HyperLab: Towards Building A Hybridized And Adaptive Remote Computer Networking Laboratory

Yongning Tang
Illinois State University
Normal IL, USA
ytang@ilstu.edu

Tom Draper
Illinois State University
Normal IL, USA
tedrape@ilstu.edu

Zhiwei Xu
University of Michigan-Dearborn
Dearborn MI, USA
zwxu@umich.edu

Abstract

Hands-on learning environment is a highly desirable component in computer networking curriculum, which also brings various educational challenges, such as rising costs, inabilities to adapt to distance learning, a lack of security, and inflexibility in meeting with the various course requirements. In this paper, we present an innovative approach called HyperLab to build a hybridized and adaptive remote computer networking laboratory. HyperLab can seamlessly integrate virtual networks with physical networks to achieve significant flexibility and effectively control the cost. Moreover, HyperLab can easily adapt to the various and even changing requirements of networking courses via its open and modularized infrastructure. This paper presents the design and implementation of HyperLab, as well as our experience in adopting HyperLab into computer networking curriculum.

I. Introduction

Hands-on learning environment is essential for students to reinforce concepts learnt in class and obtain concert understanding of details. Hands-on learning environment also keeps courses closely tied to reality, and makes students more competitive in their future careers. There is growing demand from students for obtaining hands-on experience while taking computer networking classes. However, in order to incorporate hands-on learning component into computer networking curriculum, the following challenges need be addressed:

- Cost: In order to provide technical instruction, appropriate technical equipment (e.g., routers with various network modules, switches, access points, servers, racks) is required. Moreover, several networking courses may need access these equipment at the same period. Resource conflicts and inefficient utilization often occur. Thus, a cost effective hands-on learning environment is desirable but challenging in computer networking education.
- Inabilities to distance learning: Traditionally, hands-on exercises require students to physically come to computer networking labs. While implementing a laboratory that can only be used via physical access would be less difficult, it would also greatly diminish its value and availability. This is especially true in an academic environment where students may register for distance learning courses that require a laboratory component.
- Security: A network laboratory are often used by students to mimic a production environment, which may create a big concern to network management personnel. Properly separating experimental networks from production networks is mandatory. Additionally, user authentication and resource management also require different security policies.
- Adaptability: With different learning objectives (e.g., routing and switching, VoIP, wireless) and the level of learners (e.g., undergraduate, graduate, or community trainees), the lab should be highly flexible and scalable to adapt to the corresponding requirements.

In this paper, we present a hybridized and adaptive remote computer networking laboratory system called HyperLab to tackle the above challenges. HyperLab adopts a modularized approach to achieve high flexibility and scalability, which consists of three functional modules: Access Portal module, Resource

Allocation module and Network Infrastructure module. Access Portal module authenticates users, and enforces network security policies and resource management. Resource Allocation module optimally allocates resources by using a centralized user and resource database (i.e., Active Directory). Network Infrastructure module adopts an open infrastructure to seamlessly integrate virtual network pods (a collection of information technology equipment) with physical network pods. Theatrically, there are no limit on the number and type (e.g., VoIP and wireless) of pods.

To the best of our knowledge, HyperLab is the fist remote network laboratory system that can offer the following capabilities at the same time: (1) seamlessly incorporating virtual network pods with physical ones; (2) optimally arranging hardware and software resources, and lab hours; (3) high adaptivity in system upgrading or expansion; and (4) effective security policy enforcement and monitoring.

The rest of the paper is organized as follows. After reviewing the related work in Section II, we discuss the design and implementation of HyperLab in Section III. In Section IV, we then use one case study to illustrate how HyperLab can be adopted in computer networking courses and share our experience in Section V. We conclude our work in Section VI.

II. Related Work

Providing effective hands-on learning environment in computer networking education has drawn extensive attention. There are many published papers describing the needs and importance for universities and colleges to develop and support hands-on networking exercises. This laboratory handbook, ”Hands-on Networking with Internet Technologies” by Comer & Laverell (2002), presents laboratory setups and exercises that can help universities implement TCP/IP labs for use within an introductory networking course. Similar work can also be found in Greca et al. (2003), Francia & Chao (2004) and Sarkar, N. I. (2006). In the following, we focus our work on two aspects of network laboratories that are relevant to our research: remote access to the laboratory infrastructure and virtualization of the laboratories.

A. Virtual network laboratories

Network simulations and emulations have drawn a lot of attentions. NS-2, was developed since 1995 (Network Simulator (1995)), is still widely used in many network research areas. EmuLab, described by White et al. (2002) presents a large environment for network emulations. Many software based approaches were developed using single host to emulate routers or other specific network functions, such as Gerdes, J. & Tilley, S. (2007); Kneale et al. (2004); Kohler et al. (2000); Li, P. (2009) and Stockman (2003). There are many commercial network simulation products. This range from simple simulators used for training purpose such as RouterSim’s Network Visualizer 4.1 (2005) or Boson Network Simulator (2005) to complex enterprise-class performance simulators such as Scalable Network Technologies’ Qualnet (2005).

One of the main drawbacks of any previous virtualization method is that it relies on the simulation software to mimic the behaviors/interactions of various network protocols. Thus, it presents significant limitations on its flexibilities and functionalities. We will show later that our adopted virtualization approach (i.e., Dynamips) is fundamentally different from previous ones, which directly emulates various Cisco router hardware platforms to support running IOS, other than simulate various IOS functionalities.

B. Remote Network Laboratories

With the large increase of courseware and degrees available online, educators are faced with the challenge of offering practical hands-on networking exercises to distance learning students. Several research, such as Yoo and Hovis (2004), Sloan (2002) and Sloan and Schlindwein (2003) present methods to offer remote access to network labs. The environment provided is simple and accommodate basic exercise in network environment. Yoo and Hovis (2004) discusses the fact that a virtual environment does not properly expose students to what they will encounter and stresses the merit of a concrete, realistic remotely

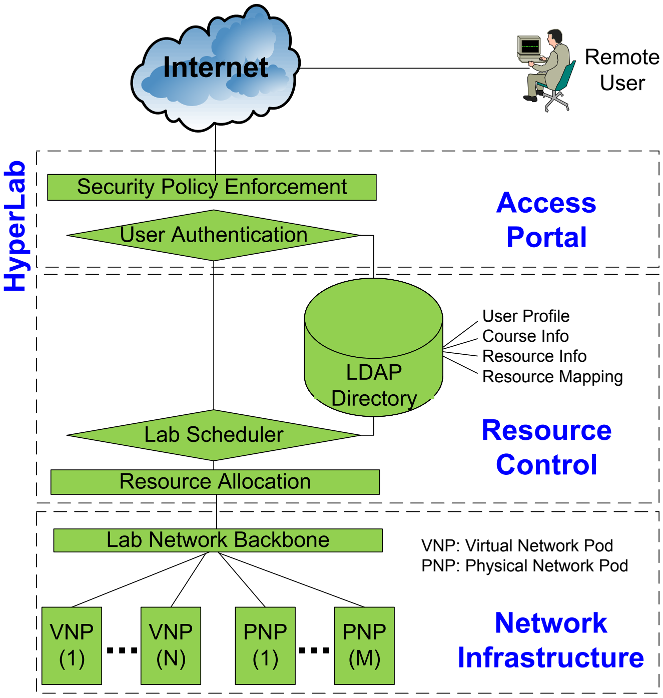


Fig. 1. HyperLab System Overview

accessible environment. Overall the existing research shows the need for appropriate protection for the lab environment as well as the authentication of remote users. The control methods presented in the research are not sufficient for a remote network laboratory. Moreover, with budget constrain, the current design of remote network labs lack of sufficient flexibility and extensibility to meet with the various and changing requirements in computer networking education.

III. System Design and Implementation

HyperLab architecture consists of three functional modules as shown in Fig. 1: Access Portal Module (APM), Resource Control Module (RCM) and Network Infrastructure Module (NIM). Access Portal Module functions as the only interface between public network (i.e., the Internet) and HyperLab. Resource Allocation Module allows students reserve available lab hours and the corresponding network equipment to efficiently utilize the lab resources. Network Infrastructure Module provides the lab backbone to connect multiple virtual and physical network pods. In the following, we elaborate each module and its corresponding features in facilitating computer networking education.

A. Access Portal Module

The main purpose of the Access Portal Module in HyperLab is two-fold: user authentication, and security policy enforcement and monitoring.

Traditionally, a hands-on network lab is completely isolated from production networks to prevent any unexpected traffic from affecting the production network operations. To enable remote access capability for distance learning, we need connect HyperLab to the corresponding production network in which it is located. Thus, a well-designed security policy enforcement component is crucial to make it acceptable by the local network administrators. In the security policy enforcement component (called SPE) of the Access Portal Module, we adopt a Windows 2003 server based VPN system to physically connect the HyperLab to the Internet, but logically separate them with double sided security policy. On the public interface of SPE, it only allows VPN connection from remote students. On the private interface of SPE, it only allows TELNET connection. We also use an IDS system (i.e., SNORT) to monitor the incoming and outgoing traffic, and automatically report any abnormal network traffic.

User Authentication component is not only to prevent illegitimate access to the HyperLab, but also enforce the resource allocation policy as we will discuss later. All user profiles are stored in a Active Directory hosted on the same Windows 2003 server as the VPN system.

B. Resource Control Module

HyperLab can be used by multiple simultaneously offered networking courses (or multiple sections of the same course). In the Active Directory of a HyperLab system, there are three type of objects: student, course, resource. Each student has an unique user profile, which is an user object in Active Directory. Each offered course has a corresponding course object, which belongs to a group object in Active Directory. Each network device has an unique object ID with the corresponding reverse TELNET port number as its extended LDAP property, which is also a group object in Active Directory. Each student can enroll into multiple networking courses, and each course can be granted access to a set of network devices. Network devices can be managed with two different granularity: individual device level or pod level. We will discuss the difference in the next section.

Resource Control module plays an important role in scheduling computer networking courses each semester. Typically, all networking courses with lab components are grouped and considered together. Special pods (e.g., the one with wireless or VoIP equipment) are reserved to the corresponding courses only. General pods are mapped to different courses to maximize the average lab hours for each student.

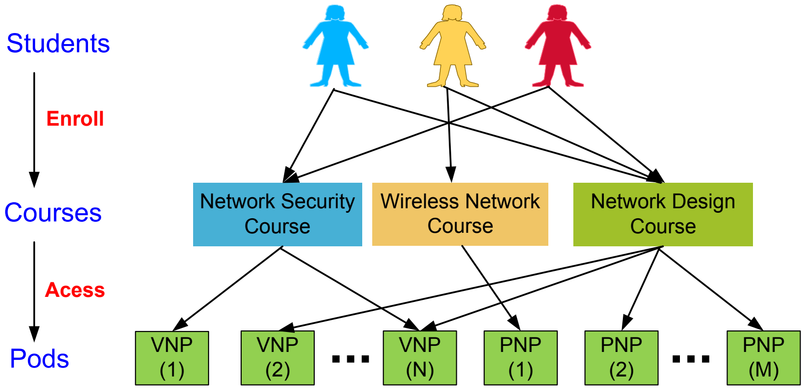


Fig. 2. HyperLab Resource Control

C. Network Infrastructure Module

In the HyperLab, we adopt a Hybridized Network Infrastructure (HNI) in the HyperLab, which consists of both Virtual Network pods (VN) and Physical Network pods (PN). The Hybridized Network Infrastructure is used to achieve the high scalability and flexibility, and balance the different learning experience from emulated/virtual networks and real networks.

The Hybridized Network Infrastructure is modularized and thus highly extensible, depending on the budgetary constrain. A minimal HNI may be composed of only one virtual or physical network pod. In the following, we discuss the implementation of the Virtual Networks and Physical Networks respectively.

1) Virtual Network Pod: There are many network training simulators in the market, such as Boson Products for Cisco Certification and Training. However, these simulators are purely software based applications, which cannot give the real interactive experience to students.

In HyperLab, we adopt Dynamips as the virtualization technology. Dynamips is a Cisco router emulator written by Christophe Fillot. It emulates Cisco 1700, 2600, 3600, 3700, and 7200 series router hardware platforms, and runs standard IOS images. Multiple Dynamips router instances can be running simultaneously on a single computer. Each router instance can have various interfaces and modules, depending on the specification of the corresponding real router series platform. The network topology and the corresponding connectivity among router interfaces are controlled by a Dynamips network topology configuration file, which can be arbitrarily modified for different learning purposes. This software was developed under the GNU GPL license and is free for downloading and distribution. The size of the memory taken by each router instance is user-specified but has to meet with the minimal requirement of the corresponding IOS. For example, the advanced Cisco 7200 series router IOS takes 128MB to 256MB of memory with IP services, VPN, Firewall, and other advanced features. The Cisco 3620 series router IOS can take as less as 64MB of memory for its basic functionalities. We have successfully run 55 Cisco 7200 router instances simultaneously on a single PowerEdge 1900 with 2.66 GHz Dual Core CPU (the actual average CPU usage is 80%) and 4GB memory (the actual used memory is 2.2GB).

Each virtual router instance communicates with each other through the configurable UDP ports within the same system memory. As a result, data throughput between devices is much faster than the physical serial or Ethernet cable connection. Students connect to these virtual router instances through the assigned TCP ports of the hosting machine. TCP connections are used for constructing TELNET sessions so that the students can remotely login and configure the virtual router instances. This process is the same as the reverse TELNET access to physical Cisco devices via Cisco terminal server. Thus, the students may not even notice the difference between the emulated and real equipment if they login remotely by a given IP address of the host machine.

2) Physical Network Pod: Dynamips is a great alternative to create Virtual Networks. However, it has its limitations. We just name a few here.

- Although Dynamips provides a simple virtual switch via the NM-16ESW network module, it does not emulate Catalyst switches, and thus advanced layer two technologies (e.g., EtherChannel) cannot be implemented using Dynamips virtualization.
- Some network physical layer characteristics cannot be observed in Virtual Networks. For example, Dynamips has no distinction on DCE and DTE serial interfaces and does not require setting up an appropriate clock rate on each DCE interface.
- With the nature of Dynamips, it cannot be used for accurate performance benchmark.
- Currently, Dynamips only emulates the vendor specific products (i.e., Cisco router platforms).

Thus, in addition to Virtual Networks, Physical Networks are often necessary to be created to increase the student learning experience. In HyperLab, we use a Cisco terminal server to provide out-of-band access for multiple devices. A Cisco terminal server is a router with multiple, low speed, asynchronous ports that are connected to other serial devices, for example, modems or console ports on routers or switches. The terminal server allows students to use a single point via reverse TELNET to access the console ports of many devices. Reverse TELNET gives users the ability to telnet to a Cisco terminal server, and then console to another device from there. For example, we use as the terminal server in the current Physical Network of HyperLab Cisco 2511, which can provide console connections to maximally 16 various network devices.

3) Network Infrastructure Topology and Integration: As we presented earlier, there should be at least one virtual or physical network pod included in a HyperLab system. With the hybridized and modularized design in HyperLab, we can flexibly add/remove both virtual and physical network pods.

The topology of a virtual network pod is defined by the corresponding Dynamips network configuration file. Dynamips can bridge virtual router interfaces with real host interfaces, allowing the Virtual Network to communicate with the other Virtual Networks and Physical Networks.

For each physical network pod, we include one pod infrastructure switch and make sure all ethernet interfaces of each router are connected to the pod infrastructure switch as shown in Fig. 3. By properly configuring VLAN membership (i.e., two interfaces are logically connected if they are in the same VLAN), the students can remotely and dynamically create various network topologies. This feature is extremely important for some network design courses, in which the students need to adopt different network topologies based on their network design blueprints.

We use a Cisco switch (called backbone switch) in HyperLab to consolidate the lab infrastructure backbone. Each network pod is connected to the backbone switch via the ethernet interface of either a hosting machine (for Virtual Network Pod) or a terminal server (for Physical Network Pod). All network pods are centrally controlled and allocated to different courses. Each pod can be identically for scalability or designed completely different (e.g., for VoIP, wireless or network security courses) for flexibility.

IV. Case Study

In the following, we use one case study to illustrate how HyperLab can be used in computer networking courses.

In the semester of Spring 2009, we offered a new course called Network Design & Analysis, in which the students are required to create different network topologies to achieve various design objectives.

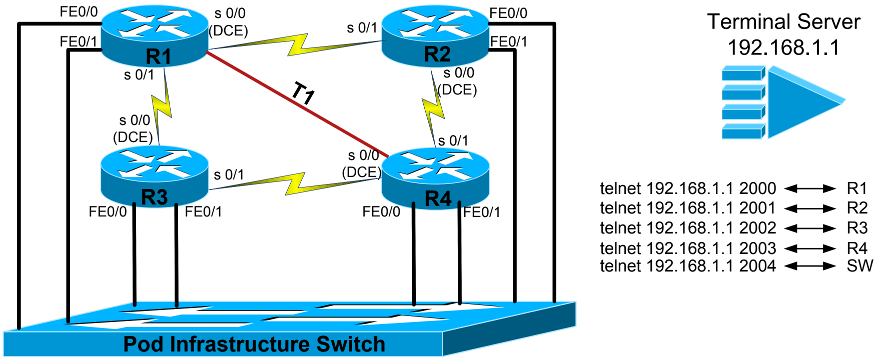
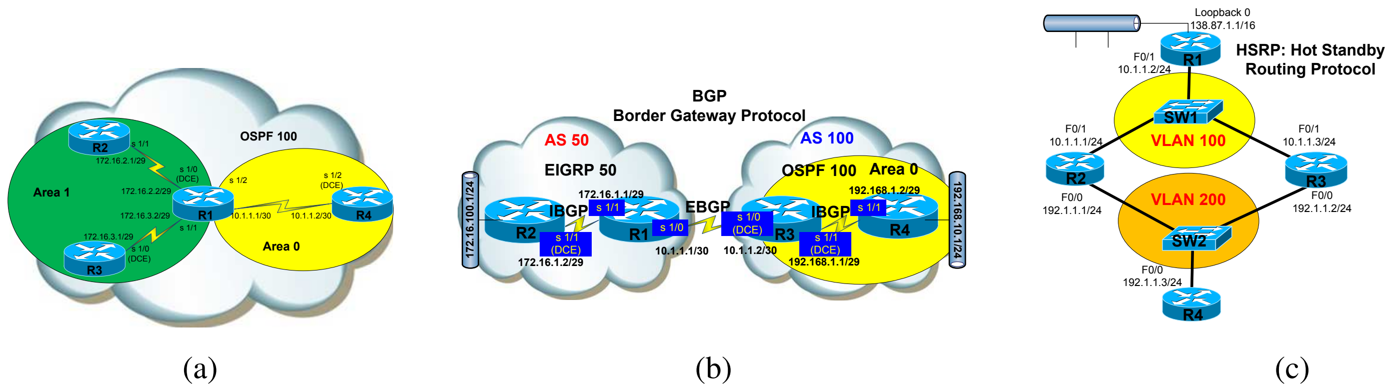

FE0/0
s 0/0
(DCE)
s 0/1
FE0/0
FE0/1
R1
R2
FE0/1
s 0/1
s 0/0
(DCE)
T1
s 0/0
(DCE)
s 0/1
s 0/1
s 0/0
(DCE)
R3
R4
FE0/0
FE0/1
FE0/0
FE0/1
Pod Infrastructure Switch
Terminal Server
192.168.1.1
telnet 192.168.1.1 2000 ⟷ R1
telnet 192.168.1.1 2001 ⟷ R2
telnet 192.168.1.1 2002 ⟷ R3
telnet 192.168.1.1 2003 ⟷ R4
telnet 192.168.1.1 2004 ⟷ SW

Fig. 3. HyperLab Physical Network Pod

With the current budget limit, we use a physical network pod as shown in Fig. 3. In this pod, there are only four routers and one switch, connected by a Cisco 2511 terminal server via console connections. Each router has two serial and two fast ethernet interfaces. All ethernet interfaces are connected to the same pod infrastructure switch. With such a simple network infrastructure, we can create many different network topologies to serve the corresponding learning objectives. To illustrate the idea, we list three network lab exercises as shown in Fig. 4 for configuring OSPF, BGP and HSRP networks respectively.

V. Experience Discussion

Designing a highly adaptive remote network lab is a challenging task. Modularized design and open infrastructure are two important factors to make HyperLab an effective approach to provide a remote network lab. In our implemented HyperLab system, we use one physical network pod and ten virtual network pods, which are all identical for scalability.



(a)

(b)

(c)

Fig. 4. **The various network topologies generated from the Same physical network pod (a) OSPF network (b) BGP network (c) HSRP network**

The Access Portal module successfully allows for the creation of two virtual autonomous systems. The boundary device will have a network interface on the campus side with a campus wide address and a private network interface with the internal laboratory environment. One key benefit of this arrangement is that the use of encapsulated tunnels provided through Layer 2 Tunneling Protocol or Point to Point Tunneling Protocol prevents the campus traffic restriction and firewall policies from influencing our traffic to the laboratory environment. This module can use a VPN server, router, pix, or ASA devices with VPN support, or even VPN concentrators. You should choose based on your budget and traffic requirements.

Central user and resource management is an effective approach to optimally coordinate resources. Typically, this service will be provided by X.500/LDAP standard services such as Active Directory or other Network Directory Service. This directory service must provide a centralized means to manage users, groups, equipment, equipment groups, classes, organizations, or even companies. We have found using Windows based solution may ease the system management tasks. This layer must allow for scheduling users, groups, classes, companies, schools, organizations, or any combination of these to groups of equipment at specific times. One key component is that this scheduling must be centralized and it must be enforceable. In other words, the students must be removed from the environment once their time has expired.

Instead of having to provide a different pod for every student or every group of students at a specific lab time, each student can utilize the same pod but at different periods of time. This can greatly reduce the amount of equipment needed, thus drastically reducing your equipment costs. In addition to the concept of time division, HyperLab introduces the concept of centralized virtualization. The HyperLab system can easily allow inexpensive workstations and servers to function as multiple routers with WIC cards and modules. Effectively, a few previously existing workstations can replace several high-end routers, modules, WIC cards, cables, and racks at a fraction of the cost while occupying only one port on a UPC. Furthermore, the use of VMware, Microsoft, or even Novell technologies can allow for the creation of multiple virtual workstations or servers that can be hosted on a single machine to reduce costs.

Additional roles that can greatly improve quality of learning can be added to HyperLab in a modular fashion. Instructors can post how to videos as links in the web pages or can conduct live or recorded lectures with Streaming Media Server live or on-demand streams respectively. Communication and collaboration technologies such as Skype, AOL, or Windows Live Messenger can be incorporated to allow for team-based communication. Additionally, virtualization technologies such as our in-house-developed Dynamips/Dynagen through Web 2.0 can be used to provide access to virtual equipment. Due to the extremely flexible capabilities of this system, the sky is the limit to the number of previously existing or custom in-house services that can be deployed at this layer.

Network Infrastructure Module consists of the equipment that will be used for labs. This layer supports virtually all Cisco devices in virtually all arrangements, any Linksys, D-Link, NetGear, or HP device, servers, workstations, Alan Bradley or GE Fanuc PLCs, VMWare ESX/ESXi server, SAN device, ADTRAN TSU, Atlas phone device, or Total Access DSLAM, or virtually any other electronic system you can imagine. In addition to supporting all of these devices, this framework supports virtually all configurations using these devices. Voice technologies such as VoIP deployment through Unified CallManager or Unified CallManager Express with H.323 and SIP. Wireless technologies such as LWAPP with light-weight access points, autonomous access points, RFID tags (passive and active), Wireless Control System, wireless with voice integration (7921 phones running VoIP over 802.11G for example), and even site surveillance and spectrum analysis technologies are supported. WAN technologies such as PPPoA or PPPoE through DSL, DSLAMs, T1 connections, T1 Inverse Multiplexing over ATM, ISDN, and frame relay are all readily supported.

In order to clarify, there are a few things to consider. First of all, the separation between layers is at a functional level not a physical level. Many devices can function at multiple levels in this arrangement. Next, bare in mind that the same physical device can have many roles. In smaller environments, a single device can successfully host many roles to reduce costs. Additionally, each level is not limited to just the functions described. This model is meant to serve as a baseline, or rule of thumb for deployment. Every environment will have unique circumstances that may require additional roles. Finally, the HyperLab system supports an infinite number of different lab configurations.

VI. Conclusion

HyperLab proposes an innovative and effective solution to design a remote network lab. HyperLab has the capacity to run telecommunication, business information, industrial electronic, and virtually any electronic technology. Therefore, this flexible framework provides separation of equipment at a course or department level through adjustable equipment groups and domains rather than physical isolation and separated administration. HyperLab can also be expanded with various network pods depending on the budgetary constrain and course requirements.

References

Armitage, W. D., Gaspar, A. & Rideout, M. (2007). Remotely Accessible Sandboxed Environment with Application to a Laboratory Course in Networking, Proceedings of the 8th ACM SIGITE conference on Information technology education.

Border, C. (2007). The Development and Deployment of a Multi-User, Remote Access Virtualization System for Networking, Security, and System Administration Classes, Proceedings of the 38th SIGCSE technical symposium on Computer science education.

Comer, E. & Laverell, W. (2002). Hands-on Networking with Internet Technologies, Prentice Hall Professional Technical Reference.

Dynamips (2007): http://www.ipflow.utc.fr/index.php/Cisco_7200_Simulator

Francia, G & Chao, C. (2004). Computer networking laboratory projects. The Journal of Computing Sciences in Colleges, 19(3), 226-237.

Gerdes, J. & Tilley, S. (2007). A conceptual overview of the virtual networking laboratory, Proceedings of the 8th ACM SIGITE conference on Information technology education.

Greca, A., Cook, R., Harris, J. (2004). Enhancing learning in a data communication and networking course with laboratory experiments. The Journal of Computing Sciences in Colleges, 19(3), 79-83.

Kneale, B., Horta, Ain Y. De & Box, I. (2004). VELNET (Virtual Environment for Learning Networking, Proceedings of the sixth conference on Australasian computing education.

Kohler, E., Morris, R., Chen, B., Jannotti, J. & Kaashoek, M. (2000) The Click Modular Router, ACM Transactions on Computer Systems, Vol 18, No. 3, page 263-297, August 2000

Krichen, J. P. & Lahoud, H. (2008). Remote labs in the online environment: indicators for success, Proceedings of the 9th ACM SIGITE conference on Information technology education.

Li, P. (2009). Exploring Virtual Environments in a Decentralized Lab, ACM SIGITE Research in IT, Vol. 6 No. 1.

Network Simulator (1995). NS-2 The Network Simulator. http://www.isi.edu/nsnam/ns/

RouterSim's Network Visualizer (2005). http://www.routersim.com/

QualNet (2005) - Network Simulator, Scalable Network Technologies - http://www.scalable-networks.com/

Sarkar, N. I. (2006). Teaching TCP/IP Networking Using Practical Laboratory Exercises, International Journal of Information and Communication Technology Education, Volume 2, Issue 4.

Sloan, J. (2002). A remotely accessible networking laboratory. The Journal of Computing in Small Colleges, v.18 n.2, p.215-222, December 2002.

Sloan, J. & Schlindwein, C. (2004). TCP/IP laboratory exercises for use with a remotely accessible networking laboratory, The Journal of Computing in Small Colleges, v.19 n.3, p.68-78, January 2004

Stockman, M. (2003), Creating remotely accessible ”virtual networks” on a single PC to teach computer networking and operating systems, Proceeding of the 4th conference on Information technology education, Lafayette, Indiana, USA. 67-71.

Wang, J., An, Y., Sheng, Y. & Li, S. (2006). IDSVL: Intrusion Detection System Virtual Lab Based on Component in the Internet, Advances in Web Based Learning ICWL.

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

Yoo, S. & Hovis, S. (2004). Remote Access Internetworking Laboratory. Proceedings of the 35th Technical Symposium on Computer Science Education, Norfolk, Virginia, USA, 311-314.