

Early Experiences with NetGUI Laboratories^{*}

Santiago Carot Nemesio, Pedro de las Heras Quirós, Eva M. Castro Barbero,
and José A. Centeno González

Grupo de Sistemas y Comunicaciones (GSyC)
Departamento de Ingeniería Telemática y Tecnología Electrónica (DITTE)
Universidad Rey Juan Carlos (URJC)
Móstoles, Spain (ES)

Abstract. Teaching computer networks to freshman on its first quarter of telecommunications engineering is a challenge due to the few, if any, previous knowledge that must be assumed in the areas of computing and computer networks. After evaluating existing alternatives that could be used by the students in the practicums, we decided we had to provide them with better tools. We have implemented a GUI named NetGUI that allows students to design and operate computer networks emulated by the Netkit software environment. NetGUI has been tested with 120 freshman students and with a smaller group of 3rd year students with prior knowledge in computer networking at two public universities in Spain. This paper describes the tool and the experience gained in the test that we made with the students.

1 Introduction

Nowadays, computer networks is a very familiar field, which every person is using each day through Internet services such as WWW or e-mail. However, many times teaching how computer networks work to computer science or telecommunications engineering students is done in a too theoretical way. The computer network concepts are explained using the typical abstractions such as layers, interfaces, communications primitives and channels, etc. It seems to us a contradiction that something students are using everyday is taught in these abstract terms.

There is a need of teaching computer networks in a practical way, so students can learn how communications between computers are carried out through practical exercises executed on their computers. A practical problem with the computer networks topic is that this discipline requires some connected computers where to observe how computers exchange information among them. In order to do so every student would need a set of communications hardware (routers, hubs, switches, PCs, etc.) to configure different network scenarios. This situation does not scale to hundreds of university students.

^{*} This work has been partially financed by a grant for teaching innovation from Universidad Rey Juan Carlos

There are networking software tools that are used to analyze, emulate and simulate computer networks. These tools can also be used for teaching computer networks. However, there is a lot of work to be done in order for these tools to be used with freshman students with low, if any, computer and networking skills. It is required that these tools are extremely easy to use.

In this paper we present NetGUI, which is a simple GUI designed for freshman students with the purpose of teaching practical skills that help them to understand the computer network abstractions explained in the lectures.

The organization of this paper is as follows. Section 2 provides a review of some software tools that exist for networking related to NetGUI. Section 3 introduces the NetGUI software. Section 4 shows some experiences carried out with NetGUI. Finally, section 5 details conclusions and shows work in progress.

2 Software tools for networking

There are many software tools for networking [1] which can be grouped in two categories: emulation environments and simulators.

Host emulation environments try to model a specific hardware architecture and are used to test functionality of kernels and its applications. Network emulation environments usually manage some instances of host emulation to configure a virtual network in order to test functionality of network protocols and applications. Another approach to network emulation is based in virtualization of the network stack instead of the complete kernel.

On the other hand, simulators are normally used to evaluate performance of protocols and algorithms, instruction sequencing problems and speed tests.

We are interested in network emulation environments based on host emulation which provide students the illusion of managing "connected real hosts" as a "real computer network". These tools help students to learn and visualize the behaviour of network protocols. Next we will describe some examples of this kind of network emulation environments that are relevant to this paper.

UML (User Mode Linux) [3] provides a Linux virtual machine running as a Linux user space process. UML helps to run experimental Linux versions, kernel configurations or trial applications in a secure way, without risking the host Linux kernel. Using UML we can run the same daemons and applications we run in a real host. Therefore, we can configure some UML virtual machines to work as a computer network. However, many advanced configuration commands are needed in order for a student to run even the simplest network scenario.

VNUML (Virtual Network User Mode Linux) [4] is an environment which gathers a set of scripts used for describing complex testbeds in XML for testing network applications and services. From the XML description of the network scenario, VNUML easily starts the UML virtual machines and configures them. This is a very useful environment for testing complex network scenarios since it hides the advanced details of booting and configuring virtual machines. However, students must describe the complete network scenario using XML language and

sometimes it requires very specific network details. VNUMLGUI [5] is a GUI for VNUML.

Netkit [6] is similar to VNUML. It is an environment for setting up and performing experiments at low cost, designed with the purpose of teaching. It is also based on UML. Netkit uses easy shell scripts to start virtual machines and to emulate network devices and their interconnections. There exists a GUI for Netkit, although as far as we know it is not currently maintained.

3 NetGUI

The NetGUI tool was specifically designed to be used by freshman students of telecommunications engineering. It was assumed that they would have little or none experience using command line interfaces such as the Unix shell.

Although Netkit and VNUML tools are used in several universities for teaching purposes, we did not consider these tools appropriate for our freshman students. Students must write shell commands when they use Netkit, or XML files when they use VNUML, to start, stop and configure virtual machines and networks. We considered our freshman students could confuse these commands with the ones they must use inside each virtual machine. This was the main reason that led us to design and implement NetGUI.

In the lectures teachers describe networks with diagrams which show hosts connected with routers and switches by means of network links. Our main objective was that freshman students could use a tool where they could draw exactly the same network diagrams that the teachers draw on the blackboard. NetGUI allows them to do so, and then, to start emulations of hosts and routers in order to experiment with TCP/IP virtual networks. The emulation is provided by means of the Netkit software environment.

3.1 Functionality

NetGUI provides a graphical interface for the Netkit system. NetGUI offers a panel with a menu bar and a collection of buttons (see figure 2).

The user can draw network configurations on the panel by choosing elements from the button bar. Network configurations can be saved and loaded using menu items. The network configuration that can be loaded/saved includes the changes made to the file system of started virtual machines.

After launching NetGUI, the user can edit network diagrams, start and stop virtual machines (both computers and routers), and interact with the started virtual machine's consoles.

Network diagrams are created by inserting in the panel the following elements chosen from the button bar: computers, routers, ethernet hubs and links. Network links can be drawn between two elements in order to join computers and routers to hubs. Routers can also be directly linked among them.

When a node (router or host) is started it must be named through a popup window. Then, a new window appears with a console for each node, and its icon

changes so that the student knows the node is started, what informs him there must be a console window somewhere on the desktop. The title of each console window is the name given to the node when it was created.

Interaction with a network diagram. The elements that have been drawn on the panel can be deleted by choosing the scissors button. All the virtual machines of a network can be started or stopped with the respective buttons.

A node can be individually started and shut down by using a toggle accessible by clicking with the right button of the mouse on the node.

The last and perhaps most useful button is the “Selection tool”. It allows to enter the Selection Mode, that provides the following functionality:

Moving a node by left clicking and dragging an element on the panel. In case the node is linked, the links are redrawn appropriately.

Moving to foreground the console of a launched node by double clicking on the icon of a launched node (router or host). This functionality is very useful even in simple diagrams, because there are multiple windows around, one for each virtual machine.

By right clicking on the background of the panel and dragging the mouse the user can zoom-in and zoom-out the whole diagram.

By left clicking on the background of the panel and dragging the mouse the user can shift the whole diagram.

Automatic visualization of information. The names of network interfaces appear automatically near the line representing a link.

When the user changes the IP address of a network interface through the command line of a console, the new IP address is automatically drawn near the icon of a node.

3.2 NetGUI implementation details

The current implementation of NetGUI described on this paper (v0.4.1) is programmed in Java and runs on GNU/Linux. NetGUI is software libre, distributed under the GPL license.

The graphical aspects of NetGUI are programmed using the Piccolo[2] graphics Framework.

When a virtual machine is launched from NetGUI, the right Netkit commands are automatically generated and invoked. This is in fact what implements the main advantage of NetGUI over the usage of Netkit commands. NetGUI stores the topology of the network being designed, what allows it to generate the Netkit commands, making this process transparent to freshman students who could confuse these commands with the commands they must issue on the virtual nodes.

The NetGUI application maintains a TCP connection with each virtual node, thus providing access to a second console, additional to the one used by the student. Currently this mechanism is used for checking the state of the virtual

machine and obtaining information to be drawn such as the IP addresses of network interfaces. The information is obtained by issuing commands to the second virtual console through the TCP connection. For example, the IP address is obtained by sending the *ifconfig* command. A series of threads guarantee that this kind of interaction is performed periodically in order to keep updated the information about each virtual node drawn by NetGUI.

4 NetGUI use in the classroom

NetGUI has been used simultaneously in a pilot experience at two public universities of Madrid with freshman students of telecommunications engineering and with 3rd year students in a Bachelor of Telematics Engineering. The software was installed in the GNU/Linux laboratories. A CD including NetGUI, Netkit and UML files was prepared for the students.

4.1 Freshman students

Three groups of freshman students of around 40 students each, used NetGUI during 10 weeks from november 2005 to january 2006 at Universidad Rey Juan Carlos.

The course is named Arquitectura de Redes de Ordenadores (Computer Network Architecture). The course is designed to provide students an introduction to computer network architecture. The syllabus includes abstractions such as protocols, protocol stacks, layers and encapsulation of data packets. Students learn for example what is the purpose of adding headers to data packets and how data packets are routed through the Internet using the TCP/IP protocol stack.

It is a challenge to teach this course because students have little knowledge of topics that are usually assumed in normal introductory courses and textbooks of computer networks, such as computer architecture, programming languages or operating systems. But it was the belief of us, the teachers, that students could learn the abstractions used in computer networks without basing them in prior knowledge, specially if the explanations were supported by a rich practical experience where students could experiment with real, or almost real networking environments from day one.

Prior to the labs taught with NetGUI, students are taught basic concepts of the Un*x shell, so that they can operate the consoles of each of the nodes of a network configuration designed with NetGUI. The Un*x shell labs were taught during 6 weeks. 8 weeks were dedicated to NetGUI labs.

Below we describe the very first compulsory assignment that students had to do individually with NetGUI. The rest of assignments can be consulted on the web page of the course [10]. All the assignments were made by a total of 120 freshman students, during sessions of two hours, monitorized by teachers. No student experienced major troubles using NetGUI or completing the assignments, and many of the students explicitly manifested the teachers their satisfaction

because they had the sensation of having configured and operated a real network. A very high percentage of the students that attended the practical classes passed with high scores. Details will be provided on the conference, once the september exams have been done.

Assignment 1: basic configuration of a network. Students must create a network diagram with NetGUI composed by three hosts and two subnetworks, interconnected through a router. Figure 1 shows the diagram created for this assignment.

A range of IP addresses is then given to students. They must assign correct IP addresses to network interfaces and then configure the routing tables of the hosts and the router. They do it by using the *ifconfig* and *route* commands.

In order to check the correct configuration of network elements, students then must use the *ping* command among the hosts and router.

Once the network is correctly configured and the connectivity among its nodes has been tested, students must explore the contents of data packets exchanged due to the *ping* commands. They do so by configuring network probes on several network interfaces with the *tcpdump* command.

Captured packets are then observed by the students using the *ethereal* application. The latest versions of several textbooks on computer networks incorporate exercises where the student is given a capture file so that he must inspect and study its contents. In our approach, it is the student who captures its own generated traffic, and then studies it on *ethereal*. We consider this approach much more useful for the student.

Once the network configuration has been created, the student would bootstrap the router and the 3 hosts, what causes that 4 windows appear on the screen, each one with a console for the router and the three hosts (see figure 1).

4.2 3rd year students

Eight groups of 3rd year students in the Bachelor of Telecommunications Engineering (Telematics) used NetGUI at Universidad de Alcalá for doing an optional practical assignment. The groups were formed by two students of the Comunicación de Datos (Data Communications) Course.

The aim of these practical exercises was to configure interior routing protocols (RIP and OSPF) and to visualize routing information exchanged between routers. RIP and OSPF lectures were taught days before in the classroom. The students performed the following tasks: Network address planification in a specific scenario (see figure 2), Address configuration in every network interface, Configuration of RIP and OSPF protocols with *zebra* and Monitorization of exchanged routing messages and routing table updates through *tcpdump* and *ethereal*.

In this case, the NetGUI tool was very useful because interior routing protocols require specific hardware (some routers, hubs and hosts) for each group

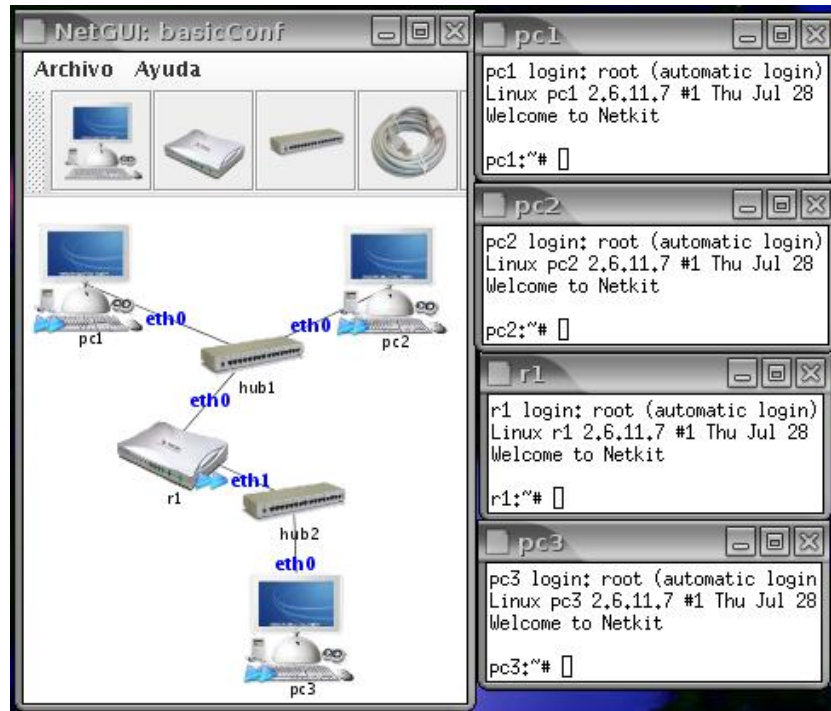


Fig. 1. Basic configuration.

of students in order to deploy different network scenarios. By using NetGUI students could work at home, using it on their laptops.

Since practical exercises were planned as students homework, without continuous teacher monitoring, we could test how simple and intuitive is the NetGUI interface. It let students quickly familiarize with the environment and carry out the aim of the exercises.

4.3 Similar experiences

The latest editions of widely used textbooks [8] on computer networks suggest laboratories where captured files are provided to students for them to analyze. We consider that laboratories created with NetGUI cover these same learning objectives and much more, thanks to the possibility that students design and operate their own network configurations, and then they capture their own generated network packets.

Others have used similar approaches for teaching network laboratories using emulated networks: Netkit [6] was in fact developed for this purpose. Netkit has usually been used without a GUI, at least judging by the way learning objects are distributed alongside the Netkit distribution. VNUML has also been used

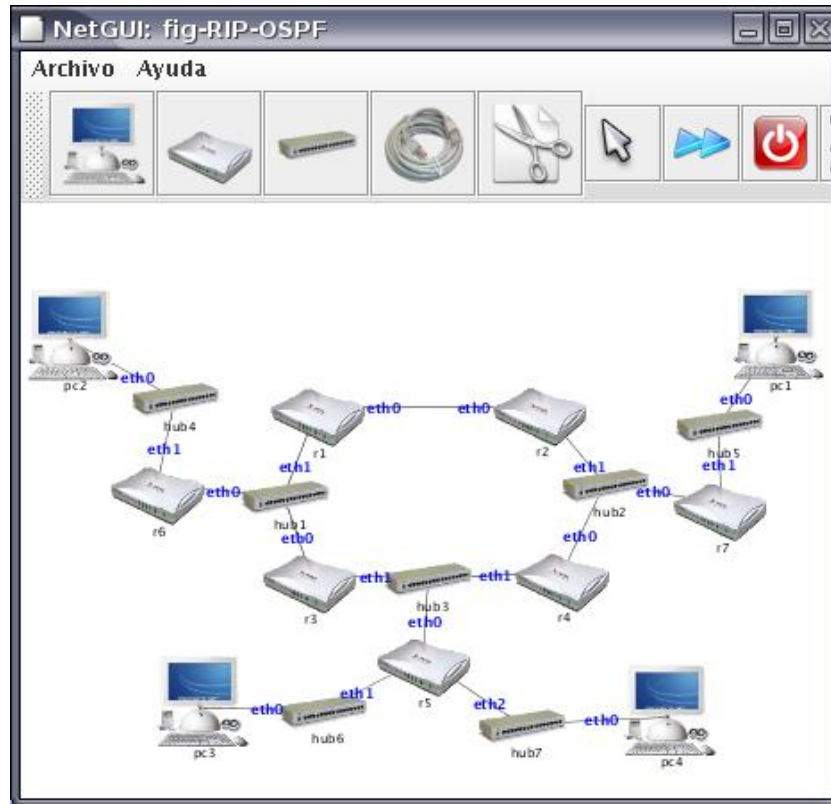


Fig. 2. Network scenario for interior routing protocols.

for teaching[9]. None of these experiences had as a target freshman students, as is the case in the NetGUI experience this paper reports.

5 Conclusions and Current Work

Using NetGUI in the classroom was satisfactory both for the students and teachers. We plan to repeat the experience next year, extending it to new groups and new courses. Although we prepared a NetGUI CD for students, it required they had installed GNU/Linux. Being freshman, many students experienced trouble with the installation process. It is so why we are designing a NetGUI Live CD that requires no installation at all.

We consider that tools such as NetGUI which allow students to perform homework assignments without requiring expensive networking hardware will play an important role in the upcoming European Space for Higher Education.

We are designing what we call a network oracle for NetGUI. It consists of automatic probes made with *tcpdump* that capture packets on all the links.

We expect to visualize graphical representations of this information on NetGUI when users request explanations to monitoring and diagnosis queries about the network behaviour. For example students will be able to see a graphic depicting the path traversed by a given packet, or will see diagrams of protocol interactions for a given packet exchange.

NetGUI allows the students to make individual assignments. We would like to design assignments that must be made collaboratively by several students. It is so why we are designing a web based NetGUI. The virtual machines will run on a server at our university, and students will access them through the web. This way several students will be able to operate a single network scenario.

References

1. Software Tools for Networking, O. Bonaventure, IEEE Network, Nov/Dec 2004.
2. Piccolo Web Page <http://www.cs.umd.edu/hcil/jazz/>
3. User Mode Linux (UML) <http://user-mode-linux.sourceforge.net>
4. Virtual Network User Mode Linux (VNUML) <http://jungla.dit.upm.es/vnuml>
5. VNUMLGUI web page <http://pagesperso.erasme.org/michel/vnumlgui/>
6. Netkit <http://www.netkit.org>
7. <http://netlab.sourceforge.net>
8. J. Kurose and K. Ross, *Computer Networking: A Top-Down Approach Featuring the Internet*, Addison-Wesley (3rd edition–May 2004).
9. Fermín Galán, David Fernández, Javier Ruiz, Omar Walid, Tomás de Miguel. *A Virtualization Tool in Computer Network Laboratories*, 5th International Conference on Information Technology Based Higher Education and Training (ITHEIT04), Istanbul, Turkey, May 2004.
10. Web page of the ARO course: <http://edukalibre.org/moodle/course/view.php?id=7>