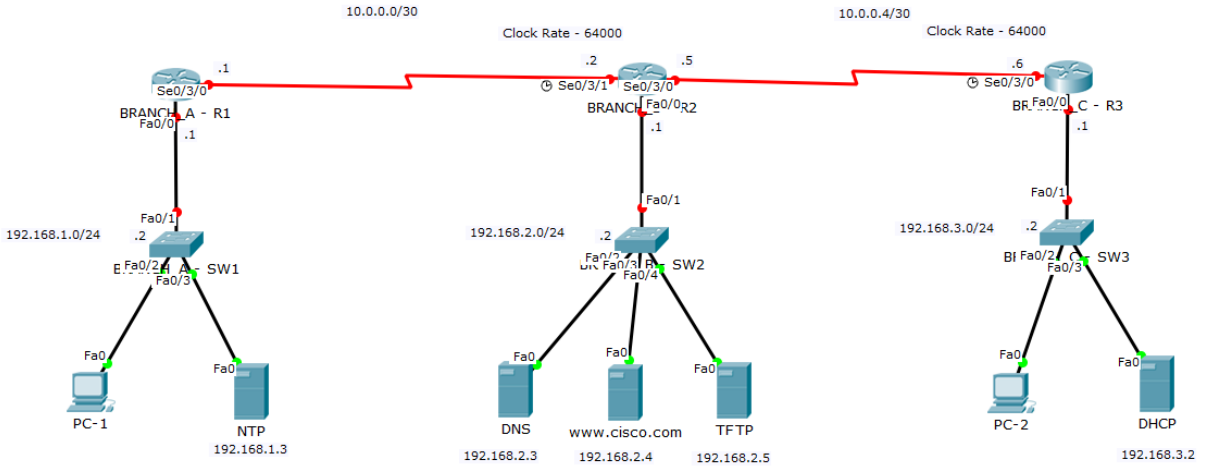
Cisco CCNA – iOS Administration Labs
Keith Gebhardt – www.learntechtraining.com

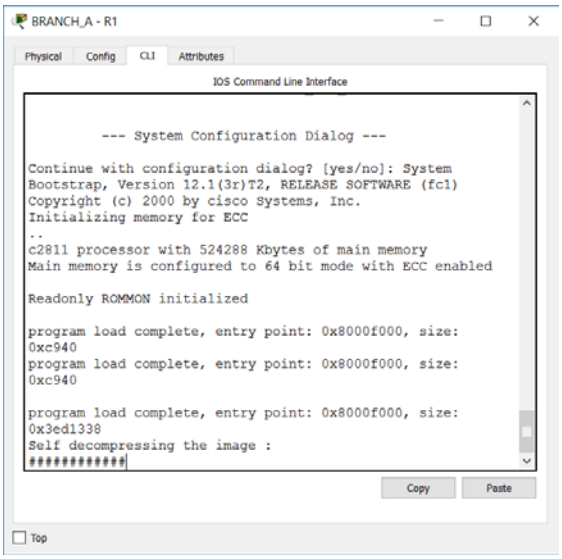
Topics to be covered:

- Cisco iOS Boot Sequence
- Navigating Cisco iOS
- Cisco iOS File System
- Router – Base / Administrative Configurations
    - Telnet/SSH
    - Interface
    - User Credentials
    - Encryption
- Switch – Base / Administrative Configurations
    - Telnet/SSH
    - Interfaces
    - VLANS
    - User Credentials
    - Encryption
- Default Routing
- Static Routing
- RIPv2
- Implementing NTP
- Implementing DNS Server
- Implementing TFTP
- Implementing SYSLOG
- Implementing DHCP (Router as a server)
- Backup/Restore Router iOS
- Password Recovery Procedure
- Verification Commands

Step 1 – Create the topology shown below

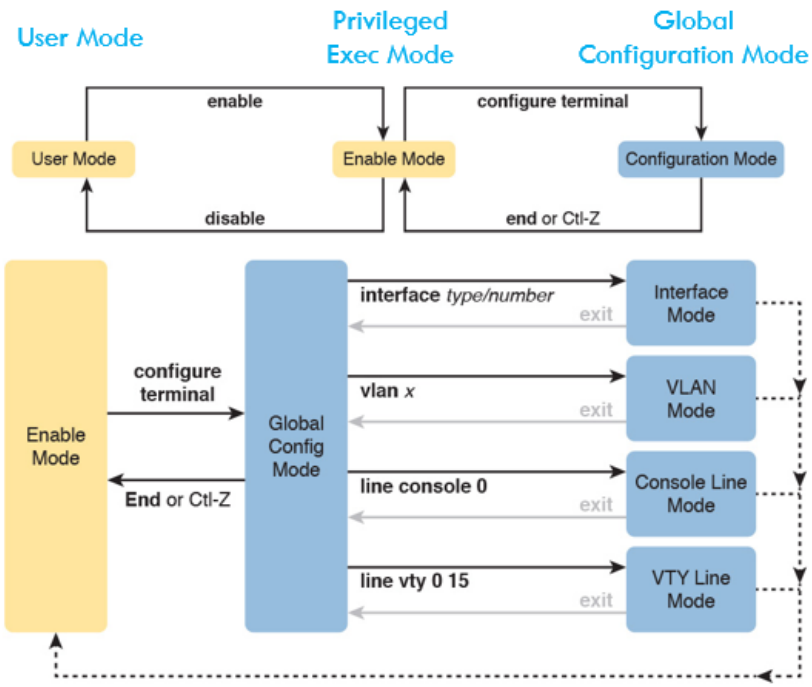


Step 2 – Familiarize yourself with router and switch boot sequence. Can view this by powering off the device, and powering the device back on.



- Notice you do not visually see POST or REGISTER during bootup from prompt.
- You do however, see the image (Cisco iOS) decompressing, represented by #
    - This is the point which Cisco iOS looks for configurations stored in NVRAM
        - If no configuration is found – On Routers – Sends you to Configuration Dialog
        - If no configuration is found – On Switches – Sends you to normal operating mode

Step 3 – Familiarize yourself with Cisco iOS Navigation



Helpful Commands:

ctrl + z = exit
ctrl + a = beginning of line text
ctrl + e = end of line text
ctrl + c = break
ctrl + shift + 6 = break
tab = finish command syntax

Step 4 – Familiarize yourself with Cisco File System

```
Router>en
Router#show flash

System flash directory:
File  Length   Name/status
  3   50938004 c2800nm-advipservicesk9-mz.124-15.T1.bin
  2   28282    sigdef-category.xml
  1   227537   sigdef-default.xml
[51193823 bytes used, 12822561 available, 64016384 total]
63488K bytes of processor board System flash (Read/Write)


Router#dir
Directory of flash:/

    3  -rw-    50938004          <no date>  c2800nm-advipservicesk9-mz.124-15.T1.bin
    2  -rw-       28282          <no date>  sigdef-category.xml
    1  -rw-      227537          <no date>  sigdef-default.xml

64016384 bytes total (12822561 bytes free)
Router#
```

Step 5 – Statically configure all Server IP Addresses to their appropriate network schemes

Step 6 – Configure Base Router & Switch Administrative Configurations.

All device configurations are below:

Cisco CCNA – iOS Administration Labs
Keith Gebhardt – www.learntechtraining.com

BRANCHA_R1

Privileged Exec Mode

_____

Router>**enable**
Router#**clock set 10:10:10 January 1 2020**
Router#**terminal history size 20**

Global Configuration Mode

_____

Router#**configure terminal**
Router(config)#**hostname BRANCHA_R1**
CCNALAB(config)#**line console 0** *[Console connection management]*
CCNALAB(config-line)#**password cisco**
CCNALAB(config-line)#**login** *[Required to be prompted for password during login]*
CCNALAB(config-line)#**logging synchronous** *[Used to keep messages from interrupting line-text place]*
CCNALAB(config-line)#**exec-timeout 0 0** *[Sets a 'idle' timer for which the console session ends]*
CCNALAB(config-line)#**exit**
CCNALAB(config)#**line vty 0 4** *[Remote Connection management]*
CCNALAB(config-line)#**password remote**
CCNALAB(config-line)#**login local**
CCNALAB(config-line)#**logging synchronous**
CCNALAB(config-line)#**exec-timeout 0 0**
CCNALAB(config-line)#**transport input all***[Set Telnet, SSH, or ALL –typically only use SSH]*
CCNALAB(config-line)#**exit**
CCNALAB(config)# **banner motd &**
Enter TEXT message. End with the character '&'.
****
AUTHORIZED USERS ONLY!!
********** **&**

[Remember, there are three different types of Banners. Should only expect to know MOTD for exam]

CCNALAB(config)#**cdp run** *[Usually on by default]*
CCNALAB(config)#**lldp run**
CCNALAB(config)#**no ip domain-lookup** *[Usually off by default – pending iOS]*
CCNALAB(config)#**service password-encryption**

Required for SSH Connection

_____

CCNALAB(config)#**enable secret cisco** *[MD5 Hash Encryption – Secure]*
CCNALAB(config)#**username admin privilege 1 secret cisco** *[MD5 Hash Encryption, privilege requires enable mode]*
CCNALAB(config)#**ip domain-name cisco.com**
CCNALAB(config)#**crypto key generate rsa**
The name for the keys will be: CCNALAB.cisco.com
Choose the size of the key modulus in the range of 360 to 2048 for your
General Purpose Keys. Choosing a key modulus greater than 512 may take
a few minutes.

How many bits in the modulus [512]: **1024**
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]
*Jan 1 10:40:2.157: %SSH-5-ENABLED: SSH 1.99 has been enabled
CCNALAB(config)#**ip ssh version 2**

CCNALAB(config)#**ip ssh time-out 120**
CCNALAB(config)#**ip ssh authentication-retries 3**
Required for Remote Connection

_____
CCNALAB(config)#**interface f0/0**
CCNALAB(config-if)#**ip add 192.168.1.1 255.255.255.0**
CCNALAB(config-if)#**description LINK TO 192.168.1.0 NETWORK**
CCNALAB(config-if)#**no shut**

CCNALAB(config-if)#

Saving Current Running-Configuration to NVRAM (Startup-Configuration)

_____
CCNALAB#**copy run start**
Destination filename [startup-config]?
Building configuration...
[OK]
CCNALAB#

BRANCHA_SW1

Privileged Exec Mode

_____
Router>**enable**
Router#**clock set 10:10:10 January 1 2020**
Router#**terminal history size 20**

Global Configuration Mode

_____
Router#**configure terminal**
Router(config)#**hostname BRANCHA_SW1**
CCNALAB(config)#**line console 0** *[Console connection management]*
CCNALAB(config-line)#**password cisco**
CCNALAB(config-line)#**login** *[Required to be prompted for password during login]*
CCNALAB(config-line)#**logging synchronous** *[Used to keep messages from interrupting line-text place]*
CCNALAB(config-line)#**exec-timeout 0 0** *[Sets a 'idle' timer for which the console session ends]*
CCNALAB(config-line)#**exit**
CCNALAB(config)#**line vty 0 4** *[Remote Connection management]*
CCNALAB(config-line)#**password remote**
CCNALAB(config-line)#**login local**
CCNALAB(config-line)#**logging synchronous**
CCNALAB(config-line)#**exec-timeout 0 0**
CCNALAB(config-line)#**transport input all***[Set Telnet, SSH, or ALL –typically only use SSH]*
CCNALAB(config-line)#**exit**
CCNALAB(config)# **banner motd &**
Enter TEXT message. End with the character '&'.
****
AUTHORIZED USERS ONLY!!
********** **&**

Cisco CCNA – iOS Administration Labs
Keith Gebhardt – www.learntechtraining.com

[Remember, there are three different types of Banners. Should only expect to know MOTD for exam]

CCNALAB(config)#**cdp run** *[Usually on by default]*
CCNALAB(config)#**lldp run**
CCNALAB(config)#**no ip domain-lookup** *[Usually off by default – pending iOS]*
CCNALAB(config)#**service password-encryption**

Required for SSH Connection

_____

CCNALAB(config)#**enable secret cisco** *[MD5 Hash Encryption – Secure]*
CCNALAB(config)#**username admin privilege 1 secret cisco** *[MD5 Hash Encryption, privilege requires enable mode]*
CCNALAB(config)#**ip domain-name cisco.com**
CCNALAB(config)#**crypto key generate rsa**
The name for the keys will be: CCNALAB.cisco.com
Choose the size of the key modulus in the range of 360 to 2048 for your
General Purpose Keys. Choosing a key modulus greater than 512 may take
a few minutes.

How many bits in the modulus [512]: **1024**
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]
*Jan 1 10:40:2.157: %SSH-5-ENABLED: SSH 1.99 has been enabled
CCNALAB(config)#**ip ssh version 2**
CCNALAB(config)#**ip ssh time-out 120**
CCNALAB(config)#**ip ssh authentication-retries 3**
Required for Remote Connection

_____

CCNALAB(config)#**vlan 10**
CCNALAB(config-vlan)#**name ADMIN**
CCNALAB(config-vlan)#**vlan 20**
CCNALAB(config-vlan)#**name MISC**
CCNALAB(config-vlan)#**exit**
CCNALAB(config)#**int range f0/1-3**
CCNALAB(config-if)#**switchport mode access** *[NEVER LEAVE MODE IN DYNAMIC!!!]*
CCNALAB(config-if)#**switchport access vlan 10** *[Assigns Ports to VLAN 10]*
CCNALAB(config-if)#**int range f0/4-24**
CCNALAB(config-if)#**switchport mode access**
CCNALAB(config-if)#**switchport access vlan 20**
CCNALAB(config-if)#**shutdown**
CCNALAB(config-if)#**exit**
CCNALAB(config)#**interface vlan 10**
CCNALAB(config-if)#**ip add 192.168.1.2 255.255.255.0**
CCNALAB(config-if)#**no shut**
BRANCHA_SW1(config)#**ip default-gateway 192.168.1.1** *[Used so other devices in other networks can remote to the switch]*

CCNALAB(config-if)#

Saving Current Running-Configuration to NVRAM (Startup-Configuration)

_____

CCNALAB#**copy run start**
Destination filename [startup-config]?
Building configuration...
[OK]

CCNALAB#

Follow Same Configurations for ALL ROUTERS and SWITCHES – MAKE SURE TO USE PROPER NAME AND ADDRESSES!

Step 7 – Configure Router Interfaces connecting all routers!

BRANCHA_R1

BRANCHA_R1(config)#**int s0/3/0**
BRANCHA_R1(config-if)#**ip add 10.0.0.1 255.255.255.252**
BRANCHA_R1(config-if)#**description LINK TO BRANCHB_R1-10.0.0.2-S0/3/1**
BRANCHA_R1(config-if)#**no shut**
BRANCHA_R1(config-if)#**exit**
BRANCHA_R1(config)#

BRANCHB_R1

BRANCEB_R2(config)#**int s0/3/1**
BRANCEB_R2(config-if)#i**p add 10.0.0.2 255.255.255.252**
BRANCEB_R2(config-if)#**clock rate 64000**
BRANCEB_R2(config-if)#**description LINK TO BRANCHA_R1-10.0.0.1-S0/3/0**
BRANCEB_R2(config-if)#**no shut**
BRANCEB_R2(config)#**int s0/3/0**
BRANCEB_R2(config-if)#**ip add 10.0.0.5 255.255.255.252**
BRANCEB_R2(config-if)# **description LINK TO BRANCHC_R1-10.0.0.6-S0/3/0**
BRANCEB_R2(config-if)#**no shut**
BRANCEB_R2(config-if)#**exit**

BRANCHC_R1

BRANCHC_R1(config)#**int s0/3/0**
BRANCHC_R1(config-if)#**ip add 10.0.0.6 255.255.255.252**
BRANCHC_R1(config-if)#**clock rate 64000**
BRANCEB_R2(config-if)# **description LINK TO BRANCHB_R1-10.0.0.5-S0/3/0**
BRANCHC_R1(config-if)#**no shut**
BRANCHC_R1(config-if)#

Copy running-config startup-config

Step 8 – Configure Both Edge/Stub routers to have a Gateway of Last Resort

BRANCHA_R1

BRANCHA_R1(config)#**ip route 0.0.0.0 0.0.0.0 s0/3/0**

BRANCHC_R1

BRANCHC_R1(config)#**ip route 0.0.0.0 0.0.0.0 s0/3/0**

Copy running-config startup-config

Show ip route for verification

Step 9 – Configure Static Routes on BRANCHB_R1 so all 3 networks can communicate!

BRANCEB_R2(config)#**ip route 192.168.1.0 255.255.255.0 10.0.0.1**
BRANCEB_R2(config)#**ip route 192.168.3.0 255.255.255.0 10.0.0.**

Ping 192.168.1.1 – Ping 192.168.3.1 : This will verfify connectivity
Show ip route for verification
copy run start

Step 10 – Configure RIPv2 on all Routers!

BRANCHA_R1

BRANCHA_R1(config)#**router rip**
BRANCHA_R1(config-router)#**version 2**
BRANCHA_R1(config-router)#**no auto-summary**
BRANCHA_R1(config-router)#**network 192.168.1.0**
BRANCHA_R1(config-router)#**network 10.0.0.0**
BRANCHA_R1(config-router)#**exit**

BRANCHB_R1

BRANCEB_R2(config)#**router rip**
BRANCEB_R2(config-router)#**version 2**
BRANCEB_R2(config-router)#**no auto-summary**
BRANCEB_R2(config-router)#**network 192.168.2.0**
BRANCEB_R2(config-router)#**network 10.0.0.0**
BRANCEB_R2(config-router)#**exit**

BRANCHC_R1

BRANCHC_R1(config)#**router rip**
BRANCHC_R1(config-router)#**version 2**
BRANCHC_R1(config-router)#**no auto-summary**
BRANCHC_R1(config-router)#**network 192.168.3.0**
BRANCHC_R1(config-router)#**network 10.0.0.0**
BRANCHC_R1(config-router)#**exit**

Copy run start
show ip route to verify
ping all networks to verify connectivity

Looking at BRANCHB_R1 Router's Routing Table, You do not see the RIP routing protocols ROUTES listed.

WHY?

- The reason is, we configured TWO STATIC ROUTES which have an AD of 1. RIP by default has an AD of 120.

SOLUTION?

1. We need to go back onto router BRANCHB_R1, and remove those two static routes.
2. We need to reconfigure the static routes to have an AD higher than RIP. We do this incase something happened to the RIP protocol, the router still knows how to forward packets to that network.

Step 11 – Configure BRANCHB_R1 Router to have static routes of a AD higher than RIP!

BRANCEB_R2(config)**#no ip route 192.168.3.0 255.255.255.0 10.0.0.6**
BRANCEB_R2(config)**#no ip route 192.168.1.0 255.255.255.0 10.0.0.1**
BRANCEB_R2(config)**#ip route 192.168.1.0 255.255.255.0 10.0.0.1 150**
BRANCEB_R2(config)**#ip route 192.168.3.0 255.255.255.0 10.0.0.6 150**

Now, look at the Routing Table for BRANCHB_R1 once again, and you will now see that RIP has been updated in the routing table!

Step 12 – Configure all ROUTERS to use NTP server for Time!

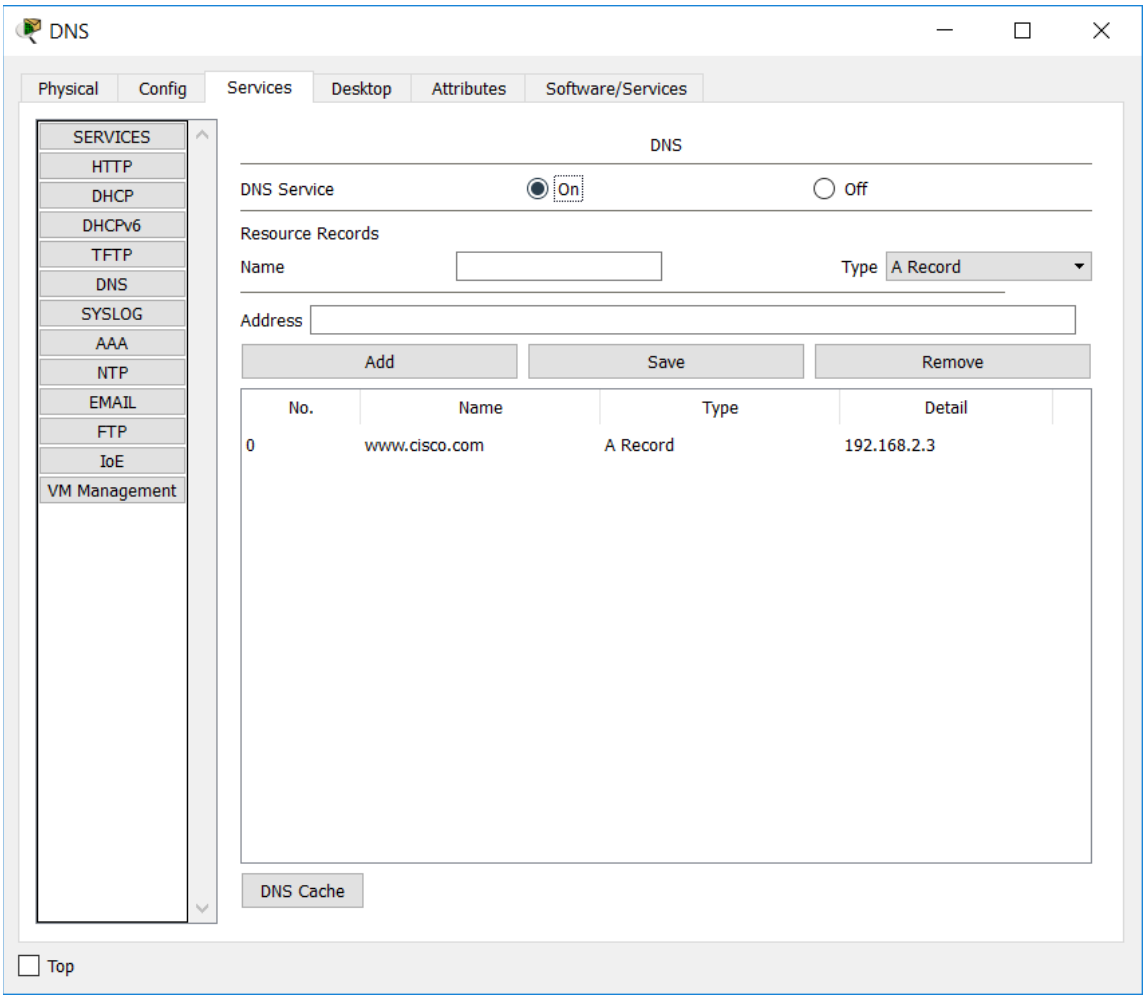
BRANCEB_R2(config)**#ntp server 192.168.1.3**

Use show ntp status command to verify

Copy run start

Step 13 – Configure DNS Server on all ROUTERS

- You will need to configure the actual DNS server to recognize Domain Names to their Servers Address.
- Also make sure all hosts computers or servers are using the DNS address!



BRANCHA_R1(config)#**ip domain-lookup**
BRANCHA_R1(config)#**ip name-server 192.168.2.3**

Step 13 – Configure TFTP, and BACKUP/RESTORE CISCO iOS

We know we have connectivity to our TFTP Server. Now we just need to backup our current running config with the iOS, and then restore it using our file system commands and our tftp server.

We are going to use BRANCHA_R1 for this section.

Step 13-A:

Verify Current Running Config by using “show running-config”
Verify Current Startup-Config by using “show startup-config”
Copy R S to make sure everything is currently SAVED!

_____

BRANCHA_R1#**show flash**

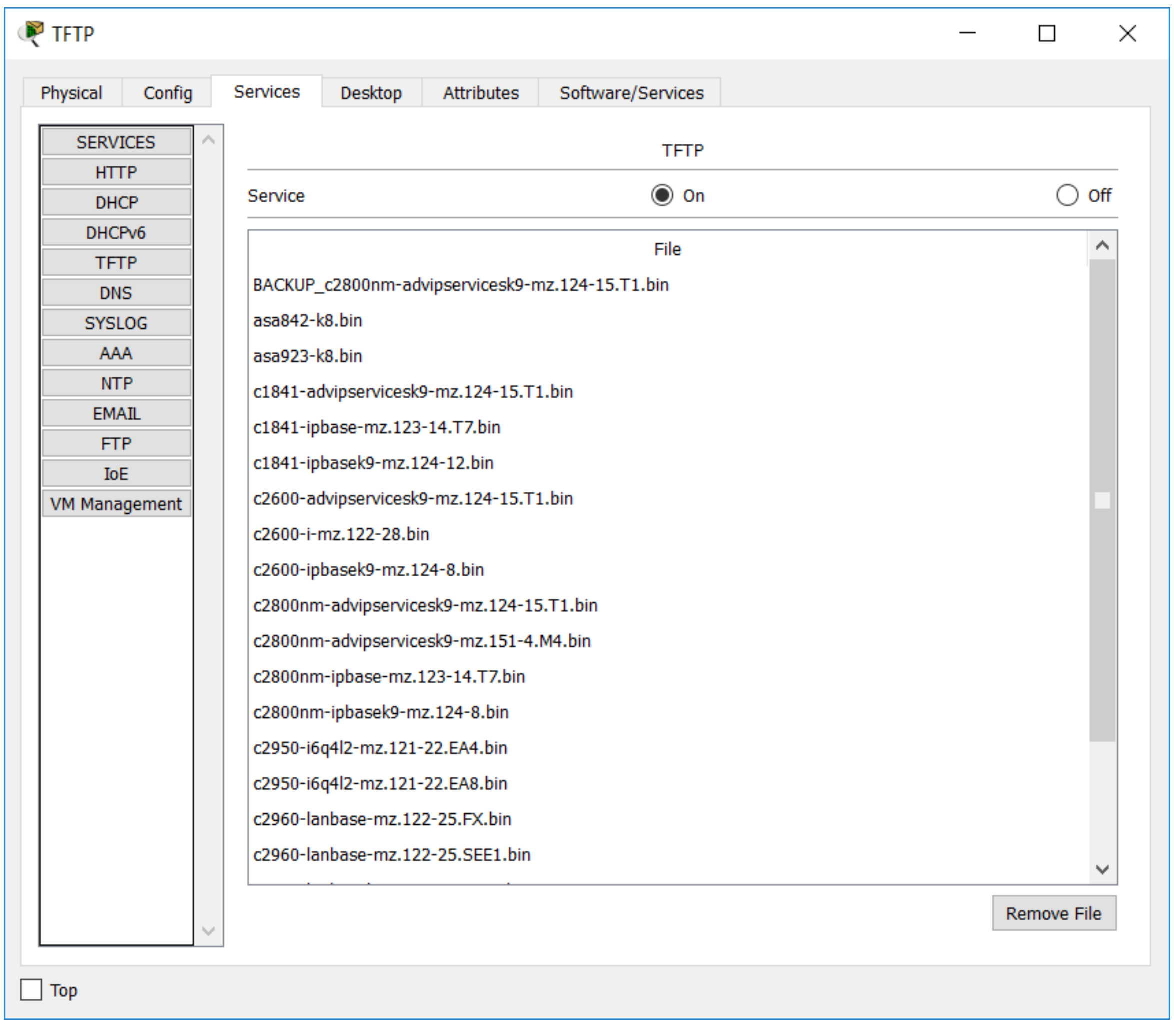
System flash directory:

File Length Name/status
3 50938004 c2800nm-advipservicesk9-mz.124-15.T1.bin
2 28282 sigdef-category.xml
1 227537 sigdef-default.xml
[51193823 bytes used, 12822561 available, 64016384 total]
63488K bytes of processor board System flash (Read/Write)

BRANCHA_R1#**copy flash tftp**
Source filename []? **c2800nm-advipservicesk9-mz.124-15.T1.bin**
Address or name of remote host []? **192.168.2.5**
Destination filename [c2800nm-advipservicesk9-mz.124-15.T1.bin]? **BACKUP_c2800nm-advipservicesk9-mz.124-15.T1.bin**

Writing c2800nm-advipservicesk9-mz.124-15.T1.bin...!!

____
Verify File was saved to TFTP server



Step 13-B:

lets go ahead and change the name of our router

Hostname R1_BRANCHA
Copy running-config startup config

This will allow us to see that we successfully imported the correct file!

Step 13-C:

Now we need to go ahead and save the startup-config file to the TFTP server

R1_BRANCHA#**copy startup-config tftp**
Address or name of remote host []? **192.168.2.5**
Destination filename [R1_BRANCHA-confg]?
Writing startup-config...!!
[OK - 1252 bytes]
1252 bytes copied in 0.005 secs (250400 bytes/sec)
R1_BRANCHA#

Step 13-D:

We now will erase the current configuration from the router!

R1_BRANCHA#**write erase**
Erasing the nvram filesystem will remove all configuration files! Continue? [confirm]
[OK]
Erase of nvram: complete
%SYS-7-NV_BLOCK_INIT: Initialized the geometry of nvram
R1_BRANCHA#**show start**
startup-config is not present
R1_BRANCHA#

Notice, when we now do a show start, we have no configurations!

We still see our hostname though. That's because we still have a running-config file. Remember that the running-config file is RAM, so we need to tell the router to reboot so we starting from scratch

Simply type in reload at privileged mode!

Step 13-E:

So how do we start communicating back to our TFTP server to recover our last configurations we backed up.

Well, since we have no current configurations – we need to at least address the interface heading to the TFTP server.

We also need to tell the router to use RIP to reach that network.

Router(config)#**int s0/3/0**
Router(config-if)#**ip add 10.0.0.1 255.255.255.252**
Router(config-if)#**no shut**
Router(config-if)#**exit**
Router(config)#**router rip**

Router(config-router)#**version 2**
Router(config-router)#**no auto-summary**
Router(config-router)#**network 10.0.0.0**
Router(config-router)#**exit**
Router(config)#**do ping 192.168.2.5**

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 6/9/16 ms

Router(config)#

Step 13-F:

We will now pull our old configuration back from the TFTP server.

Router#**copy tftp running-config**
Address or name of remote host []? **192.168.2.5**
Source filename []? **R1_BRANCHA-confg**
Destination filename [running-config]?

Accessing tftp://192.168.2.5/R1_BRANCHA-confg...
Loading R1_BRANCHA-confg from 192.168.2.5: !
[OK - 1252 bytes]

1252 bytes copied in 0 secs
R1_BRANCHA#
%SYS-5-CONFIG_I: Configured from console by console

R1_BRANCHA#

In the output above, we can see we successfully retrieved our configurations. We know this already by seeing the "Hostname" has already been changed.

You can even do a "show run" to verify.

Make sure you do a copy running-config startup-config to save the file back to NVRAM!

R1_BRANCHA#**copy r s**

Step 14 – Configure SYSLOG

Were going to simply configure our R1_BRANCHA router to log informational information to a syslog server. In our lab, the SYSLOG server will be the same as our TFTP server.

This will be the basic level of configurations as Cisco Packet Tracer really limits you on the complexity of certain commands. This will give you experience at least getting SYSLOG running, and then when you have the opportunity to play with real equipment (or even GNS3) and use your computer as a server, you can really get into the details.

Note for your CCNA, you should only need to have an understanding of SYSLOG and its Purpose.

First, we need to turn SYSLOG on and tell it where to store the logging information.

R1_BRANCHA(config)#**logging trap debugging**
R1_BRANCHA(config)#**logging 192.168.2.5**

If you look at your SYSLOG server, running on our TFTP server – you will start seeing logging data. However, the only time shown is the hardcoded time we set during our Base configurations.

Lets sync our Logging Information with the NTP server so we have a better way of tracking the information. This is very useful for trying to find root cause of an issue.

R1_BRANCHA(config)#**service timestamps log datetime msec**

And that is it. Go ahead and shut down fa0/0 and turn it back on – go back to your syslog server and view the new output that is shown!

Step 15 – Configure DHCP Server and Relay

There are two ways we can set up DHCP services.

1.) We can configure a server to run as a DHCP server
2.) We can configure a router to act as a DHCP server

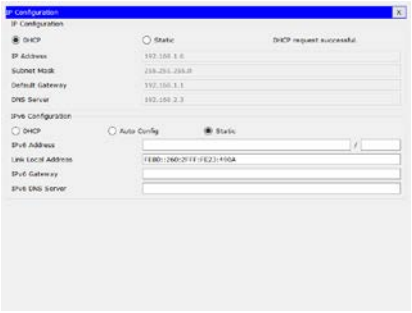
Each network has different needs. Remember, no two networks are the same. We will take a look at setting up DHCP on both!

First – We are going to Setup DHCP on R1_BRANCHA Server, to act as the DHCP Server and dynamically distribute IP addresses to the devices that connect to that switch.

R1_BRANCHA(config)#**ip dhcp excluded-address 192.168.1.1 192.168.1.5**
R1_BRANCHA(config)#**ip dhcp pool BRANCHA_NETWORK**
R1_BRANCHA(dhcp-config)#**network 192.168.1.0 255.255.255.0**
R1_BRANCHA(dhcp-config)#**default-router 192.168.1.1**
R1_BRANCHA(dhcp-config)#**dns-server 192.168.2.3**
R1_BRANCHA(dhcp-config)#**exit**

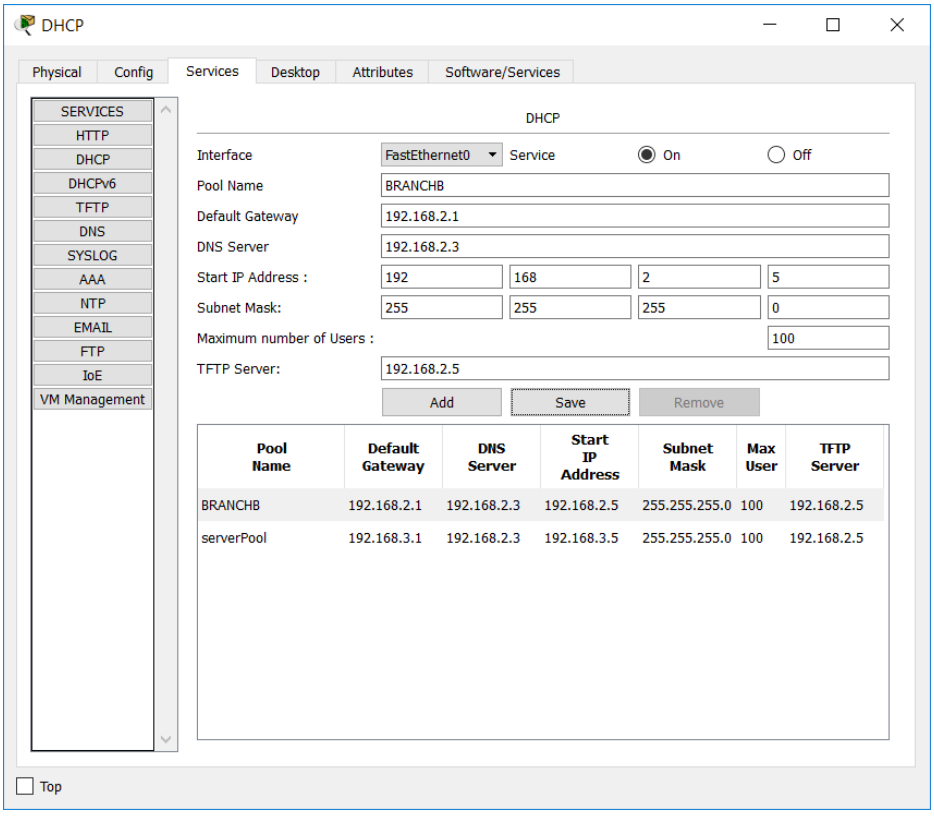
In the output above, we excluded the address range from 1.1 to 1.5, this way we can statically configure our default gateway, switch management, and server IP addresses. These are addresses we do not want being dynamically learned because we need to know the addresses to those devices and they need to remain consistant.

If we go to our Desktop PC, click on Dynamic DHCP, we will see the addressing populates.



Next, Lets go ahead and configure our DHCP Server to dish out the IP Addresses, and then we will implement the "ip helper" command on the routers so other networks can pull from it.

You can see the first server pool is named "serverPool", that is PacketTracer's default – its just easier to leave it in there. But we needed to configure two server pools, one for each network we are handing out IP address to!



We will only configure BRANCHB_R1 and BRANCHC_R1 to obtain DHCP Ip addresses from this server.

To accomplish this, we really only need to tell BRANCHB_R1's F0/0 interface to have the ip helper relay command because the 192.168.3.0 network is sitting where the DHCP server resides.

BRANCEB_R2(config)**#int f0/0**
BRANCEB_R2(config-if)**#ip helper-address 192.168.3.3**

Add a laptop, and verify it receives DHCP address.

Test all network connectivity.

Step 16 – Gateway of Last resort | Connect to the Internet

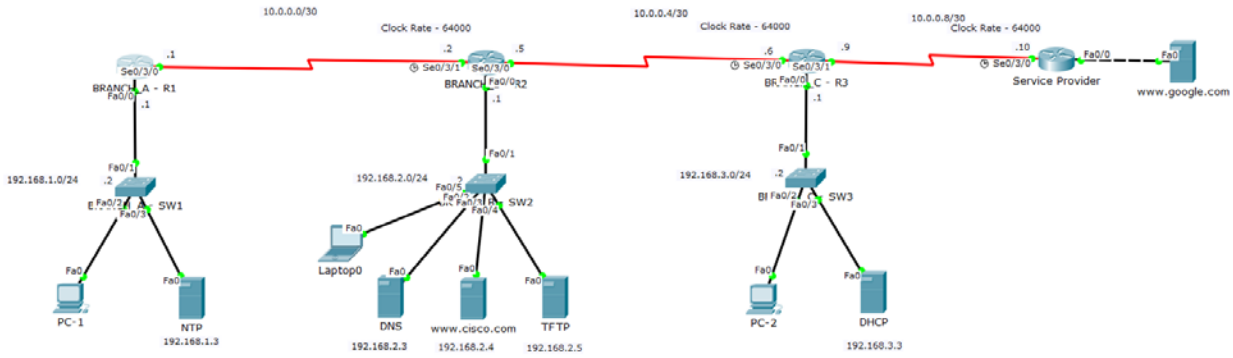
Finally, we need to finish configuring out routed network to utilize a Default Route which is advertised by our routing Protocol – RIP

We will also configure Passive Interface on our routers to help eliminate the possibility of rouge Routers or devices plugging in and potentially taking down our network.

To accomplish this.. we simply create a “mock” service provider network.
(configure RIP, and a google webserver to have a device to connect to)

We then will remove the default routes on our two edge routers.

We will then set up BRACHC_R1 to act as our new “Edge or STUB” router, assign it a new default route which points to the internet service provider, and we issue RIP the command “**ip default information originate**” command and simply watch the magic happen!



Be sure to verify all connectivity.

Step 17 – OH NO! I FORGOT MY PASSSWORD!

Well, Monday arrives and its time to login to your Switch or Router, and try after try – It fails.

I deliberately saved this topic for the end because its very good to understand how to perform password recovery – and it helps you understand your register values. I didn’t want to bring this topic up right as we backup/restored our iOS to the TFTP server, because although it is similar, there are a few things that are different.

Its not often we forget our passwords, so by waiting until the end, it also simulates some “time” has passed and now you need to perform this process.

Also keep in mind, that we typically control our accounts through a TACAS or RADIUS server for AAA authentication.

Step 17 – A:

We need to first shut the router down. Something you do not want to do during operational hours.

Once you power down the router, wait a few seconds and power it back on.

In the CLI, you want to use your “BREAK” command.
CTRL+SHIFT+C

You should now be in rommon.

rommon 1 > **confreg 0x2142**
rommon 2 > **reset**

Using the confreg 0x2142 command, you are bypassing the startup-config and going right into RAM.

When you use the reset command, you are rebooting the router.

At this point you will be prompted with a standard interface as if the switch or router was never configured before.

Step 17 – B:

- Now we can go ahead and go to privilege mode.
- Do a show start and you will see the startup configuration stored in NVRAM which we bypassed during boot up.
- If you do a show run, you will see a default router or switch configuration file.
- We now need to copy that Startup configuration file to running configuration (RAM)

Router#**copy start run**

- Now, do a show run and you should see everything as normal.
- **RECONFIGURE PASSWORD**
- **NOW CHANGE THE BOOT REGISTER BACK TO 0x2102**!
- **SAVE YOUR RUNNING CONFIG TO STARTUP CONFIG**!

R1_BRANCHA#**conf t**.
R1_BRANCHA(config)#**config-reg 0x2102**
R1_BRANCHA(config)#**exit**
R1_BRANCHA#**copy run start**

You are not DONE! Do a show version and you will see:

R1_BRANCHA#**show version**

```
Configuration register is 0x2142 (will be 0x2102 at next reload)
```

ONCE YOU SAVE YOUR RUNNING CONFIG (RAM) TO STARTUP CONFIG (NVRAM), Reboot the device!

R1_BRANCHA#**reload**

Test it!

VERIFICATION COMMANDS TO KNOW:

show running-config
show startup-config
dir
show flash
show interface #
show ip interface brief
show ip route
show ip protocols
show ip rip database
show cdp neighbors
show controllers
show session
show ntp status
show vlan brief
show arp
show version
ping
traceroute
debug