International Journal of Engineering Education Vol. 33, No. 2(B), pp. 855–864, 2017
Printed in Great Britain

0949-149X/91 $3.00+0.00

Design and Deployment of Hands-on Network Lab Experiments for Computer Science Engineers*

ÁNGELES Mª MORENO MONTERO and DAVID RETORTILLO MANZANO
Department of Computer Science and Automation, University of Salamanca, Plaza de los Caídos s/n, 37008 Salamanca, Spain.
E-mail: amoreno@usal.es, davidretortillo@usal.es

Computer networks is an essential area in the training of computer science engineers. Delivery of networking laboratory experiments with specialized equipment is a challenge for computer networks teachers. The high cost of networking equipment makes it necessary to use simulation software to overcome the challenges of a large groups of students and working with complex topologies. This paper presents the first version of practical teaching materials to be used in networking courses for the Computer Science Engineering degree at the University of Salamanca (Computer Networks I and II). This material will allow students to work with real equipment in some cases and with simulation environments in others, maximizing the benefits of both.

Keywords: computer networks; networking laboratories; network simulation; network emulation; hands-on experiments; virtualization; cisco equipment; Packet Tracer (PT); GNS3; Cisco IOS

1. Introduction

The ACM/IEEE includes computer networks in the core body of knowledge for computer engineering [1]. Because the global job market demands well-trained professionals in this area, and more specifically in internetworking, universities everywhere offer specialised courses that follow the ACM/IEEE guidelines. Cisco press provides a good definition of interworking as ". . . *the industry, products, and procedures that meet the challenge of creating and administering internetworks. An internetwork is a collection of individual networks, connected by intermediate networking devices, that functions as a single large network*" [2]. The best example of an internetwork is the Internet. The deployment and maintenance of an internetwork requires practical experience in routing, switching, network applications, protocols, services and network security.

The best way to acquire this experience is through hands-on network lab experiments, and high quality practical exercises are essential for universities to produce work-ready networking graduates [3]. These hands-on labs are a necessary part of the educational process in order to improve not just student understanding of the subject but also their competences for the job market [4]. However, internetworking laboratories are costly in terms of both human and material resources. Their deployment and maintenance requires professional switches, routers, cables, mounting rack and several computers playing different roles in network applications (web, mail, cloud servers, agents and so on), as well as the technical staff needed to manage and maintain these laboratories. For this reason, network simulation tools represent a viable alternative to working with real equipment.

Deciding which tool to use also presents challenges. There are network simulation tools whose main purpose is as platforms for experimenting with network performance, generating traffic according to given models and parameters [5, 6]. Numerous work-focused virtual laboratories [7–9] also fall outside the scope of this work. In our case, the goal is to design classroom experiments without necessarily ruling out other possibilities. Here, we are more interested in those tools oriented to learning of networking concepts. There are several alternatives, as will be discussed later, but we consider the following to be essential features: (1) a GUI for designing the network topology; (2) easy setup of network equipment; (3) integrated traffic network analyser and (4) a learning curve compatible with the time constraints of our courses. Content design to complement and instantiate the concepts presented in lectures that can be performed by students within the available time and that contributes effectively to their training is another challenge we set ourselves here.

The rest of the paper is organized as follows. The teaching context within which this proposal was developed will be discussed along with our goals in section 2. Section 3 briefly analyses the different virtualization platforms and network simulators. Section 4 describes the methodology followed in the design of the laboratory guidelines. Section 5 details some hands-on lab experiments, and conclusions are presented in section 6.

* Accepted 13 December 2016.

2. Context and goals

The undergraduate degree in Computer Science Engineering at the University of Salamanca is a four-year program. In the networking stream, there are two level three subjects (Computer Networks I, Computer Networks II) and one at level four (Computer System Security). All of these are 60-hour (6 ECTS) mandatory subjects, with four teaching hours weekly during the semester (15 weeks).

The networking experiments utilized Netkit simulation software [10, 11] that emulates network devices such as Linux computers. Although it meets most of the educational objectives, this software has a number of limitations, principal of which is that the student is not working with either real devices or emulators of real internetwork equipment. For this reason, we decided to acquire real network equipment from one of the more recognized manufacturers, Cisco Networks. In light of financial limitations, mid-range equipment was acquired with a view to building a scale prototype that would enable us to validate our proposal and develop a scalable solution for the future networking laboratory.

Having validated our prototype [12], we consider how we might design a networking lab and associated experiments, taking account of the available technical and human resources and the number of students.

Table 1 summarizes student numbers for academic year 2015–2016.

Buying real internetworking equipment for this large number of students involves a significant financial commitment. Although this might be met, it would not solve the problem of recruiting technical staff for setup and maintenance. For that reason, the proposal presented here necessarily incorporates the use of network simulation tools; selecting the most appropriate of these in each case is one of the most important contributions of the present work.

Our objectives were as follows:

- Design hands-on lab experiments for internetworking training.
- Define a common structure for such experiments.
- Develop step-by-step experimental guidelines for students.
- Create predesigned network topologies as a starting point for experiments.
- Develop educational material for the instructor.
- Ensure that the user experience is as close as possible to working with real equipment.
- Select the best laboratory (real equipment or simulation tool) most appropriate to each experiment.
- Group materials into difficulty levels in terms of the previous experience needed by the student.
- Build a repository to make all the material accessible both to students and teachers of the University of Salamanca and to any other interested collective (engineers, network administrators, teachers and researchers in networking etc.).

Table 1. Number of students in academic year 2015–2016

Computer Networks I	140
Computer Networks II	152
Computer System Security	88

3. Virtualization platform and network simulator

When studying networks, one of the most important aspects of the subject is to perform laboratory practices that closely replicate real-life situations. We considered several possible tools for designing our experiment. The first option was to create a network laboratory using real hardware, which would be ideal for the student as they would work with real equipment and therefore with fully known problems and real solutions. Against this, the solution would entail an extremely high cost that would not be sustainable for large groups of students.

For this reason, network simulators and virtualization platforms were assigned a fundamental role in the study of computer networks. These allow network topologies to be created without the restrictions of a real network, such as financial costs or logistical problems in creating large-scale networks. A network simulator provides a global overview of the network and can verify how packets travel over the network. This makes it possible to assimilate information about the network and its operation in a much simpler form that in the case of a real network, which would require use of a network analyzer and several devices to obtain the same result. The main problem with network simulators is that they are not 100% exact, and commonly, not all router functions and commands are available on these simulators.

This leads to the third possibility: software emulators, which are programs that can emulate the functions of a particular primary hardware on a secondary and independent hardware platform. The main difference from simulators is that emulators require an operating system for each network element (for example, a router operating system). This brings the student closer to reality, as they see the same functions as in real equipment (i.e., the router's operating system is the same). Emulators also allow real equipment or virtual machines to be connected to the network, so linking the first possibility (real hardware) with the third (emulated hardware).

Of the many available software tools for networking [13], we will focus on two that relate directly to Cisco equipment—Packet Tracer and Graphical Network Simulator-3 (GNS3)—with a view to linking the work with our real hardware. Packet Tracer [14] is a Cisco network emulator that approaches the real user experience of computer networks. It includes a graphical user interface (GUI) for configuring network devices and is suitable for inexperienced users. Packet Tracer incorporates a graphical simulation and traffic analyzer, enabling observation of each packet and how it travels across the network. However, advanced users may consider it limited in its implementation of some services, such as DNS and routing multicast.

GNS3 [15] is a free software emulator for networks that is open-source and multi-platform. It allows simulation of complex networks and is quite accurate because of its link with external software such as Dynamips (Cisco IOS emulator), Dynagen (text-based front-end for Dynamips), QEMU (generic machine emulator, open-source and virtualizer) and VirtualBox (free virtualization software). GNS3 allows emulation on multiple platforms of Cisco routers (IOS operating system images) and firewalls and connection of simulated and real networks. In addition, it has built-in Wireshark [16], one of the most popular network analyzers (along with tcpdump), enabling capture and monitoring of network traffic. In designing the practical materials, we chose one or other tool, principally according to student skill level and previous handling experience.

4. Design of practical materials

In this section, we describe the design of the educational materials and the phases of their elaboration summarized in Table 2. In the initial phase, we selected the topics to be treated and their priority. Next, we defined their objectives and competencies (skills to be acquired by the student at the end of the experiment). The third step provides the necessary theoretical knowledge. The fourth step, perhaps the most important (as it determines the others), relates to the selection of the tool with which the experiment is designed. After selecting the tool, a simple example will be developed with the double goal of understanding the concept and serve as training in the use of the tool.

Next, we elaborated the guidelines in two versions—one for the student and one for the teacher. The student version includes practice targets, scenarios and/or predesigned configurations for the selected tool, step-by-step descriptions and assessment tests. The teacher version details all the procedures for configuring the equipment for the situation analyzed in each case and the answers to all the questions.

Table 2. Steps in the design of experiments

1. Select topics and assign difficulty level.
2. Objectives and competencies.
3. Development of theoretical concepts.
4. Selecting the tool: network simulator, network emulator or real hardware.
5. Development of practical examples (exercises performed with the selected tool).
6. Guidelines for students
   6.1. Development of scenarios and/or predesigned configurations for the selected tool
   6.2 Step-by-step guide
   6.3. Evaluation questions
7. Guidelines for teacher
   7.1. Solution development (scenarios and/or configurations scripts)
   7.2. Suggestions about solving practice and evaluation
8. Estimated completion time.
9. References.

The teaching materials were grouped into three levels of difficulty, based on the knowledge required to solve the questions raised. The first set of materials relate to the basic level, in which students work with concepts like collision domain, frame forward/filter, IP address, IP subnetting and static routing. The second set of materials can be characterized as mid-level, working with essential Internet services like DNS and DHCP, dynamic routing (RIP and OSPF) and the new version of the IP protocol, IPv6. Finally, a third set of advanced materials includes learning about virtual local area networks (VLANs) and voice over IP (VoIP).

5. Hands-on lab experiments

The lab experiments cover the following topics: hubs and switches, IP addressing, static routes, dynamic routing (RIP and OSPF), IPv6, DNS, DHCP, multicast routing (PIM-DM and PIM-SM), VLAN, ACL and VoIP. In this section, we provide one example of the experiments realized at each level. As an example of the basic level material, we detail the hubs and switches experiment developed for Packet Tracer. For the second level, we describe the multicast routing experiment (implemented with GNS3). Finally, we demonstrate work with real hardware in a prototype integration of voice and data.

5.1 Hubs and switches

In the hubs and switches experiment, the student must work with concepts such as collision domain, MAC tables and frames forward and filter. The proposed network topology is shown in Fig. 1. The student has to draw the topology in Packet Tracer, assign IP addresses and observe in simula-

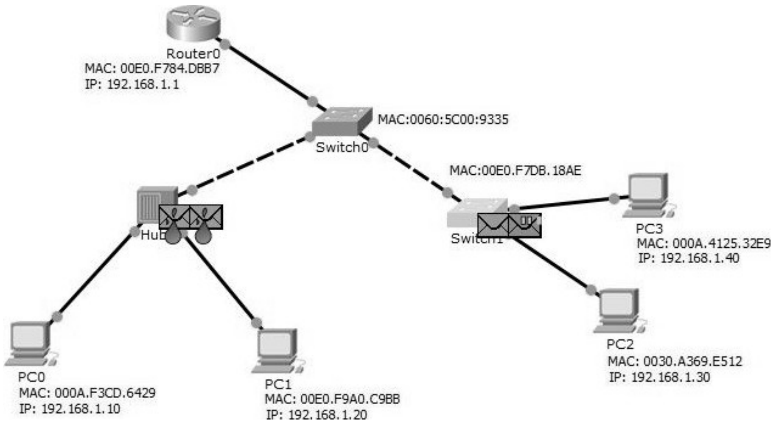

Fig. 1. Example of a first level lab experiment: working with hub and switches.

tion mode how the switches learn the MAC directions. The student must also identify collision domains.

This experiment is designed to be carried out by students with little or no experience of internetworking. As one of their first exposures to network simulation tools, we chose PT for this experiment because of its very easy graphic user interface (GUI), and especially for its simulation mode, which allows one to see step by step how packets travel across the network. In this way, students can see and so more easily understand what is actually going on in the network.

5.2 Multicast routing

The lab experiment of multicast routing protocols, specifically PIM-SM, is one example of the second-level materials. It is designed in GNS3, which provides a GUI for drawing the topology, but configuration of equipment must be done from the command line interface (CLI). This should not be a problem, as the students will already have acquired some experience of the IOS command from the Cisco router, in the previous experiments realised in PT. GNS3 allows one to work with routers that support multicast routing (PT does not incorporate any router with this feature) and, although it does not include a simulation mode step by step, it does allow use of the network packets analyzer Wireshark at points of interest. This enables observation of traffic at individual links rather than all together as in PT. The advantage is that it is possible to use in a native way a more widely used and more complete sniffer like Wireshark. Using this analyzer, students

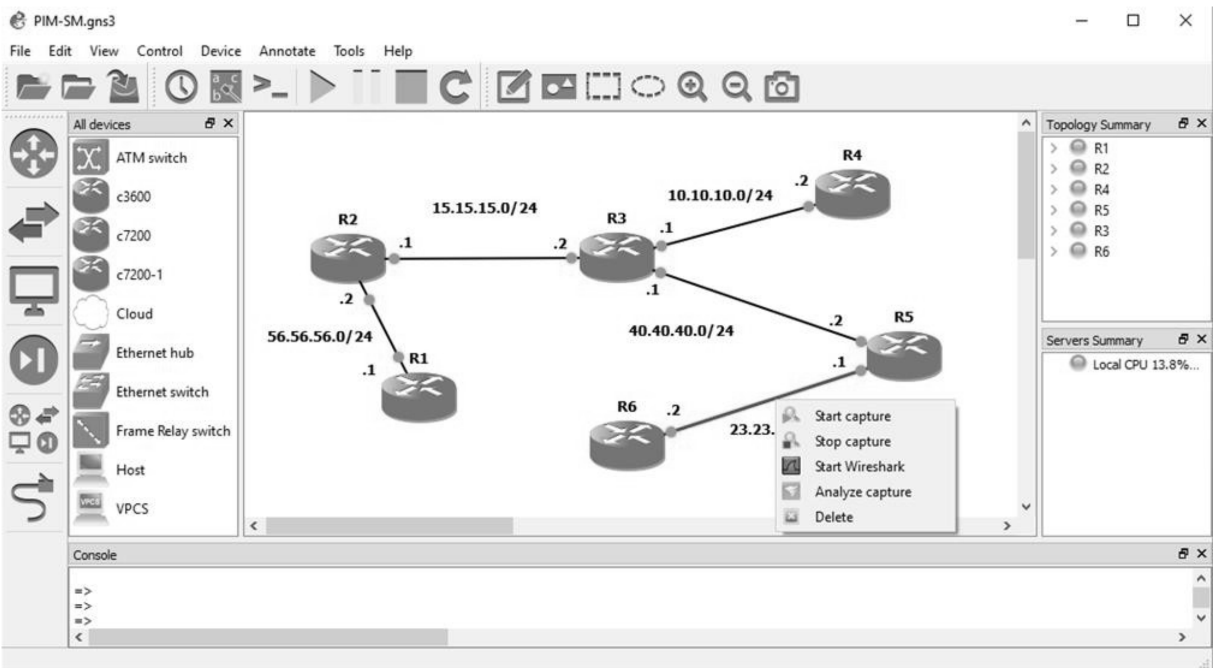

Fig. 2. Second level: working with PIM-SM in GNS3.

Table 3. Configuring multicast routing PIM-SM from a router console

```
conf t
ip multicast-routing
int fa0/0
        ip pim sparse-mode
int fa1/0
        ip pim sparse-mode
int fa2/0
        ip pim sparse-mode
end
```

acquire practical experience in the use of sniffers that can be applied in real environments.

The network topology proposed is shown in Fig. 2. IP addresses have already been assigned, and dynamic routing is enabled. The student's work focuses on configuration of the multicast routing protocols. The IOS commands enabling the protocol multicast PIM-SM for router interfaces are shown in Table 3.

Once students complete the configuration, they can observe the operation of PIM-SM. The protocol messages on the network link of interest will be analysed with Wireshark. Fig. 3 shows how Wireshark enables analysis of the details of a join message to multicast groups 224.0.1.40 and 239.20.20.20.

5.3 Voice over IP (VoIP)

From the third-level materials, the example below involves a simple scenario on PT, designed to equip students with the skills and experience necessary to work with experiments elaborated in GNS3 or directly with real hardware.

The configuration of a VoIP system in PT must be realized from the command line interface (CLI). The steps for its correct settings are as follows (see Table 4):

1. Configure the Cisco Unified Communications Manager Express (CME)
2. Define VLAN for the voice traffic so that voice traffic will travel separately from data traffic.
3. Configure the telephone numbers.

VLAN (Virtual Local Area Network) is an advanced topic also included in the experiments designed for level three. PT GUI enables the student to see the communication between both phones (Fig. 5) and to analyse in simulation mode the protocols involved in the voice traffic (Fig. 4).

Along with the VLANs, this experiment makes network simulation the perfect training environment for working with more complex network topologies that mix voice and data and also work with real hardware. This experiment is the most complex of all those elaborated, as we describe below.

5.4 Integration of voice and data traffic

To perform this set of experiments, we used one Cisco 1861 router, two Cisco 861 routers and three Cisco 6911 IP telephones (Fig. 6). Before attempting these experiments, the student has already completed previous lab experiments with IP addresses,

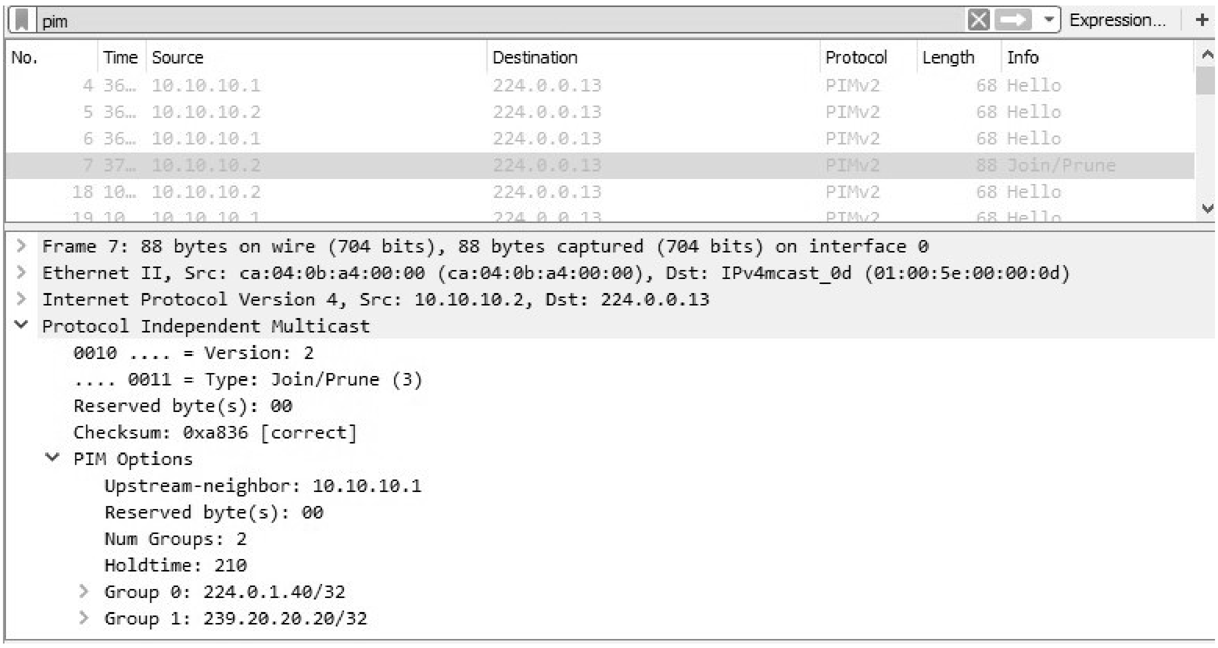

Fig. 3. Link between the routers R3 and R4, the Join/Prune PIM-SM message detail.

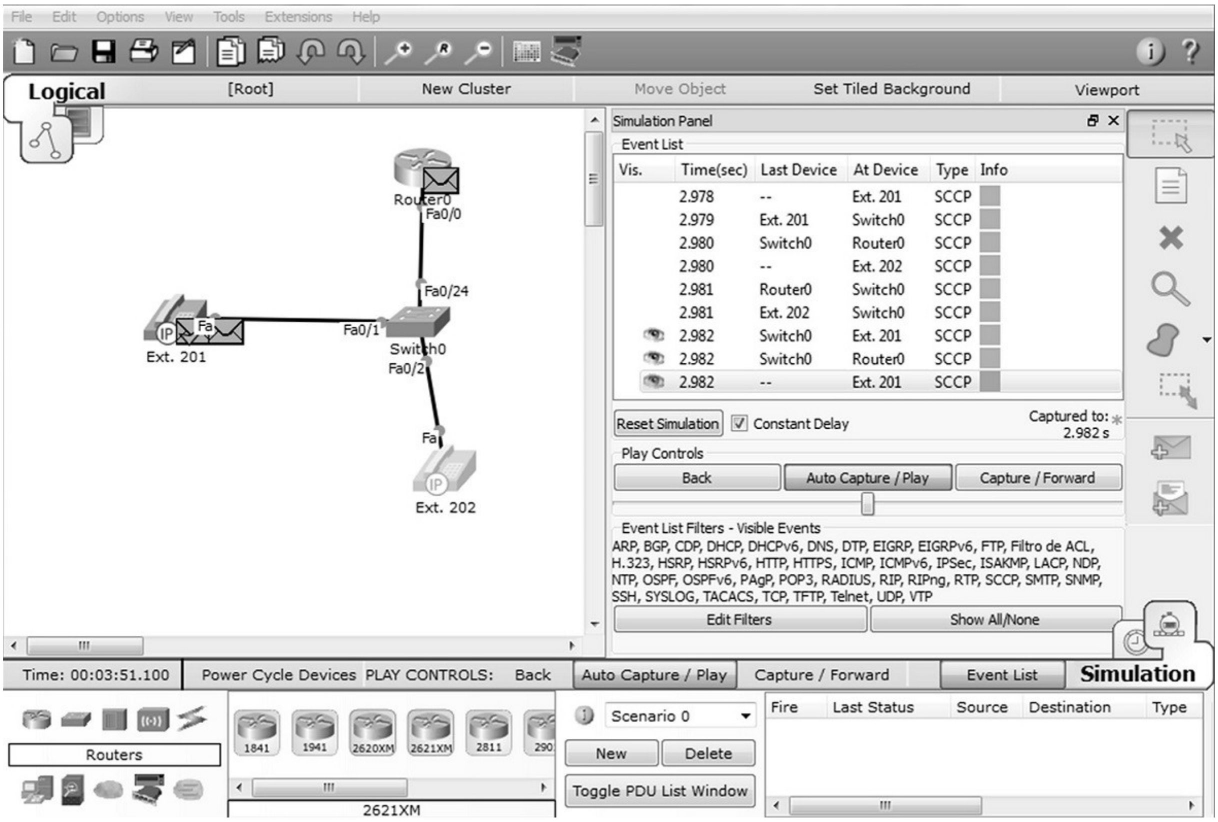

Fig. 4. Analysing VoIP traffic.

Table 4. Configuring VoIP service

Configuring the router	Configuring the switch
enable	enable
conf t	conf t
telephony-service	interface range fa0/1-5
max-dn 10	switchport mode access
max-ephones 10	switchport voice vlan 1
ip source-address 192.168.1.1. port 2000	exit
exit	exit
ephone-dn 1	wr
number 201	
exit	
ephone-dn 2	
number 202	
exit	
wr	

static routing, dynamic routing, essential services like DNS and DHCP, IPv6, VLANs and VoIP. All of these involve network simulation tools in which the internetworking equipment is configured from the command line. On that basis, we would expect that the student can successfully configure a complex network that integrates voice and data technologies in real equipment as proposed.

Bearing in mind the equipment's features, and in an effort to exploit our resources to the maximum, we designed experiments of increasing difficulty, the more complex of which we describe below. The main target was to simulate the network of three companies, which would be able to communicate between their own offices but never with others. In addition to a conventional data network, one of these companies would have a VoIP telephony service.

Fig. 7 shows a schematic layout identifying the companies as A, B and C. A_Router, B_Router1 and C_Router1 represent the virtual routers of each company to be physically implemented using VRF (Virtual Routing and Forwarding) technology for the Cisco 1861. B_Router2 and C_Router2 represent access (which may be remote) to another company's offices, implemented here using the Cisco 861.

To separate data traffic in the companies' offices,

Fig. 5. Connecting the IP phone.

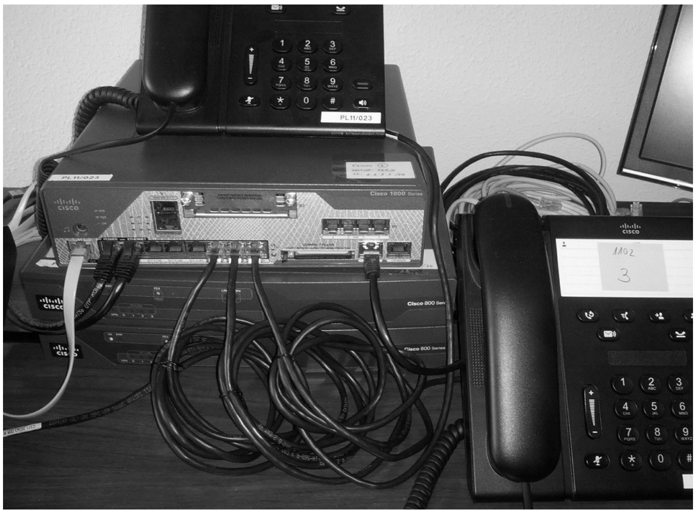

Fig. 6. Networking laboratory prototype.

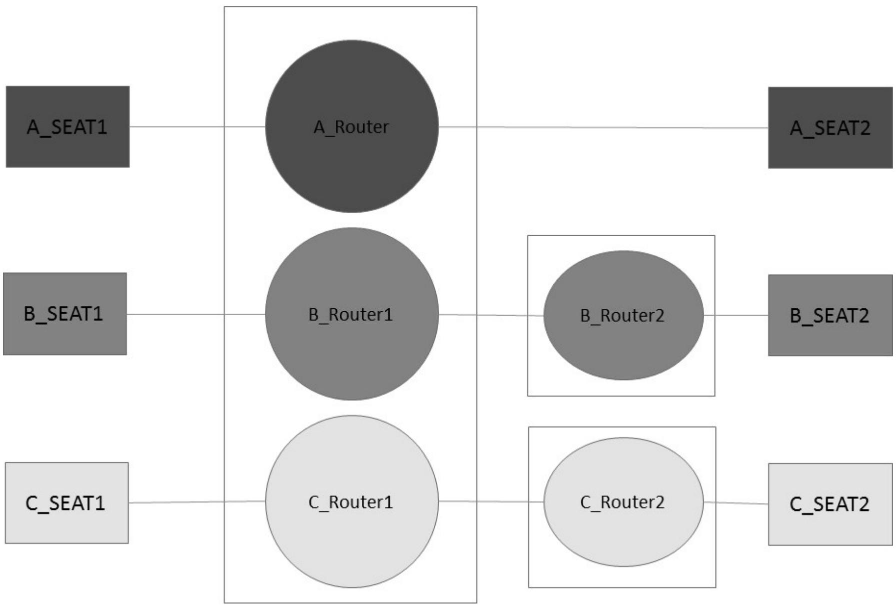

Fig. 7. Schematic layout of voice and data integration lab experiment.

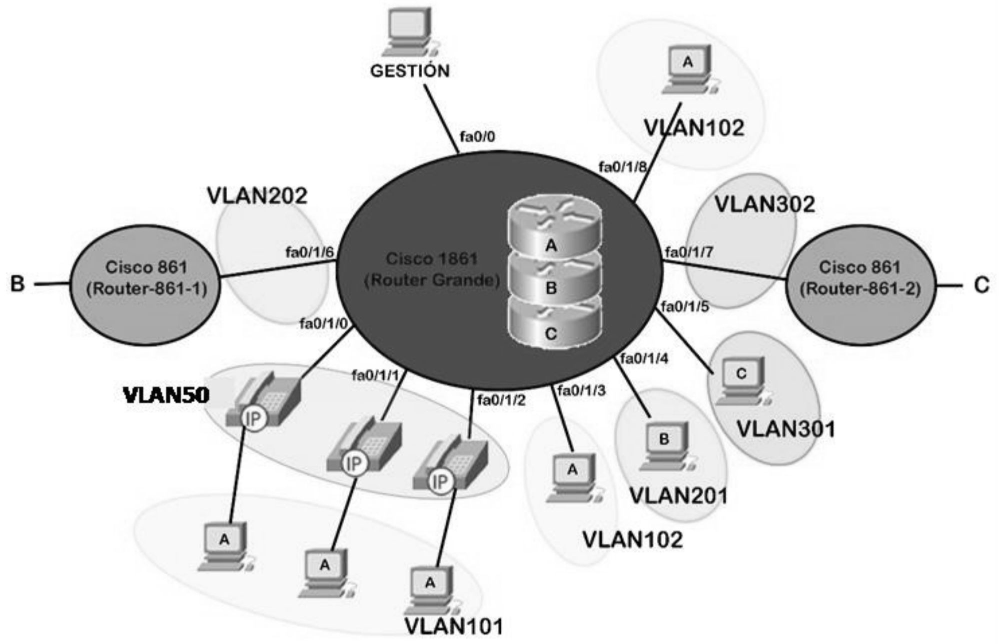

Fig. 8. Logical view of voice and data integration lab.

Table 5. Address space of voice and data integration lab

Interface	Company – Seat	VLAN	Network
Fa0/0	Management	Native	1.1.1.0/24
Fa0/1/0-2	A_SEAT1	VLAN101	172.10.0.0/16
	A_SEAT1_VOICE	VLAN50	172.50.0.0/16
Fa0/1/3	A_SEAT2	VLAN102	172.11.0.0/16
Fa0/1/4	B_SEAT1	VLAN201	172.20.0.0/16
Fa0/1/5	C_SEAT1	VLAN301	172.30.0.0/16
Fa0/1/6	B_SEAT2	VLAN202	172.21.0.0/16
Fa0/1/7	C_SEAT2	VLAN302	172.31.0.0/16
Fa0/1/8	A_SEAT2	VLAN102	172.11.0.0/16

we defined different VLANs and a virtual router. In this way, each company will have independent routing tables. Voice traffic will have its own VLAN (VLAN50). A logical view of the experiment ready for implementation is shown in Fig. 8. The address space to be assigned by DHCP to each VLAN is shown in Table 5.

Fig. 9 shows a physical view of the wiring diagram as well as the assigned VLANs and address space.

The student is guided step by step in configuring

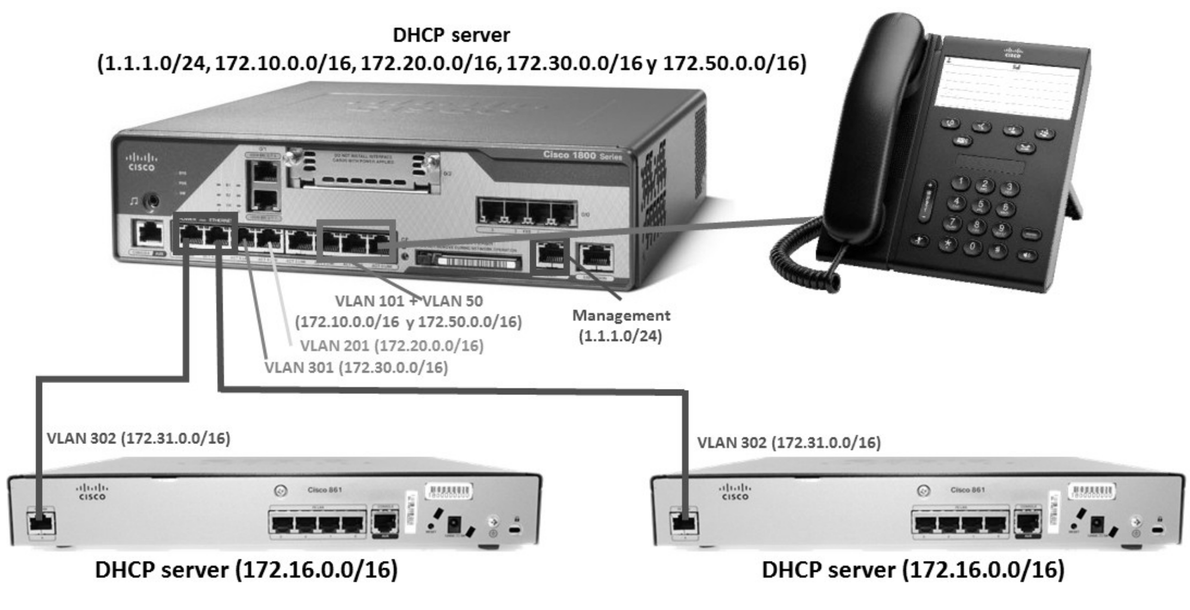

Fig. 9. Physical view of integration voice and data lab.

the three routers, in which the following are the principal steps:

- Set router clock.
- Configure console and VTY (Virtual Teletype) ports.
- Configure one ethernet interface for management tasks (usually FastEthernet0/0).
- Configure DHCP service.
- Define VRF routers (only for 1861 router).
- Define VLANs and associated virtual interfaces.
- Configure physical interfaces.
- Configure a dynamic routing protocol (RIPv2).
- Configure router 1861 as a TFTP server; IP telephones load their firmware from this server.
- Configure Cisco Unified Communications Manager Express (CME) for our telephony network (1861 router only).
- Configure IP phones (1861 router only).

When working with real hardware, differences from the network emulators are quickly observed, including setting of the internal router clock, configuring the console and configuring the virtual interface ports. In addition, router ports clearly differ from switch ports, making it possible to directly assign an IP address to a router port, whereas for switch ports, it is mandatory to associate a virtual interface to a VLAN.

Other details of relevance include the following:

- If you connect two routers, you must define subinterfaces.
- The interfaces allowing traffic from multiple VLANs need to be configured in trunk mode.
- To restrict traffic between VLANs, access control lists (ACL) must be defined.
- Our license limits the number of phones to eight.

Implementation of these experiments involves several two-hour sessions, and scripts are designed to make this possible. By restoring equipment to factory defaults, as well as intermediate situations, students can save their work and resume at the point where they left it in the previous session.

6. Conclusions

This paper details our efforts to design and build a networking laboratory, using both physical equipment and network simulators tools to provide experimental lab environments for undergraduate students of Computer Science Engineering at the University of Salamanca. The experiments meet our curriculum goals but also address topics at masters level.

The experiments were designed around the key idea of always choosing the tool best suited to the students' educational level and of introducing skills that facilitate learning of operating system commands for networking equipment—initially through a graphical interface and then using a command line. Acquiring skills with network simulation tools assures students' success in setting up real network equipment.

Our idea is to exploit the advantages of network emulation tools as low cost solutions that complement work with real devices. The work carried out with real equipment in the prototype of voice and data integration enabled us to validate our proposal and to consider buying new hardware in the future. While a large number of students can be a disadvantage, the experiments are geared to masters degrees, where student numbers are smaller.

Our assessment of PT and GNS3 is that they complement each other, and both are essential in the deployment of network laboratories. Neither involves a steep learning curve that would be unfeasible for use in the classroom. PT is suitable for beginning students of networks, as it has a powerful graphical user interface for design and analysis. Its main advantage is that while the equipment is configured, the equivalent IOS commands are also showing, allowing students to learn its syntax. In addition, its simulation mode uses animation to provide a view of generated traffic. Its main limitation is the traffic analyser and supported protocols; traffic is shown all together, which complicates study of traffic at a particular link in more complex network designs.

As a graphical network simulation tool that allows one to work with a greater number of protocols and traffic analysers such as Wireshark, we would choose GNS3. It runs images directly of the operating systems of real network equipment, narrowing the distance between working with simulated and real equipment. GNS3 generates configuration files that can be loaded to real hardware, allowing integration of real and simulated labs and offering almost limitless possibilities for the design of complex networks. The only limiting factor is the machine running the virtualized network elements, but this could also be scaled.

The designed experiments include concepts ranging from IP addressing (basic level) to the integration of voice and data traffic (third level). Following on from essential services such as DNS and DHCP, dynamic and multicast routing, the new version of IP (IPv6), VLANs and VoIP, we are already working on the design of laboratory exercises for subjects such as wireless networks and network security.

The repository with all the elaborated material can be found at http://diaweb.usal.es. At present, it is available only in Spanish.

References

1. Association for Computing Machinery (ACM). Joint Task Force on Computing Curricula and IEEE Computer Society, Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science, ACM, New York, NY, 2013.
2. http://www.cisco.com/cpress/cc/td/cpress/fund/ith/ith01gb.htm Cisco Press Internetworking basics.
3. Ka Ching Chan, Integration of physical equipment and simulators for on-campus and online delivery of practical networking labs, Technical Report CSIT 20151002, La Trobe University, Bendigo, Australia, October 2015.
4. M. Gregg, Build Your Own Security Lab: A Field Guide For Network Testing, Indianapolis, IN: Wiley Publishing, Inc. April 2008.
5. Virtual InterNetwork Testbed (NS and NAM) http://www.isi.edu/nsnam/vint/. Accessed 26 May 2016
6. SSFNet, http://www.ssfnet.org/. Accessed 26 May 2016.
7. M. Wannous and H. Nakano, NVLab, a networking virtual web-based laboratory that implements virtualization and virtual network computing technologies, IEEE Trans Learn Tech **3** (2010), pp. 129–138.
8. A. Castiglione, G. Cattaneo, L. Catuogno, E. Cerrelli, C. D. Giampaolo, F. Marino and R. Rotondo, Virtual lab: A concrete experience in building multi-purpose virtualized labs for computer science education, *Proceedings of the 20th International conference on software, telecommunications and computer networks (SoftCOM)*, 2012, pp. 1–5.
9. D. M. Dobrilovic, V. Z. Jevtic and B. Odadzic, Expanding usability of virtual network laboratory in IT engineering education, *Int. J. Online Engg.*, **9**, 2013, pp. 26–31.
10. S. Carot Nemesio, P. de las Heras Quirós, E. M. Castro Barbero, and J. Centeno González, Early Experiences with NetGUI Laboratories, SIIE'06: Simposio Internacional de Informática Educativa, Leon, Spain, October 2006.
11. NetGUI, a graphical interface for the Netkit system, Grupo de Sistemas y Comunicaciones, Departamento de Ingeniería Telemática y Tecnología Electrónica, Universidad Rey Juan Carlos, Madrid, Spain, http://mobiquo.gsyc.es/netgui/, Accessed 21 January 2016.
12. A. M. Moreno Montero and A. Paredero de Dios, LabRedes. Net—Laboratorio de prácticas para la materia de Redes de Computadores. Technical Report, October 2012.
13. O. Bonaventure, Software Tools for Networking, IEEE Network, Nov/Dec 2004.
14. Cisco Career Certification, http://www.cisco.com/web/learning/certifications/. Accessed 29 January 2016.
15. GNS3, virtual network emulator, https://www.gns3.net, Accessed 28 January 2016.
16. Wireshark, network protocol analyzer, https://www.wireshark.org, Accessed 28 January 2016.

Ángeles Mª Moreno Montero received her PhD degree in computer science from the University of Salamanca (2013). She is currently an Assistant Professor in the area of Artificial Intelligence and Computer Science at the University of Salamanca, and her current teaching focuses on computer networks. She is a member of the Robotics & Society Research Group. Her research interests include computer networks, robotics and intelligent systems. Additionally, she has co-authored several published papers in recognized journals, workshops and symposia, mostly related to computer networks, distributed systems, robotic, intelligent systems and technology-enhanced learning.

David Retortillo Manzano received his B.S. degree in computer science engineering from the University of Salamanca (2015). He is currently working at GMV (http://www.gmv.com) in the area of Computer Security.