The Virtual Internet and Telecommunications Laboratory of Switzerland

Torsten Braun and Marc-Alain Steinemann
Institute of Computer Science and Applied Mathematics, University of Bern
Neubrückstrasse 10, CH-3012 Bern, Switzerland
[braun|steine]@iam.unibe.ch, www.iam.unibe.ch/~rvs

Introduction

The Virtual Internet and Telecommunications Laboratory of Switzerland (VITELS, www.vitels.ch) is a course providing practical hands-on exercises for computer science students with the goal to strengthen and extend the knowledge in computer networking. The course is built on different modules that have been developed and implemented by different course designers and developers at several universities. The course is based on a common didactical approach, a common graphical interface and a common implementation architecture. The resulting requirements and guidelines must be fulfilled by each module in order to provide a homogeneous course to the students. The distributed approach has the advantage that equipment and maintenance costs can be shared by the collaborating universities (Universities of Bern, Geneva, Fribourg, and Neuchatel, Engineering School Fribourg) and their costs for providing such a course can significantly be reduced compared to traditional network laboratory courses. In particular, human and technical resources can be used much more efficiently. The modular approach allows using the course contents in many different ways, e.g. as a stand-alone laboratory course or as a practical part integrated into a traditional lecture. Teachers are able to compose an individual set of modules that are targeted to their preferences and the curriculum needs at their respective university.

Course Prerequisites

Usually, we expect that a student participating in the course has already attended a traditional lecture in computer networking of approximately 40 hours. At University of Bern, the traditional lecture includes topics such as communication architectures, transmission media, physical layers, medium access layers, data link protocols, circuit switching, packet switching, routing protocols, transport protocols, Internet applications, network management, and network security. Each of the topics mentioned above can be considered as atomic units and are each being covered in a single two hours lecture, enhanced by accompanying theoretical exercises. This enables students to study such a topic within a few days completely.

Course Contents

The practical course modules focus on very particular subtopics. The modules have been chosen such that the various layers are being covered uniformly. In addition, the students shall get in touch with different types of exercises and with various tools. Currently, the following modules are either available or in preparation: A module on ***IP network simulation*** focuses on Internet routing issues. Students have to configure network interfaces and routes in both end systems and routers. Both types of network nodes are being emulated, which allows performing experiments with larger network topologies. In contrast to that, students work with real router equipment in the ***IP security*** module, where they have to establish security associations (authentication and encryption) between two routers. The module on ***IP network performance evaluation*** introduces performance metrics and measurement / diagnosis tools such as traceroute or tcpdump in real IP networks. In the ***protocol analysis*** module, Ethernet packets are captured in real-time and various protocols on different layers can be visualized, for example to study packet loss effects on transport layer. The ***firewall management*** module allows configuring filtering policies in a commercially available firewall. ***Socket programming*** and distributed applications using remote procedure calls have to be developed by the students. Another course module focuses on ***remote method invocation*** as this

technology is becoming increasingly important in distributed Java based applications. Another module on application level is focusing on the ***hypertext transfer protocol***. Finally, a module has been developed that allows students to remotely install and configure ***network services*** such as email, http, and DNS on Linux systems.

Implementation Architecture

The course modules are using a common implementation architecture. Each university implements its own experimental laboratory networks and connects them via a portal system to the Internet. The portal acts as an application level gateway / firewall that protects unauthorized access to the distributed laboratory infrastructure. Portal systems are currently connected to a central directory which contains user authentication data. The directory is also being used for scheduling mechanisms. Students can reserve time slots to get exclusive access to the distributed laboratory equipment. Access to the directory is provided via Lightweight Directory Access Protocol (LDAP). Moreover, recovery mechanisms have been implemented that allow students performing remote experiments, to reset the laboratory hardware in case of failures. A key feature of the system is that students can work on laboratory equipment as if they would sit in front of the device console. The only requirement for students is to have access to a standard web. Another key component is the commercial web-learning platform for provisioning web content and for course management functions such as student management, automated rating of exercises, discussion forums, chat etc.

Didactical Approach

Teaching on the Internet requires a well elaborated concept, for the course's pedagogy but as well for the user interfaces and the graphical layout. The result is a homogeneous learning environment where all course modules use exactly the same didactical structure and graphical layout. Two crucial factors of the course are the strong constructivist approach on one side and the claim to produce an easy navigational and understandable course on the other side. Therefore, a didactical and graphical design guideline document has been developed in a collaboration of technical and didactical experts.. Special graphical design tools facilitate the fast generation of uniform web pages. The course starts with a common introduction chapter explaining the VITELS course structure, the teaching approach and especially all the components that reappear in all the different modules. All modules have exactly the same required structure and consist of four chapters:

1. An introduction chapter with a short abstract to the module content and relevant study information such as the module's study goals.
2. A theory chapter with information that might not be covered in previous computer science lectures.
3. A knowledge application and exploration chapter with the hands-on sessions on real laboratory devices or emulation software. This is the main section, where students use the trial-and-error method to configure devices or program applications. Students have to write a logbook that is used to write down notes, calculations, questions and more.
4. An evaluation of acquired knowledge and skills chapter where the achieved knowledge is tested and graded.

Experiences

About fifty students attended already the course in its current form. Students appreciated the clear concept behind the VITELS course and its uniform appearance. Especially e-learning demands from course designers to follow clearly defined didactical concepts and structures when designing a course. Although students have been initially concerned about some new didactical elements such as writing synthesis essays or formulating own learning goals, the overall feedback and learning impact was very positive. An important remaining issue is that students should perform at least one or two modules in a real laboratory to get in touch with real laboratory devices.