A Virtualization with Xen Using an Exploration of Computer Network

Azham Hussain
School of Computing, Universiti Utara Malaysia, 06010 UUM Sintok, Kedah, Malaysia

Article Info

Article history:

Received Mar 19, 2025
Revised Apr 20, 2025
Accepted May 13, 2025

Keywords:

Virtualization
Xen
Network Programming
Education
Information Technology

ABSTRACT

As universities have security policies in place to safeguard the integrity of network infrastructure, computer science students who take a network programming or system administration class that includes a hands-on lab are usually not granted administrative privileges on lab computers. By implementing Virtual Machines (VM) technology such as VMware and Xen, the research community is starting to tackle this. The objective is to provide a virtualized environment so that learners have complete control over their operating system and can execute the required classroom duties. To date, most of the classroom trial deployments of VMs have been tailored to the classes of network security and system management. There may be beneficial advantages for hands-on laboratories that support networking ideas taught in lectures. A hands-on laboratory also gives students the chance to engage with' real-world' issues that involve creativity and resourcefulness. Students of computer science are introduced to networking either using the seven-layer OSI stack as the model or using a ' top-down ' strategy that begins with the implementation [a reference to Comer, Stevens, Kurose textbooks]. Virtualized designs must be able to provide an atmosphere to enable learners to operate their TCP / UDP programs and assess realistic network performance through WAN metrics such as throughput, drop rates, and packet delays to promote a deeper understanding of the content. This article uses Xen virtualization to accompany network programming courses to investigate three distinct technical designs. This article uses Xen virtualization to accompany network programming courses to investigate three distinct technical designs. Designs include a physical network lab configuration comparable to the job performed by other scientists, a method of installing a laptop and using a Live CD / DVD based on Xen. In two of the models, activities and outcomes based on a network programming class are shown and future research is suggested with Xen Live CD / DVD.

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

General Terms: Design, Experimentation, Performance.

Corresponding Author:

Azham Hussain,
School of Computing,
Universiti Utara Malaysia, 06010 UUM Sintok, Kedah, Malaysia
Email: azham.h@uum.edu.my

1. INTRODUCTION

Computer networking is packed with a variety of techniques, operating systems, and equipment that seeks to provide a reliable means of managing people, colleges and businesses ' computer communication needs. Networking education aims to create an atmosphere in which a student can learn such a wide range of data, but often faces the challenge of not only simulating a network environment for instructional reasons but also bringing together laboratories and events to enhance student retention and fulfill class goals. One way of setting up a laboratory atmosphere for networking courses from prior work experience beginning in 2001[1] would be to locate the amount of pcs in a classroom together with a switch and/or router. One challenge with this strategy is to provide a range of simultaneously running operating systems to enable learners to finish a growing and more complicated series of exercises on their pcs. One way to handle this would be to partition the hard drives into distinct boot choices for the operating system so learners could select which entry to boot for their present assignment. This, however, required lengthy reboot times during laboratory hours when modifications were needed.

Students now have many choices to learn from networking with the advent of new technologies and the World Wide Web. Schools can provide online learning alternatives such as eLearning from Clemson with Skill Port [2], a self-paced web-based courseware tool that helps learners learn about different topics. While this technology technique offers a strong means for a student to learn at their speed, it does not provide the complicated atmosphere that computer networks need to emulate. The Virtual Computing Lab (VCL) is used by students and teachers at North Carolina State University [24]. This mentioned service enables users to access separate on-campus computers operating systems and distinct apps remotely. The ns2 application [3], a tool that can simulate many networking protocols, can also be downloaded and run by any networking student. This is very helpful for studies and can significantly improve networking student learning, but the virtualization structure of this article does not cover this software.

Different (guest) operating system (OS) instances can operate simultaneously on a single physical computer thanks to Xen, a publicly accessible computer virtual machine monitor (VMM). It employs para-virtualization, in which an application conceptualization that differs slightly from the physical system is exposed via the VMM. Today, most computer laboratories in schools are linked to the infrastructure of a university network. Students have access to resources from the university and can go to further studies online. In order to handle different vehicle functions, computational resources in cars include hardware elements including consciousness, storage, machine cores, and the ECU. Balancing these technological assets has grown increasingly difficult due to the quick development and complexity of vehicle operations. The downside to this design is the university's necessary safety to preserve its networks and the constraints it places on learners in computer science who need administrative privileges to achieve duties such as: altering IP data on a network interface card, setting up connectivity routing tables, running packet data analysis programs, etc. This challenge is overcome by creating a computer network using virtualization technology and providing a resource for learners to perform administrative duties as learners have full control over their assigned operating system.

From an undergraduate view, if these administrative rights were present in computer laboratories, our personal experiences in network programming and system administration courses would have been further improved. Dating back to the mainframe architecture of IBM [25], virtualization technology has been around for many years. Nevertheless, in the last century, VMware [5] and Xen. Intel is now working on new technology that will give virtual machines (VMs) unfettered access to equipment I/O functions in an effort to enhance the I/O technique of emulation systems. For devices with this type of hardware support, we think the IDD overhead may be decreased or even eliminated.

[6] virtualization software have produced important progress in business and research fields. Virtualization software also offers many educational possibilities. Universities can use this technology to create study platforms and teachers can carry out a broad range of operations to achieve their classroom goals by offering a virtualized design to enable laboratory exercises. Because computers now need to be able to manage various operating systems simultaneously, networking, safety, and system management classes can incorporate a broad range of networking operations. We provide an overview of instructional methods using virtualization in the background chapter of this document and they are applied to education in networking.

While these methods concentrate on network subjects, safety, and system administration, there is an important programming element in all networking classes provided in Clemson's Computer Science (CS) department. Students of CS networking are brought to networking either using the seven-layer OSI stack as the model or using a' top-down' strategy starting with the application [23]. Students also learn the basics of networking and network protocols comparable to the above courses but also include programming projects. In this article, we concentrate on assessing technical models for network programming and performance analysis, which are not necessarily discovered in small- scale and hands-on student approaches in generic network courses. Three virtualized topologies are analyzed so that TCP / UDP programming and the study of WAN features fit into a paradigm required to complete a computer science networking course successfully.

This article introduces three technical virtualized models:

1) A centralized Xen topology on a physical network comparable to present study projects;
2) A virtualized laptop assembly design, and
3) The implementation of using Live CD / DVD technology as a networking education platform.

In their corresponding parts, each of the technical models is outlined. We spent a few semesters in networking courses using the first model and presenting our results from that experience. Another semester was spent as a project in a system modeling course on the laptop assembly technique and outcomes are also stated. At the end of the paper, we propose a design and use for the Live CD and finalize our results in the section concluding / future work. Sample projects and laboratories used in this document are included in the appendix.

2. BACKGROUND

Xen is an open-source instrument for virtualization created by Cambridge Laboratories [6,7]. In a broad sense, virtualization is the encapsulation of multiple operating systems running on the same physical computer simultaneously. Figure 1 is a virtualization diagram instance.

As in the case of Xen, called the hypervisor, the virtual machine monitor (VMM) manages the system resources between multiple OS instances. This sort of structure enables testing environments to be created and network topologies to be mapped. Virtual machines (VMs) are called instances of OS operating on top of a hypervisor. These machines can be configured with a modified Xen kernel, called a para-virtualized machine or set up as a fully-featured independent operating system, such as a fully virtualized machine, Windows. Using a para- virtualized Xen OS has many advantages, particularly better access to the I / O system and interaction between VMM and VM [7,8,9].

Full virtualization enables management of VMs such as Windows and Solaris by the Xen hypervisor. Full virtualization enables the independence of the OS kernel, but sometimes it has a big overhead [6].

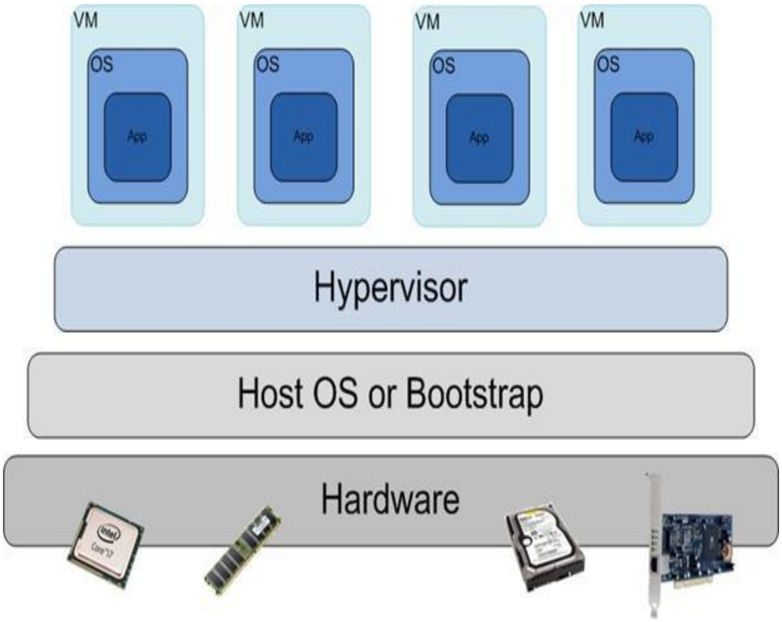

Figure 1. Virtualization Diagram

2.1 Virtualized Networking Education

Virtualization offers many instructional possibilities. For a variety of functions, educators can set up virtual networks for student labs. Programmers can test their apps before pressing their code into a' live' setting. Virtualization can also be a cost-effective strategy for hosting various operating systems, reducing software expenses and improving the use of CPUs [5]. Marist College utilizes the IBMz/VM virtualization hypervisor to provide a key service for a variety of courses, emails, websites, etc. [26] Users request a service from this VM technology without understanding the underlying structure; they think that they are using this service as if it were on their desktop. The partnership between Marist College and IBM will enable the technology to continue to be supported and future improvements to the VM scheme to be implemented. With the increasing popularity of virtualization over the past few years, especially VMware and Xen, there has also been an increase in virtualized environment publications for use in network, network security, and system management classes. Over the past year, this paper's physical designs have been researched and used in live settings, and its timing coincides with some of the studies presently being done at other universities. A few of these journals are discussed and how they use virtualization technology in their next course work.

User-Mode Linux (UML) is a technology for virtualization that is currently used for company and educational purposes [10]. In a computer security class, Walden at Toledo University introduced UML virtualization by dividing learners into organizations and making them compete on a virtual network by playing' Capture the Flag' [11]. In a competitive and high-rise setting, this is a very engaging and innovative way for learners to learn about computer and network security. It could also apply this sort of exercise to a variety of other exercises.

One of the market leaders and virtualization software manufacturers is VMware [5]. EMC purchased VMware and is anticipated to have this year's original public offer (IPO). Also used as instructional instruments are VMware products. We used VMware products from private experience [1] to teach introduction to ECPI network classes last year.

Students could begin Fedora or Windows pictures to use with their networking laboratories. This is given them with a distinctive chance to see differences in how Linux and Windows approach simultaneously running networking ideas on their computer. At the University of New Mexico, Butler, W. et al. created a virtualized environment for the network, safety, and database classes using VMware Workstation software operating various Windows, Windows, and Free BSD instances [12]. Students were able to learn a range of subjects from within these courses and the teachers were empowered to provide a range of exercises over the traditional hard drive partition strategy described in the introduction, thus saving time in class configuration and configuration.

The border at the Rochester Institute of Technology extends the approach of the University of New Mexico to students wishing to take part in laboratory exercises remotely using Microsoft's Remote Desktop Connection, Microsoft Terminal Services, VMware Workstation and Remote Laboratory Emulation System (RLES) [13]. Xen virtualization is of specific concern and the concentrate of assessment in this paper's virtualized models. Xen is an open-source, cost- effective approach to virtualization [6] and we are talking briefly about two journals using Xen in their instructional framework.

M. Alexander and J. A. Lee [15] uses a mixture of Xen and a LAMP-based Weblab application to design and use Weblab. Weblab was initially implemented as a VMware design; with their virtual machines running Linux CentOS operating systems, the authors then added Xen to the structure. Another Xen setup was used at the Electrical and Computer Engineering Department of Iowa State University [14].

T. Daniels and B. Anderson created a virtual machine grid called Xen World to manage a virtualized infrastructure for their courses in safety education. Their strategy was to offer a low-cost, flexible and tailor-made strategy to learners who control their own assigned VMs. Their original project included the use of thirty learners of one server handling VMs. Their current configuration in the publication now can run 300 VMs for a variety of computer engineering classes with the hope of one day supporting a 1000 VMs educational structure.

A kernel's source code compilation includes a CPU-intensive program that stresses the system CPU by calling several functions. 2.6.12 is the kernel version in use. Each of the above-virtualized settings must go through a life-cycle of design to achieve instructional goals. Due to safety issues, infrastructure design, technology interoperability, and classroom dynamics, changes to the setting are often needed. Another issue is the ease of OS updates to help the ongoing improvements in the releases of Linux. When technology and students ' needs are placed together in a manner that is cohesive and adaptable, a healthy virtualized design is encountered. Many of the design modifications will be produced over some time. From the point of perspective of the school in providing learners with these kinds of settings, teacher training time and design expenses are often variables that determine their execution.

3. MOTIVATION

In the background chapter, each of the virtualization methods is mainly oriented towards the network, network security, system management, or database classes. Many of the subjects in the above-mentioned classes are discussed in a computer science course at Clemson University from private experience in undergraduate computer science courses. However, one of the areas not necessarily found in them is the additional element required for computer science students to develop and run software programming labs and projects. This does not state that programming is not included in those courses, but likely not to the extent of someone who is majoring in computer science.

In the Clemson network infrastructure, computer science labs are integrated. Students in the first and second years have credit hours set up and are allocated to a laboratory to perform classroom exercises in them. This configuration is a great way to start programming courses. An issue may occur when a student reaches a CS networking or system administration class and these laboratories are used for additional training (at the moment of writing this document there were no lab hours needed for CS networking or system administration courses. Students have the chance to use these laboratories outside the classroom to program their projects). Students conduct their laboratories in these schools, but due to the university's security policies, there are no administrative rights on the computers. In studying networking concepts, having administrative privileges and controlling your operating system is crucial. Students on the computer science network learn subjects such as DNS, DHCP, NIS, and NFS, to name a few, but the primary elements of their teaching are network programming and performance analysis. Most of their out-of-class time is spent on a few tiny projects or a big semester programming socket apps for TCP / UDP.

For learners to obtain a better knowledge of networking by evaluating packets traveling over a physical media, virtualized technical designs need to be able to provide an atmosphere for learners to operate their TCP / UDP programs and assess realistic network performance via WAN metrics such as throughput, drop rates, and packet delays. We look at and evaluate three designs in the next section of the paper.

Before looking at the designs for Xen virtualization, we will define the scope of a CS networking class, what areas of learning we focus on in our Xen research, and what concerns we might have in a virtualized environment. A description of Clemson's CS networking class is shown below:' CP SC 360 Networks and Network Programming 3(3,0) Introduction to a fundamental computer network and network programming concepts. Themes include network programming, layered protocol architectures, local and wide area networks, concepts of internet and intranet work, safety. Socket level programming is introduced and used throughout the course. Prereq: 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 the 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 TCP dump, TCP trace, and MTRG.
- Enriching administrative skills -- provides an opportunity to do network administration and for students to set up and maintain their 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?

How does this virtualized technical design affect WAN characteristics like throughput, drop rates, and packet delays?

After each of the three technical models for inquiry, we will answer these questions. We look at using 2 Xen servers in our first model with 1 other device acting as a router. This virtualized design is the primary network to be evaluated as it has been used more extensively than the others over a longer period. The second design includes using a student laptop to install Xen. The final design is a technique suggested to use Live CD / DVD as a network programming device.

4. VIRTUALIZED NETWORK DESIGN – 2 XEN SERVERS

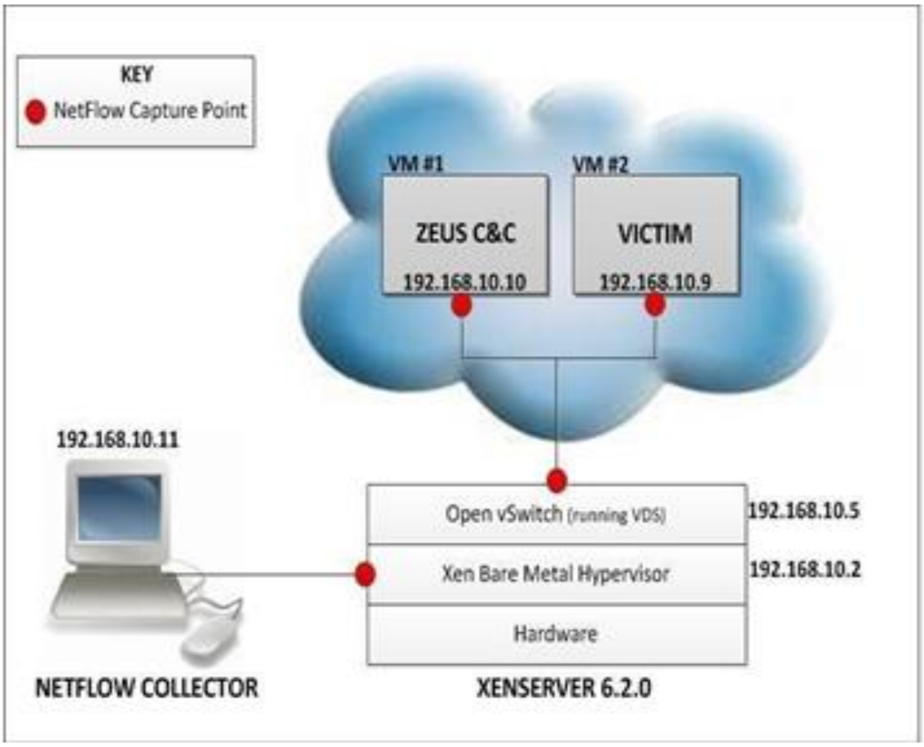

Figure 2. Virtualized Network Design – 2 Xen Servers

A. Topology

Figure 2 is a network description that was set up for a class of CS networking undergraduates and a class of internetworking graduates. This layout was used over a two-semester period, first with the graduate course and then configuration for the graduate course.

This topology contains three physical devices: 1 Dell server and 2 Sun servers. The Dell machine, named borg, had 3 NICs set up to provide connectivity to the Clemson network, but also designed to enable the Xen servers to have two private subnets, csvm1 and csvm2.

Each of the Xen servers consisted of 148 Sun Opteron computers, each running at 2.2 GHz, 4 GB RAM, and two 250 GB hard drives. Borg has a FreeBSD operating system with a DummyNet [16] network emulator that enables a user to define WAN features over a specified NIC. The Fedora Core 5 hypervisor operating system (VMM) and 10 distinct VMs were used by each of the Xen servers. Overall, the learners were able to use 20 VM. Each VM was set to run on a memory of 256 MB and the hard drive's VM space was 5 GB.

B. Usage

For the graduate course, a pair of VMs, one on each Xen server, were assigned to each group in the class so that the VMs would be on different subnets. There were not a lot of learners in the school, so the number of organizations was tiny. The graduate students were able to use the Xen network during the semester for the projects allocated to them in school. The Xen network was set up the next semester for undergraduate school. This class made extensive use of the virtualized network, primarily for laboratories developed for this design, but also made accessible for use with the semester project they had to do. This class ranged from 3 to 4 learners each with many learners and group numbers. Again, each group had a pair of VMs, each one on different Xen servers.

Borg and both Xen servers were allocated to each group in both categories. These machines were accessed by students via SSH and shelled as root users into their corresponding VMs. Students then developed their directories of users and conducted duties as needed. Dummy Net was used if the WAN features required to be changed by any lab or project.

C. Impressions

We first explored this design by attempting to use Fedora Core (FC) 4 as the operating system, but at that moment it seemed difficult to set up matching versions of an FC kernel and Xen release that would work with some measure of achievement. Then we moved to Fedora Core5 and were able to operate from a stable setting. Each VM was set up on 5 GB of hard drive space and it took a lot of time to set up 10 VMs on one server. Although we did not notice any problems with the graduate course using this design, there were some remarks from learners attempting to access their corresponding VMs. Their remarks involved running various SSH commands just to get to their own VM. A student would be required to shell Borg, a Xen server, and then their VM. In the undergraduate course, this was also obvious. Another observation was to copy back and forth their files from the Clemson network and their VMs for testing their programs. In the undergraduate course, we developed laboratories for the learners to get them started on their VMs and how to generate SSH, copy/move files, generate accounts. It seemed that these were going well.

When our laboratories were then concentrated on using instructions such as TCP dump to comprehend packet kinds that go through the NIC and then advance from there. There are two sample laboratories in the appendix. Labs seemed to be going well and the virtualized network offers a range of assignments that learners can complete. Setting up the Xen VMM for bridging and NAT for network requirements adds versatility to future laboratory workflows. We will realize that if there are many groups simultaneously using the 2 Xen servers, testing their programs, and then attempting to get excellent outcomes for WAN features, an issue will be generated. Each server uses a gigabit NIC, but if you have a few groups competing over the same media, there may be obvious incorrect throughput, drop rates, or packet delays. If the activity is interactive and designed to create this (for which some projects are designed), this sort of network is beneficial, but if learners expect drop rates of .04 percent as we set up Dummy Net with this environment, learners may see much greater packet drop levels and wonder what's wrong with their apps. Probably the most significant factor in socket programming is accurate results on WAN features, enabling accurate prediction of traffic pattern outcomes through TCP and UDP. Figure 3 portrays VM resource utilization vs. performance.

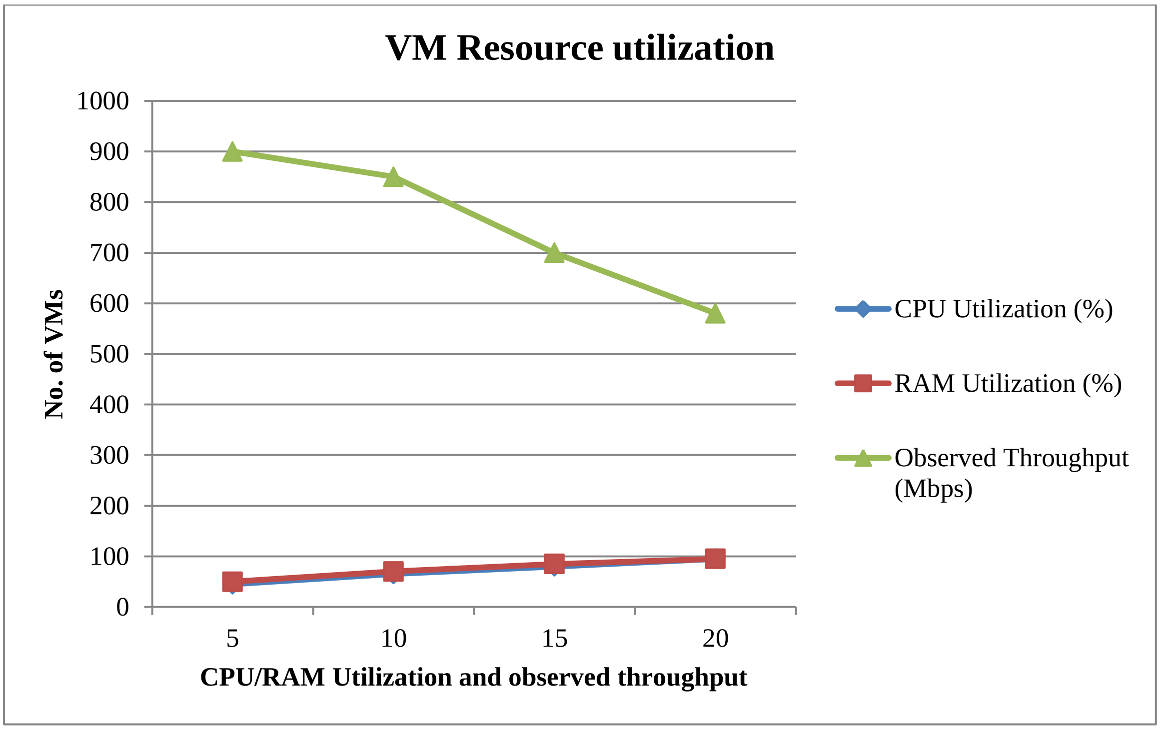

Figure 3. VM Resource utilization vs. Performance

These values help you show how increasing VM count impacts host system resources and throughput, reinforcing your recommendation for scalable server capacity.

D. 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 the 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

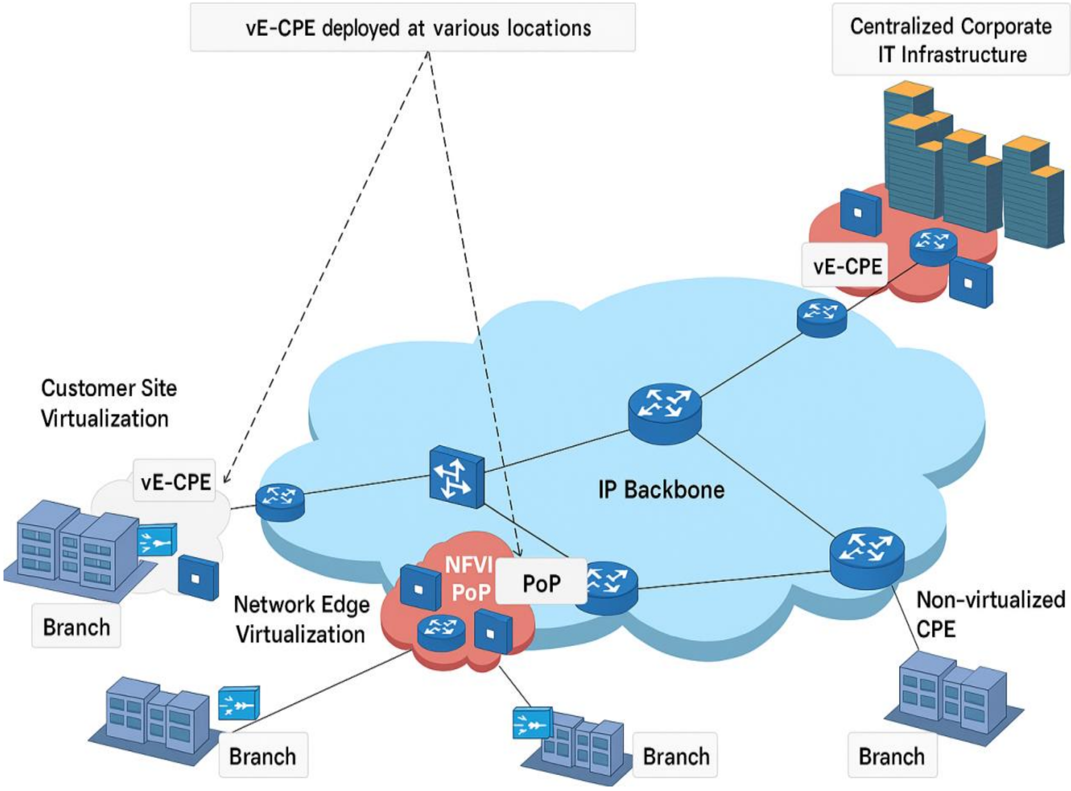

Figure 4. Virtualized Network Design – Laptop Installation

A. Topology

Figure 4 is a description of a single laptop of a virtualized Xen network. On the same machine, the Xen hypervisor and 5 VMs are running. This topology was installed with 2 GB of memory and 80 GB of the hard drive on a Dell Latitude 620, dual-core 2.0 GHz. To boot to Windows XP or Fedora Core 6, which was Xen's OS, the hard drive was partitioned. Each VM was set to have a hard disk space of 5 GB and run on a RAM of 256 MB. XEN bridging had to be used to build this topology. The laptop comes with one NIC, but for each virtual bridge shown in the figure, a PCMCIA network card was purchased and connected to this topology which permitted the use of two NIC's. Iperf was mounted on each VM. Iperf [18] is a NIC bandwidth measurement tool and can also display packet delay and packet loss stats. The Linux kernel 2.6 has a network tool, Netem [19], which can be used in laboratories or projects for WAN features and configuration to operate when needed. Item was running on a VM that had two virtual NICs and was configured to operate between two subnets as a router.

B. Usage

This setup was not used in a networking class but was conducted as a project for the modeling project of a graduate system exploring the use of a Xen laptop model as a networking student instructional instrument. This was a personal project designed to coincide with the 2 Xen servers' first design. The purpose of this was to see if it was possible to install a virtual environment on one laptop and to what extent it could be used in a CS networking course by undergraduate learners. Fedora, CentOS, and RHEL provide a virt-install Xen utility that makes installing VMs simpler than designing the 2 Xen servers.We didn't have to build distinct groups, but for this design, we were able to use one account. It's simple to build and handle users and organizations on one laptop. Students would set up their lab and project settings. Unlike the 2 Xen server design, this is an autonomous strategy. The only restriction is left to the student who utilizes that laptop in terms of network management.

C. Impressions

Using the virt-install utility provided by Fedora Core 6, it was not difficult to set up the VMs, although it was very time-consuming. Xen's bridging identified the PCMCIA network card automatically and set up two distinct bridges to operate on the VMM (not using the wireless NIC for this model). For their corresponding bridge, each VMs NIC had to be configured, but that was only a shift in a few lines of a configuration file. Packet forwarding had to be switched on to get routing to operate on the VM that acted as a router. To be able to send packets via the VM router, each VM required a routing table entry for the other subnet. The first two laboratories published for this class might be general directions for setting up Xen to get to this topology on their laptops.

The level of knowledge that an undergraduate student has installed and used Linux and Xen is one issue that would likely emerge from this design. The class intends to concentrate on the goals of network programming and not be diverted by taking excessive time to prepare their laptop. While it may be too distracting to install and use Xen as a great learning instrument. This layout is very flexible and is suitable for many laboratories and prospective projects. The appendix contains a few experiments from this design.

Another observation observed (and also shown in the appendix outcomes) was some very interesting WAN behaviors among VMs. Significant distinctions between para-virtualized computers and fully virtualized machines showed throughput experiments between two VMs using Iperf. This variability could damage learners attempting to operate TCP / UDP performance programs. It is a nice environment from which to operate programs on virtualized computers researching network performance, but not on any project seeking anticipated outcomes in socket programming. Published papers explain the distinction between para- and fully virtualized machines, but this goes beyond this investigation's grasp. Item conducted the router tests accordingly.

The variability in student laptop hardware will probably be one of the most challenging obstacles to this design. There will be apparent variations in hardware and the issues connected with installing on them a Linux OS and Xen. To work with this design, students would likely need at least 2 GB of RAM and 60 GB or more room on their hard drive.

D. Recommendations

- Set up 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 –Live CD/DVD/USB

We had no time and couldn't use a LiveCD in a networking class. We see this technology as a feasible solution to the assembly models of the 2 Xen servers and laptops. We are discussing the technology because it requires no setup and can be used as a manner to manage a virtualized network more easily. Live CD's, or the term' live,' allows a user to insert a CD, DVD, USB, or another bootable device into a computer and power the machine to an OS — all without a hard drive being used. One of the most well-known LiveCDs currently in use in Knoppix [20]. Xensource [6] has a Xen Live CD accessible, and Xenoppix [21] has been created in Japan. Both were experimented with and from our knowledge, we discover that a virtualized network, albeit a restricted one, could be used in a model.

To use a Live CD in a CS networking course, a Xen hypervisor with at least two VMs must be adapted to operate a network for network programming projects. The Xen Source Live CD can launch two different VMs, but it doesn't have all the capacities we want for programming and WAN management reasons in a virtual machine. Now that Live CD's use is becoming more common, there are some choices. There are software instruments out there, for instance, Revisor, which is now accessible on Fedora 7 and comes from the Fedora Unity Project [22], which can assist build a tailored Live CD. Some constraints come with Live CD. Not necessarily the environment is stable. You lose everything you had in memory once you power the laptop down. For instance, you will need to discover a way to save your files, by saving your information to a USB device. Because your CD-ROM drive is slower than memory, there are performance hits when you use a Live CD. The BIOS of your computer must also be able to identify your Live bootable machine.

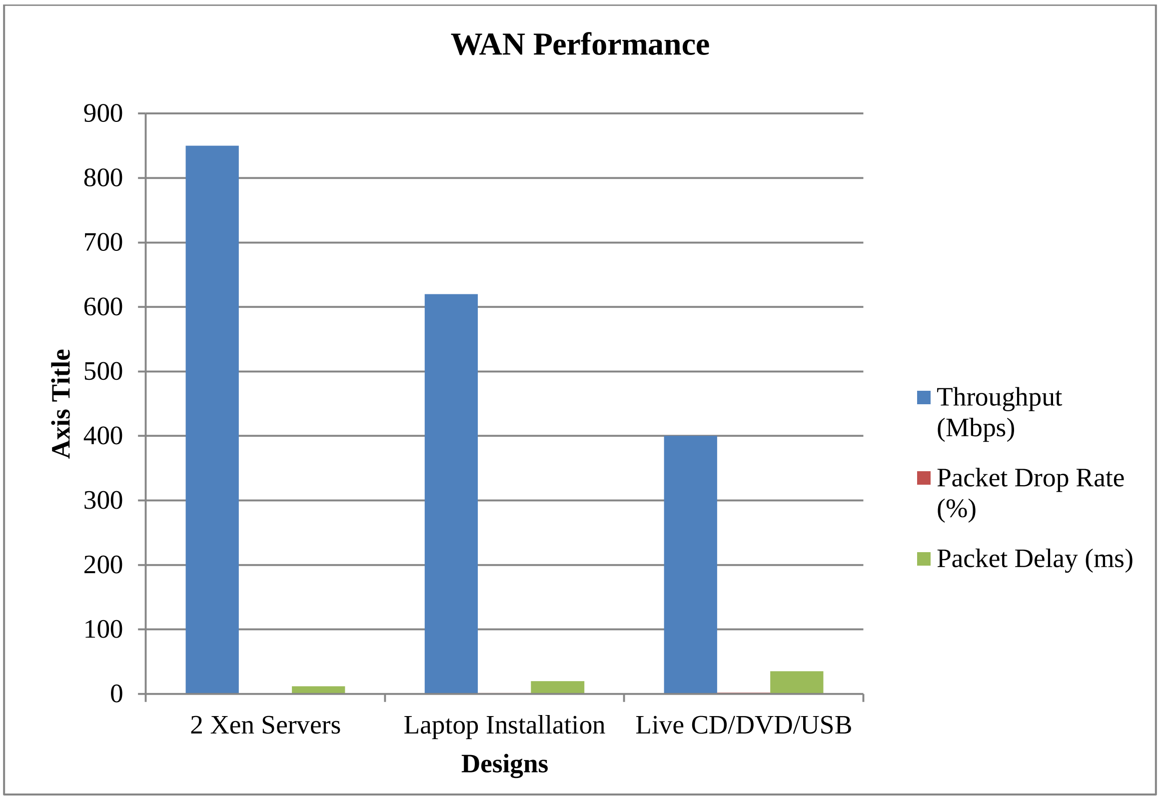

Figure 5. WAN performance metrics across designs

Figure 5 graph compares WAN performance metrics across the three virtualization models, showing the trade-offs in throughput, packet drops, and delays. It highlights the practical differences in network behavior encountered in different setups.

7. CONCLUSIONS / FUTURE WORK

A total of 20 VMs, 10 on each server, were managed by the 2 Xen server layout. This was in line with the number of learners that this design was used in each of the courses. Typically, group numbers had two or three members, tiny enough to manage laboratory tasks. If the amount of # # s rises, to manage more VMs, RAM would have to be added to the servers. Every Fedora Core 5 VM was running on RAM 256 MB. It is possible to create smaller, stripped-down pictures and possibly use less memory, so more VMs without adding more memory. It went a lot of time to set up the servers. It was necessary to set configuration files and it took some time to copy the base VM. To assist reduce this time, setup scripts would be written. Because the servers were centrally managed, it didn't take too much effort to change the atmosphere for laboratories. We set up the servers to prevent the students from spending time organizing the design. To begin their laboratories, students required a fundamental knowledge of Linux commands. We got feedback from the learners and they mentioned irritation that they had to have SSH many times to get to their VMs; this also occurred when each VM got programming files. The concurrent running of TCP / UDP programs over the same NIC between servers was one of the most noticeable factors affecting this design. This led to unpredictable results of WAN on very tiny occasions but it would be common if more organizations tried to get their laboratories completed the night before the project was due.

As Xen environments are developed on distinct laptops, the laptop assembly design would be able to manage any amount of learners. Students need to have a better knowledge of Linux systems, virtualization, and console experience to be able to manage the configuration of a WAN simulating Xen setting. Again, a local Linux User Group's assistance might assist start the process. Changes in design would not involve much effort between laboratories, and the state of the virtualized laptop would stay continuous due to the hard drive installation of the OS. It would be necessary to set hardware specifications before the learners begin school. As recommended by the university, most laptops have the required hardware: more than a 40 GB hard drive and CD / DVD drives. Determining the amount of memory sold with preconfigured laptops is difficult, but a good size would be 1.5 GB of RAM or higher to suit the laptop design. The effect on WAN outcomes would be another hardware factor in this design. The appendix offers some variants of the throughput investigated in the modeling project of the system and demonstrates that erroneous figures encountered throughput between VMs on the laptop. Students can learn about the features of WAN through their experiences with this model, but they need to configure their TCP or UDP programs to regulate throughput and not rely on media between VMs.

Experiences were restricted without a thorough inquiry of the Live CD / DVD design. If you can create a Live CD / DVD picture, you need to specialize in simulating a WAN. VMs on the CD should be stripped because only about 700 MB of CD media and DVDs from four to eight GB can accommodate. To generate Live CD / DVD pictures, there are utilities and we are researching the use of multiple utilities for their building. The design of the two Xen servers is a feasible virtualized network. It was used for two courses and should be able to manage one CS networking class with some changes. Further facilities would need to be added if more courses need to access this design. If closely enforced, the laptop assembly design should be an entrepreneurial approach to virtualized schooling. While it requires more individual work on the part of the student, they are more flexible in running their networks and can develop a variety of semester projects. The Xen LiveCD method is being investigated and a LiveCD is being worked on to customize it. We also plan to use Xen virtualization to write an associated laboratory guide for CS networking lessons. Xen's ongoing use as a network programming tool would also rely on its marketplace strength and XenSource and Cambridge Laboratories ' ongoing growth.

REFERENCES

[1] Previous work experience teaching Networking Courses at ECPI College of Technology, Greenville, SC, 2001-2006.

[2] Clemson’s eLearning with SkillPort, Clemson University Computing and Information Technology http://ccit.clemson.edu/services/training/elearning.php Ns2 simulator. http://www.isi.edu/nsnam/ns/edu/.

[3] Example Computer Science Network ProgrammingClasssyllabus.http://people.clemson.edu/~jmarty/courses/Spring- 2007/CPSC360/syllabus.html.

[4] VMware, virtualization software. http://www.vmware.com/.

[5] Xen, virtualization software. http://www.xensource.com/.

[6] P.Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield. Xen and the art of virtualization. In 19th ACM Symposium on Operating Systems Principles, Oct 2003.

[7] A.Menon, J. R. Santos, Y. Turner, G. J. Janakiraman, and W. Zwaenepoel. Diagnosing performance overheads in the xen virtual machine environment. In First ACM/USENIX Conference on Virtual Execution Environments (VEE'05), June 2005.

[8] A.Menon, A. Cox, and W. Zwaenepoel. Optimizing network virtualization in xen. In USENIX Conference, June 2006.

[9] User-Mode Linux, virtualization software. http://user-modelinux.sourceforge.net/.

[10] J. Walden. A real-time information warfare exercise on a virtual network. In Proceedings of the 36th SIGCSE technical symposium on Computer science education (SIGCSE ’05) (St. Louis, Missouri, February 23-27, 2005).ACM Press, New York, NY, 2005, 86-90.

[11] W.Buller, S. Burd, and A. Seazzu. Virtual machines – an idea whose time has returned: application to network, security, and database courses. In Proceedings of the SIGCSE technical symposium on computer science education (SIGCSE’06)(Houston, TX, March 1-5, 2006). ACM Press, New York, NY, 2006, 102-106.

[12] C.Border. The development and deployment of a multiuser,remote access virtualization system for networking, security, and system administration classes. In Proceedings of the 38th SIGCSE technical symposium on Computer science education (SIGCSE ’07) (Covington, KY, March 7- 10, 2007). ACM Press, New York, NY, 2007, 576-580.

[13] B.Anderson, T. Daniels; Xen Worlds: Xen and the Art of Computer Engineering Education; In Proceedings of 2006 ASEE Annual Conference and Exposition; Chicago, IL; June 2006.

[14] M.Alexander, and J. A. Lee. A scalable xen and web-based networking course delivery platform. In Proceedings of the Second IASTED International Conference on Education and Technology (ICET ’06) Calgary, AB, Canada, July 2006.

[15] Clemson University 2006-2007 Courses of Instruction PDF file, Clemson University Registrars Office, p. 9.

[16] http://www.registrar.clemson.edu/publicat/catalo g/2006_PD F/133_146CME_ED.pdf

[17] Dummy Net, network emulator. http://info.iet.unipi.it/~luigi/ip_dummynet/.Iperf, network performance tool. http://dast.nlanr.net/Projects/Iperf/.

[18] Netem, network emulation utility. http://linuxnet. osdl.org/index.php/Netem/.

[19] Knoppix, bootable LiveCD. http://www.knoppix.org/.

[20] Xenoppix, bootable Xen--Knoppix liveCD. http://unit.aist.go.jp/itri/knoppix/xen/index-en.html.

[21] Revisor, LiveCD application. http://fedoraunity.org/.

[22] Kurose, J. F., and Ross, K. W. (2004). Computer Networking: A Top-Down Approach Featuring the Internet. Boston, MA: Addison Wesley.

[23] North Carolina State’s Virtual Computing Lab (VCL). http://vcl.ncsu.edu/.

[24] Adair, R. J., Bayles, R. U., Comeau, L. W., and Creasy, R. J. A Virtual Machine for the 360/40. IBM Corp., Cambridge Scientific Center, Report No. 320- 2007, (May 1966).

[25] Marist College virtualization framework, z/VMfromIBM. http://www.marist.edu/it/clientservices/datacenter/vm.html/.
[26] Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., ... & Warfield, A. (2003). Xen and the art of virtualization. ACM SIGOPS operating systems review, 37(5), 164-177.
[27] Zhang, Y., Bestavros, A., Guirguis, M., Matta, I., & West, R. (2005, June). Friendly virtual machines: leveraging a feedback-control model for application adaptation. In Proceedings of the 1st ACM/USENIX international conference on Virtual execution environments (pp. 2-12).