Ryan Munger

Internetworking

Professor Cannistra

9/19/23

Lab 1

Description: In this lab, we practiced basic commands in Cisco IOS. We created a network in packet tracer and configured the virtual networking equipment. We verified every device in the topology was connected and ensured that the switches were configured correctly as well as able to maintain a console or telnet connection.

Topology/Diagram:
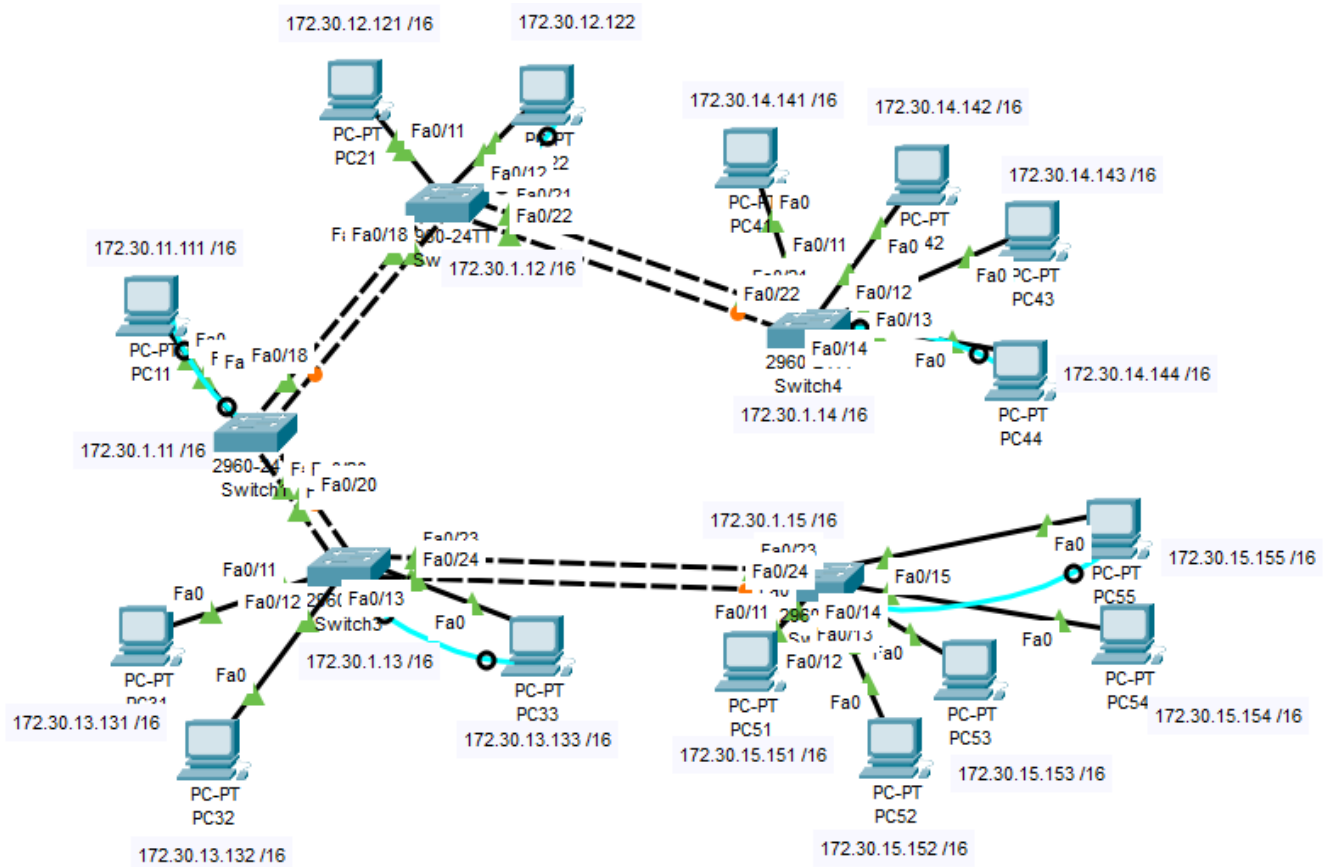
Syntax:

CLI	Command	Description	Mode
Cisco IOS	enable	Enter privilege exec mode	User
Cisco IOS	configure terminal	Enter global configuration mode	Priv Exec
Cisco IOS	hostname	Change the hostname of the device	Global Config
Cisco IOS	line console 0	Enter line configuration mode for console port 0	Config Modes
Cisco IOS	password	Sets password for selected configuration	Config Modes
Cisco IOS	login	Enables the set password to be prompted for	Config Modes
Cisco IOS	interface vlan 1	Enter configuration for vlan 1 interface	Config Modes
Cisco IOS	ip address	Set ip address for interface	Interface Config
Cisco IOS	line vty 0 15	Enter line configuration mode for virtual terminal connection lines 0-15	Config Modes
Cisco IOS	show ip interface brief	Shows information about the ip interfaces installed in brief format	User
Cisco IOS	show interface vlan 1	Displays information about vlan 1 interface such as ip address	User
Cisco IOS	end	Returns to privilege exec mode	Any but user
Cisco IOS	show running-config	Shows info about the current device configuration stored in RAM	Priv Exec
Cisco IOS	copy running-config startup-config	Copies the current config to the saved config used on startup in NVRAM	Priv Exec
Cisco IOS	no shutdown	Enables an interface	Interface Config
Cisco IOS	show mac address-table	Displays switch's mac address table	User
CMD	ping	tests for connectivity to specified end device by sending packets	N/A
CMD	arp -a	displays full arp cache	N/A

Verification:

Is this feasible to test connectivity to each host within the network manually?
It is not feasible to manually test each host on a mid to large-scale network.

What could you do to improve upon the efficiency of testing connectivity?
A few ways you could improve the efficiency are: writing a script that pings the required hosts, checking only one host from each switch (if there were a problem it would be with the host or cable itself), or broadcasting a ping to the entire network (although many hosts block these for security reasons).

Testing point: PC44, screenshots organized by switch#

Switch1 and Switch2 Hosts: Switch3 Hosts:

```
C:\>ping 172.30.11.111

Pinging 172.30.11.111 with 32 bytes of data:

Reply from 172.30.11.111: bytes=32 time=8ms TTL=128
Reply from 172.30.11.111: bytes=32 time<1ms TTL=128
Reply from 172.30.11.111: bytes=32 time<1ms TTL=128
Reply from 172.30.11.111: bytes=32 time<1ms TTL=128

Ping statistics for 172.30.11.111:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 8ms, Average = 2ms

C:\>ping 172.30.12.121

Pinging 172.30.12.121 with 32 bytes of data:

Reply from 172.30.12.121: bytes=32 time<1ms TTL=128
Reply from 172.30.12.121: bytes=32 time=8ms TTL=128
Reply from 172.30.12.121: bytes=32 time<1ms TTL=128
Reply from 172.30.12.121: bytes=32 time<1ms TTL=128

Ping statistics for 172.30.12.121:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 8ms, Average = 2ms

C:\>ping 172.30.12.122

Pinging 172.30.12.122 with 32 bytes of data:

Reply from 172.30.12.122: bytes=32 time<1ms TTL=128
Reply from 172.30.12.122: bytes=32 time<1ms TTL=128
Reply from 172.30.12.122: bytes=32 time<1ms TTL=128
Reply from 172.30.12.122: bytes=32 time<1ms TTL=128

Ping statistics for 172.30.12.122:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

```
C:\>ping 172.30.13.131

Pinging 172.30.13.131 with 32 bytes of data:

Reply from 172.30.13.131: bytes=32 time<1ms TTL=128
Reply from 172.30.13.131: bytes=32 time<1ms TTL=128
Reply from 172.30.13.131: bytes=32 time<1ms TTL=128
Reply from 172.30.13.131: bytes=32 time<1ms TTL=128

Ping statistics for 172.30.13.131:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>ping 172.30.13.132

Pinging 172.30.13.132 with 32 bytes of data:

Reply from 172.30.13.132: bytes=32 time=1ms TTL=128
Reply from 172.30.13.132: bytes=32 time<1ms TTL=128
Reply from 172.30.13.132: bytes=32 time<1ms TTL=128
Reply from 172.30.13.132: bytes=32 time<1ms TTL=128

Ping statistics for 172.30.13.132:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>ping 172.30.13.133

Pinging 172.30.13.133 with 32 bytes of data:

Reply from 172.30.13.133: bytes=32 time<1ms TTL=128
Reply from 172.30.13.133: bytes=32 time<1ms TTL=128
Reply from 172.30.13.133: bytes=32 time<1ms TTL=128
Reply from 172.30.13.133: bytes=32 time<1ms TTL=128

Ping statistics for 172.30.13.133:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Switch4 Hosts:

```
C:\>ping 172.30.14.141

Pinging 172.30.14.141 with 32 bytes of data:

Reply from 172.30.14.141: bytes=32 time<1ms TTL=128
Reply from 172.30.14.141: bytes=32 time=8ms TTL=128
Reply from 172.30.14.141: bytes=32 time<1ms TTL=128
Reply from 172.30.14.141: bytes=32 time<1ms TTL=128

Ping statistics for 172.30.14.141:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 8ms, Average = 2ms

C:\>ping 172.30.14.142

Pinging 172.30.14.142 with 32 bytes of data:

Reply from 172.30.14.142: bytes=32 time<1ms TTL=128
Reply from 172.30.14.142: bytes=32 time=9ms TTL=128
Reply from 172.30.14.142: bytes=32 time<1ms TTL=128
Reply from 172.30.14.142: bytes=32 time<1ms TTL=128

Ping statistics for 172.30.14.142:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 9ms, Average = 2ms

C:\>ping 172.30.14.143

Pinging 172.30.14.143 with 32 bytes of data:

Reply from 172.30.14.143: bytes=32 time<1ms TTL=128
Reply from 172.30.14.143: bytes=32 time=1ms TTL=128
Reply from 172.30.14.143: bytes=32 time<1ms TTL=128
Reply from 172.30.14.143: bytes=32 time<1ms TTL=128

Ping statistics for 172.30.14.143:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>ping 172.30.14.144

Pinging 172.30.14.144 with 32 bytes of data:

Reply from 172.30.14.144: bytes=32 time=4ms TTL=128
Reply from 172.30.14.144: bytes=32 time=5ms TTL=128
Reply from 172.30.14.144: bytes=32 time<1ms TTL=128
Reply from 172.30.14.144: bytes=32 time=5ms TTL=128

Ping statistics for 172.30.14.144:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 5ms, Average = 3ms
```

Switch5 Hosts:

```
Ping statistics for 172.30.15.151:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 10ms, Average = 2ms

C:\>ping 172.30.15.152

Pinging 172.30.15.152 with 32 bytes of data:

Reply from 172.30.15.152: bytes=32 time<1ms TTL=128
Reply from 172.30.15.152: bytes=32 time=10ms TTL=128
Reply from 172.30.15.152: bytes=32 time=10ms TTL=128
Reply from 172.30.15.152: bytes=32 time=1ms TTL=128

Ping statistics for 172.30.15.152:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 10ms, Average = 5ms

C:\>ping 172.30.15.153

Pinging 172.30.15.153 with 32 bytes of data:

Reply from 172.30.15.153: bytes=32 time=10ms TTL=128
Reply from 172.30.15.153: bytes=32 time=1ms TTL=128
Reply from 172.30.15.153: bytes=32 time=1ms TTL=128
Reply from 172.30.15.153: bytes=32 time=1ms TTL=128

Ping statistics for 172.30.15.153:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 10ms, Average = 3ms

C:\>ping 172.30.15.154

Pinging 172.30.15.154 with 32 bytes of data:

Reply from 172.30.15.154: bytes=32 time<1ms TTL=128
Reply from 172.30.15.154: bytes=32 time=1ms TTL=128
Reply from 172.30.15.154: bytes=32 time<1ms TTL=128
Reply from 172.30.15.154: bytes=32 time=1ms TTL=128

Ping statistics for 172.30.15.154:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>ping 172.30.15.155

Pinging 172.30.15.155 with 32 bytes of data:

Reply from 172.30.15.155: bytes=32 time<1ms TTL=128
```

Local Arp Cache of PC52:   It only contains PC44, the testing point of our connectivity test

```
C:\>arp -a
  Internet Address      Physical Address      Type
  172.30.14.144         0003.e42d.144d        dynamic
```

Mac Address Table of Switch3:

```
Switch3>show mac address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address        Type        Ports
----    -----------        --------    -----

   1    0001.43ee.a713     DYNAMIC     Fa0/19
   1    0001.6309.a995     DYNAMIC     Fa0/12
   1    0003.e42d.144d     DYNAMIC     Fa0/19
   1    000a.415c.6d43     DYNAMIC     Fa0/23
   1    0030.f2dd.2618     DYNAMIC     Fa0/23
   1    0040.0bca.2216     DYNAMIC     Fa0/23
   1    0050.0fa3.529b     DYNAMIC     Fa0/11
   1    0090.0cc9.d7be     DYNAMIC     Fa0/13
   1    0090.2b04.d917     DYNAMIC     Fa0/23
   1    0090.2b04.d918     DYNAMIC     Fa0/24
   1    00d0.bc18.770c     DYNAMIC     Fa0/23
   1    00e0.f9b0.d36a     DYNAMIC     Fa0/23
```

What do you think is occurring with the redundant switch to switch connections? Are they both actively forwarding traffic?
Since we are using a vlan enabled version of spanning tree protocol (gathered from startup-config), the redundant connections are blocked to avoid a loop in the network. However, if one of the cables were to fail, the redundant one would then be enabled.

Conclusion: Everything worked in my lab except enabling logging synchronous on the first 16 lines of the virtual terminal lines because I only configured that for Switch1 and forgot to set up the others.