The purpose of this lab was to get us used to a potentially real-world situation; setting up VoIP and to give us an idea of how IP phones work. Plus, we should be taught about how this common protocol is set up, more than just a 20 second google search. Originally, I thought this lab would be a quick walk in the park, but it turned out to be a real challenge that I did get frustrated on.

Here is some background of the lab. **V**oice **o**ver **IP** is a protocol that is meant to carry voice traffic using IP. Traditionally, voice was carried over landlines with an analog port using POTS, but it has since been upgraded to a digital version. Most companies that used to run POTS have switched to VoIP due to several benefits.

First, POTS needed to use PBX, or Private Branch Exchange. This needed experts to set up and could be a real pain. VoIP has eliminated the need for PBX, which makes it much easier to configure. This is because VoIP routes calls via IP packets using a router and is sent to the destination.

Second, digital has been taking over most of the world. This means that the analog POTS system will be looked at as out of date and not very relevant anymore, VoIP, being a more digital counterpart, is more modern and better equipped for our digital world. These are why companies prefer to use VoIP over POTS;

In summary we set up the phones to obtain a number from the router and call each other internally. We then got the phone to call other numbers outside the network.

While DHCP is “new” for this lab, I have already done DHCP a ton in CCNA, so I will skip that. I will also skip trunking and VLANs on switches because I also did that a lot in CCNA. I will talk about how to set up phone numbers and dial out.

First, getting the phones their numbers. You need to set up a DHCP pool that the phones will use. One command that you need to use in the DHCP pool is

option 150 ip <tftp ip address>

This allows the VoIP phones to obtain an IP address. Usually this is from a TFTP server, but here the router is the device giving the phones their configuration.

There is also a command that will be used for most address later, which is:

ip host admin <X.X.X.X>

Next, go into a new config mode called:

telephony-service

From there, you input several commands:

max-ephone <#> // specifies how many phones will be added

max-dn <#> // specifies how many phone numbers will be needed

ip source-address <host admin> port 2000 // identifies the host address and port used for the service

max-conferences 8 gain –6 // for the switch conf

create cnf-files // creates the files that will be sent to the phones

There are a few more commands to give the phones their numbers, which are:

ephone-dn <#> // opens config of whatever phone number specified

number <4-digit number> // gives the phone a number

And

ephone <#> // opens config of whatever phone specified

mac-address <aaaa.bbbb.cccc> // specifies the MAC address of the phone

type <#> // specifies the type of phone ie. 7940, 7960, etc.

These commands will allow the phones to call each other internally. There is a lot more work required to get the phones to call out. First thing is that we needed a dial peer, which was done with:

dial-peer voice 1 pots

Dial peers are used to configure dial plans and identify source and destination endpoints. The **P**lain **O**ld **T**elephone **S**ervice is used to convert to analog from the digital VoIP line. From there add the following commands:

destination-pattern 91[2-9]......... // the 12 digits in the destination number. The 91 is needed to call out, and the remaining 10 identify the wanted phone number. To prevent misdialing 911, there is a [2-9] so that a second 1 isn't allowed.

port X/X/X // specifies the outgoing voice port

forward-digits all // forwards all digits in the pattern

Next you need to set up the proper voice port which is done by typing:

voice-port X/X/X

The X/X/X is the same as in the “port X/X/X” command from earlier. Next input the following:

connection plar <4-digit number> // also analog, matches the number of one of the phones

caller-id enable // allows the phones to have an ID. I'm not sure what the ID is because I can’t call myself due to strict calling rules imposed by the school.

Next, go into the interface (sub interface for my case) and under the voice sub interface, add this:

h323-gateway voip bind srcaddr X.X.X.X

Which sets the voice router as the designated gateway via h323. A couple other commands needed were:

voice service voip // configuration mode to make changes on VoIP in general

allow-connections h323 to h323 // allowing h323 connections on both endpoints

Once you have all of this, you are halfway done. Now you have to type several “ccm-manager” commands that, when I added them, I had no idea about what they did, but here is what I believe they all do:

ccm-manager mgcp // allows ccm to use mgcp

ccm-manager music-on-hold // enables a feature that isn't necessary for this lab

ccm-manager config server X.X.X.X // says where to get the files from, in this case the host admin

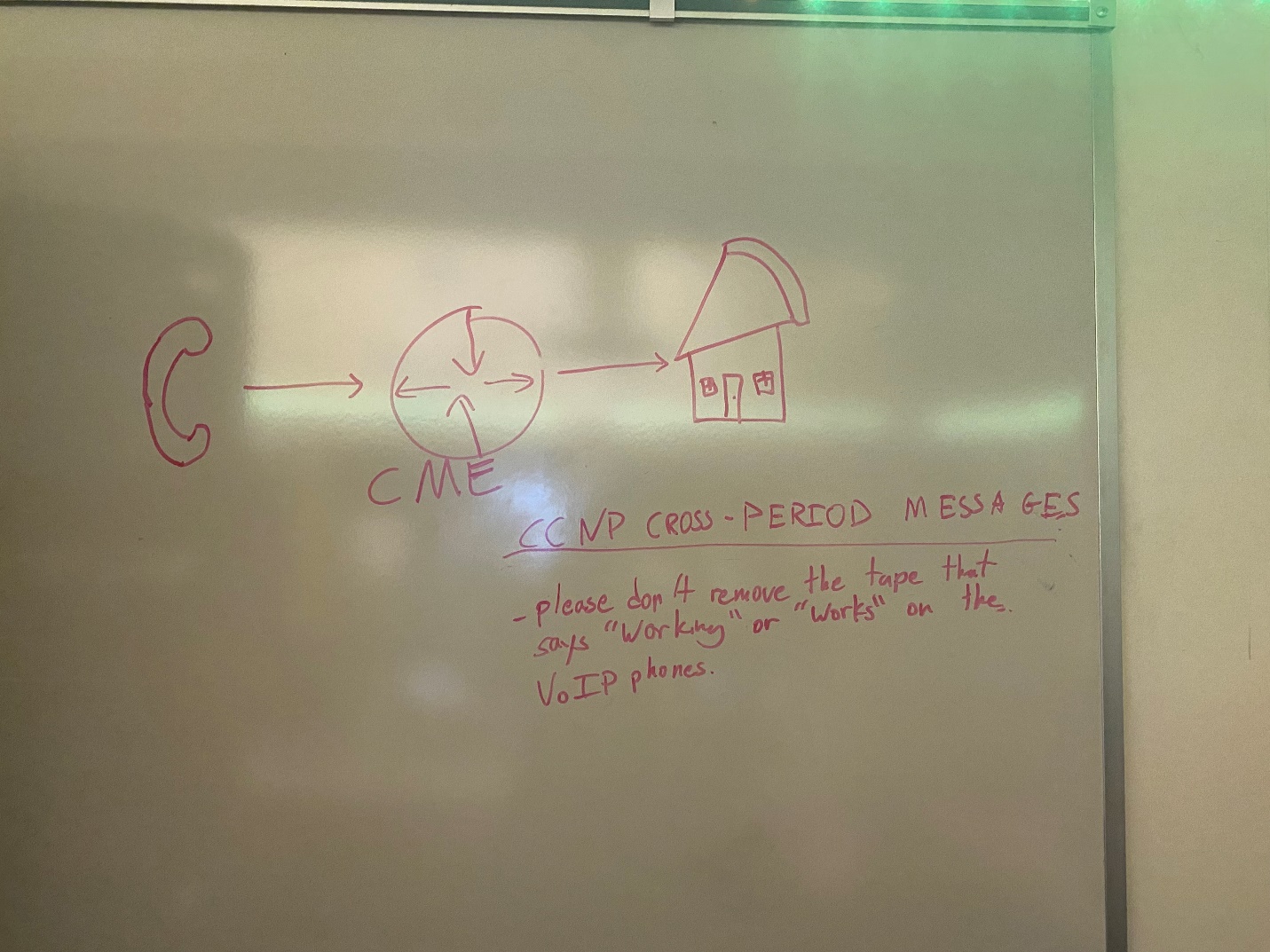
ccm-manager config // allows the needed

mgcp // enables mgcp

mgcp call-agent admin <port #> service-type mgcp version 0.1 // specifies the port number (either 2427 or 2428) and will use the aforementioned admin IP address

mgcp profile default // tells the profile to use whatever is default

Here is the network diagram that is conveniently laid out for me to use:



The footnote there is just because most of the phones that we could use didn’t work; they either didn’t start when we plugged them in, didn’t accept a number, or just broke.

Here is the running configuration for the router and switch:

VRouter#show run

Building configuration...

Current configuration : 3025 bytes

version 12.4

service timestamps debug datetime msec

service timestamps log datetime msec

no service password-encryption

hostname VRouter

boot-start-marker

boot-end-marker

logging message-counter syslog

no aaa new-model

memory-size iomem 10

no network-clock-participate slot 1

dot11 syslog

ip source-route

ip cef

ip dhcp excluded-address 192.168.10.1 192.168.10.5

ip dhcp excluded-address 192.168.20.1 192.168.20.5

ip dhcp pool Voice

network 192.168.20.0 255.255.255.0

default-router 192.168.20.1

option 150 ip 192.168.20.1

ip dhcp pool Data

network 192.168.10.0 255.255.255.0

default-router 192.168.10.1

ip host admin 192.168.20.150

no ipv6 cef

multilink bundle-name authenticated

voice service voip

allow-connections h323 to h323

voice-card 0

no dspfarm

voice-card 1

no dspfarm

vtp domain cisco

vtp mode transparent

archive

log config

hidekeys

vlan 2,10,20

vlan 996

name CUSTOMER\_NATIVE

interface FastEthernet0/0

no ip address

shutdown

duplex auto

speed auto

interface FastEthernet0/1

no ip address

duplex auto

speed auto

interface FastEthernet0/1.10

encapsulation dot1Q 10

ip address 192.168.10.1 255.255.255.0

interface FastEthernet0/1.20

encapsulation dot1Q 20

ip address 192.168.20.1 255.255.255.0

h323-gateway voip bind srcaddr 192.168.20.1

interface FastEthernet0/1.50

encapsulation dot1Q 50 native

interface FastEthernet0/0/0

interface FastEthernet0/0/1

interface FastEthernet0/0/2

interface FastEthernet0/0/3

interface Serial0/1/0

no ip address

shutdown

interface Serial0/2/0

no ip address

shutdown

clock rate 2000000

interface Serial0/2/1

no ip address

shutdown

clock rate 2000000

interface Serial0/3/0

no ip address

shutdown

clock rate 2000000

interface Serial0/3/1

no ip address

shutdown

clock rate 2000000

interface Vlan1

no ip address

ip forward-protocol nd

no ip http server

no ip http secure-server

control-plane

voice-port 1/0/0

connection plar 1020

caller-id enable

voice-port 1/0/1

voice-port 1/0/2

voice-port 1/0/3

voice-port 1/1/0

voice-port 1/1/1

ccm-manager mgcp

ccm-manager music-on-hold

ccm-manager config server 192.168.20.150

ccm-manager config

mgcp

mgcp call-agent admin 2427 service-type mgcp version 0.1

mgcp profile default

dial-peer voice 1 pots

destination-pattern 91[2-9].........

port 1/0/0

forward-digits all

telephony-service

max-ephones 2

max-dn 2

ip source-address 192.168.20.150 port 2000

max-conferences 8 gain -6

transfer-system full-consult

create cnf-files version-stamp Jan 01 2002 00:00:00

ephone-dn 1

number 1010

ephone-dn 2

number 1020

ephone 1

device-security-mode none

mac-address 0015.62B5.E72C

type 7960

button 1:1

ephone 2

device-security-mode none

mac-address 001D.A219.FA62

type 7940

button 1:2

line con 0

line aux 0

line vty 0 4

login

scheduler allocate 20000 1000

end

VSwitch#show run

Building configuration...

Current configuration : 4351 bytes

version 12.2

no service pad

service timestamps debug datetime msec

service timestamps log datetime msec

no service password-encryption

hostname VSwitch

boot-start-marker

boot-end-marker

no aaa new-model

system mtu routing 1500

vtp domain CCNP

vtp mode transparent

authentication mac-move permit

ip subnet-zero

crypto pki trustpoint TP-self-signed-1177695488

enrollment selfsigned

subject-name cn=IOS-Self-Signed-Certificate-1177695488

revocation-check none

rsakeypair TP-self-signed-1177695488

crypto pki certificate chain TP-self-signed-1177695488

certificate self-signed 01

30820240 308201A9 A0030201 02020101 300D0609 2A864886 F70D0101 04050030

31312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D 43657274

69666963 6174652D 31313737 36393534 3838301E 170D3933 30333031 30363136

35315A17 0D323030 31303130 30303030 305A3031 312F302D 06035504 03132649

4F532D53 656C662D 5369676E 65642D43 65727469 66696361 74652D31 31373736

39353438 3830819F 300D0609 2A864886 F70D0101 01050003 818D0030 81890281

81009904 1EE01ADF C4CFB67C 5BCEBC42 EAC69DBA D13E9E77 A205B3A0 181E508B

A5B705AE 970826A9 D9537A6F 63B692A7 CE884E2F C513A935 E2BCEEDF C218482F

6260377A F8FBFA47 90465E38 D1761D5A F49D8CAD 490218AF 4D9F1CC2 0C09E423

8A683579 E8F36C29 1EE4EC23 8B1CAD1F 63B319DA E72E5039 C8229F44 F4616971

DA030203 010001A3 68306630 0F060355 1D130101 FF040530 030101FF 30130603

551D1104 0C300A82 08565377 69746368 2E301F06 03551D23 04183016 80149287

C4E4DD29 D6247C64 B85F813A 43CE2AD1 9F0D301D 0603551D 0E041604 149287C4

E4DD29D6 247C64B8 5F813A43 CE2AD19F 0D300D06 092A8648 86F70D01 01040500

03818100 52759CEB 15E4A612 0A7498E7 624578B0 ADFA81B5 EB75732B 3A44C433

33937B4F 521EC30F 4A9C2E6F 66ED5C04 779A6729 BBB4BF7D 6FBC60C3 99496D7B

3BF2A7B1 BBB24EE5 E9AAF7A3 4F94E6E9 920F6B8C 37D806E4 63E28289 F02ABF36

3130A67A CF30EE48 61CE6305 149F5B87 F10BEAB8 951A8DAE 1B7695FE 2C07A044 B064742F

quit

spanning-tree mode pvst

spanning-tree portfast default

spanning-tree etherchannel guard misconfig

spanning-tree extend system-id

vlan internal allocation policy ascending

vlan 2-5,7

vlan 10

name DATA

vlan 11-12

vlan 20

name VOICE

vlan 29

vlan 30

name MGT

vlan 40

name MISC

vlan 50

name NATIVE

vlan 99

name MANAGEMENT

vlan 100,192,400,999

interface FastEthernet0/1

switchport trunk encapsulation dot1q

switchport trunk native vlan 50

switchport mode trunk

interface FastEthernet0/2

switchport access vlan 10

switchport mode access

switchport voice vlan 20

spanning-tree portfast

interface FastEthernet0/3

switchport access vlan 10

switchport mode access

switchport voice vlan 20

spanning-tree portfast

interface FastEthernet0/4

interface FastEthernet0/5

interface FastEthernet0/6

interface FastEthernet0/7

interface FastEthernet0/8

interface FastEthernet0/9

interface FastEthernet0/10

interface FastEthernet0/11

interface FastEthernet0/12

interface FastEthernet0/13

interface FastEthernet0/14

interface FastEthernet0/15

interface FastEthernet0/16

interface FastEthernet0/17

interface FastEthernet0/18

interface FastEthernet0/19

interface FastEthernet0/20

interface FastEthernet0/21

interface FastEthernet0/22

interface FastEthernet0/23

interface FastEthernet0/24

interface FastEthernet0/25

interface FastEthernet0/26

interface FastEthernet0/27

interface FastEthernet0/28

interface FastEthernet0/29

interface FastEthernet0/30

interface FastEthernet0/31

interface FastEthernet0/32

interface FastEthernet0/33

interface FastEthernet0/34

interface FastEthernet0/35

interface FastEthernet0/36

interface FastEthernet0/37

interface FastEthernet0/38

interface FastEthernet0/39

interface FastEthernet0/40

interface FastEthernet0/41

interface FastEthernet0/42

interface FastEthernet0/43

interface FastEthernet0/44

interface FastEthernet0/45

interface FastEthernet0/46

interface FastEthernet0/47

interface FastEthernet0/48

interface GigabitEthernet0/1

interface GigabitEthernet0/2

interface GigabitEthernet0/3

interface GigabitEthernet0/4

interface Vlan1

no ip address

shutdown

ip classless

ip http server

ip http secure-server

ip sla enable reaction-alerts

line con 0

line vty 0 4

login

line vty 5 15

login

end

We encountered several problems while doing this lab. The first one was just finding the few phones that would work in the collection of many. The second problem was that the article that we used did include most of the commands that we needed to get VoIP to work, but the ccm-manager and mgcp commands were missing and instead had parts that weren’t necessary for VoIP at all, including using an interface called “ServiceEngine,” which didn't even exist on our routers. Also, sometimes the phones that did work would reject the configuration, giving a “registration rejected” error and starting without a number. The router would give a syslog message saying that the phone “unregistered abnormally” and would “fallback” to the “g711ulaw” codec. I have no idea why that happened, and the only way that I knew to fix it was restarting the router and hoping the phones wouldn’t do that again. To save time, we eventually just saved any changes to the startup config (even though we weren’t really supposed to do that) so that we could just restart the router if something went wrong. The configuration register was 0x2142, so the router would ignore startup config when it boots, and people can just copy it into their running config when they need to work on the router. It took several days to weeks of virtually no progress until we found some information on the ccm-manager commands, which were apparently necessary for the router to work. Finally, I thought the destination pattern just had to start with 2-9 followed by the 10 digits, but it had to start with 91 followed by the 10 digits to make a call. Fortunately, we didn't have an accident where we misdialed 911

In conclusion, I set up VoIP on phones so they can call each other internally, then allowed them to dial out and call external numbers. I had several issues that I had to deal with, such as finding the right phones, getting the phones that did work to accept their configuration, realizing after a week that ServiceEngine wasn’t an interface, and that eventually our period and all other periods started putting our work into a configuration that we aren’t supposed to use just to get it done a little faster. Overall, I did get a better understanding of how VoIP works and how to set it up. Also, we now have a configuration still sitting around so that our classes can call a pizza store whenever we feel like, which is obviously a massive benefit. However, I still think I will use the stores’ websites because it didn't take several weeks to understand how to use those.