William Bonanno

Jonathan VanFossen

COMT 21100

27 April 2015

Final Project Report

To become successful in whatever career one chooses, one must seek out extra learning opportunities that are designed to enhance or build upon the knowledge that he or she already possesses. Therefore, I chose to participate in the Kent State Honors College throughout my educational career. Per the Honors College's rules, each participating student must complete one honors experience per semester. For my most recent semester, the spring of 2015, I chose to turn a course called "LAN Troubleshooting" taught by Jonathan VanFossen into an honors experience.

Through the honors course, Mr. VanFossen and I developed a final project that would be thought-provoking and would provide invaluable hands-on experience that some students never get during their education. This class's expectations were created together by Mr. VanFossen and myself throughout the summer of 2014 via email communications. Said expectations included weekly journals and meetings, two progress reports, a final project, and a final report. The final project that Mr. VanFossen and I agreed upon was a fully-functioning virtualized small-office network. Three key steps were needed to begin the project: planning, setting up the operating systems within VirtualBox, and creating a virtual network on which all the systems could send and receive data.

Before I even started work on the final project, Mr. VanFossen and I spoke about the outcomes that we wanted to see with the project. At first there were some deliberations, but in the end, we decided upon the important challenges which I was to overcome. First, we decided that I was going to host my clients inside of a program called “VirtualBox,” as mentioned previously. Next, we decided upon the different operating systems that I was going to use: Windows 2008 Server, Windows 7, Windows 8.1, and Ubuntu. Alongside the major operating systems for the clients and server, I also chose to include DD-WRT running on a virtual Linux router to connect everything and act as a DHCP server. This planning stage of the project took roughly one to two weeks to complete.

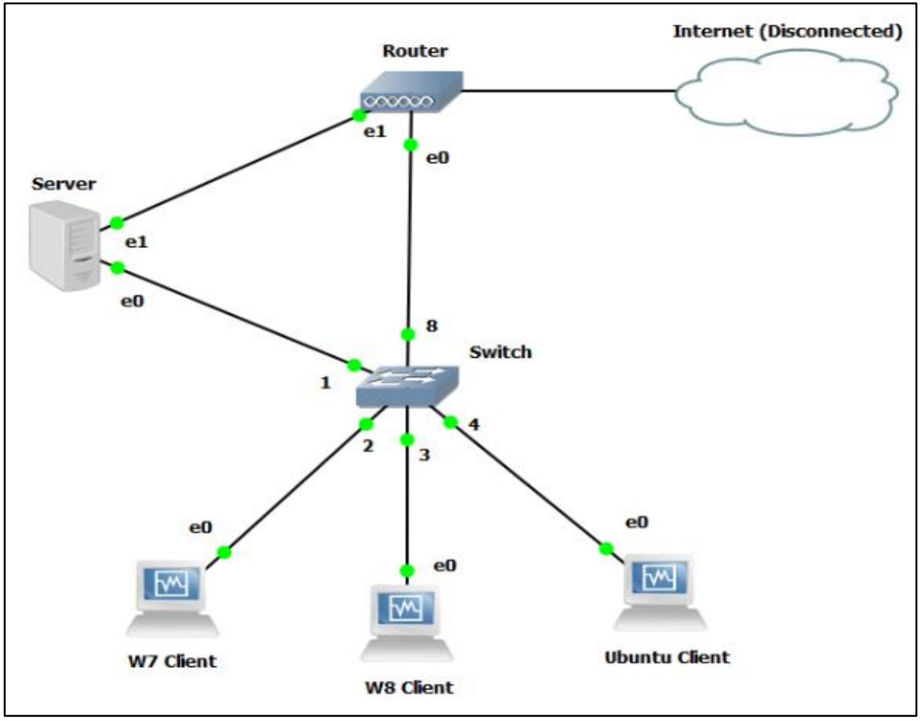
Once we were finished completing the planning stage, it was up to me to get all the operating systems up and running before I could even think about connecting them together. Because of KSU’s policy on piracy, I could not simply steal the Windows operating systems that I chose to use for the project. Instead, I was able to utilize DreamSpark, an opportunity provided by KSU and Microsoft for students to acquire legal versions of different Microsoft products. Through DreamSpark’s website, www.DreamSpark.com, I was able to acquire all of the Microsoft operating systems that were previously mentioned. Ubuntu, a popular version of Linux, is free to the public; therefore, I did not need to go through an activation process similar to the one I went through with Windows.

Now that I had those operating systems out of the way, I needed only to find an open-source router operating system. For the aforementioned OS, I decided upon DD-WRT. From personal experience, I knew that DD-WRT was a great firmware that is available for free. Its ease of use and great features is one of the reasons why it is such a popular open-source operating system in the router-modding community (Nawrocki).

After I acquired the operating systems, I then, of course, had to install them to my virtual PCs. I chose to dedicate 20GB of my SSD to each of the Windows systems and to the Ubuntu system. Alongside this decision, I also allocated 1-2 GB of RAM to each system. For my virtual router, I did not have to dedicate these great amounts of volatile and non-volatile memory in order for it to run. These settings can be observed in the beginning of the attached project video (0:04-0:28).

Upon completion of the actual installation of the operating systems, I needed to work on interconnecting my virtual systems into a small network. This was achieved by logically mapping the components of the network inside of GNS3, a program used to design virtual networks. The finalized network topology included redundancy (see figure 1) and prioritization for the server's traffic (4:29-5:23). These features were included in the network to ensure that faulty hardware and network congestion during times of stress may be overcome. In the project video, I even simulate a hardware fault to demonstrate that the network could compensate (5:35-6:20).

Figure 1. Finalized GNS3 Network Topology



This figure is the final network topology taken directly from a screen capture inside of GNS3. Notice the redundant connection running from the server to the router that would help in case of hardware fault.

Once I completed my work inside of GNS3, I started setting up the file server. Since I had never worked with Server 2008 R2 before, I found this process new and interesting. For the file server, I chose to make a directory on the server's desktop that would be shared on the network (1:41-1:43). I also selected the SMB protocol for my file server because SMB is the preferred method for network sharing on the Windows platform ("AFP vs. SMB and NFS"). While I do have one Linux client on the network, I figured the benefits of the SMB protocol on the Windows machines would outweigh the trouble of configuring the SMB share on just one Linux client.

Upon configuration of the file server, I needed to set up different user accounts to access it. Instead of tweaking with an active directory, I chose to create several user accounts on the server machine itself. My decision to do this was based on the assumption that there would be only a finite number of users on the virtual network; therefore, policy creation and modification would be rather simple to manage. I figured messing with an active directory would be counterproductive in this case.

Since I decided to rely on different user accounts, all that was needed to configure their different permissions was a couple of tweaks to their user accounts (1:47-1:59). These settings were found directly inside of Windows' file server management console. When creating the different accounts, I made sure that the various users had different permissions, such as read only and directory listing only. I then tested and saved the clients' different configurations on their respective machines (2:23-3:19).

While the permission and file server configuration were important parts of the project, I found the analysis of packets on the virtual network the most interesting part. I managed to capture said packets through a network diagnostic program that I installed on the server called

WireShark. Once it was installed and configured, WireShark let me view various types of packets running to and from the server itself (3:32-4:17). Valuable information can be found through analyzing packets such as incomplete packets or redundant information traveling over the network. These types of packets must be kept to a minimum to reduce network congestion (Franchitti).

Through this final project, I have learned the basics of managing a small-office network. This is because my final result appeared quite similar to an actual small-office network that businesses may have in place. As a result of this final project, alongside the required readings and assignments, I have been able to hone my skills in network administration through Kent State’s Honors College.

In conclusion, I am certain this project has been a success. It has taught me several things that are crucial to network administration such as policy creation and management, how to configure an SMB file server, and the basics of virtualization. These aforementioned skills will prove to be invaluable later in my career. As I initially stated, further education can aid one to succeed in whatever field he or she participates in. Personally, I am glad that I had this honors experience not only to enhance my computer networking skills, but also to form a lasting bond with my favorite professor, Mr. VanFossen.

Work Cited

“AFP vs. SMB and NFS File Sharing for Network Clients.” *Helios*. Helios Software GmbH, n.d. Web. 28 Apr. 2015.

Nawrocki, Matt. "Free DD-WRT Network Router Firmware." *TechRepublic*. N.p., 15 Aug. 2013. Web. 29 Apr. 2015.

Franchitti, Jean-Claude. *Data Communication & Networks G22.2262-001*. New York: New York University, Fall 2012. PowerPoint Presentation.