Use of Virtualization Tools in Computer Network Laboratories

Fermin Galán 1, David Fernández 2, Javier Ruiz2, Omar Walid2, Tomás de Miguel2
1Agora System, S. A., 2Universidad Politécnica de Madrid
Aravaca, 12
28040 Madrid
SPAIN
e-mail: fermin.galan@agora-2000.com

Abstract – The paper describes how virtualization tools can be used in computer network laboratories to simplify and dramatically reduce its deployment and management costs. In particular, the Virtual Network User Mode Linux (VNUML) free-software tool (developed as part of the Euro6IX IST research project) is introduced, showing how it can be used to easily build complex virtual network scenarios.

I. INTRODUCTION

Computers laboratories are a must in many colleges and high schools, especially in those related with computer science or telecommunication. Apart from its general purpose, they are often used for other specific purposes like teaching network services administration, protocol design, implementation and testing or network management and configuration.

The laboratories used to teach that subjects are typically made of two different parts: a production network made of a Local Area Network (LAN) interconnecting general purpose personal computers used by the students to execute standard ofimatic applications, programming environments or simulation tools, as well as typical Internet applications like e-mail or www; and an experimental network made of other computer network specific laboratory equipment like hubs, switches, routers, servers, as well as the physical infrastructure (wires, etc) that joins them.

For example, a laboratory for teaching network administration should be made of some servers running typical Internet services like DNS, WWW or e-mail and some network nodes (routers), as well as some clients to allow students to test the services in an almost real scenario. The students need complete access to that equipment in order to change its configuration (that is, they need “administrator rights” as they are learning to be administrators of that network services), and, ideally, there should be several laboratory scenarios, in order to allow several students or student groups to work in parallel without interfering among them.

The deployment and management of a computer network training laboratory like the one described above is in general a costly task in terms of time and resources. First of all, the equipment used is relatively expensive. Although personal computers and basic LAN equipment prices are affordable, the cost of network specific equipment like routers or ATM or FrameRelay nodes is high. Besides, there are software costs associated with that equipment, related with the operating systems and applications running in clients, servers and nodes.

Furthermore, the management of such laboratories is often a very resource consuming activity. Apart from the initial effort dedicated to the physical installation of the equipment, the installation of their software and their configuration, a lot of effort has to be invested in reconfiguring all the equipment at the end of the exercises, in order to get them to their initial state or to use them for a different exercise. In our experience, this can be the most resource consuming task if the laboratory is not carefully designed, as it can involve the reinstallation of operating systems or the physical reconfiguration of network connections.

This paper presents how the use of software virtualization techniques makes possible an alternative way of building computer network laboratories. In particular, it describes and analyses how a free-software tool named VNUML is being used to enhance and simplify the management of an already running computer network laboratory at the Technical University of Madrid.

VNUML tool was initially developed as part of the Euro6IX research project (partially funded by the European Union) to simulate and experiment complex next generation IPv6 based Internet scenarios, in particular the ones related with the new model of Internet Exchange Points (IX) [1]. However, VNUML is a general purpose tool that can be used to simulate general Linux based network scenarios, for example, to create network testbeds to test new applications and network services and reduce its development cost; or to complement computer network laboratories as it is the case described in this paper.

Tools like VNUML can dramatically reduce the investment needed to deploy a computer network laboratory, as well as its management costs, simplifying and adding flexibility to the creation of exercises. As it is later explained in the paper, VNUML can be used either to completely implement computer network laboratories or to complement existing laboratories made of real equipment, extending them to cover more complex and realistic scenarios.

1 See Euro6IX project web server in http://www.euro6ix.org
The paper is organized as follows. Section II introduces software virtualization techniques. Section III presents the computer network laboratory where VNUML tool is being introduced and describes the different usage options, showing their advantages and drawbacks. Later, section IV describes some technical details about VNUML tool, and section V presents interesting considerations about the use of virtualization tools in laboratories. Finally, section VI summarizes the conclusions and presents some future research lines related to the work presented in the paper.

II. VIRTUALIZATION TECHNIQUES

In general, virtualization techniques consist on running a process unit -a single program, and operating system or even an entire box- inside a program environment -usually called jail or sandbox- which is running in a physical machine named the hosting machine or simply the host.

By means of these techniques, a powerful enough hosting machine can be used to start a set of “virtual machines” interconnected by one or more “virtual networks”, creating a “virtual network scenario” that closely emulates the behaviour of the same scenario implemented with real equipment.

The main advantage of this approach is that the processes running in these virtual machines behave almost exactly as if they were running on a real environment. In this sense, the most execution transparency achieved, the better the virtualization technique is.

Complete virtualization transparency is very difficult to reach, due to the fact that virtualization layers introduce some overhead. Besides, as several virtual machines are emulated on a single hosting machine, these techniques can not be used, for example, to evaluate performance, as the host resources have to be shared between all virtual machines.

Anyway, given the maturity of the current virtualization techniques and being the present hardware used in standard personal computers (the ones typically used as hosting machines) powerful enough, this is not problematic, at least not for the computer network scenarios used in laboratories.

The main advantage of virtualization techniques when used to emulate network scenarios is the significant reduction in equipment and management costs, compared to the equivalent real scenarios. First of all, only one device (the hosting machine) has to be used to implement the entire scenario, saving the cost of all the real equipment and infrastructure (wire, hubs, spare room, etc.) needed to create it. But also, the administration of the whole scenario is highly simplified, as all the configurations of its components are now stored in files on the hosting machine, being easy to restart the scenario, take it to a known state or make any modification. We will go deeply into these and other advantages and drawbacks in section V, applied to the VNUML virtualization tool presented in this paper.

Applied to computer network laboratories, the use of virtualization techniques will simply consist on completely or partially emulating the network scenarios habitually used in laboratory exercises -typically made of clients, servers, nodes and networks- inside one or more hosting machines. As it will be shown in the next section, this approach allows the easy creation and replication of complex network scenarios using a very small number of pieces of equipment -ideally only one personal computer-, significantly reducing laboratory hardware and management cost.

III. PRACTICAL EXAMPLE: COMPUTER NETWORKS TRAINING LABORATORY

As an example about how virtualization techniques can contribute to improve computer network laboratories, in this section we will describe some on-going and future activities related to the use of VNUML tool in the Computer Networks Laboratory of the Telematics Engineering Department of the Technical University of Madrid [2].

That laboratory was created with the general objective of giving the last year grade students, as well as postgraduate students, practical skills in computer networks, complementing what they have learnt in theoretic courses. The idea was to push the students to go deeper in the study of network protocols by means of exercises based on the use of network simulators and real equipment, in order to get the “real feeling” of computer networks.

In particular, the laboratory covers or has covered in the past the following areas:

- Study of the behaviour of network protocols. It can be done by using protocol analysis tools, like Ethereal2, and also by using network simulation programs (mainly NS3).
- Network Planning and Design, mainly covered by a set of laboratory exercises carried out with network simulation programs.
- Network and Services Configuration, covered by some exercises where the students have to configure network cards and nodes in several heterogeneous network scenarios including LAN, ATM, Frame Relay and ISDN, as well as typical network applications and services like DNS, web, ftp or DHCP.
- Network Management, covered by some exercises with SNMP-based network management software and some other diagnostic and performance measuring tools.

One of the main design premises of the laboratory was that it should be mostly based on real commercial equipment, if possible from several different manufacturers, as it was considered very important that the students get experience on the configuration of real equipment, similar to the one they will later use on their professional career.

The laboratory is made of several exercises, each one focused on some specific objectives and network technologies. Although virtualization techniques can be applied to most of the exercises, we will concentrate the attention on one of the final exercises, where the advantages of using such techniques are more outstanding.

2 http://www.ethereal.com
3 http://www.isi.edu/nsnam/ns/
This exercise is based on the network scenario shown in Fig. 1. It represents partially the network of a typical medium-size organization made of:

- Sites, which represent the different offices of the organization.
- Regional sites, which represent the offices which group the activity of the organization in a region and concentrate all its traffic.
- Backbone network, which transports all the traffic among regional sites and headquarters.
- Central site (headquarters), where the main organization servers and applications are located.

This scenario is designed to allow the students to gain experience on the design, configuration and management of a real network. The main focus of the exercise is on addressing and routing, which is one of the most important functions related to the design and management of IP networks.

For that purpose, the students have to design the network addressing plan and later configure the network using static and dynamic routing based on OSPF (Open Shortest Path First) routing protocol [3]. Related to that, the scenario allows the students to experiment with interesting aspects like measuring convergence times in case of link failures, using address aggregation techniques to improve the scalability of the network, load balancing traffic, controlling the injection of external routes to the network or measuring network performance.

Moreover, the students have to optionally configure all the typical services needed in a network like that, for example, the DNS service, the autoconfiguration service based on DHCP, WWW and FTP servers, as well as network management.

This exercise is made cooperatively among up to twelve student groups (typically made of two students). Each group is responsible of one site, having to configure the site router and host, as well as the routers of the regional site and optionally the backbone routers, cooperating with other groups connected to the same regional site.

This exercise has been in place for more than four years and more than 500 undergraduate and postgraduate students have made it. Although the results have been in general very positive, with a high degree of interest and acceptance from the students, the effort to set-up and maintain the scenario has been high and several problems have been experienced, most of them related to the fact that, as it can be appreciated in Fig. 1, the complete scenario involves a high number of systems (more than 30 routers and more than 20 hosts). To be mentioned:

- High cost of the equipment involved.
- High management cost of the whole scenario in terms of resources and time. Although the use of several ad-hoc program scripts and auxiliary equipment like console servers simplify a lot the management, the cost is still high.
- Changes in the scenario require “physical” changes on the equipment (some cables have to be reconnected when changing from one exercise to another).
- The students have to make the exercise at the same time in order to experiment with the complete scenario.

The use of virtualization techniques can greatly contribute to improve and simplify the management of this exercise, solving most of the problems mentioned. The following subsections describe different options related to the use of VNUML tool in this context. Some of them are already in place or will be soon introduced.

A. Whole scenario emulated using VNUML

The first option is to emulate the whole scenario using VNUML. As already tested, a standard PC with enough memory (at least 512 Mbytes) can emulate the whole scenario without problems.

This option is clearly the cheapest one, as only one PC is needed for each student or group of students. Besides, the exercise can be made by students at home. However, the main disadvantage is that the students do not use real equipment.

Fig. 1. Virtual network laboratory
B. Auxiliary systems emulated using VNUML

Several auxiliary systems are needed to run the exercise, for example, DNS or web servers on regional sites and headquarters, as well as other systems that allow the students to run performance measurement applications or capture traffic using protocol analyzers.

All these systems can be emulated using VNUML on just one personal computer, drastically reducing the number of systems involved and their management cost. Besides, the students will not see any difference when using that emulated systems, as they normally do not have physical access to them (they access them through remote login).

C. Mixed real/emulated scenario

The most interesting option consist on emulating using VNUML the part of the scenario that each group of students do not directly manage. For example, as shown in Fig. 2, all systems but site 9, site 10 and regional site E can be emulated using VNUML.

As explained in detail in next section, each server and router belonging to the emulated part of the exercise will become a Linux “virtual machine” running inside the host. Besides, UML provides the internal interconnection between these virtual machines by means of “virtual networks” that resemble the real scenario, as well as the connection to the external real network.

This option allows any group of students to make the exercise using the whole network scenario independently of the rest of students. However, in contrast to option A, now the students configure real equipment.

D. Adding complexity to the scenario

Finally, VNUML capability to emulate network scenarios externally connected to real networks opens the door to a wide range of possibilities to extend the scenario. For example:

- More sites or a new topology hierarchy representing another concentration level could be added to make the network bigger.
- Connections to external networks (autonomous systems) could be emulated in order to practise with BGP [4] routing protocol and the interaction between BGP and OSPF.
- A more complex backbone could be emulated using VNUML, including when available MPLS linux implementations, to allow testing traffic engineering or QoS routing capabilities.
- Level IV Linux switching implementations could be introduced in the headquarters network, in order to test server farm techniques used nowadays.

IV. THE VNUML TOOL

Among the available virtualization alternatives, that range from process virtualization -like FreeBSD jails [5]- to virtual machine hardware emulation -like VMware [6]-, VNUML is based on User Mode Linux (UML [7]), an operating system virtualization environment that allows running completely functional Linux kernels as conventional user processes inside a standard Linux box.

UML is based on a modification to the standard Linux kernel that allows it to run as a conventional user process on top the Linux kernel running in the hosting machine. Several “UML processes” can be executed simultaneously on a host, each one being a “virtual machine” with its own resources and its own running processes.

The functionality included in this “virtual kernels” is basically the same than the one that can be found on a standard Linux kernel (for example, IPv6 or multicast support can be enabled on virtual machines), with the exception mentioned about the performance penalty caused by the overhead introduced by the virtualization layers.

Apart from providing the functionality to create virtual machines, UML provides the mechanisms to interconnect virtual machines using “virtual networks” implemented at operating system level. The virtual networking techniques used are very flexible: they allow the interconnection to be configured at level 2 (bridge) or 3 (host acting as a router), as well as the transparent interconnection of the emulated scenario to external equipment using the physical network interfaces of the host. Moreover, VLANs (Virtual Local Area Network [8]) are supported, allowing the creation of several simultaneous scenarios independently interconnected to external networks, as it is supported in VNUML.

As an example, Fig. 3 shows the process space of a hosting machine executing two interconnected virtual machines (each one executing two processes) using UML.

UML is a powerful but complex tool: it is generally difficult to manage medium-to-big size simulations “by hand”. VNUML tool was designed to allow the easy creation of UML based network scenarios hiding the user all the underlying UML complex details. VNUML allows the users to enjoy the advantages of the virtualization without having to deal with the advances topics in GNU/Linux administration that raw-UML requires (like virtual bridges, tap devices or UNIX sockets).

VNUML is a free software tool developed made of two components: a simple descriptive XML-based language that allows specifying in a text file the scenario to be emulated; and an

---

4 A complete description of VNUML and all its functionalities, as well as the tool itself, can be found at http://www.dit.upm.es/vnuml.
application that interprets the language and builds and manages the scenario inside the hosting machine, hiding all the complex UML details to the user.

VNUML users have to write the specification of the desired emulated scenario in a text file. The XML-based language used is very simple and descriptive. That allows the user to just concentrate on his problem, for example, in our case, on the design and implementation of the computer network laboratory exercise scenarios, detaching from low level implementation details.

Fig. 4 shows a simplified example of the VNUML specification language, derived from the network training laboratory exercise presented in section III. In particular, the listing included in the figure shows a fragment of the description of site 10. Each virtual network is specified in a listing included in the figure shows a fragment of the description of site 10. Each virtual network is specified in a

The VNUML application is a parser that reads the simulation specified in the text file and automates all the tasks needed to build and manage the simulation scenario, for example, copying files from the host to the virtual machines, starting and stopping process inside, running command scripts, etc. No in-depth knowledge about Linux or UML is needed to use it.

V. CONSIDERATIONS ABOUT VNUML-BASED SIMULATIONS

Network scenarios based totally or partially on virtual systems built with VNUML present many advantages compared to the same systems build using real elements.

Firstly, apart from the important savings in the initial investment needed to create the scenario, VNUML allows significant savings in the upgrade costs. Routers are generally expensive and, although new functionalities can be added by software upgrades, after four or five years, the hardware has to be replaced. On the contrary, the replacement cost of VNUML based parts of the laboratory is much lower, as it is based on standard personal computers.

As mentioned before, the virtualization approach greatly simplifies the laboratory management tasks, due to the fact that a big part of the laboratory runs inside one or more emulation boxes and it is basically made of software and data files. For example, all the configuration files needed to create the OSPF training laboratory can be stored in less than a CD. Besides, once the scenario is started, the configuration of any of the systems can be easily changed by means of the VNUML management interface.

VNUML virtual machines are controlled from the host machine through a management interface automatically configured when the simulation starts. Commands and configuration files can be passed through this interface in order to control virtual machines behaviour, even in the case a misconfiguration has isolated them from the rest of the scenario. VNUML management interfaces are conceptually similar to the “console servers” typically used in data processing centers to remotely access equipment consoles. But, even in the worst case that the management interface is not working properly, the whole scenario can be simply restarted by executing a software command.

VNUML can be considered as a Zero Administration technique applied to virtual systems. With VNUML, it is very easy to regenerate the emulated part of a laboratory exercise when a crash or serious misconfiguration occurs (a frequent situation when dealing with training environments), avoiding the cost (mainly in terms of administrator time) to locate and fix the problem.

In this sense, VNUML can be easily combined with other techniques used in computer laboratories to achieve zero administration in client machines and even reduce to a minimum server administration (see, for example, the solutions being used in our laboratory [9]).

Although using virtualization in general and VNUML in particular brings many advantages, there also some drawbacks that have to be considered. One of them is the overhead (mainly in CPU usage, physical RAM memory and storage capability) introduced in the hosting machine by the virtualization layer. Anyway, as the UML backend is highly optimized, even standard personal computers can cope with most of the typical scenarios found in laboratories. Tests performed so far have demonstrated that the training networking laboratory scenario presented in this paper (more than 50 machines) runs properly in a state-of-the-art PC (Pentium IV at 2,8 GHz with 1 Gbyte of RAM).

However, as mentioned before, the overhead introduced by the virtualization layer prevents this approach to be used for performance evaluation, as the overhead penalty that can distort throughput and latency measures made over processes running inside virtual machines, for example, about the traffic crossing virtual networks.

Finally, an important drawback of VNUML is that real equipment (for example, commercial routers) cannot be emulated, due to the fact that UML is only able to virtualize Linux kernels. In any case, when the use of real equipment is really important -for example, in laboratories designed to train in the use of that specific equipment- virtualization techniques can be used to complement real equipment, allowing testing them in virtual complex scenarios, as shown in case C of section III.

---

For a comprehensive description of the VNUML language, refer to project web page (http://www.dit.upm.es/vnuml).
VI. CONCLUSIONS

The paper has shown how the use of virtualization tools and techniques can reduce significantly the cost and management effort invested in deploying and managing computer network laboratories.

In the case of the example laboratory exercise presented, the number of systems involved can be dramatically reduced, either in the ideal case of emulating the complete virtual scenario (about 50 nodes) inside a single host, or in the case the scenario is only partially emulated. Even if the hosting machine would need powerful hardware that raises its price four times the medium cost of a standard PC, as much as 90% cost savings can be reached.

The saving in management cost is even higher, because the time dedicated by staff personal to laboratory administration is dramatically reduced, as it was described so far.

Although, techniques like UML have already been used for testing and training purposes in projects like ADIOS\(^7\) or FreeS/WAN\(^8\), the solutions applied lacked the flexibility and simplicity needed to be used for teaching. VNUML tool comes as a flexible solution to easily define, create and test computer network laboratory scenarios, hiding UML and network related complex details to users and laboratory administrators. VNUML is a flexible solution that complements virtualization techniques giving the possibility to easily maintain tenths or hundredths of different scenarios.

This flexibility also arises when VNUML is compared with vendor-specific networking laboratory learning tools, like the ones used in CISCO networking academy programs. These tools usually achieve great emulation accuracy of specific router and switches models, but they are limited, in general, to a fixed set of scenarios and functionalities (although the latest versions of some tools already include flexibility in the free definition of scenarios; see, for example, the well-known Boson NetSim simulator\(^9\)). VNUML, in contrast, allows great flexibility for the creation of new scenarios, as well as the facility to include in them any new functionality, network protocol or application developed for Linux.

Furthermore, VNUML can also be easily integrated in Linux bootable CDs like KNOPPIX\(^9\), giving the students the possibility to “carry” the network laboratory home for stand-alone training.

Among others, current working lines on VNUML are towards including virtual machines using more flexible backends, like VMware (that allows emulate MS Windows or FreeBSD systems) and to introduce schedule events in simulations (for example, in order to study the adaptation of OSPF to changes in network topology, a fail in a network link can be scheduled).

ACKNOWLEDGMENT

The work presented in this article has been partially funded by the European Union in the context of IST Euro6IX project (European IPv6 Internet Exchanges Backbone). Several people has contributed to the VNUML project with good ideas, examples or just helping to test and debug the tool: Jorge López, María José Perea, José Luis Rubio and Antonio Tapiador from UPM, as well as Gabriel López and Antonio Gómez-Skarmeta from Murcia University.

REFERENCES


---

\(^7\) http://dc.qut.edu.au/adios/adios-project.html
\(^8\) http://www.freeswan.org/
\(^9\) http://www.knoppix.org