity (Users LAN numbers).

This platform has been allowing us to validate complex resource algorithm for QoS management over next generation satellite systems that would have not been possible over real satellite systems. In particular, it was possible thanks to the ability to generate realistic traffic, in particular for background traffic.

V. CONCLUSION

This work has addressed the design and setting-up of an experimental platform for leading researches in networking. The objectives in terms of experimental conditions control for getting reproducible and easy to analyze experiments have been successfully reached. The originality of this platform stands in its ability to provide both real as well as emulated network support seamlessly and this with the same guarantees in terms of experiment control.

This platform also comes with a methodology cookbook and tools for performing realistic experiments. The methodology relies on the use of actual Internet and Internet traffic characterization, analysis and modeling. It gives rich information on how to configure network emulators, but also leads to the design and development of a realistic traffic generator. Generated traffic ranges from all kinds of normal traffic to all kinds of anomalous traffic. The assessment of this platform and its associated methodology for running experiments proved to provide very realistic experimental conditions. Laasnetexp then appears as the perfect tool for network experiments.

REFERENCES

[1] J. Cleary, S. Donnelly, I. Graham, A. McGregor, M. Pearson, "Design principles for accurate passive measurement", PAM 2000, Hamilton, New Zealand, April 2000

[2] Fall, K., *Network emulation in the Vint/NS simulator*, proceedings of ISCC'99, July 1999

[3] S. Floyd and V. Paxson, *Difficulties in Simulating the Internet*, IEEE/ACM Transactions on Networking, Vol.9, No.4, pp.392-403, August 2001

[4] K. Park, G. Kim, M. Crovella, "On the Effect of Traffic Self-similarity on Network Performance", SPIE International Conference on Performance and Control of Network Systems, November, 1997

[5] K. Park, W. Willinger, "Self-similar network traffic: an overview", In "Self-similar network traffic and performance evaluation", edited by K. Park and W. Willinger, J. Wiley & Sons, 2000

[6] Antoine Scherrer, Nicolas Larrieu, Pierre Borgnat, Philippe Owezarski, Patrice Abry, "Non Gaussian and Long Memory Statistical Modeling of Internet Traffic", 4th International Workshop on Internet Performance, Simulation, Monitoring and Measurements (IPS MoMe'2006), Salzburg, Austria, February 27-28, 2006

[7] A. Scherrer, N. Larrieu, P. Owezarski, P. Borgnat, P. Abry, "Non Gaussian and long memory statistical characterization for Internet traffic with anomalies", IEEE Transaction on Dependable and Secure Computing, Vol. 4, No. 1, pp 56-70, January-March, 2007

[8] K. Thompson, G. Miller, M. Wilder, "Wide-area internet traffic patterns and characteristics", IEEE Network, Vol. 11, n° 6, pp. 10-23, November/December 1997

[9] W. Willinger, M. S. Taqqu, R. Sherman, D. V. Wilson, "Self-similarity through high-variability: statistical analysis of Ethernet LAN traffic at the source level", IEEE/ACM Transactions on Networking, Vol. 5, n° 1, pp. 71-86, 1997

[10] W. Willinger, V. Paxson, M. Taqqu, "Self-Similarity and Heavy Tails: Structural Modeling of Network traffic", In "A Practical Guide To Heavy Tails: Statistical Techniques and Applications", ISBN 0-8176-3951-9, 1998