ABSTRACT
Computer science students who take a network programming or system administration class that involves a hands-on lab are generally not given administrative rights on lab machines because universities have security policies in place to protect the integrity of the network infrastructure. The research community is beginning to address this by applying Virtual Machine (VM) technology such as VMware and Xen. The goal is to provide a virtualized environment so students have full control over their own operating system and can perform necessary classroom tasks. To date, most of the trial deployments of VMs in the classroom have been geared to network security and system administration classes. Hands-on labs that support networking concepts taught in lecture can have positive benefits. A hands-on lab also provides the opportunity to engage students with ‘real-world’ problems that require creativity and resourcefulness.

Computer science students are introduced to networking either using the OSI seven layer stack as the model or using a ‘top-down’ approach that starts with the application [reference to Comer, Stevens, Kurose text books]. To facilitate a deeper understanding of the material, virtualized designs need to be able to provide an environment to allow students to run their TCP/UDP programs and evaluate realistic network performance via WAN metrics like throughput, drop rates, and packet delays. This paper investigates three different technical designs using Xen virtualization to accompany network programming classes. Designs include a physical network lab setup similar to work being done by other researchers, a laptop installation method, and the use of a Xen based LiveCD/DVD. Activities and results based on a network programming class are shown in two of the designs and future work with a Xen LiveCD/DVD is proposed.

Categories and Subject Descriptors
K.3.2 Computer and Information Science Education

General Terms
Design, Experimentation, Performance.

Keywords
Virtualization, Xen, Network Programming, Education

1. INTRODUCTION
Computer networking is filled with varying technologies, operating systems, and equipment trying to provide a reliable means of handling the computer communication needs of individuals, universities and businesses alike. Networking education seeks to deliver an atmosphere in which a student can learn such a broad spectrum of information, but often faces the challenge of not only simulating a network environment for educational purposes, but also putting together labs and activities to increase student retention and meet the class objectives. From previous work experience starting in 2001 [1], one way of setting up a laboratory environment for networking classes would be to situate the number of computers in a classroom together with a switch and/or router. One difficulty with this approach is in providing a variety of operating systems running at the same time so the students could complete an increasing and more complex sequence of exercises on their computers. One way of handling this would be to partition the hard drives into separate operating system boot options so the students could pick which entry to boot for their current task. However, this required long reboot times when changes were needed during laboratory hours.

With the emergence of newer technologies and the World Wide Web, students now have many options from which to learn networking. Schools can provide online learning options like Clemson’s eLearning with SkillPort [2], a self-paced courseware web tool that helps students learn about varying subjects. While this method of technology provides a solid means for a student to learn at their own pace, it doesn’t provide the complex environment needed to emulate computer networks. Students and faculty at North Carolina State University use the Virtual Computing Lab (VCL) [24]. This noted service allows users to remotely access different operating systems and various applications from on-campus computers. Any networking students can also download and run the ns2 application [3], which is a tool that can simulate many networking protocols. This is very useful for research and can greatly enhance the learning of the networking student, but this software isn’t covered within the virtualization framework of this paper.

Most computer labs in colleges today are connected to a university network infrastructure. Students have access to university resources and can go online to do additional research. The downside to this design is the required security needed by the university to maintain their networks and the limitations it puts on computer science students who need administrative privileges to accomplish tasks like: changing IP information on a network...
interface card, setting up routing tables for connectivity, running programs that analyze packet information, etc. Creating a network of computers using virtualization technology gets around this difficulty and provides a resource for students to accomplish administrative tasks because students have complete control of their designated operating system. From an undergraduate perspective, our personal experiences in network programming and system administration classes would have been further enhanced had those administrative rights been present in the computer labs.

Virtualization technology has been around for many years, dating back to IBM’s mainframe architecture [25]. Recently, however, VMware [5] and Xen [6] virtualization software has made significant strides into areas of business and research in the last decade. Virtualization software provides many opportunities in education also. Universities can develop research platforms with this technology and instructors can implement a wide variety of activities to accomplish their classroom objectives by providing a virtualized design to facilitate laboratory exercises. Because machines now have to ability to handle multiple operating systems at the same time, networking, security, and system administration courses can integrate a wide variety of activities within the area of networking. In the background section of this paper, we provide a summary of educational approaches using virtualization and they are applied to networking education.

While these approaches focus on network topics, security, and system administration, all networking courses offered in Clemson’s Computer Science (CS) department have a significant programming component. CS networking students are introduced to networking either using the OSI seven layer stack as the model or using a ‘top-down’ approach that starts with the application [23]. Students also learn the networking fundamentals and network protocols similar to the classes mentioned above but also incorporate programming projects. In this paper we focus on evaluating technical designs for network programming and performance analysis, those objectives not necessarily found in generic network classes in a small-scale and hands-on student approach. We analyze three virtualized topologies so that TCP/UDP programming and the study of WAN characteristics fit in a paradigm needed for successful completion of a computer science networking course.

Three technical virtualized designs are introduced in this paper: 1) a centralized Xen topology on a physical network similar to current research projects, 2) a virtualized laptop installation design, and 3) the introduction of using LiveCD/DVD technology as a platform for networking education. Each of the technical designs is described in their respective sections. We have spent a couple of semesters using the first design in networking classes and present our findings from that experience. Another semester was spent on the laptop installation method as a project in a systems modeling course and results are also mentioned. We propose a design and use for the LiveCD at the end of the paper and finalize our results in the conclusion/future work section. The appendix includes sample projects and laboratories used in this paper.

2. BACKGROUND

2.1 Xen

Xen is an open-source virtualization tool developed at Cambridge Laboratories [6, 7]. Virtualization, in a broad sense, is the encapsulation of multiple operating systems running concurrently on the same physical computer. Figure 1 is an example of a virtualization diagram.

The virtual machine monitor (VMM), as in Xen’s case, called the hypervisor, manages the systems resources between multiple instances of OS’s. This type of structure allows for the creation of testing environments and the mapping of network topologies. Instances of OS’s running on top of the hypervisor are called virtual machines (VMs). These machines can be configured with a modified Xen kernel, called a para-virtualized machine or setup as a fully featured independent OS, like Windows, a fully virtualized machine. Use of a para-virtualized Xen OS has many benefits, especially better access to the system’s I/O and communication between the VMM and VM [7, 8, 9]. Full virtualization allows the Xen hypervisor to manage VMs such as Windows and Solaris. Full virtualization allows for OS kernel independence but sometimes has large overhead [6].

![Virtualization Diagram](image)

Figure 1 – Virtualization Diagram

2.2 Virtualized Networking Education

Virtualization provides many opportunities for educational use. Educators can set up virtual networks for student laboratories for a variety of tasks. Programmers can test their applications before pushing their code into a ‘live’ environment. Virtualization can also be a cost-effective approach for hosting multiple operating systems, thus saving on computer costs and increasing CPU utilization [5, 6, 7, 10]. Marist College uses the IBM z/VM virtualization hypervisor to provide a core service for a variety of classes, email, websites, etc. [26] Users request a service from this VM technology without knowing the underlying structure; they believe they are using this service as though it was on their own computer. Marist College’s partnership with IBM will allow for continued support of the technology and implement future advances to the VM system.

With the popularity of virtualization increasing in the past few years, especially VMware and Xen, there has also been an increase in publications of virtualized environments for use in network, network security, and system administration classes. The physical designs of this paper have been investigated and used in live environments over the last year and its timing coincides with some of the research currently being worked on at other universities now. We discuss a few of these publications and how they use virtualization technology in their coursework next.

User-Mode Linux (UML) is a virtualization technology that is being used today for business and educational purposes [10]. J.
Walden at the University of Toledo implemented UML virtualization in a computer security class by separating students into groups and having them compete on a virtual network by playing “Capture the Flag” [11]. This is a very engaging and innovative way for students to learn about computer and network security in a competitive and high-paced environment. This type of activity could also be applied to a multitude of other exercises.

One of the leaders in the marketplace and producers of virtualization technology is VMware [5]. VMware was bought out by EMC and is expected to have an initial public offer (IPO) this year. VMware products are used as educational tools too. From personal experience [1], we used VMware products to teach introduction to network courses at ECPI last year. Students were able to start images of Fedora or Windows to use with their networking labs. This provided a unique opportunity for them to see differences in how Linux and Windows approach networking concepts running at the same time on their computers. Butler, W. et al. at the University of New Mexico developed a virtualized environment using VMware Workstation software running multiple instances of Windows, Windows Server, and FreeBSD for network, security and database courses [12]. Student were able to learn a variety of topics from within these classes and the instructors were empowered in providing a complexity of exercises over the traditional partition hard drive approach mentioned in the introduction, thus saving time in the configuration and setup between classes. C. Border at the Rochester Institute of Technology extends the University of New Mexico’s approach for students who wish to participate in laboratory exercises remotely by using Microsoft’s Remote Desktop Connection, Microsoft Terminal Services, VMware workstation and Remote Laboratory Emulation System (RLES) [13].

Of particular interest, and the focus of evaluation in the virtualized designs of this paper, is Xen virtualization. Xen is an open-source, cost effective approach to virtualization [6] and we briefly discuss two publications that have used Xen in their educational frameworks. M. Alexander and J. A. Lee [15] use a combination of Xen and a LAMP based application of Weblab for the design and use of a web-based virtual environment for networking courses. Weblab was initially implemented as a VMware design; the authors then added Xen to the structure with their virtual machines running Linux CentOS operating systems. Another Xen configuration was used at Iowa State University’s Department of Electrical and Computer Engineering [14]. T. Daniels and B. Anderson have developed a grid of virtual machines, called Xen World, to handle a virtualized infrastructure for their security education classes. Their approach was to be able to deliver a low-cost, flexible, and tailored to students who have control over their own designated VMs. Their initial project incorporated the use of one server handling VMs for thirty students. Their current configuration in the publication now has the ability to run 300 VMs with the hope of one day supporting an education structure of a 1000 VMs for a variety of Computer Engineering classes.

Each of the virtualized environments above has to go through a design life-cycle to be able to meet educational objectives. Changes to the environment often are required because of security concerns, infrastructure design, technology interoperability, and classroom dynamics. Another concern is the ease of OS updates in order to support the continuous changes in Linux releases. A good virtualized design is met when the technology and the needs of the students are put together in a way that is cohesive and adaptable. Many of the changes to design are made over a period of time. From the school’s point of view in delivering these types of environments to students, instructor preparation time and costs for the design are often determining factors in their implementation.

3. MOTIVATION

Each of the virtualization approaches in the background section is primarily geared to network, network security, system administration or database courses. From personal experience in undergraduate computer science courses, many of the topics in the classes listed above are covered in a computer science course at Clemson University. One of the areas not necessarily found in them, though, is the additional element of having to develop and run software programming labs and projects that are required for computer science students. This is not stating that those classes don’t incorporate programming, but probably not to the extent of someone majoring in computer science.

The computer science laboratories are incorporated within the Clemson network infrastructure. First and second year students have hours set up for credit and are assigned to a lab to do classroom exercises in them. This setup is excellent for beginning programming classes. A problem might arise when a student reaches a CS networking or systems administration class and these laboratories are used for supplementary instruction (There weren’t designated lab hours required for CS networking or systems administration classes at the time of writing this paper. Students have the opportunity to use these labs outside of class to program their projects). The students perform their labs in these classrooms but there aren’t any administrative rights on the machines because of the security policies set by the university. Having administrative rights and control of your operating system is essential in learning networking concepts.

Computer science network students learn such topics as DNS, DHCP, NIS and NFS, to name a few, but the main components of their learning are focused on network programming and performance analysis. A majority of their time outside of class is spent on a few small projects or a large semester one that is geared towards programming TCP/UDP socket applications. In order for students to gain a better understanding of networking by analyzing packets that travel over a physical media, technical virtualized designs need to be able to provide an environment to allow students to run their TCP/UDP programs and evaluate realistic network performance via WAN metrics like throughput, drop rates, and packet delays. We look at and evaluate three designs in the next section of the paper.

Before looking at the Xen virtualization designs, we will define the scope of a CS networking class, what areas of learning we concentrate on in our investigation of Xen, and what concerns we might have in a virtualized environment. A description of a CS networking class here at Clemson is shown below:

“CP SC 360 Networks and Network Programming 3(3,0)
Introduction to basic concepts of computer network technologies and network programming. Topics include network programming, layered protocol architectures, local and wide area networks,
internetwork and intranetwork concepts, security. Socket level programming is introduced and used throughout the course. Preq: CP SC 212, 215 with a C or better.”

An example list of objectives from a syllabus might have [4]:

- Demonstrate an understanding of basic networking concepts.
- Demonstrate an understanding of the set of protocols that make up TCP/IP.
- Implement moderately complex networked applications using the BSD sockets interface.

You will note that the description above falls within what would be taught in the virtualized classes the paper discussed earlier in the background section. For this paper we investigate the Xen technical designs and how they apply to a CS networking class by seeing how they work with virtualization. We investigate subjects in virtualized technical plans based on:

- **General network topics** – these would include traditional items like: DNS, DHCP, NFS, TCP, IP, etc.; those found in network, security, or administration classes
- **TCP/UDP socket programming** – student’s labs and projects are primarily geared toward the programming of TCP and UDP programs and how the network would accommodate this
- **Network topologies** – what type of environment can be created with these designs and what the students can learn from them, for example, setting up a realistic WAN network
- **WAN characteristics** – learning how networks communicate by running programs or tools that allow for an understanding of throughput, drop rates, and packet delays with tools like tcpdump, tcptrace, and MTRG
- **Enriching administrative skills** – provides opportunity to do network administration and for students to setup and maintain there own apache web site

There are some questions related to the topics above that might draw some concern when an instructor or graduate assistant is developing a virtualized environment to accompany network programming classes:

1) Computer science classes are generally small in number, does this design allow for flexibility to user/group management and can the technology handle the number of students?
2) What level of understanding and experience would a student need for Linux and virtualization in this design?
3) What is the preparation time needed for this design? How much flexibility is there in this approach to accommodate changes needed between labs and projects?
4) How does this virtualized technical design affect WAN characteristics like throughput, drop rates, and packet delays?

We’ll answer these questions after each of the three technical designs up for investigation. In our first design we take a look at using 2 Xen servers with 1 other machine acting as a router. This virtualized design is the primary network that will be evaluated because it was used more extensively and over a longer period of time than the others. The second design incorporates the use of installing Xen on a student laptop. The final design is a proposed method at using LiveCD/DVD technology as a device for network programming.

### 4. VIRTUALIZED NETWORK DESIGN – 2 XEN SERVERS

#### 4.1 Topology

Figure 2 is a description of the network that was set up for an undergraduate CS networking class and a graduate internetworking class. This design was used over a period of two semesters, first with the graduate course and then setup for the undergraduate one. There are three physical machines in this topology: 1 Dell server and 2 Sun servers. The Dell machine, named borg, had 3 NIC’s that was setup to provide connectivity to the Clemson network but also designed to allow for two private subnets to the Xen servers, csvm1 and csvm2. Each of the Xen servers were Sun Opteron 148 computers, each running at 2.2 GHz, with 4GB of RAM, and two 250GB hard drives. Borg has a FreeBSD operating system with a network emulator called DummyNet [16] that allows a user to define WAN characteristics over a designated NIC. Each of the Xen servers used the Fedora Core 5 operating system for the hypervisor (VMM) and 10 separate VMs. Overall, there were 20 VM available for use by the students. Each VM was set to run on 256MB of memory and the VM space on the hard drive took up 5GB.

#### 4.2 Usage

For the graduate course, each group in the class was assigned a pair of VMs, one on each Xen server so that the VMs would be on different subnets. There weren’t many students in the class, so groups were small in number. The students in the graduate course were able to use the Xen network for the projects assigned to them in class during the semester. The Xen network was setup for the undergraduate class the next semester. This class used the virtualized network extensively, mainly for labs that were created for this design, but also made available for use with the project they had to do for the semester. This class had many students and group numbers ranged from 3 to 4 students each. Again, each group had a pair of VMs, each one on different Xen servers.

Each group in both classes was assigned accounts on Borg and both Xen servers. Students accessed these machines via SSH then shelled into their respective VMs as root users. Students then created their own user directories and then performed tasks as
necessary. DummyNet was used if any lab or project needed changes set for WAN characteristics.

4.3 Impressions
We first investigated this design by trying to use Fedora Core (FC) 4 as the operating system, but at that time there seemed to be difficulty in setting up matching a FC kernel and Xen release versions that would function together with some measure of success. We then switched to Fedora Core 5 and were then able to get a stable environment from which to work. Each of the VMs were setup on 5GB of hard drive space and setting up 10 VMs on one server took much time. Although we didn’t notice many difficulties in using this design with the graduate course, there were some comments from students trying to get access to their respective VMs. Their comments were about having to run multiple SSH commands just to get to their own VM. A student would have to shell into Borg, a Xen server, and then their VM. This was also evident in the undergraduate course. Another observation was for copying their files back and forth from the Clemson network and their VMs on which to test their programs.

We created labs for the students in the undergraduate course designed to get them started on their VMs and on how to SSH, create accounts, copy/moving files. These seemed to go well. When then focused our labs in using commands like tcpdump to be able to understand types of packets that go through the NIC and then advanced from there. Two sample labs are located in the appendix. Labs seemed to go well and the virtualized network provides for a variety of tasks that can be completed by the students. Setting up the Xen VMM to use bridging and NAT for network specifications adds versatility to future laboratories exercises.

We will notice that if there are many groups using the 2 Xen servers at the same time, testing out their programs, and then trying to get good results for WAN characteristics, a problem is created. Each server is using a gigabit NIC, but when you have a few groups competing over the same media, inaccurate throughput, drop rates, or packet delays might be evident. If the activity is interactive and is intended to produce this (which some projects are intended for), this type of network is favorable, but if students are expecting drop rates of .04% because we have setup DummyNet with this setting, students may see much higher levels of packet drops and wonder what is wrong with their applications. Accurate results on WAN characteristics is probably the most important factor in socket programming—being able to accurately predict results from traffic patterns through TCP and UDP.

4.4 Recommendations
• Provide scripts to handle multiple SSH commands and the moving of files, or connect the Xen servers directly to the university network.
• Add one more Xen server to decrease number of students in each group.
• Assign time segments to teams for them to run their programs so WAN characteristics can be accurate.

5. VIRTUALIZED NETWORK DESIGN – LAPTOP INSTALLATION

5.1 Topology
Figure 3 is a description of a Xen virtualized network on one laptop. The Xen hypervisor and 5 VMs are running on the same machine. This topology was installed on a Dell Latitude 620, dual core 2.0 GHz, with 2GB of memory and an 80GB hard drive. The hard drive was partitioned to boot to Windows XP or Fedora Core 6, which was the OS used by Xen. Each VM was set to have 5GB of hard drive space and run on 256MB of RAM. In order to create this topology, XEN bridging had to be used. The laptop comes with one NIC but a PCMCIA network card was bought and attached to this topology that allowed for the use of two NIC’s, one for each virtual bridge shown in the figure. Each VM had Iperf installed on them. Iperf [18] is a tool for measuring bandwidth on a NIC and can also show statistics on packet delay and packet loss. The Linux kernel 2.6 has a network tool, Netem [19], which can be used for WAN characteristics and setup to run when required in labs or projects. Netem was run on a VM that had two virtual NIC’s and was setup to run as a router between two subnets.

5.2 Usage
This setup wasn’t used in a networking class, but run as a project for a graduate system’s modeling project investigating the use of a Xen laptop model as an educational tool for networking students. This was a personal project that was developed to coincide with the first design of the 2 Xen servers. The intent of this was to see if installing a virtual environment could be done on one laptop and the extent that it could be used by undergraduate students in a CS networking course. Fedora, CentOS, and RHEL provide a Xen utility called virt-install that makes it easier to install VMs than in the 2 Xen server design.

We didn’t have to create different groups but were able to use one account for this design. Users and groups are easy to create and manage on one laptop. Students would setup their own configurations for their labs and projects. This is an independent approach as opposed to the 2 Xen server design. The only limit in terms of management of the network is left up to the student who uses that laptop.

5.3 Impressions
It wasn’t difficult to setup the VMs using the virt-install utility provided by Fedora Core 6, although it was very time consuming. Xen’s bridging automatically detected the PCMCIA network card and set up two separate bridges to run on the VMM (The wireless NIC was not used for this design). Each VMs NIC had to be
configured for their respective bridge, but that was only a change in a couple of lines of a configuration file. In order to get routing to work on the VM that acted as a router, packet-forwarding had to be turned on. Each VM needed a routing table entry for the other subnet in order to be able to send packets through the VM router. The first two labs written for this class could be general directions for setting up Xen on their laptops to get to this topology.

One problem that would probably arise from this design is the level of experience an undergraduate student has installing and using Linux and Xen. The intent of the class is to focus on the network programming objectives and not be diverted by taking inordinate amounts of time to prepare their laptop. While a great learning tool for installing and using Xen, it might be too distracting. This design is very flexible and accommodates many labs and potential projects. A few tests from this design are provided in the appendix.

Another observation noted (and also shown in results in the appendix) was some very interesting WAN behaviors between VMs. Throughput tests between two VMs using Iperf showed significant differences between para-virtualized machines and fully virtualized machines. This variability could produce havoc on students trying to run TCP/UDP performance programs. It is a good environment from which to run programs investigating the network performance on virtualized machines, but not on any project looking for expected results in socket programming. There are published articles explaining the difference between para- and fully virtualized machines but that goes beyond the grasp of this investigation. Netem performed accordingly in the tests on the router.

Probably one of the most difficult challenges to this design will be the variability in student laptop hardware. There will be obvious hardware differences and the problems associated with installing a Linux OS and Xen on them. Students would probably need at least 2GB of RAM and 60GB or more space on their hard drive for this design to work.

5.4 Recommendations

- Setup an installation lab with a local Linux Users Group to help with installing Fedora Core 6 and Xen.
- Mandate minimum hardware requirements for the laptop.
- Find a Linux tool to use on the NIC that would determine the amount of throughput it can achieve.
- Designate two class lectures to Xen virtualization.

6. VIRTUALIZED NETWORK DESIGN – LIVECD/DVD/USB

We didn’t have time and weren’t able to employ the use of a LiveCD in a networking class. We look at this technology as a possible alternative to the 2 Xen server and laptop installation designs. We discuss the technology because it doesn’t require any installation and can be used as a simpler way of managing a virtualized network. LiveCD’s, or the term ‘live’, allow a user to insert a CD, DVD, USB, or some other bootable device into a computer and power up the machine to an OS—all without the use of a hard drive. One of the best known LiveCD’s in use today is called Knoppix [20]. There is a Xen LiveCD available from Xensource [6] and one developed in Japan, called Xenoppix [21]. Both have been experimented with and from our experience we find that is could be used in a design for a virtualized network, albeit a limited one.

In order for a LiveCD to be used in a CS networking course, it must be customized to run a Xen hypervisor with at least two VMs so that a network can be run for network programming projects. The Xensource LiveCD can start two different VMs, but doesn’t have all the capabilities we would like in a virtual machine for programming purposes and WAN management. There are some options now that the use of LiveCD’s is becoming more prevalent. There are software tools out there that can help create a customized LiveCD, for example, Revisor, which is now available on Fedora 7 and comes from the Fedora Unity Project [22].

LiveCD’s come with some limitations. The environment isn’t necessarily stable. Once you power down the computer, you lose everything you had in memory. You will need to find some way in which to save your files, for example, save your info to a USB device. If you use a LiveCD, there are performance hits because your CD-ROM drive is slower than memory. Your computer’s BIOS also needs to be able to detect your bootable Live device.

7. CONCLUSIONS/FUTURE WORK

The 2 Xen server design was able to manage a total of 20 VMs, 10 on each server. This was accommodating to the number of students in each of the classes this design was used. Group numbers usually had two or three members, small enough to handle lab assignments. If the number of students increases, RAM would need to be added to the servers to handle more VMs. Each Fedora Core 5 VM ran on 256MB of RAM. Smaller, stripped-down images could be created and probably use less memory, thus more VMs without adding additional memory. It took much prep time in setting up the servers. Configuration files had to be set and the copying the base VM took awhile. Setup scripts would be written to help decrease this time.

Because the servers were centrally managed, changes to the environment for labs didn’t take too much effort. We setup the servers so the students didn’t have to spend time in organizing the design. Students needed a basic understanding of Linux commands to start their labs. We received input from the students and they mentioned irritation at having to SSH many times to get to their VMs; this also happened when getting programming files to each VM. One of the most noticeable variables that affected this design was the simultaneous running of TCP/UDP programs over the same NIC between servers. This lead to unpredictable WAN results on very small occasions, but would be prevalent if there were more groups trying to get their labs done the night before the project was due.

The laptop installation design would be able to handle any number of students because Xen environments are created on separate laptops. The students would need to have a higher understanding of Linux installations, virtualization, and console experience to be able to handle the setup of a Xen environment simulating a WAN. Again, help from a local Linux User Group could help initiate the process. Changes to the design wouldn’t require much effort in

...
between labs and the state of the virtualized laptop would remain constant because the OS is installed on the hard drive.

Hardware requirements would have to be set before the students start class. Most of the laptops have the required hardware as recommended by the university: more than a 40GB hard drive and CD/DVD drives. It is hard to determine the amount of memory that is sold with preconfigured laptops but a good size to accommodate the laptop design would be 1.5GB of RAM or greater. Another hardware factor in this design would be the effect on WAN results. The appendix provides some throughput variations investigated in the system’s modeling project and shows that throughput between VMs on the laptop experienced erratic numbers. Students can learn about WAN characteristics through their experiences with this design, but would have to configure their TCP or UDP programs to control throughput and not depend on the media between VMs.

Without thorough investigation of the LiveCD/DVD design, experiences have been limited. If a LiveCD/DVD image can be created, it would have to be specialized to simulate a WAN. VMs on the CD would have to be stripped down because CD media can only accommodate about 700MB and DVDs from four to eight GB. There are utilities to create LiveCD/DVD images and we are investigating the use of several utilities for its construction.

The 2 Xen servers design is a viable virtualized network. It has been used for two classes and with some modifications, should be able to handle one CS networking class. If more classes need to access this design, more equipment would need to be added. The laptop installation design, if carefully implemented, should be an enterprise approach to virtualized education. Although it requires more individual work on the student’s part, they have more flexibility to run their own networks and can come up with a variety of semester projects. The Xen LiveCD approach is currently under investigation and customization of a LiveCD is being worked on. We also intend to write an accompanying lab manual for CS networking courses using Xen virtualization.

The continued use of Xen as a tool for network programming would also depend on its strength in the marketplace and continued development from Xensource and Cambridge Laboratories.

7. REFERENCES

[2] Clemson’s eLearning with SkillPort, Clemson University Computing and Information Technology http://ccit.clemson.edu/services/training/elearning.php