VoIP Lab Part 1

# CCNP Lab 7

## Axel Li

## 1 April 2019

# Purpose

The purpose of the lab was to configure a Voice over IP network. Our task was to configure two phones with Cisco Unified Communications Manager in a network with a router and switch, and to be able to place a call between the two phones.

# Background

Voice over IP is a technology which transmits voice data using IP packets rather than a traditional phone line. This allows phones to be used on a LAN without needing to purchase phone lines for a phone network, providing an alternative to PTSN which requires a dedicated network of phone lines. This allows for lower costs from not needing traditional copper wire phone systems in addition to simpler management. VoIP also can combine different forms of communication, such as telephones, voice and video conferencing, and email, into a single system that can be used by computers, phones, and mobile phones.

VoIP implements a best-effort system without QoS, and typically handles data first-come first-serve with packets sent sequentially. Thus, VoIP may suffer from data loss, latency, and jitter, especially in networks with high traffic. Despite these drawbacks, VoIP has become more popular in use in corporate and consumer settings due to the number of benefits it provides in cost, use, and management.

# Summary

For this lab my partner and I set up voice and data VLANs on a Power over Ethernet Switch, and configured a router as a DHCP and NTP server in a router-on-a-stick configuration to allow the voice and data VLANs to communicate. We also set up Cisco Unified Communications Manager on a virtual machine using VMware 14. We then configured it to act as a TFTP server for the phones to access configuration files. Through the web interface, we activated all services on CUCM and registered two phones in a device pool. We then verified that everything was working properly by calling between the two phones.

# Commands

The key commands used in this lab for the routers were:

option 150 ip [ip address] – points phones to the CUCM call manager

max-ephones [# of phones] – sets the maximum number of phone numbers

max-dn [# of directory numbers] – sets the maximum number of directory numbers

ip source-address [ip address] port [port #] – points to the CUCM call manager

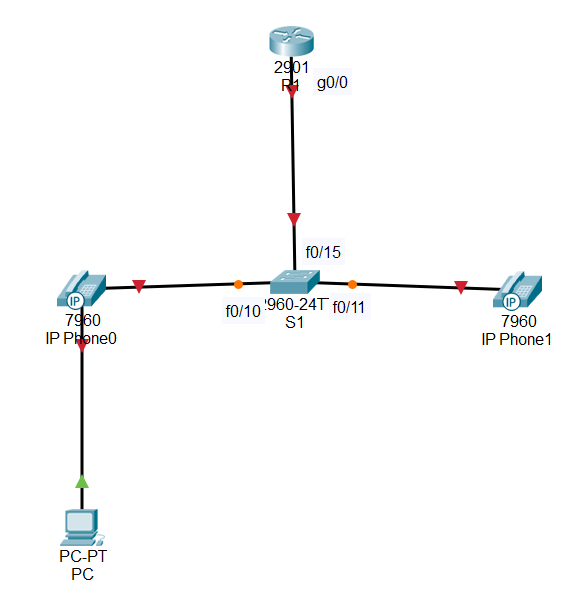
create cnf-files – configures CUCM to act as a TFTP server

# Tables and Diagrams

Addressing:

|  |  |  |
| --- | --- | --- |
| Device | IP Address | Subnet Mask |
| R1 | 192.168.10.1 | 255.255.255.0 |
| R1 | 192.168.11.1 | 255.255.255.0 |
| CUCM | 192.168.10.24 | 255.255.255.0 |

Topology:



# Configurations

Router Configuration:

Enable

Conf t

Hostname R1

No ip domain-look

Ntp master

Int g0/0

No shut

Exit

Int g0/0.10

Encap dot 10

Ip add 192.168.10.1 255.255.255.0

No shut

exit

Int g0/0.11

Encap dot 11

Ip add 192.168.11.1 255.255.255.0

No shut

exit

Ip dhcp pool Data\_Pool

Network 192.168.10.1 255.255.255.0

Def 192.168.10.1

Option 150 ip 192.168.10.24

exit

Ip dhcp pool Voice\_Pool

Network 192.168.11.1 255.255.255.0

Def 192.168.11.1

Option 150 ip 192.168.10.24

exit

telephony-service

Max-ephones 2

Max-dn 2

ip source-address 192.168.10.24 port 2001

Create cnf-files

Exit

Switch Configuration:

en

config t

hostname S1

No ip domain-look

vlan 10

name Data\_Vlan

exit

vlan 11

name Voice\_Vlan

Exit

int range fa0/10-11

switchport mode access

switchport access vlan 10

switchport voice vlan 11

exit

int fa0/15

switchport trunk encapsulation dot1q

switchport mode trunk

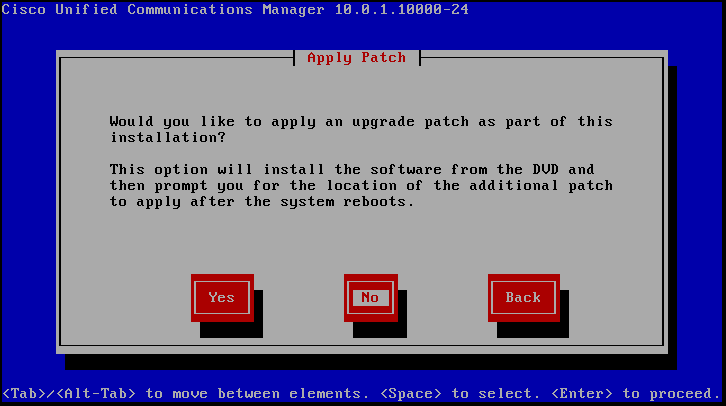
exit

CUCM VM Specifications:

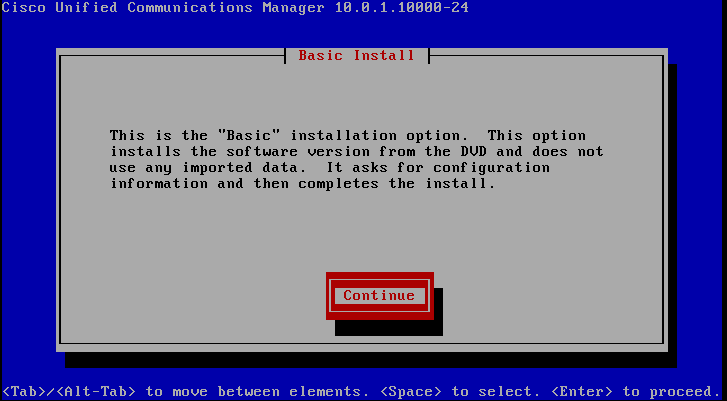
|  |  |
| --- | --- |
| Processers | 2 cores |
| RAM | 4 GB |
| Hard Drive | 80 GB |
| Connection Type | Bridged |

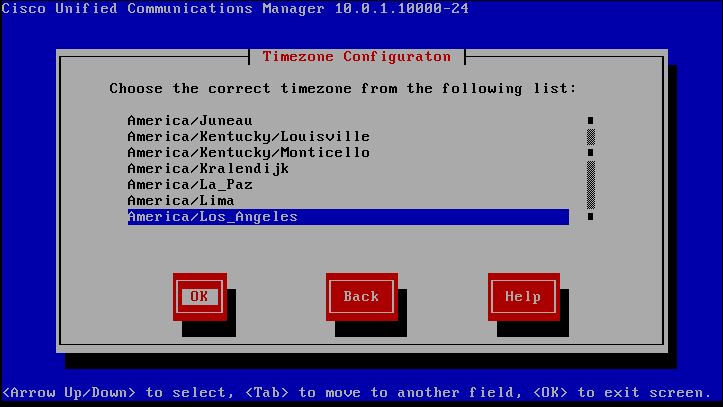
CUCM VM Setup:

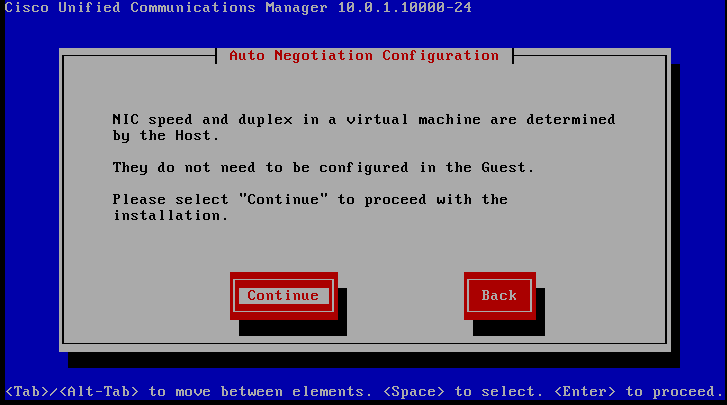


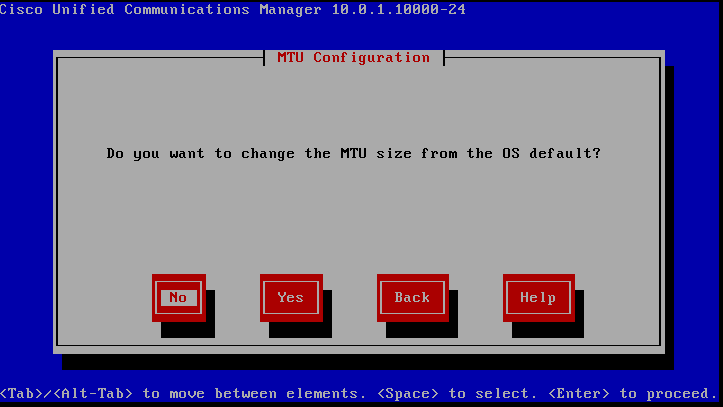


There is no need for the upgrade patch, so we will opt out.

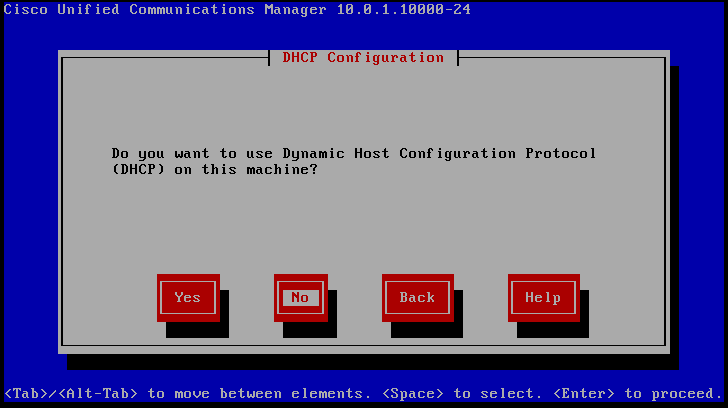




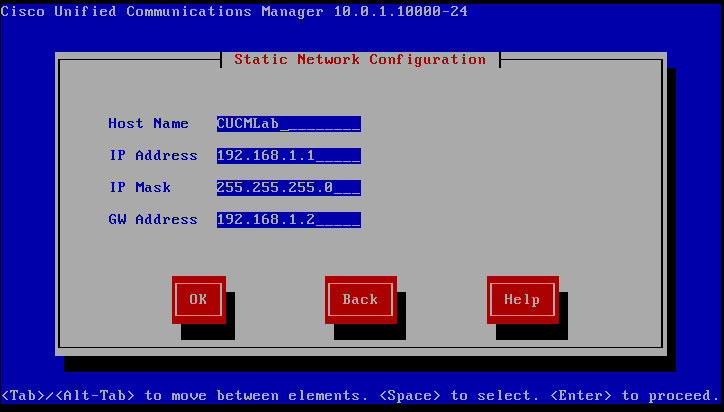




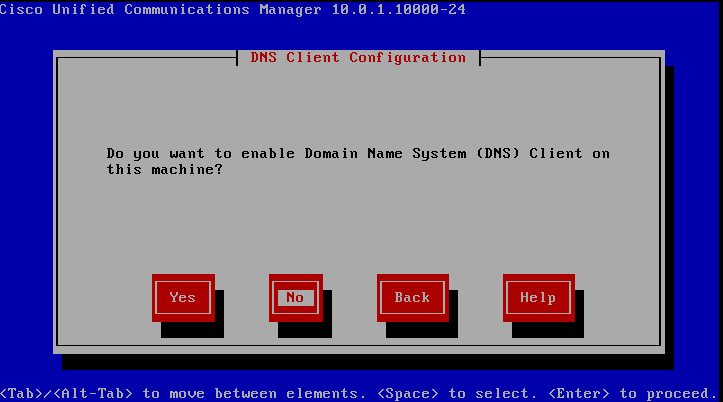
We will not have to do anything with the MTU size.



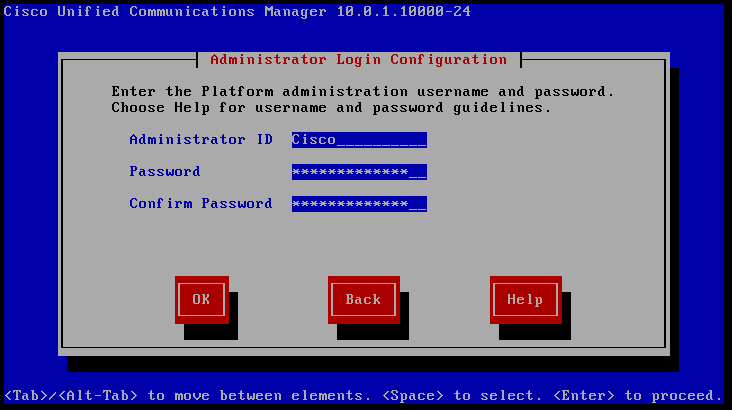
We will configure DHCP on the router rather than on the CUCM server.



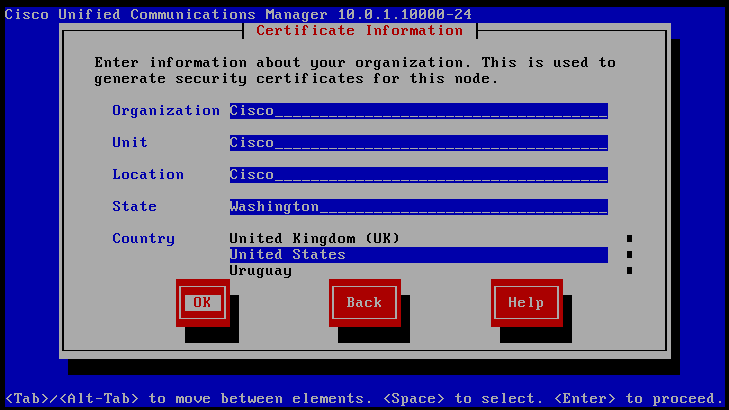
Here we need to set up basic IP addressing for the call manager and give it a name.



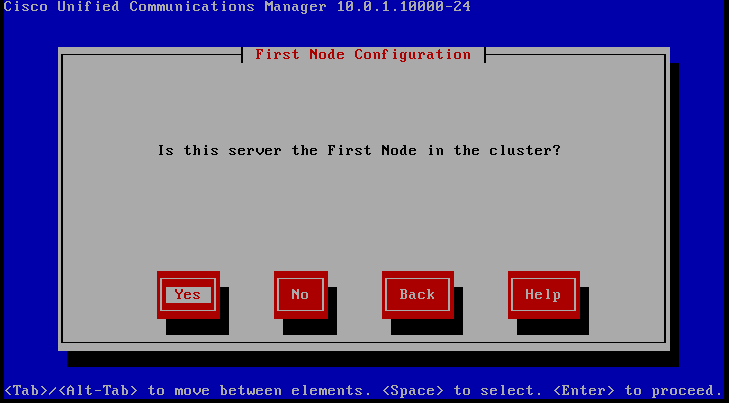
DNS is not necessary for this lab.



Here we will configure the administrator login credentials. The password needs to meet specific complexity and length requirements, which made selecting a memorable password difficult.



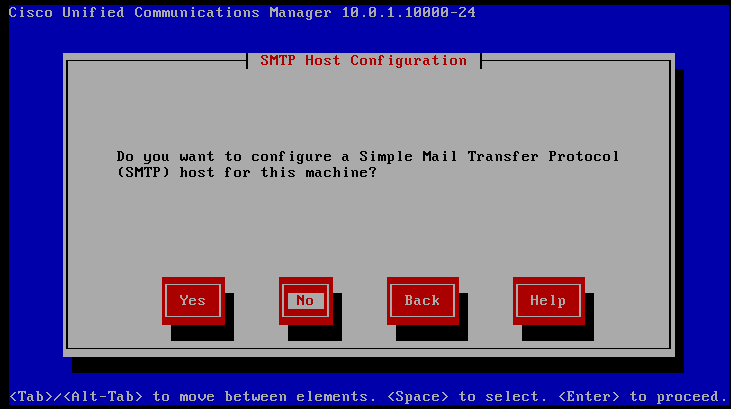
The information entered here is not of any importance.



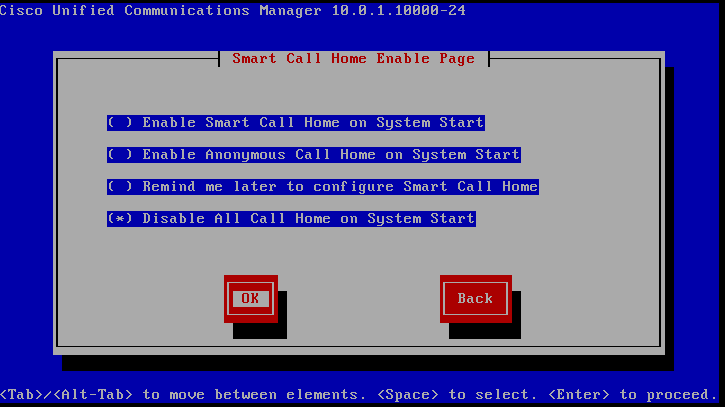
We will only be using one server, so we will put “Yes.”



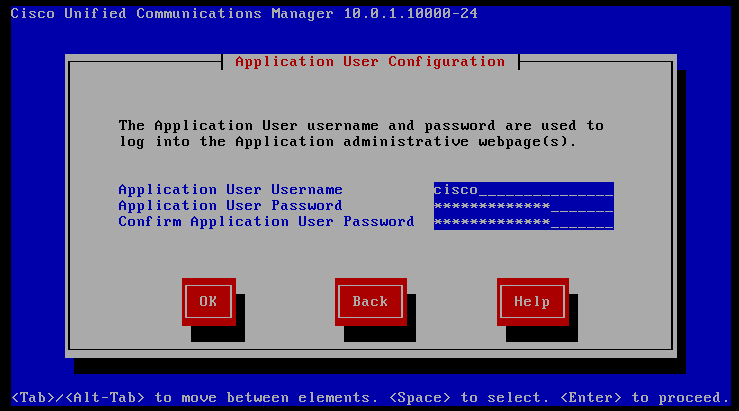
This password can be the same as the administrative password for convenience.



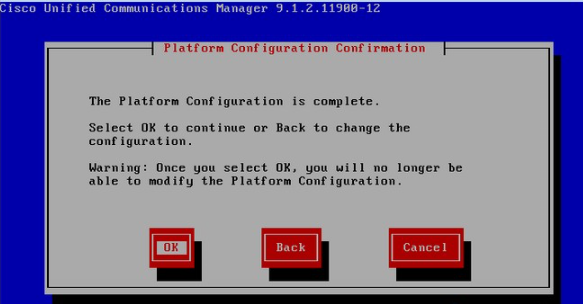
We will not need SMTP.



We will not need Smart Call Home, so we will select the disable option.

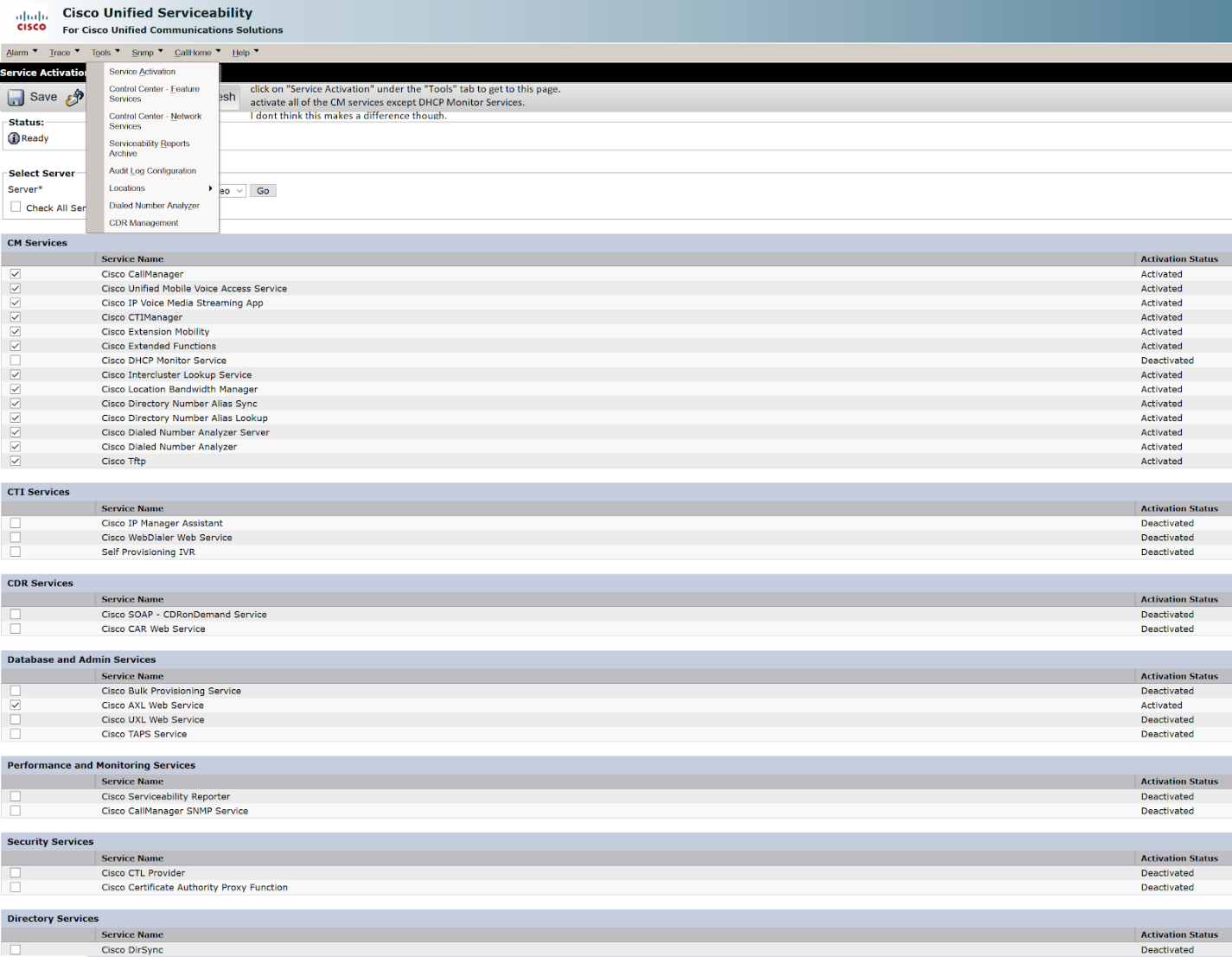


These credentials are needed for the web interface. For convenience we can use similar or the same credentials as the administrative login.

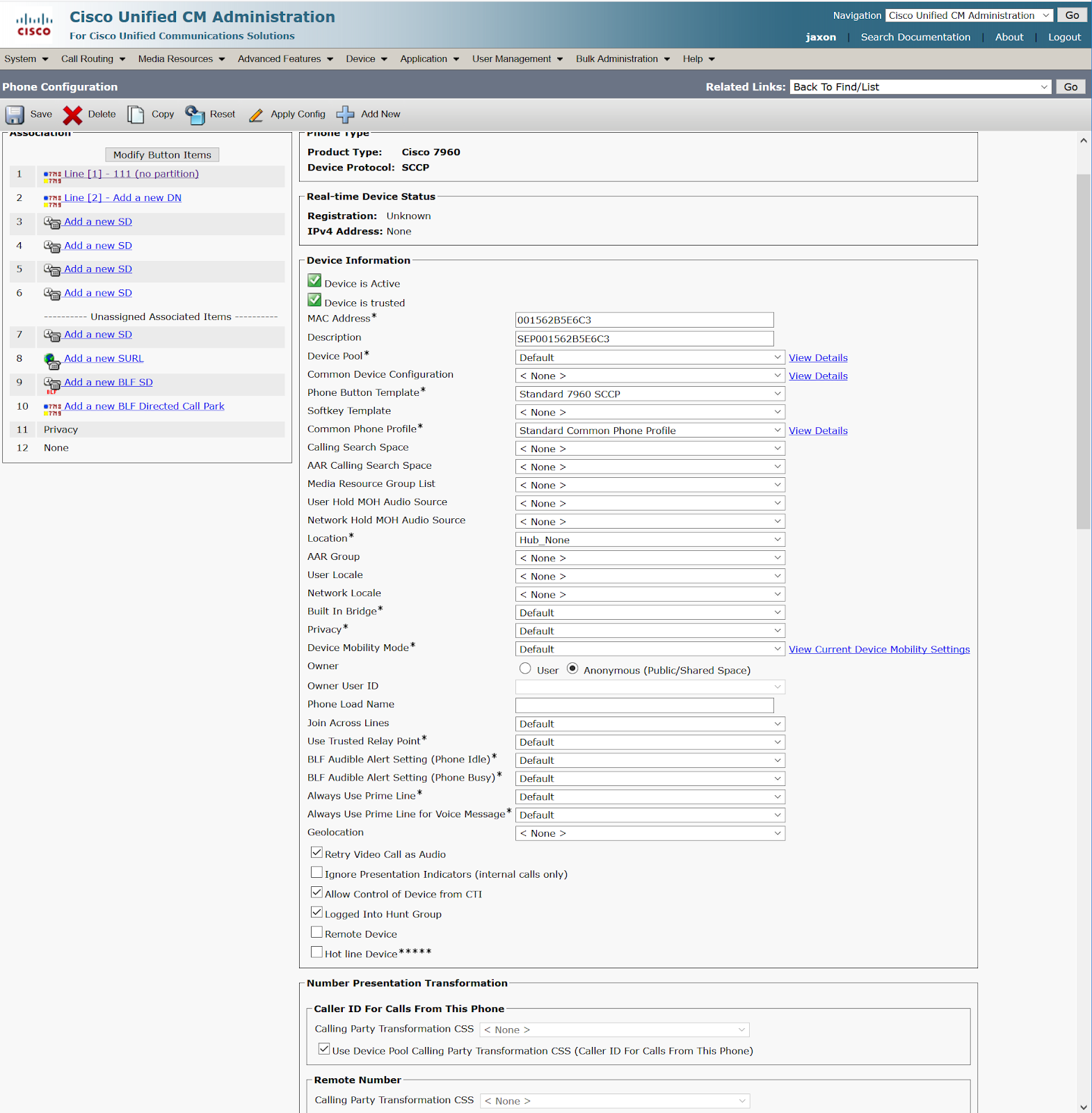


CUCM Web Interface:

We can reach the web interface by typing the IP address of the server in a browser. If there is a security error, click continue anyway. On the web interface we will be asked to log in on the top right of the page, and there are also a number of options in a drop-down list.

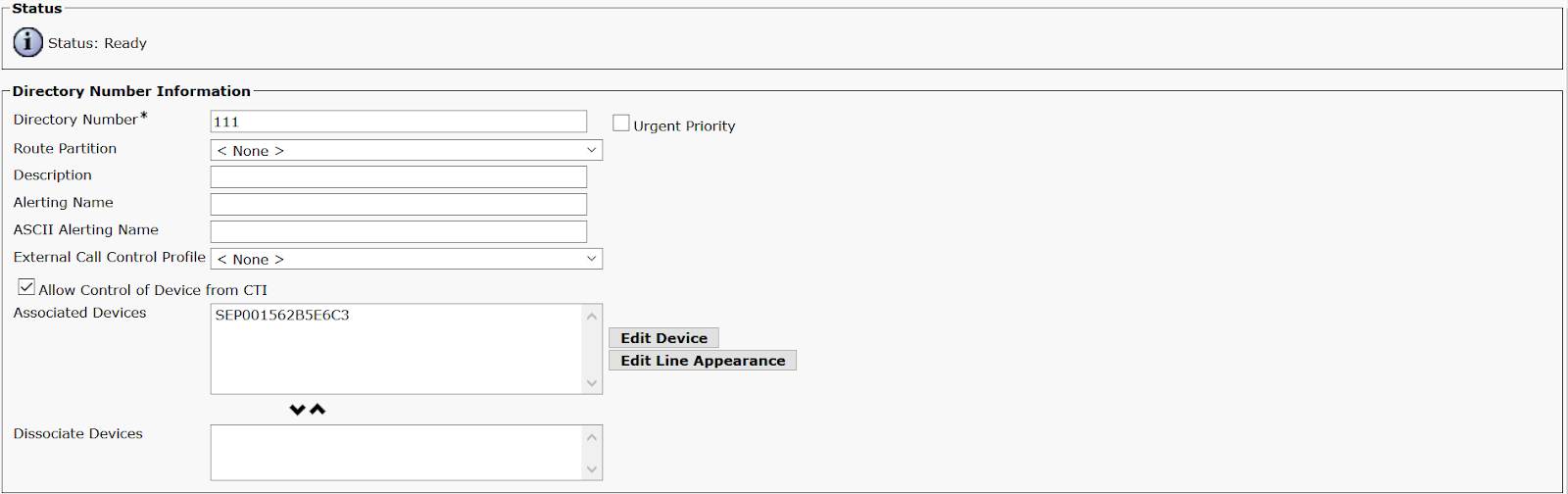


Under “Cisco Unified Serviceability,” in “Tools” select “Service Activation.” We will need to activate Cisco CallManager, TFTP, and CTIManager and save the changes, which can be done all at once by selecting “Check All Services” and implementing the changes.



Prior to setting up phones we will need to set up a Communications Manager Group (Cisco Unified CM Administrator > System > Cisco Unified CM Group) and device pool (Cisco Unified CM Administrator > System > Device Pool) with basic configurations. We can then configure phones under “Device” in the “Cisco Unified CM Administration” tab. We will need to specify the phone type, mac address, and device pool. For “Owner” we will select “Anonymous.” We will also change the phone button template and device security profile to the defaults provided.

In the top left corner we can configure directory numbers.



We can assign phones any directory number desired here. After we are done, we will need to select “Apply Config” for the changes to take place.

# Problems

Throughout this lab we encountered a variety of minor problems due to being unfamiliar with the CUCM web interface and the new commands necessary. One such example was when the phones were unable to connect despite all the configurations being correct. It turns out that we had to press the “Apply Configuration” button in order for the configurations to be established. While simple, this issue took an unusually long time to resolve due to us being confused with what we needed to do.

We also experienced some problems setting up CUCM in VMware in the first place. One such problem was creating a password for the administrative login, which has minimum complexity and length requirements. The requirements were surprisingly strict, and it took us many attempts to find a satisfactory, memorable password that would be accepted. Another such problem was simply being unable to connect to the web interface. To resolve this, we used the command “utils service restart Cisco Tomcat” in the command-line interface, which restarted the web service and seemed to solve the problem just fine.

# Conclusion

This lab was a basic, but challenging introduction to configuring VoIP on a network. It was particularly difficult at beginning navigating the CUCM web interface and figuring out what was necessary for a basic configuration, but the steps we took became quite simple in hindsight. While helpful as an introduction, this lab only covered the very basics and the next lab will involve additional advanced concepts within VoIP.