Global Architecture and Partial Prototype Implementation for Enhanced Remote Courses

Marc-Alain Steinemann, Stefan Zimmerli, Thomas Jampen, Torsten Braun
Institut für Informatik und Angewandte Mathematik, Universität Bern
Neubrückstr. 10, CH-3012 Bern
Email: [steine/szimmer/jampen/braun]@iam.unibe.ch

Abstract. Internet based courses require new architectures that support features like authentication, authorization, accounting, scheduling and in some cases controlling of additional hardware resources. This article describes an open architecture for remote courses in which many locally distributed clients attend exercises on many locally distributed servers. The architecture includes lightweight directory access protocol for user management and public key infrastructure for the security needs. All traffic is encrypted, either with IPsec between the course servers or secure shell and secure sockets layer between clients and course servers. Although all connections are secured, students can work with any standard web browser and don't need to install additional software. In a partly prototype implementation a module of a remote network laboratory with student's access to real network hardware will be presented, showing the technical and didactical difficulties encountered.

Keywords: remote learning, network laboratory, architecture

1. Introduction

New technologies smooth the way to new methods. The Internet with its manifoldness of services from 3W to Email, from discussion boards to real time chat is the technology in talk. Mostly, new technologies don't cause big impact in humankind if they are not accessible to a broad community. In the case of the Internet the user community has already reached such a size that the Internet has become a matter of course. Hence, one conclusion that results is that offering on-line courses is no longer a temptation for freaky teachers but simply a must for most educational instances. It is a demanding task to develop new learning methods like on-line tutorials, courses and laboratories, seen from the didactical and the technical perspective, especially after visiting many of the bad designed currently available remote courses. It is not possible to copy paper scripts page per page to the Internet. Theoretical sections must be mixed with exercises, linked to background information, supplemented with interactive content and finally, exercises must be automatically processed and evaluated to reduce the teachers' work. Well-designed on-line courses allow teachers to offer the content accompanied by a higher level of support to more students than in a traditional way.

A step beyond the above mentioned on-line courses includes the integration of remotely controllable hardware. Imagine a biologist that would like to offer a microscopy course or an engineer that would like to offer a micro electronic course.

The new enhanced courses should be easily accessible for the respective users, wherever they are and whatever kind of hardware they possess, regardless of the connection speed. Architectures that offer those features should also contain authentication, authorization and in most cases accounting functions.

This paper presents an architecture that brings technologies and methods together, offering possibilities for the student management and covers current security needs. Our new architecture opens the Internet for effective distance learning.

The impulse to work in the field of remote education came from a project where several Swiss universities are working together towards a common course in the area of computer communication. The goal is to design a course consisting of several modules developed rather independently by the various partners. The motivation behind this activity is to combine the limited available human and equipment resources required to develop and maintain such a course.

The work is being performed within the project called Virtual Internet and Telecommunications Laboratory of Switzerland VITELS [1], which is one of several projects within the Swiss Virtual

Campus SVC [2] program funded by the Swiss ministry of education and science. Each partner of the VITELS project - four universities (Bern, Fribourg, Genève, Neuchâtel) and one engineering school (Fribourg) - is currently developing modules based on the own competence and equipment. The seven modules focus on Linux System Installation and Configuration, IP Network Simulation, Configuration and Performance Evaluation of a Real IP Network, Client/Server Programming, Protocol Analysis, IP Security, and Firewall Management.
Each participating university develops and maintains its modules within its own laboratory environment, but allows remote students to access and use the laboratory infrastructure via Internet technology. The entire course must appear to the user as being homogeneous, although it is distributed over several locations in Switzerland. A web-learning environment called WebCT [3] is used to lead through the course modules.

2. Related Work

The best example of an on-line remote laboratory course is Mentor Technologies' [4] vLab. It offers laboratory modules with real network hardware (routers and switches) and scheduling for user access. Unfortunately, vLab is a centralized system, they use a single location for the servers and only the clients are geographically distributed. Mentortech is not a charitable organization and therefore no detailed technical information is available.
An excellent remote course is offered by the TU Chemnitz [5]. A big difference to our approach lies in the authentication and authorization architecture. In the courses of the TU Chemnitz entire classes get access to one specific laboratory during a longer time period, whereas we only open modules to pre-registered students that have booked the respective laboratory on-line.
Another SVC Project, called Nano-World [6] deals with remotely controllable microscopes and video transmission. Their demands to the hardware are not quite the same as ours, but the restricted hardware resources especially and the thereby resulting one user per time slot problem is exactly the same.

3. The Architecture in Detail

The here-proposed and in Figure 1 depicted architecture bases on several components as described in brief:

- Lightweight directory access protocol (LDAP) [7] directory servers for user and module data management as well as for scheduling.
- Security infrastructure for issuing the entire necessary keys.
- A web-learning platform that leads through the courses.
- Portal servers that build the gates to the different locally distributed course modules with third party hardware.

All the possible connections of this architecture are visible on Figure 1. Students and administration personal can access module and scheduling information on the central directory server. Students can connect to the laboratory modules and portal servers to query the directory, to check the student data, to update their module reservation data and to see the module state. An own certification authority issues keys for secure socket layer (SSL) [8] connections and secure shell (SSH) [9] connections as well as for Internet protocol security (IPsec) [10] connections.

Described below are the important parts of the architecture with their functionalities in more detail.

Course Servers
We propose to use a course platform like WebCT that offers many useful functions for student management and especially for creating and automatically rating exercises. Although WebCT offers news boards, chat, student mail, white boards and more it is restricted in terms of designing web pages. As a consequence, the portal servers run their own web servers and provide parts of the course module content directly to the student. The result is that the course platform leads through the entire course, like a red thread, but is supplemented with external content from external sources.

LDAP Servers
This architecture is designed for stimulating collaboration among different educational institutes. Each institute nowadays maintains a database of its own students and many institutes use LDAP directories already. For the others it is easy to export the data in LDAP format.

Therefore we propose to use LDAP directory servers. LDAP has significant advantages to other databases; especially in cases where mostly read accesses occur as LDAP is designed for a fast handling of read accesses. The implementation can be obtained for free from OpenLDAP.

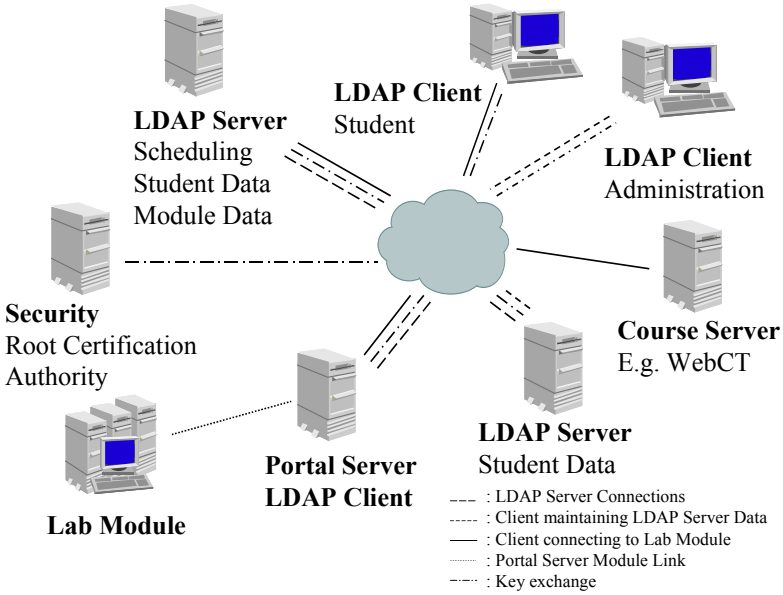

Figure 1: The components of the remote course architecture at a glance

Based on LDAP a rather powerful student management system, with a minimum student account administration work can be outlined. As a consequence of the limited hardware resources that many exercise modules will have (think for example on the prices of a remote controllable microscopy), many exercise modules can only be accessed by a limited number of students simultaneously, according to the available hardware.

To manage the bottleneck that expensive third party hardware in the laboratories can cause, the architecture includes scheduling functions where students can automatically book, change or delete course module reservations on-line on a graphical user interfaces (GUI) in the form of a timetable.

Student and module data are stored on the LDAP directory server, with a scheduling script as an interface between the GUI and the data sets. The scheduling script has the ability to read and write into the LDAP data sets and thereby set access attributes for the modules. Students can query the timetable of the modules, create entries, and change or cancel entries from that table. LDAP clients such as portal servers or administrational staff query the LDAP directory with the respective protocol directly. Student data sets can contain personal information like matriculation number, institute, email, and must contain login and password. The current user of a module is copied in the respective module data set by the scheduling script and read out by the portal servers. Reservation data is kept for each module separately. Only registered students that have previously booked the respective course module are then allowed to proceed to the practical laboratory session. This means that students have to authenticate in order to access the scheduling system, and are authorized by the scheduling system to perform the exercise. During the registered time slot, nobody else can access the reserved hardware. Scheduling functions take place in the LDAP directory and in the portal servers. The portal queries the LDAP directory at regular intervals for the access values of the module that runs on the respective portal. The portal has the ability to announce the remaining time and close the connections to the student. Our LDAP directory structure is shown in Figure 2. We integrated as many as possible of the already broadly used LDAP attribute names in our directory tree.

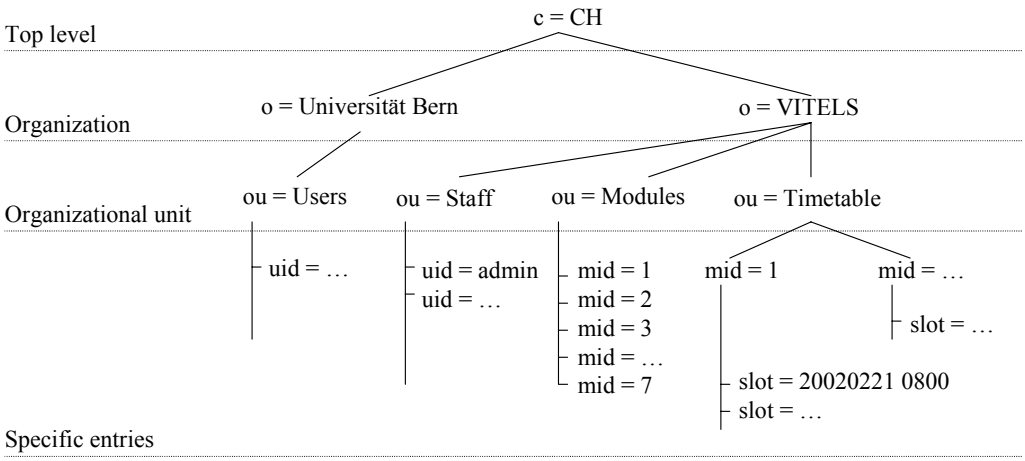

Figure 2: LDAP directory structure

Security
All connections in this architecture, except those behind the portal servers in the laboratory environment, are secured. The own certification authority is used to issue own keys for IPsec connections, for SSH connections, for SSL connections and could be extended to a full public key infrastructure (PKI) in case an institute with increased security needs (for example bank or insurance companies) would like to use smart cards or simply distribute secret keys on a disk.
Students and teachers connect with SSL or with secure shell SSH through the portal to the laboratory equipment. There is no technical obstacle against using Telnet or clear text http connections for connections to the portal servers but as students will transmit passwords and personal data it seems already old-fashioned not using secured connections.

Portal servers on the distributed laboratory sites
The portal server is the entry point to each site and module's laboratory hardware equipment. Portal servers get their authentication and authorization information from LDAP servers and have no student data saved.
Portal servers allow connecting the whole spectrum of available third party hardware, with the only restriction that the hardware must be controllable by a PC. Many devices can be connected by a serial connection, by an internal TCP/IP network or more and more by USB.

Students
Nowadays, it is not unusual that students are international and live distributed over the world. Remote course architectures therefore must obviously scale to this demand. A big issue is that nobody can foresee the hardware and the connection the student will use when attending the course. This architecture is not magic, and if a tutor needs to provide video streams he will encounter the same bandwidth problems as everywhere else too when a student with a 28,8 kbps modem wants to watch a 300 kbps movie. But this architecture is designed that way that students can do anything without installing additional software or having a special operating system for accessing the course servers and the third party hardware network laboratories. The absolutely necessary traffic for course reservation and module access is more or less like traffic from a Telnet session.

4. Prototype Implementation of a Remote Network Laboratory with Third Party Hardware

The VITELS project initiated not only the development of the open course architecture but also the start of the course implementation. For the first prototype implementation we selected the module IPsec from our traditional in house laboratory and adapted it to the needs of a remote laboratory. For a further description of the steps from a traditional laboratory to an on-line laboratory please read Architectural Issues of a

Remote Network Laboratory [11] and Didactical Issues of a Remote Network Laboratory [12].

Below we report from our ongoing implementation work.

Operating system and hardware
All our systems run on hosts with Debian Linux 2.2 [13]. All our hosts and servers are common PCs. The third party laboratory hardware are two Cisco [14] routers, a 2620 and a 3620.

Security
The connections between the portal server and the LDAP server are secured with IPsec tunnels using AH and ESP. The http connections between portals and students are secured with SSL, also between the LDAP server and the students. The portal server acts like a firewall, equipped with one Ethernet interface connected to the Internet and 3 interfaces to the laboratory network repeaters. Like this the Internet is isolated from the laboratory network.

How students connect to the laboratory hardware
Students are directed by the course platform to the web server of the portal server and see a web page with the hardware constellation of the laboratory equipment like showed in Figure 3.

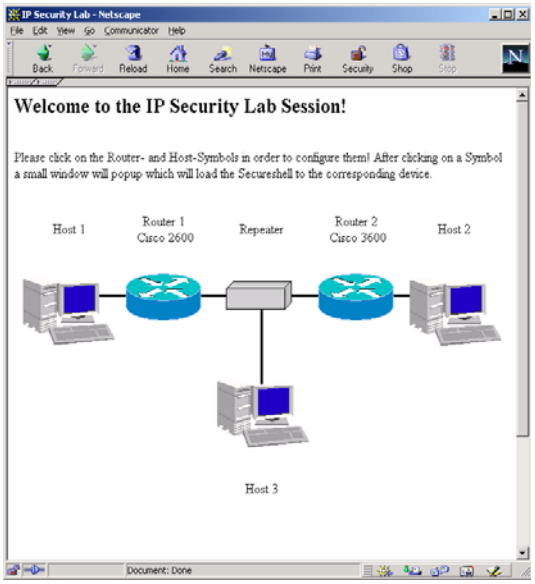

Figure 3: Hardware constellation of the IPsec Laboratory

With clicking on each of the devices a new SSH Java applet from appgate [15], called MindTerm is started, if it is the first time it is downloaded to their computers. The student can then open a SSH connection to the portal server and thereby login to the desired device. For that, he uses the names of the devices and his own LDAP login, as this login is delivered from the LDAP server to the portal server. The hosts are then accessed with rlogin and the routers with Minicom sessions. The student is not aware of this at all, he has his SSH window for each of the connected devices and seems to be connected directly to those devices. Figure 4 shows the SSH session with one of the laboratory routers.

Students use the internal Ethernet to access routers and hosts. As router interfaces can be shutdown or routers can fail, additional connections to the routers were made, using the serial ports and the terminal program Minicom [16]. Over these links, the course software can control and reset the routers.

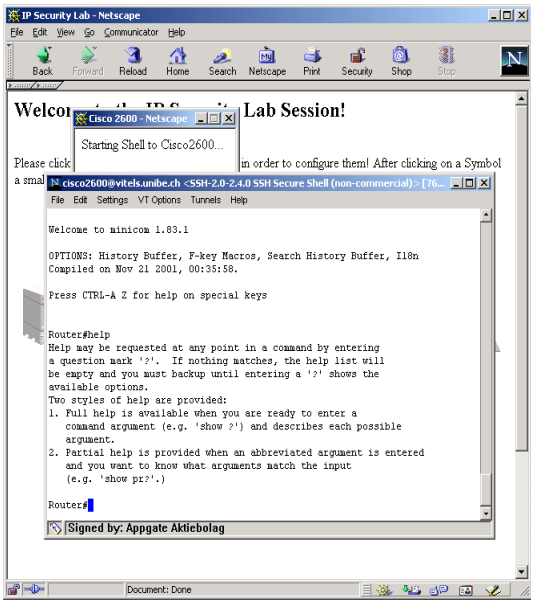

Figure 4: SSH session with a laboratory router

Error Recovery from third party hardware
For the proper function of the exercises, error recovery mechanisms have to be implemented. Students, with or without bad intentions, can crash hardware or lock out themselves from the laboratory infrastructure. A part of the recovery functions is a virtual “emergency button” that is placed on the course pages, allowing a restart with the original settings at any time. After a student has finished an exercise, the hardware

has to be re-initialized and set to the original settings for the next student. Automatic reset should be performed after timeouts.
In our prototype implementation, we log the entire dialog from the students with the routers as these connections use Minicom. At the end of every session or on demand (emergency button), the log is parsed and the passwords are filtered out and used to restart the hardware. This feature is not fully functional yet. A further reset possibility is not yet implemented and will be used in case the filtering fails. The commercial router hardware used in this module allows resetting the passwords during the boot process. This is a difficult procedure, because the router boot dialog must be read out and digits must be sent back to proceed. Currently, we are developing a power interrupter that is controlled over a serial port of the portal server and thereby can shut down the routers.

Automatically rating students
The preparation and post practical exercises are covered with multiple choice and text questions. For the evaluation of the practical session, students have to send back the configuration files of the routers and the dumps of programs like traceroute. Teachers must manually analyze those self-written solutions.
Another approach that we intend to integrate in our architecture is to extract the above described configuration logs and to search for specific configuration phrases. This would allow a much more automated progress control. The data is already logged, with the already described Minicom sessions and also with the history files of the shells.

5 Conclusions

This paper presents a global architecture and partial prototype implementation for enhanced remote courses. The architecture is designed to scale with many situations where remote courses are set up with additional third party hardware and/or where parts of the course content is provided by locally separated content deliverers to students with Internet access, whatever connection or operating system they use. The implementation work has shown that many issues are not easily resolvable but that the inclusion of third party hardware is possible without investing endless amounts of time and money, thereby opening new horizons of remote teaching. The inclusion of third party hardware has at least one significant advantage to computer simulations: in case of technical upgrades it is possibly to update firmware or add new features to the hardware without having to write new simulation software. The gained experiences during work and on meetings indicate that there is a great demand for the proposed architecture.

6. References

[1] Virtual Internet and Telecommunications Laboratory of Switzerland, http://www.vitels.ch
[2] Swiss Virtual Campus, http://www.virtualcampus.ch/
[3] WebCT, http://www.webct.com
[4] Mentor Technologies, vLab Technology, http://www.mentortech.com/vlab/index.shtml
[5] TU Chemnitz, Informations- und Kommunikationssysteme, http://iuk.tu-chemnitz.de/
[6] Virtual Nanoscience Laboratory, http://www.nanoworld.unibas.ch/zope/nano/en
[7] OpenLDAP, http://www.openldap.com
[8] How SSL Works, http://developer.netscape.com/tech/security/ssl/howitworks.html
[9] Secure Shell, http://www.ssh.com/
[10] FreeS/WAN Project, http://freeswan.org/
[11] Steinemann, Zimmerli, Jampen, Braun, Architectural Issues of a Remote Network Laboratory, 2001, submitted and accepted for publication, NL2002
[12] Steinemann, Zimmerli, Jampen, Braun, Didactical Issues of a Remote Network Laboratory, 2001, submitted and accepted for publication, ICNEE 2002
[13] Debian, Linux distributor, www.debian.org
[14] Cisco, Network hardware producer, www.cisco.com
[15] appgate, SSH Java Applet called MindTerm, www.appgate.com
[16] Minicom, Terminal program, http://www.pp.clinet.fi/~walker/minicom.html