Spring 2016

Home Networking with Enterprise Equipment

Alex Lowers
all77@zips.uakron.edu

Home Networking with Enterprise Equipment

Alex Lowers

Project Name:

1. Home Networking with Enterprise Equipment

Team Member:

1. Alex Lowers

Project Description

1. Using enterprise layer 2 and layer 3 switches, a media server will be connected to a home network. Music, movies, and video games will be streamed and speeds will be benchmarked on both wired and wireless connections between the server and clients.

Equipment:

1. Windows 10 computer, the server

2. Linux computer, the client

3. Cisco Catalyst 2950 layer 2 switch

4. Cisco Catalyst 3550 layer 3 switch

5. TP-Link TL-WA801ND Wireless access point

6. Cat 5e cables, both straight-through and crossover

7. Crimp tool, cable tester, and any other tools necessary to build Ethernet cables

Location of Work:

1. Home

Detailed Objectives:

1. Research

    a. Software that readily streams media between Linux and Windows will be found.

    b. FTP software to allow file transfer between the client and server for offline viewing will also be found.

    c. Media to stream for quality testing will be determined (testing will require music files, video files, and video games to benchmark).

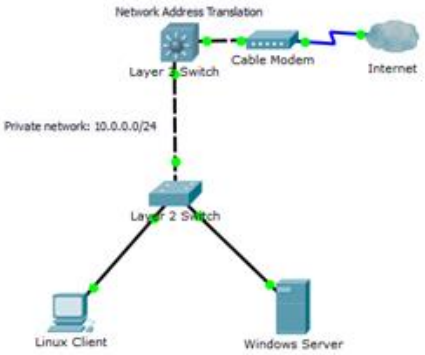
d. Best methods to benchmark performance of each media type will be determined.

e. Methods of accessing the server over the internet will be researched.

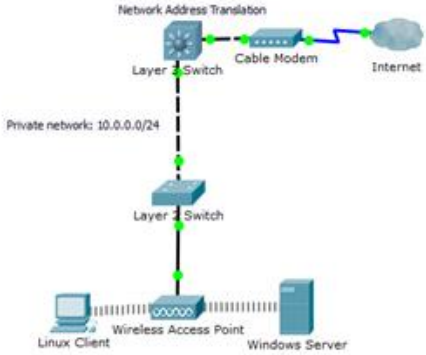
i. Ideally the equipment can support it using network address translation and port forwarding. Otherwise third party software may be required.

2. Design

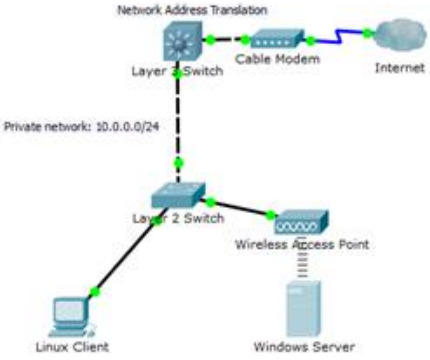
a. Here are some mockups of what the network would look like made in Packet Tracer:



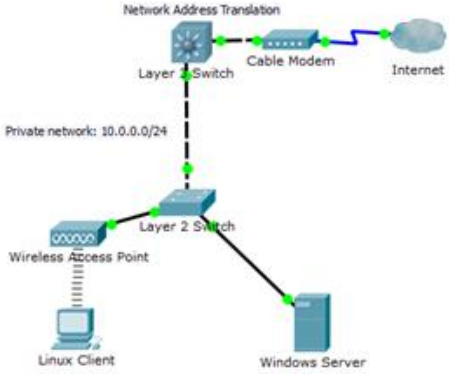
Wired Topology – both server and client are connected to the network via Cat 5e cable



Wireless Topology – both server and client are connected to the network wirelessly



Wired/Wireless Topology – the client is connected to the network via Cat 5e cable and the server is connected wirelessly



Wireless/Wired Topology – the client is connected to the network wirelessly and the server is connected via Cat 5e cable

3. Implementation

a. The parts have already been purchased.

b. Setup network infrastructure

i. The switches, mounted together on a small rack, will be connected with multiple crossover cables to form a large Etherchannel between them.

ii. A wireless access point will be connected to the layer 2 switch and a cable modem will be plugged into the layer three switch.

iii. Then the devices will be configured for LAN and internet connectivity.

- This will require at least layer three addressing, DHCP, and NAT to be setup on the layer three switch.
- Layer three addressing will be based off the 10.0.0.0/24 range. This range was chosen for its simplicity and the ability to expand to further subnets if needed.
- The range will be broken into four subnets and four VLANs will also be created. The subnets will be utilized as follows:
    - 10.0.0.0/26 - Server network
    - 10.0.0.64/26 - Client network
    - 10.0.0.128/26 - Guest network
    - 10.0.0.192/26 - Management VLAN
- Inter-VLAN routing will be implemented at the layer 3 switch so the Server and Client VLANs can communicate.
- For security purposes, the other two VLANs will only be able to access the other devices in their respective VLAN and the internet. This will be accomplished using access control lists.
- These subnets will be translated to a public IP address at the layer 3 switch when accessing the internet.

c. Server setup

i. The operating system is already installed.

ii. Any software deemed necessary after research conducted above will be installed and tested.

iii. The server will be attached to the network either wirelessly or via Ethernet cable, depending on what stage of testing is happening.

d. Client setup

i. The client operating system is already installed.

ii. Any software deemed necessary after research conducted above will be installed and tested.

iii. The client will be attached to the network either wirelessly or via Ethernet cable, depending on what stage of testing is happening.

4. Testing

a. Music, video, and games will be selected for testing purposes.

b. Performance will be measured on the server based on factors such as sound quality, framerate, and other factors found in research to establish a baseline.

c. Using each network configuration above, the same media will be streamed to the client and the same metrics will be measured.

d. Lastly, using an outside internet connection such as a cellular phone as a hotspot or another house, a connection will be established remotely to the server and media will be transferred from there to a local machine. Download speeds will be benchmarked as well.

5. Documentation

a. The network setup, complete with IP addressing scheme, topology, etc. will be documented and diagramed.

b. The server software installation will be documented.

c. The client software installation will be documented.

d. Test results will be organized into tables and graphs.

e. The results will then be analyzed. The topologies will be compared to find the strengths and weakness of each.

f. Lastly, future improvements that could be implemented will be noted.

Time Estimates

Research	Design	Implementation	Testing	Documentation	Total
15	5	30	30	25	105

Project Analysis

The focus of this project was leveraging enterprise equipment and networking techniques to streamline media sharing within a home network. Using several software tools, streaming can be accomplished both within the network, using the CIFS file system and VLC media player, or over the internet using Plex. Additionally, the program Steam allows a powerful gaming computer to run a game and stream it over the LAN to another, much less powerful computer, making living room gaming a breeze. The server and client were separated into separate VLANs. Because of the limitations of the Catalyst 3550, only the server VLAN has internet access, however this did not prove problematic for the scope of this project. As long as the server has internet access, online games can still be streamed to the client and remote access of media files can still be accomplished.

Problems arose with each piece of software, despite its ultimate success. The creators of Steam OS, Valve, have their own Debian repository used by their operating system named Alchemist. The software available in this repository is very limited. Specifically, the packages needed for this project, *cifs-utils* and VLC media player, were not in the Alchemist repository. Adding the Wheezy repository to /etc/apt/sources.list allowed those programs to be downloaded. VLC media player also required that the repository be specified in the *apt-get* command.

Despite its ubiquity in the gaming space, Steam also posed some challenges for this project. The in-home streaming option uses UDP broadcasts to find other Steam-enabled computers in the LAN and initialize connections between the computers. Because the client and server were on separate networks, these broadcasts were not getting through. Luckily, Cisco IOS allows forwarding of UDP packets to certain IP addresses. The command *ip forward-protocol udp 27036* allows the switch to forward broadcast packets that have the destination port of UDP 27036 outside the broadcast domain and *ip helper-address 10.0.0.2* specifies the address to send

the broadcasts to. Once these were entered, as the client sent broadcasts to find other Steam computers, L3 forwarded those packets to the server. The server then responded and was able to start communication and streaming with the client across different subnets.

Plex media server was very straight forward to setup. The only difficulty came from the fact that the phone application costed $4.99 to unlock, despite being advertised as free on the Google Play store. Despite the cost, or possibly because of it, Plex's installation was relatively hassle-free and works surprisingly well out-of-the-box.

Network configuration ran into some trouble as well. Outside of testing periods, in order to ensure other internet users in the house were unaffected by any network tests, the existing wireless LAN setup on the Netgear router was enabled. However, late in the project's development, the router was connected to L3 and DHCP functionality was moved from the WNR2000 to L3. After this migration, users on the wireless network experienced loss of internet connectivity. After some ping tests were performed, it was discovered that the users had access to the internet but DNS was not working. The DHCP pools were set to use the default router, L3, as the DNS server. However, L3 had ip domain-lookup disabled. This was remedied by changing the DNS server of the DHCP pools to 8.8.8.8, one of Google's DNS servers. After refreshing their IP addresses using *ipconfig /release* and *ipconfig /renew*, the computers on the network were able to browse the internet normally.

One last issue that arose was the fact that the TP-Link WAPs only support traffic from one LAN at a time, so the fully wireless topology would need a second WAP if the client and server were to stay on separate VLANs. The WAP was easy to order online, but this was an additional unexpected cost that delayed testing slightly.

The project scope was adhered to well with a few modifications. FTP was not used to

transfer files over the LAN or internet because it was not needed. CIFS easily allowed copying files manually for offline viewing and Plex has device storage for offline viewing available as a premium feature. Also, the necessity of the Netgear router was unexpected, but did not affect the scope of the project, as explained above. Network configuration otherwise stayed very close to the original design and the client and server behaved exactly as expected with regards to network capabilities.

From the results of this project, further improvements and adjustments could be made. The use of enterprise equipment allows experimentation with network monitoring software. Also, as testing shows, using wired connections is extremely important and Ethernet wiring within the testing space could be greatly improved by running cat 5e or even cat 6 cable within the walls of the house. Lastly, a more robust device such as a 2600 series router could be added to the network to replace the Netgear WNR2000.

Overall, research, configuration, and testing was quicker than initially predicted but documentation time far exceeded expectations. Predictions and actual time used can be seen in the table below:

	Research	Design	Implementation	Testing	Documentation	Total
Predicted	15	5	30	30	25	105
Actual	3.5	2.5	12	21	43	82
Difference	-11.5	-2.5	-18	-9	18	-23

Initial research was concluded much faster than expected. Solutions to the questions asked by this project at the beginning were easy to find on the internet. Also, some research was conducted during the implementation that did not count towards research hours as it was used to solve specific problems in the installation (for instance, the Steam broadcast issue outlined above). Design and implementation overall went very smoothly and fell far under predictions.

Testing also fell short of predictions. Once the network was setup, it was relatively easy to gather data for the project. The lion's share of effort went to documentation. Collecting screenshots, explaining implementation steps and commands, and analyzing the data took almost twice as long as first predicted. That said, the project as a whole took about 20% quicker than anticipated to complete.

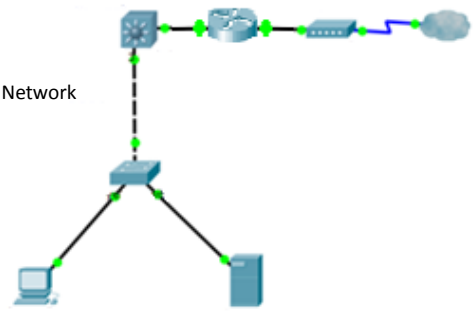
Project Description

Installation and Configuration

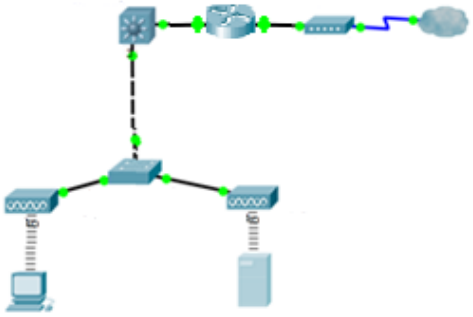
Network Design

Topologies

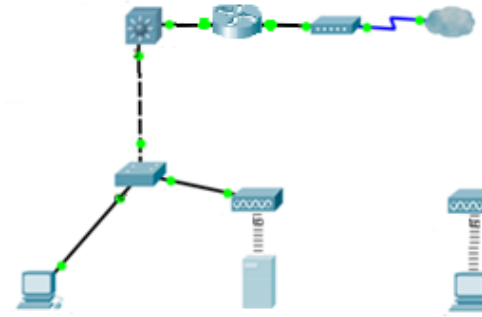
10.0.0.0/24 Network



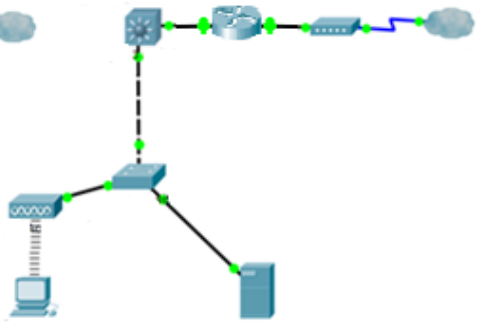
Wired Topology – both server and client are connected to the network via Cat 5e cable



Wireless Topology – both server and client are connected to the network wirelessly



Wired/Wireless Topology – the client is connected to the network via Cat 5e cable and the server is connected wirelessly



Wireless/Wired Topology – the client is connected to the network wirelessly and the server is connected via Cat 5e cable

Legend			
	Steamos (Linux client)		L3 (Catalyst 3550)
	Alex-PC (Windows 10 server)		Netgear WNR2000
	L2 (Catalyst 2950)		Cable Modem
	TP-Link TL-WA801ND WAP		Internet

IP Addressing:

VLAN 10 SERVER: 10.0.0.0/26

Hostname	**IP Address**
Alex-PC (server)	10.0.0.2
L3 (gateway address)	10.0.0.1
TP-Link (WAP)	10.0.0.3
NAT Router	10.0.0.4
Any (DHCP scope)	10.0.0.6 - 63

VLAN 20 CLIENT: 10.0.0.64/26

Hostname	**IP Address**
Steamos (client)	DHCP assigned
L3 (gateway address)	10.0.0.65
TP-Link (WAP)	10.0.0.66
Any (DHCP scope)	10.0.0.70 - 127

VLAN 30 GUEST: 10.0.0.128/26

Hostname	**IP Address**
L3 (gateway address)	10.0.0.129
Any (DHCP scope)	10.0.0.134 - 192

VLAN 40 MANAGEMENT: 10.0.0.192/26

Hostname	**IP Address**
L3	10.0.0.193
L2	10.0.0.194

Network Configuration

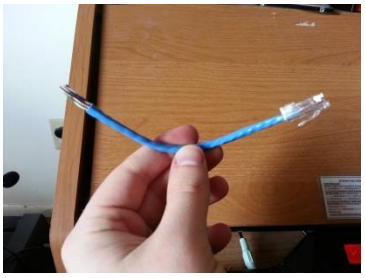
Before any software was installed, the network was setup. The switches needed to be connected and configured. Addressing and VLANs were setup, and then the server and client were connected to the network, as detailed below. The VLANs were configured such that the server, client, and any other guests on the network were segregated. Also, management traffic had its own dedicated VLAN.

Initial configuration was done using the PuTTY terminal program. Communication was provided by via a USB to serial adapter that connected into a console cable plugged into the back of the switches. VTY lines were configured on the switches to allow telnet access to the CLI for future configurations. The console, vty, and enable secret passwords were all set to "cisco" for the purposes of the project; however a more secure password would be needed after testing. Also, the command *logging synchronous* was entered on the console line to prevent debug messages from interrupting commands being entered.

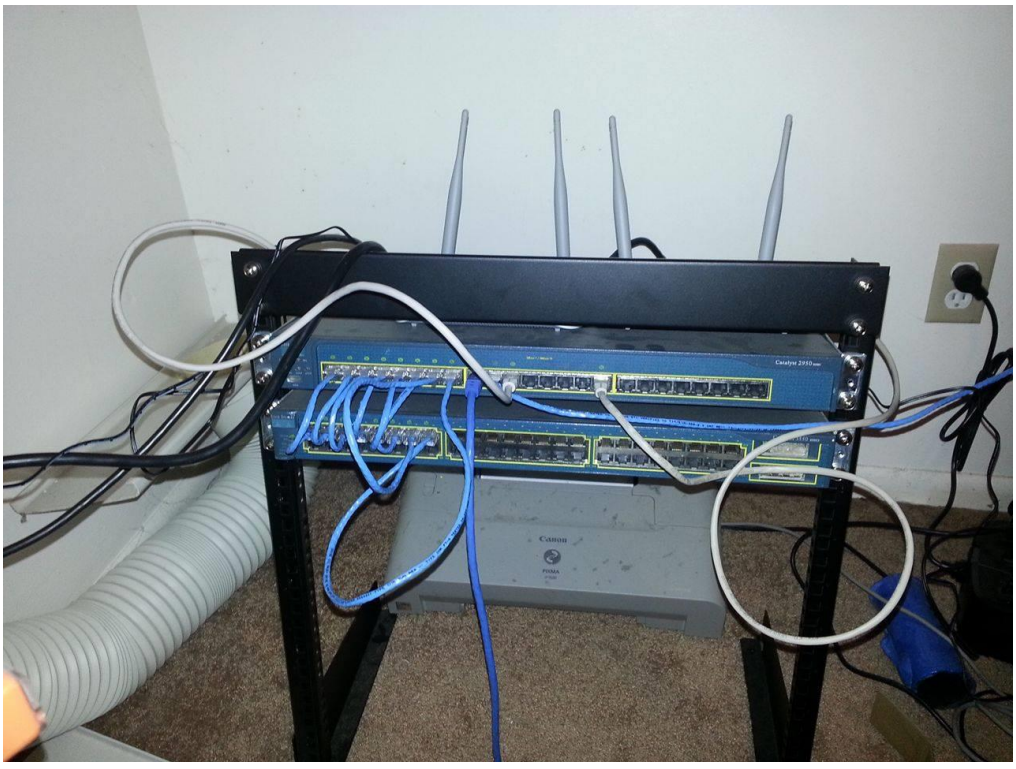
The 2950 was given the hostname "L2" using the *hostname* global config command. Likewise, the 3550 was given the host name "L3". Once basic configuration was completed, the rest of the network setup was done. This document will outline the steps in order of OSI model layer.

Layer 1: Physical

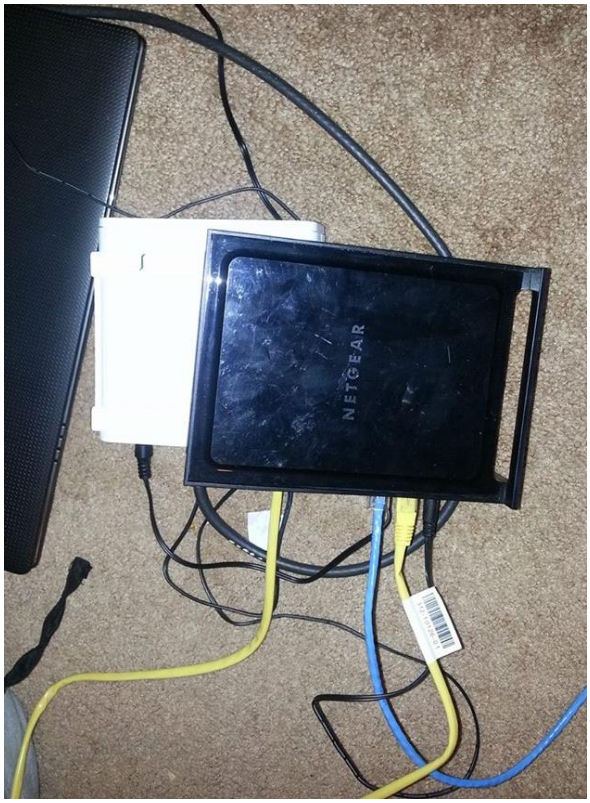
In order to connect all the devices in the network, some cables needed to be created. Eight short crossover cables were created to connect the switches together. Two straight-through cables were created to link the client and server to the switches.



In one end, the crossover cables were connected to the Catalyst 2950 in ports Fast Ethernet 0/1 through 0/8. The other ends were connected to the odd ports on the Catalyst 3550 in range Fast Ethernet 0/1 through 0/15. This made for a more complex software configuration but physically made the setup cleaner because the odd ports are closer to the 2950 above than the even ports. The server was plugged into L2 in port Fast Ethernet 0/9. The client was plugged into L2 in port Fast Ethernet 0/13. The wireless access points, after being configured, were plugged into L2 ports Fast Ethernet 0/11 and/or 0/16, depending on the subnet they were configured for.



The Netgear WNR2000, which was already plugged into the cable modem, was also plugged into L2 in port Fast Ethernet 0/10.



Layer 2: Data Link

The ports the crossover cables were plugged into were configured into an Etherchannel in both switches. LACP was used to negotiate setup of the port channel. The following command was entered into both switches for each interface that had a crossover cable plugged into it:

```
L3(config-if)#channel-group 1 mode active
```

The interface command *channel-group 1 mode active* specifies the port channel the interface will belong to as well as the mode it will be in for negotiation. As long as both sides of the link are set to active, the port will form a port channel with the other switch and all ports in the same

channel group on the same switch will be part of the same Etherchannel.

Four VLANs were configured on each switch and named, per the table above. The commands below were entered on both routers for each VLAN:

```
L3(config)#vlan 10
L3(config-vlan)#name SERVER
```

The command *vlan x* creates a VLAN of ID "x" while the *name* command describes the VLAN for easier management.

Since ports 0/1 through 0/8 were in the EtherChannel on the 2950, there were sixteen ports available to be set up as access ports. Fast Ethernet 0/9 through 0/12 were configured as access ports on VLAN 10 for the Server network and 0/13 through 0/16 were configured as access ports on VLAN 20 for the Client network. Ports 0/17 through 0/24 then were configured as access ports on VLAN 30 for Guest access:

```
L2(config)#interface fa0/9
L2(config-if)#switchport mode access
L2(config-if)#switchport access vlan 10
L2(config-if)#spanning-tree portfast
```

The interface command *switchport mode access* makes the port only accept traffic from a single VLAN while the *switchport access vlan* command specifies what VLAN the port accepts. The last command, *spanning-tree portfast* prevents ports facing end devices from waiting to receive BPDUs for spanning-tree before transmitting and receiving data on the port. Essentially, end devices connected to ports with this command can start using network resources faster.

The EtherChannel and included ports were configured as trunk ports that allowed the four VLAN's configured as follows:

```
L2(config)#interface port-channel 1
L2(config-if)#switchport mode trunk
L2(config-if)#switchport trunk allowed vlan 10,20,30,40
```

The command *switchport mode trunk* allows the interface to accept multiple VLAN's, unlike its access mode counterpart. This command should only be used on ports between switches, never on ports that connect to end devices, for security purposes. The command *switchport trunk allowed vlan* specifies the VLANs that can be carried over the trunk link. VLAN 40 was made the native VLAN for all trunk interfaces using the command *switchport trunk native vlan*. This VLAN is used by the switches to transport untagged traffic and should match on all switches in the broadcast domain. Also, on the layer 3 switch, 802.1q encapsulation was required on the interfaces. The command *switchport trunk encapsulation dot1q* accomplished this. L2 defaulted to 802.1q encapsulation and did not need the command entered on its trunk links:

```
L3(config)#interface f0/15
L3(config-if)#switchport trunk encapsulation dot1q
L3(config-if)#switchport trunk native vlan 40
```

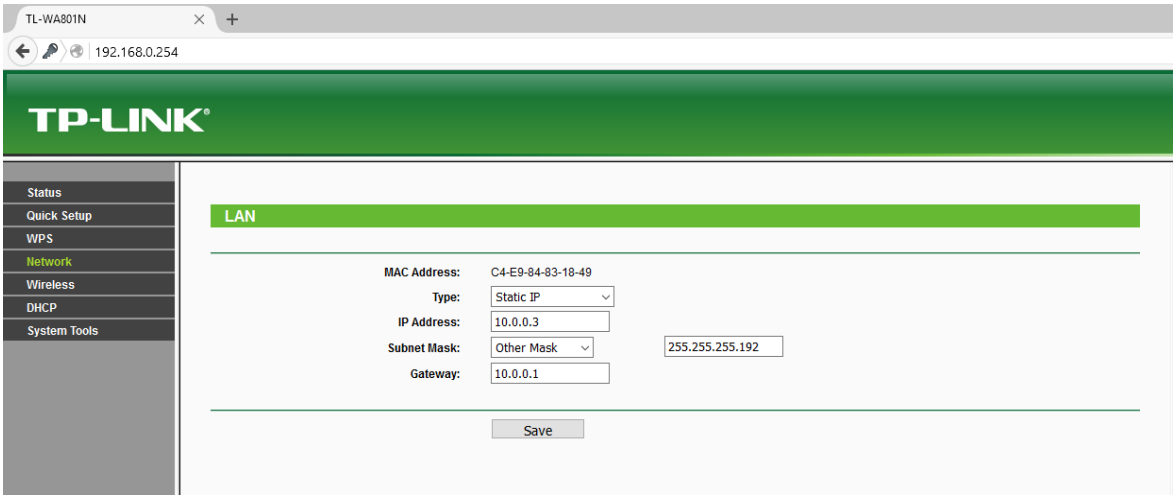
Lastly, all unused Fast Ethernet ports on the 3550 were shutdown for security reasons:

```
L3(config)#interface f0/4
L3(config-if)#shutdown
```

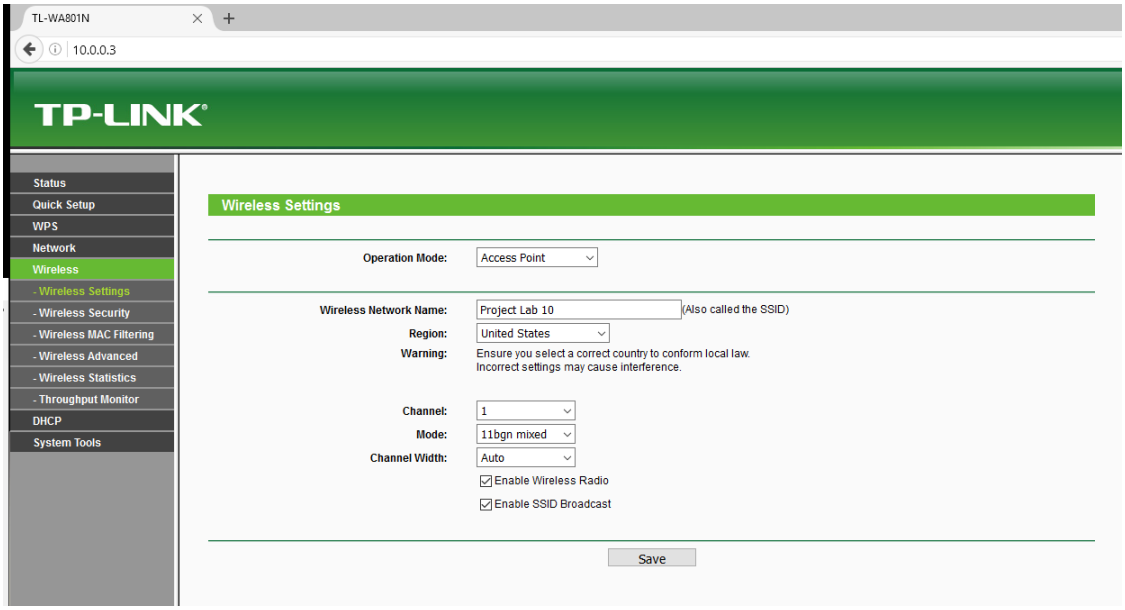
The shutdown command administratively shuts down the Fast Ethernet port. Anything plugged into the port will not be able to communicate to the switch at all through that port. This command was repeated across all unused ports.

Wireless access point setup was done entirely through the GUI provided on the device. A WAP was needed for each VLAN, since they were unable to support multiple VLAN traffic at the same time. They were assigned the addresses above and given the gateway of the

corresponding SVI's of L3. Below is the IP address configuration of the WAP for VLAN 10:

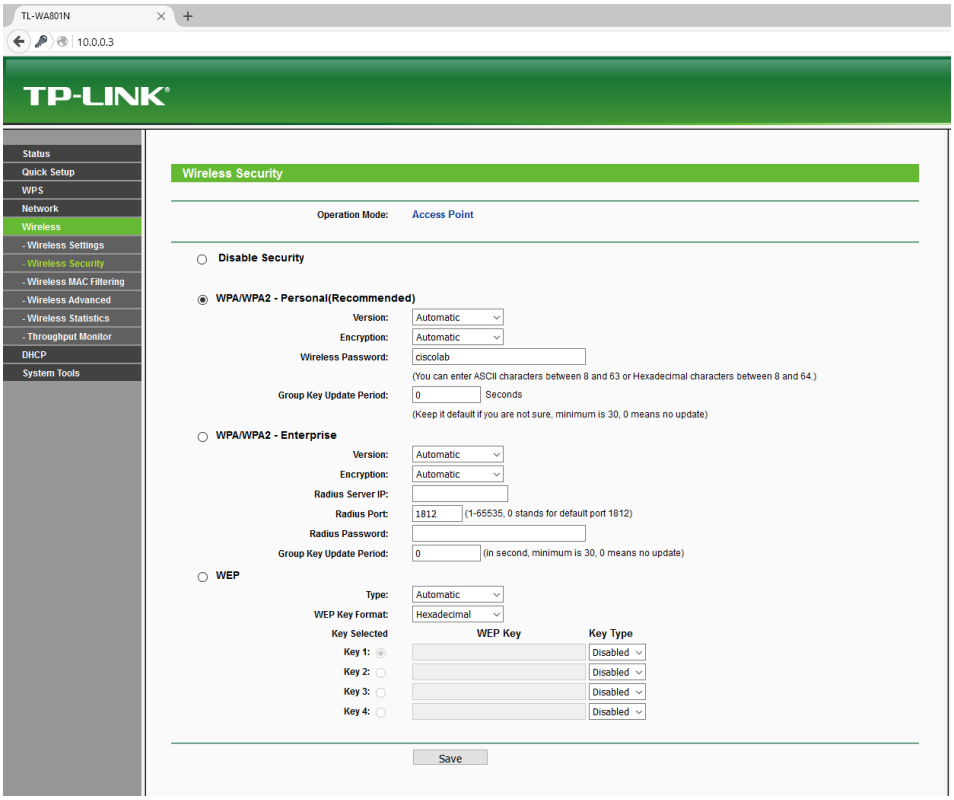


Next, the wireless settings were configured. The SSID was set to "Project Lab X", where X was the VLAN and the channel was set to either 1 or 11 on either WAP to prevent interference from one another.

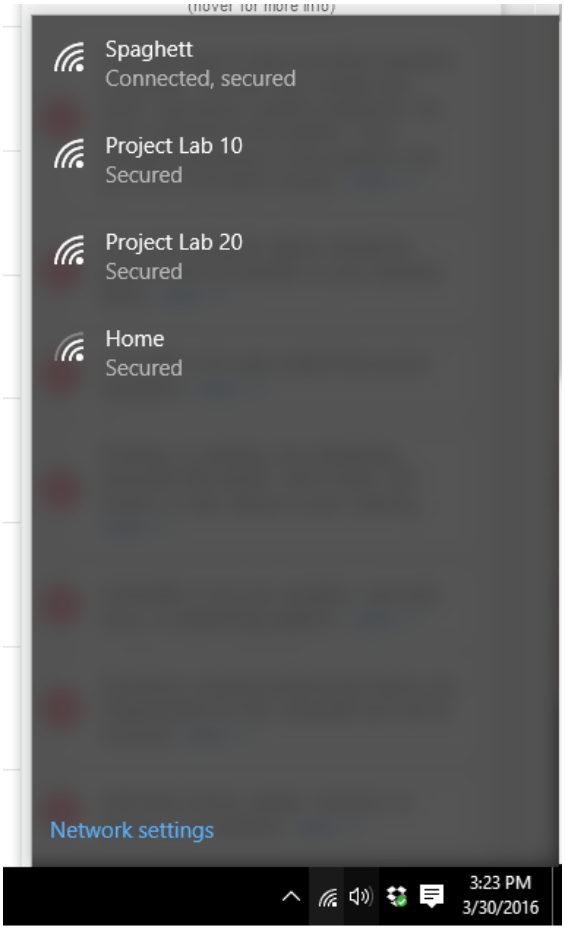


For wireless security, WPA Personal encryption was used and the password was set as "ciscolab". Like the enable secret above, once testing was over, this would need to be switched

to a more secure password featuring alphanumeric characters.



When it came time to connect the client and server to the wireless networks, the respective operating systems' GUI tools were used. Both had wireless icons that when clicked popped up a window with available wireless networks:



The proper network was selected and then the password configured on the access point was entered. Also, the Ethernet cable was connected from the computer when it was put onto the wireless network.

Layer 3: Network

Layer 3 routing was enabled on the 3550 in global config mode using the command *ip routing*. The VLAN interfaces on both switches were given the gateway addresses specified in the tables above with the command *ip address*:

```
L3(config)#ip routing
L3(config)#interface vlan 10
L3(config-if)#ip address 10.0.0.1 255.255.255.192
```

While that created gateway addresses for each VLAN, DHCP needed configured for VLANs 10, 20, and 30 for end devices connecting onto the network. The first five addresses in each subnet were reserved for static assignment to the gateway and any other devices that needed a static address such as the WAPs and server. Then, pools were defined for the three subnets. The corresponding gateway address was assigned to the pool. Lastly, Google's DNS server was setup as the DNS server address for the DHCP pools so clients could direct all DNS resolution requests directly to Google:

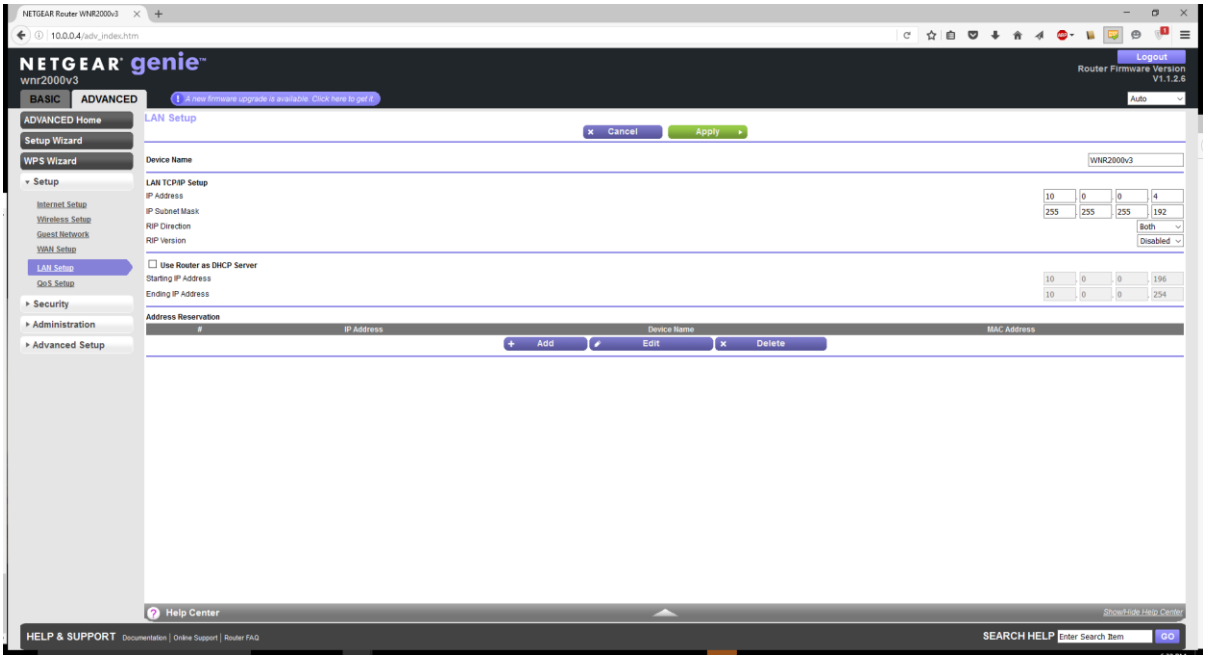
```
L3(config)#ip dhcp excluded-address 10.0.0.1 10.0.0.5
L3(config)#ip dhcp pool SERVER_DHCP_POOL
L3(dhcp-config)#network 10.0.0.0 255.255.255.192
L3(dhcp-config)#default-router 10.0.0.1
L3(dhcp-config)#dns-server 8.8.8.8
```

The command *ip dhcp excluded-address* prevents addresses within a certain range from being handed out to hosts. It was repeated for the first five addresses of each VLAN. The command *ip dhcp pool* creates a new DHCP pool and enters dhcp-config mode. From there, the network to be used in the DHCP pool is defined with *network* and the default gateway address and DNS server for DHCP hosts is defined with *default-router* and *dns-server* respectively. Again, these commands were repeated for the three pools set up for the three VLANs

While the 3550 did not support Network Address Translation, to set this up within the Cisco IOS, the command *ip nat inside* would be used on the VLAN interfaces and the *ip nat outside* commands would be used on the interface connected to the cable modem.

For this setup though, a Netgear WNR2000 router was needed to perform network address translation for internet access. First, the router was logged into by connecting the server to the router and browsing to the default address of 192.168.1.1. The first setup step was to

change the IP address of the router to 10.0.0.4 and disable DHCP, since that would be handled by L3.



This configuration was applied and then the server was assigned its IP address of 10.0.0.2/26 so it could continue to communicate to the router without needing DHCP. This was accomplished from the web based GUI by going to Network and Sharing Center > Change Adapter Settings > right clicking Local Area Connection > clicking “Properties” > highlighting “Internet Protocol Version 4” and selecting properties, then manually entering the IP address and gateway information.

Next, on L3 a default route was configured to go to the Netgear’s LAN address. This allowed any traffic destined outside the lab network to be forwarded to the Netgear router, have the address on the packets translated, and then be sent over the Internet. This was accomplished with the global config command *ip route*:

```
L3(config)#ip route 0.0.0.0 0.0.0.0 10.0.0.4
```

For security purposes, the Guest network should have only had access to the Server network. All other communication was blocked from the Guest VLAN. To accomplish this, an extended ACL was created and applied inbound to the VLAN 30 SVI. Because ACLs have an implicit deny all, only a single permit statement was required. The following commands were entered:

```
L3(config)#access-list 100 permit ip 10.0.0.128 0.0.0.63 10.0.0.0 0.0.0.63
L3(config)#interface vlan 30
L3(config-if)#ip access-group 100 in
```

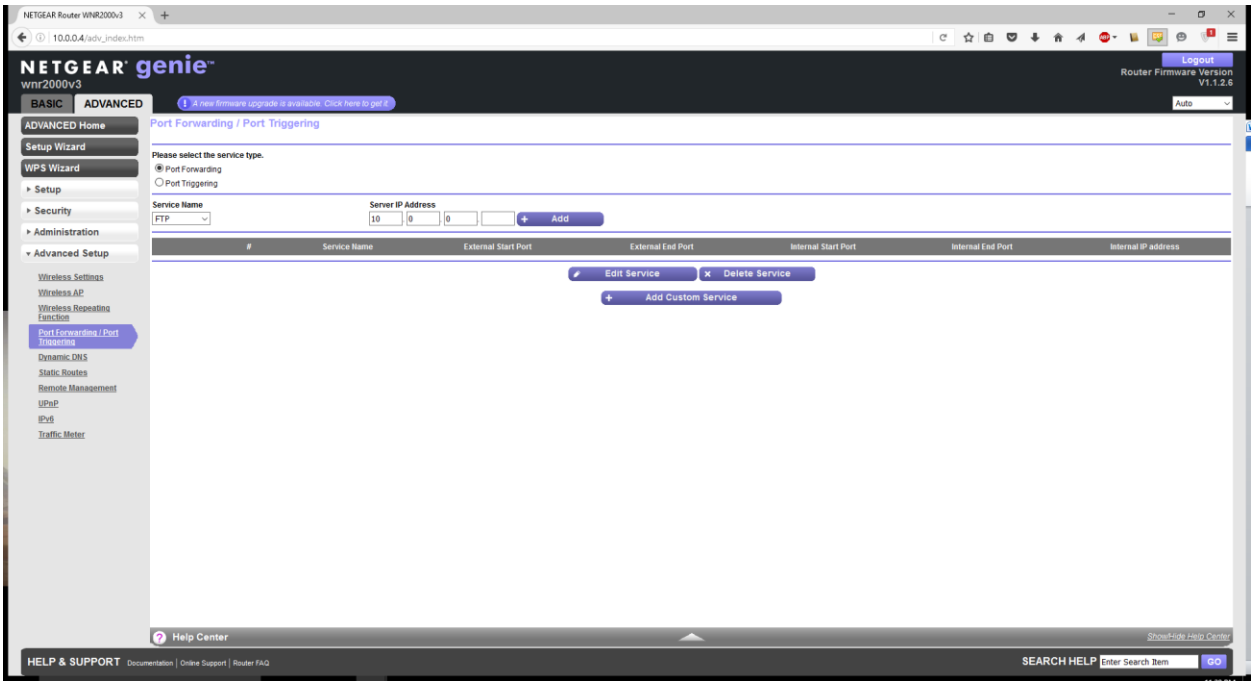
The *access-list* command creates a new list with number 100 and permits traffic using any protocol that has a source address within the 10.0.0.128/26 subnet. The *ip access-group 100 in* command then applies the ACL to all traffic that comes into interface VLAN 30, the Guest subnet.

Layer 4: Transport

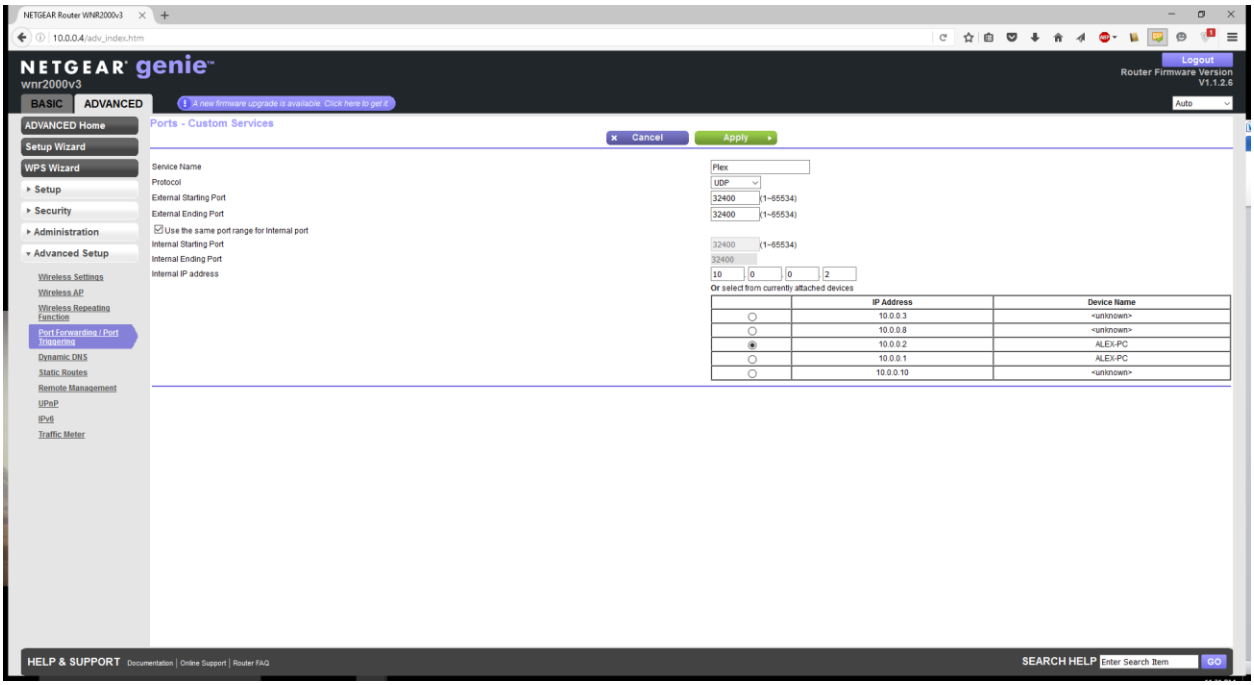
Steam's in-home streaming service uses UDP broadcasts to find other Steam computers on the LAN. Obviously, because the server and client were on separate subnets, those broadcasts would never reach each other. Two commands were used to remedy this. The first, in global config mode, UDP port 27036, which Steam uses for this functionality, was set to be forwarded using the command *ip forward-protocol udp 27036*. Then, on the client VLAN, the server was configured as a destination for UDP packets of that port number using the command *ip helper-address 10.0.0.2*.

```
L3(config)#ip forward-protocol udp 27036
L3(config)#interface vlan 20
L3(config-if)#ip helper-address 10.0.0.2
```

Also, since port forwarding was not possible on the 3550, Plex required UDP port 32400 to be forwarded using the Netgear WNR2000. This was accomplished by logging into the router's web interface at 10.0.0.4. Once there, port forwarding can be found by going to the "Advanced" tab > expanding "Advanced Setup" > selecting "Port Forwarding/Port Triggering".



After clicking "Add custom service", the configuration window was filled out with the port information and destination IP address, as seen below.



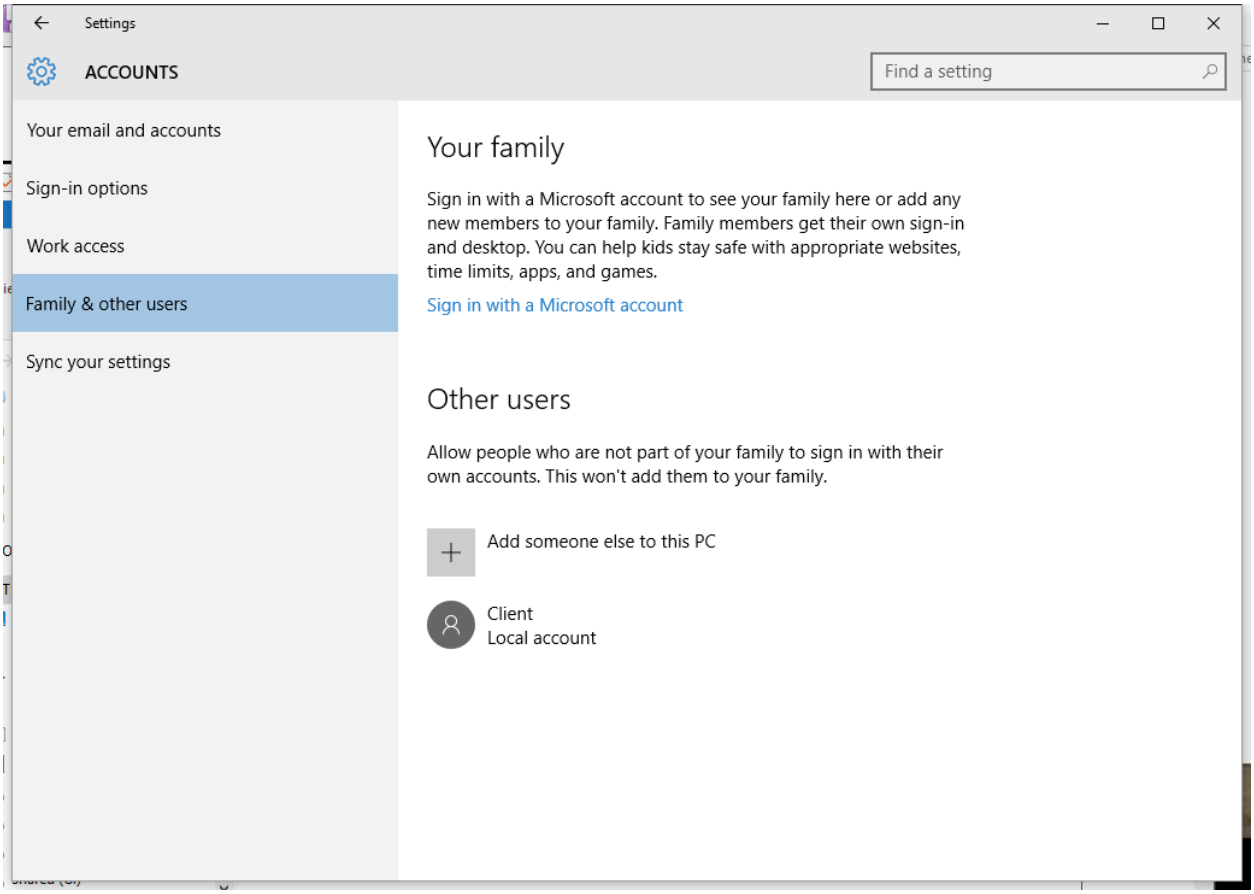
“Apply” was then selected. This completed the port forwarding setup.

Software Setup

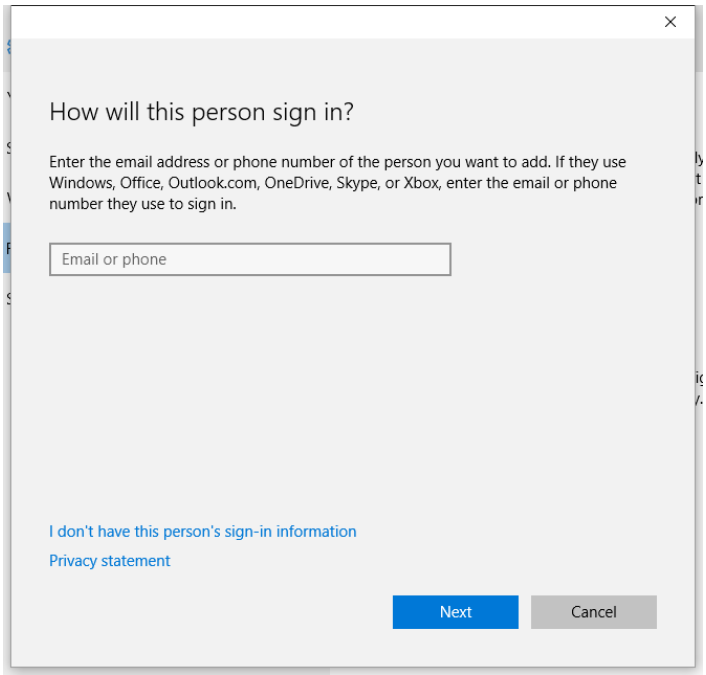
Cifs-Utils and LAN sharing Installation

CIFS, or Common Internet File System, is a file system that allows computers of different platforms to share files across a network using the Server Message Block (SMB) protocol. Because it is supported in both Windows and Linux, CIFS was used to share files between the server, Alex-PC, and the client, steamos, within the LAN.

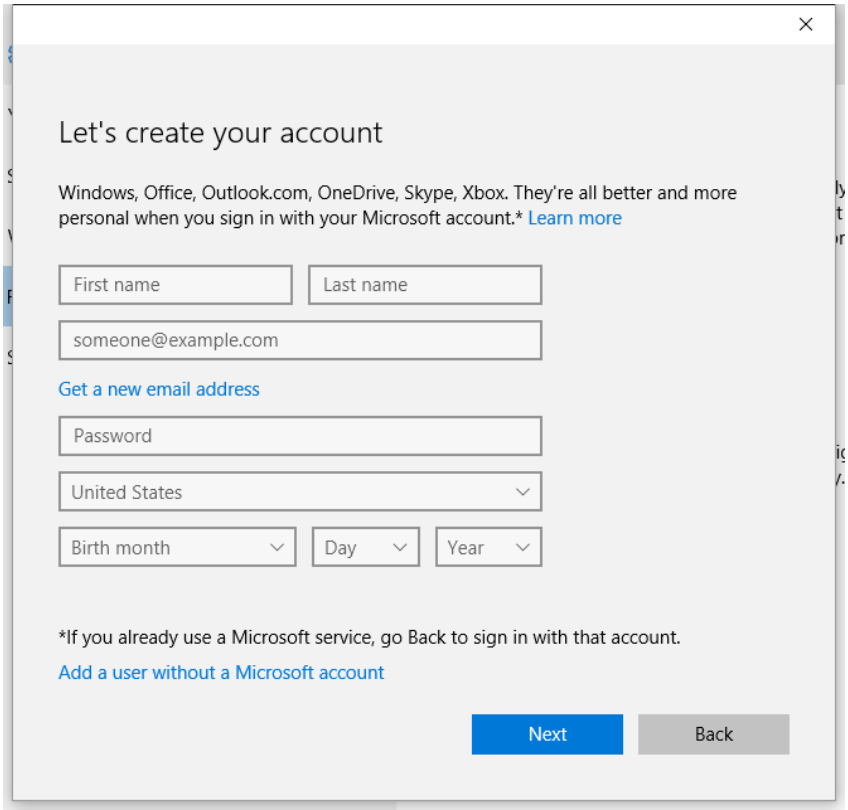
In order to share files between the computers, a remote user account needed to be setup and a network share needed to be created on the server. The user account for steamos was created Windows by going to Start > searching for “account” > selecting “Manage your account (system settings)” > Family & other users on the left pane > Add someone else to this PC



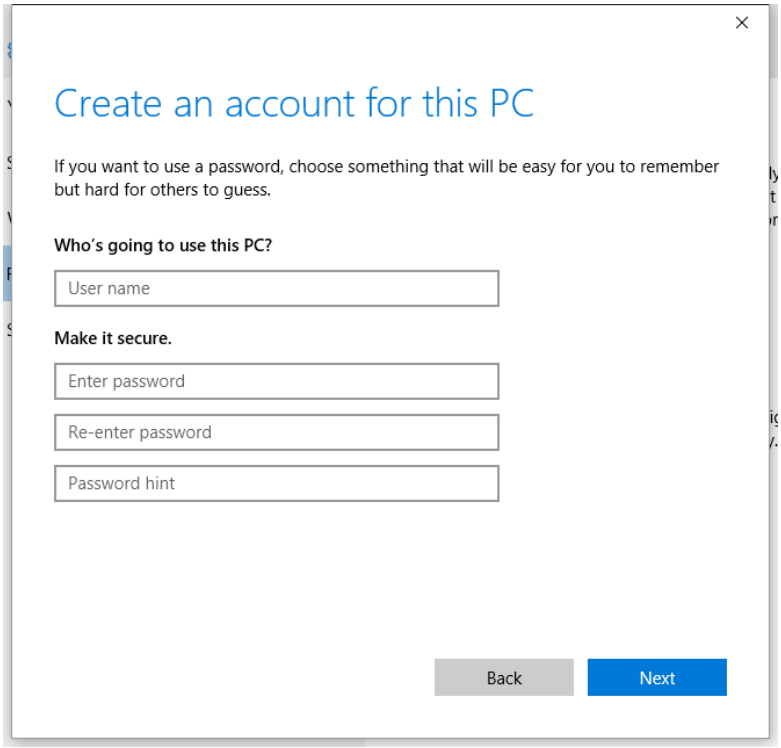
A window popped up asking how the user would sign in. “I don’t have this person’s sign-in information” was selected.



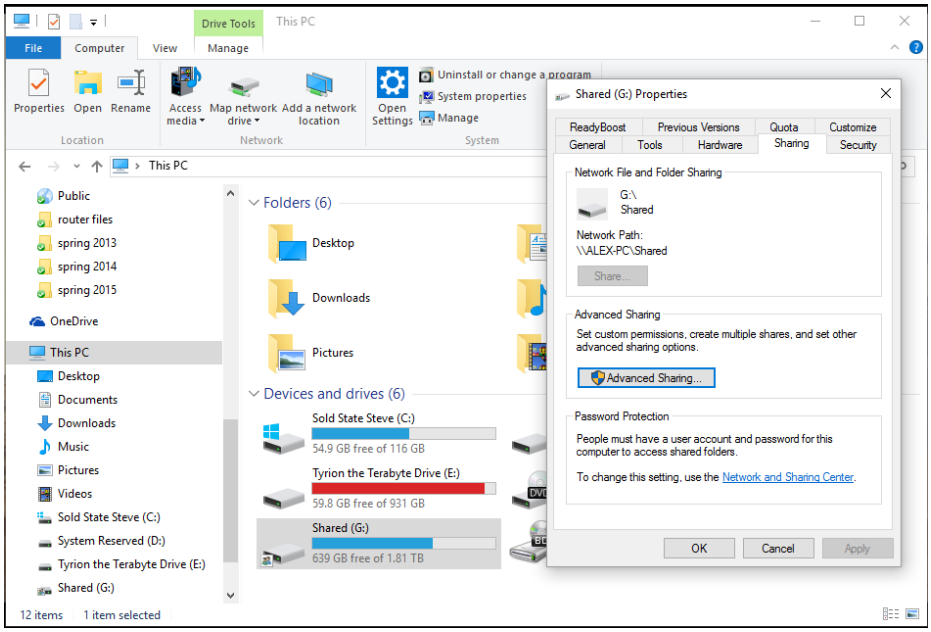
The window then prompted to create a Microsoft account. As this account would only be used to access resources on this server, “Add a user without a Microsoft account” was selected.



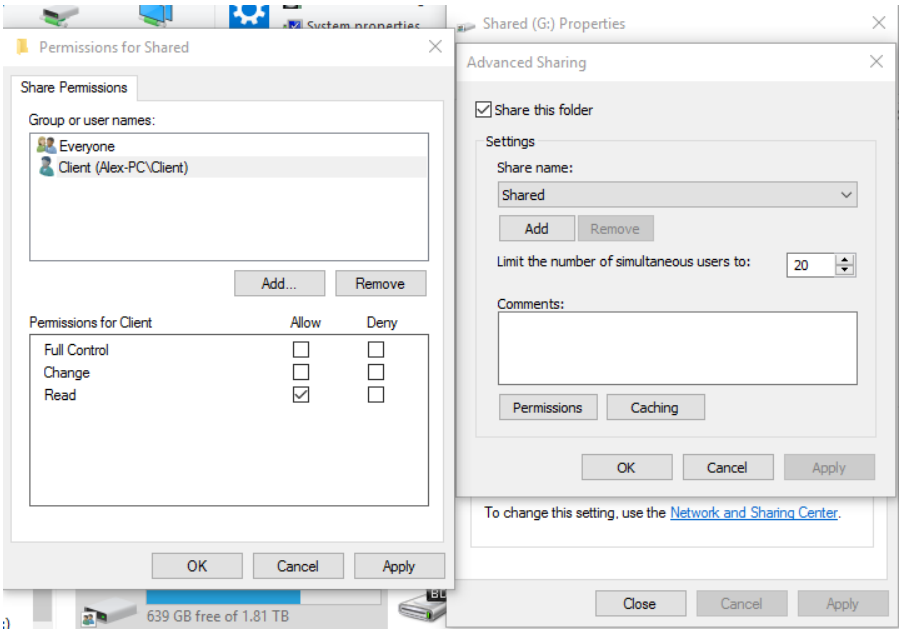
The username was set to “Client” and the password was set to “cisco” to keep it consistent with other passwords in the lab. In an enterprise or production environment this would not be secure. “Next” was then selected.



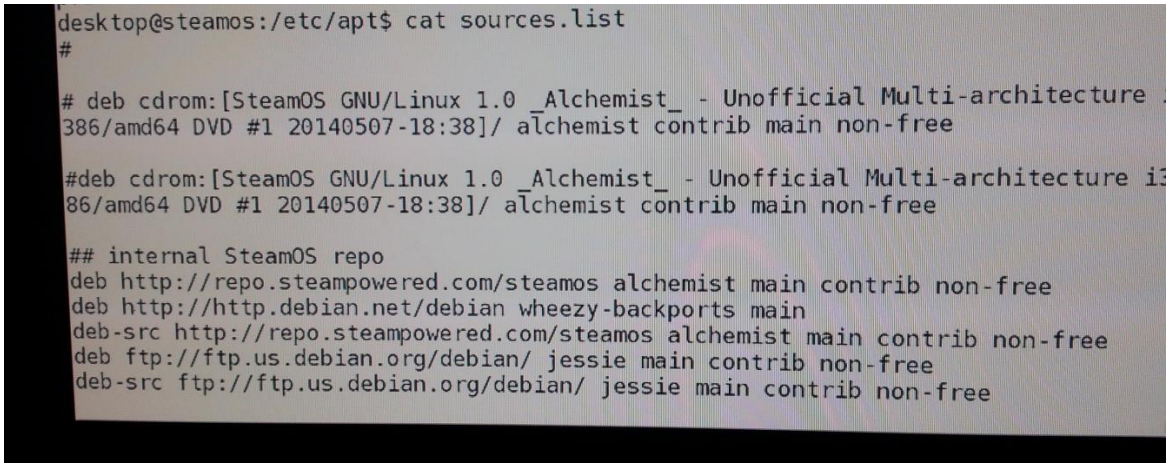
Next, a 2TB external hard drive, storing the requisite media files, was setup to be shared over the network in Windows. This was accomplished by right clicking the drive in file explorer > clicking “Properties” > “Sharing” tab > “Advanced Sharing” (this requires administrator access).



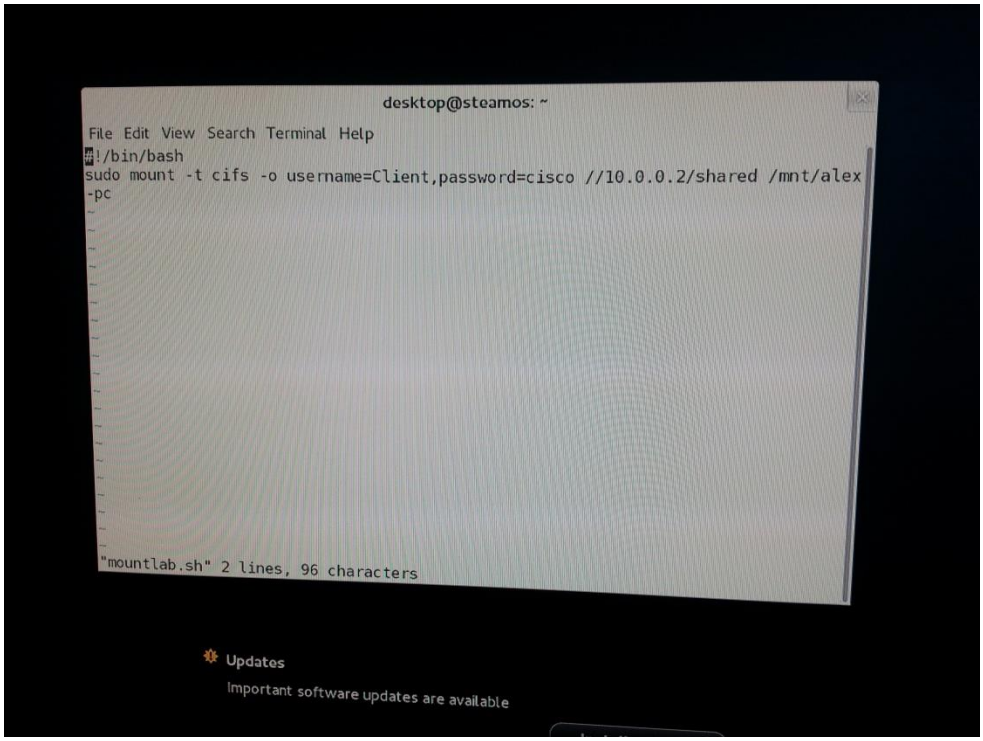
Once there, “Share this folder” was selected. The share name was named “Shared”, the name of the drive, for simplicity. “Permissions” was selected, and “Client (Alex-PC\Client)” was highlighted. This account was given read only permission by only having the “Allow” box checked next to “Read”. “OK” was selected in both windows to close out of the sharing options. Then “Close” was selected on the G drive window to close out of the drive properties window this concluded setup of the share and user account setup on the server:



On the Linux client side, the first step to mounting the share drive was to install *cifs-utils*, a program that allows CIFS drives to be mounted within the operating system. However, as outlined in the testing documentation ("Software Testing" starting on page 5), the Alchemist repository did not include the dependencies necessary for *cifs-utils.* A new repository needed to be added to the sources.list file. A terminal was opened from the Steam OS desktop. The sources.list file was edited by entering the command *sudo vi /etc/apt/sources.list*. The line "deb http://http.debian.net/debian wheezy-backports main" was added to the file. Then the file was saved and exited by typing ":wq" in command mode. Below is a screenshot of the contents of the file. It includes the Jessie repository which was not used in this project:



With the sources file updated, *cifs-utils* was installed by typing *sudo apt-get install cifs-utils*. After confirming the disk space to be taken up, the program was installed. The network share was then mounted within Linux command *sudo mount –t cifs –o username=Client,password=cisco //10.0.0.2/shared /mnt/alex-pc*. This command, *mount*, used the option –t to specify the file system to be mounted, CIFS, the –o option for mounting options such as the username and password to authenticate with the remote server (the credentials match the account setup on the server above), as well as the source of the share, //10.0.0.2/shared, and the mount point within the local file system, /mnt/alex-pc. For future use, this was turned into a script by entering "sudo vi mountlab.sh". A shebang was entered into the first line, *#!/bin/bash*, followed by the mount command above on the line below it. This script was saved using ":wq" in command mode, then given execute permissions by all users, groups, and others by entering *sudo chmod 711 mountlab.sh*. With these permissions, only the root user that created the file can read or change its contents, but other users can still use the file.



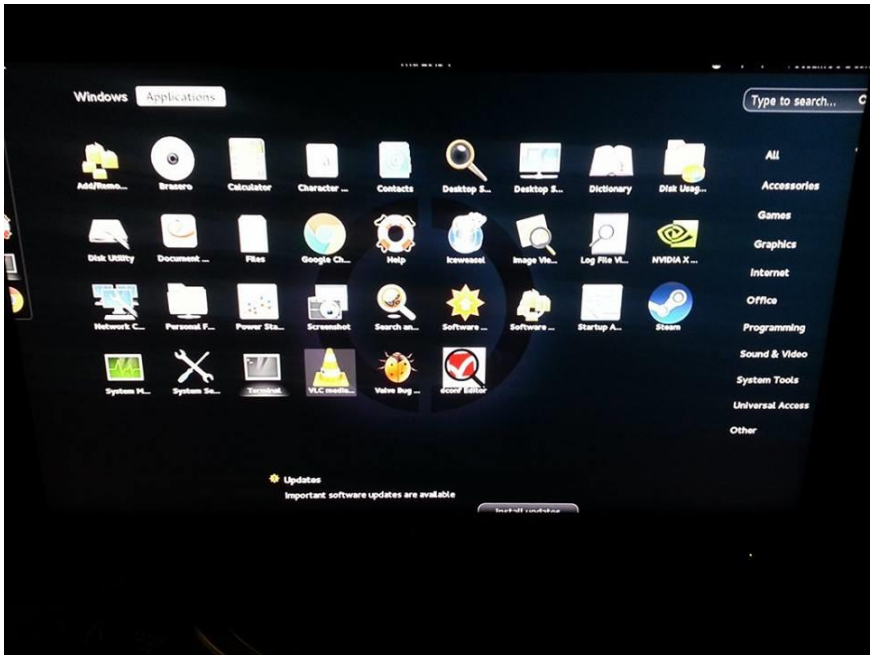
Above is a screenshot of the contents of mountlab.sh.

VLC Media Player Installation

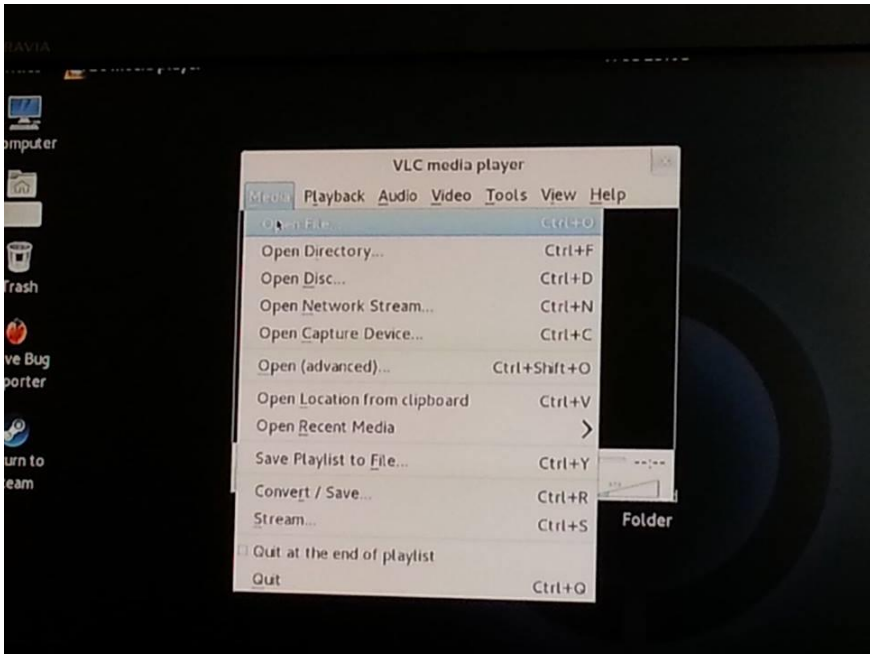
VLC Media Player is an open source program that allows viewing of media files on a computer. Because it is supported in Linux and can play various media formats including the movies and music tested, it was the perfect candidate for use on the client.

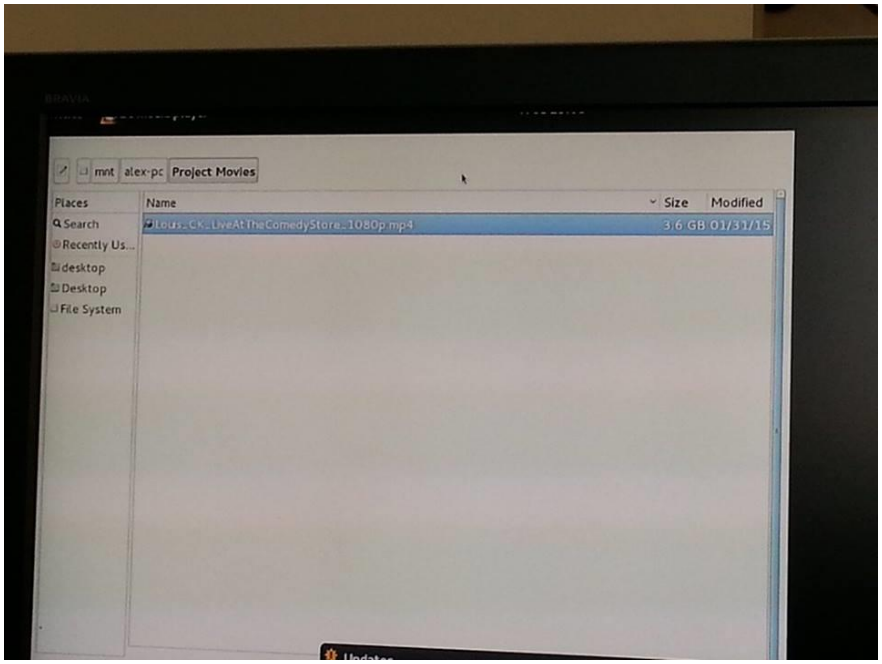
Like *cifs-utils*, VLC was not included in the Alchemist repository. The installation also required the use of the Wheezy repository. Refer to the "Cifs-Utils and LAN sharing Installation" section of this document for instructions on how to add the Wheezy repository to the /etc/apt/sources.list file on a Linux host.

Once the repository was setup, VLC was installed using the command *sudo apt-get -t wheezy install vlc.* Unlike *cifs-utils*, VLC would not install all the required dependencies without specifying the repository with the –t option of the command. The root password was entered. After confirming the disk space requirements by typing "y" and Enter when prompted, installation completed. VLC was available by going to Applications and clicking the traffic cone icon.



In VLC Media Player, media were played by navigating to Media > Open Media and browsing to the appropriate directory in /mnt/alex-pc. In the screenshots below, the Louis CK special was selected from /mnt/alex-pc/Project. Once selected, playback began.

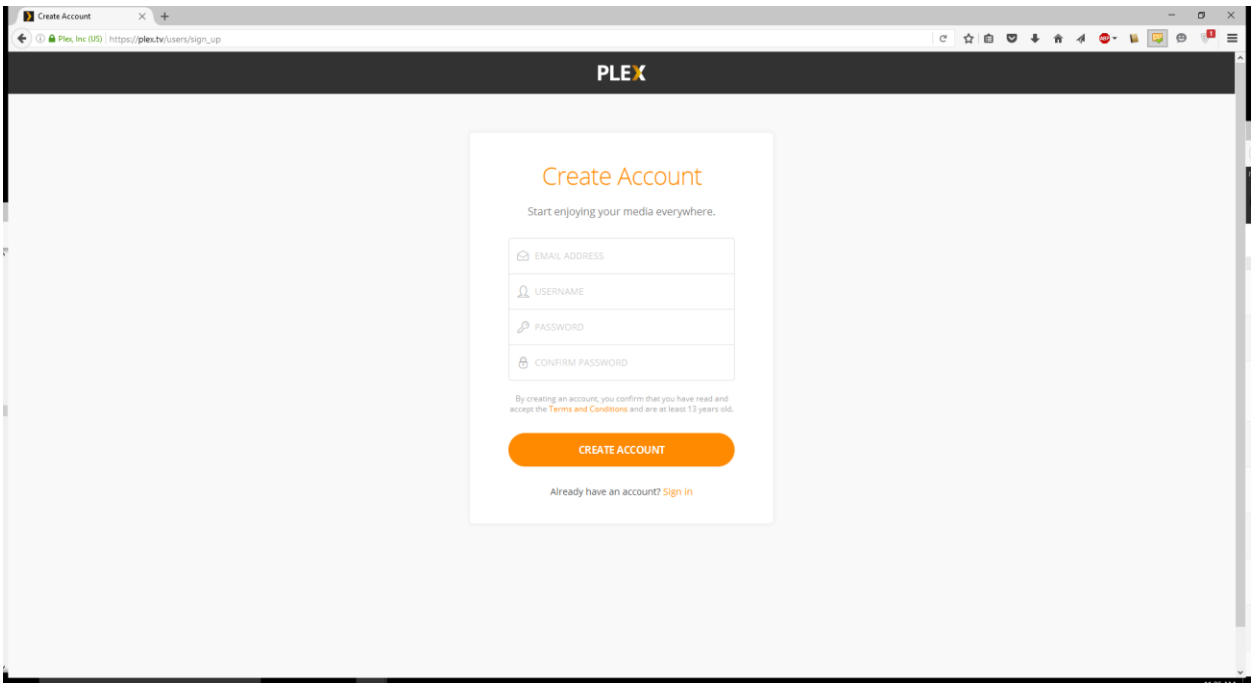




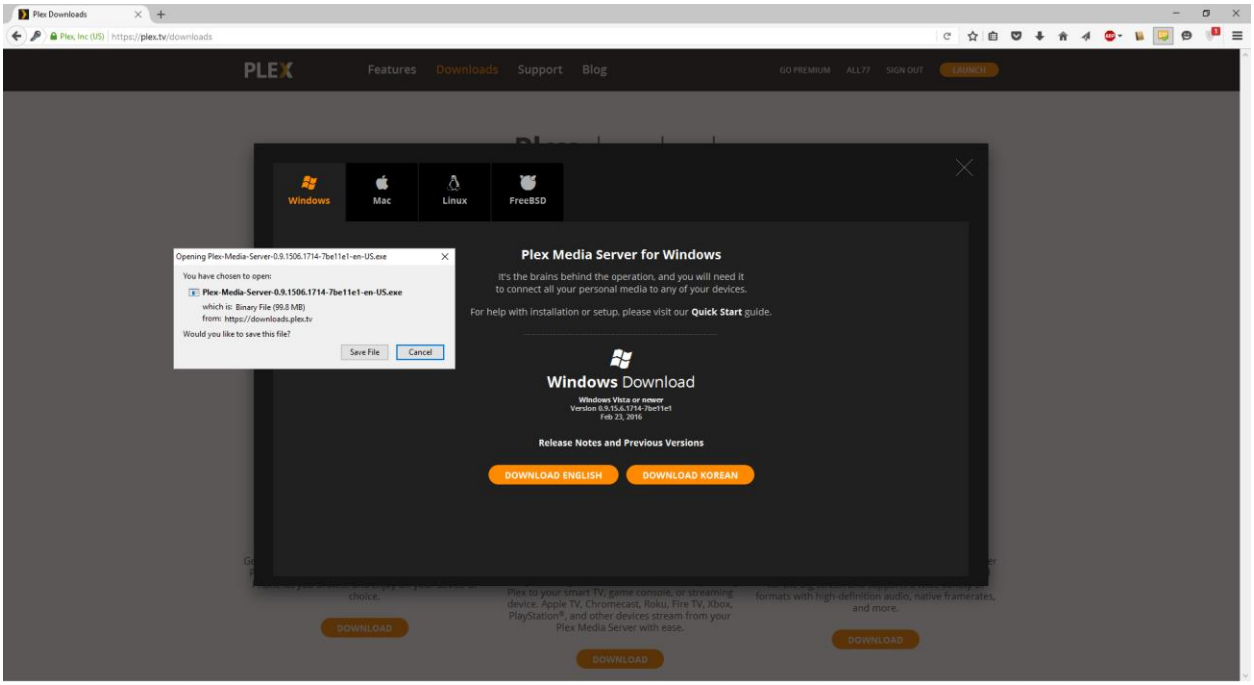
Plex Media Server Installation

Plex is a program that allows streaming of movies and TV shows over the LAN or the Internet to other computers and mobile devices. While the media server application is free, the mobile application does cost $4.99 to unlock, even though it says it is free on the Google Play store. Other free applications exist, but Plex was chosen for its ease of installation and Android support. The server acted as the media server for Plex, and a Samsung Galaxy SIII was the client, running the Android Plex app.

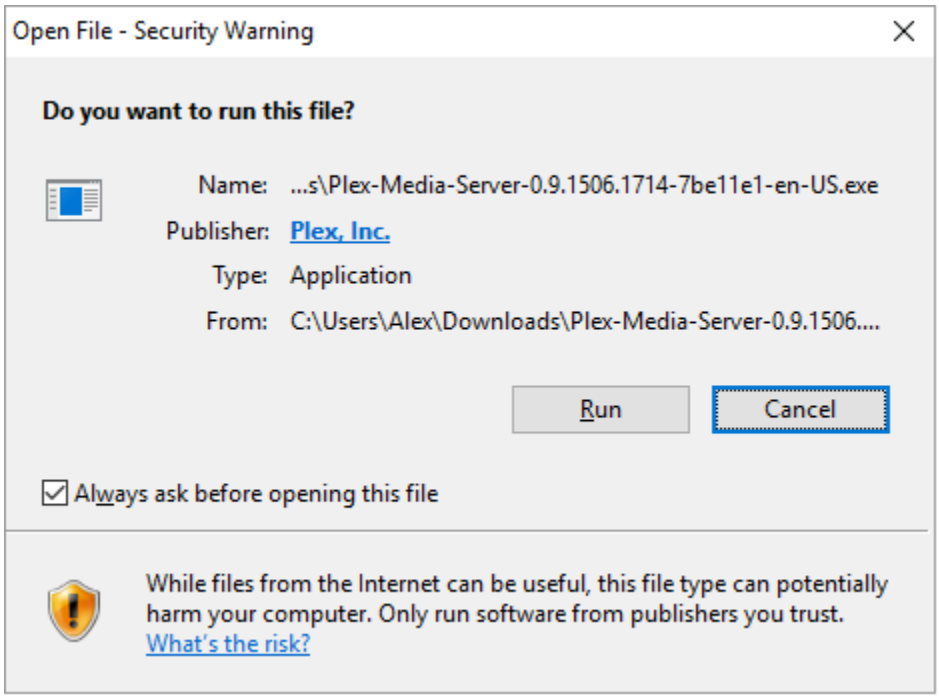
Plex also requires an account be setup. This can be accomplished by navigating to https://plex.tv/users/sign_up. The form requires a valid email address as well as a username and password. Once this information is entered, “Create Account” can be clicked.



On the server, the installation began by downloading the Plex Media Server executable from plex.tv.

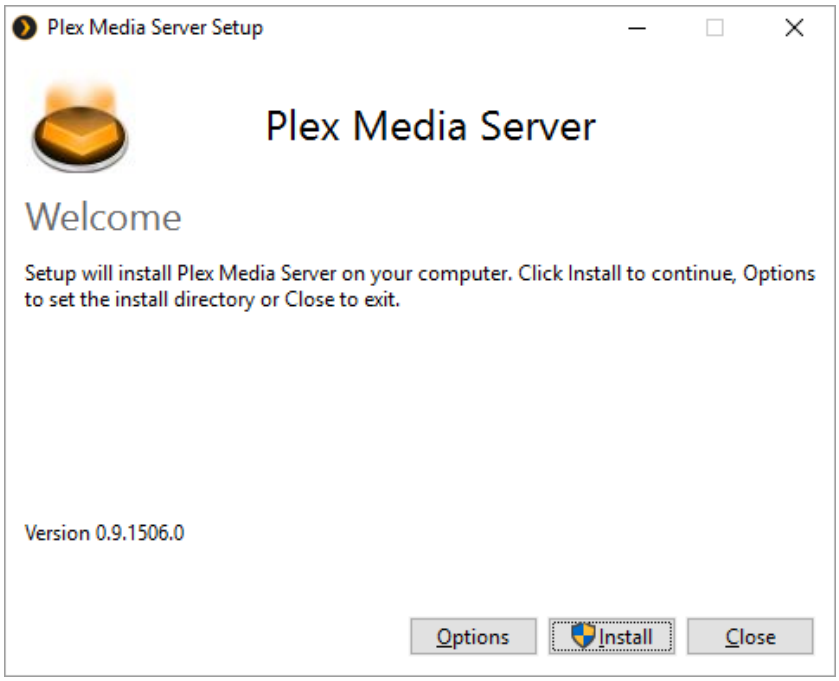


Next, the executable was run to begin the installation process:

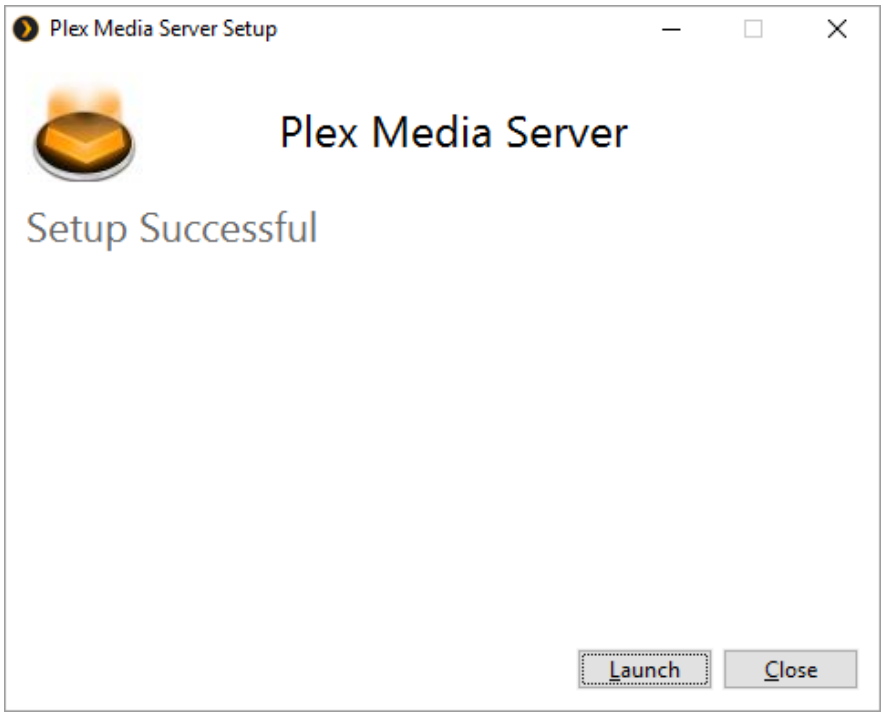


To continue installation, Administrator access is needed. “Install” was clicked to begin

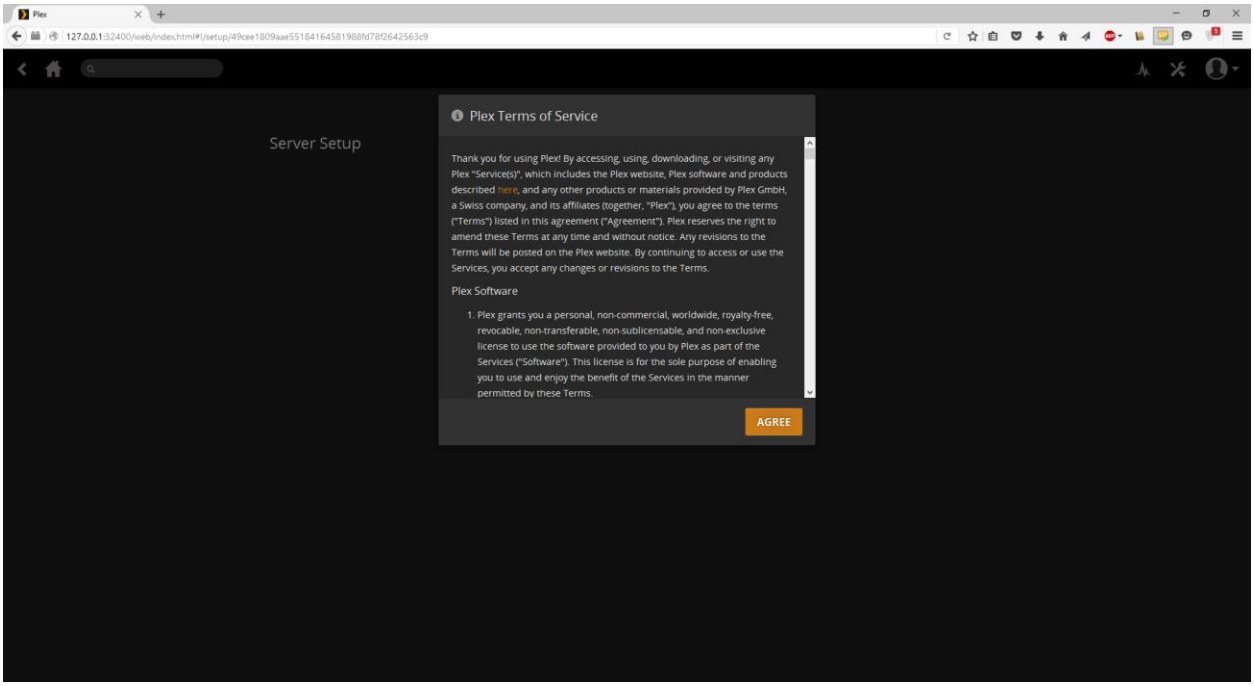
installation:



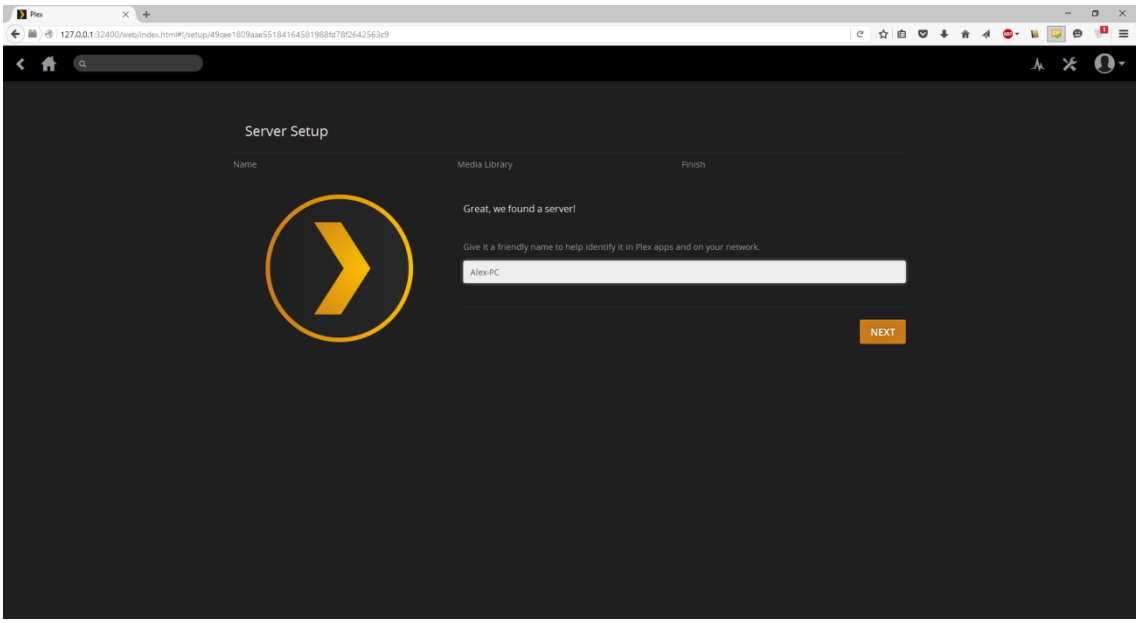
Once it finishes installing itself, it is able to be launched by hitting “Launch”:



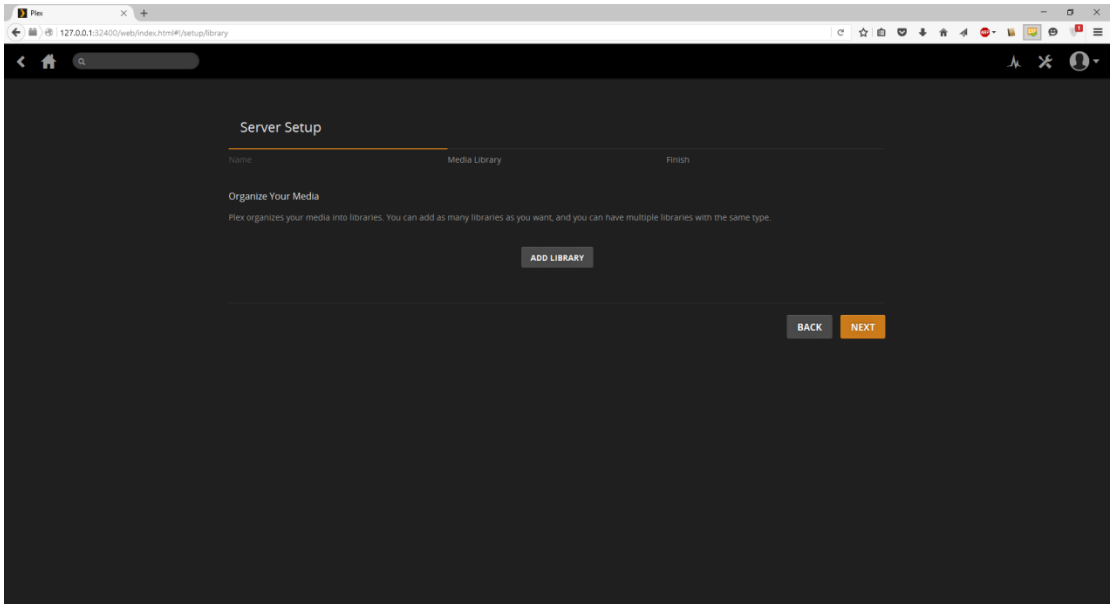
Plex opened up in Firefox by browsing to 127.0.0.1:32400. This socket is made up of the IPv4 loopback address, referring to the local computer, and the port 32400 which Plex uses to communicate over the network. The terms of service were read and agreed to by clicking "Agree":



The application used the local machine as the server and it was named the same as the computer name, Alex-PC. "Next" was then selected:

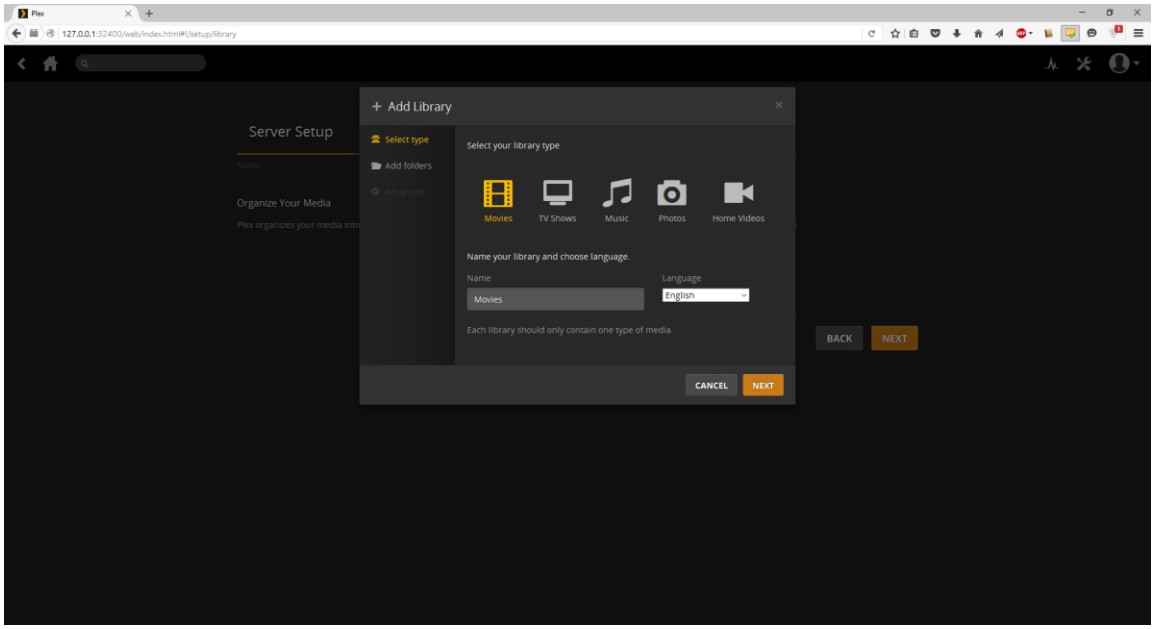


Once the server is named, a library must be setup. Libraries allow the user to organize his/her media based on type, such as movies, music, photos, etc. “Add Library” was selected to add a library to the server:

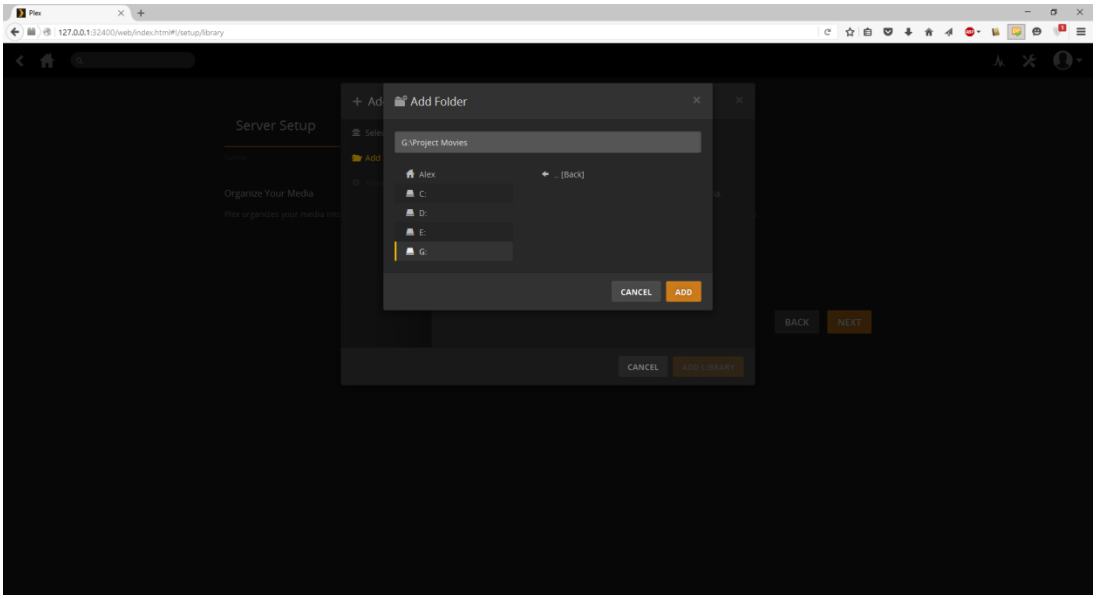


Movies was the first library created, with the type “Movies”. This would include videos such as

Louis CK's standup special used for testing:

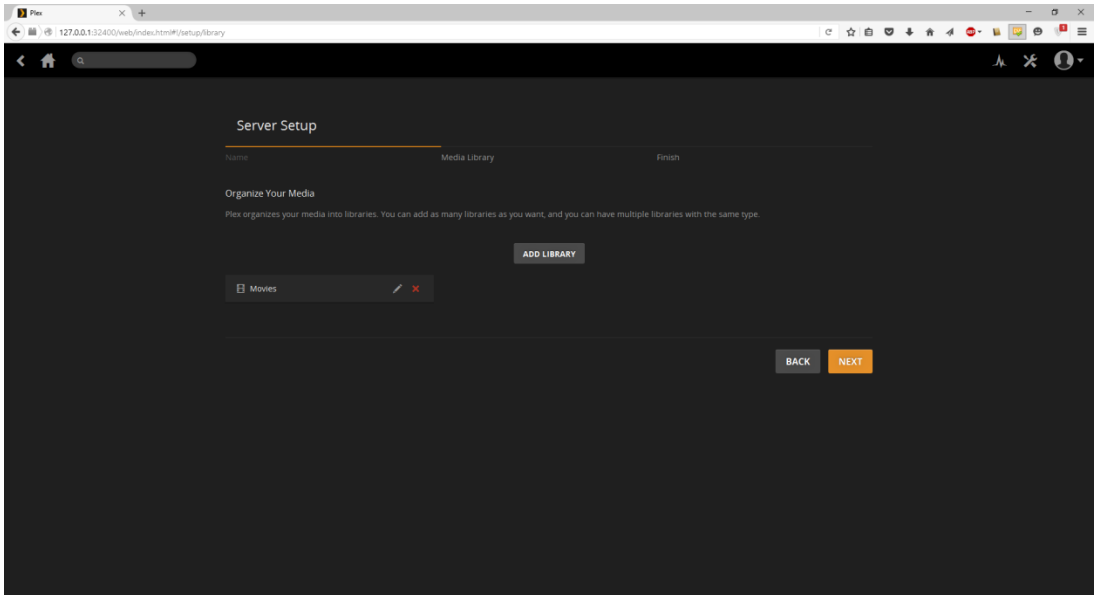


After hitting "Next" above, the folder G:\Project Movies was selected as the media folder. This folder included the Louis CK special. "Add" was selected to import the directory and its contents to the library:

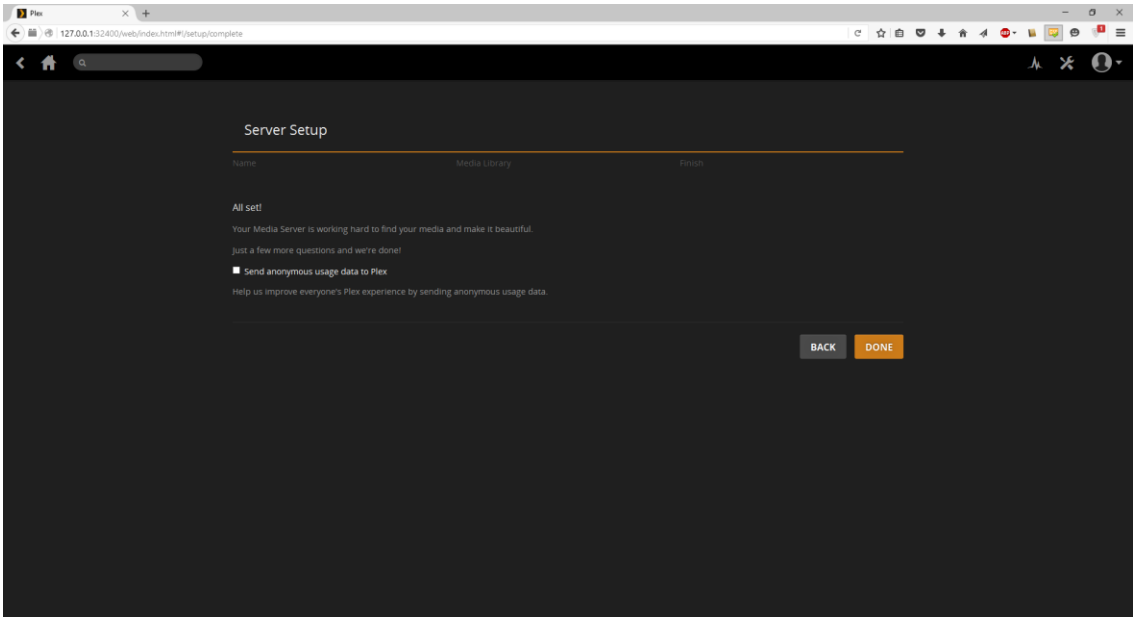


Plex gives the user the option to add other libraries at this point. For testing, only a single library

was added, however, the above steps can be repeated for each additional library needed. “Next” was selected:

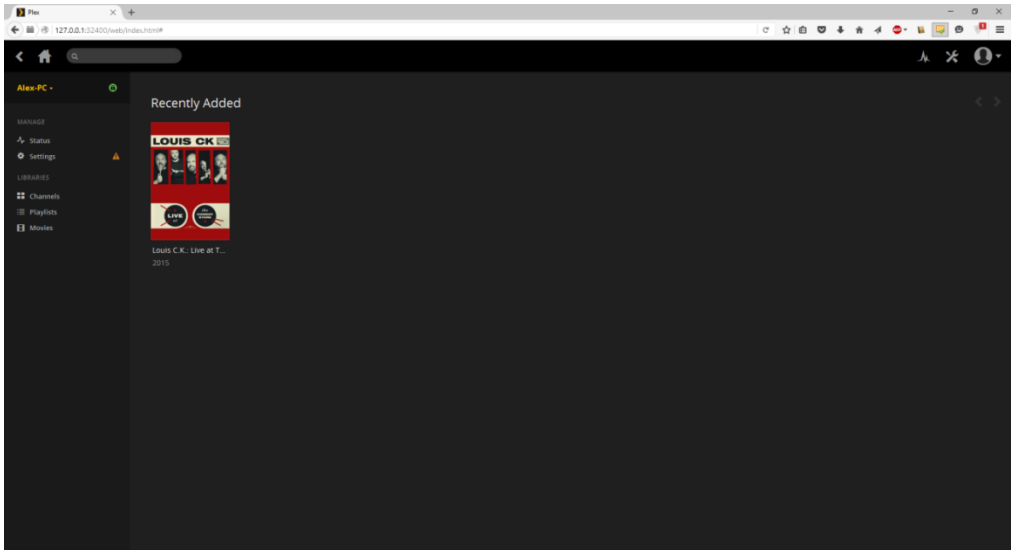


Plex imported the media and was ready to start sharing. “Send anonymous usage data to Plex” was unchecked for security purposes and then “Done” was selected:

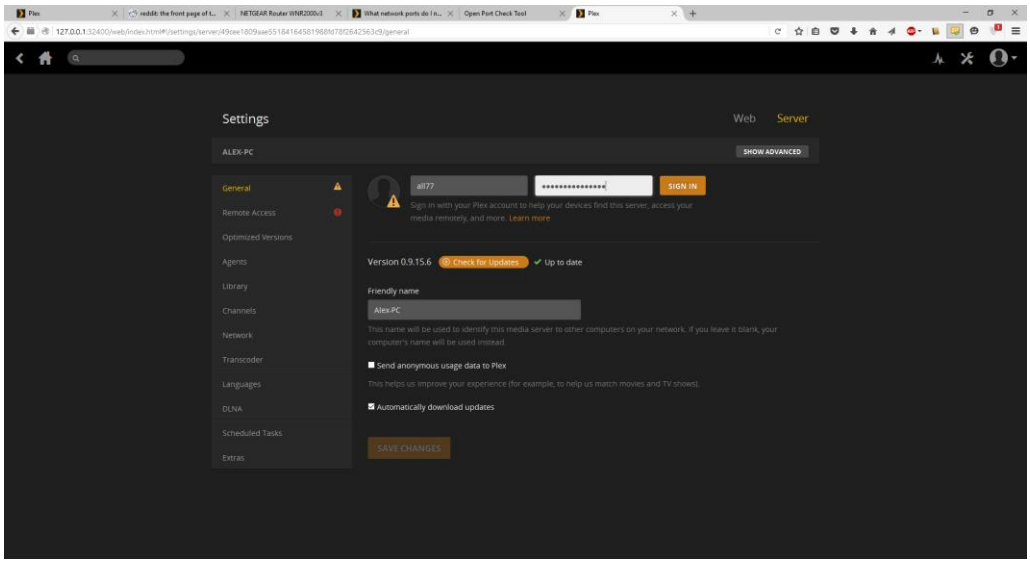


Plex then goes back to the home page. There, imported media and libraries can be viewed. The

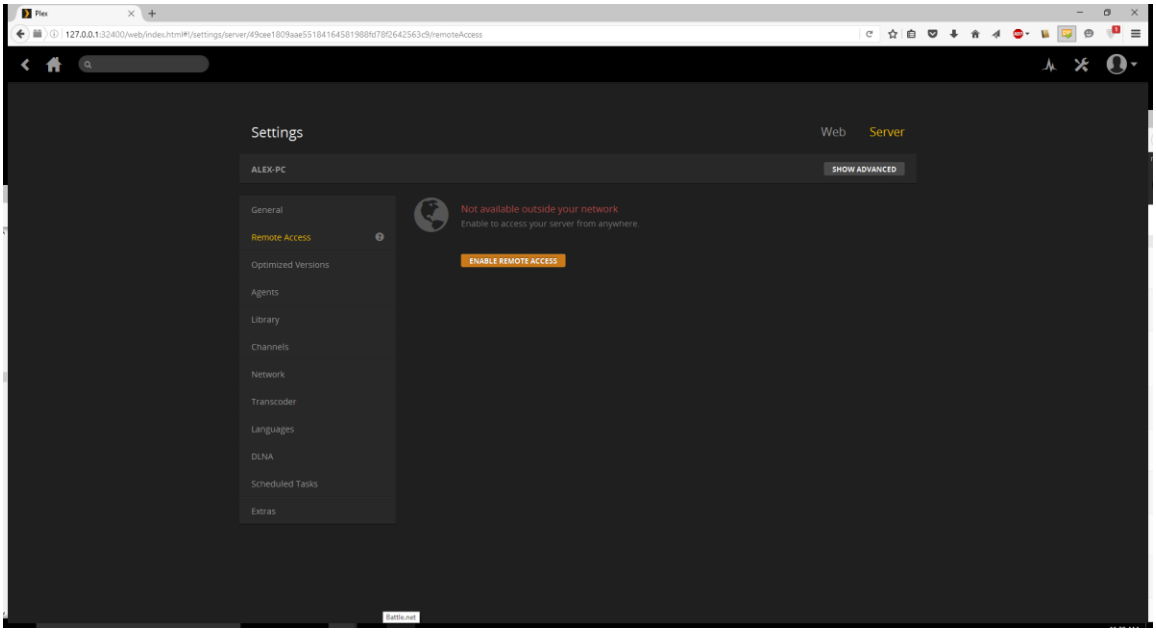
created library, “Movies” can be seen on the left side panel. The media, Louis CK’s special, can also be seen under “Recently Added”:



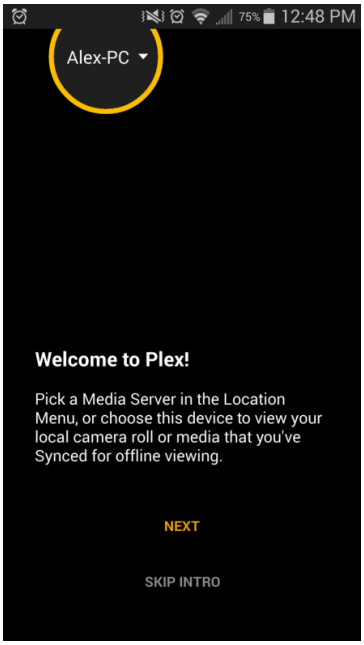
While the server was setup, remote access was not yet enabled. “Settings” on the left panel was clicked. On the “General” tab, the account information setup above was entered to allow other devices to find the server. Then, “Remote Access” was clicked on the left pane:



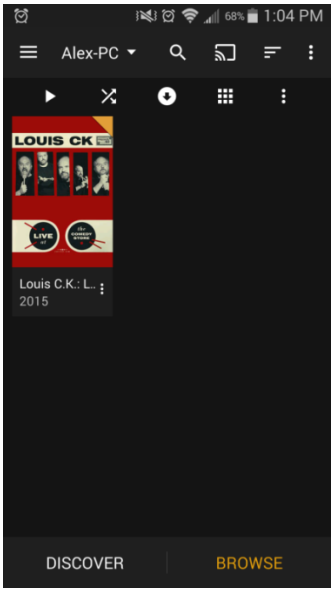
To enable remote access to the media server over the internet, the “Enable Remote Access” button needed selected.



As stated above, the Plex app can be downloaded from the Apple or Google Play store, depending on the device. Once downloaded, the app asks for the login information setup above. Then, the app looks for servers to connect to. The server named Alex-PC was found and then “Next” was hit:

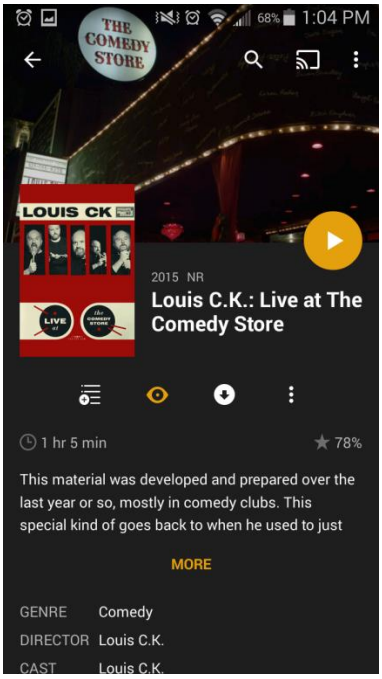


Within the app, the top dropdown menu allows the user to select the server to browse and the three lines to the left of it open up the navigation pane that allow the user to open the libraries. With the server Alex-PC selected, the navigation window was opened and the Movies library was viewed.



Within the Movies library was the Louis CK special. Selecting the title or the thumbnail

icon above it moved to a screen for the material itself. Plex automatically downloaded information about the media such as runtime, synopsis, and genre based on the media's file name and then updated the library to reflect this information. The yellow button with the white triangle begins playback.

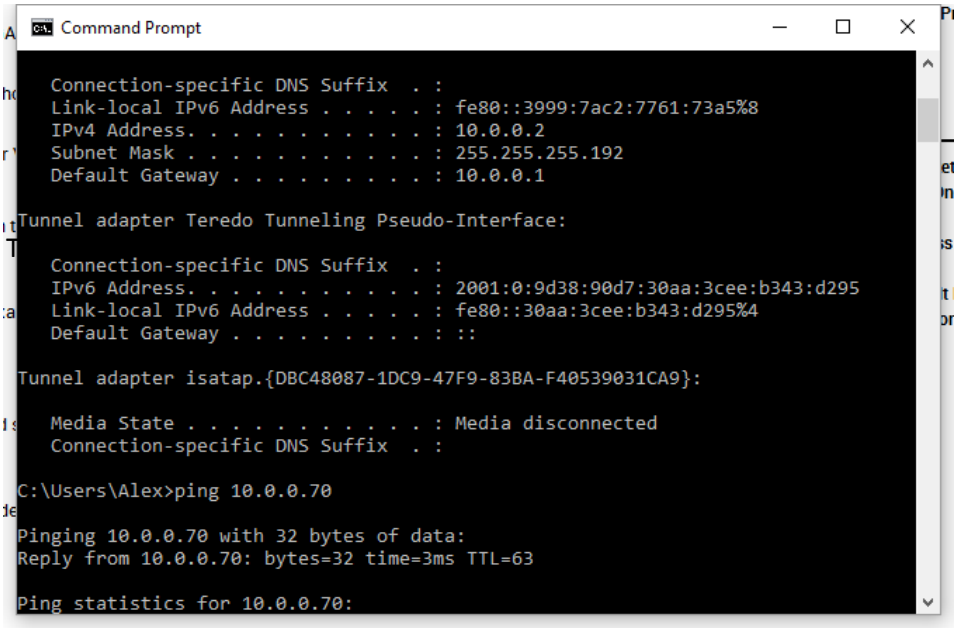


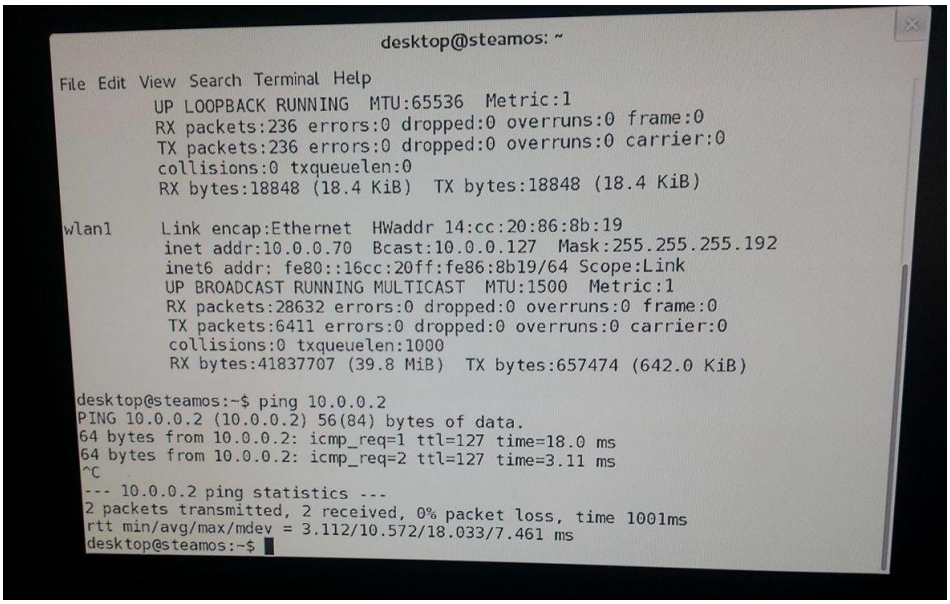
These steps were repeated for the music library that was set up for testing purposes. That library was of type "Music" and named "Music" instead of "Movies".

Testing Documentation

Network Testing

Network testing was completed simply through pinging across the network. Once network setup was complete hosts on VLAN 10, 20, and 30, Alex-PC, steamos, and a laptop plugged into L2's fa0/17 port, pinged each other. Below are some screenshots of the client and server pinging each other:





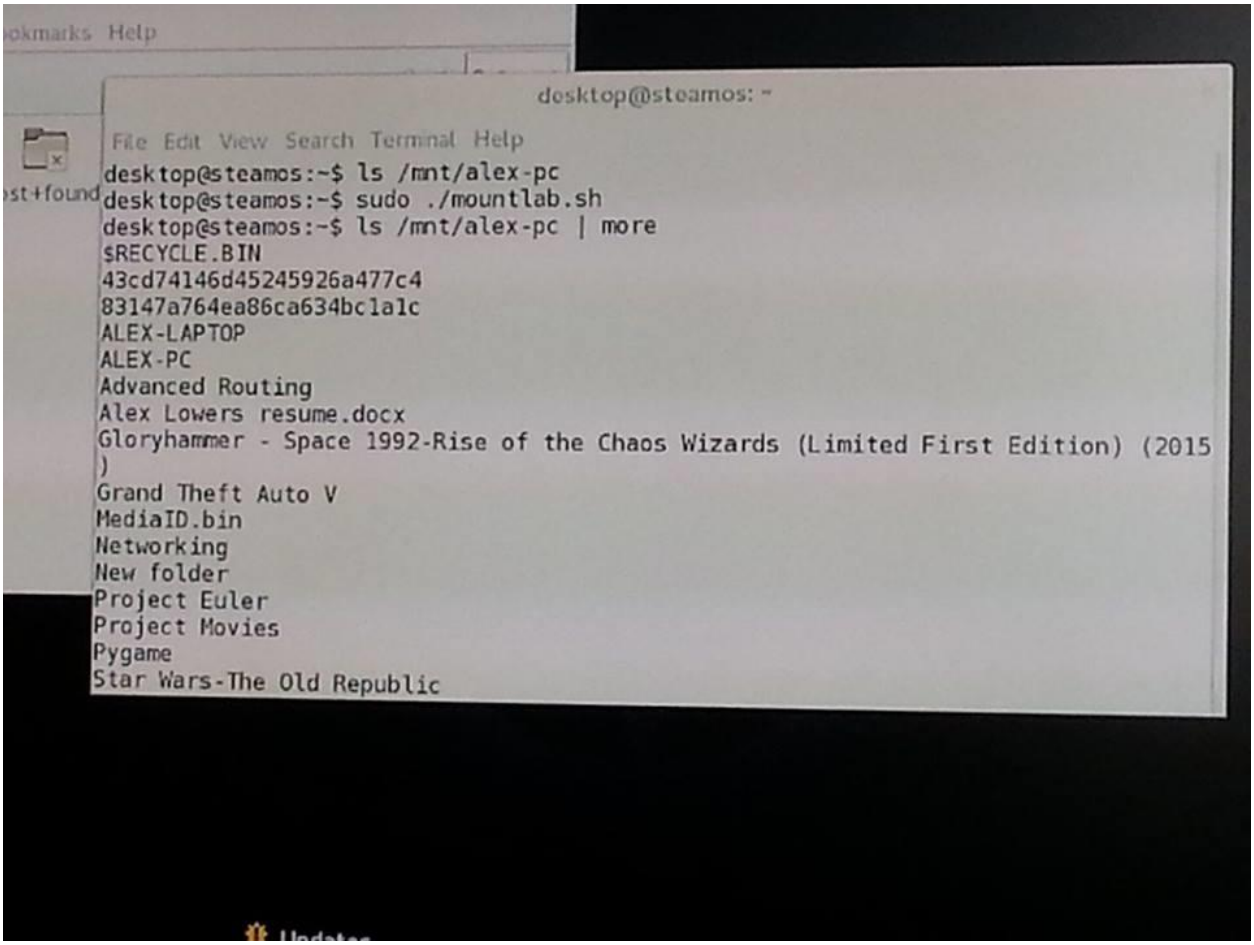
Also, the management SVIs of the switches pinged and were pinged to make sure the switches could be reached. The following matrix displays the results of the pings:

From	Alex-PC (10.0.0.2)	Steamos (10.0.0.70)	Laptop (10.0.0.134)	L2 Management SVI (10.0.0.194)	L3 Management SVI (10.0.0.193)
Alex-PC (10.0.0.2)	Success	Success	Success	Success	Success
Steamos (10.0.0.70)	Success	Success	Fail	Success	Success
Laptop (10.0.0.134)	Success	Fail	Success	Fail	Fail
L2 Management SVI (10.0.0.194)	Success	Success	Fail	Success	Success
L3 Management SVI (10.0.0.193)	Success	Success	Fail	Success	Success

The failures made sense because hosts on VLAN 30 were not supposed to be able to communicate with hosts on VLAN 20 or 40, only hosts on VLAN 10.

Software Testing

Within the LAN, the first configuration to test was the mount point. If the client did not mount the server's share properly, VLC would not have any media to stream. Once the *mount* command was entered, either from the terminal or by running the mountlab.sh script, the command *ls /mnt/alex-pc* was entered. This listed the contents of the mount point, which matched the contents of the server's shared drive. This confirmed the drive mounted properly.

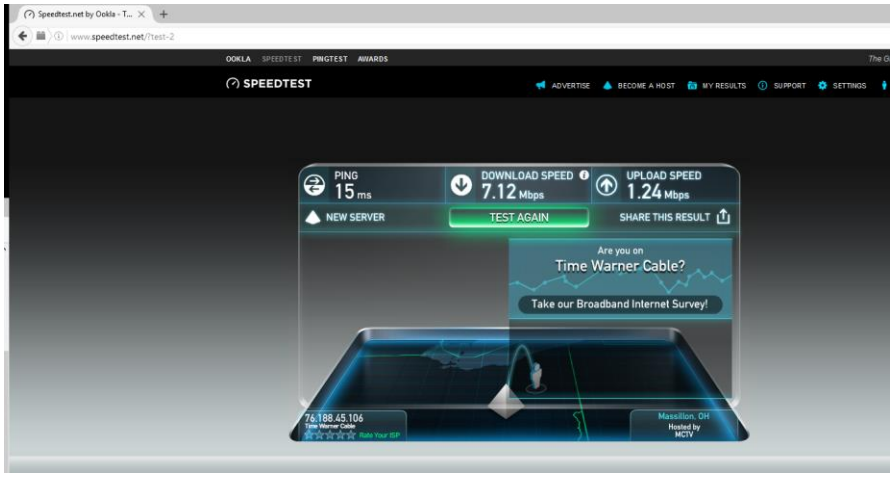


Above, the terminal displays the results of running the *ls* command before and after mounting the share.

Steam tests connectivity by itself every time it starts up. When the client first booted up after the *ip forward-protocol* and *ip helper-address* commands were entered, on the bottom right

of the screen a message appeared saying “Alex-PC is available for streaming”. Once this message appeared, the client was able to stream games from the server.

Plex was ready to stream once the library was able to be viewed on the phone app. See “Project Description” for details about the installation process for the Plex phone app. Upload speed was measured however on *speedtest.net* on the server as that has an impact on streaming performance over the internet.



Stream Testing

With the computers connected and media accessible from the server to both the client and the phone, it was time to begin streaming. For music, a CD rip of *1992: Rise of the Chaos Wizards* by Gloryhammer was streamed across all topologies and sound quality was rated on a scale of 1 to 10, 10 being the highest possible score. Scoring was based on audio clarity and range. For movies, Louis CK's special *Live at the Comedy Store* because the file was DRM free and in 1080p. Video and audio qualities were both rated on a scale of 1 to 10 as well. Audio again was scored based on audio clarity and range while video was scored based on framerate, smoothness, and whether or not it was synched with the audio. Lastly, three video games were selected of varying graphical quality. The game *Papers, Please* is a very graphically simple game with slow paced gameplay. *Civilization V* has medium quality graphics so would be more sensitive to network latency. Being a turn-based game, gameplay would not suffer as much from some latency, however. Lastly, the newly released *Tom Clancy's: The Division* was used as an example of high end gaming. The graphical fidelity is quite high and the third-person shooter gameplay is not very tolerant of network lag. Across all topologies, scores were assigned to the video, audio, and gameplay aspects of each game. Scores were again on a scale of 1 to 10. Video scores were based on framerate, smoothness, and if it is synched with the audio. Audio was scored based on clarity and range, like music and video tests. Lastly, gameplay was based on control delay, network delay, and overall playability.

Framerate information from video games while streaming was provided by Steam. It provides streaming statistics on an overlay as the game is being streamed. Video and audio quality however was more subjective and judgment was based on third parties viewing and listening to the audio.

Below are the computer's specs. Obviously, the server was much more powerful. The

client, Steamos, would struggle with new games such as *The Division* were it to render the game by itself.

Alex-PC	**Steamos**
Windows 10 AMD FX 8320 @ 3.5GHz NVidia GeForce GTX 960 4GB GDDR5 16GB DDR3	Steam OS Intel Core 2 Duo @ 2.8GHz NVidia GeForce GTS 450 1GB GDDR5 4GB DDR3

Baseline

In order to put the rest of the testing results into context, a baseline needed to be produced on the server alone. All media were played on the server first and rated on the same scale using VLC media player and Steam, just like the client.

Music:

1992: Rise of the Chaos Wizards features cheesy power metal anthems about space wizards and warriors fighting to save the galaxy in the distant future (or is it the past?). Not only is it an enjoyable listen, the songs vary quite a bit. The album features songs of different tempos and keys, allowing for a full range of different sounds. Listening to it straight off the hard drive was as good as it was going to get, so the quality was 10/10.

Movie:

Despite being only about an hour long, *Live at the Comedy Store* is more than 3 GB. That data density means there is a lot of video and audio needing to be sent over the network to stream an enjoyable experience. 1080p means any drop in video quality would be noticeable. There was no drop in video or audio quality when read directly off the hard though, as expected, so both are 10/10 for the baseline.

Video Games:

Papers, Please is a small indie game that features a border patrol agent working for a totalitarian government. The player, as the agent, looks through people's papers and based on varying criteria, allows or denies them into the country. The graphics are simple, reminiscent of the Nintendo Entertainment System so the GTX 960 had no trouble rendering the game. Audio likewise was not very advanced. Overall, video, audio, and gameplay were silky smooth on the

computer, though that is not difficult to achieve. This game got a 10/10 across the board for the baseline. *Civilization V* features addictive turn-based gameplay, as the player seeks to expand their territories, advance their cities and dominate the world. Featuring various biomes, buildings, and units, the game presents a modest graphical challenge that the server handled well. Again, this game received 10/10 across the board for quality. Lastly, *Tom Clancy's: The Division* is a newly released third-person shooter with incredible graphics. While the settings were only set to about the "medium" fidelity with in the game, it was still gorgeous and able to be rendered at or around 60 frames per second. Controls were responsive, as they needed to be in a shooter. Sound effects were clear and very immersive, giving punch and weight to the guns. Overall, graphic fidelity made video take a small hit to 9/10 but the audio and gameplay were both 10/10.

	Music	**Movie**		**Video Game**								
				Papers, Please			*Civ V*			*The Division*		
	Audio	Video	Audio	Video	Audio	Play	Video	Audio	Play	Video	Audio	Play
Base-line	10	10	10	10	10	10	10	10	10	9	10	10

Fully Wired Topology

In this setup, both the server and client were connected to the network via cat 5e cable. Music and movie streaming was conducted using VLC media player and video games were streamed through Steam.

Music:

When streamed over the fully wired topology, the album sounded just like it was coming off the CD. The drum beats were powerful and guitar riffs were crystal clear. From the bass lines to the falsetto vocals, the network was able to handle the full range of sound from the album. Overall, testers found audio quality of the music streamed over the network to be a 10/10.

Movie:

When streamed over the fully wired topology, the video quality was excellent. Frame rate stayed solid throughout the piece. Skipping to different parts of the special produced no lag or synch issues. No visual artifacts were found and overall quality was good. Audio clarity was likewise great, as if watching directly off the hard drive. Video quality and audio quality were both 10/10

Video games:

When streamed over the fully wired topology, *Papers, Please* had no framerate issues. Audio stayed synched throughout the experience and controls were responsive. Audio, video, and gameplay were all 10/10. They were exactly as though playing on the computer. Similarly, *Civilization V* was an enjoyable experience. Scrolling across the landscape quickly did not impose any screen tearing or lag. Controls were very responsive and audio and video were both like playing on the computer. Lastly, *The Division* took a slight performance hit. While 60 fps

was easily achievable on the server, when streaming the client version stayed around 30 fps. This was still very playable but was noticeable. Controls though were not affected. There was no recognizable delay between clicking the mouse button, the character on screen firing a round, and the shot playing through the speakers. Swinging the camera around did not invoke any screen tearing or frame rate drops. Overall, video was 8/10, while audio and gameplay were a 10/10.

	Music	**Movie**		**Video Game**								
				Papers, Please			*Civ V*			*The Division*		
	Audio	Video	Audio	Video	Audio	Play	Video	Audio	Play	Video	Audio	Play
Base-line	10	10	10	10	10	10	10	10	10	9	10	10
Fully Wired	10	10	10	10	10	10	10	10	10	8	10	10

Wireless/Wired Topology

In this configuration, the client was moved to the wireless network Project 20 while the server stayed connected via cat 5e cable. Streaming was performed using the same methods.

Music:

Again, quality while streaming was very good. Sound was clear and the full ranges of the songs were produced. The introduction of wireless networking to the streaming path did not impact performance in any discernable way. Quality was 10/10

Movie:

When viewing the piece from the beginning all the way through, no loss in quality could be seen. Audio was crisp, clear, and synched well to the video. However, when VLC was used to skip ahead in the video, a delay between streaming video and audio data caused them to go out of synch. The only remedy to this was closing and reopening the file. Also some audio in the very high range became very static-y. Because of these issues, video and audio quality took a hit down to 7/10.

Video Games:

Papers, Please did not suffer at all from the addition of wireless. Graphic fidelity, audio clarity, and gameplay all stayed at 10/10. *Civ V* had some issues however. When sweeping the camera across the landscape, there was some noticeable screen tearing. Likewise, cycling through units caused some delay and drop in frame rate. Because of its turn-based gameplay, the experience did not suffer from these issues. Audio remained unaffected. Overall, video was a 7/10, while gameplay and audio were 10/10. Lastly, *The Division* had similar issues to *Civ.* Moving the camera around caused screen tearing and while walking kept the normal frame rate

of around 30 fps, sprinting caused a noticeable drop. Audio was not affected, but gameplay did suffer. The drops in frame rate also corresponded to intense moments of action, with many enemies on screen. The video quality drop to 6/10 also caused gameplay to drop to 7/10. Audio maintained its quality at 10/10.

	Music	**Movie**		**Video Game**								
				Papers, Please			*Civ V*			*The Division*		
	Audio	Video	Audio	Video	Audio	Play	Video	Audio	Play	Video	Audio	Play
Base-line	10	10	10	10	10	10	10	10	10	9	10	10
Fully Wired	10	10	10	10	10	10	10	10	10	8	10	10
Wireless/ Wired	10	7	7	10	10	10	7	10	10	6	10	7

Wired/Wireless Topology

With this topology, the client and server connections were basically swapped. Steamos was attached again to the network with a cat 5e cable and the server was connected wirelessly to the Project Lab 10 network. Streaming methods stayed the same.

Music:

Like the previous tests, music was unaffected by the topology change. No discernable drop of quality occurred, so this scored 10/10 again.

Movie:

The issues that plagued the video stream in the previous topology plagued this topology as well. The same problem with skipping ahead occurred. While watching the video straight through caused very little issue, save for the audio static, the scores stayed the same for video and audio quality: 7/10 for both.

Video Games:

Papers again proved low graphical quality has low streaming requirements. Video, audio, and gameplay all stayed consistent with the baseline in this topology. *Civ* continued to experience issues the screen tearing, though it did not seem any worse than the previous topology. *The Division* had the same fate as well. Screen tearing and frame rate issues continued. Overall there was no discernable difference in video, audio, or gameplay quality when one device was connected wirelessly and one was connected via cable.

	Music	**Movie**		**Video Game**								
				Papers, Please			*Civ V*			*The Division*		
	Audio	Video	Audio	Video	Audio	Play	Video	Audio	Play	Video	Audio	Play
Base-line	10	10	10	10	10	10	10	10	10	9	10	10
Fully Wired	10	10	10	10	10	10	10	10	10	8	10	10
Wireless/ Wired	10	7	7	10	10	10	7	10	10	6	10	7
Wired/ Wireless	10	7	7	10	10	10	7	10	10	6	10	7

Fully Wireless Topology

In this topology both the client and server were connected to the network wirelessly. Streaming methods stayed the same as before.

Music:

For the last topology, *1992* had no surprises. No drop in quality was observed and it maintained its 10/10 rating.

Movie:

The movie quality took a big hit in this topology. Problems skipping followed into this topology but on top of that, artifacts would sometimes form in the picture. Occasionally the artifacts would start occurring as soon as the video began, accompanied by the audio static, and the only remedy was to close the player and reopen it. Overall this topology caused a large drop in audio and video quality, earning it a 4/10 for both.

Video Games:

Papers continued its tradition of being unaffected by topology changes. Its low fidelity and simple gameplay withstood all topology changes and the game maintained a solid 10/10 in all categories. *Civ* did not fare so well. Screen tearing issues increased and delays when selecting units became more noticeable. Again, being turn-based, it did not impact gameplay too much. While video dropped to 6/10, audio and gameplay stayed steady at 10/10. *The Division* suffered the worst. Frame rate decreased even when walking, and running caused it to go as low as 10 fps. Gameplay was impacted as well. Controls felt floaty; there was a noticeable delay between when a button was pressed and the character on screen performed the corresponding action. There was a gap between when the mouse button was pressed and when the character fired a round, for

instance. The sound effect for the gun even had a delay. In competitive, player-versus-player environments, this delay is not just an inconvenience; it can be a death sentence. While fidelity stayed high, the frame rate drop caused the video rating to dive to 3/10. Audio, despite being delayed, stayed relatively smooth and did not have the characteristic static of the movie streams, so it stayed at 10. Gameplay dropped the most, as the game was almost unplayable; this area scored 2/10.

	Music	**Movie**		**Video Game**								
				Papers, Please			*Civ V*			*The Division*		
	Audio	Video	Audio	Video	Audio	Play	Video	Audio	Play	Video	Audio	Play
Base-line	10	10	10	10	10	10	10	10	10	9	10	10
Fully Wired	10	10	10	10	10	10	10	10	10	8	10	10
Wireless/ Wired	10	7	7	10	10	10	7	10	10	6	10	7
Wired/ Wireless	10	7	7	10	10	10	7	10	10	6	10	7
Fully Wireless	10	4	4	10	10	10	6	10	10	3	10	2

Remote Streaming

The last stream to test was Plex internet streaming. The Galaxy SIII was connected to Sprint's 4G network and the server was connected to the network using an Ethernet cable. Upload speeds were previously measured at 1.24 Mbps.

Music:

1992: Rise of the Chaos Wizards suffered more from the quality of the phone's speakers than from the streaming quality of Plex. Low bass tones were not produced very well given the small size of the speakers. There was also a slight buzzing that could be heard in the background. This may have been a symptom of the upload speed. Nevertheless, Plex proved to be serviceable for streaming music and received a score of 7/10

Movie:

The upload speed affected the quality of the Louis CK stream as well. Despite the file being in 1080p, or perhaps because of that high fidelity, the streaming quality had to drop all the way down to 96kbps to ensure smooth playback. This had a noticeable impact on video quality. The screen was very low resolution, like a Youtube video from 2006. Audio synched well and didn't have that same buzzing that the song did. Ultimately, while video quality was a 6/10 and audio was a 9/10, the problems stemmed from the low upload speed more than from a fault in the software.

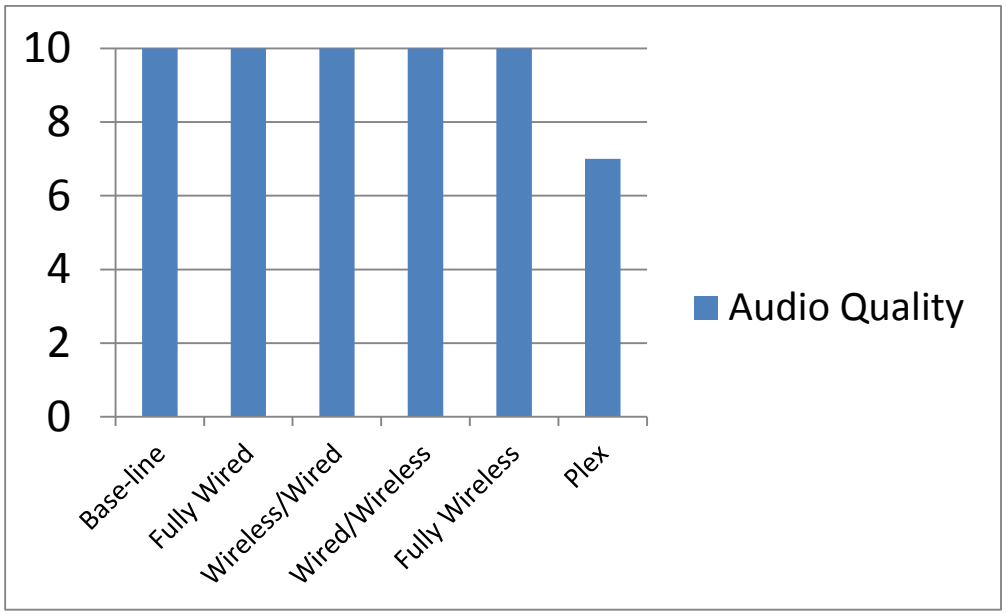
	Music	**Movie**		**Video Game**								
				Papers, Please			*Civ V*			*The Division*		
	Audio	Video	Audio	Video	Audio	Play	Video	Audio	Play	Video	Audio	Play
Base-line	10	10	10	10	10	10	10	10	10	9	10	10
Fully Wired	10	10	10	10	10	10	10	10	10	8	10	10
Wireless/ Wired	10	7	7	10	10	10	7	10	10	6	10	7
Wired/ Wireless	10	7	7	10	10	10	7	10	10	6	10	7
Fully Wireless	10	4	4	10	10	10	6	10	10	3	10	2
Plex over Net	7	6	9	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Conclusion

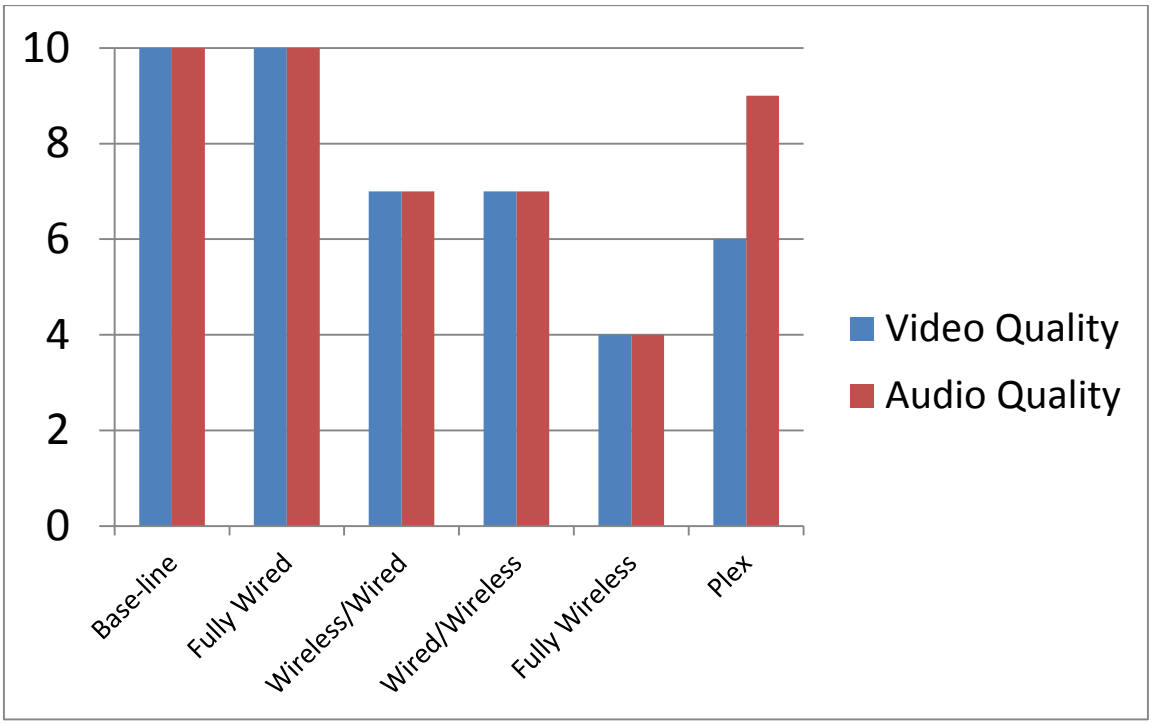
Final Scores

	Music	Movie		Video Game								
				Papers, Please			*Civ V*			*The Division*		
	Audio	Video	Audio	Video	Audio	Play	Video	Audio	Play	Video	Audio	Play
Base-line	10	10	10	10	10	10	10	10	10	9	10	10
Fully Wired	10	10	10	10	10	10	10	10	10	8	10	10
Wireless/ Wired	10	7	7	10	10	10	7	10	10	6	10	7
Wired/ Wireless	10	7	7	10	10	10	7	10	10	6	10	7
Fully Wireless	10	4	4	10	10	10	6	10	10	3	10	2
Plex over Net	7	6	9	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

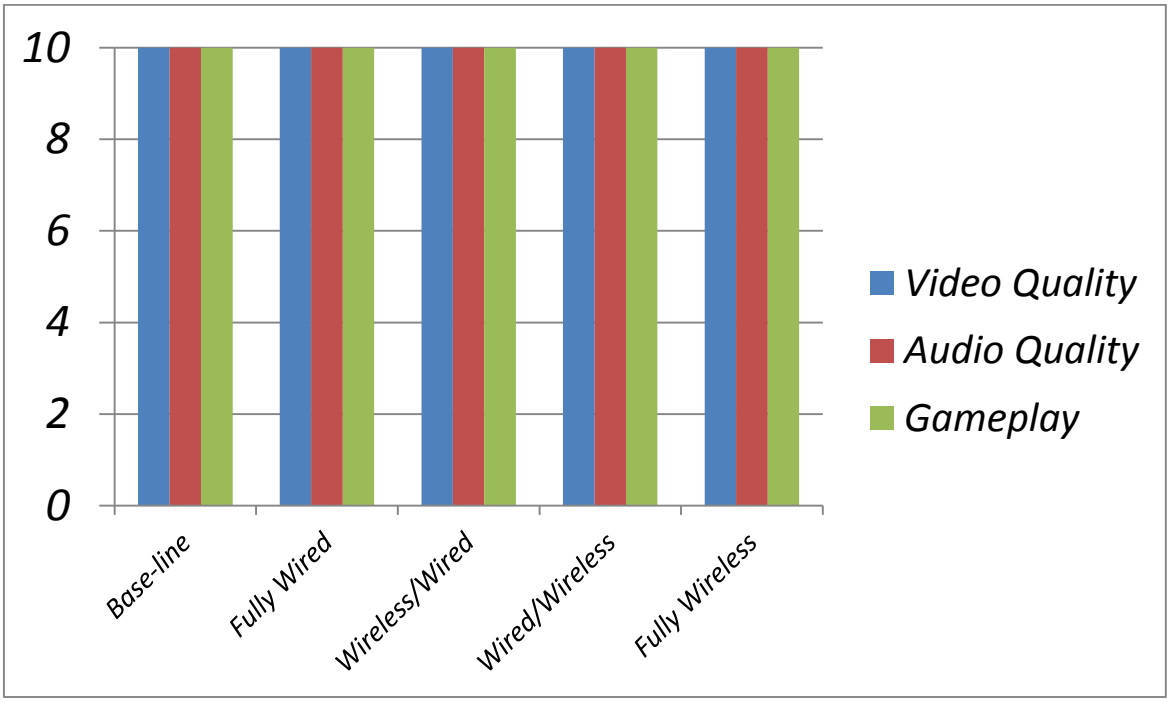
Media Quality (Music) vs Topology



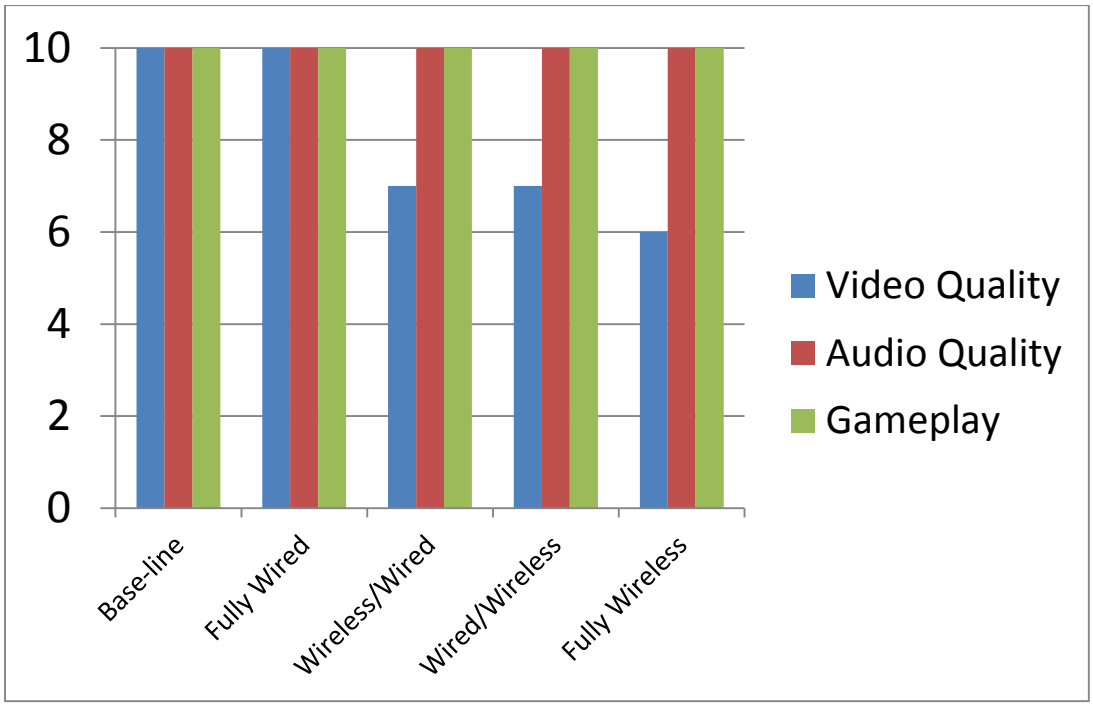
Media Quality (Movie) vs Topology



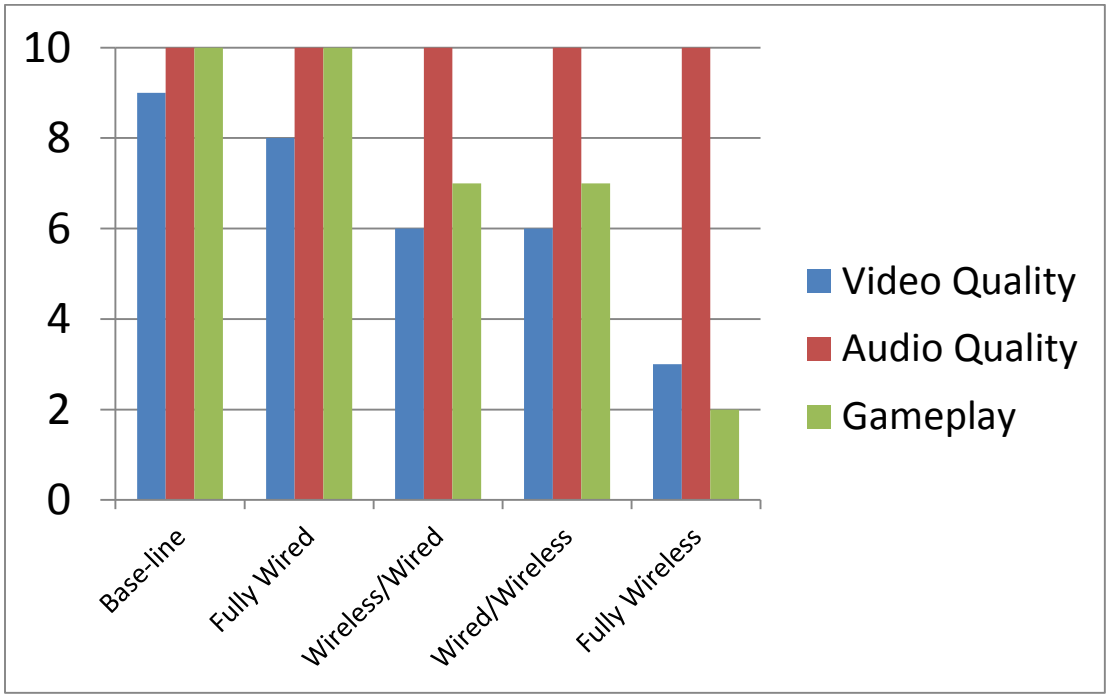
Media Quality (*Papers, Please*) vs Topology



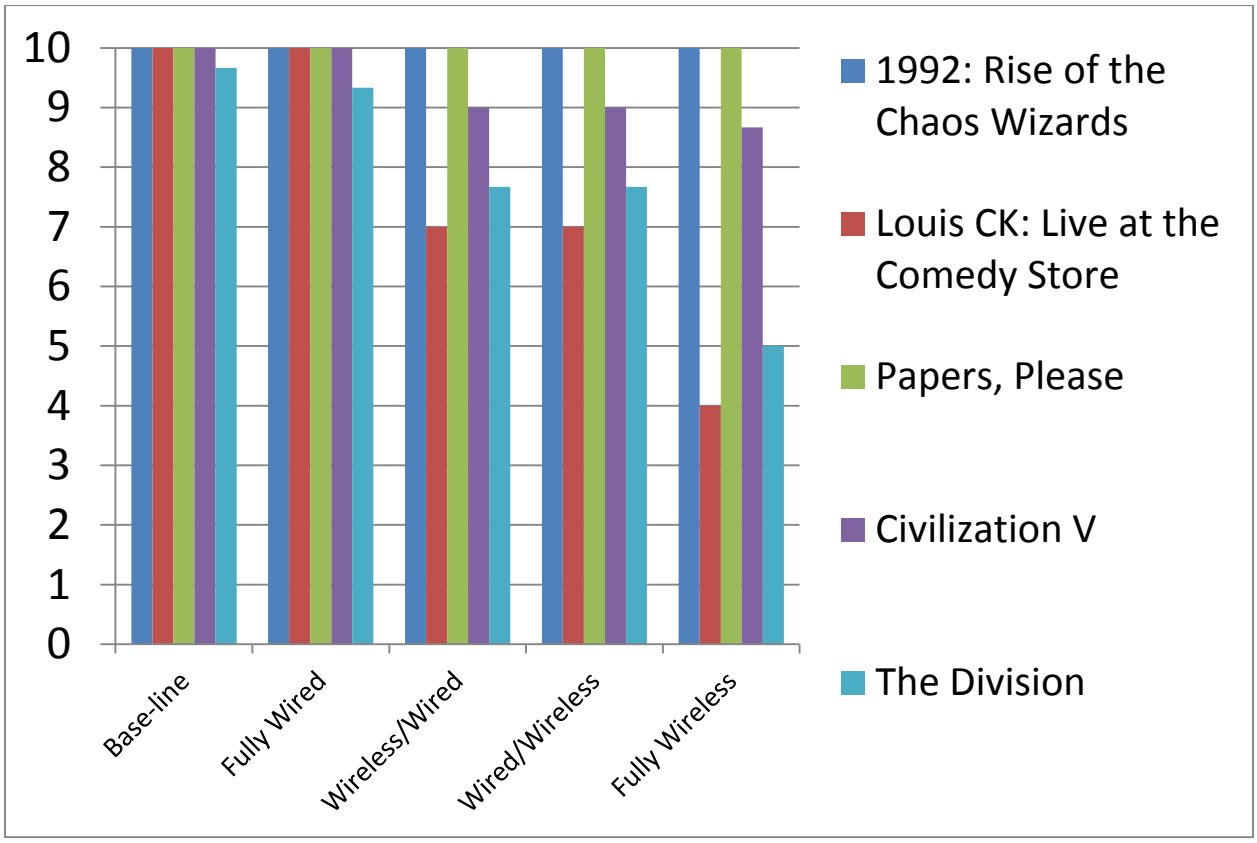
Media Quality (*Civilization V*) vs Topology



Media Quality (*The Division*) vs Topology



Media Quality (Mean of Criteria) vs Topology



Music did not seem to be impacted at all across the topologies. The low bandwidth requirements mean that even in a fully wireless topology, audio quality did not suffer. Likewise, low resolution games such as *Papers, Please* were unaffected by the change in topology. While *Civ* did notice a drop in video quality, this did not impact gameplay much. The only media that the topology severely impacted the experience with was the 1080p video and high resolution video game. Based on these results, a fully wireless network is not conducive to streaming video or playing intense video games. Overall, a fully wired topology provides the best experience across all media types and wireless should be avoided if at all possible.

The most surprising aspect about this result is that it holds true even when the Ethernet ports are slower than the theoretical maximum wireless transmission speed. Both switches had

100 Mbps Fast Ethernet ports and the access points as well as wireless NICs of both PCs were all using 802.11n with a theoretical maximum of 300 Mbps. Despite the boost to transmission speed, wireless still negatively impacted streaming performance.

For internet streaming, Plex was an easy to use tool. Ultimately, performance was hindered more by internet connection upload speed and the client device's hardware. For its simple installation and management though, Plex cannot be recommended highly enough as an internet streaming program.

L3 Running Config:

```
!
version 12.1
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname L3
!
enable secret 5 $1$KTIo$4RwP/PNFfp7SADj7tq2RE0
!
ip subnet-zero
ip routing
ip dhcp excluded-address 10.0.0.1 10.0.0.5
ip dhcp excluded-address 10.0.0.65 10.0.0.69
ip dhcp excluded-address 10.0.0.129 10.0.0.133
ip dhcp excluded-address 10.0.0.193 10.0.0.194
!
ip dhcp pool SERVER_DHCP_POOL
   network 10.0.0.0 255.255.255.192
   default-router 10.0.0.1
   dns-server 8.8.8.8
!
```

```
ip dhcp pool CLIENT_DHCP_POOL
   network 10.0.0.64 255.255.255.192
   default-router 10.0.0.65
   dns-server 8.8.8.8
!
ip dhcp pool GUEST_DHCP_POOL
   network 10.0.0.128 255.255.255.192
   default-router 10.0.0.129
   dns-server 10.0.0.129
!
ip name-server 8.8.8.8
ip ssh time-out 120
ip ssh authentication-retries 3
!
spanning-tree mode pvst
spanning-tree extend system-id
!
!
!
!
!
interface Port-channel1
 switchport trunk encapsulation dot1q
 switchport trunk native vlan 40
 switchport mode trunk
```

```
!
interface FastEthernet0/1
 switchport trunk encapsulation dot1q
 switchport trunk native vlan 40
 switchport mode trunk
 channel-group 1 mode active
!
interface FastEthernet0/2
 switchport mode dynamic desirable
 shutdown
!
interface FastEthernet0/3
 switchport trunk encapsulation dot1q
 switchport trunk native vlan 40
 switchport mode trunk
 channel-group 1 mode active
!
interface FastEthernet0/4
switchport mode dynamic desirable
 shutdown
!
interface FastEthernet0/5
 switchport trunk encapsulation dot1q
 switchport trunk native vlan 40
 switchport mode trunk
```

```
 channel-group 1 mode active
!
interface FastEthernet0/6
switchport mode dynamic desirable
 shutdown
!
interface FastEthernet0/7
 switchport trunk encapsulation dot1q
 switchport trunk native vlan 40
 switchport mode trunk
 channel-group 1 mode active
!
interface FastEthernet0/8
switchport mode dynamic desirable
 shutdown
!
interface FastEthernet0/9
 switchport trunk encapsulation dot1q
 switchport trunk native vlan 40
 switchport mode trunk
 channel-group 1 mode active
!
interface FastEthernet0/10
 switchport mode dynamic desirable
 shutdown
```

```
!
interface FastEthernet0/11
 switchport trunk encapsulation dot1q
 switchport trunk native vlan 40
 switchport mode dynamic desirable
 channel-group 1 mode active
!
interface FastEthernet0/12
 switchport mode dynamic desirable
 shutdown
!
interface FastEthernet0/13
 switchport trunk encapsulation dot1q
 switchport trunk native vlan 40
 switchport mode dynamic desirable
 channel-group 1 mode active
!
interface FastEthernet0/14
 switchport mode dynamic desirable
 shutdown
!
interface FastEthernet0/15
 switchport trunk encapsulation dot1q
 switchport trunk native vlan 40
 switchport mode dynamic desirable
```

```
 channel-group 1 mode active
!
interface FastEthernet0/16
 switchport mode dynamic desirable
 shutdown
!
interface FastEthernet0/17
 switchport mode dynamic desirable
 shutdown
!
interface FastEthernet0/18
 switchport mode dynamic desirable
 shutdown
!
interface FastEthernet0/19
 switchport mode dynamic desirable
 shutdown
!
interface FastEthernet0/20
 switchport mode dynamic desirable
 shutdown
!
interface FastEthernet0/21
 switchport mode dynamic desirable
 shutdown
```

```
!
interface FastEthernet0/22
 switchport mode dynamic desirable
 shutdown
!
interface FastEthernet0/23
 switchport mode dynamic desirable
 shutdown
!
interface FastEthernet0/24
 switchport mode dynamic desirable
 shutdown
!
interface FastEthernet0/25
 switchport mode dynamic desirable
 shutdown
!
interface FastEthernet0/26
 switchport mode dynamic desirable
 shutdown
!
interface FastEthernet0/27
 switchport mode dynamic desirable
 shutdown
!
```

```
interface FastEthernet0/28
 switchport mode dynamic desirable
 shutdown
!
interface FastEthernet0/29
 switchport mode dynamic desirable
 shutdown
!
interface FastEthernet0/30
 switchport mode dynamic desirable
 shutdown
!
interface FastEthernet0/31
 switchport mode dynamic desirable
 shutdown
!
interface FastEthernet0/32
 switchport mode dynamic desirable
 shutdown
!
interface FastEthernet0/33
 switchport mode dynamic desirable
 shutdown
!
interface FastEthernet0/34
```

```
 switchport mode dynamic desirable
 shutdown
!
interface FastEthernet0/35
 switchport mode dynamic desirable
 shutdown
!
interface FastEthernet0/36
 switchport mode dynamic desirable
 shutdown
!
interface FastEthernet0/37
 switchport mode dynamic desirable
 shutdown
!
interface FastEthernet0/38
 switchport mode dynamic desirable
 shutdown
!
interface FastEthernet0/39
 switchport mode dynamic desirable
 shutdown
!
interface FastEthernet0/40
 switchport mode dynamic desirable
```

```
 shutdown
!
interface FastEthernet0/41
 switchport mode dynamic desirable
 shutdown
!
interface FastEthernet0/42
 switchport mode dynamic desirable
 shutdown
!
interface FastEthernet0/43
 switchport mode dynamic desirable
 shutdown
!
interface FastEthernet0/44
 switchport mode dynamic desirable
 shutdown
!
interface FastEthernet0/45
 switchport mode dynamic desirable
 shutdown
!
interface FastEthernet0/46
 switchport mode dynamic desirable
 shutdown
```

```
!
interface FastEthernet0/47
 switchport mode dynamic desirable
 shutdown
!
interface FastEthernet0/48
 switchport mode dynamic desirable
 shutdown
!
interface GigabitEthernet0/1
 switchport mode dynamic desirable
 shutdown
!
interface GigabitEthernet0/2
 switchport mode dynamic desirable
 shutdown
!
interface Vlan1
 no ip address
 shutdown
!
interface Vlan10
 ip address 10.0.0.1 255.255.255.192
!
interface Vlan20
```

```
 ip address 10.0.0.65 255.255.255.192
 ip helper-address 10.0.0.2
!
interface Vlan30
 ip address 10.0.0.129 255.255.255.192
 ip access-group 100 in
!
interface Vlan40
 ip address 10.0.0.193 255.255.255.192
!
ip classless
ip forward-protocol udp 27036
ip route 0.0.0.0 0.0.0.0 10.0.0.4
ip http server
!
access-list 100 permit ip 10.0.0.128 0.0.0.63 10.0.0.0 0.0.0.63
!
line con 0
 password cisco
 logging synchronous
 login
line vty 0 15
 password cisco
 login
!
```

!

end

L2 Running Config

```
!
version 12.1
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname L2
!
enable secret 5 $1$p.7W$Lyhk51JUDHRkTlH3j.zu90
!
ip subnet-zero
!
ip ssh time-out 120
ip ssh authentication-retries 3
!
spanning-tree mode pvst
no spanning-tree optimize bpdu transmission
spanning-tree extend system-id
!
!
!
!
```

```
interface Port-channel1
 switchport trunk native vlan 40
 switchport mode trunk
 flowcontrol send off
!
interface FastEthernet0/1
 switchport trunk native vlan 40
 switchport mode trunk
 channel-group 1 mode active
!
interface FastEthernet0/2
 switchport trunk native vlan 40
 switchport mode trunk
 channel-group 1 mode active
!
interface FastEthernet0/3
 switchport trunk native vlan 40
 switchport mode trunk
 channel-group 1 mode active
!
interface FastEthernet0/4
 switchport trunk native vlan 40
 switchport mode trunk
 channel-group 1 mode active
!
```

```
interface FastEthernet0/5
 switchport trunk native vlan 40
 switchport mode trunk
 channel-group 1 mode active
!
interface FastEthernet0/6
 switchport trunk native vlan 40
 switchport mode trunk
 channel-group 1 mode active
!
interface FastEthernet0/7
 switchport trunk native vlan 40
 switchport mode trunk
 channel-group 1 mode active
!
interface FastEthernet0/8
 switchport trunk native vlan 40
 switchport mode trunk
 channel-group 1 mode active
!
interface FastEthernet0/9
 switchport access vlan 10
 switchport mode access
 spanning-tree portfast
!
```

```
interface FastEthernet0/10
 switchport access vlan 10
 switchport mode access
 spanning-tree portfast
!
interface FastEthernet0/11
 switchport access vlan 10
 switchport mode access
 spanning-tree portfast
!
interface FastEthernet0/12
 switchport access vlan 10
 switchport mode access
 spanning-tree portfast
!
interface FastEthernet0/13
 switchport access vlan 20
 switchport mode access
 spanning-tree portfast
!
interface FastEthernet0/14
 switchport access vlan 20
 switchport mode access
 spanning-tree portfast
!
```

```
interface FastEthernet0/15
 switchport access vlan 20
 switchport mode access
 spanning-tree portfast
!
interface FastEthernet0/16
 switchport access vlan 20
 switchport mode access
 spanning-tree portfast
!
interface FastEthernet0/17
 switchport access vlan 30
 switchport mode access
 spanning-tree portfast
!
interface FastEthernet0/18
 switchport access vlan 30
 switchport mode access
 spanning-tree portfast
!
interface FastEthernet0/19
 switchport access vlan 30
 switchport mode access
 spanning-tree portfast
!
```

```
interface FastEthernet0/20
 switchport access vlan 30
 switchport mode access
 spanning-tree portfast
!
interface FastEthernet0/21
 switchport access vlan 30
 switchport mode access
 spanning-tree portfast
!
interface FastEthernet0/22
 switchport access vlan 30
 switchport mode access
 spanning-tree portfast
!
interface FastEthernet0/23
 switchport access vlan 30
 switchport mode access
 spanning-tree portfast
!
interface FastEthernet0/24
 switchport access vlan 30
 switchport mode access
 spanning-tree portfast
!
```

```
interface Vlan1
 no ip address
 no ip route-cache
 shutdown
!
interface Vlan40
 ip address 10.0.0.194 255.255.255.192
 no ip route-cache
!
ip http server
!
line con 0
 password cisco
 logging synchronous
 login
line vty 0 15
 password cisco
 login
!
!
end
```

Project References

"Cisco IOS IP Application Services Command Reference - Ip Accounting through Ip Sctp Authenticate [Support]." Cisco Systems, Inc. Cisco IOS IP Application Services Command Reference. Cisco, n.d. Web. 23 Feb. 2016. <http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipapp/command/iap-cr-book/iap-i1.html#wp1413119578>

"How to I Create an Open Smb Share on a Windows 7 Box?" *Linux*. N.p., n.d. Web. 11 Feb. 2016. <http://superuser.com/questions/386782/how-to-i-create-an-open-smb-share-on-a-windows-7-box>

LinusTechTips. "Stream Your Home Media Files Away from Home." *YouTube*. YouTube, 13 Mar. 2016. Web. 13 Mar. 2016. <https://www.youtube.com/watch?v=GYf9m1Ck8u0>

"Share Files and Printers between Windows 7 and XP." *HowTo Geek RSS*. N.p., n.d. Web. 11 Feb. 2016. <http://www.howtogeek.com/howto/windows-7/share-files-and-printers-between-windows-7-and-xp/>

"Steam Community :: Guide :: [SteamOS] Adding Samba Shares for Music Libraries." *Steam Community :: Guide :: [SteamOS] Adding Samba Shares for Music Libraries*. N.p., n.d. Web. 11 Feb. 2016. <http://steamcommunity.com/sharedfiles/filedetails/?id=407004166>

"Stream across Subnets? :: Steam In-Home Streaming." Stream across Subnets? :: Steam In-Home Streaming. N.p., 4 Feb. 2015. Web. 23 Feb. 2016. <https://steamcommunity.com/groups/homestream/discussions/0/606068060815627379/>

"The Difference between CIFS and SMB." *Varonis Blog RSS*. N.p., 14 Feb. 2014. Web. 11 Feb. 2016. <https://blog.varonis.com/the-difference-between-cifs-and-smb/>

"VLC Player on SteamOS? :: Steam Universe." *VLC Player on SteamOS? :: Steam Universe*. N.p., 18 Oct. 2013. Web. 11 Feb. 2016.

<http://steamcommunity.com/groups/steamuniverse/discussions/1/810938810601778753/
>

"Watch Movies on SteamOS." *Steam Universe*. N.p., 18 Oct. 2014. Web. 11 Feb. 2016.
<https://steamcommunity.com/groups/steamuniverse/discussions/1/619568793940320490/>

"Windows - HOW-TO Share Folders over SMB Using Windows 7." HOW-TO Share Folders over SMB Using Windows 7. N.p., n.d. Web. 11 Feb. 2016.
<http://forum.kodi.tv/showthread.php?tid=63123>