2006-1824: REMOTE INTERNETWORKING LABORATORY

Imad Jabbour, Massachusetts Institute of Technology

Imad W. Jabbour received his B.E. in Computer and Communications Engineering with distinction from the American University of Beirut in 2005. He is currently an M.S. candidate in the Information Technology program at MIT, and is working as a graduate Research Assistant at MIT's Center for Educational Computing Initiatives. His current research includes the implementation of software tools for online laboratories, as part of Microsoft-MIT's iLabs project. He holds a Microsoft Certified Systems Administrator certification since 2003, and is a Student Member of the IEEE since 2002.

Linda Haydamous, New Jersey Institute of Technology

Linda A. Haydamous received a B.E. in Electrical Engineering with a minor in Information Technology from the American University of Beirut in 2005. She is currently an M.S. candidate in the Engineering Management program at New Jersey Institute of Technology, and is working as a graduate Research Assistant in Operations Research. She is a Student Member of the IEEE since 2003.

Wissam Kazan, Stanford University

Wissam S. Kazan received his B.E. in Computer and Communications Engineering with distinction from the American University of Beirut in 2005. He is currently an M.S. candidate in the Computer Science program at Stanford University, and is specializing in Artificial Intelligence. He holds a Microsoft Certified Systems Administrator and a Cisco Certified Network Associate certifications since 2003.

Amine Hayek, Stanford University

Amine Hayek received his B.E. in Computer and Communications Engineering with distinction from the American University of Beirut in 2005. He is currently a candidate for an M.S. in Engineering at Stanford University.

Remote Internetworking Laboratory

Abstract

Remote experimentation is the process of conducting real lab experiments without being physically in contact with lab devices. Recently, the development of Internet technologies fostered the spread of online laboratories, and significant benefits helped establish remote experimentation as a potential substitute to real experimentation. Internetworking lab experiments usually involve very expensive hardware; and providing remote access to this type of laboratories is highly desired, especially when a large number of students must share a limited set of devices. Very few remote internetworking laboratories have previously been implemented, and they had several drawbacks. In this paper, we discuss our design of a Remote Internetworking Laboratory which provided access to the internetworking lab at the American University of Beirut (AUB). The uniqueness of our approach is that it uses VLANs (Virtual LANs) to build network topologies, a technique that separates network subnets into different broadcast domains before allowing users to configure each lab device. The proposed scheme overcomes the limitations of other designs, and allows users to build any desired network topology and configure the lab devices in real-time. Many technical and educational features were integrated into the system, and possible enhancements were suggested based on several evaluation criteria.

I. Introduction

Recently, remote experimentation was introduced as a potential way of overcoming the limitations of traditional laboratory experience. It can be defined as the process of conducting real lab experiments while being physically located away from laboratory devices. The general scheme for remote experiments consists of a user that connects to a remote server through a graphical interface, via the Intranet or Internet; the server directly interfaces the equipment and is responsible for forwarding the user's requests to the intended lab device.

The internetworking lab that we are considering was offered as a course to undergraduate engineering students at the Electrical and Computer Engineering (ECE) department at AUB, starting 2003, and an average lab session, held weekly, lasted for more than two hours. In this lab, students do not work on fixed topologies; instead, they work on a fixed number of network components which they interconnect and assign different roles to, depending on the experiment. Many students requested to be offered supplementary lab sessions so that they can familiarize themselves with previous experiments or acquaint themselves with prospective ones. However, it was very difficult to match these requests in view of the inadequate amount of space and time, and the limited availability of teaching assistants.

In this paper, we present the design of a Remote Internetworking Laboratory, which we fully implemented, tested and allowed for use by a sample group of students. Our design allows the lab to be accessed remotely, anywhere and at anytime. Furthermore, it overcomes the limitation on the number of devices, by making them accessible 24 hours/7 days a week. Students can use the remote lab by reserving a lab session and then executing the experiment interactively, by connecting any network topology remotely and configuring the network devices in real-time.

Our design is based on the Virtual Local Area Network (VLAN) technique which isolates the network subnets from each other, as if they were connected to different hubs. Moreover, our system embeds several educational features that make it easier for both students and graduate assistants to use the online laboratory.

The paper proceeds in Section II with a description of related work. Section III provides an overview of the internetworking lab at AUB. Section IV gives a thorough description of the system design. Section V shows how the different design elements are used by the system to enable the execution of a remote experiment. Section VI draws an assessment of the online lab based on ABET criteria and on students' reactions. Section VII concludes this paper.

II. Related Work

In this section, we start by giving an overview of remote experimentation, and then we discuss and evaluate two designs of remote internetworking laboratories.

A. Remote Experimentation

Laboratory experimentation is a hallmark activity in any engineering discipline. Throughout their curricular education, prospective engineers learn how to apply their theoretical background into practical situations in the lab, and investigate their understanding of the subject matter by placing their knowledge in a useful context.¹ However, traditional laboratory experimentation suffers from several disadvantages that can be overcome by remote experimentation.

Remote experimentation is the process through which real lab experiments can be conducted by a user without her having to be physically interacting with lab devices. In other words, remote experimentation consists of allowing users to connect to actual hardware devices and gain accessibility to laboratories via the Internet technology. Recently, the development of the Internet as a communication medium fostered the spread of remote laboratories.²

Remote experimentation offers many benefits over traditional experimentation, and its increasing utilization through out the globe justifies its advantages. First, remote experimentation provides access to remote laboratories anywhere and at anytime; lab experiments are not limited by time or space constraints³ as in traditional laboratory experiments. Second, distant learners can be given the opportunity to benefit from laboratory courses, when such courses have until recently been regarded as impractical for remote students.² Moreover, access to expensive state-of-the-art material becomes simpler and the cost of buying numerous devices is mitigated^{3,4} if we allow multiple users to remotely share a single set of hardware devices. Finally, unsafe or hazardous use of expensive hardware equipment² is diminished, especially when safeguards are built into the software at the lab server level. Mougharbel et al.⁵ present an extensive overview and analysis of remote experimentation models.

B. Remote Internetworking Lab at the Middle Tennessee State University (MTSU)

This remote access internetworking lab⁶ has a Management Station (MS), several hosts and several routers, all of which are connected to an Ethernet switch. The MS has control over the whole network, and is the only station with an interface that connects to the Internet. Furthermore, it has another interface that connects to the Management LAN (MLAN): the MLAN is the subnet to which all machines are connected (through one of their interfaces) via the switch. In order to establish a given network configuration, the remote user must first login to the MS, via an assigned username and password, and then access each network component

separately via the MLAN. The interfaces of the devices which are interconnected in the network topology must be assigned IP addresses that belong to the same subnet.

This design presents several drawbacks. First, the user does not “build” a network topology in the real sense; indeed, the topology is configured only after accessing the desired hosts and routers and configuring some of their interfaces to be part of a given subnet (by explicitly specifying an IP address and a subnet mask). In addition, if in a given network two LANs have the same IP range but are located many hops apart, they will still be able to ping each other, because they are physically connected to the same switch. Moreover, the internal routers are accessed through their standard Ethernet interfaces, and not through their console lines, which might prevent users to configure them remotely in case their routing configuration is erroneous.⁷ Furthermore, since all nodes have an IP address on the MLAN, users cannot use the MLAN IP addresses when they are configuring a network, nor configure default gateways that point to the MLAN IP addresses range. Finally, given that all nodes are connected to the Ethernet switch, any node can be reached in a single hop, which led the designers of the system to suppress ICMP redirect messages at system setting.

C. Collaborative Remote Computer Networks Lab⁷

This is another interactive remote computer networks laboratory, where many users can collaborate on a single experiment. In this design, real-time collaboration between several students is enabled by using Microsoft ConferenceXP Client. Unlike the system designed at MTSU, lab routers are connected via their console lines to a terminal server (see subsection IV-B), and thus can be accessed even if their configuration is incorrect. However, the main drawback in this design is that students can only work on predefined topologies and cannot build a network topology remotely.

III. Internetworking Lab Overview

The internetworking laboratory at AUB aims at showing how computer networking concepts, which students mainly study theoretically, are applied in a real network. The lab is equipped with six sets of devices, which allows six groups of students to work on the same assignment during the same lab session. Each set of devices consists of four PCs (rack servers), four hubs and four routers. All PCs have Linux Red Hat 9 operating system installed, and each PC has two 10/100 Mbps Ethernet interface cards labeled eth0 and eth1. The four routers are Cisco 2600-series routers; each router has two Ethernet interfaces operating at 100 Mbps, and a console port. Finally, each hub consists of eight ports and operates at a dual speed of 10/100 Mbps.

This internetworking laboratory is mainly concerned with implementing an application layer protocol and observing the packets transmitted. In this lab, students also learn how to configure Linux PCs and make them act as specific servers such as DHCP or NAT, as well as how to configure Cisco routers. Students monitor the network by using special applications such as “tcpdump” or “Ethereal”, and might need to check the routing tables of some devices or perform Telnet sessions. In other cases, students trace the path of network packets using the *traceroute* command, ping other network devices and perform various other functions that enhance their understanding of computer networks.

IV. System Design

The goal of our design is to allow a user to execute an experiment remotely, in such a way that the resulting interaction with network devices is as high as possible. In addition to the available laboratory equipment described in the previous section, we added three major components to the overall design: the Lab Server (LS), the *iSwitch* and the *iRouter*.

Our proposed scheme (Figure 1) allows a remote user to connect to the LS via the Internet (or Intranet) and perform experiments by using a web site hosted by the LS. A Cisco Ethernet switch, the *iSwitch*, plays a key role in building network topologies, and a Cisco router, the *iRouter*, enables the user to configure the rack-mounted routers via their console lines. This section proceeds with a detailed explanation of the major design elements.

A. The *iSwitch*

Remote users are unable to physically interconnect the lab devices when they need to establish a given network topology. This is why we needed an Ethernet switch (that we named *iSwitch*) that internally connects any network configuration requested by the user. The *iSwitch* is a Cisco Catalyst 2950 switch and has 24 ports; it is physically connected to all the Ethernet interfaces of the PCs and routers in the lab. By using this switch, we no longer need the four hubs to establish connections between interfaces that are on the same subnet.

Figure 1: Overall Design of the Remote Internetworking Laboratory

The *iSwitch* connects to all the other devices which are: the *iRouter* (one port), the LS (one port), the PCs (2×4 ports) and the routers (2×4 ports). We installed an additional Ethernet card (which we called *management interface*) on each of the four PCs so that the user can access these machines and configure them remotely. The *iSwitch* ports connected to the *iRouter*, the LS and the four management interfaces are assigned static IP addresses, whereas the remaining ones are connected to the built-in interfaces of PCs and routers, and thus their IP addresses depend on the experiment.

A VLAN allows network devices found on the same LAN to be separated into different broadcast domains. In our case, all the interfaces of all devices are connected to the *iSwitch* and are thus on the same physical broadcast domain. However, by using VLANs, the user can establish any network topology by logically making every subnet in the topology a separate broadcast domain. For instance, if in a given topology, eth0 of PC1 and eth0 of Router1 are connected to the same subnet, we should configure the *iSwitch* in such a way that the ports corresponding to PC1/eth0 and Router1/eth0 belong to the same VLAN.

B. The *iRouter*

In general, the preferred way to configure a router is through its console line, because in most cases we neither know the router's IP address nor its configuration. *Reverse Telnet* is a technique used to connect remotely to the serial console lines of several routers (or switches) using a serial router. The *iRouter* is a Cisco 2522 serial router and is used to route a console line to a remote user: whenever a connection request reaches a specific port on *iRouter*, the latter forwards the request to the corresponding console line.

The *iRouter* has eight serial synchronous/asynchronous ports and one Ethernet port. As stated earlier, the Ethernet interface is connected to the *iSwitch* and is assigned a static IP address. On the other hand, each serial port is a 60-pin (DB60) port that needs to connect to the Ethernet (RJ45) console lines; thus, what we needed was a DB60-to-RJ45 converter which we built using a DB60-to-DB25 cable followed by a DB25-to-RJ45 cable. We connected the *iRouter* to the console line of each of the four lab routers and to that of the *iSwitch*^{*} and we configured it as a terminal server, i.e. a router "with multiple, low speed, asynchronous ports that are connected to other serial devices"⁸ and which allows a user to remotely configure the internal lab routers, via their console lines.

Theoretically, whenever a user wishes to connect to the console line of an internal router, she needs to telnet the IP address of the *iRouter*, as follows: *telnet <iRouter IP address> <2000+x>*, where *x* is the port number (ranging from 1 to 8) of the *iRouter*'s serial line to which a console line is connected. However, the LS's external IP address is the only interface to the user, since IP addresses of internal lab devices cannot be reached remotely. This is why the LS needs to act as a port redirector (see subsection V-B). A complete description of how to configure a Cisco router as a terminal server and how to remotely access console lines through the Reverse Telnet technique was found on Cisco's website.⁸

C. The Lab Server (LS)

The LS is a server that handles the execution of lab experiments as specified by the remote user. The role of the LS is three-fold: it is a web server (running IIS), a database server and a *laboratory server* (a server interfacing the laboratory devices).

Users need to connect to the LS remotely in order to execute their experiments. For this purpose, we have implemented an ASP.NET web application (using C#) that enables users to login to the website, schedule an experiment by reserving an online lab session (see Figure 2), execute their experiments, upload their lab reports, and manage user accounts and experiments (only for faculty members and graduate assistants).

On the other hand, we created a database on the LS, which is used to store users' and experiments' records. The database we used is a standard relational database administered by MySQL Server. This database is used by the ASP.NET code to authenticate users, save

login/logout records, save user and experiment records, schedule experiments, delete lab reservations, etc.

As a web server, the LS has an external IP address that allows remote users to connect to its website and perform their experiments. However, and as we mentioned earlier, the LS has also an internal static IP address since it is connected to the *iSwitch*, and hence can communicate with the *iRouter* (consequently, with the lab routers) and with the internal PCs (via their management interfaces). In this sense, the LS is a *laboratory server* since it interfaces laboratory devices and allows the remote user to gain access to them, as will be discussed further in the next section.

V. Experiment Execution

So far we have presented the various design elements (the *iSwitch*, the *iRouter* and the LS) and how they interact with each other. In this section, we will show how we used these components in order to dynamically interconnect the lab devices and come up with the desired network topology. Furthermore, we will discuss the different techniques used to access the lab PCs and routers remotely and configure them in real-time.

Figure 2: Screenshot of the LS website – scheduling an experiment

A. Connecting Network Topologies

An experiment can be executed by first connecting the desired network topology using the *iSwitch* and the lab server. As indicated in Section IV, the *iSwitch* is responsible for connecting network topologies dynamically, by adding, to the same VLAN, the Ethernet interfaces of several devices. In the *iSwitch*, the way each port is configured is stored in the *running configuration* of the switch, which is a memory that stores the current configuration of the switch and that is erased once the switch is rebooted. Hence, our target is to send the network configuration from the LS to the *iSwitch* that contains the desired VLAN memberships of the lab devices in a given experiment.

In order to configure the *iSwitch* via the ASP.NET web application hosted on the LS, we needed a special C# library, “CSharpTelnet”[†], which allows the LS code to Telnet the *iSwitch*. We use Reverse Telnet to access the *iSwitch* from the LS, by telnetting the IP of the *iRouter* followed by $[2000 + iSwitchSerialPort]$, where *iSwitchSerialPort* is the *iRouter*’s serial port connected to the console line of the *iSwitch*.

Furthermore, we used a TFTP[‡] server and made it run continuously on the LS while users are executing their experiments. The rationale behind using such a server is to allow the LS to send to the *iSwitch* a configuration file that contains the VLAN assignments corresponding to the desired network topology. As a result, the Telnet library uses the Cisco IOS[§] command *copy tftp running-config*, followed by the LS’s internal IP address and the configuration file name (see Figure 3) to transfer the desired file to the *iSwitch*’s memory.

```
string ip = iRouterIP;
int port = 2000 + iSwitchSerialPort;
Terminal tn = new Terminal (ip, port, ...);
tn.Connect( );
...
tn.SendResponse ("enable");
...
tn.SendResponse (iSwitchPassword);
...
tn.SendResponse ("copy tftp running-config");
...
tn.SendResponse (LabServerIP);
...
tn.SendResponse (configFileName);
...
tn.Close( );
```

Figure 3: C# skeleton code for connecting a topology using a configuration file. Note that *Terminal* is a class in the “CSharpTelnet” library

B. Accessing Lab PCs and Routers

After having the desired network topology connected, the user needs to access the network devices available in the laboratory, i.e. routers and PCs. The most suitable way to access a device remotely is to use the Telnet protocol. But since the external IP address of the LS is the only interface available for a remote user, we need to allow the LS to act as a port redirector. In this case, a remote port on a specific lab device is mapped to a physical port on the lab server itself.

Every time a user requests to configure a lab device, the web application hosted by the LS starts a port redirection application^{**}, using appropriate parameters. The physical ports that are opened on the LS are generated randomly and range from 10,000 to 65,000, and each port is mapped into the desired port (*remote port*) on some lab device’s interface (*remote IP address*). In the case of a lab PC, the remote port is port 23 (the default Telnet port) and the remote IP is the IP address of the management interface of that PC. Whereas in the case of a lab router, the remote port is $[2000 + routerSerialPort]$, where *routerSerialPort* is the *iRouter*’s port connected to the internal router’s console port, and the remote IP is the IP address of the *iRouter*.

C. Design Features and Educational Scope

A registered user in the Internetworking Lab course can login to the system by first accessing the “Login” web page on the LS, and then specifying her username and password. In order to perform an experiment, the user needs to acquire full control over the lab equipment for a specific period of time. This is why we created a reservation system on the LS that allows the user to perform any experiment during her scheduled time. Reservations are structured as follows: nine two-hour slots per day (with a thirty-minute interval between two consecutive slots), 7 days a week. A user that logs in will be able to execute an experiment if the login time is within her time slot, i.e. within the reservation time plus two hours.

Figure 4: Real-time configuration of a PC and a router during an experiment execution. The message box shows the lab devices being used along with their randomly-generated ports.

In each experiment, one or several topologies are presented to the user; the latter needs to connect the desired topology by pressing on the “Connect Topology” button; this action will send the configuration file corresponding to the topology from the LS to the *iSwitch*, and will enable the latter to configure the VLAN memberships dynamically. After that, the user can click on any PC or router in the network and configure it; every device will be assigned a random port on the LS, and the user willing to access that device needs to initiate a Telnet session to the public IP of the LS on that port. Figure 4 shows a sample lab experiment as the user is executing it. Unfortunately, the GUI-enabled “Ethereal” application couldn’t be made accessible to users, since they are using a non-graphical Telnet command window to access lab devices; however, users may use “tcpdump” which is a powerful tool that allows them to sniff network packets and make statistical analysis of the network traffic.

Besides, a real countdown timer always appears on the experiment's web page, indicating the remaining duration in the reserved time slot. When the allotted time slot expires, the user is notified, and is given an extension of fifteen minutes to wrap up. If the extra time expires, or if the user finishes within her time slot, all opened ports on the LS are closed and a special configuration file will be sent to the *iSwitch* in order to reset all VLAN memberships.

In addition to the predefined lab experiments^{††}, our system offers the users the powerful feature of building any network topology they desire by accessing a “free-experiment” web page. As shown in Figure 5, this option allows students to connect any topology by using the four PCs and four routers. The user adds the interfaces of these devices to the desired subnets, and the network diagram will be drawn in the web page on-the-fly. In this case, the configuration file will have to be written dynamically using the C# code, saved on the LS and then sent to the *iSwitch* via the TFTP server. Configuring the PCs and routers in the free-style experiment is similar to the case where the experiment topology is predefined.

Figure 5: Building any network topology on-the-fly

Finally, our system enables the students to upload their lab reports into the LS, and the graduate assistants (GAs) to download these reports in order to grade them. Furthermore, GAs can manage user accounts and experiments in order to check the students' progress in the course.

VI. Assessment

In this section, we investigate the students' reaction to the advent of the Remote Internetworking Laboratory (RIL) and use their feedback to suggest possible enhancements to our system. Moreover, we explore how far the RIL meets the ABET criteria for Engineering programs in general, and laboratory experience in particular.

A. Student Response and Future Work

The RIL design was fully implemented as part of an eight-month Final Year Project at the ECE department at AUB, during the 2004-2005 academic year. Students and faculty members who tested the remote lab expressed considerable enthusiasm to the idea of the project, and believed that it would be a very important complement to the weekly laboratory sessions.

The system is not yet integrated in the internetworking lab course, but nevertheless, was tested by a number of students who already took the course or who were familiar with networking concepts. When asked to comment on the online lab, many students found the reservation scheme very useful but believed that the current system will get crowded once it gets fully integrated in the lab course. We also anticipate that in the future the remote lab will be allowed for use by students from other departments and possibly from other institutions. As a potential enhancement, we can allow multiple racks (i.e. sets of four PCs and four routers) in the internetworking lab to be used simultaneously; this property seems effective especially in the presence of traffic of users needing to access the system. The design should make it possible to check for the availability of racks and the traffic on them, and accordingly, reserve a session on an available rack. Thus, in such a system, a user is more liable to find an empty time slot to reserve.

Another insightful feedback we got from students is that since they work in groups of two in the actual lab, they would also like to be able to carry out the experiments jointly with other students. This comment leads us to think of possible ways of extending our system to allow collaboration among several remote users; techniques such as those described by Hua and Ganz⁷ may be used in the future to enable collaborative execution of experiments.

B. ABET Requirements

In what follows, we show to what extent the RIL meets some of the key requirements set forth in Criterion 3 of the ABET Engineering Accreditation Criteria.⁹ We also extend our analysis by showing how the RIL meets the instructional laboratories' objectives proposed in the ABET/Sloan Colloquy^{††} (these were enumerated by Feisel and Rosa¹⁰).

ABET Engineering Practice Requirements	ABET/Sloan Colloquy Engineering Lab Objectives	Contribution of the RIL
a) An ability to apply knowledge of mathematics, science, and engineering e) An ability to identify, formulate, and solve engineering problems	1) Instrumentation 2) Models	The RIL allows students to work with network-related engineering problems, and enables them to solve them remotely
b) An ability to design and conduct experiments, as well as to analyze and interpret data c) An ability to design a system, component, or process to meet desired needs	3) Experiment 4) Data analysis 5) Design	Students can execute pre-set or free-style experiments anywhere and at anytime. The "Free-Experiment" web page allows users to design any topology on-the-fly; students can use this page to design a network that meets certain criteria. The ability to analyze and interpret data is enabled by applications such as the "tcpdump" monitoring tool

d) An ability to function on multi-disciplinary teams	11) Teamwork	Students from different disciplines can execute experiments as long as they have a valid account on the online system. Allowing collaboration in the future will enhance teamwork capabilities
j) A knowledge of contemporary issues	—	Faculty/GAs can integrate technical articles or other publications into the online system. Students can thus combine education with knowledge of contemporary problems related to computer networks
k) An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice	—	Students use the latest software tools and Internet-based techniques to access the lab remotely and configure devices in real-time.
—	9) Safety	Unsafe use of network devices is diminished, since safeguards are built into the software at the LS level

Table 1: The RIL versus some of the key ABET requirements. Note that letters assigned to the ABET requirements and the numbers assigned to the ABET/Sloan objectives correspond to their actual numbering in their original source

As we can see from Table 1, the RIL meets the key engineering accreditation requirements and allows students to satisfy the major laboratory accreditation objectives. It is important to note that incorporating collaboration techniques into the system will certainly enhance teamwork capabilities and will enable remote users to share knowledge and experience; but still, any major improvement over the current implementation has to take into consideration the expected increase in design complexity and cost.

Certainly, the RIL has a main value added to the traditional engineering experience: it is the ability of students to execute experiments on their own schedule. Nevertheless, it is unclear if this feature will enhance their design and data analysis capabilities.

VII. Conclusion

In this paper, we presented the design of a remotely accessible internetworking laboratory at the American University of Beirut. Our project allowed users to access the lab from any computer and conduct real-time experimentation on the lab's equipment while being physically distant from the lab facility. This project made it possible to overcome the limitations of standard internetworking experiments conducted within the lab, mainly the fixed time and space limitation, and the constraint on the availability of expensive network devices.

Our design is essentially interactive, in the sense that users can configure lab devices and send network commands in real-time, as if they were physically located inside the lab facility. Connecting lab devices through cables is eliminated by using a network switch that interconnects the devices internally and dynamically via the VLAN technique. An important aspect of the

design is that, in addition to pre-set network topologies, users are given the option to build *any* topology on-the-fly. Finally, the system includes features such as time-slot reservations, and management of user accounts, lab reports and experiments, all of which make learning of networking concepts more efficient and the online experience much closer to reality. Based on students' feedback, we believe that the design can be enhanced by allowing multiple racks to be used, and by potentially enabling collaboration between remote users.

Bibliography

- ¹ C. Baillie and E. Hazel. (2004) Teaching materials laboratory classes. [Online]. Available: <http://www.materials.ac.uk/guides/labclasses.asp>
- ² T. A. Fieldly and M. S. Shur, *Lab on the Web: Running Real Electronics Experiments via the Internet*. Wiley-IEEE Press, October 2003.
- ³ D. Z. Deniz, A. Bulancak, and G. Özcan, "A novel approach to remote laboratories," in *33rd ASEE/IEEE Frontiers in Education Conference*, Boulder, Colorado, November 2003.
- ⁴ B. A. Foss, K. E. Malvig, and T. I. Eikaas, "Remote experimentation - new content in distance learning," in *Proceedings of the International Conference on Engineering Education (ICEE'01)*, Oslo, Norway, August 2001.
- ⁵ I. Mougharbel, A. E. Hajj, H. Artail, and C. Riman, "Remote lab experiments: A comparative study," *International Journal of Engineering Education*, 2005, accepted for publication.
- ⁶ S. Yoo and S. Hovis, "Remote access internetworking laboratory," in *Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education*, Norfolk, Virginia, March 2004.
- ⁷ J. Hua and A. Ganz, "A new model for remote laboratory education based on next generation interactive technologies - A generic laboratory plug-in using MS ConferenceXP learning infrastructure," in *Proceedings of the ASEE New England Regional Conference*, Orono, Maine, 2003.
- ⁸ Cisco Systems. (June 2005) Configuring a terminal/comm. server. [Online]. Available: <http://www.cisco.com>, document ID: 5466.
- ⁹ The Accreditation Board for Engineering and Technology. (October 2005) Criteria for accrediting engineering programs. [Online]. Available: <http://www.abet.org/forms.shtml>.
- ¹⁰ L. D. Feisel and A. J. Rosa, "The role of the laboratory in undergraduate engineering education," *Journal of Engineering Education*, ASEE, Vol. 94, No. 1, January 2005.

* The *iSwitch* and the *iRouter* are transparent to users. The *iRouter* is also connected to the console line of the *iSwitch* because the LS web application needs to access the *iSwitch* and configure it, e.g. during topology setup

† The library is available for download from <http://www.klausbasan.de/misc/telnet/index.html>

‡ Trivial File Transfer Protocol. A simpler and faster version of FTP

§ Internetwork Operating System. An operating system used by Cisco as the main control program in its routers and switches

** We used FPipe.exe. This is a free port-redirection tool that can be downloaded from <http://www.foundstone.com/resources/proddesc/fpipe.htm>

†† The lab experiments that we put online correspond to those available in the book currently used by the lab students: "Mastering Networks: An Internet Lab Manual" (Addison-Wesley, 2003), by J. Liebeherr and M. E. Zarki

‡‡ The ABET and the Alfred Sloan Foundation organized a colloquy in San Diego, CA on January 6-8, 2002 and agreed on thirteen essential objectives of instructional engineering laboratories