Implementation of a Distance Learning Module Based on Emulated Routers

Florian Baumgartner, Torsten Braun, Eveline Kurt,
Marc Steinemann, Attila Weyland

Forschungsgruppe “Rechnernetze und Verteilte Systeme”
Institut für Informatik und Angewandte Mathematik, Universität Bern
Neubrückstrasse 10, CH-3012 Bern
Phone: +41 31 631 8957, Fax: +41 31 631 3261
[baumgart|braun|kurt|weyland|steine]@iam.unibe.ch

Abstract. This paper presents the implementation of a tool aiming to allow students to perform virtual experiments within a computer network course. Students can create an arbitrary virtual IP network consisting of emulated routers and experiment with their configuration. The web-based user interface allows students to interact remotely with the emulated routers, but simultaneously it is very similar to commonly available configuration interfaces of network devices in reality. This enables students to configure routers like in the real world but also to experiment in a much more robust and safe environment.

1 Introduction

The Swiss Virtual Campus [1] is supporting a series of projects in order to develop learning material for distance learning over the Internet. Within the project Virtual Internet and Telecommunications Laboratory of Switzerland [2], a set of modules is currently being developed and tested that allow students to perform practical exercises remotely from any Internet workstation instead of being present in a laboratory room.

The different modules being developed by the various project partners can be classified as remote and virtual exercises. In the case of remote exercises, students work with real devices that are located in a university’s laboratory room. Students control and configure the behaviour of the devices using web technologies from any workstation connected to the Internet. An example for such a module is the IP Security module [3], where students have to manage IP Security tunnels between end systems across a network infrastructure consisting of Ethernet repeaters and IP routers. Students performing this exercise module need a certain knowledge and experience level since potential mistakes during network device configuration can cause significant error states. These might cause that the devices will not be accessible over the Internet or in the worst case need to be reset manually. Therefore, there is a need for students with a lower knowledge level to gain the experiences they need for performing such advanced modules in a more smooth way.

For this reason, a second class of modules is required, which are called virtual modules. In the case of virtual modules, the experimentation environment is not existing in reality, but it is simulated. This allows to offer a much more safe and robust experimentation environment. Students can make errors without the need to manually reset any devices. However, the simulated environment should match reality as close as possible in order to prepare the students for later remote experiments.

In this paper we present the implementation of a virtual exercise module that shall be used by students to make first experiences with IP router configuration. In particular, the students shall learn how to set up network interfaces and static routing tables. Router devices are emulated by software entities called virtual routers. We will describe related work in Section 2 and the implementation architecture of virtual routers in Section 3. For the creation of larger networks out of the virtual routers, we have developed a web-based user interface that is described in Section 4. Section 5 will conclude the paper.

2 Related Work

We propose to use emulated routers for distance learning and allowing students to perform computer network experiments remotely and within a safe environment. Emulation of routers usually requires one computer for emulating a single router [4][5]. In contrast to that, multiple virtual routers can emulate multiple routers on a single computer. Like the virtual routers, the network simulator ns [6] supports to interconnect real networks to the simulated environment. However, due to time-consuming simulation processing, the number of simulated routers is rather limited if experiments shall be performed in real-time. The ns implementation is also rather monolithic and therefore not very flexible. In addition, the ns user interface is not appropriate for a course in that students shall learn how to configure routers.

The main goal of laboratory experiments with emulated routers is to provide a safe environment, where students can prepare themselves for later experiments that are performed remotely with real network devices [3]. Remote experiments are very popular in various areas such as nano-science [7], engineering [8][9], computer networks [10] and others.

3 Virtual Routers

3.1 Virtual Router Networks

Virtual routers are small entities (Unix user space processes) that are emulating single IP routers [11]. Links that are normally used to connect real routers are replaced by communication channels between these entities in order to create larger networks. Each virtual router runs as an independent process not interfering with other virtual routers and only exchanging packets by communication channels.

Figure 1 illustrates the interconnection of several virtual routers (VR) that are distributed to three different computers. The type of the communication channels between the virtual routers depend on whether they run on the same or on different computers. Virtual routers running on the same computer are interconnected by inter-process communication channels, while virtual routers running on different computers are interconnected by channels based on UDP tunnels. Up to 64 virtual routers have been created on a single computer for various research experiments. Virtual routers significantly reduce the required amount of resources for network emulation. The performance evaluation and other technical details is beyond the scope of this paper and is described in [12] and [13].

Virtual routers shall behave like real routers and therefore have to process traffic in real time. Like normal routers they have to receive packets, process, and forward them. Virtual routers can also be connected to real networks. This allows to route traffic generated by real end systems over a network consisting of both real and emulated subnetworks. For the interconnection of virtual routers to real networks, so-called softlink devices (Figure 2) have been implemented. A softlink device behaves like other network devices within a computer, e.g. an Ethernet device. However, packets transmitted over a softlink device are not transmitted over a real network but to a virtual router that is connected to the softlink device via file system IO. In the other direction, packets from a virtual router can be sent to the IP stack on top of the softlink device. If in scenario depicted in Figure 2 a web server transmits a packet to a remote web client (web browser), the packet is sent via the socket, TCP/IP stack, and the local softlink device (sol0). Virtual router 1 receives the packet and forwards it via an UDP tunnel to virtual router 2. Virtual router 3 receives it via an inter-process communication channel from virtual router 2 and forwards the packet to virtual router 4, again via an UDP tunnel. Virtual router 4 delivers the packet to the softlink device at the receiving end system and the web client receives the packet from the socket it is connected to.

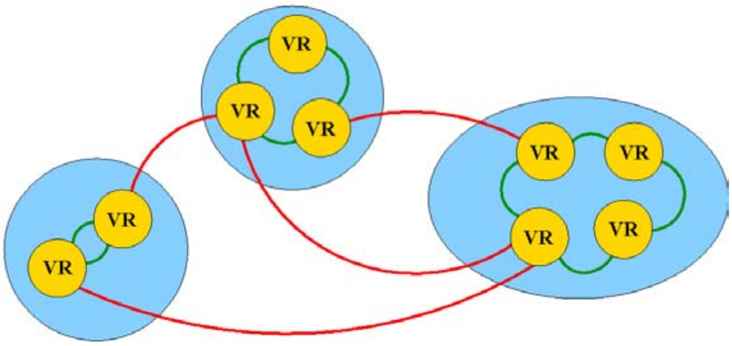

Figure 1: Network of Virtual Routers

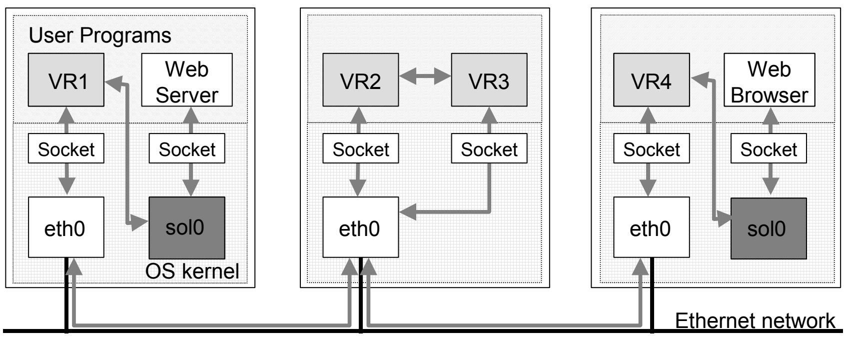

Figure 2: Softlink Device

3.2 Implementation Architecture

Figure 3 shows the implementation architecture of a virtual router. The lower part consists of the core components required for packet forwarding and routing. IP packets are received and transmitted via interfaces (dashed boxes). Interfaces consist of several configurable subcomponents such as network address translators (NATs), queuing systems (e.g., DiffServ traffic conditioners, token bucket filters, schedulers etc.), rate limiters, and interconnection handlers. Interconnection handlers interconnect a virtual router with other virtual routers or softlink devices. The packets received from an interface are processed by a programmable filter and the forwarding unit. Packets might be forwarded to other interfaces or to higher-layer components / protocols. A virtual router can be extended by dynamically loadable objects such as an active router extension (Python-based active router, pybar), a graphical user interface (command line interface), a traffic monitoring component (dump), and virtual network diagnosis utilities (ping, traceroute).

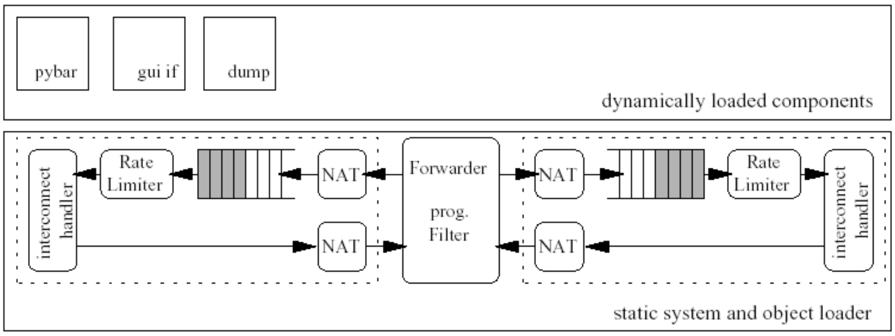

Figure 3: Virtual Router Implementation Architecture

3.3 Application Programming Interface (API)

Virtual routers provide a high degree of flexibility: interface components can be created and modified dynamically, higher-layer objects can be loaded dynamically, filters are programmable, and routing tables can be adapted. Virtual routers support different configuration interfaces by API channels. The configuration program, e.g. a graphical user interface, has to establish an API channel to the virtual router and can exchange configuration messages over this duplex channel. The communication is based on virtual router control blocks (VCRBs) and virtual router response blocks (VRRBs). A virtual router receiving a configuration command within a VCRB, parses the control block, executes the configuration command, and returns the configuration result via a VRRB. Note that the configuration program and the virtual router that communicate via an API channel can run on different computers. Currently implemented commands support adding and deleting virtual router interfaces, retrieving interface information, changing interface characteristics and queuing systems. In addition routing table entries, loadable objects, filters, and protocol stacks can be added, deleted and read. Also, IP packets can be delivered for further processing to the virtual router.

4 Web Interface for the Configuration of Virtual Networks

Virtual routers offer a platform for rapid development, prototyping and testing of new communication subsystems but also can serve as a platform for distance learning. Virtual routers not only help to keep the costs for building large experimentation networks very low but also offer a robust environment for performing network device configuration exercises. In order to be able to use virtual routers for web-based distance learning courses, it was required to extend virtual routers by an appropriate web-based user interface [14]. The goal was to offer students an environment in which they could create or import their own network topologies, perform the required interface and routing table configurations in order to get the network running, and to perform tests whether the router configurations have been correct.

In this Section, we will first describe the user interface to be used for performing configuration and evaluation experiments with virtual routers. Finally, we will describe the implementation architecture of the required components.

4.1 User Interface

While a student uses the virtual routers for her configuration experiments, she will visit four web pages. First, the student has to login and will then be directed to the create/change network applet (Figure 4). If she has created a network before, the network configuration will be loaded and displayed. The student can now modify an existing network or create a new one. The applet allows her to create or delete new routers, to interconnect the interfaces of two routers, and to delete connections between two routers. By creating and deleting connections between the virtual routers, virtual router interfaces are implicitly created and removed. This step basically is equivalent to installing wires between two real routers. After the network has been designed, the student can allocate the required resources (virtual routers, communication channels). This causes the applet to ask for the creation of the network consisting of virtual routers on the server computer. “Quit” causes to release the allocated resources.

After allocating the resources, the student switches to the select applet (Figure 5). By clicking on the corresponding router symbol, she can switch to the configuration applet that allows her to enter configuration commands for the selected (virtual) router. Otherwise, she has also the choice to return to the create/change network applet in order to change the network configuration again.

The configuration applet (Figure 6) allows the student to enter router configuration commands using an interface very similar to a command line interface of commercially available routers. We have chosen a command line interface intentionally, since such an interface is more common and usually more powerful, e.g. providing a browsable history of executed commands and the corresponding replies. By clearly separating the command line input from the command line history we aim to make the interface more convenient to use. The command language is oriented to UNIX commands. Figure 6 shows an entered sequence of ifconfig and route commands with the replies from the virtual router.

The student mainly will use ifconfig or route commands to setup the interfaces of the router (e.g., assigning IP addresses and network masks) and to configure the static routing table of each router. After the configuration process has been finished, she might want to test whether her network configuration is correct. This can be done using traceroute and ping commands.

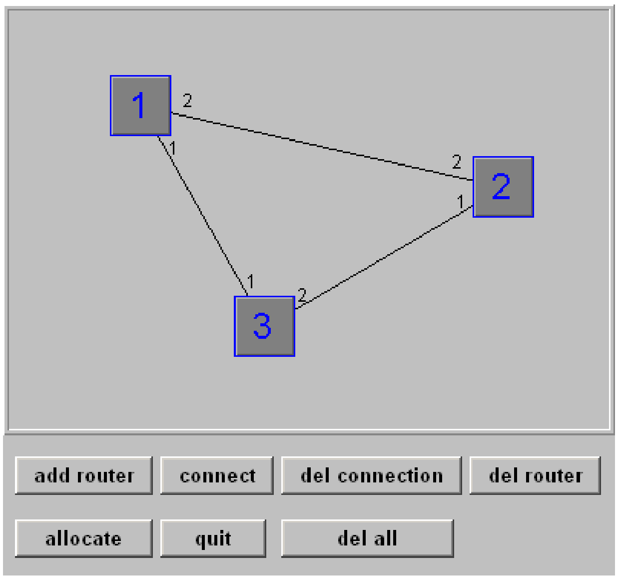

Figure 4: Create/Change Network Applet

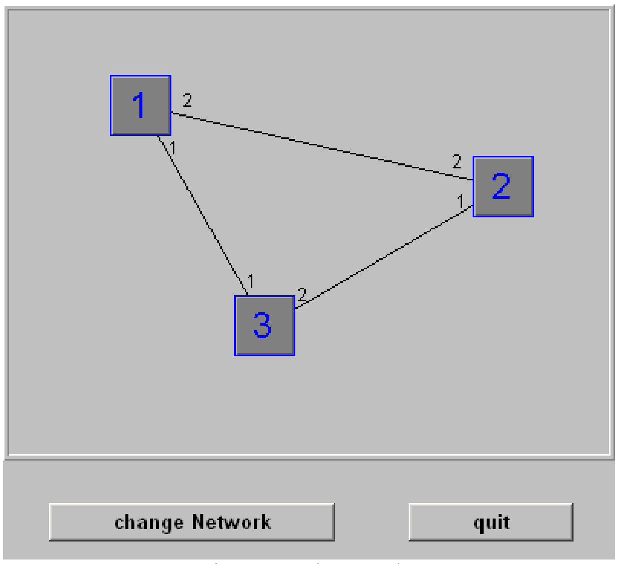

Figure 5: Select Applet

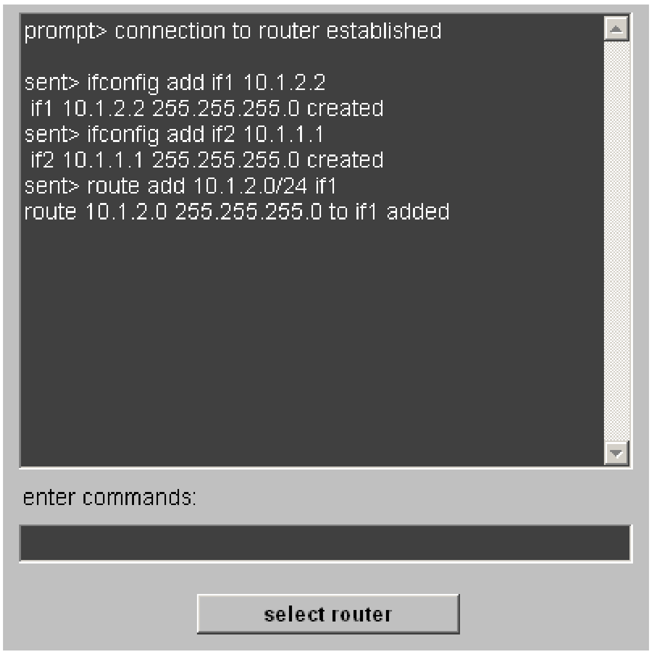

Figure 6: Configuration Applet

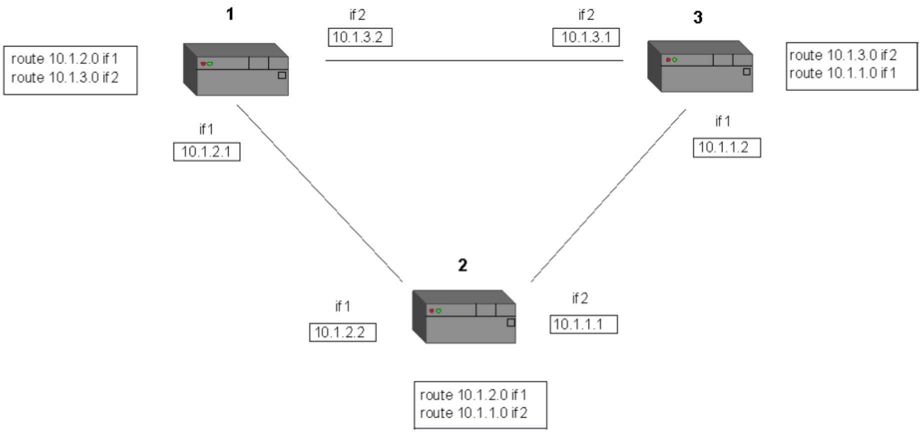

Figure 7: Network Example

4.2 Configuration Example

Given the simple IP network of Figure 7, the student has first to design the network topology (three routers and their interconnections) in the create/change network applet. Then she has to trigger the allocation of the required network resources. After that she can configure the interfaces and the routing tables of the virtual routers in the configuration applet. For example, for router 1 the student has to enter the following command sequence:

```
ifconfig add if1 10.1.2.1 255.255.255.0
ifconfig add if2 10.1.3.2 255.255.255.0
route add 10.1.2.0/24 if1
route add 10.2.3.0/24 if2
```

4.3 Implementation Architecture

This section describes the implementation architecture of the web-based virtual router configuration interface. Figure 8 shows two computers: a client where the student has launched a web browser and a server including a web server, a Java program called administrator, and the virtual routers (microvar). The student navigates through dynamic web pages generated using PHP and downloads/executes Java applets that are embedded in the web pages. As described in the previous section, the student switches between applets. These applets share common data such as the network configuration. Due to Java security restrictions, the data can not be saved on the client computer. Therefore, they are transmitted to the administrator program which stores the data and provides it for subsequent applets.

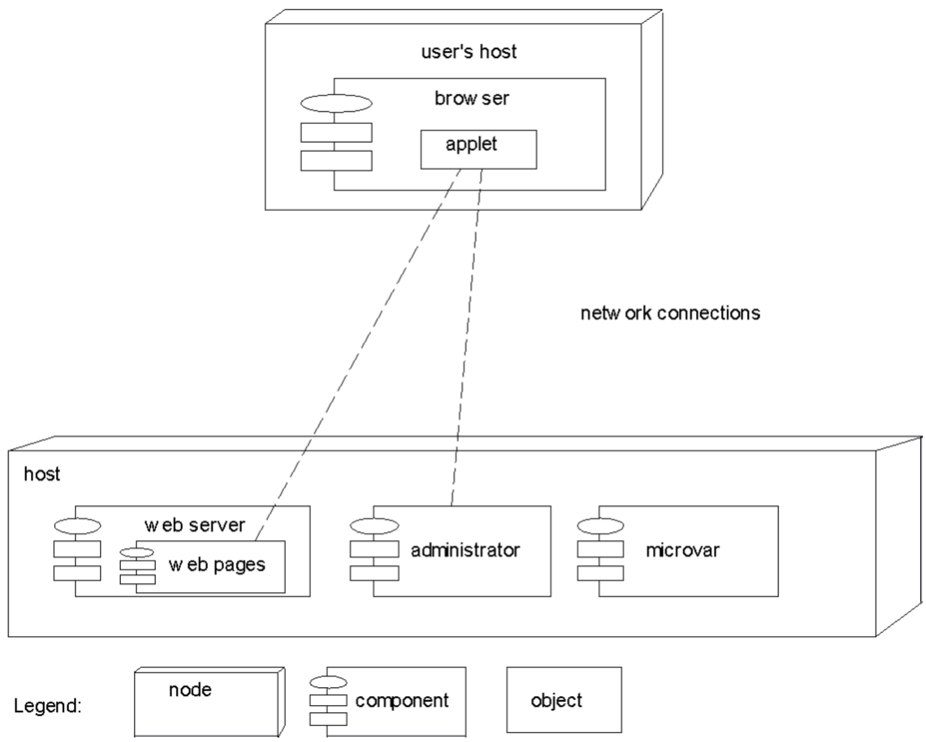

Figure 8: Implementation Architecture

The Java applets on the client then open TCP connections to the administrator program (written in Java) running on the server and exchange request / response messages. The most important commands are for saving data to a file at the server, allocation of virtual router resources, retrieval of virtual network topology data, and closing a session.

The administrator program is the interface between the applet running on the client and the virtual routers. It shares common data with the component for dynamic web page creation. The administrator program receives commands from the applet and translates it into appropriate configuration API calls to the designated virtual router. However, there is no 1:1 relationship between commands issued by the student and commands sent to virtual routers. As depicted in Figure 9, a command from the applet triggers a sequence of virtual router API commands. Figure 9 shows the example of deleting a virtual router interface via the configuration applet. After receiving this user command, the administrator program causes the virtual router to disconnect the specified interface and then to delete it.

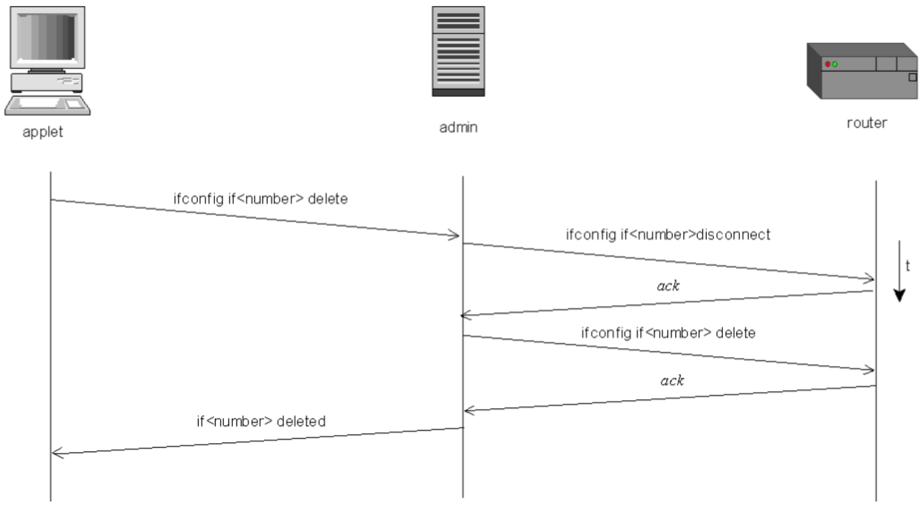

Figure 9: Message Exchange between Client, Server, and Virtual Routers

5 Summary and Conclusions

This paper described the implementation of a course module for a practical exercise course in the area of computer networks. The module is based on emulating IP networks and allows to configure emulated routers via a web-based command line interface. Therefore, the student desiring to perform the exercise only needs a web browser on her client computer. With these minimal requirements, the course module can be easily accessed by all students and can be deployed with low costs.

The course module will be tested within a regular computer network laboratory course in the near future and be improved based on the student's feedback. Although we explained how students can configure router interfaces and routing tables, the virtual routers are much more powerful and can be used for more advanced student exercises. It would also be possible to let students configure more advanced networking functions such as traffic shaping, address translation, and IP-in-IP encapsulation. Also, the behaviour of TCP and Internet applications could be investigated in more detail.

The virtual routers as the basic underlying components have already been successfully used for research purposes, in particular for research projects in the area of Quality-of-Service management and monitoring as well as active networking. Future extensions will support multicast extensions and dynamic routing protocols. Also it might be interesting to create laboratory exercises consisting of mixed networks, i.e. networks including real and emulated routers. However, the main goal of laboratory experiments with emulated routers is to provide a safe environment, where students can prepare for later experiments with real network devices.

6 Acknowledgements

The work described in this paper has been partially supported by the Swiss National Science Foundation project 2100-055789.98.

7 References

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

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

[3] M. Steinemann, T. Jampen, S. Zimmerli, T. Braun: Architectural Issues of a Remote Network Laboratory, Networked Learning 2002 (NL 2002), Berlin, May 1-4, 2002

[4] B. White, J. Lepreau, S. Guruprasad: Lowering the Barrier to Wireless and Mobile Experimentation, First Workshop on Hot Topics in Networks (HotNets-I), 28-29 October 2002, Princeton, New Jersey, USA

[5] Yongguang Zhang and Wei Li: An Integrated Environment for Testing Mobile Ad-Hoc Networks, Third ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc 2002, Lausanne, July 9-11, 2002

[6] Network Simulator, URL: http://www.isi.edu/nsnam/ns/

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

[8] D. Bühler, W. Küchlin, G. Gruhler, G. Nusser: The Virtual Automation Lab - Web-based Teaching of Automation Engineering Concepts, 7th Annual IEEE International Conference on the Engineering of Computer Based Systems, Edinburgh, April 2000

[9] A. Böhne, N. Faltin, B. Wagner: Self-directed Learning and Tutorial Assistance in a Remote Laboratory, Interactive Computer Aided Learning Conference, September 25-27, 2002, Villach, Austria

[10] R. Sontag: Berufsbegleitend lernen: Informations- und Kommunikationssysteme, it+ti: Informationstechnik und Technische Informatik, Oldenbourg Verlag, 3/2001, pp. 167

[11] F. Baumgartner, T. Braun: Virtual Routers: A Novel Approach for QoS Performance Evaluation, QofIS'2000, September 25-26, 2000, Berlin, Germany

[12] F. Baumgartner: Quality-of-Service Support by Active Networks, Ph.D. Dissertation, Universität Bern, February 2002

[13] F. Baumgartner, T. Braun, B. Bhargava: Virtual Routers: A Tool for Emulating IP Routers, 27th Annual IEEE Conference on Local Computer Networks, Tampa, November 6-8, 2002

[14] E. Kurt: Implementation of a Web-Based Interface for Virtual Router Configuration, Diploma Thesis, Universität Bern, May 2002