Architectural Issues of a Remote Network Laboratory

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. Remote learning is slowly gaining new territories and substituting traditional teaching methods. However, not only a transfer of paper scripts to the Internet is required, but courses for remote learning also need to be developed, offering the possibility to embed practical sessions as well as automated learning control. This paper presents a platform for remote courses in which many locally distributed clients attend exercises on many locally distributed servers. The course platform includes automated student data administration and scheduling for the laboratory modules. Although all connections are secured, students can work with any standard web browser and don't need to install additional software.

1. Introduction

Today's distance learning in the Internet often consists of on-line available scripts, sometimes enhanced with exercises that are not automatically processed and evaluated. Although integrated web-learning environments seem to get more and more accepted, there is still a strong need for advanced concepts and architectures in order to make use of the Internet for distance learning. Several Swiss universities are working together towards a common course in the area of computer communication. The goal is to develop 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 presented in this paper is being performed within the project called Virtual Telecommunications Laboratory of Switzerland [VITELS], which is one of several projects within the Swiss Virtual Campus [SVC] 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. As a consequence, a common adequate architecture had to be developed. This relates to user interface issues as well as to the security architecture. The students must have access to the laboratory equipment such as routers and PCs or workstations over the Internet from any computer in the Internet through any web browser on any operating system. A web-learning environment called [WebCT] is used to lead through the course modules.

2. Related Work

There appear many tries to bring remote laboratory courses on-line. [Mentor Technologies] vLab offers laboratory modules with real network hardware (routers and switches) and scheduling for user access. A big difference to our work is that in their case the users only are geographically distributed but the laboratories are all centralized. Unfortunately, no detailed technical information is available.

An excellent remote course is offered by the [TU Chemnitz]. 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] deals with remotely controllable microscopes and video transmission. Their demands to the hardware are not quite the same as ours, but especially the restricted hardware resources and the therefore resulting one user per time slot problem is exactly the same.

3. Lessons Learned with a Traditional Exercise

Before we started building up remotely accessible exercises, we have offered the laboratory modules based on a traditional approach where small student groups visited the laboratory together. All students have first attended a compulsory introduction course on computer networks.

In order to collect didactical and practical experiences, user feedback forms have been collected after each laboratory exercise. Also, the students have been observed during their laboratory sessions and the preparing homework has been analyzed. First of all, most of the students showed a rather strong motivation for the practical exercises. Many of them liked to get in touch with real hardware. This is probably hard to achieve with purely distance learning exercises. Therefore, laboratory visits should be organized at the beginning of a distance-learning course, giving the students a feeling about the hardware they will work with.

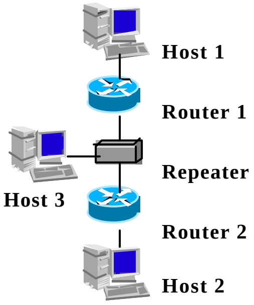

Figure 1: VPN exercise in our traditional laboratory

Figure 1 shows the IP Security exercise of our traditional laboratory, which mainly consists of two parts. In the first part students learn to configure two routers in order to set up a secured VPN tunnel between the routers. In the second part, students have to perform measuring tasks like traffic analysis on the VPN link, using a bandwidth measurement (network capacity) tool called [NetPipe] with and without encrypted traffic and analyzing traffic with [Tcpdump] to verify whether the exchanged traffic is really encrypted.

All our course development work at the university of Bern is done with this module, developing it from a traditional laboratory network to a remotely accessible test network.

Preparation work is urgently needed for the students to successfully perform an exercise. We have discovered that several students did not prepare themselves sufficiently prior to the practical work. Those in particular had significant problems to solve the exercise tasks within a reasonable time. In addition, they had to ask many questions to the tutors. In a remote lab, good preparation becomes even more important to successfully pass the exercises. Also, a well-structured approach to solve the tasks seems to become more important.

However, the available on-line material has been a problem in some cases since still a lot of good information is only available in books. Copyright issues prohibit distributing digitized copies among the students.

4. Architecture

4.1. Distributed Laboratory Sites

Each module can be accessed via a portal and each participating site maintains one portal server per module. Portal servers don't contain any student data but they are connected to the central [LDAP] directory server, where they get authentication and authorization information. Students and teachers connect with secure socket layer [SSL] through the portal to the laboratory equipment. The architecture allows connecting all kind of lab modules to the portals.

Figure 2 shows the central LDAP directory server and one portal server. On the right the laboratory equipment is shown as a black box. Two clients, typically students, connecting through the Internet to the module, are depicted in the upper left corner.

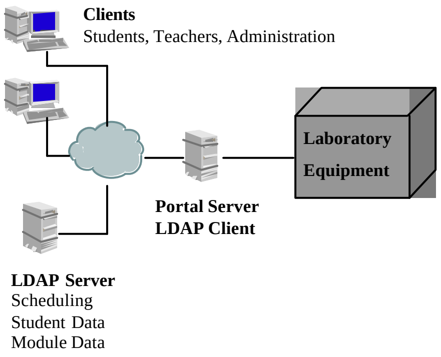

Figure 2: Module Architecture

We are developing a rather powerful student management system. The student account administration work must be kept on a minimum here. As a consequence of the limited hardware resources, exercise modules can only be accessed by a limited number of students simultaneously, according to the available hardware.

Scheduling helps to manage those problems. State of the art is an automated booking system with a timetable where students can book, change or delete module reservations on-line on a graphical user interfaces (GUI).

Our technical approach is to have student and module data on the LDAP directory server, with a scheduling script as an interface between the GUI and the data sets. The scheduling script is still under development and has the ability to read and write into the LDAP data sets and thereby set access attributes for the modules. As Figure 3 indicates, a student 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.

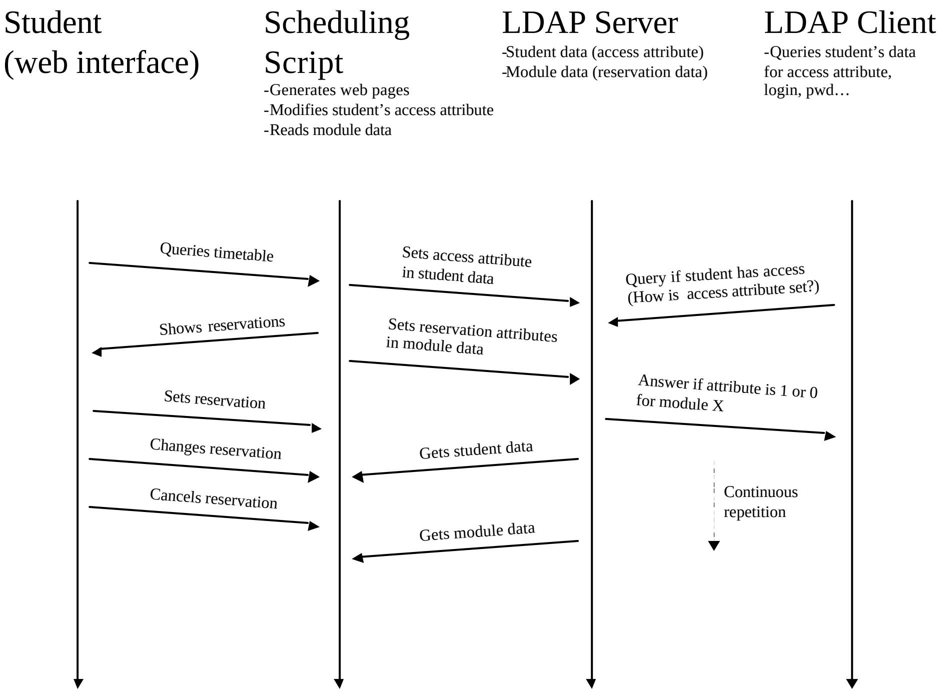

Figure 3: Booking interactions

Student data sets contain personal information like login, password, matriculation number, institute, email, and a very important attribute that contains module access flags. The access flags are set by the scheduling script and contain the number of the modules booked and actually available for the respective students.

Only registered students are then allowed to proceed to the practical session of a module. 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 are realized in the LDAP directory and in the portal. 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 4. We integrated as many as possible of the already broadly used LDAP attribute names in our directory tree.

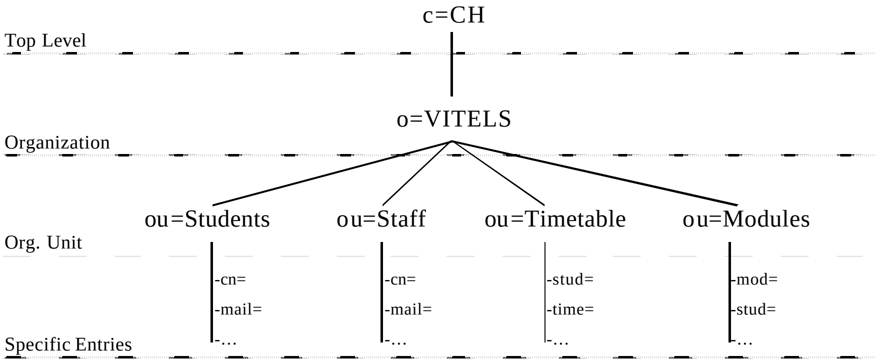

Figure 4: LDAP directory structure

4.2. Distributed Clients and Distributed Laboratory Sites

There are many issues to be solved for building such a course in an open and distributed environment. One important issue and at the same time a main obstacle to overcome is the distributed course infrastructure with several modules placed in five geographically distributed locations all over Switzerland. Each partner should be able to create the content of its own course modules rather independently. Other important issues are user administration and security. The courses are open to pre-registered students only. Since users from several administratively independent institutes need to access the entire course, we decided to build a centralized database containing authentication and authorization data in the first phase of the project. In a later phase we also intend to include accounting. On each site, an entry point called gateway or portal checks the accessing persons based on the authentication result it receives from the central LDAP directory as shown in Figure 5. There are two LDAP servers shown because LDAP and our architecture allow maintaining distributed directories, which can be an advantage if already existing LDAP directories at partner sites should be integrated in the remote course structure.

Figure 5 shows the possible connections in our architecture. Students and administration personal can access module and scheduling information on the central directory server. Students can connect to the lab modules and portal servers to query the directory, to check the student data and the module state.

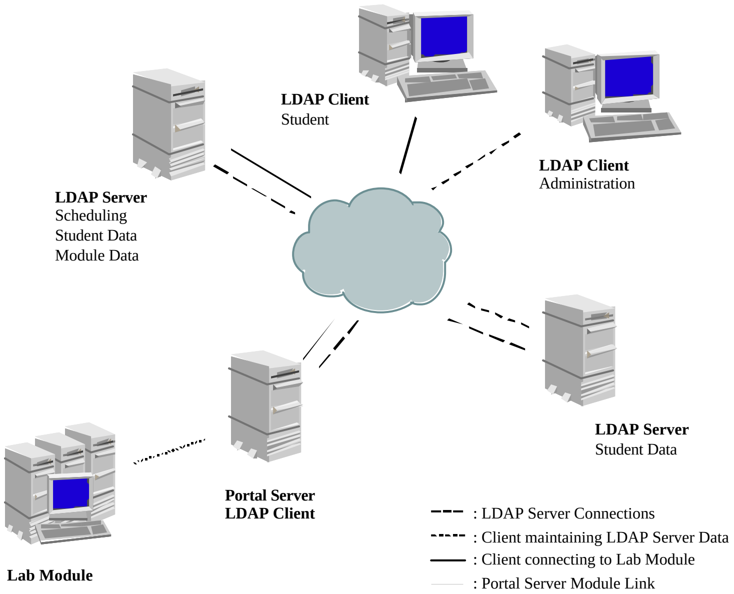

Figure 5: Global Architecture for the VITELS course

5. Implementation Issues

5.1. Security

The whole architecture for distance learning must include authentication, authorization and accounting functions. As the proposed architecture could be used for future courses from other organizations with probably higher security needs, links between the LDAP directory servers and the portal servers are secured with [IPsec] protocols. Students and administration staff use SSL connections for data exchange with the booking and portal servers, because it is widespread and easy to use.

In this context, it is important that clients do not need to install additional security software to be able to use the remote network laboratory.

5.2. Error Recovery

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. 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. A further reset possibility is not yet implemented and will be used in case the filtering fails. For example Cisco hardware 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.

5.3. Rating Students

A remote course structure that lacks control mechanisms for the student work is not very useful for tutors and students. WebCT offers all sorts of multiple-choice questions, yes or no questions and additionally discussion boards, white boards, and chat rooms.

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.

6. Example Exercise Module Implementation

6.1. IP Security Exercise Experimentation Network

For the first prototype implementation of the VITELS course, we took the IP Security (Ipsec) module from our traditional in house laboratory and adapted it to the needs of a remote laboratory as shown in Figure 6. The isolated laboratory network was connected to a portal server and additional connections were made with the routers over serial links. The portal server of this prototype implementation manages the routers and the hosts. It does not store any student data; all the needed student data is retrieved from the LDAP directory as described above. The hosts only allow students to use network-testing tools. We are planning an additional implementation of network configuration and set-up possibilities for the hosts.

The IP Security module consists mainly of two parts. In the first part students learn to configure two routers in order to set up a secured VPN tunnel between the routers. In the second part, students have to perform measuring tasks like traffic analysis on the VPN link, using a bandwidth measurement (network capacity) tool called [NetPipe] with and without encrypted traffic and analyzing traffic with [Tcpdump] to verify whether the exchanged traffic is really encrypted.

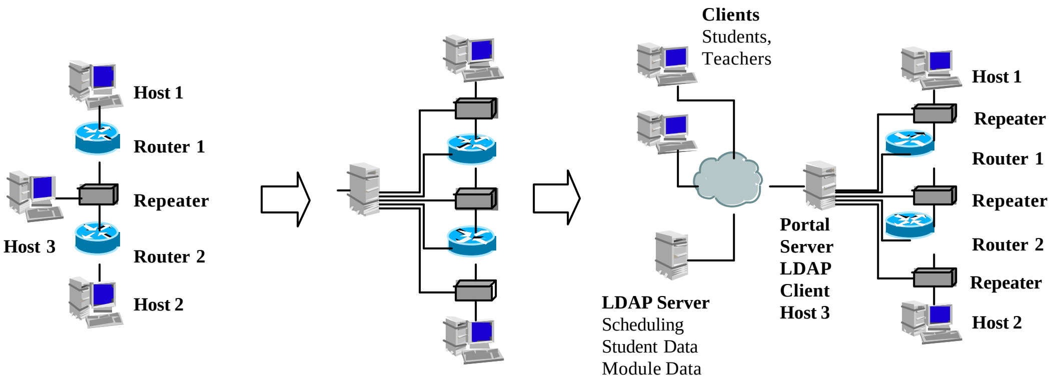

Figure 6: Transition from a traditional laboratory to a remote laboratory

7. Conclusions

This paper presents an architecture for implementing a remote computer network laboratory. The course consists of various modules that are being developed and maintained in a distributed environment. A strong focus is required to solve issues such as security, scheduling and error recovery. In parallel, first didactical experiences have been gained with a traditional laboratory that will be valuable for structuring the remote course modules.

8. References

[Ipsec] FreeS/WAN Project, http://freeswan.org/

[LDAP] OpenLDAP, http://www.openldap.com

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

[Nano-World] Virtual Nanoscience Laboratory, http://www.nanoworld.unibas.ch/zope/nano/en

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

[SSL] How SSL Works, http://developer.netscape.com/tech/security/ssl/howitworks.html

[SVC] Swiss Virtual Campus, http://www.virtualcampus.ch/

[Tcpdump] TCPDUMP public repository, www.tcpdump.org

[TU Chemnitz] Informations- und Kommunikationssysteme, http://iuk.tu-chemnitz.de/

[WebCT] WebCT, http://www.webct.com

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