Resource Management Portal for Laboratories Using Real Devices on the Internet

Stefan Zimmerli
University of Bern
Neubrückstr. 10
CH-3012 Bern
+41 31 631 86 47
szimmer@iam.unibe.ch

Marc-Alain Steinemann
University of Bern
Neubrückstr. 10
CH-3012 Bern
+41 31 631 86 47
steine@iam.unibe.ch

Torsten Braun
University of Bern
Neubrückstr. 10
CH-3012 Bern
+41 31 631 49 94
braun@iam.unibe.ch

Abstract

Internet-based distance learning is slowly gaining new territories and substituting current teaching methodologies. However, distance learning not only consists of transferring documents to web pages, but also of developing new concepts, methods, and implementation architectures. This article presents concepts and implementation issues for an example remote hands-on networking laboratory. The described course gives access to real network hardware via the Internet. In particular, authentication, authorization, scheduling, and error recovery issues had to be solved.

General Terms

Management, Measurement, Performance, Design, Experimentation, Human Factors.

Keywords

Distance learning, resource management, computer networks laboratory, hands-on training.

1 Introduction

Internet-based distance education in Switzerland is closely related to the initiative of Swiss government and the formation of the Swiss Virtual Campus (SVC) [1, 8] promoting Internet-based studies at university level. The Virtual Internet and Telecommunications Laboratory of Switzerland (VITELS) [10] is one of fifty projects within the SVC program.

Five Swiss universities elaborate a laboratory for virtual and remote exercises in the area of telecommunications and computer networks. Each of the participating institutes develops the laboratory modules rather independently from each other but within a well defined common framework. The term module in the area of e-learning is similar to the meaning of a book chapter. To the time this article has been written, five VITELS modules are opened for student access (Simulation of IP Network Configuration, IP Security, Firewall Management, Sockets and Remote Procedure Calls, Remote Method Invocation) and more are in preparation. The entire course is integrated into a commercial web course platform. This platform offers helpful tools such as an exercise design and management system with quizzes and self tests, student collaboration tools such as discussion boards, chat and a glossary but also student rating and tracking tools. It also leads the students through the course with its modules such as a red thread. Content that cannot be integrated in a commercial course platform is provided by external servers but can then be reached via the course platform.

At University of Bern we have developed the module IP Security for the VITELS course. Students can remotely configure commercial network router devices. To achieve the manageability of such devices, new techniques and mechanisms that allow remote access for configuration of real network devices from the Internet had to be developed and implemented. The goal was to have the same graphical user interfaces for the configuration tasks as students would find in reality.

A general requirement in VITELS is that all the exercises must be accessible by any common web browser on any wide spread operating system. Students must not be forced to install additional software on their private computers.

For managing course access, a highly flexible and open architecture has been developed that provides authentication, authorization and scheduling functions [7]. Students get a unique user name with a single password that can be used throughout the whole course. Scheduling functions for expensive laboratory network equipment help to share the limited resources. Students must be prevented from disconnecting themselves from the course hardware by misconfiguration. In the worst case, the course hardware can fail and must be reset automatically. Each time before a new student starts an exercise session, the hardware must be reset.

This article describes the functionality and the technical background of one VITELS module. The second chapter describes related work and compares it with our approach. The third chapter is devoted to didactical issues. The fourth chapter describes the VITELS architecture. The fifth chapter describes the module IP Security, which serves as an example module for the design and implementation of our course architecture. Chapter six contains a summary and conclusions.

2 Related Work

VITELS combines theory sections, exercise sections, hands-on work on real network devices and simulation tasks in a very modular course framework. Many other network courses do not integrate work on real devices. Others open the devices to the entire course during a certain time interval for the whole class. A disadvantage of this method is that one never knows who exactly accesses the hardware and one cannot exclude the possibility of duplicate log-ins. Nevertheless, this system works fine for courses such as provided by Verband der Elektrotechnik Elektronik Informationstechnik e.V. (VDE), called Internet: Vom Basiswissen zum Netzmanagement. [9]. In that course, basis knowledge about the Internet and network management is taught.

Another well established course system with hardware management and scheduling is run by Mentor Technologies [3]. The course offers theory, exercise and hands-on work on real network devices. They are dedicated to a particular vendor of router products and offer the possibility to configure the whole range of those products. Unfortunately no details about the technical solutions behind the course are available.

There exist several federated, huge network laboratories built on the EmuLab Technology. Within the project "EmuLab Classic" they offer access to several hundred PCs to do network emulation and experimentation. Universities are able to request these resources for experiments and get full access to a certain number of nodes. The access is granted through secure shell clients or WWW interfaces. The PC nodes can be used as edge nodes running arbitrary programs, simulated routers, traffic-shaping nodes, or traffic generators. While an experiment is running, the acting students get exclusive use of the assigned machines, including root access. A big difference to our approach is that EmuLab emulates hardware and does not provide real device as they are. EmuLab emulates everything on PCs and not on commercial router devices as we do. Experiments done on real devices are closer to reality than emulations [11].

Another Swiss Virtual Campus project called Nano-World deals with nano technologies and provides access to real electron microscopes. The course management problems are closely related to those of VITELS and so is the solution. Remote experiments are also very popular in engineering [2].

3 Didactical Issues

Many changes had to be done for the switch from a traditional computer networks laboratory to an Internet-based distance laboratory. A major difference between the two cases is the way students work and study. While students worked as teams in the traditional in-house hands-on work sessions, they are alone in front of their computer screens in the distance hands-on work sessions. Figure 1 shows the main differences of the studying styles in the two learning environments.

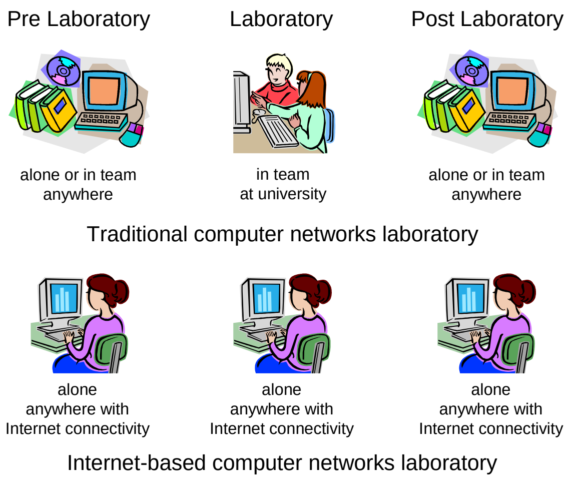

Pre Laboratory
Laboratory
Post Laboratory
alone or in team anywhere
in team at university
alone or in team anywhere
Traditional computer networks laboratory
alone anywhere with Internet connectivity
alone anywhere with Internet connectivity
alone anywhere with Internet connectivity
Internet-based computer networks laboratory

Figure 1. Different learning environments in traditional and distance learning.

The didactical concept of the traditional course had to be changed completely. The theoretical sections for the distance course must contain the complete learning material to gain the required knowledge level for the course modules. Exercises had to be developed for the tools offered by the chosen commercial course platform. Interactive content has been created to enrich the theory. New tools such as on-line discussion boards have been included to help replacing the teamwork in the in-house laboratory. Although many valuable aspects of the traditional laboratory work have been eliminated by the change to the distance course, many other good aspects such as the permanent availability of discussion threads have been added. A major novelty is the availability of the laboratory around the clock and seven days a week. Now it is also possible to give access to external students to the laboratory. These issues raised many questions about student support. A first test of the module IP Security with a special emphasis on the differences between traditional and distance education is described in [6].

4. Design and Implementation of a Remote Computer Networks Laboratory

4.1 Remote Laboratory Architecture

The whole VITELS course architecture is depicted in Figure 2. A central part is the lightweight directory access protocol (LDAP) directory-based course reservation system. With the reservation system, students can book time slots for each of the laboratory modules. If they want to access a module with a limited number of hardware devices, they get authorized by the reservation system, if the module was booked before by the respective user.

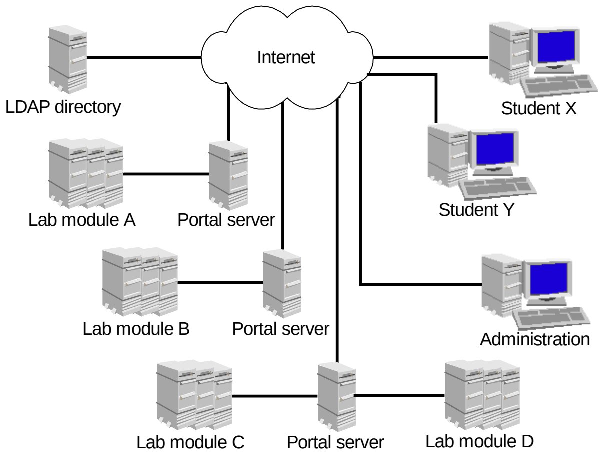


Figure 2. VITELS Architecture.

Another part of the architecture is the administration. Administrators maintain the data base, add or delete students, set module slots or change / add new modules. Students are users such as administrators but with less permission rights. They only get into contact with the reservation system over graphical user interfaces.

In the VITELS architecture, hands-on session modules are connected by a gateway that is called portal server. Portal servers manage the connected laboratory equipment and are located between the module specific hardware on one side and the students on the other side (Figure 2). Normally, there is one portal server per module but it is also possible to connect more than one module to a portal server. It is possible to connect any device that has an interface to a computer (for example a serial interface) to a portal server. Portal servers also provide firewall functions as they have at least two network interfaces, one to the internal network and one to the external network, i.e. the Internet. The VITELS reservation system and the portal server of the module IP Security are running on Linux.

4.2 Laboratory Reservation System

Access to the computer networks course is only granted to subscribers. Subscriber and module scheduling information is stored in a central or in distributed LDAP data bases. The collected information contains standard attributes such as names, given names, user names and passwords as well as special attributes such as module scheduling information. Module-specific information is for example the time slot table of a specific module. Figure 3 shows the chosen LDAP tree structure.

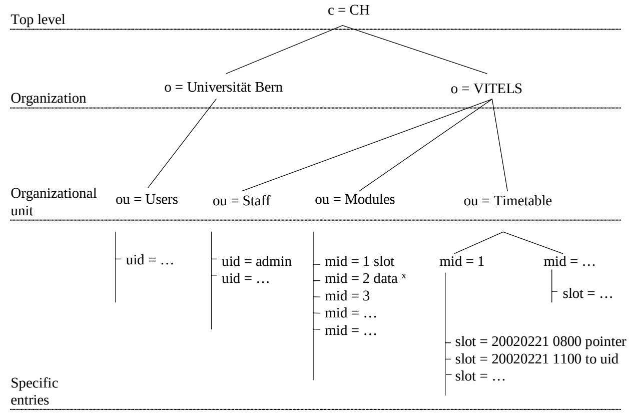
The organization VITELS splits up in three organizational units, the staff where module administrators are stored, the modules, where slot data for the modules are stored (first slot of a day, slot length, amount of slots per day, and a pointer to the user identity from the current module user in the timetable), and the timetable, where the time slots for each module are stored.

LDAP has been chosen because it is designed for fast read access and because querying clients can use user names and passwords stored in the directory as if they were stored locally on the client computers. Students book modules rather infrequently but the modules' entry points, the so called portal servers, query the LDAP directory up to sixty times per hour to retrieve the current module user. The directory is maintained by several staff members at each institution and is accessible for all of the portal servers of the course. Based on the experience with the traditional course, for the Internet-based course, four-hour timeslots were defined for the module IP Security.

Each module entry in the LDAP directory provides a "current" user to the respective querying portal server. In the case of the module IP Security, the portal server retrieves user name and password of the current user and stores them for the duration of the slot in the local password file at the beginning of each of the time slots. This is done by a PERL script on the portal server that is called by a CRON job. The script searches for any processes that have been created by previous students that did not log out correctly. Still running processes are terminated and possibly existing lock files written by the terminal program are deleted. Subsequently all the router and host passwords on the portal server are changed to the scheduled user's password. This guarantees that only the current user can access the laboratory devices with his personal password stored in the LDAP data base.

This is achieved by changing the encrypted host and router user's password entries in the /etc/shadow file on the portal server. If there is no current user for a time slot all the router and host logins get blocked. Because the password entries are stored encrypted in the /etc/shadow file and have a fixed length, a login can be blocked by setting the encrypted password string to a random string with a length smaller than the usual fixed length. It is impossible to find a plaintext password string whose encrypted result will match the stored string.

At the end of the time slot, any of the router and host user's processes are terminated and their respective passwords are set to the next students' password. This mechanism ensures that at the beginning of each time slot the previous student has logged out, even in the case that he or she did not close the secure shell applets or the browser or did not terminate his measuring processes on the Linux hosts. Since the connection from the students' computer to the portal server is encrypted by using a secure shell login, it is not easy to sniff the student's password.



slot data ^{x} = firstslot = 0000 slotlen = 3 numslot = 8 [pointer to uid from current mid user in ou = Timetable]

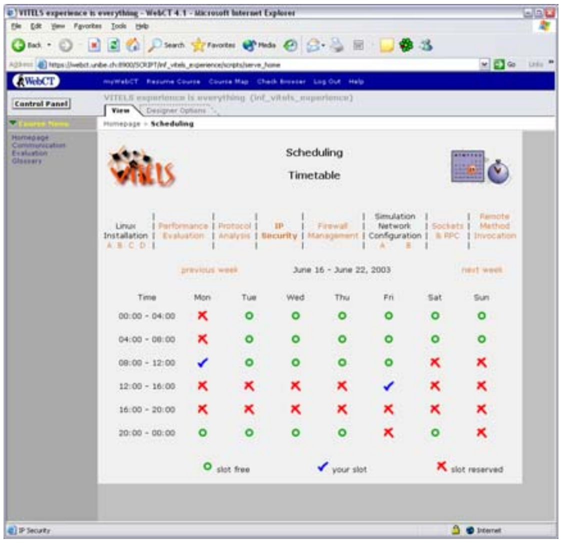
Figure 3. LDAP tree of the data base.

4.3 WWW Interfaces

Before a student can login to a laboratory module, he or she has to book a time slot for the desired module. The WWW interface for booking modules looks as shown in Figure 4. The booking page is a PHP-based front-end to the scheduling data stored in the LDAP directory.

By clicking on a symbol for free slots or for own slots, students can reserve or release time slots. Slots indicated as reserved are booked by other students. Staff members possess additional rights and can view the students' names, free slots and define slot times and durations.

When a student tries to access a module on a portal server, he or she first gets to the WWW interface for accessing the laboratory. The login page for the hands-on session contains user name and password fields. A drop down menu named institute allows students to select their home organization. This is only needed, if more than one user LDAP data base are linked together. In this case, the portal server might query the respective LDAP server of the user's home organization for authentication purposes. After a successful login to the module, a session cookie is stored in the user's browser. This session cookie is valid up to the end of the booked time slot. The students get to the laboratory device configuration as depicted in Figure 5.



Scheduling

Timetable

June 16 - June 22, 2003

Time	Mon	Tue	Wed	Thu	Fri	Sat	Sun
00:00 - 04:00	✗	○	○	○	○	○	○
04:00 - 08:00	✗	○	○	○	○	○	○
08:00 - 12:00	✓	○	○	○	○	✗	✗
12:00 - 16:00	✗	✗	✗	✗	✓	✗	✗
16:00 - 20:00	✗	✗	✗	✗	✗	✗	✗
20:00 - 00:00	○	○	○	○	✗	○	✗

○ slot free ✓ your slot ✗ slot reserved

Figure 4. The Module Reservation system interface.

For each network device that is configurable, a corresponding link has been created on the web page. By clicking on one of these links a new browser window pops up, which then downloads a signed java applet to the student's computer. This applet is a complete secure shell client written and provided by [4].

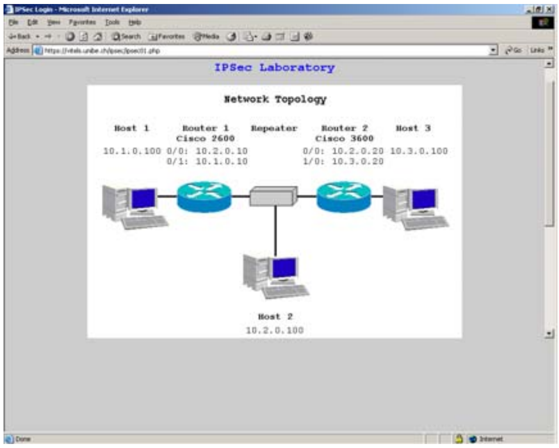
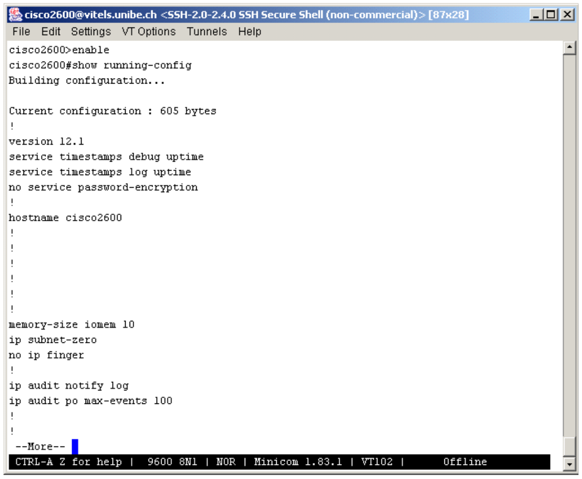


Figure 5. IP Security module, laboratory configuration.


cisco2600@vitels.unibe.ch <SSH-2.0-2.4.0 SSH Secure Shell (non-commercial)> [87x28]
File Edit Settings VT Options Tunnels Help

```
cisco2600>enable
cisco2600#show running-config
Building configuration...

Current configuration : 605 bytes
!
version 12.1
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname cisco2600
!
!
!
!
!
!
memory-size iomem 10
ip subnet-zero
no ip finger
!
ip audit notify log
ip audit po max-events 100
!
!
 --More--
 CTRL-A Z for help |  9600 8N1 | NOR | Minicom 1.83.1 | VT102 |      Offline
```

Figure 6. Remote access to router through web browser with secure shell applet.

After starting the applet on the client computer, the student is able to establish a secure connection to the selected laboratory device, which is in fact a secure shell connection from the student's client computer to the portal server. One of these connections is showed in Figure 6. The student then gets a set of open windows to the respective devices as it is in real life too.

5 Details to the Implementation of the Module IP Security

5.1 The Traditional Laboratory Course Module IP Security

The remote laboratory originates from a traditional, in-house computer networks laboratory course performed at University of Bern. In the traditional network laboratory, students have to be present in a network laboratory for several hours and perform exercises with real network hardware. In one course module, students had to connect two routers and three hosts, to set up the routing tables and to perform traffic measurements with and without established Virtual Private Network (VPN) tunnels. For these tasks, a network consisting of repeaters between the routers as shown in Figure 7 had to be used. The practical work was preceded by theoretical work and reading preparation material in books or WWW pages. Before students could begin with the hands-on training they had to pass a knowledge test. Students documented their performed laboratory work with notes and log files from routers, hosts and measurement programs such as Tcpdump and NetPipe [5]. The last section of the module consisted of exercises about the performed work and of analyzing the gained laboratory results and log files.

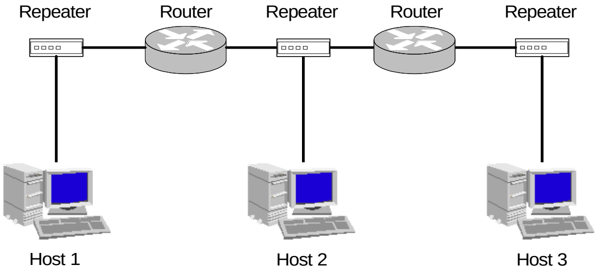


Figure 7. Configuration in the traditional module IP Security.

5.2 Technical Aspects of the Remote Module IP Security

Figure 8 gives an overview of the IP Security module implementation. The portal server is connected to the Internet and to the laboratory hardware by Ethernet links and acts as a gateway between them. There are three network interfaces to the laboratory devices, i.e. one to each repeater. The repeaters are between host 1 and router 1, host 2 and router 2, and between the routers and host 3. In this way, all the network interface cards of the laboratory devices can be reached by Ethernet links. However, for managing the routers it is not enough to have Ethernet links to the routers because they can easily be shut down and the routers might become unreachable. For that reason, the portal server also has serial links to each of the router's console ports. These serial links allow permanent configuration access to the routers by the portal server.

For each laboratory device (router 1, router 2, host 1, host 2 and host 3) a dedicated user account has been created on the portal server. When the laboratory user logs into the portal server he or she is automatically forwarded to the corresponding laboratory device and is able to configure it. This mechanism was realized for the three hosts by changing the users' login scripts that automatically forwards them to the corresponding host via the standard "rlogin" command and logs in. "rlogin" does not represent a security risk, because these logins are transmitted over an internal subnet, which is separated from the Internet through the portal server.

Unfortunately, the same mechanism could not be used in the same manner for the router users as for the host users. A login to the routers via the Ethernet link is not possible as long as the routers' links are not set up (for example after a reset) or are down due to a misconfiguration (for example by the student). To prevent management problems of the routers, the router users' login scripts start the terminal program directly after login. The terminal program is installed at the portal server. The settings of the terminal program are pre-configured to connect the user to the router's console port via the serial link. With this mechanism it is always possible to access the routers independent of whether their Ethernet links are up or not. When the router user quits the terminal program, he or she is automatically logged off from the portal server. With these two procedures, access to the laboratory hardware is granted through the portal server.

The portal server runs an Apache web server with the modules PHP, Perl and SSL loaded. PHP scripts creating dynamic HTML pages have been developed. These pages are the main interfaces between the students and the network hardware and allow them to connect to the laboratory hardware.

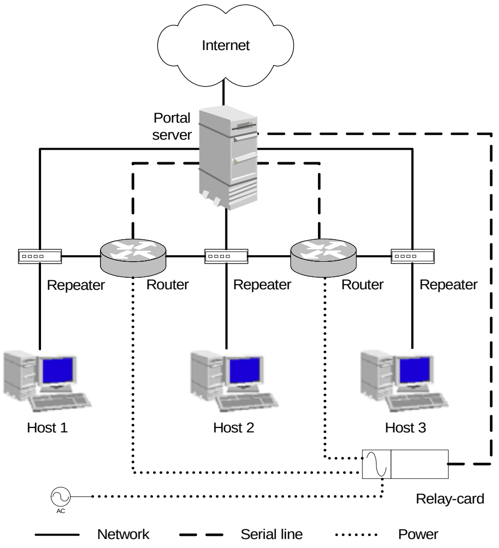


Figure 8. Portal server and laboratory hardware.

5.3 Error Recovery for Connected Laboratory Devices

Configuring routers is not a simple task and students without experience can get lost during the configuration. The routers might even be configured such that only a reset can help or it might happen that students lock out themselves. Students must have administrator access to the routers to setup Ethernet links and to configure the routing. A person with administrator access to the routers can also set a new administrator password and block the routers for others. To prevent this, a hard reset mechanism has been implemented for the two Cisco routers that are used in the IP Security module. By means of this mechanism, routers can be reset to a minimal configured state and administrator passwords deleted. The mechanism uses a documented Cisco password recovery solution. The router has to be power-cycled and then the routers can be configured over the console port as described below.

Usually this is done by connecting a laptop with its serial-line to the router's console port. After having the router power-cycled (power off and power on), one has to send a break signal to the router. This brings the router into the bootstrap program where it is possible to load another system image (called Internet Operating System IOS by Cisco) into the router. In this special maintenance mode one can configure the router such that it “forgets” its current configuration, especially a set administrator password. This is achieved by changing a configuration register which holds the address from where the routers initially boot the IOS. This current stored address has to be saved for later use. After setting it to an invalid value and rebooting the router, its configuration is set to a minimal state and the administrator password is deleted. Then one has to log-in again and set the configurations-register value back to the old address saved before. A reboot is performed again and the router is again ready to be used.

To automate the power cycling a relays card has been inserted between the routers and the electrical power socket. The relays card is connected to the portal server’s serial port and consists of a small microprocessor that controls the relays’ state. The communication between the portal server and the relays cards’ microprocessor is based on a proprietary protocol that uses four-byte frames commands. For each command frame sent from the portal server to the micro-controller, an answer frame is sent back to the portal server.

A Perl-based driver library has been developed, which implements the frame-based protocol. Because the library communicates with the relays-card’s microcontroller over the serial line, the existing Perl module Device:SerialPort was included. This Perl module provides functions for sending and receiving data over a serial-line interface and is the basis for our developed driver library. Further the module Device:SerialPort offers a function to send a break-signal over a serial-line, which is needed to put the booting router into the bootstrap program.

Using the developed library functions it is possible to control the relays card and to set and get the relays’ state. Therefore, it is possible to power on and power off each of the routers by switching the corresponding relays of the relays card on and off. As described above, for fulfilling the password recovery, a break signal must be sent to the router and the changes of the configuration register has to be done. This has been achieved by implementing a Perl script that uses the relays card library. This script brings the router into the bootstrap program after the power cycle and sends the confreg-command over the serial line as if an administrator would do by connecting a laptop to the router. Then the script sends the reboot command to the router and waits till it is done. At that moment a before set router password has been deleted, but the saved configuration register’s value must be written back. This is done by the script and the router is rebooted a second time. Because the router has to be rebooted twice during the recovery procedure it consumes about five minutes to get to the initial configuration state.

5.4 Future Extensions

More features can be implemented for enhancing the IP Security module. Students could select among different IOS versions for the routers. This could be achieved by setting up a Trivial File Transfer Protocol-server (TFTP) that stores a set of IOS files on the portal server. The already described reset script that brings the routers into the bootstrap program could be changed in a way that it does not change the routers configure register, but sends the commands for connecting to the portal server and for downloading the chosen IOS image. After a scripted reboot the router is ready for use with a different IOS.

In the current state, the network configuration of the above described Linux hosts is pre-configured, because students need them only for measuring tasks. If students should learn to configure the hosts based on a fresh installation, they should have full administrator access to the hosts. Because full access to the hosts might result in an unstable or corrupt system, a possibility to reset them to a known state would be desirable. This could be done by plugging the relays card between the hosts’ power and the power socket and power cycling them in the same manner as the routers. The hosts BIOS would have to be configured to boot from the built-in CD ROM drive or the disk drive. This bootable medium should contain a small Linux system that would get an IP address from the portal server via dynamic host configuration protocol (DHCP) and then fetch an image file from the TFTP server. The image would be extracted on the host and the Linux system stored in the image would be installed. After a final reboot the hosts would be ready again.

The measuring tasks of the above described module results in output from tools such as Tcpdump and NetPipe. At the moment, the students collect theses files and send them to the teacher for reviewing them. Therefore, a mechanism that checks automatically the student’s configuration and measurement results could be developed. To copy the actual router configuration to the portal server the routers command “copy” can be used because it is able to copy files to a TFTP server. This operation could be scripted by enhancing the existing reset script. Then, the file on the portal server has to be parsed to check if all of the needed configure steps have been done and if they are in the right order. Unfortunately, the parser would have to be changed for different IOS and for different exercises. To copy the obtained measuring results, which are stored on the Linux hosts to the portal server, the command “rcp” can be used, but parsing this file strongly depends on the kind of measurement exercise and the used tools.

6 Summary and Conclusions

This article demonstrated that it is possible to build up remote hands-on laboratories for Internet-based computer networks courses. The inclusion of real devices and not only simulations allows teaching with the same means as in traditional in-house laboratories. From the didactical point of view there are still many obstacles to overcome until reaching the same high level of education as in the traditional courses. From the technical point of view most obstacles are already left behind and the main tasks lie in connecting the devices to the portals.

The above presented VITELS architecture is presented on the VITELS Internet page (www.vitels.ch), where it is also possible to download the described software and to visit a demonstration course.

The architecture allowed connecting the exemplar module IP Security to the VITELS course system and to perform tests with students. The remote control of the routers and hosts works fine and no problems have arisen.

The use of a portal server that serves as an interface to the rest of the course architecture, to the students and to the laboratory hardware simplifies Internet-based course architectures for distance education significantly. This example shows that it is possible to manage a wide range of devices by a portal server and thereby to offer Internet-based distance learning courses with real devices and hands-on work.

Connecting commercial hardware to hands-on laboratories is not an easy and cheap way. Simulations of devices would allow much more users at the same time and require much less hardware than real devices. But there are major disadvantages of simulations. Implementations of simulations of devices such as routers are time and manpower demanding projects. If device's operating systems are updated, real devices can be updated and still work as before. Simulations have to be developed further. Another advantage of real devices lies in the possibility to exchange those devices without having a lot of additional work.

6 References

[1] Bachmann G., Haefeli O., Kindt M.: Campus 2002. Die Virtuelle Hochschule in der Konsolidierungsphase. Waxmann, 2002, ISBN 3-8309-1191-2

[2] Guggisberg M., Fornaro P., Gyalog T. and Burkhart H.: An Interdisciplinary Virtual Laboratory on Nanoscience, Electronic Notes in Future Generation Computer Systems, Elsevier, Vol. 1 (2001)

[3] Mentor Technologies, vLab Technology, http://www.mentortech.com/vlab/index.shtml/

[4] Mindterm, http://www.appgate.com/mindterm/

[5] NetPipe, a Network Protocol Independent Performance Evaluator, http://www.scl.ameslab.gov/netpipe/

[6] Steinemann M.-A., and Braun T.: Remote versus Traditional Learning in a Computer Networks Laboratory, Communications and Computer Networks (CCN 2002), Cambridge, USA, November 4-6, 2002

[7] Steinemann M.-A., Zimmerli S., Jampen T. and Braun T.: Global Architecture and Partial Prototype Implementation for Enhanced Remote Courses, Computers and Advanced Technology in Education (CATE 2002), Cancun, Mexico, May 20-22, 2002

[8] Swiss Virtual Campus, http://www.swissvirtualcampus.ch/

[9] Verband der Elektrotechnik Elektronik Informationstechnik e.V., Internet: Vom Basiswissen zum Netzmanagement, http://iuk.in-chemnitz.de/

[10] Virtual Internet and Telecommunications Laboratory of Switzerland, http://www.vitels.ch

[11] White et al., An Integrated Experimental Environment for Distributed Systems and Networks (full report), Technical Report, May 2002; Revised version to appear at OSDI 2002, December 2002.