Creating a Virtual Network Laboratory

Yen-Jen Lee, Wei-hsiu Ma, David H.C. Du
Distributed Multimedia Research Center
Department of Computer Science
University of Minnesota
Minneapolis, Minnesota 55455, USA
{ylee,hjlee,wma,du}@cs.umn.edu

James A. Schnepf
Department of Computer Science
College of St. Benedict/St. John's University
Collegeville, Minnesota 56321, USA
jschnepf@cs.csbsju.edu

1 Introduction
As modern society moves into the information age, electronic communication has taken on increased importance in many facets of life. There is increasing recognition of the need to train a more technically qualified workforce with an understanding of the concepts of computer networking and how to effectively apply those concepts to real world problems. In addition, many professionals currently in the networking field need further education and training to maintain their mastery of this area.

Science and engineering departments have long recognized the need for laboratory experience that it is through these experiences that students deepen their understanding of the conceptual material. In contrast, networking courses are traditionally conceptual in nature with little opportunity for students to apply what they have learned in order to strengthen their knowledge. To support these courses, the University of Minnesota (joined with North Hennepin Community College) established a "hands-on" network lab three years ago. However, educational and operational concerns have surfaced throughout the years due to the physical lab constraints and difficulty of satisfying the high demand.

The issues involved can be summarized as follows: 1) the need to bridge the theory and practice of networking technologies and provide students "hands-on" experience; 2) a cost effective way to leverage expensive network equipments and maintain physical laboratory by lab attendants; 3) access to hardware and software of emerging technologies before they reach the market place through a mix of simulated or state-of-the-art experimental setup; 4) a paradigm to provide concurrent on-line instruction, visualization, repeated practice, and feedback to break the geographical, lab space, and time constraints; and 5) a large class enrollment size to meet the strong demand.

To address these issues, and to overcome the shortcomings of current approaches, we are developing a virtual network laboratory environment to eliminate most of the constraints, though it is not necessarily replacing all of the physical settings at once. The kernel will consist of a collection of on-line multimedia learning modules which will integrate the key lab devices with instructional material. The modules are designed to allow students carry out hands-on networking labs in a virtual environment or through remote access. This virtual laboratory will be easily accessible over the Internet.

2 Framework
To support the development of the virtual lab environment, we proposed a three-stage approach to the design and implementation. In each of these stages, we will develop a collection of multimedia learning modules using Java, an object-oriented programming language. We will center (but not limit) these modules on the series of instructional lab exercises we have developed.

2.1 Approach
The stages of development are as follows: 1) establish an environment where students can remotely access equipments in the existing lab (and potentially access to industrial partners' state-of-the-art networking devices) and carry out experiments without physically presented in the lab; 2) develop a set of virtual laboratory components that can be incorporated into existing lab instructional material so that experiments can be performed by providing interaction and feedback through captured data or pre-orchestrated simulation sequence; and 3) based upon the framework that has been developed for part two, plan on and develop a set of experiments based on emerging technologies.

Students will perform the experiments via a collection of multimedia learning modules. The seamless integration of WWW and Java enables a platform-neutral implementation of the system modules.

1This work is supported in part by the NSF under Grant CDA-9502979 and a gift from IBM.
2.2 Module Design using Java

The module design consists of three correlated components: client runtime, server runtime, and physical networking device. As shown in Figure 1 using Java for the design framework, client runtime consists of: a graphical user/command front-end interface to assimilate the look-and-feel of a physical device, a back-end access object to simulate the state and command control of the physical device, and a communication entity for the back-end to retrieve pre-orchestrated simulation sequence or communicate with a physical device through a proxy service. The server runtime provides services to multiple clients concurrently. It uses a *pseudo device* object to respond to clients’ back end access requests for simulated sequences; or, it may access a physical device if the network access mechanism is available through a *proxy agent*.

The coherent design of client runtime and server runtime for lab modules promotes common reusable objects and standard implementation paradigm without adhering to a specific hardware platform.

3 Prototype Implementation

The direction we are leaning towards is to gradually transform the lab instruction and experiments to an on-line learning space which provides a virtual setting and interaction mimicking the real physical environment. We have developed two functional components according to the framework and design described earlier. One is a simple protocol analyzer. The other is the performance analyzer which provides real-time network throughput and latency measurement. These two components are being incorporated into several existing lab experiments, e.g., the TCP/IP lab experiment.

3.1 Protocol Analyzer

The Protocol Analyzer implements a protocol data display which is self-executable in a web page with a Java-enabled browser or appletviewer. The appearance of the browser component is depicted in Figure 2. There is a stand-alone server application written in Java to feed simulated sequences to the client. The operation of a protocol analyzer has distinct phases of data capture and data display. Hence, the network traffic is captured and stored beforehand rather than displayed on-the-fly. The prototype is flexible enough to display different protocol suites as long as they are in a layered structure.

Figure 1: Framework for Virtual Network Lab Module.

[Diagram of framework]

The current implementation of the performance analyzer provides monitoring of real-time performance for TCP and UDP protocols. For both of the TCP and UDP protocols, the analyzer measures the round-trip latency and throughput over the target network between two machines with varying message sizes. The result is visualized on the user site. In addition, the receipt or loss of each UDP packet is also explicitly shown. Figure 3 demonstrates the UDP performance analysis.

3.2 Performance Analyzer

The current implementation of the performance analyzer provides monitoring of real-time performance for TCP and UDP protocols. For both of the TCP and UDP protocols, the analyzer measures the round-trip latency and throughput over the target network between two machines with varying message sizes. The result is visualized on the user site. In addition, the receipt or loss of each UDP packet is also explicitly shown. Figure 3 demonstrates the UDP performance analysis.

Figure 2: Data display in protocol analyzer.

[Diagram of protocol analyzer]

Figure 3: Performance analyzer for UDP.