Teaching Advanced Concepts in Computer Networks: VNUML-UM Virtualization Tool

Antonio Ruiz-Martínez, Fernando Pereñíguez-García, Rafael Marín-López,
Pedro M. Ruiz-Martínez, *Member*, *IEEE*, and Antonio F. Skarmeta-Gómez, *Member*, *IEEE*

Abstract—In the teaching of computer networks the main problem that arises is the high price and limited number of network devices the students can work with in the laboratories. Nowadays, with virtualization we can overcome this limitation. In this paper, we present a methodology that allows students to learn advanced computer network concepts through hands-on experience with the VNUML-UM virtualization tool, which is offered freely as a resource for the practical teaching of mobility, load balancing, and high availability. To verify the utility of using the VNUML-UM virtualization tool in the teaching of advanced computer network concepts, we have performed some opinion polls to the students during the last three academic years. The obtained results confirm that our students agree that the VNUML-UM enables an enhanced learning process of the different concepts and their practical skills. This perception is also confirmed by the final marks obtained by the students, which have considerably improved along the years. To the best our knowledge, this paper presents the first experience that provides results on the use of virtualization to teach advanced concepts in the field of computer networks.

Index Terms—Computer and information science education, virtualization software tools, virtual labs, educational simulations, computer networks, teaching/learning strategies

✦

---

1 INTRODUCTION

IN the teaching of basic and advanced computer networks concepts (routing, mobility, load balancing, and so on) students not only have to acquire the theoretical knowledge but also the practical skills to configure and manage real computer network scenarios. In Engineering education, these practical skills are acquired through the development of the so-called *laboratories* [1]. The use of these labs should be made in groups of reduced number of students so that they can have the maximum availability to practice with the different network elements to correctly acquire the required skills. However, to attain this goal in real laboratories (hereinafter *labs*), the development of complex network topologies is necessary but costly (in terms of both economic and time).

Virtualization tools [2], [3], which allow the emulation of network infrastructures in a highly realistic way, have revealed as an alternative solution to avoid the aforementioned problems in an efficient and cheap way. Moreover, the integration of computer technology can help students to perceive the improvement of their learning process [4]. For this reason, in computer science education, these tools have traditionally assisted the teaching of different subjects [5], [6], [7], [8], [9], [10], [11], [12], [13], [14]. In particular, several authors have successfully applied virtualization for teaching basic network configuration and network routing [10], [11], [12], [15], [16]. Nevertheless, to the best of our

knowledge, virtualization tools are not being used to teach more advanced computer network concepts such as mobility, load balancing or high availability.

This paper presents the methodology and the tool we have developed and applied during several courses to teach these concepts in the *computer networks* subject (within the Degree in Computer Science Engineering of the University of Murcia). The tool, called VNUML-UM [17], has been adapted from VNUML [18]. To the best of our knowledge, VNUML-UM is the first virtualization tool that allows the teaching of such advanced contents in a practical way. Moreover, the success of our proposal is not only proved by examining during several courses the evolution of students' marks but also confirmed by students themselves through some opinion polls.

The rest of the paper is organized as follows: Section 2 introduces background related to virtualization and teaching. Section 3 describes the computer networks subject, where we are applying the learning methodology based on VNUML-UM. Section 4 presents the VNUML-UM tool and some examples of scenarios developed by our students. Section 5 points out the specific results we have obtained as result of applying the proposed methodology. Finally, Section 6 concludes the paper and outlines some future work lines.

2 VIRTUALIZED NETWORK LABORATORIES

In the following, we detail the advantages of virtualization techniques that have motivated the development of virtualized network laboratories based on the VNUML-UM virtualization tool.

2.1 Virtualization for Teaching Computer Networks

Unlike the teaching of subjects such as *operating systems* and *system administration* where a reduced number of

---

• *The authors are with the Department of Information and Communications Engineering, Faculty of Computer Science, University of Murcia, E-30100 Murcia, Spain. E-mail: {arm, pereniguez, rafa, pedrom, skarmeta}@um.es.*

TABLE 1
Comparison of Virtualization Tools

	License	Topology configuration	Scenario execution	Type of virtualized hosts
Imunes	BSD	Configuration centralized in a single configuration file. Optional graphical configuration.	Graphical simulation or through command line interface	Virtual hosts run a specific FreeBSD kernel that cannot be extended
Velnet	Restricted (based on VMware Workstation)	Topology is configured through a graphical interface. Virtual machines can be only configured once they are started.	Graphical execution where each virtual host must be independently started/stopped.	Virtual hosts can be based on any operative system.
MLN	GNU GPL	Single configuration file containing both topology and virtual machines configurations.	Single command line parser to manage the virtual network scenario.	Limited set of specific file systems supported for virtual nodes.
Netkit	GNU GPL	Configuration is split across various files to specify the topology and virtual machines configuration. Optional graphical configuration (NetLab).	Execution is controlled through different command line programs for managing virtual hosts, scenarios, etc.	Virtual machines run a Debian GNU/Linux distribution that can be extended with new software.
Dynamips	GNU GPL	Configuration is specified in a single file to indicate topology and virtual machines configurations.	Tedious execution based on command-line interface commands.	Only supports de virtualization of Cisco routers.
NVLab	Not available	Configuration is specified through a Web interface that is saved in a server as a XML file.	Executed in the server and the user accesses by means of VNC and works manually with the nodes. Only accepts one user logged into the system. The tool has not been tested in a real environment.	Nodes based on Xen
VNUML	GNU GPL	Topology and virtual machines are configured in a single specification file by using a simple XML-based language. Optional graphical configuration (VNUMLGUI)	Simple execution through a command line parser that not only controls the basic interaction with the scenarios (start/stop) but also allows the automatic execution of commands in virtual machines.	Virtual hosts can run any customized UML kernel that can be extended with new utilities.

computers is required to practice with them (typically two *personal Computers*—PCs—is enough to set up a scenario), the teaching of *computer networks* may require a large number of computers, especially when dealing with some advanced concepts.

The use of labs based on real equipment has some important disadvantages. More precisely, because there is a single infrastructure that cannot be used by all the students at the same time, the learning process must be organized in small groups of students that will practice in shifts and the time the students can work with the computer network scenario is limited. Conversely, the application of virtualization techniques in labs are of great benefit because they enable each student to work with his/her own network scenario.

Apart from solving different problems present in labs based on real equipment, virtualization facilitates the practical learning of computer network concepts. First, a key advantage highly appreciated by students is the mobility. Given that students can install the virtualization software in their own computer or laptop, they can work with the computer network scenario either at home or in the university premises. Thus, the students can improve their practice in the computer network scenario and gain in flexibility because they are not constrained to the timetable that is scheduled to work with the different physical components in a real lab. Second, virtualization also offers flexibility and students can design their own network topologies or teachers can create and distribute the scenario that the students will work with (by defining a topology

and its components). Finally, also thanks to virtualization, students can gain experience in configuring, deploying and managing tools in realistic topologies that they can find in their future professional activities. In summary, virtualization is a powerful tool that allows students to consolidate the concepts learned in theoretical lessons by practicing in a flexible and realistic environment.

2.2 Virtualization Tools for Virtual Network Scenarios

Currently, some well-known virtualization tools are: *Qemu*, *Xen*, *WMware server*, *virtual box*, and *User Model Linux* (UML). Nevertheless, these tools do not provide a specific mechanism to facilitate the deployment of virtual network scenarios. For this reason, in the latest years, some initiatives oriented to simplify the deployment and configuration of virtual network infrastructures have appeared: *Imunes* [19], *Velnet* [20], *My Linux Network* (MLN) [21], *Netkit* [22], *Dynamips* [23] and *Virtual Network User-Mode Linux* (VNUML) [18]. Their use in research and educational environments has already been studied in [12], where it has been concluded that VNUML is one of the most flexible and powerful tools for this purpose. Despite other virtualization tools have recently appeared (e.g., NVLab[24]), VNUML is still nowadays the most adequate solution for virtualizing network scenarios (see Table 1).

Thus, VNUML represents a promising option to teach advanced computer network concepts. Indeed, it allows us to execute a reduced version of both a Linux kernel and a file system file used by the machines of the virtualized
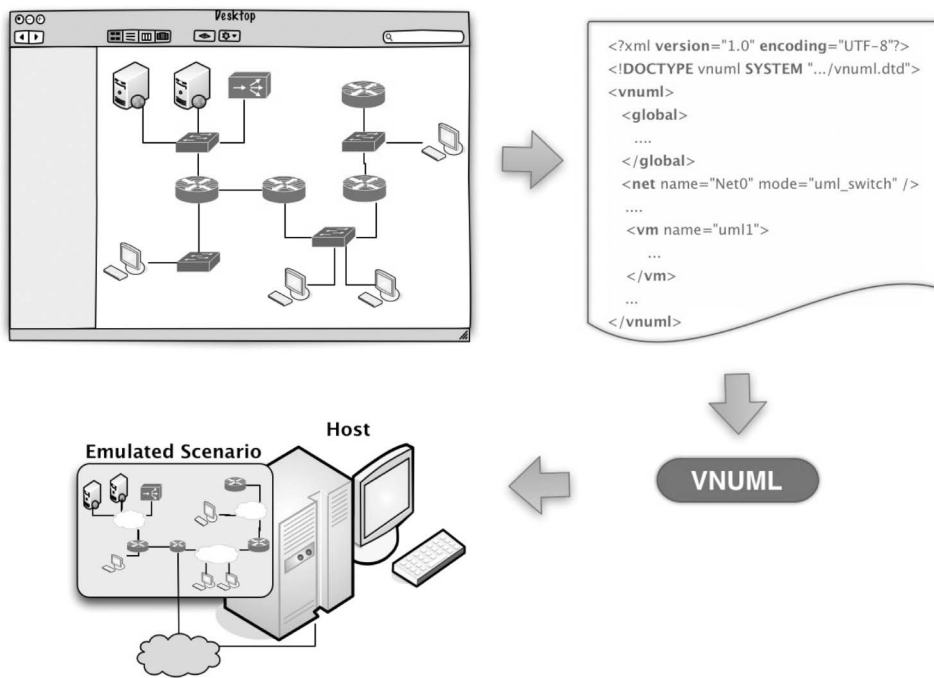
Fig. 1. VNUML development phases.

scenario. Therefore, the configurations used within the virtualization software can be used in real scenarios without any change because the virtualized environment employs the same tools used in real scenarios. Furthermore, the tool also shows a good performance and a reasonable memory consumption [25] as well as an excellent virtualization grade.

Moreover, VNUML is distributed under GNU Public License, easy to install and can be used once installed in a computer or directly in a live CD/DVD version. In general, the installation procedure comprises the VNUML parser, documentation, and its associated XML Document Type Definition (DTD). During the installation process we are also required to install two files used by the virtual machines: A file containing the Linux kernel and another one with the file system of the virtual machine. In fact, it allows the installation of different kernels and file system files to be able to test different virtual machine configurations. For example, we could have a kernel with support for only IPv4 (*Internet Protocol (IP) version 4*) and another one with support for IPv4 and IPv6.

VNUML also provides two main components to work with computer network topologies: 1) The VNUML language that allows the specification of computer network topologies; and 2) The parser that generates the virtual topology from a file with the specification of the computer network topology expressed in the VNUML language.

The process of using VNUML is organized in three phases (see Fig. 1). First of all, there is a *design phase* where the user (a student or a professor) conceptually describes the network scenario (first step in Fig. 1). In this phase, the user details the different aspects and entities involved in the network topology: Computers, routers, connections, and the different IP addresses of each interface of the equipment. Next, there is an *implementation phase* where the user generates a file with the specification of the scenario in the VNUML language (second step in Fig. 1), which is based on a well-defined XML DTD. This specification may be automatically generated if the network topology is designed with the VNUML graphical user interface tool [26]. Finally, in the *execution phase*, the VNUML parser (written in Perl language) processes the specification of the scenario (third step in Fig. 1) and automatically launches the topology containing the virtual machines and specified networks (fourth step in Fig. 1).

As shown in Fig. 2, VNUML launches a window for each node, representing a Linux console which allows users to interact with virtual machines: Establishing a proper configuration, launching or stopping applications and/or daemons, executing commands, and so on. Moreover, the tool offers the possibility of executing different commands in the virtual machines defined in the scenario file. Furthermore, VNUML also allows the real computer, where the simulation is executed (typically named *host*), to participate in the virtualized network topology and even allowing the simulated topology to connect with real Internet through the host. Once we have finished our interaction with the topology, the simulation can be stopped by invoking the parser. We can perform this process in two different modes: Either saving the current status of the simulation or purging any state configured in the scenario.

Since the definition and execution of computer network scenarios is quite simple, students' learning curve is shallow. Thus, they can focus more on the design of the virtualized topology than the tool itself (i.e., VNUML does not require users to know internal details about the system operation). For these reasons, we have applied VNUML for teaching different network concepts in our computer networks course, which is taught in the fourth course of the Computer Engineering degree. In the following section, we will comment on how we have developed an extended version called VNUML-UM, where we have installed new
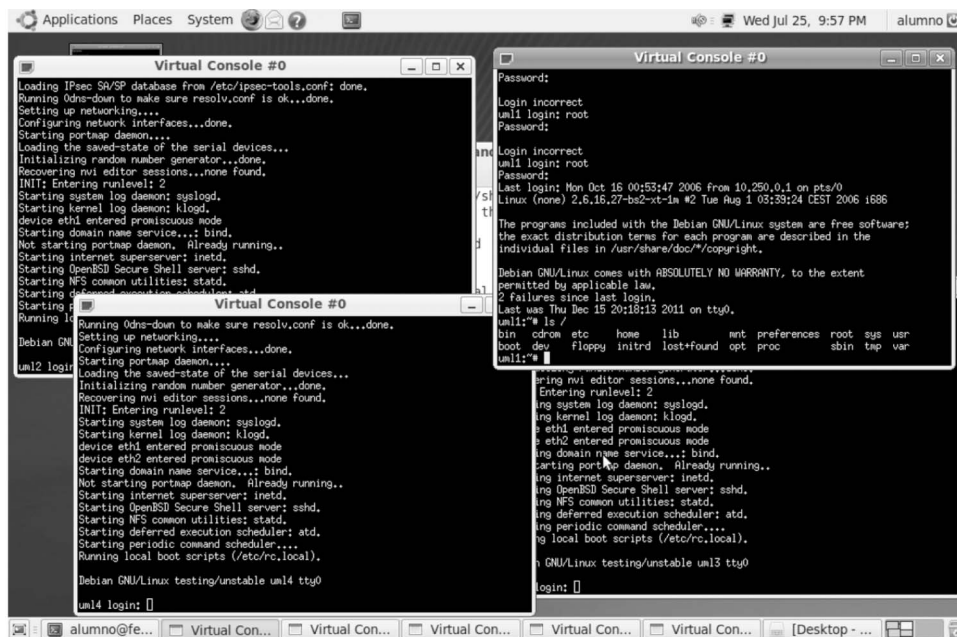
Fig. 2. VNUML in execution.

applications in the file system file and properly configured kernel modules to teach advanced network concepts.

3 COMPUTER NETWORKS SUBJECT

Next, we explain the context of the practical part of the computer networks subject, as well as the potentially necessary resources for its development.

3.1 Subject Description

The *computer networks* subject is located in the fourth course of the degree in *Computer Science Engineering* and is an annual subject taught through two four-month periods. The theoretical part of the subject (with a workload equivalent to 8 ECTS—*European Credit Transfer System*) is distributed in 10 different units that cover the following fundamental aspects: Dynamic routing, *Transmission Control Protocol* (TCP) operation and quality of service, addressing in IP versions 4 and 6, routing and mobility, high availability, load balancing, and related network security aspects. The practical part (with a workload equivalent to four ECTS) is intended to reinforce the most important theoretical concepts through the development of a computer network scenario that must accomplish a set of requirements and solve certain problems. By means of the VNUML tool, students are required to design and validate the correct operation of their proposed solution. In particular, in the practical part, three main topics are covered: Dynamic routing, IP mobility and, high availability, and load balancing.

These fundamental contents are developed through the proposal of various practical works to be solved by the students. These proposals are provided during the course in sync with the exposition of the related theoretical concepts. It is important to note that, unlike previous approaches followed in teaching computer networks, the practical works we propose to the students are not based on a preestablished computer network topology. Our approach is to specify a set of requirements used by the students to design the network topology, the elements integrating such topology and the proper configuration of those elements.

3.2 Laboratory Methodology

To teach the practical part of the subject, we organize several 1-hour laboratory sessions, which are distributed throughout the academic year. The objective of these laboratory sessions is to provide the student (using exposition-based lectures) a general overview about the different tools and configuration options available to resolve the practical works. More specifically, the sessions are distributed in the following manner:

- *Sessions 1, 2, 3, and 4*. The goal of these sessions is to explain the general use of the VNUML and help the student in becoming familiar with the environment.
- *Sessions 5, 6, and 7*. In these sessions, professors not only explain the software implementing dynamic routing protocols and details regarding its configuration but also discuss computer network designs associated to routing protocols.
- *Sessions 8 and 9*. In these sessions, we explain and discuss on the software and topologies that can be used by students to develop the practical work about load balancing and high availability.

Apart from these general sessions, we also offer students the possibility of interacting with professors to resolve doubts through online tools to facilitate and speed up the learning process [27]. In general, to increase both the comfort of students and the teaching quality, the laboratory sessions are composed of no more than 15 groups (30 students) and are distributed through three different turns in different days along the week.

Considering the three main topics covered by the subject in the theoretical part (discussed in Section 3.1), we propose three practical works to deal with them:

- *Practical work 1: Design of computer networks and routing.* In this practical work (related to sessions 5-7), the students are required to design a network topology integrated by a set of organizations satisfying certain requirements. Then, they validate their design by using different routing tools and configuring different dynamic routing protocols (RIPv1, RIPv2, RIPng, OSPFv2, OSPFv3, and BGPv4) within the network infrastructure. By means of this practical work, the students acquire a global comprehension of the operation of different routing protocols that are explained in the theoretical part of the subject: IPv4/IPv6 address assignment, computer network design, and dynamic routing.
- *Practical work 2: Autoconfiguration and mobility in IPv6.* This practical work allows the students to become familiar with certain relevant concepts of autoconfiguration and IPv6 mobility. In particular, the exercise consists of a sketch describing the configuration of these features on real scenarios. For this reason, the methodology we follow is based on the description of several steps that guide the student throughout the process and show the different configuration files and required commands. The student must verify the operation of the scenario proposed by the professor, answer a set of questions to probe the correct student's comprehension, and perform a set of requested extensions/modifications.
- *Practical work 3: High availability and load balancing.* This practical work (related to sessions 8 and 9) proposes the design of a high available and load balanced system that satisfies a set of well-defined requirements. Namely, it enables students a practical interaction with some theoretical concepts related to computer network design, high availability, and load balancing systems. In particular, students are required to configure redundant routers (according to the *Virtual Router Redundancy Protocol*—VRRP) and webservers, domain name system (DNS) and web proxies using caches and load balancers based on IP or cookies.

It is important to highlight that Practical Work 2 does not have associated sessions since both the concepts explained in the theoretical part and sessions 1-4 provide students the necessary elements to correctly develop the sketch and answer the different formulated questions. Finally, we would like to mention that the different scenarios required to carry out the aforementioned practical exercises are developed by means of the VNUML-UM distribution (see Section 4).

3.3 Students' Learning Process

In this section, we describe the process followed by students to acquire the knowledge and skills required in the subject. The successful application of our methodology requires students to assist the lectures where theoretical background is provided. A week after theoretical lessons have started, the first laboratory session is given.

The first four sessions are devoted to the explanation of the VNUML-UM virtualization tool: Definition of a topology, configuration of the nodes participating in the network scenario and interaction with a simulation. Once the initial sessions are completed, the specific sessions associated to the practical works are given in sync with the contents explained in theoretical lessons. In these sessions, professors explain the software installed in the file system file used by the virtual machines providing dynamic routing, autoconfiguration and IPv6 mobility, and high availability and load balancing. More precisely, in these sessions, professors explain how to configure this software and how it can be used to design and develop different computer network systems. In this process, we promote active learning by means of class discussion. Thus, we can relate different theoretical concepts to its practical use. When both theoretical concepts and specific laboratory sessions associated with a practical work are successfully given, the practical work is introduced. Each practical work consists on the development of a computer network system that satisfies a set of requirements on some of the theoretical concepts introduced.

During the course, students must develop three practical works. For the resolution of each practical work, students are organized in groups integrated by two members. Once the different laboratory sessions associated to a practical work are completed, students proceed in the following manner. First, they discuss the requirements and establish the design of the solution. This solution is, generally, presented to the teacher to get his feedback. Next, when the design is validated, they start with the development of the solution. This development can be made either during the weekly laboratory sessions scheduled for the subject that are not associated with the explanation of tools and concepts or during their personal work time either at home or within the laboratory (when is free and no classes are scheduled). Regardless of the way students proceed, professors are available to provide tutorship by means of interviews with them or through a virtual campus platform.

In general, in each group, students collaborate to produce the design of the system, the choice of the tools to develop the system, and configuration options for the development of the core elements by articulating their perspectives and negotiating their knowledge or beliefs [28]. However, they work cooperatively by sharing out some tasks when some specific issues that are similar have to be developed, e.g., the configuration of a routing protocol in two different autonomous systems. Finally, in general, they cooperate to produce the final report associated with the practical work. This report serves to reinforce their knowledge of the system, and discuss and justify again the different design solutions they have chosen for their system.

3.4 Students' Evaluation

In the computer networks subject the result of the students' learning process is evaluated in two different ways. On the one hand, by means of a theoretical exam. On the other hand, by means of the evaluation of the three practical works they are required to develop.

In the theoretical exam, we distinguish two parts. First, a test with 20 multiple choice questions (choosing the one answer out of three options). With the aim of reducing the possibility of guessing, we use a scoring rule which

subtracts points for incorrect responses. This rule is as follows: We give 1 point for a correct response, we subtract 1/(number of choices-1) points for each incorrect response, and we give no points if the question has not been answered [29]. These questions evaluate whether the student has acquired the basic knowledge on the topics of the subject. This knowledge is necessary to be able to solve the different problems contained in the second part of the exam. Thus, if the student does not obtain a mark of 10 points or higher, he/she is not able to take the second part of the theoretical exam and the exam is considered failed. Second, the exam contains three meaningful problems where the students have to put into practical works the knowledge acquired in the teaching-learning process.

Similarly to the development and evaluation of the three practical works, students are grouped in groups of two members. Each group has to develop the three proposed practical works. For each practical work, they must develop and implement the designed network scenario through VNUML-UM, to generate a report explaining how they have solved the practical work and the justification on the different decisions they have taken in the resolution. Additionally, each group has an interview with a professor. In this interview, each member of the group is individually evaluated by means of two aspects: A written test, in which the student has to answer to some questions about different issues arisen during the resolution of the practical work; and by making some tests and some changes on the scenario they have developed with the VNUML-UM system.

The final mark assigned by the student is based on the results obtained both in theoretical exam and the resolution of the practical works as follows: The theoretical exam is weighted as a 70 percent and the practical works are weighted with a 30 percent. The weight of each practical work is the following: 40 percent for Practical Work 1, 20 percent for Practical Work 2, and 40 percent for practical work 3. Furthermore, to proceed with the final mark calculation, students are required to pass both the theoretical exam and practical works, i.e., students must obtain at least five points over 10 in each one.

4 VNUML-UM: CUSTOMIZED VNUML DISTRIBUTION FOR THE PRACTICAL LEARNING OF COMPUTER NETWORKS

Based on VNUML version 1.7.3, we have created an enhanced VNUML distribution called VNUML-UM, which is available as a teaching resource [17]. In this distribution, we have extended both kernels and file system files of basic VNUML to include the modules and software necessary to perform the different practical works described in Section 3.2. It is important to point out that some of the tools included in VNUML-UM for IP-based mobility, high availability, and load balancing are powerful enough and are being used in professional environments. Thus, VNUML-UM could also be used in the professional design of networks. In fact, the use of VNUML in professional testbeds has already been proved [12].

Despite there exists newer VNUML versions (1.8.9 and 1.9.0beta8), for our laboratories we have opted by version 1.7.3. The reason is twofold. On the one hand, newer versions require more memory than versions 1.7 to launch a virtual machine, which obviously limits the number of virtual machines running in the same virtualized scenario. On the other hand, the new features included in the newer versions of VNUML do not improve either the aspects developed in our virtualized scenarios or the methodology followed in the practical part of the subject. In fact, these latest versions have mainly focused on the simplification of the language to describe the virtualized network topology and on the improvement of the *parser* used to launch the simulation, which has been proven to be already stable in version 1.7.3. Nevertheless, for those potential users with more powerful labs, we have also developed a customized version based on VNUML 1.8.9. The same action is planned for version 1.9 once a stable release is available.

Regarding the virtual machines, VNUML-UM includes support for Linux kernel version 2.6.16.27 and the filesytem version 0.4.1 of VNUML version 1.7.3. For VNUML 1.8.9 , the Linux kernel version 2.6.29 and the file system file version 0.6.0 are supported. The original files for these versions are available at the VNUML project webpage [18].

The different problems faced during the generation of this customized distribution of VNUML are derived from the fact that the Linux distribution provided by VNUML does not work in graphical mode. Thus, only software supporting command-line operation mode can be installed. Despite this feature allows executing a number of virtual machines without using an excessive amount of memory, it limits the different tools that we can incorporate to the system. For this reason, we had to make an in-depth search and analysis of the different tools satisfying this restriction. Additionally, to minimize the size of the virtual machine file system file, we carefully checked both the dependencies and the required disk space for the installation of each tool. When selecting the tools to be installed in the file system file, our intention was to achieve a tradeoff between the provided functionality and the required disk space. In fact, we discarded some tools because the functionality offered was similar to others while the size required for the installation was considerable. After the selection and installation process, we tested the operation and performance of each tool, paying special attention to the memory needs during the execution. This was an analysis of paramount importance because we identified some tools not able to operate with the default virtual machine memory configuration. For this reason, we carried out an exhaustive study to determine an optimal configuration allowing a fluent operation with the virtual network scenario.

From the VNUML-UM distribution we could develop a centralized system, a web-based system or include this new distribution in a virtualized system, what would prevent students from installing the software. Thus, the new distribution is the building block for creating any kind of system based on this functionality. In our case, we decided to use VNUML-UM as is instead of developing a centralized system because the performance is better when the students are developing the different labs.

In our case, in addition to the distribution and the installation available in the laboratories, we provided to our students with a virtualized solution based on NoMachine NX [30], which is basically a remote desktop. In this

solution, the students access a system with identical features to the system in the laboratories (same version of VNUML-UM installed in a Ubuntu operating system). Although this solution provides access by means a web browser without the installation of the software, it offers a worse performance because the different computers used to provide this functionality are not powerful enough. For this reason, we decided to install the software in each computer of the labs and, in general, the students have preferred this environment to use VNUML-UM in the NoMachine NX platform.

4.1 Tools Supported by VNUML-UM

The tools included in the VNUML-UM distribution can be grouped according to the provided functionality into: 1) Tools for routing and computer network configuration; 2) tools for IP-based mobility; and, 3) tools for high availability and load balancing. In the following, we provide a detailed description of the different software installed in VNUML-UM as well as the provided functionality.

4.1.1 Tools for Routing and Network Configuration

Since the VNUML file system file is based on a Linux system, it includes common tools to administrate both IPv4 and IPv6: *ifconfig*, *ping*, *ping6*, *traceroute*, *traceroute6*, *route*, *tcpdump*, and so on. Additionally, the *Router Advertisement Daemon* (radvd) is also included to support the route advertisement (performed by routers) that enables the IPv6 autoconfiguration process.

For supporting the different routing protocols, the VNUML project relies on the *Quagga* routing software (version 0.99.5) which implements the following protocols: *RIPv1*, *RIPv2*, *RIPng*, *OSPFv2*, *OSPFv3*, *BGP-4*, and *BGP-4+*. Therefore, we can propose practical exercises related to *interior gateway protocol* (IGP) and *exterior gateway protocol* (EGP). The different OSPF versions support the configuration of the different areas defined in the protocol specification. Similarly, the BGP implementation supports features such as prefix-based and attribute-based filtering, private autonomous systems, prepending, BGP attribute modification, multihoming, and so on.

4.1.2 Tools for IP-Based Mobility

VNUML-UM employs the *mobile IPv6 for Linux* (MIPL) package to deploy scenarios based on IPv6 mobility. This software implements the *mobile IPv6* protocol and requires the use of the previously mentioned *radvd* daemon. MIPL supports advanced features such as routing optimization between mobile nodes and correspondent nodes, or the use of security options (IPsec).

4.1.3 Tools for High Availability and Load Balancing

Unlike the aforementioned group of tools, which are already included in the file system file provided by the VNUML project [18], we have installed the software necessary to develop the practices dealing with high availability and load balancing. Given that the Linux virtual machines in the simulation do not work in graphical mode, only new software supporting command-line operation mode can be installed. Nevertheless, we have found a wide amount of tools providing high availability and load balancing capabilities supporting this mode. In the following, we enumerate the installed tools grouped into different categories according to the specific functionality provided. Note that some tools may appear several times if they can be used for different purposes:

- Tools for network redundancy. With this purpose, we have installed two different tools supporting the VRRP protocol for router redundancy: *vrrpd* and *keepalived*. Additionally, the VRRP protocol can be also used for load balancing between routers within a specific network.
- Proxy DNS tools that allow both to reduce the number of DNS requests sent to authoritative servers and to check the availability of remote DNS servers: *DNRD*, *DNSProxy*, *dproxy*, *dnsmasq*, and *pdns*.
- DNS resolution cache, DNS server backup, support of different DNS forwarder servers, and load balancing among a set of local DNS servers: *squid*, *DNS Balance*, and *DNRD*.
- TCP and Layer 7 load balancing: *Balance*, *balanceNG*, *pen*, *penbw*, *pound*, *haproxy*, *LVS*, *lvs-kiss*, and *keepalived*.

4.2 Using the VNUML-UM Virtualization Tool

In this section, we describe various aspects which should be taken into account when using the VNUML-UM distribution for teaching network-related concepts. First, it is worth noting the installation of VNUML in the Ubuntu distribution is quite simple and the process is partially automated. For this reason, numerous groups of students opt for installing the VNUML-UM distribution provided in [17] in their PCs at home, so that they can gain flexibility when they work outside the university premises. Furthermore, given that we have been employing this virtualization tool several years, we can affirm that the installation process has been simplified in the most recent versions.

An important aspect related to the VNUML-UM operation relies on the permissions that need to be granted to students to properly employ the different virtualization features and, ultimately, develop the practical exercises explained in Section 3.2. Typically, the policy followed by the laboratories administrators is to not granting administrator permissions (*root*) to the student's personal account. This situation represents a problem when developing virtualized scenarios based on VNUML-UM that need to interact with the real machine (*host*). The reason is that certain functionalities (e.g., the management network used for facilitating the launching process of the scenario by automatically executing certain commands in the virtualized hosts) require administrator-level permissions in the host to carry out some configuration actions. To solve this problem, we propose the employment of the *sudo* utility (available in Linux systems) which can be used by the administrator to specify that certain scripts can be executed by other users, not owning administrator privileges, as if they were the administrator.

As a consequence of including support for a wide variety load balancing tools, compared with the original file system file provided by the VNUML development team, our extended file system file requires more disk space (about 800 MB). Nevertheless, this is a reasonable size
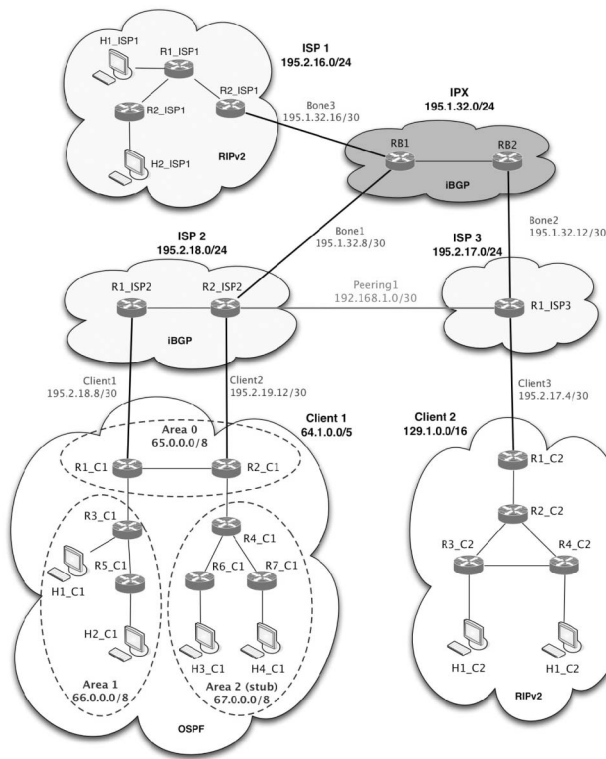
Fig. 3. Example of scenario for Practice 1.

Fig. 4. Example of scenario for Practice 3.

considering the complete set of load balancing software that has been installed. Furthermore, this size does not affect the VNUML performance when virtualizing network scenarios since, thanks to the *copy-on-write* optimization, a single copy of the file system file is shared by all the virtual machines and only modifications to the basic file system file are saved for each one.

4.3 Network Scenarios Developed with VNUML-UM

As an example of the power that VNUML-UM can provide us for the testing and validation of computer networks scenarios, we show some topologies developed during the courses. For example, Fig. 3 depicts a representative topology for a scenario developed in Practical Work 1. In this scenario, there exists different autonomous systems that are connected through different links using both peering and transit configurations. Additionally, we observe that the autonomous systems *Client1* and *Client2* are based on different routing protocols. More precisely, while the former is organized in several OSPF areas, the latter uses RIP as routing protocol.

In Fig. 4, we show a scenario implemented by students according to the requirements demanded in Practical Work 3. As observed, the network topology is integrated by two organizations: *Service provider* and *client's organization*. Both organizations deploy their own DNS system with master and backup DNS servers. The service provider provides high available web services achieved by means of multiple webservers and a load balancer distributing requests among them. These webservices are accessed by users belonging to the client's organization, where a set of web proxy servers is installed to minimize the number of accesses.
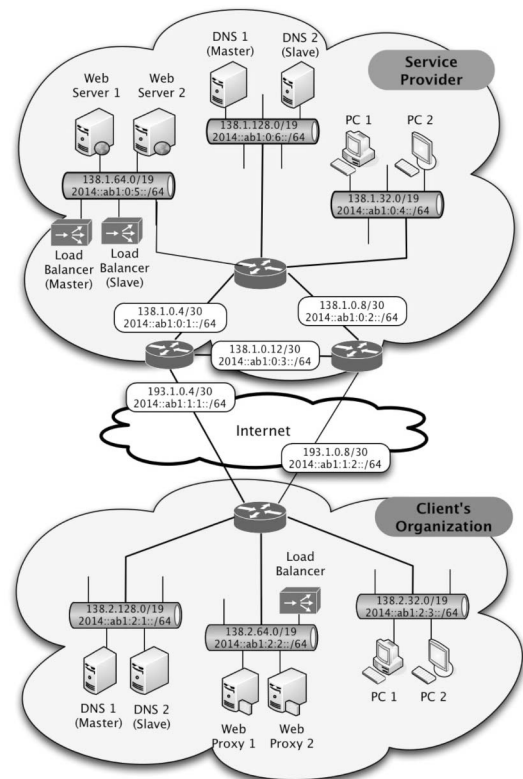
5   RESULTS AND EVALUATION

In this section, we present the different results that the students have obtained with the methodology and activities we propose throughout three academic courses: 2009-2010, 2010-2011, and 2011-2012. The results are based on the evaluation method presented in Section 3.4. We have also made an opinion poll between the students to know whether they consider useful the tools and the different practical works we have proposed to facilitate the understanding of the different concepts taught in theoretical lessons.

5.1   Analysis of Marks Obtained by Students

Table 2 depicts a comparison of the different marks obtained by the students in the three practical works during the different academic courses. The mark is shown in a numerical interval between 0 and 10 points. We consider that a student passes a practical work if he/she has obtained a mark equal or greater than five points. In the table, we can appreciate a decrease in the number of students who takes each course. We would like to clarify that this variation is provoked by a transition period caused by this degree to be adapted to the *European Credit Transfer and Accumulation System* (ECTS) according to the *Bologna process*. Students can decide to adapt themselves to the new degree or finish their studies according to the current degree. Since some students have opt for entering into the new degree, the number of students in this subject has decreased slightly.

From the analysis of the marks obtained by students in the different practical works, we observe some fluctuation values (specially in practical work 3) during the first course (2009-2010). From our experience, we can affirm that these

TABLE 2
Labs Marks

Course \ Issue	2009-2010	2010-2011	2011-2012
Number of evaluated students	45	25	21
Practical Work 1			
Marks (7-10)	38	22	17
Marks (5-7)	7	3	4
Marks (<5)	0	0	0
Marks 7-10 (%)	84,4	88	81
Practical Work 2			
Marks (7-10)	45	25	19
Marks (5-7)	0	0	2
Marks (<5)	0	0	0
Marks 7-10 (%)	100	100	90
Practical Work 3			
Marks (7-10)	20	23	15
Marks (5-7)	25	2	5
Marks (<5)	0	0	0
Marks 7-10 (%)	44,4	92	75

TABLE 3
Relationship between Passing Practices and Theory

Course \ Issue	2009-2010	2010-2011	2011-2012
Theoretical part passed with practices passed			
Marks (7-10)	12	10	9
Marks (5-7)	24	8	6
Passed (%)	84,1	81,8	83,3
Passed (7-10) (%)	27,3	45,4	50
Theoretical part passed before passing practices			
Marks (7-10)	4	2	1
Marks (5-7)	3	2	2
Passed (%)	15,9	18,2	16,7
Passed (7-10) (%)	9,1	9,1	5,5

variations are derived from the application of our methodology for the first time, which is a process prone to unexpected problems and where reiterative adjustments are necessary. Particularly, to overcome the different problems we detected during the first year, we applied the feedback received from our students and added more examples of usage, documentation related to the different tools installed in VNUML-UM, and a better guidance to the students. These actions had a positive effect in subsequent course (2010-2011) where students improved their marks. For example, in practical work 3, the percentage of marks between 7 and 10 points passed from 44.4 percent (course 2008-2009) to 92 percent (course 2011-2012).

For this reason, we consider that the results of the courses 2010-2011 and 2011-2012 are more representative since they are obtained when applying a stable methodology overcoming the different problems detected during the first academic year. In this regard, we observe that the marks the students obtained show a high level of performance. More precisely, the mean percentage of students who received a mark between 7 and 10 points in practical work 1, practical work 2, and practical work 3 were ≈ 85, ≈ 95, and ≈ 84 percent, respectively. Finally, it is also important to note that all the students performing the different practical works have passed it (Mark ≥ 5 points). This is due to they can check with VNUML-UM whether their scenarios are properly configured and solve the different proposed problems. Thus, until they are not sure that the computer network scenario works properly, they do not present it to the teacher for evaluation.

To determine if our methodology and the VNUML-UM tool really help students in the practical learning of computer network concepts, we have also carried out an additional analysis of the students marks in the theoretical part of the subject. In particular, we verify if there exists a relationship between passing the practical and theoretical parts of the subject. For the students passing both parts, in Table 3 we show the marks obtained in the theoretical exam organized in two groups: The students who have passed the

theoretical part once they have developed the different practical works, and the students who have managed to pass the theoretical exam without making practical works. From the results, we conclude that the students who have passed the theoretical part without carrying out the practical works is reduced (15.9 percent in course 2009-2010, 18.2 percent in 2010-2011, and 16.7 percent in 2011-2012). Thus, the students who pass the theoretical part having made practical works before or at the same time that the theoretical exam is over 81 percent. Furthermore, we can affirm that our methodology helps students in the acquisition of knowledge related to computer network concepts because the percentage of students who pass the theoretical part with good marks (between 7 and 10 points) has increased from 27.3 percent in course 2009-2010 to 50 percent in the last course 2011-2012.

Therefore, we might conclude that the work developed in the practical works with VNUML-UM contributes to improve the learning process, or at least, the students consider that is more valuable to make them. To confirm this hypothesis, we performed an opinion poll to obtain feedback from the students. The obtained results are presented in the following section.

5.2 Students Evaluation of the Proposed Methodology

Apart from the conclusions extracted from the analysis of the students' marks, we also decided to perform an opinion poll to know students' perception and whether the different practical works had allowed them to understand better the different theoretical concepts. Their feedback is essential to enhance our work through a continuous improvement process. In this opinion poll, 15 students participated in course 2009-2010, whereas 10 students did it in course 2010-2011. Finally, the number of participants was 20 in 2011-2012.

The opinion poll consisted of different questions (see Table 4) related to the learning of computer networks concepts through our labs and the use of VNUML-UM. In particular, students were required to evaluate the following aspects: Question 1 (Q1) asked students whether the different practical works had allowed them to improve their understanding of the different theoretical concepts related to the network design. Questions 2 and 3 (Q2 and Q3) were related to the understanding of routing protocols

TABLE 4
Questions Related to the Students' Learning through the Labs

Code	Question related to
Q1	Improvement of the understanding of network design
Q2	Better understanding of IGP
Q3	Better understanding of EGP
Q4	Consideration of VNUML as useful tool to understand network concepts
Q5	Better understanding of mobile IPv6 concepts
Q6	Better understanding of load balancing and high availability concepts

(both internal—Q2—and external—Q3—routing protocols). Question 4 (Q4) evaluated the use of VNUML-UM. Finally, questions 5 and 6 (Q5 and Q6) aimed to know whether our methodology and tools were useful for the learning process of some advanced concepts, such as mobility, load balancing, (Q5) and high availability (Q6).

Our students evaluated each question by choosing a value between 0 and 10, where 0 express a negative evaluation and 10 an excellent evaluation of a certain aspect. The students' evaluation obtained through the different questions are shown in Fig. 5. Apart from this

information, we have calculated mean, standard deviation (SD), and the 95 percent confidence interval (CI) values, which are depicted in Table 5. Results show that our students are highly satisfied with the contribution of the practical works in its learning process of advanced computer network concepts. In fact, the results of the first three questions (Q1-Q3) were as expected because virtualization has been generally used to teach these concepts, as explained in Section 2.

According to question Q4, our students consider that VNUML-UM is an adequate tool for the virtualization of computer network scenarios, specially to teach advanced network concepts such as Mobile IPv6, load balancing, and high availability.

From the results obtained in questions Q5 and Q6, we can conclude that our students consider the learning process with VNUML-VM more useful for mobility, load balancing, and high availability than using it for network routing. In fact, they got better marks in the theoretical part when they perform the practical works beforehand (see Table 3).

Finally, it is worth mentioning that, apart from the data obtained in the opinion poll, we have received students' feedback personally and, in general, they mentioned us that, thanks to the practical works, they have managed to
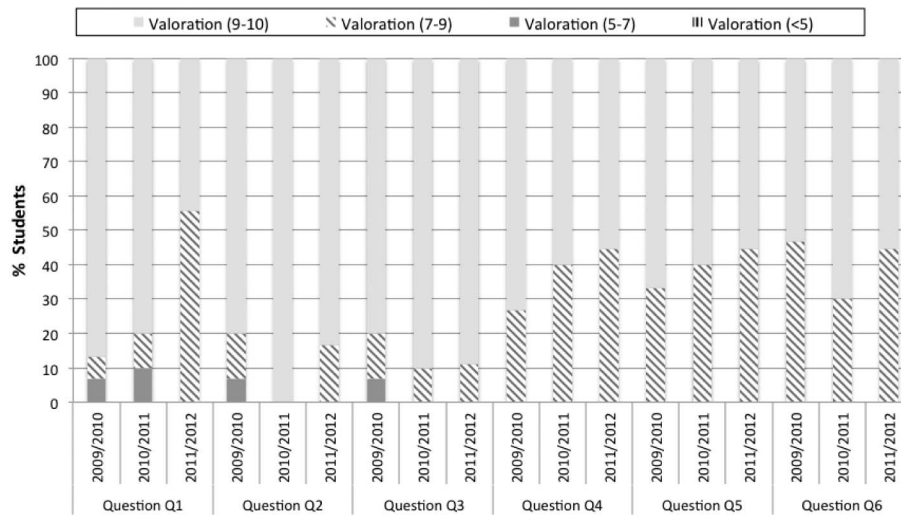

Fig. 5. Questions valoration.

TABLE 5
Detailed Pool Results

Course / Issue	2009-2010	2010-2011	2011-2012	2009-2010	2010-2011	2011-2012
Question Code	Q1			Q2		
Mean	8,7	9,1	8,1	8,6	9,5	8,9
SD	0,8	0,6	1,9	0,9	0,5	2
CI	0,4	0,3	0,9	0,5	0,3	0,9
Question Code	Q3			Q4		
Mean	8,6	8,9	8,8	8,4	8,5	8,5
SD	0,9	0,7	2	0,8	0,9	2
CI	0,5	0,4	0,9	0,4	0,6	0,9
Question Code	Q5			Q6		
Mean	8,3	8,6	8,5	8,2	8,6	8,5
SD	0,9	0,5	2	0,9	0,7	2
CI	0,5	0,3	0,9	0,5	0,3	0,9

understand some concepts that initially were too abstract for them. This feedback has been verified by the anonymous opinion pools made in the diverse courses.

6 CONCLUSIONS AND FUTURE WORK

Virtualization is useful in computer science education because it allows a better availability and management of resources to set up laboratories for the students. Up to now, virtualization has been used mainly for teaching administration of operating systems and the basic concepts of computer networks. In this paper, we have presented how virtualization can be used not only to teach these basic concepts but also advanced aspects of computer networks such as mobility, load balancing, and high availability. Namely, we present a new teaching resource called VNUML-UM, how we teach these concepts with it in a computer network course, the used methodology, and the different results we have obtained. VNUML-UM has been used during several academic courses, where we have obtained successful results.

The main conclusion extracted from our experience is that both basic and advanced computer network concepts can be taught successfully with the developed VNUML-UM distribution. This fact has been contrasted with the increase in the number of students that have passed the theory when they have made the practical works (before or at the same time) and with better marks. Other important aspect that has endorsed our approach is the students' opinion on their learning process and how this tool and the labs have contributed to improve their knowledge on the different computer network concepts taught in lectures. Another relevant conclusion relies on how our results demonstrate that VNUML-UM distribution is a virtualization tool valid for teaching advanced computer network concepts, which is an issue not covered so far by other experiences.

ACKNOWLEDGMENTS

REFERENCES

[1] I.A. Garcia, C.L. Pacheco, and J.N. Garcia, "Enhancing Education in Electronic Sciences Using Virtual Laboratories Developed with Effective Practices," *Computer Applications in Eng. Education,* http://dx.doi.org/10.1002/cae.20554, 2011.

[2] S. Nanda and T.-C. Chiueh, *A Survey of Virtualization Technologies,* Technical Report TR-179, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.74.371, 2005.

[3] J. White and A. Pilbeam, "A Survey of Virtualization Technologies with Performance Testing," *The Computing Research Repository,* http://arxiv.org/abs/1010.3233, Oct. 2010.

[4] J. Keengwe and L.O. Anyanwu, "Computer Technology-Infused Learning Enhancement," *J. Science Education and Technology,* vol. 16, no. 5, pp. 387-393, June 2007.

[5] D. Dobrilovic and Z. Stojanov, "Using Virtualization Software in Operating Systems Course," *Proc. Int'l Conf. Information Technology: Research and Education (ITRE '06),* pp. 222-226, May 2006.

[6] A. Gaspar, S. Langevin, and W.D. Armitage, "Virtualization Technologies in the Undergraduate IT Curriculum," *IT Professional,* vol. 9, no. 4, pp. 10-17, 2007.

[7] G.D. Steffen and H.I. Abu-Mulaweh, "Teaching Local Area Networking in a Secure Virtual Environment," *Computer Applications in Eng. Education,* vol. 18, no. 3, pp. 547-554, 2010.

[8] A. Gaspar, S. Langevin, W. Armitage, and M. Rideout, "Enabling New Pedagogies in Operating Systems and Networking Courses with State of the Art Open Source Kernel and Virtualization Technologies," *J. Computer Small Colleges,* vol. 23, no. 5, pp. 189-198, 2008.

[9] D. Dobrilovic, V. Brtka, I. Berkovic, and B. Odadzic, "Evaluation of the Virtual Network Laboratory Exercises Using a Method Based on the Rough Set Theory," *Computer Applications in Eng. Education,* vol. 20, no. 1, pp. 29-37, http://dx.doi.org/10.1002/cae.20370, 2012.

[10] D. Dobrilovic and B. Odadzic, "Virtualization Technology as a Tool for Teaching Computer Networks," *World Academy of Science, Eng. and Technology,* vol. 13, no. 13, pp. 126-130, Jan. 2006.

[11] L. Fàbrega, J. Massaguer, T. Jové, and D. Mérida, "A Virtual Network Laboratory for Learning IP Networking," *Proc. Seventh Ann. Conf. Innovation and Technology in Computer Science Education (ITiCSE '02),* pp. 161-164, http://doi.acm.org/10.1145/544414.544462, 2002.

[12] F. Galán, D. Fernández, W. Fuertes, M. Gómez, and J.E.L. de Vergara, "Scenario-Based Virtual Network Infrastructure Management in Research and Educational Testbeds with VNUML," *Annals of Telecomm. - Annales Des Tcommunications,* vol. 64, nos. 5/6, pp. 305-323, http://dx.doi.org/10.1007/s12243-009-0104-3, 2009.

[13] F. Galán and D. Fernández, "Use of VNUML in Virtual Honeynets Deployment," *Proc. Actas de IX Reunión Española sobre Criptología y Seguridad de la Informaci (RECSI '06),* pp. 600-615, Sept. 2006.

[14] M. Pollitt, K. Nance, B. Hay, R.C. Dodge, P. Craiger, P. Burke, C. Marberry, and B. Brubaker, "Virtualization and Digital Forensics: A Research and Education Agenda," *J. Digital Forensic Practice,* vol. 2, pp. 62-73, Apr. 2008.

[15] S.C. Nemesio, P. De Las Heras Quirs, E.M.C. Barbero, and J.A.C. González, "Early Experiences with NetGUI Laboratories," *Proc. Simposio Internacional de Informática Educativa (SIIE '06),* Oct. 2006.

[16] R. Nabhen and C. Maziero, "Some Experiences in Using Virtual Machines for Teaching Computer Networks," *Proc. IFIP 19th World Congress Education for the 21st Century - Impact of ICT and Digital Resources,* D. Kumar and J. Turner, eds., vol. 210, pp. 93-104, http://dx.doi.org/10.1007/978-0-387-34731-8_11, 2006.

[17] A. Ruiz-Martínez, F. Pereñiguez, R. Marín-López, P.M. Ruiz, and A.F. Gómez-Skarmeta, "VNUML Kernel and Filesystem for Teaching Advanced Computer Network Concepts," http://ants.inf.um.es/arm/computernetworks/vnuml-um.html, 2013.

[18] VNUML Web Server, "Virtual Network User Mode Linux (VNUML)," http://www.dit.upm.es/vnumlwiki, 2013.

[19] "MUNES - An Integrated Multiprotocol Network Emulator / Simulator," http://old.tel.fer.hr/imunes, 2013.

[20] B. Kneale, A.Y. De Horta, and I. Box, "VELNET: Virtual Environment for Learning Networking," *Proc. Sixth Conf. Australasian Computing Education (ACE '04),* pp. 161-168, 2004.

[21] "My Linux Network (MLN)," http://ostatic.com/mln, 2013.

[22] "Netkit," http://wiki.netkit.org/index.php, 2013.

[23] Dynamips, "Dynamips: Cisco 7200 Simulator," http://www.ipflow.utc.fr/index.php, 2013.

[24] M. Wannous and H. Nakano, "NVLab, A Networking Virtual Web-Based Laboratory that Implements Virtualization and Virtual Network Computing Technologies," *IEEE Trans. Learning Technologies,* vol. 3, no. 2, pp. 129-138, Apr.-June 2010.

[25] W.M. Fuertes and J.E. López de Vergara, "A Quantitative Comparison of Virtual Network Environments Based on Performance Measurements," *Proc. 14th HP Software Univ. Assoc. Workshop,* July 2007.

[26] M. Blanc, "Virtual Network User Mode Linux (VNUML)," http://sourceforge.net/projects/vnumlgui, 2009.

[27] J. MacDonald, *Blended Learning and Online Tutoring: Planning Learner Support and Activity Design.* Gower, Apr. 2008.

[28] C.M. Duh and L.R. Wen, "From Collaborative Argumentation to Knowledge Creation," *Proc. Conf. Information Management (ICIM),* Feb. 2002.

[29] Y. Bereby-Meyer, J. Meyer, and O.M. Flascher, "Prospect Theory Analysis of Guessing in Multiple Choice Tests," *J. Behavioral Decision Making,* vol. 15, no. 4, pp. 313-327, http://dx.doi.org/10.1002/bdm.417, 2002.

[30] "NoMachine NX Software," http://www.nomachine.com, 2013.

Antonio Ruiz-Martínez received the MS and PhD degrees in computer science from the University of Murcia. Since 2005, he has been an assistant professor in the Department of Information Engineering and Communications, University of Murcia, Spain. He teaches computer network architectures and security subjects in the degree of computer sciences. He also teaches in the degree of advertising and public relationships and in the MSc degree on new technologies in computer science. He also participates in teaching and in the coordination of the master studies in teacher training in secondary school and vocational training. He has been involved in different national and international research projects. His research interests include electronic commerce, payment systems, security, privacy, electronic government, and web services. He has published several papers in national and international conferences and journals. He is also a technical program committee member for various conferences and a reviewer for several international journals.

Fernando Pereñíguez-García received the BE, ME, and PhD degrees in computer sciences from the University of Murcia in 2007, 2008, and 2011, respectively. He is a postdoctoral researcher within the Department of Information and Communications Engineering, the University of Murcia, Spain. He has participated in several projects funded by the Spanish government and the European Union, such as SEISCIENTOS, SWIFT, or ITSSv6. Additionally, he has been involved in standardization activities within the IETF and has served as technical program committee for several international conferences. His research interests include security, privacy, and handoff management in mobile networks.

Rafael Marín-López received the BE, ME, and PhD degrees in computer sciences from the University of Murcia in 1998, 2000, and 2008, respectively. He is a full time assistant lecturer in the Department Information and Communications Engineering, the University of Murcia, Spain, from 2004. He teaches and coordinates the subjects advanced computer network architectures and telematic services and computer networks. Additionally, he is collaborating actively in standardization bodies such IETF and IEEE. His primary research interests include network access authentication, key distribution, and security in mobile networks. He serves as technical program member for several conferences and as a reviewer and editor for international journals.

Pedro M. Ruiz-Martínez received the BSc, MSc, and PhD degrees in computer science from the University of Murcia, Spain, in 1999, 2001, and 2002, respectively. He is an associate professor in Telematics in the Department of Information and Communication Engineering (DIIC) at the University of Murcia (UMU). In 2003, he was awarded a Ramón y Cajal research position by the Spanish MEC. He has also held postdoctoral research positions at ICSI in Berkeley, King's College London, and University of California at Santa Cruz. During these years, he acted as principal investigator for a number of research projects mainly funded by the European Union, Spanish government, and private companies, and has published a large number of refereed papers in international journals and conferences. He received an outstanding research trajectory recognition from the spanish MEC in 2007. He is on the editorial board for the *Ad Hoc Networks Journal*, *Computer Communications Journal*, *International Journal on Parallel, Emergent, and Distributed Systems*, *International Journal of Network Management*, and *International Journal on Smart Home*. He has organized several workshops on localized algorithms and protocols colocated with IEEE MASS, ACM MobiHoc, and IEEE DCOSS. He has also served on a number of Organizing and Technical Committees of highly relevant conferences such as ACM MobiCom, IEEE INFOCOM, IEEE MASS, ACM MobiHoc, and IEEE SECON. He also serves as a reviewer for major IEEE journals and conferences. His primary research interests include vehicular networks (VANET), sensor networks, mobile and ad hoc wireless networks, and distributed systems. He is a member of the IEEE, ACM, and IEEE Communications Society.

Antonio F. Skarmeta-Gómez received the MS degree in computer science from the University of Granada, and the BS (Hons.) and PhD degrees in computer science from the University of Murcia, Spain. Since 2009, he has been a full professor at the same department and university. He has worked on different national and international research projects, like Euro6IX, 6Power, Positif, Seinit, Deserec, Enable, and Daidalos. His primary interest is in the integration of security services at different layers like networking, management, and web services. He is an associate editor of the *IEEE Transactions on Systems, Man, and Cybernetics-Part B* and reviewer of several international journals. He has published more than 90 international papers and is a member of several program committees. He is a member of the IEEE.