

Creating a Small VoIP-supported Network using SIP

Drina Hobson

CIS Networking Senior Projects

CISS 490-001

April 13th, 2023

Janet Kropff

Project Plan

The original plan for my project was to create a small VoIP-supported hybrid network that supports SIP and implements advanced SIP configurations. I would be installing GNS3 and implementing a virtual router on GNS3 and downloading Cisco IOS for this router. I would connect this router to my computer using a Cloud on GNS3. My computer would act as a server between my virtual GNS3 topology and my physical topology. My physical topology would consist of a physical Cisco Catalyst 2960 switch, as well as two Cisco 6941 VoIP phones. To achieve this, I would need to install a loopback adapter onto my computer and give it a static IP address to connect to the virtual GNS3 Cloud. Once this topology is configured, I would install the SIP server onto my computer and have it connect to my router. Once the VoIP phones are registered onto the SIP server, I would then implement the advanced SIP configurations including multicast paging, voice message system & message waiting indicator, and music-on-hold.

- Weeks 1 & 2: Planning and researching
- Weeks 3-9: Constructing/cabling the physical topology, installing Cisco IOS on router, configuring hybrid topology using GNS3, installing SIP server
- Week 10: Implementing advanced VoIP features
- Weeks 11-13: Documentation, putting together binder, creating presentation, recording working project

Project Analysis

When starting to work on this project idea, I originally installed GNS3 and configured my topology with the virtual router and the cloud that would then connect to my PC. I installed the Cisco IOS needed for the router and was successful in connecting it to the cloud. However, when trying to connect the cloud to my PC, I was unsuccessful. I set the IP address of my cloud to the same subnet as my PC, but still didn't have any luck. I tried to install a loopback adapter on my computer as well and tried to connect the cloud to the loopback adapter, but my router was still unsuccessful in pinging my computer and vice versa. I utilized the GNS3 community forums to help me come up with a reason as to why I was having these issues and hopefully a solution. Since the GNS3 VM acts as a connection from the cloud to my computer and it was installed on VirtualBox, I tried to install VMWare to see if the VirtualBox software might be incompatible with my Windows IOS version. After installing VMWare and trying again to connect my cloud to my virtual router, I still had no luck. After consulting the forums again, I came to the conclusion that GNS3 typically does not work well with Windows 11 and is not currently optimized for it. Since I did not want to downgrade back to Windows 10, I thought that I would try to install a fully virtual topology instead.

For the fully virtual topology to work, I would still utilize GNS3 and connect the virtual router to the cloud, but instead of the cloud connecting to my computer, it would instead connect solely to the GNS3 VM. I would also have to install two other Windows 10 virtual machines and install the Cisco IP Communicator software on both. The Cisco IP Communicator would act as a virtual softphone and would, in theory, be able to call each other and would be configured with the telephony configurations via the virtual router. After doing some research into the Cisco IP Communicator software and how to acquire it, I quickly realized that this would not be viable either. The Cisco IP Communicator software was not being offered by Cisco anymore, and had been replaced by Jabber. Jabber may have been able to work with what I was attempting to do,

but I was unable to find any sort of tutorial or compatibility matrix regarding GNS3 and Jabber. And with the deadline creeping up on me, I finally decided it was in my best interest to do what I am most familiar with and work with completely physical topology.

After acquiring the necessary components - a physical router, physical switch, and two physical Cisco IP phones - I was able to get started on my project, in what felt like square one. I began to cable the topology and configure the basic commands on the router and everything was going smoothly. Once I got to configuring the SIP commands, I ran into the issue of the Cisco IP phone type not showing up in the configuration options. After doing some research, I came to the conclusion that I needed to install a different CME version onto the router via a TFTP server. Not only did I need to install the CME files, but I also needed to install the correct SIP files for my phone type since the CME version I installed only came with the SCCP files for my phones. Once these files were installed, I just needed to install the SIP server and register the IP phones to the SIP server and configure the advanced VoIP commands on the router. Configuring the multicast paging and the voice mail configurations were fairly straight forward. The router allowed for both of these commands so there wasn't any additional firmware that I had to install. However, I did not allow myself enough time to configure the music-on-hold. I failed to do enough research to realize that I would need to configure a MOH server before being able to actually configure the music-on-hold. However, I will still go over the configurations of music-on-hold and how to do it in my documentation. As well, the SIP server that I had chosen is not compatible with the 6941 Cisco IP phones anymore. I had tried to follow tutorials that would allow for the 3CX server to register 6941 IP phones, but these tutorials did not work either. I feel as if I still learned a lot about SIP and how it operates, and I feel as though I could configure it easily now that I've done research into the topic. My main takeaways from this project are that I need to do more research before attempting to configure or install certain things, and that I can't always learn things as I am doing them. Having knowledge before going into something like this would have been a lot more helpful than the methods I was attempting.

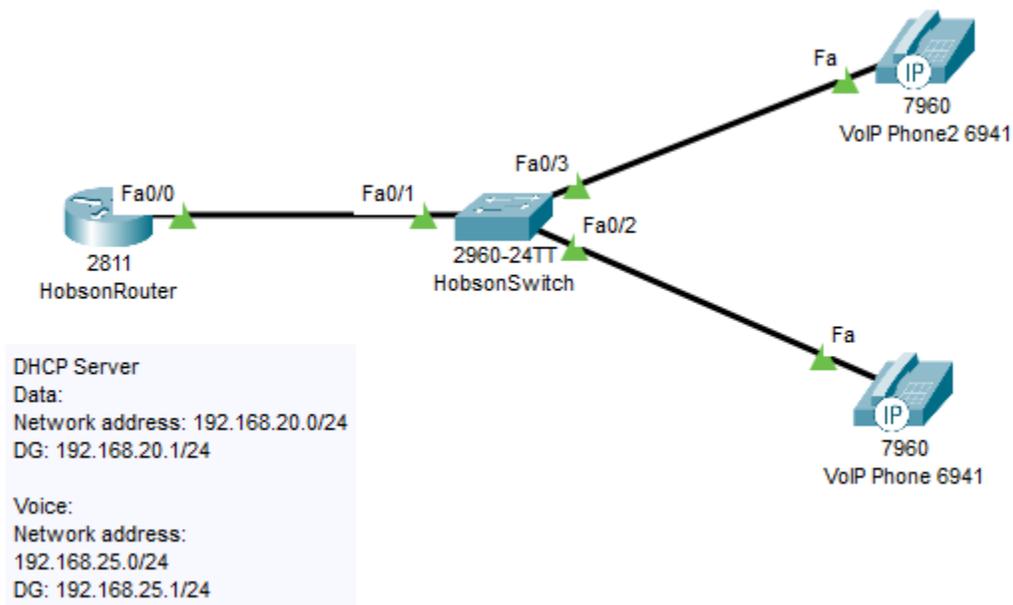
As well, I believe that I personally would achieve more if I had set stricter deadlines for myself. Since this was a semester-long project, I found myself procrastinating a lot and assuming that I could get anything that I needed done, done before the deadline. I was very wrong about this assumption. Overall, I learned a lot about virtualization, the SIP server, advanced configurations, and compatibility between products. These will all be things that I will take into account before undergoing a project like this in the future.

Project Description

Components Used

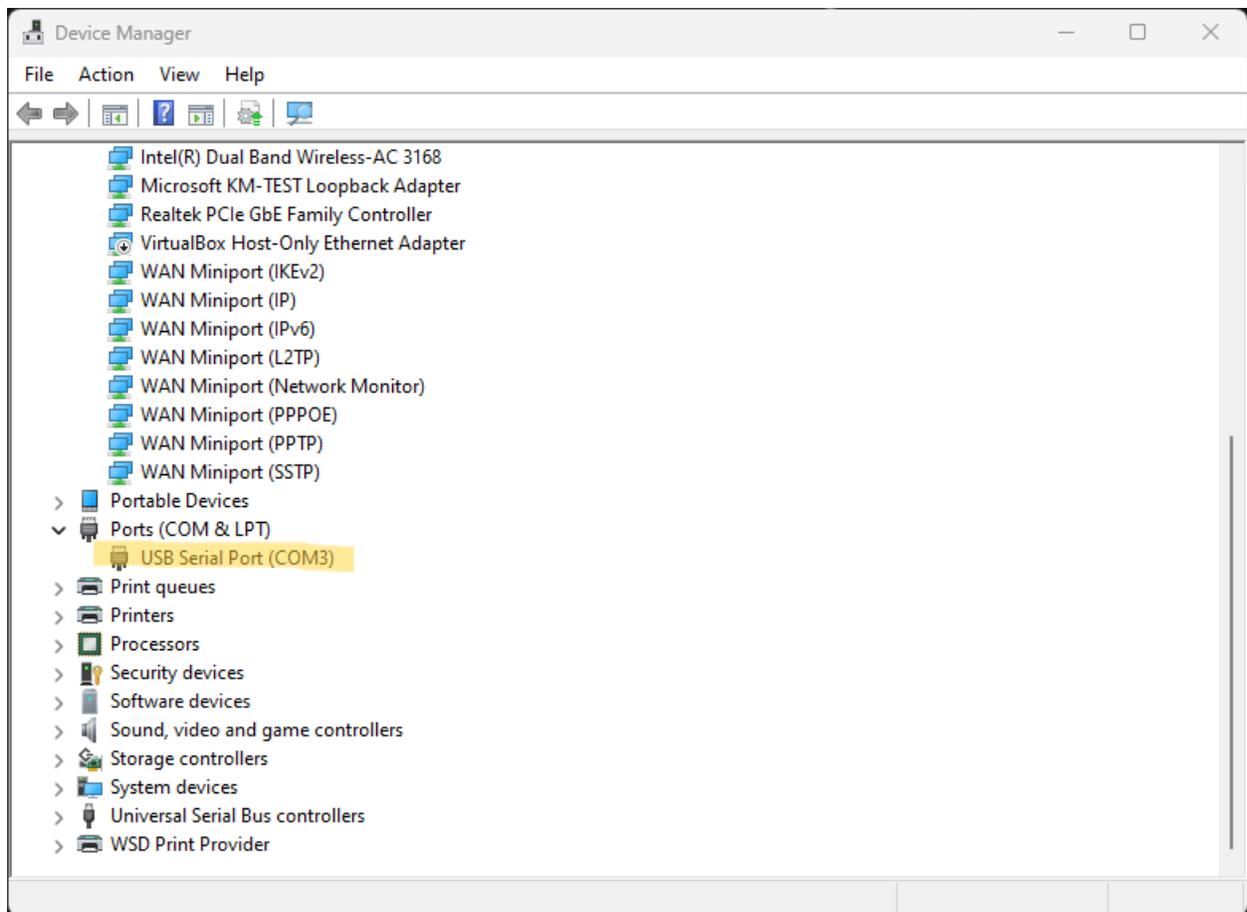
- Cisco 2811 that runs Cisco Unified Communications Manager Express 8.0 with Cisco IOS Software Release 15.1
- Cisco Catalyst 2960 PoE-24 port switch
- (2) Cisco 6941 SIP Phones

Simulation of physical topology

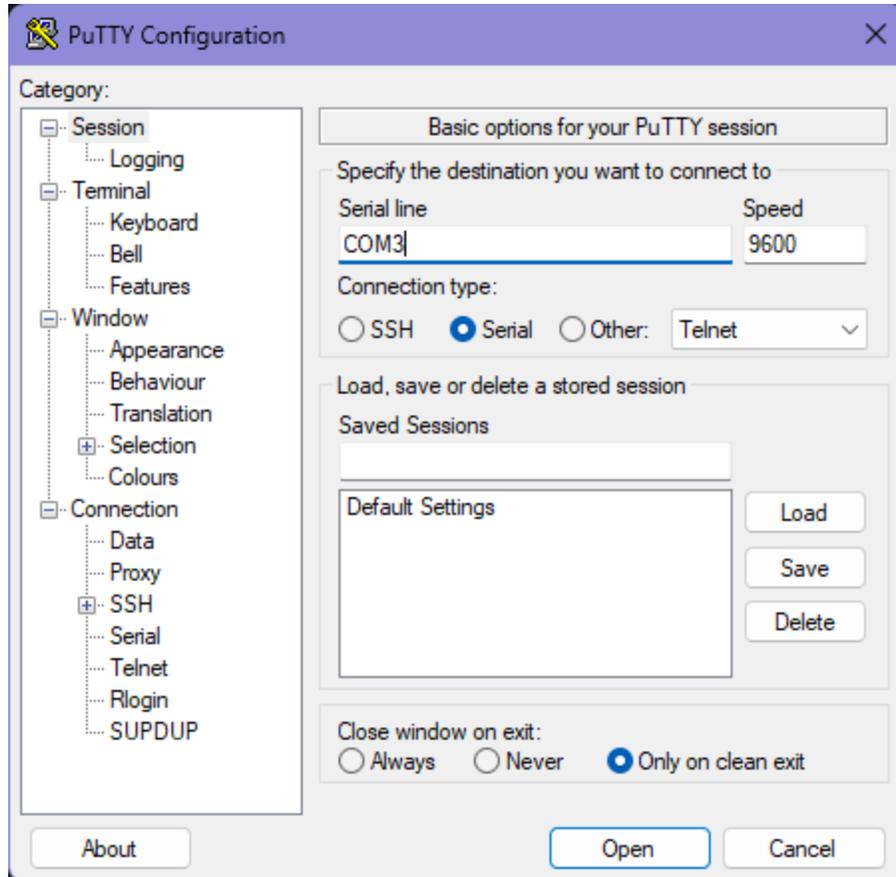


Installing PuTTY

The installation of PuTTY is very straightforward. I will have the link to the installer in the references section. Once PuTTY has been installed, open the service and select 'Serial'. By default, the serial line is set to COM1. However, this was not the case for my device. I had to navigate to my Device Manager > Ports to see which port my console cable was plugged into.



Once you've located the port that your console cable is plugged into, type that into PuTTY. It should look like this:

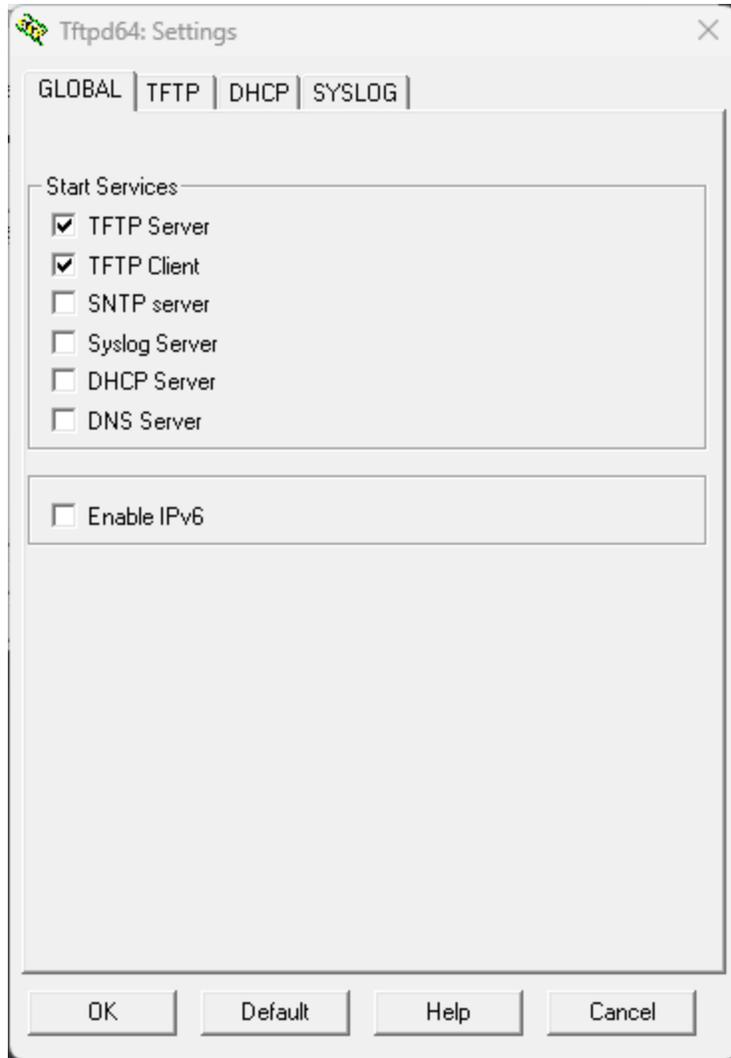


Click 'Open' whenever you are ready to begin configuring your devices.

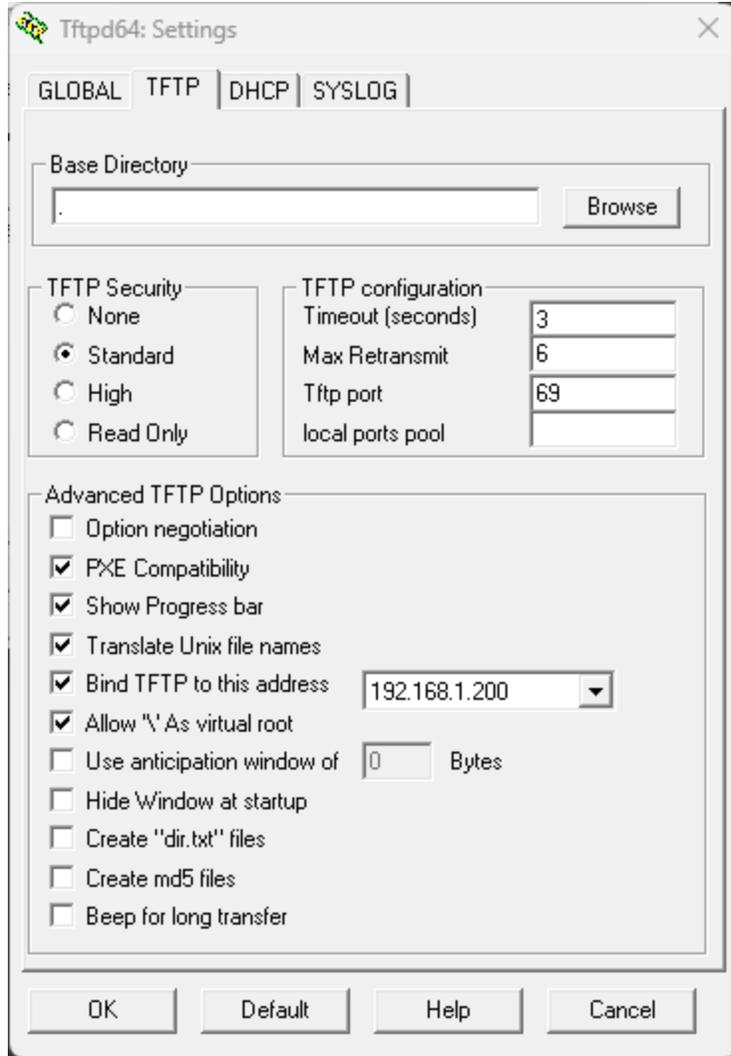
Installing TFTP Server

Since the 2811 router does not originally come with the CME 8.0 software or the SIP firmware for the 6941 VoIP phones, a TFTP server is needed to load these files onto the router. For this project, I used Tftpd64, an open-source TFTP server. The link for the Tftpd64 download will be in the references section. Once I downloaded the TFTP server onto my computer, I needed to change some settings in order for the router to communicate with the TFTP server. I changed the static IPv4 address of the interface to 192.168.1.200/24 and the default gateway to 192.168.1.100/24 - which will be the address to the router. Using PuTTY to enter the CLI of the router and entering terminal configuration mode, I entered the interface configuration by using the 'interface fa0/1' command, and then changed the IP address of this router port to

192.168.1.100/24 so that my computer was able to communicate with the router. Use a network cable and plug it into the computer's network port and into router port FastEthernet0/1. Open command prompt on the computer and enter in the command 'ipconfig' to ensure that the computer has received the static IPv4 address, then enter the command 'ping 192.168.1.100' to ensure that the computer can ping the router. Now open the PuTTY CLI and enter the command 'ping 192.168.1.200' to ensure that the router is able to ping the computer. After you've verified that both devices are able to ping each other, we will now configure the settings on the TFTP server. Once I opened the TFTP server, I clicked Settings > Global. I unchecked Syslog Server and DHCP Server and ensured that TFTP Server and TFTP Client were checked. The global settings should look like this:



Once I configured the global settings, then I will now configure the TFTP settings under the TFTP tab. Under the Advanced TFTP Options, I checked the 'Bind TFTP to this address' checkbox, which then allowed me to set the address to the address of my computer's interface - 192.168.1.200. The TFTP Settings should now look like this:

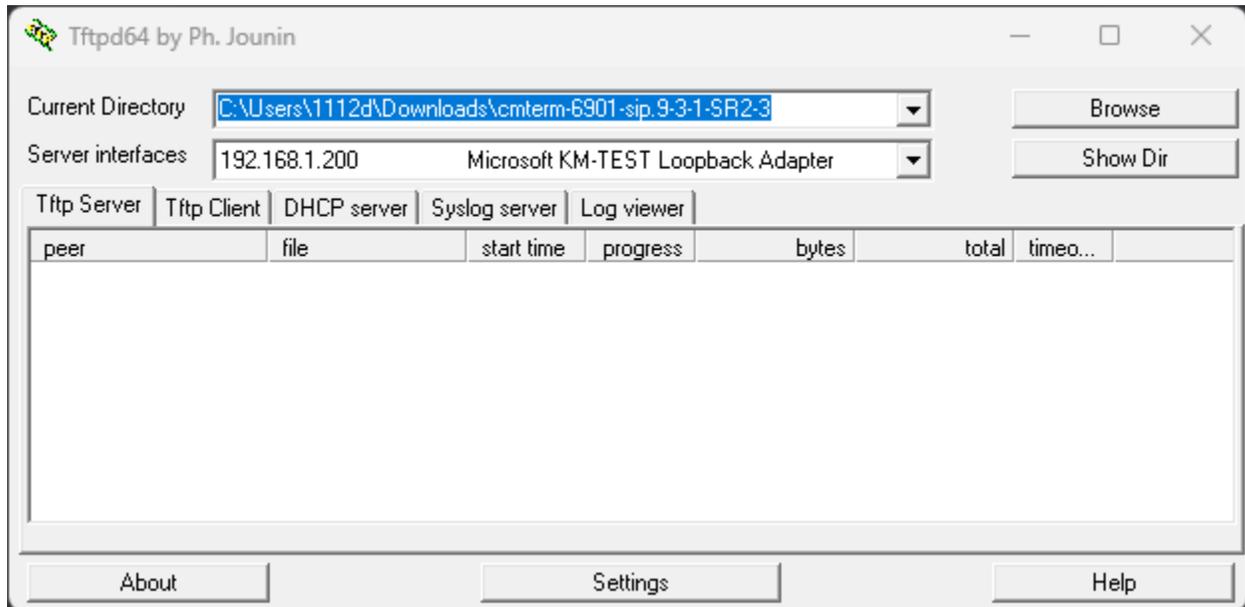


Click 'OK' once the Global settings and TFTP settings are configured. Tftpd64 will prompt you to restart the service, but this is not necessary to transfer the firmware to the router.

Loading the Firmware & CME Files

Once the TFTP server has been installed, now it will be possible to load the firmware files for the 6941 VoIP phone and the CME 8.0 files onto the router. Since the CME 8.0 files only include files for SCCP for the 6941 VoIP phone, I needed to also download the SIP firmware files for the 6941 phone. The link to this will be provided in the reference section. Once the files are downloaded, then I was able to point to the directory in which they were located in the TFTP

server. Then I had to set the server interface to the computer interface that I set the static IP address for. Once configured, the TFTP server should look like this:



Now that the TFTP server is ready to transfer the files over, I had to enter in the appropriate commands on the router. Under privileged EXEC mode on the router, enter the following commands:

- Copy tftp: flash - This command copies the contents of the TFTP server to the flash of the router.
- After this, it will then prompt for the remote host's IP address. This will be the address of the computer. Enter: 192.168.1.200.
- Finally, it will prompt for the name of the specific file in the directory that you would like to copy over. Enter: cmterm-6901-sip.9-3-1-SR2-3. I had to do this process twice. Once for the SIP firmware, and other for the CME files. For the CME files, enter:
cme-full-8.0.0.1.tar.

The TFTP server should display a message indicating that the files are being copied over, and it will disappear once the files are finished being copied over. After the files are copied over, use the 'show flash' command to display the contents of the router's flash. You will notice that they

are .tar files, meaning that they will need to be extracted. To extract the files, use the following command in privileged EXEC mode: 'archive tar /xtract flash:cmterm-6901-sip.9-3-1-SR2-3.tar' and 'archive tar /xtract flash: cme-full-8.0.0.1.tar'. Once these files have been extracted, reload the router and continue.

Implementing DHCP Server

When configuring the DHCP server for a VoIP network, you will typically need two DHCP pools. One will be the voice pool and the other will be the data pool. The voice pool will be used to give the VoIP phones IP addresses, while the data pool will be used to give other devices on the network IP addresses. You will need to exclude IP addresses to account for the router ports receiving a static IP address.

```
ip dhcp excluded-address 192.168.20.1 192.168.20.10
ip dhcp excluded-address 192.168.25.1 192.168.25.10
```

Once you have excluded the IP addresses, you will then need to actually configure the DHCP pools. You will need to set a network address for each pool and an IP address for the default router. This IP address will be the same IP address that you will configure for the inter-VLAN routing ports later on. You will also need to configure the 'option 150 ip' command with the IP address that the VoIP phones will contact as a DHCP server. This IP address will be the same address as the default router.

```
ip dhcp pool Data
 network 192.168.20.0 255.255.255.0
 default-router 192.168.20.1 255.255.255.0
 option 150 ip 192.168.20.1
!
ip dhcp pool Voice
 network 192.168.25.0 255.255.255.0
 default-router 192.168.25.1
 option 150 ip 192.168.25.1
```

Inter-VLAN Routing

Now we must configure inter-VLAN routing and single point routing. The point of this is to enable the router to route traffic between VLANs as well as routing voice and data networks using a single port between the router and the switch. The first step of this is to enable and configure the VLANs on the switch. You must declare the VLAN number, name, and optionally a description. On the switch, I created three VLANs - data VLAN, voice VLAN, and management VLAN. Respectively assigned to VLANs 20, 25, and 21. Once these VLANs have been declared, then you will want to assign these VLANs to each switchport that you will be using for your VoIP phones. In this case, I am only using FastEthernet0/2-03 for my VoIP phones. Set these ports to access ports, set the access VLAN to 20, and the voice VLAN to 25.

```
interface FastEthernet0/2
  switchport access vlan 20
  switchport mode access
  switchport voice vlan 25
  spanning-tree portfast
!
interface FastEthernet0/3
  switchport access vlan 20
  switchport mode access
  switchport voice vlan 25
  spanning-tree portfast
```

Once this has been completed, now the router will be configured to allow for Inter-VLAN routing. On the router, I created a subinterface for each VLAN that will be on the switch. I set each subinterface with an IP address that corresponds to the network address for each VLAN and set the 'encapsulation dot1q' command to correspond with the VLAN ID as well.

```
!
interface FastEthernet0/0.20
  description Data VLAN
  encapsulation dot1Q 20
  ip address 192.168.20.1 255.255.255.0
!
interface FastEthernet0/0.21
  description Management VLAN
  encapsulation dot1Q 21
  ip address 192.168.21.1 255.255.255.0
!
interface FastEthernet0/0.25
  description Voice VLAN
  encapsulation dot1Q 25
  ip address 192.168.25.1 255.255.255.0
!
```

Now there is a subinterface to correspond with every VLAN that I just created. That is the final step of inter-VLAN routing configuration and I will now move onto trunking and STP configurations.

Trunking & STP

Since the switch and the router will be directly connected to each other via FastEthernet0/1, the switch needs to be configured as a trunk on that port. To do this, use the 'switchport mode trunk' command on FastEthernet0/1 on the switch. This will then allow data to flow across the port from the router to the VLANs.

After trunking has been configured, now we will configure STP on the other switchports. For this network topology, I'll be setting the other switchports to the portfast state. This will cause the port to enter the forwarding state immediately and is meant for ports that are connecting to end devices, i.e. the VoIP phones. Since my VoIP phones will be connected via the FastEthernet ports 0/2-3, I will use the command 'spanning-tree portfast' on both of these ports.

Telephony Configuration & SIP Configuration

```
telephony-service
auto assign 1 to 10 type 6941
cnf-file perphone
load 6941 XMLDefault6941.cnf.xml
```

Before starting to configure the telephony configuration, first we must go into the telephony-service configuration mode. The first command I've input here is 'auto assign 1 to 10 type 6941'. This command will set all ephones to type 6941 automatically, since this is the only ephone type that I will be using. This command will also ensure that we do not have to set the ephone-type in the telephony configuration mode, since this command will take precedence. The 'cnf-file perphone' command loads the individual CNF file for each phone. The last command shown in this screenshot is the 'load 6941 XMLDefault6941.cnf.xml' command. This command will load the correct firmware onto the 6941 phones that we originally loaded into the router from the TFTP server. When entering these commands, you should see a message displayed that says "Updating CNF files. CNF files updating complete" - this is how you will know that the operation was successful.

Next, we will configure the router for SIP connections. To enable the router for SIP connections, enter the 'voice service voip' configuration mode. Then enter the command 'allow-connections sip to sip' to allow SIP-enabled ephones to contact other SIP-enabled ephones. If you wanted a SIP ephone to contact a H323 ephone, you would instead use the command 'allow-connections sip to h323', but for this project we will not be needing to utilize this command. Then enter the 'sip' configuration command to enter the SIP configuration. Enter 'registrar server expires max 1200 min 300' to enable Cisco IOS SIP registrar and set the maximum time to 1200 and the minimum time to 300 seconds.

```
voice service voip
allow-connections sip to sip
sip
registrar server expires max 1200 min 300
!
```

Now we will move onto configuring Voice Register Global parameters. This configuration mode will set global parameters for all SIP-enabled Cisco phones in the SIP environment. The configurations are similar to the SCCP configuration commands. Use the 'voice register global' command to enter this configuration mode. Use the 'mode cme' command to set the Cisco IOS SIP registrar to CME mode. Then, use the 'source-address' command to set the source address for phone registration and the 'port' parameter to set the port number. This source address should be the same address as the default router for the Voice DHCP pool. Set the maximum number of extensions that can be enabled with the 'max-dn' command, and set the maximum number of phones that can be enabled with the 'max-pool' command. Use the 'authenticate register' command to set authentication for phone registration. Lastly, use the 'create profile' command to create a configuration file for all phones.

```
voice register global
mode cme
source-address 192.168.25.1 port 5060
max-dn 20
max-pool 10
authenticate register
file text
create profile sync 0001081529528165
!
```

Now we will configure the extensions for each phone and their parameters. Since I am only using two IP phones, we will only be configuring two extensions. Use the 'voice register dn [number]' command to create an extension and enter the configuration per extension. I will be creating 'voice register dn 1' and 'voice register dn 2' for each phone. In the 'voice register dn 1' configuration, we will first set the label as 'Receptionist'. This is what will appear on the phone screen and what other users will see when receiving a call from this phone. Then set the extension number with the 'number 1001' command. This is what internal users will use to call this phone. Lastly, you can also set a name for each phone with the 'name' command. Users will also see this appear when you make calls. In the 'voice register dn 2' configuration, I set the number to 2002, and set the label to Manager. Here is the full configuration:

```
voice register dn 1
  number 1001
  name Drina
  label Receptionist
!
voice register dn 2
  number 2002
  name Allen
  label Manager
.
```

Lastly, we will be configuring each individual SIP phone. To enter the individual SIP phone configuration mode, use the 'voice register pool [number]' command. Set the MAC address for each phone with the 'id mac [MAC-address]' command. Then assign the buttons to each phone to speed-dial each other with the 'number [number] dn [number]' command. Specify the codec with the 'codec g711ulaw' command. Ensure to use the same codec for each phone, or else your calls will not go through. Last, set the username and password for each phone for the SIP registrar. The full commands should be as follows:

```
voice register pool 1
  id mac C414.3C8B.17FC
  number 1 dn 1
  username drina password password
  codec g711ulaw
!
voice register pool 2
  id mac C414.3C8B.179F
  number 2 dn 2
  username Allen password password
  codec g711ulaw
.
```

Multicast Paging SIP Configuration

Multicast paging allows you to page groups of phones directly without a PBX or SIP server being involved with the page. To configure multicast paging, you have to configure some SCCP configurations before implementing it under the SIP configuration. Enter the command 'ephone-dn 1' to configure our first VoIP phone. The dn-tag was set to 1 already in our voice pool register configuration for our first VoIP phone and the dn-tag was already set to 2 for our second VoIP phone. Now, you will be in the ephone-dn configuration prompt. Enter the

command 'number 1001' for our first VoIP phone. While still in the ephone-dn configuration, enter the command 'paging ip 239.0.1.1'. IP phones do not support multicast address 224.x.x.x. This multicast address is the address that will be used to multicast voice packets for audio paging. Now, while still in the ephone-dn configuration, type 'paging group 10' to create a paging group for the VoIP phones. For the second IP phone, set the ephone-dn to 2, set the number to 2002, set the multicast IP address, and add the second IP phone to paging group 10 as well. Once both IP phones have been configured for the multicast group, enter the SIP voice pool configuration with the 'voice register pool 1' command for the first IP phone and 'voice register pool 2' for the second IP phone. On both IP phones, enter 'paging-dn 10' to add both SIP phones to the multicast paging group. The commands should be as follows:

```
HobsonRouter(config)# ephone-dn 1
HobsonRouter(config-ephone-dn)# number 1001
HobsonRouter(config-ephone-dn)# paging ip 239.0.1.1
HobsonRouter(config-ephone-dn)# paging group 10
HobsonRouter(config-ephone-dn)# exit
HobsonRouter(config)# ephone-dn 2
HobsonRouter(config-ephone-dn)# number 2002
HobsonRouter(config-ephone-dn)# paging ip 239.0.1.1
HobsonRouter(config-ephone-dn)# paging group 10
HobsonRouter(config-ephone-dn)# exit
HobsonRouter(config)# voice register pool 1
HobsonRouter(config-register-pool)# paging-dn 10
HobsonRouter(config)# voice register pool 2
HobsonRouter(config-register-pool)# paging-dn 10
```

Voicemail Configuration

The voicemail mailbox feature for IP phones allows you to press the “Message” button on an IP phone and then listen to voicemails that have been left for the IP phone user. Before configuring voicemail, the directory number and number for the voicemail phone number must be configured. First, enter the voice register global configuration mode to set parameters for all SIP phones. Then I will define a telephone number that is speed-dialed when the Messages button on a Cisco IP phone - for my configuration I used the number ‘1111’. Once the global voicemail number has been configured, you must go into each individual phone’s configuration and set a back-to-back user agent, a voicemail box to use at the end of a chain of call forwards, and a call forwarding number for a SIP back-to-back user agent.

Enter the voice register dn configuration and set the SIP back-to-back user agent with the command ‘call-forward b2bua busy 1000’. The 1000 extension is where calls will be forwarded if the number that was called is busy. Next, configure the voice mailbox to use at the end of a chain of call forwards with the ‘call-forward b2bua mailbox 2200’ command. This extension is where calls will be forwarded if incoming calls have been forwarded to a busy extension. Lastly, enable call forwarding for a SIP back-to-back user agent so that incoming calls to an extension that does not answer will be forwarded to a specific extension with the ‘call-forward b2bua noan 2201 timeout 15’. The ‘15’ in this command specifies the amount of seconds before the call gets forwarded to the designated extension. I have set these extensions to the same extension number on both IP phones. This means that all calls will be sent to the same extension, however you can have them sent to different extensions. The full configuration is as follows:

```
HobsonRouter(config)# voice register global
```

```
HobsonRouter(config-register-voice)# voicemail 1111
```

```
HobsonRouter(config-register-voice)# exit
```

```
HobsonRouter(config)# voice register dn 1 **This is the first IP phone’s config
```

```
HobsonRouter(config-register-dn)# call-forward b2bua busy 1000
```

```
HobsonRouter(config-register-dn)# call-forward b2bua mailbox 2000
HobsonRouter(config-register-dn)# call-forward b2bua noan 2200 timeout 15
HobsonRouter(config)# voice register dn 2 **This is for the second IP phone's config
HobsonRouter(config-register-dn)# call-forward b2bua busy 1000
HobsonRouter(config-register-dn)# call-forward b2bua mailbox 2000
HobsonRouter(config-register-dn)# call-forward b2bua noan 2200 timeout 15
```

Music-on-Hold Configuration

Music-on-hold was the one configuration that I was not able to properly implement due to not having the time to do so. Instead, I will go over the information that I researched here and go through a step-by-step of how I would have configured MOH. Before configuring MOH, you must have access to the CUCM server to configure it. Music-on-hold is not able to be configured through the CLI on Cisco IP phones. To access CUCM, you must connect your router to your computer and ensure that the router interface is on 'no shutdown'. On your computer, open a browser window (preferably Chrome or Firefox), and type in the router's IP address on the address bar. By default, this should be <https://10.1.1.10/>. Then click on "Cisco Unified Communications Manager" under the "Installed Applications" section. You will then be prompted to enter your login credentials.

Now that we have accessed CUCM, we will now want to configure music-on-hold audio sources. First we must add a music-on-hold audio source. To do this, while in CUCM, choose Service > Music On Hold. Choose 'Add new MOH Audio Source' in the MOH Audio Sources list. You will now want to enter in the appropriate settings according to this chart:

Field	Description
MOH Audio Source Information	
MOH Audio Stream Number	Use this required field to choose the stream number for this MOH audio source. To do so, click the drop-down arrow and choose a value from the list displayed. For existing MOH audio sources, this value displays in the MOH Audio Source title.
MOH Audio Source File	Use this required field to choose the file for this MOH audio source. To do so, click the drop-down arrow and choose a value from the list displayed.
MOH Audio Source Name	Enter a unique name in this required field for the MOH audio source. This name can comprise up to 50 characters. Valid characters include letters, numbers, spaces, dashes, dots (periods), and underscores.
Play continuously (repeat)	To specify continuous play of this MOH audio source, check this check box. Note Cisco recommends checking this check box. If continuous play of an audio source is not specified, only the first party placed on hold, not additional parties, will receive the MOH audio source.
Allow Multicasting	To specify that this MOH audio source allows multicasting, check this check box.

Ensure to allow multicasting for this project environment, since we have configured multicast paging previously. Now that we have added the audio to be played when music-on-hold is in action, we will now configure a music-on-hold server by adding a new one.

To add a new music-on-hold server, navigate to Service > Music On Hold. Click 'Configure Music On Hold Servers', and choose 'Add new Music On Hold Server' from the MOH Servers list. You will be prompted for a host server, server name, description, device pool, location, maximum half duplex streams, maximum multicast connections, fixed audio source device, and whether or not you would like to choose a run flag for the MOH server. Essentially, if you select 'no' on the run flag option, you will disable the MOH server. So you will want to ensure that the run flag is set to 'yes'. After you have input the correct configurations, click 'Insert'. The status will change from 'Ready' to 'Insert completed'. The MOH Servers list will now include the new music-on-hold server. Ensure to select your new server from the drop down box.

SIP Server Installation & Configuration

The SIP server that I will be using for my project is the 3CX free SIP server. I am using a SIP server opposed to a PBX system - meaning that the calls my phones can make will only be internal. If I were to use a PBX system instead, then the phones would be able to make calls outside of my organization. I will be linking the 3CX SIP server website in my research references. This requires no installation, you only need to sign up for your free account. Once you've signed up for your free account, you will be given an extension number and login information via email, which you will then use to access the 3CX configuration page. Once you've accessed the configuration page for 3CX, hover over your name in the top right corner and click 'Add System' to add a new SIP system.

Drina Hobson
dmh240@uakron.edu

Add System
My Systems
My Account
Support
Certification
Downloads
Logout

When you click 'Add System', you will be prompted to select a type of system that you would like to configure. For this project, I will be selecting the 3CX start up system.

Which 3CX?

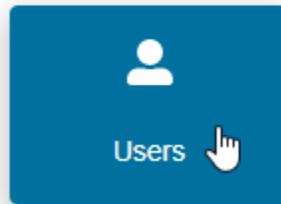
Choose the correct 3CX install based on your company size and requirements. See detailed feature and pricing comparison [here](#).

3CX StartUP - Up to 20 users - Hosted - FREE

A shared instance hosted by 3CX perfect for smaller companies. FREE tier gives core telephony features & WhatsApp integration.

We will now begin to set up our SIP server and register our phones for our 3CX SIP server. At the bottom left of the page, you will see a button that says 'Admin'. This is where we will navigate to to fully configure our SIP server. Click on 'Users' so that we can add our IP phones.

Configure



Now click on 'Add user' and enter in their general info (full name and email). This is required for every user that we will be registering. The email does not have to be a real one. Now, navigate to 'IP Phone' and click on 'Configure a phone'. A window will appear asking you what type of phone you would like to configure. Select 'Router Phone', since our router is fully configured to support the SIP server now.

What type of phone? ×



In a hosted / public cloud environment all voice traffic must be proxied via a router phone or an SBC. If you do not have an SBC or router phone already, your first phone must be a router phone. Select the type of IP Phone to add:

- Router Phone — This phone will proxy all voice traffic for the location.
- Normal Phone — This phone will work via another router phone.

It will then prompt you to enter in your phone model, in this case will be the Cisco 6941 for both phones, and the MAC address. Enter the same MAC address that you entered for our SIP configurations that matches the back of the phone. Ensure that the registrar SIP port is the same as we configured earlier, port 5060. Set the Outbound Proxy address as the address of the router IP address for the Voice VLAN, being 192.168.25.1. Once you have this information on the user page, double-check that your extension number matches for the user account as well as each phone on the router. Also ensure that the router is pointing towards the 3CX SIP server's IP address, in this case being 192.168.25.2/24. After you have completed this configuration, you will want to reboot both of your phones, and then you should be able to make calls to one another when both user profiles are fully configured.

Research References

Software Download Links

PuTTY : <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

Tftpd64 : <https://tftpd64.software.informer.com/download/>

SIP 6941 Firmware : <https://software.cisco.com/download/home/282601278>

CME 8.0 : http://auto-download.com/download/get/_rWrc9vf/cme-full-8_0_0_1_tar.html

3CX SIP Server : <https://www.3cx.com/phone-system/sip-server/>

References

“Cisco Unified CME 8.0 Supported Firmware, Platforms, Memory, and Voice Products.” Cisco,

Cisco, 14 Nov. 2018,

https://www.cisco.com/c/en/us/td/docs/voice_ip_comm/cucme/requirements/guide/cme80s/pc.html.

“Cisco Unified Communications Manager Express Command Reference - Cisco Unified CME

Commands: E [Cisco Unified Communications Manager Express].” Cisco, 12 Mar. 2018,

https://www.cisco.com/c/en/us/td/docs/voice_ip_comm/cucme/command/reference/cme_cr/cme_e1ht.html#wp2012233745.

“Cisco Unified Communications Manager Express System Administrator Guide - Configuration

Files for Phones [Cisco Unified Communications Manager Express].” Cisco, 30 June

2022,

https://www.cisco.com/c/en/us/td/docs/voice_ip_comm/cucme/admin/configuration/manual/cmeadm/cmefiles.html.

Cisco. (2022, June 30). *Cisco Unified Communications manager express system administrator guide - paging [Cisco Unified Communications manager express]*. Cisco. Retrieved March 24, 2023, from https://www.cisco.com/c/en/us/td/docs/voice_ip_comm/cucme/admin/configuration/manual/cmecm/cmepage.html#config_page

“Cisco Unified Communications Manager Express: SIP Implementation Guide.” Cisco, Cisco, 13 July 2021, <https://www.cisco.com/c/en/us/support/docs/voice-unified-communications/unified-communications-manager-express/99946-cme-sip-guide.html>.

“Sip Configuration Guide, Cisco Ios Release 12.4T - Basic Sip Configuration [Support].” Cisco, Cisco, 7 Oct. 2013, https://www.cisco.com/c/en/us/td/docs/ios/voice/sip/configuration/guide/12_4t/sip_12_4t_book/sip_cg-basic_cfg.html.

“Troubleshooting Phone Registration in Cisco Unified CME.” Cisco, Cisco, 22 Nov. 2013, https://www.cisco.com/c/en/us/td/docs/voice_ip_comm/cucme/troubleshooting/guide/ts_ph_reg.html.

Convert Cisco 6921/6941/6961 phone from SCCP to SIP protocol - 58VOIP-3CX. 58VOIP. (2019, August 20). Retrieved March 12, 2023, from <https://en.58voip.com/convert-cisco-6921-6941-6961-phone-sccp-sip-protocol/>