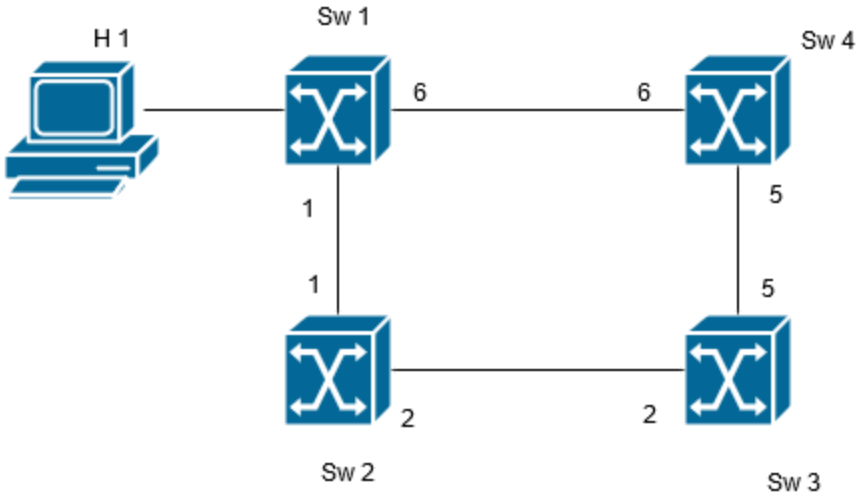
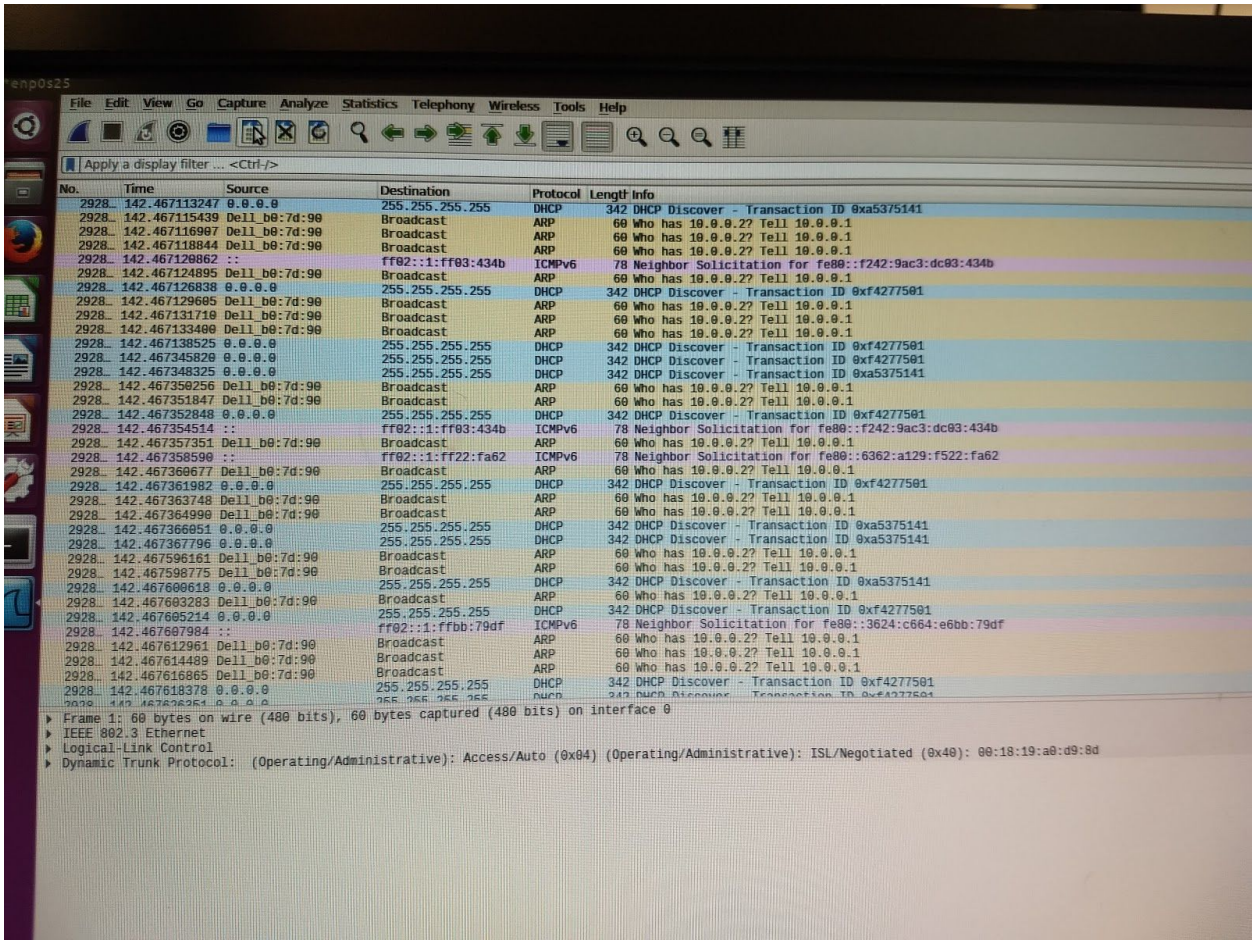
James Cross
Michael Patel
ECE 577
Lab Hw 3

Problem 1

1. **Objective**: To implement a mechanism that breaks loops in an L2 topology
2. **System Under Observation**: L2 topology of 4 switches, PC screen
3. **Input**
   a. **Verification Problem:** To verify that the topology contains a loop, can observe Wireshark capture on PC screen. Filter on ARPs. Once many ARP broadcasts have been captured, alarm network administrator that there is a loop in the topology. Physically locate administrator to resolve issue by breaking link to stop broadcasts.
   b. **Measurement Problem**: To measure how fast the Albert mechanism is at reacting to failures, the ARP broadcasts on the Wireshark capture need to be first noticed. This is dependent on humans, so it could take seconds to possibly hours. We expect it to be on the order of minutes that a human might notice, however it would only take milliseconds for the broadcast to actually happen. Then, the administrator needs to be alerted, which could also take minutes.
4. **Expected Output**:
   a. **Verification Problem:** The expected output to verify that the topology contains a loop is to observe ARP broadcasts in Wireshark capture.
   b. **Measurement Problem**: The expected measurement of how fast the Albert scheme reacts to a failure is approximately 1s for a human to notice, but actually in milliseconds. Then, the time to contact administrator might take seconds if already co-located or minutes.
5. **Actual Output**



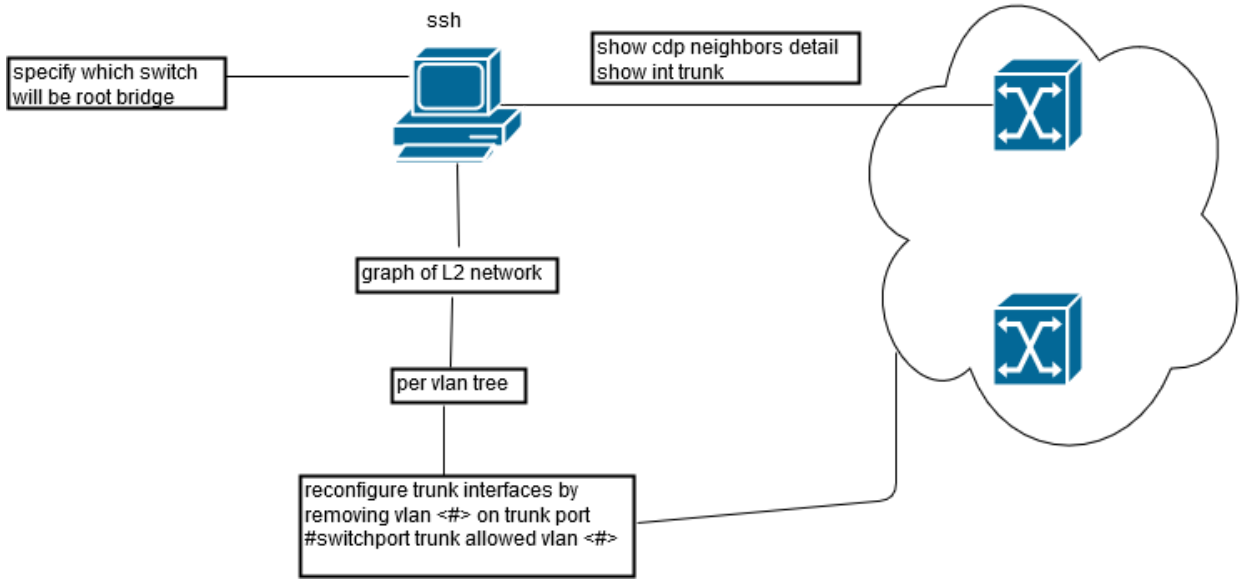


Overall, the Albert scheme is not very effective or scalable. For one, it requires constant human attention in order to quickly find ARP broadcasts and then to alert a network administrator in a timely fashion. Image an enterprise network with lots of LANs and VLANs having to be overseen by human eyes. There are also lapses with regards to mission critical applications where any downtime or inability to be useful is too much inconvenience.

A better implementation scrapes the Wireshark for ARP broadcasts automatically and then alerts an administrator via a messaging system. This allows the individual monitoring the Wireshark to perform more useful and complex tasks such as customer service. An even better approach is to fully automate the system using artificial intelligence mechanisms that can be proactive and reactive without human intervention entirely.

Problem 2

1. **Objective**: To create an automated system that provides a loop-free L2 topology per each vlan in the L2 network
2. **System Under Observation:** The system under observation is each vlan in an L2 network
3. **Input**
   a. **Verification Problem**: The spanning trees for each vlan can be verified by using “show spanning-tree vlan <#>” command on each switch, and using all information to abstract a loop-free per vlan tree
4. **Expected Output**:
   a. **Verification Problem**: The expected output is that each vlan will become loop free via automated script and that after sufficient time, the STP implementations in each switch will mean that the script’s tree and the STP’s tree will converge to the same tree.
5. **Actual Output**



Our implementation for loop-free vlan topologies uses a globalized approach which is effective in small topologies such as in the lab/classroom environment. Additionally, it is nice that one central user can learn many different vlan trees using a single script. Unfortunately, this does not scale when there are hundreds to thousands of switches with many more vlans. The amount of information would become overwhelming to a central user, and may become slow as a result. Instead, STP is decentralized and operates in each switch independently of topology changes. Perhaps by converting the network system into an SDN system, we can leverage the global nature of our implementation with statistics feedback without the inability to scale effectively. Even more so, the central user must be prompted as to which switch will serve as root bridge

per vlan. Again, this is ineffective with many switches and vlans in the topology. Some STP features we would incorporate would be timers, so that actions can happen without the need for a central human user. STP is proactive in that loops are never allowed at any time. However, our implementation may contain loops, detects them, and then reacts by re-configuring trunk vlan access and spanning-tree primary designation. This is quite slow and limited by dynamic topologies.

See readme for instructions and output details. Below is a screen capture for breaking a loop in a vlan 10 topology

```
##############################

A3>terminal length 0
A3>enable
Password:
A3#show int trunk

Port        Mode             Encapsulation  Status        Native vlan
Fa0/12      on               802.1q         trunking      1
Fa0/13      on               802.1q         trunking      1

Port        Vlans allowed on trunk
Fa0/12      1,10,20
Fa0/13      1,10,20

Port        Vlans allowed and active in management domain
Fa0/12      1
Fa0/13      1

Port        Vlans in spanning tree forwarding state and not pruned
Fa0/12      1
Fa0/13      1
A3#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
A3(config)#interface FastEthernet0/13
A3(config-if)#switchport trunk allowed vlan 1,20
A3(config-if)#end
A3#show int trunk

Port        Mode             Encapsulation  Status        Native vlan
Fa0/12      on               802.1q         trunking      1
Fa0/13      on               802.1q         trunking      1

Port        Vlans allowed on trunk
Fa0/12      1,10,20
Fa0/13      1,20

Port        Vlans allowed and active in management domain
Fa0/12      1
Fa0/13      1

Port        Vlans in spanning tree forwarding state and not pruned
Fa0/12      1
Fa0/13      1
A3#

##############################
```

```
##############################

D1>terminal length 0
D1>enable
Password:
D1#show int trunk

Port        Mode             Encapsulation  Status        Native vlan
Fa0/1       on               802.1q         trunking      1
Fa0/2       on               802.1q         trunking      1

Port        Vlans allowed on trunk
Fa0/1       1,10,20
Fa0/2       1,10,20

Port        Vlans allowed and active in management domain
Fa0/1       1
Fa0/2       1

Port        Vlans in spanning tree forwarding state and not pruned
Fa0/1       1
Fa0/2       1
D1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
D1(config)#interface FastEthernet0/1
D1(config-if)#switchport trunk allowed vlan 1,20
D1(config-if)#end
D1#show int trunk

Port        Mode             Encapsulation  Status        Native vlan
Fa0/1       on               802.1q         trunking      1
Fa0/2       on               802.1q         trunking      1

Port        Vlans allowed on trunk
Fa0/1       1,20
Fa0/2       1,10,20

Port        Vlans allowed and active in management domain
Fa0/1       1
Fa0/2       1

Port        Vlans in spanning tree forwarding state and not pruned
Fa0/1       1
Fa0/2       1
D1#
PS C:\Users\micha\PycharmProjects\ECE 577\Lab Hw3>
```

```
#################################################################

#################################################################

List of nodes by IP address:  ['192.168.1.10', '192.168.1.20', '192.168.1.30']

For vlan 1, specify root node by IP address: 192.168.1.30

Edges:  [('192.168.1.30', '192.168.1.20'), ('192.168.1.30', '192.168.1.10'), ('192.168.1.10', '192.168.1.20'), ('192.168.1.10', '192.1
68.1.30'), ('192.168.1.20', '192.168.1.30'), ('192.168.1.20', '192.168.1.10')]

Tree: (start_node, dest_node)  [('192.168.1.30', '192.168.1.10'), ('192.168.1.30', '192.168.1.20')]

BEFORE CONFIG

middle_switch>terminal length 0
middle_switch>en
Password:
middle_switch#show spanning-tree vlan 1

VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    24577
             Address     0015.6285.aa80
             Cost        19
             Port        8 (FastEthernet0/6)
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    24577  (priority 24576 sys-id-ext 1)
             Address     001a.e37d.7b00
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Fa0/5               Desg FWD 19        128.7    P2p
Fa0/6               Root FWD 19        128.8    P2p


middle_switch#

####################

AFTER CONFIG
conf t
Enter configuration commands, one per line.  End with CNTL/Z.
middle_switch(config)#spanning-tree vlan 1 root primary
middle_switch(config)#end
middle_switch#show spanning-tree vlan 1

VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    20481
             Address     001a.e37d.7b00
             This bridge is the root
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    20481  (priority 20480 sys-id-ext 1)
             Address     001a.e37d.7b00
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  15  sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Fa0/5               Desg FWD 19        128.7    P2p
Fa0/6               Desg FWD 19        128.8    P2p


middle_switch#
```

```
################################################################

################################################################

List of nodes by IP address:  ['192.168.1.10', '192.168.1.20', '192.168.1.30']

For vlan 10, specify root node by IP address: 192.168.1.10

Edges:  [('192.168.1.10', '192.168.1.20'), ('192.168.1.10', '192.168.1.30'), ('192.168.1.20', '192.168.1.30'), ('192.168.1.20', '192.1
68.1.10'), ('192.168.1.30', '192.168.1.20'), ('192.168.1.30', '192.168.1.10')]

Tree: (start_node, dest_node)  [('192.168.1.10', '192.168.1.20'), ('192.168.1.10', '192.168.1.30')]

BEFORE CONFIG

TOP>terminal length 0
TOP>en
Password:
TOP#show spanning-tree vlan 10

VLAN0010
  Spanning tree enabled protocol ieee
  Root ID    Priority    20490
             Address     0015.6286.6e00
             This bridge is the root
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    20490  (priority 20480 sys-id-ext 10)
             Address     0015.6286.6e00
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time 300

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Fa0/1               Desg FWD 100       128.3    P2p
Fa0/2               Desg FWD 100       128.4    P2p
Fa0/5               Desg FWD 19        128.7    P2p

TOP#

####################

AFTER CONFIG
conf t
Enter configuration commands, one per line.  End with CNTL/Z.
TOP(config)#spanning-tree vlan 10 root primary
TOP(config)#end
TOP#show spanning-tree vlan 10

VLAN0010
  Spanning tree enabled protocol ieee
  Root ID    Priority    20490
             Address     0015.6286.6e00
             This bridge is the root
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    20490  (priority 20480 sys-id-ext 10)
             Address     0015.6286.6e00
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time 300

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Fa0/1               Desg FWD 100       128.3    P2p
Fa0/2               Desg FWD 100       128.4    P2p
Fa0/5               Desg FWD 19        128.7    P2p

TOP#

################################################################
```

```
#################################################################

#################################################################

#################################################################

List of nodes by IP address:  ['192.168.1.10', '192.168.1.20', '192.168.1.30']

For vlan 20, specify root node by IP address: 192.168.1.20

Edges:  [('192.168.1.20', '192.168.1.30'), ('192.168.1.20', '192.168.1.10'), ('192.168.1.10', '192.168.1.20'), ('192.168.1.10', '192.1
68.1.30'), ('192.168.1.30', '192.168.1.20'), ('192.168.1.30', '192.168.1.10')]

Tree: (start_node, dest_node)  [('192.168.1.20', '192.168.1.10'), ('192.168.1.20', '192.168.1.30')]

BEFORE CONFIG

Bottom>terminal length 0
Bottom>en
Password:
Bottom#show spanning-tree vlan 20

VLAN0020
  Spanning tree enabled protocol ieee
  Root ID    Priority    20500
             Address     0015.6286.6e00
             Cost        38
             Port        8 (FastEthernet0/6)
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    24596  (priority 24576 sys-id-ext 20)
             Address     0015.6285.aa80
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time 300

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Fa0/1               Altn BLK 100       128.3    P2p
Fa0/6               Root FWD 19        128.8    P2p

Bottom#

####################

AFTER CONFIG
conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Bottom(config)#spanning-tree vlan 20 root primary
Bottom(config)#end
Bottom#show spanning-tree vlan 20

VLAN0020
  Spanning tree enabled protocol ieee
  Root ID    Priority    20500
             Address     0015.6285.aa80
             This bridge is the root
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    20500  (priority 20480 sys-id-ext 20)
             Address     0015.6285.aa80
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time 15

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Fa0/1               Desg LIS 100       128.3    P2p
Fa0/6               Desg FWD 19        128.8    P2p

Bottom#
PS C:\Users\micha\PycharmProjects\ECE 577\Lab Hw3>
```

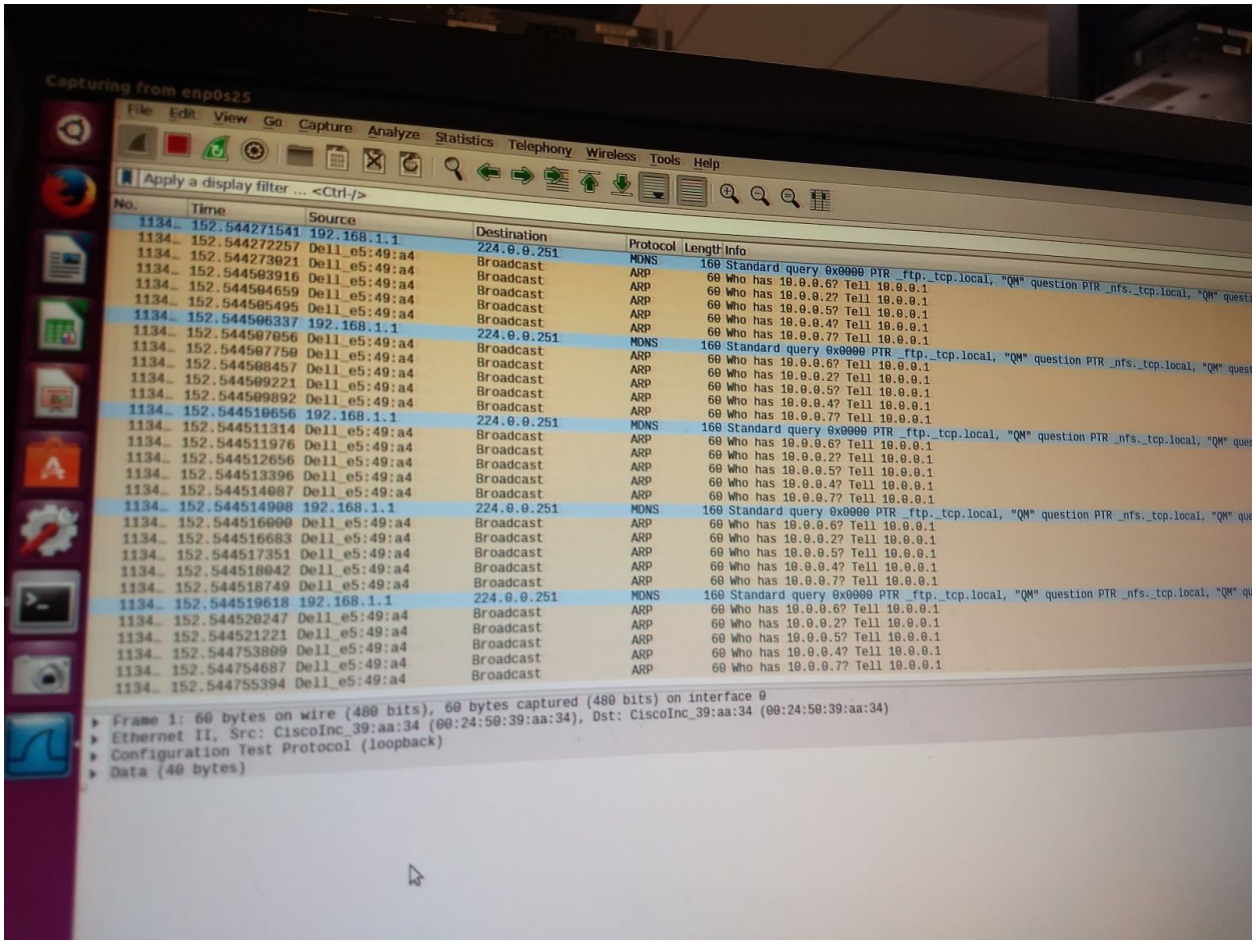
```
#################################################################

#################################################################

#################################################################

List of nodes by IP address:  ['192.168.1.10', '192.168.1.20', '192.168.1.30']

For vlan 20, specify root node by IP address: 192.168.1.20

Edges:  [('192.168.1.20', '192.168.1.30'), ('192.168.1.20', '192.168.1.10'), ('192.168.1.10', '192.168.1.20'), ('192.168.1.10', '192.1
68.1.30'), ('192.168.1.30', '192.168.1.20'), ('192.168.1.30', '192.168.1.10')]

Tree: (start_node, dest_node)  [('192.168.1.20', '192.168.1.10'), ('192.168.1.20', '192.168.1.30')]

BEFORE CONFIG

Bottom>terminal length 0
Bottom>en
Password:
Bottom#show spanning-tree vlan 20

VLAN0020
  Spanning tree enabled protocol ieee
  Root ID    Priority    20500
             Address     0015.6286.6e00
             Cost        38
             Port        8 (FastEthernet0/6)
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    24596  (priority 24576 sys-id-ext 20)
             Address     0015.6285.aa80
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time 300

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Fa0/1               Altn BLK 100       128.3    P2p
Fa0/6               Root FWD 19        128.8    P2p

Bottom#

####################

AFTER CONFIG
conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Bottom(config)#spanning-tree vlan 20 root primary
Bottom(config)#end
Bottom#show spanning-tree vlan 20

VLAN0020
  Spanning tree enabled protocol ieee
  Root ID    Priority    20500
             Address     0015.6285.aa80
             This bridge is the root
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    20500  (priority 20480 sys-id-ext 20)
             Address     0015.6285.aa80
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time 15

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Fa0/1               Desg LIS 100       128.3    P2p
Fa0/6               Desg FWD 19        128.8    P2p

Bottom#
PS C:\Users\micha\PycharmProjects\ECE 577\Lab Hw3>
```

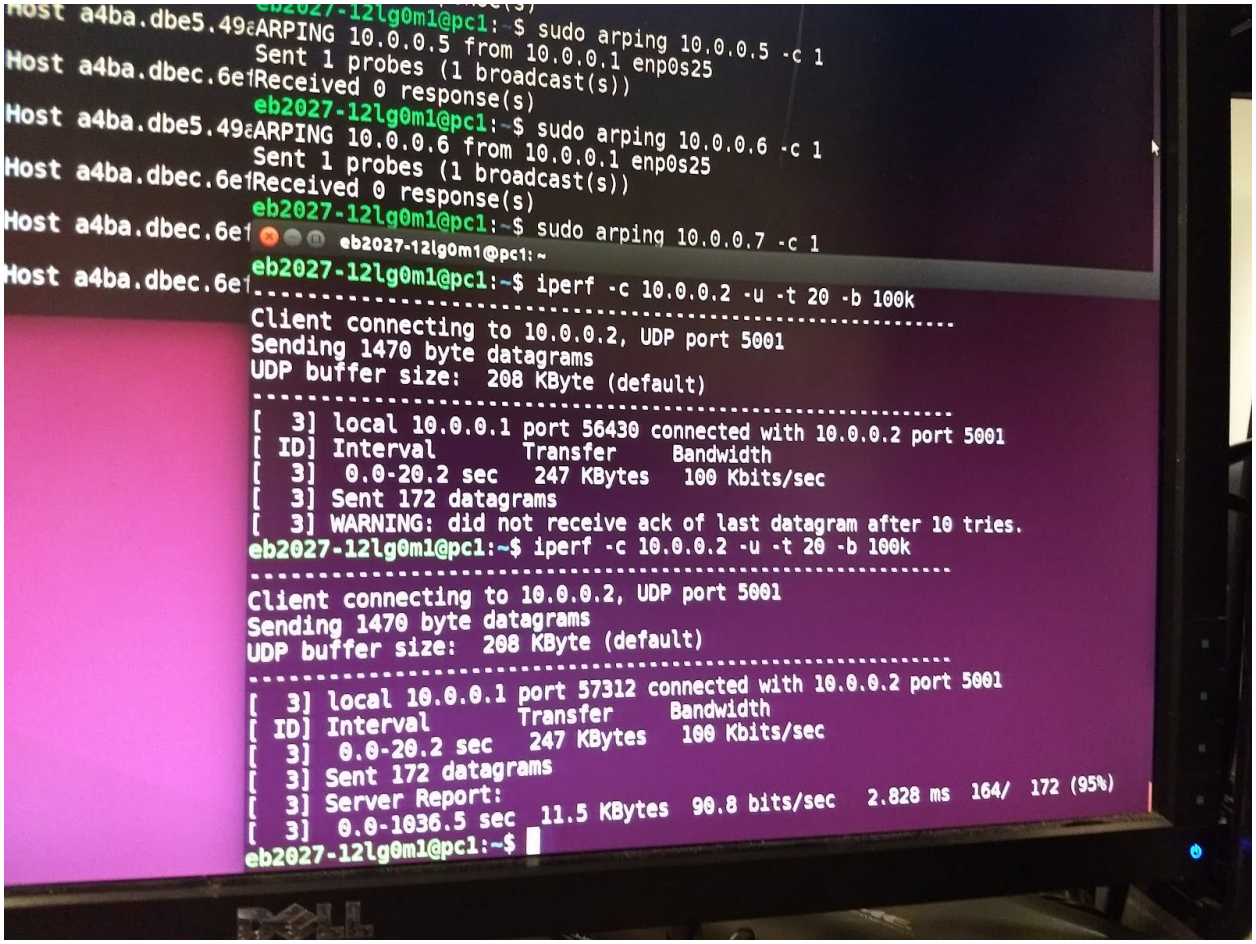
Problem 3

1. **Objective**: To investigate the consequences of broadcasts in a loop topology

2. **System Under Observation**: L2 network of 3 switches that has a loop, the bandwidth of a particular link between the switches
3. **Input**
    a. **Measurement Problem**:
        i. 3.2 Unusable will mean that the number of iperf udp misses will be at or near 100% loss. To measure, an iperf udp client and server will be set up on 2 hosts. Then, incrementally, ARP broadcast frames will be introduced until the network becomes unusable.
        ii. 3.3 To measure the bandwidth consumed by a single ARP broadcast frame, we timed 10s worth of broadcasting over 100 Mbps links using Wireshark captures.
        iii. 3.4 We can calculate the number of max ARP frames before the network becomes unusable as the ARP frame size is 60 bytes. A max frame size for Ethernet is 1518 bytes. Therefore, we can use a frame length of 1518 bytes instead of 60 and calculate the number of frames accordingly.
        iv. 3.5 The command “show proc cpu history” provides cpu usage and “show buffers” provides buffer information
4. **Expected Output**:
    a. **Measurement Problem**:
        i. 3.2 The expected output will probably be 3-5 ARP broadcast frames given ARP frame size and link bandwidth.
        ii. 3.3 The expected output will probably be between 25-33% according to previous homework calculations given by estimating the number of ARP frames in a 10s interval. Then, using link bandwidth and ARP broadcast frame length, we can expect a percentage due to that single frame between 25 and 33%.
        iii. 3.4 The expected output will be less than 4 ARP frames. The max ARP frame size is 60 bytes which is what can be seen by L2 network.
5. **Actual Output**

Q3.2 - 4 ARP broadcast frames


Capturing from enp0s25
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
No. Time Source Destination Protocol Length Info
1134… 152.544271541 192.168.1.1 224.0.0.251 MDNS 160 Standard query 0x0000 PTR _ftp._tcp.local, "QM" question PTR _nfs._tcp.local, "QM" questi
1134… 152.544272257 Dell_e5:49:a4 Broadcast ARP 60 Who has 10.0.0.6? Tell 10.0.0.1
1134… 152.544273021 Dell_e5:49:a4 Broadcast ARP 60 Who has 10.0.0.2? Tell 10.0.0.1
1134… 152.544503916 Dell_e5:49:a4 Broadcast ARP 60 Who has 10.0.0.5? Tell 10.0.0.1
1134… 152.544504659 Dell_e5:49:a4 Broadcast ARP 60 Who has 10.0.0.4? Tell 10.0.0.1
1134… 152.544505495 Dell_e5:49:a4 Broadcast ARP 60 Who has 10.0.0.7? Tell 10.0.0.1
1134… 152.544506337 192.168.1.1 224.0.0.251 MDNS 160 Standard query 0x0000 PTR _ftp._tcp.local, "QM" question PTR _nfs._tcp.local, "QM" quest
1134… 152.544507056 Dell_e5:49:a4 Broadcast ARP 60 Who has 10.0.0.6? Tell 10.0.0.1
1134… 152.544507750 Dell_e5:49:a4 Broadcast ARP 60 Who has 10.0.0.2? Tell 10.0.0.1
1134… 152.544508457 Dell_e5:49:a4 Broadcast ARP 60 Who has 10.0.0.5? Tell 10.0.0.1
1134… 152.544509221 Dell_e5:49:a4 Broadcast ARP 60 Who has 10.0.0.4? Tell 10.0.0.1
1134… 152.544509892 Dell_e5:49:a4 Broadcast ARP 60 Who has 10.0.0.7? Tell 10.0.0.1
1134… 152.544510656 192.168.1.1 224.0.0.251 MDNS 160 Standard query 0x0000 PTR _ftp._tcp.local, "QM" question PTR _nfs._tcp.local, "QM" ques
1134… 152.544511314 Dell_e5:49:a4 Broadcast ARP 60 Who has 10.0.0.6? Tell 10.0.0.1
1134… 152.544511976 Dell_e5:49:a4 Broadcast ARP 60 Who has 10.0.0.2? Tell 10.0.0.1
1134… 152.544512656 Dell_e5:49:a4 Broadcast ARP 60 Who has 10.0.0.5? Tell 10.0.0.1
1134… 152.544513396 Dell_e5:49:a4 Broadcast ARP 60 Who has 10.0.0.4? Tell 10.0.0.1
1134… 152.544514087 Dell_e5:49:a4 Broadcast ARP 60 Who has 10.0.0.7? Tell 10.0.0.1
1134… 152.544514908 192.168.1.1 224.0.0.251 MDNS 160 Standard query 0x0000 PTR _ftp._tcp.local, "QM" question PTR _nfs._tcp.local, "QM" que
1134… 152.544516000 Dell_e5:49:a4 Broadcast ARP 60 Who has 10.0.0.6? Tell 10.0.0.1
1134… 152.544516683 Dell_e5:49:a4 Broadcast ARP 60 Who has 10.0.0.2? Tell 10.0.0.1
1134… 152.544517351 Dell_e5:49:a4 Broadcast ARP 60 Who has 10.0.0.5? Tell 10.0.0.1
1134… 152.544518042 Dell_e5:49:a4 Broadcast ARP 60 Who has 10.0.0.4? Tell 10.0.0.1
1134… 152.544518749 Dell_e5:49:a4 Broadcast ARP 60 Who has 10.0.0.7? Tell 10.0.0.1
1134… 152.544519618 192.168.1.1 224.0.0.251 MDNS 160 Standard query 0x0000 PTR _ftp._tcp.local, "QM" question PTR _nfs._tcp.local, "QM" qu
1134… 152.544520247 Dell_e5:49:a4 Broadcast ARP 60 Who has 10.0.0.6? Tell 10.0.0.1
1134… 152.544521221 Dell_e5:49:a4 Broadcast ARP 60 Who has 10.0.0.2? Tell 10.0.0.1
1134… 152.544753809 Dell_e5:49:a4 Broadcast ARP 60 Who has 10.0.0.5? Tell 10.0.0.1
1134… 152.544754687 Dell_e5:49:a4 Broadcast ARP 60 Who has 10.0.0.4? Tell 10.0.0.1
1134… 152.544755394 Dell_e5:49:a4 Broadcast ARP 60 Who has 10.0.0.7? Tell 10.0.0.1
Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
Ethernet II, Src: CiscoInc_39:aa:34 (00:24:50:39:aa:34), Dst: CiscoInc_39:aa:34 (00:24:50:39:aa:34)
Configuration Test Protocol (loopback)
Data (40 bytes)



Q3.3

arp bcast frame length = 60 Bytes = 240 bits
link bandwidth = 100 Mbps
time = 10s
1 arp bcast frame

ARP frames = 565727

(565727 frames)(240 bits/frame)/(10s) = 27154896 bps

27.15%

Q3.4 - max ARP frame size is 60 byte, Ethernet can support 1518 byte frames, so 4 frames will completely make the network unusable

Q3.5

Buffer occupancy and MAC address flapping

Terminal

eb2027-12lg0m1@pc1: ~

```
 flapping between port Fa0/3 and port Fa0/2
*Mar  1 16:45:27.352: %SW_MATM-4-MACFLAP_NOTIF: Host a4ba.dbe5.49a4 in vlan 1 is
 flapping between port Fa0/2 and port Fa0/3
Top_Switch#sh
*Mar  1 16:45:28.451: %SYS-5-CONFIG_I: Configured from console by consoleow buff
ers
Buffer elements:
     1506 in free list (500 max allowed)
     36482 hits, 0 misses, 1024 created

Public buffer pools:
Small buffers, 104 bytes (total 50, permanent 50, peak 185 @ 16:44:55):
     49 in free list (20 min, 150 max allowed)
     28057 hits, 45 misses, 135 trims, 135 created
     0 failures (0 no memory)
Middle buffers, 600 bytes (total 25, permanent 25, peak 67 @ 16:44:56):
     25 in free list (10 min, 150 max allowed)
     1003 hits, 14 misses, 42 trims, 42 created
     0 failures (0 no memory)
Big buffers, 1536 bytes (total 50, permanent 50, peak 83 @ 16:44:32):
     24 in free list (5 min, 150 max allowed)
     315799 hits, 13 misses, 39 trims, 39 created
     0 failures (0 no memory)
VeryBig buffers, 4520 bytes (total 14, permanent 10, peak 14 @ 16:44:59):
```

MSK_VC

```
Sending 1470 byte d
UDP buffer size:  2
------------------
[  3] local 10.0.0.
[ ID] Interval
[  3]  0.0-20.2 sec
[  3] Sent 172 datag
```

Terminal

eb2027-12lg0m1@pc1: ~

```
     315799 hits, 13 misses, 39 trims, 39 created
     0 failures (0 no memory)
VeryBig buffers, 4520 bytes (total 14, permanent 10, peak 14 @ 16:44:59):
     0 in free list (0 min, 100 max allowed)
     88 hits, 2 misses, 0 trims, 4 created
     0 failures (0 no memory)
Large buffers, 5024 bytes (total 0, permanent 0):
     0 in free list (0 min, 10 max allowed)
     0 hits, 0 misses, 0 trims, 0 created
     0 failures (0 no memory)
Huge buffers, 18024 bytes (total 0, permanent 0):
     0 in free list (0 min, 4 max allowed)
     0 hits, 0 misses, 0 trims, 0 created
     0 failures (0 no memory)

Interface buffer pools:
Syslog ED Pool buffers, 600 bytes (total 132, permanent 132):
     100 in free list (132 min, 132 max allowed)
     949 hits, 0 misses
RxQFB buffers, 2040 bytes (total 300, permanent 300):
     240 in free list (0 min, 300 max allowed)
     11130297 hits, 16378 misses
RxQ1 buffers, 2040 bytes (total 128, permanent 128):
     2 in free list (0 min, 128 max allowed)
```

MSK_VC

```
Sending 1470 byte
UDP buffer size:
[  3] local 10.0.0
```

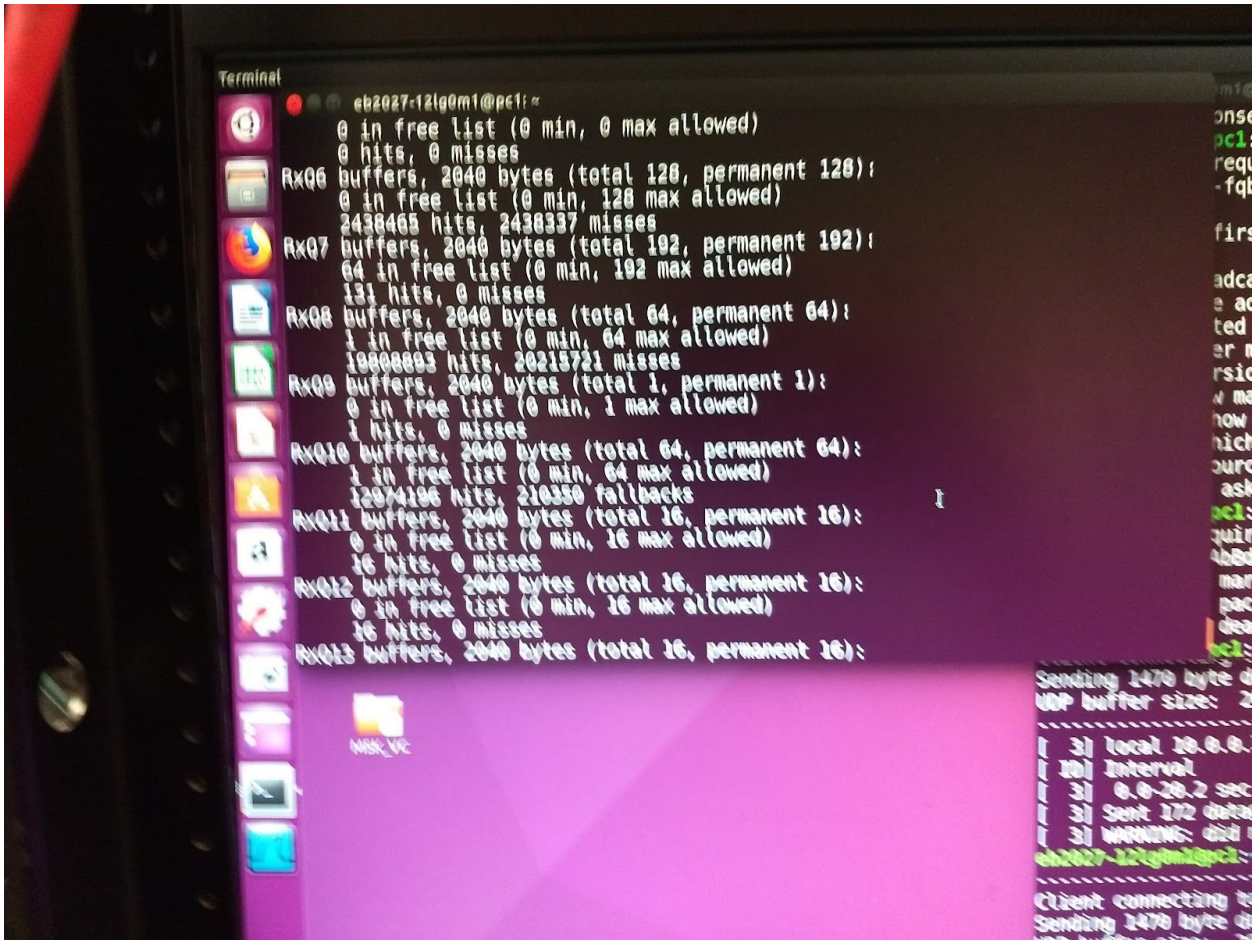
Terminal

eb2027-12ig0m1@pc1: ~

```
Interface buffer pools:
Syslog ED Pool buffers, 600 bytes (total 132, permanent 132):
     100 in free list (132 min, 132 max allowed)
     949 hits, 0 misses
RxQFB buffers, 2040 bytes (total 300, permanent 300):
     240 in free list (0 min, 300 max allowed)
     11130297 hits, 16378 misses
RxQ1 buffers, 2040 bytes (total 128, permanent 128):
     2 in free list (0 min, 128 max allowed)
     207 hits, 2 fallbacks
RxQ2 buffers, 2040 bytes (total 12, permanent 12):
     0 in free list (0 min, 12 max allowed)
     12 hits, 0 fallbacks, 0 trims, 0 created
     0 failures (0 no memory)
RxQ3 buffers, 2040 bytes (total 128, permanent 128):
     0 in free list (0 min, 128 max allowed)
     60314132 hits, 10936349 fallbacks
RxQ4 buffers, 2040 bytes (total 64, permanent 64):
     0 in free list (0 min, 64 max allowed)
     4016 hits, 9444 misses
RxQ5 buffers, 2040 bytes (total 0, permanent 0):
     0 in free list (0 min, 0 max allowed)
     0 hits, 0 misses
RxQ6 buffers, 2040 bytes (total 128, permanent 128):
```

MSK_VC

```
Sending 1470 byte data
UDP buffer size:  208
[  3] local 10.0.0.1 p
[ ID] Interval       T
[  3]  0.0-20.2 sec
```


Terminal
eb2027-12ig0m1@pc1: ~
0 in free list (0 min, 0 max allowed)
0 hits, 0 misses
RxQ6 buffers, 2040 bytes (total 128, permanent 128):
0 in free list (0 min, 128 max allowed)
2438465 hits, 2438337 misses
RxQ7 buffers, 2040 bytes (total 192, permanent 192):
64 in free list (0 min, 192 max allowed)
131 hits, 0 misses
RxQ8 buffers, 2040 bytes (total 64, permanent 64):
1 in free list (0 min, 64 max allowed)
19808893 hits, 20215721 misses
RxQ9 buffers, 2040 bytes (total 1, permanent 1):
0 in free list (0 min, 1 max allowed)
1 hits, 0 misses
RxQ10 buffers, 2040 bytes (total 64, permanent 64):
1 in free list (0 min, 64 max allowed)
12974196 hits, 210350 fallbacks
RxQ11 buffers, 2040 bytes (total 16, permanent 16):
0 in free list (0 min, 16 max allowed)
16 hits, 0 misses
RxQ12 buffers, 2040 bytes (total 16, permanent 16):
0 in free list (0 min, 16 max allowed)
16 hits, 0 misses
RxQ13 buffers, 2040 bytes (total 16, permanent 16):
MSK_VC
Sending 1470 byte
UDP buffer size:
[3] local 10.0.0.
[ID] Interval
[3] 0.0-20.2 sec
[3] Sent 172 data
[3] WARNING: did
eb2027-12ig0m1@pc1:
Client connecting
Sending 1470 byte

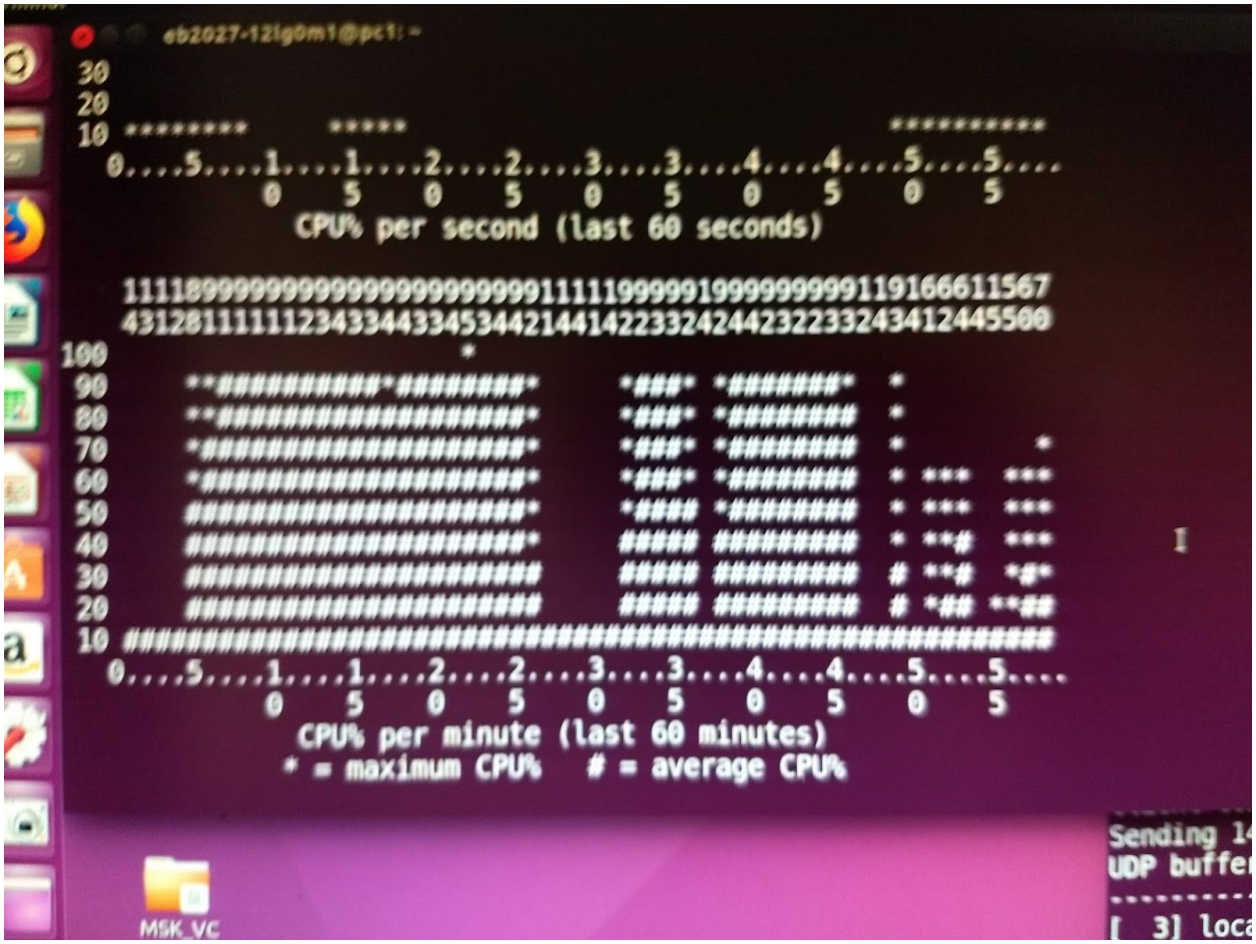
CPU usage

```
eb2027-12lg0m1@pc1: ~
          0    5    0    5    0    5    0    5    0    5    0    5
           CPU% per hour (last 72 hours)
          * = maximum CPU%   # = average CPU%

Top_Switch#show proc cpu history

                                                          11111
    55555555444445555544444444444444444444444444444444444466666
100
 90
 80
 70
 60
 50
 40
 30
 20
 10 ********     *****                                 **********
   0....5....1....1....2....2....3....3....4....4....5....5....
            0    5    0    5    0    5    0    5    0    5
             CPU% per second (last 60 seconds)

    111189999999999999999999991111199999199999999911919166611567
    431281111111234334433453442144142233242442322332434124455 00
```

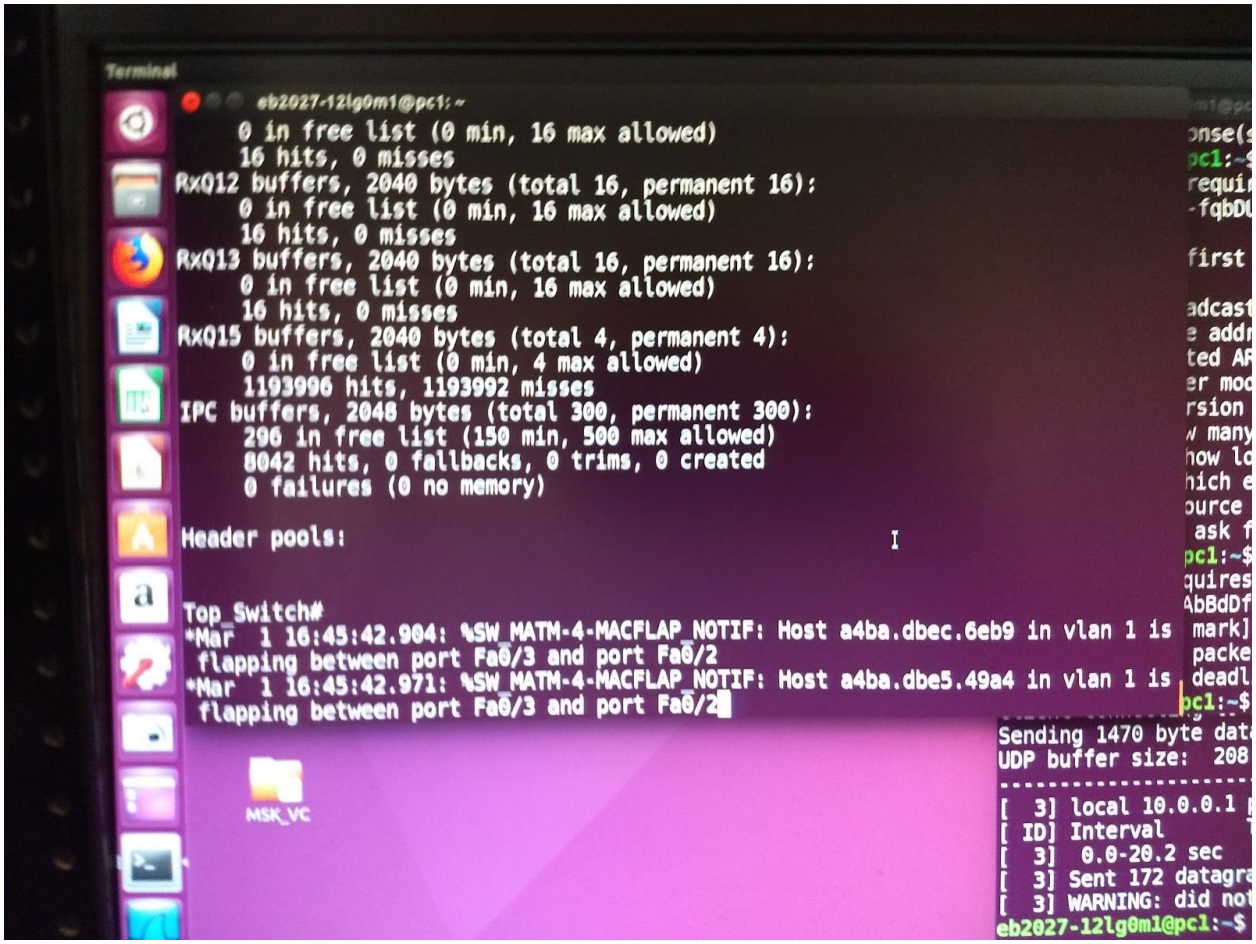
```
10 ##################################################################
   0....5....1....1....2....2....3....3....4....4....5....5....
             0    5    0    5    0    5    0    5    0    5
              CPU% per minute (last 60 minutes)
             * = maximum CPU%   # = average CPU%

    9111111111119999
    3545555455556646
100                 ** *
 90 *               ****
 80 *               ****
 70 *               ****
 60 *               **##
 50 *               **##
 40 *               **##
 30 *               **##
 20 ** **** ****####
 10 ################
   0....5....1....1....2....2....3....3....4....4....5....5....6....6
             0    5    0    5    0    5    0    5    0    5    0    5
                 CPU% per hour (last 72 hours)
                * = maximum CPU%   # = average CPU%

Top_Switch#
```

Send
UDP

[3



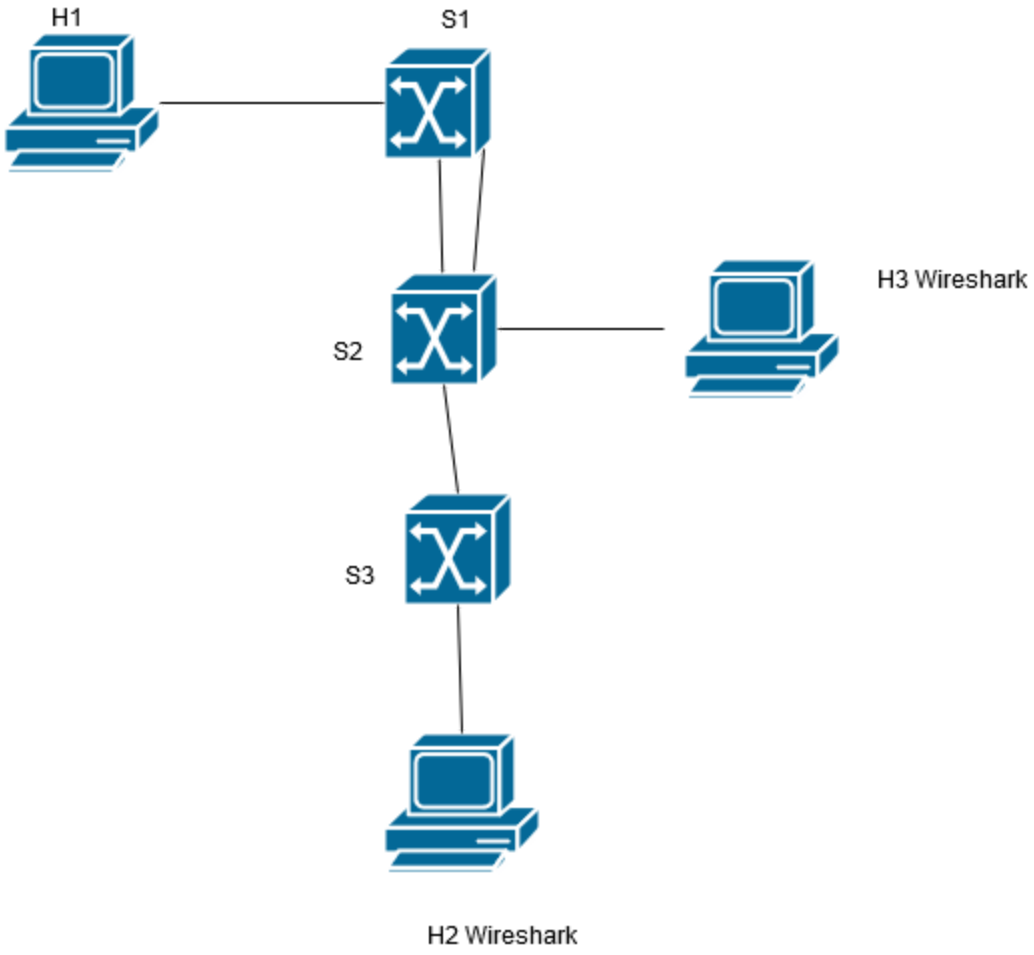
MAC address flapping



Problem 4

1. **Objective**: To understand the benefits and operations of EtherChannel
2. **System Under Observation**: the aggregated EtherChannel link composed of several physical links
3. **Input**
   a. **Verification Problem**: To verify that EtherChannel group still functions with a single link failure, ping connectivity from one host (H1) to two different hosts (H2, H3) over the EtherChannel. The default link bandwidth is 100 Mbps for the pings from H1. The EtherChannel is configured as 10 Mbps links.
4. **Expected Output**:
   a. **Verification Problem**: The expected output with a link failure is that the EtherChannel group will still function as the other links in the group handle the data traffic. The traffic is redistributed onto the other remaining up link in the group, while still being received on hosts H2 and H3.

5. Actual Output



The topology is appropriate to study EtherChannel because there are 2 links between switches S1 and S2 that will be aggregated to behave as a single link with twice the bandwidth. The principal advantage of EtherChannel is to bundle several links as one in an effort to gain more bandwidth without buying higher-bandwidth single cables as substitutes. Additionally, the EtherChannel provides some degree of redundancy and load balancing.

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

No.	Time	Source	Destination	Protocol	Length	ID	Info
1355	854.860	10.0.0.3	10.0.0.1	ICMP	98		Echo (ping) reply id=0x6609, seq=
1356	855.647	CiscoInc_85:aa:8e	CDP/VTP/DTP/PAgP/UDLD	DTP	60		Dynamic Trunk Protocol
1357	855.647	CiscoInc_85:aa:8e	CDP/VTP/DTP/PAgP/UDLD	DTP	90		Dynamic Trunk Protocol
1358	855.860	10.0.0.1	10.0.0.3	ICMP	98		Echo (ping) request id=0x6609, seq
1359	855.860	10.0.0.3	10.0.0.1	ICMP	98		Echo (ping) reply id=0x6609, seq
1360	856.860	10.0.0.1	10.0.0.3	ICMP	98		Echo (ping) request id=0x6609, seq
1361	856.860	10.0.0.3	10.0.0.1	ICMP	98		Echo (ping) reply id=0x6609, seq
1362	857.860	10.0.0.1	10.0.0.3	ICMP	98		Echo (ping) request id=0x6609, seq
1363	857.860	10.0.0.3	10.0.0.1	ICMP	98		Echo (ping) reply id=0x6609, seq
1364	858.860	10.0.0.1	10.0.0.3	ICMP	98		Echo (ping) request id=0x6609, seq
1365	858.860	10.0.0.3	10.0.0.1	ICMP	98		Echo (ping) reply id=0x6609, seq
1366	859.860	10.0.0.1	10.0.0.3	ICMP	98		Echo (ping) request id=0x6609, seq
1367	859.860	10.0.0.3	10.0.0.1	ICMP	98		Echo (ping) reply id=0x6609, seq
1368	860.454	CiscoInc_85:aa:8e	CiscoInc_85:aa:8e	LOOP	60		Reply
1369	860.860	10.0.0.1	10.0.0.3	ICMP	98		Echo (ping) request id=0x6609, seq
1370	860.860	10.0.0.3	10.0.0.1	ICMP	98		Echo (ping) reply id=0x6609, seq
1371	861.860	10.0.0.1	10.0.0.3	ICMP	98		Echo (ping) request id=0x6609, seq
1372	861.860	10.0.0.3	10.0.0.1	ICMP	98		Echo (ping) reply id=0x6609, seq
1373	861.939	CiscoInc_85:aa:8e	CDP/VTP/DTP/PAgP/UDLD	CDP	400		Device ID: Bottom_Switch Port ID:
1374	862.860	10.0.0.1	10.0.0.3	ICMP	98		Echo (ping) request id=0x6609, se
1375	862.860	10.0.0.3	10.0.0.1	ICMP	98		Echo (ping) reply id=0x6609, se
1376	863.860	10.0.0.1	10.0.0.3	ICMP	98		Echo (ping) request id=0x6609, se
1377	863.860	10.0.0.3	10.0.0.1	ICMP	98		Echo (ping) reply id=0x6609, se
1378	864.860	10.0.0.1	10.0.0.3	ICMP	98		Echo (ping) request id=0x6609, se
1379	864.860	10.0.0.3	10.0.0.1	ICMP	98		Echo (ping) reply id=0x6609, se
1380	865.860	10.0.0.1	10.0.0.3	ICMP	98		Echo (ping) request id=0x6609, se
1381	865.860	10.0.0.3	10.0.0.1	ICMP	98		Echo (ping) reply id=0x6609, se
1382	866.860	10.0.0.1	10.0.0.3	ICMP	98		Echo (ping) request id=0x6609, se
1383	866.860	10.0.0.3	10.0.0.1	ICMP	98		Echo (ping) reply id=0x6609, se
1384	867.860	10.0.0.1	10.0.0.3	ICMP	98		Echo (ping) request id=0x6609, se
1385	867.860	10.0.0.3	10.0.0.1	ICMP	98		Echo (ping) reply id=0x6609, se
1386	868.661	10.0.0.2	224.0.0.251	MDNS	183		Standard query 0x0000 PTR _nfs._tc
1387	868.860	10.0.0.1	10.0.0.3	ICMP	98		Echo (ping) request id=0x6609, se
1388	868.860	10.0.0.3	10.0.0.1	ICMP	98		Echo (ping) reply id=0x6609, se
1389	869.860	10.0.0.1	10.0.0.3	ICMP	98		Echo (ping) request id=0x6609, se
1390	869.860	10.0.0.3	10.0.0.1	ICMP	98		Echo (ping) reply id=0x6609, se
1391	870.465	CiscoInc_85:aa:8e	CiscoInc_85:aa:8e	LOOP	60		Reply
1392	870.860	10.0.0.1	10.0.0.3	ICMP	98		Echo (ping) request id=0x6609, se
1393	870.860	10.0.0.3	10.0.0.1	ICMP	98		Echo (ping) reply id=0x6609, se

```
▸ Frame 485: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
▸ Ethernet II, Src: Dell_b0:7d:90 (84:2b:2b:b0:7d:90), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▸ Address Resolution Protocol (request)

0000  ff ff ff ff ff ff 84 2b  2b b0 7d 90 08 06 00 01   .......+ +.}.....
0010  08 00 06 04 00 01 84 2b  2b b0 7d 90 0a 00 00 01   .......+ +.}.....
0020  00 00 00 00 00 00 0a 00  00 02 00 00 00 00 00 00   ........ ........
0030  00 00 00 00 00 00 00 00  00 00 00 00               ........ ....
```

wireshark_pcapng_enp0s25_20181106224003_z3491D

Packets: 1393 · Displayed: 1393 (100.0%

Capturing from enp0s25

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1497	1244.3962490…	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x6608, seq=187/47872, ttl=64 (request in 1496)
1498	1244.7293421…	CiscoInc_a0:d9:8d	CDP/VTP/DTP/PAgP/UD…	CDP	403	Device ID: Middle_Switch.random.com Port ID: FastEthernet0/11
1499	1245.3962213…	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x6608, seq=188/48128, ttl=64 (reply in 1500)
1500	1245.3962470…	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x6608, seq=188/48128, ttl=64 (request in 1499)
1501	1245.4800093…	10.0.0.2	224.0.0.251	MDNS	183	Standard query 0x0000 PTR _nfs._tcp.local, "QM" question PTR _ipp._tcp.local, "QM" question PTR _ipps
1502	1246.3961882…	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x6608, seq=189/48384, ttl=64 (reply in 1503)
1503	1246.3962132…	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x6608, seq=189/48384, ttl=64 (request in 1502)
1504	1247.3961406…	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x6608, seq=190/48640, ttl=64 (reply in 1505)
1505	1247.3961655…	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x6608, seq=190/48640, ttl=64 (request in 1504)
1506	1248.3961310…	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x6608, seq=191/48896, ttl=64 (reply in 1507)
1507	1248.3961565…	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x6608, seq=191/48896, ttl=64 (request in 1506)
1508	1249.3176279…	CiscoInc_a0:d9:8d	CiscoInc_a0:d9:8d	LOOP	60	Reply
1509	1249.3960955…	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x6608, seq=192/49152, ttl=64 (reply in 1510)
1510	1249.3961180…	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x6608, seq=192/49152, ttl=64 (request in 1509)
1511	1250.3960512…	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x6608, seq=193/49408, ttl=64 (reply in 1512)
1512	1250.3960767…	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x6608, seq=193/49408, ttl=64 (request in 1511)
1513	1251.3960277…	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x6608, seq=194/49664, ttl=64 (reply in 1514)
1514	1251.3960531…	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x6608, seq=194/49664, ttl=64 (request in 1513)
1515	1252.3960156…	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x6608, seq=195/49920, ttl=64 (reply in 1516)
1516	1252.3960398…	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x6608, seq=195/49920, ttl=64 (request in 1515)
1517	1253.3959712…	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x6608, seq=196/50176, ttl=64 (reply in 1518)
1518	1253.3959954…	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x6608, seq=196/50176, ttl=64 (request in 1517)
1519	1254.3959613…	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x6608, seq=197/50432, ttl=64 (reply in 1520)
1520	1254.3959852…	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x6608, seq=197/50432, ttl=64 (request in 1519)
1521	1255.3959126…	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x6608, seq=198/50688, ttl=64 (reply in 1522)
1522	1255.3959359…	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x6608, seq=198/50688, ttl=64 (request in 1521)
1523	1256.3958793…	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x6608, seq=199/50944, ttl=64 (reply in 1524)
1524	1256.3959030…	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x6608, seq=199/50944, ttl=64 (request in 1523)
1525	1257.3958594…	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x6608, seq=200/51200, ttl=64 (reply in 1526)
1526	1257.3958852…	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x6608, seq=200/51200, ttl=64 (request in 1525)
1527	1258.3958420…	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x6608, seq=201/51456, ttl=64 (reply in 1528)
1528	1258.3958661…	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x6608, seq=201/51456, ttl=64 (request in 1527)
1529	1259.3252162…	CiscoInc_a0:d9:8d	CiscoInc_a0:d9:8d	LOOP	60	Reply
1530	1259.3958063…	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x6608, seq=202/51712, ttl=64 (reply in 1531)
1531	1259.3958302…	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x6608, seq=202/51712, ttl=64 (request in 1530)

```
▸ Frame 1517: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
▸ Ethernet II, Src: Dell_b0:7d:90 (84:2b:2b:b0:7d:90), Dst: Dell_ec:6e:12 (a4:ba:db:ec:6e:12)
▸ Internet Protocol Version 4, Src: 10.0.0.1, Dst: 10.0.0.2
▸ Internet Control Message Protocol
```

```
0000  a4 ba db ec 6e 12 84 2b  2b b0 7d 90 08 00 45 00   ....n..+ +.}...E.
0010  00 54 60 f2 40 00 40 01  c5 b4 0a 00 00 01 0a 00   .T`.@.@. ........
0020  00 02 08 00 1d 94 66 08  00 c4 1a 5f e2 5b 00 00   ......f. ...._[..
0030  00 00 b4 11 04 00 00 00  00 00 10 11 12 13 14 15   ........ ........
0040  16 17 18 19 1a 1b 1c 1d  1e 1f 20 21 22 23 24 25   ........ .. !"=$%
0050  26 27 28 29 2a 2b 2c 2d  2e 2f 30 31 32 33 34 35   &'()*+,- ./012345
                                                         67
```

Packets: 1531 · Displayed: 153

Problem 5

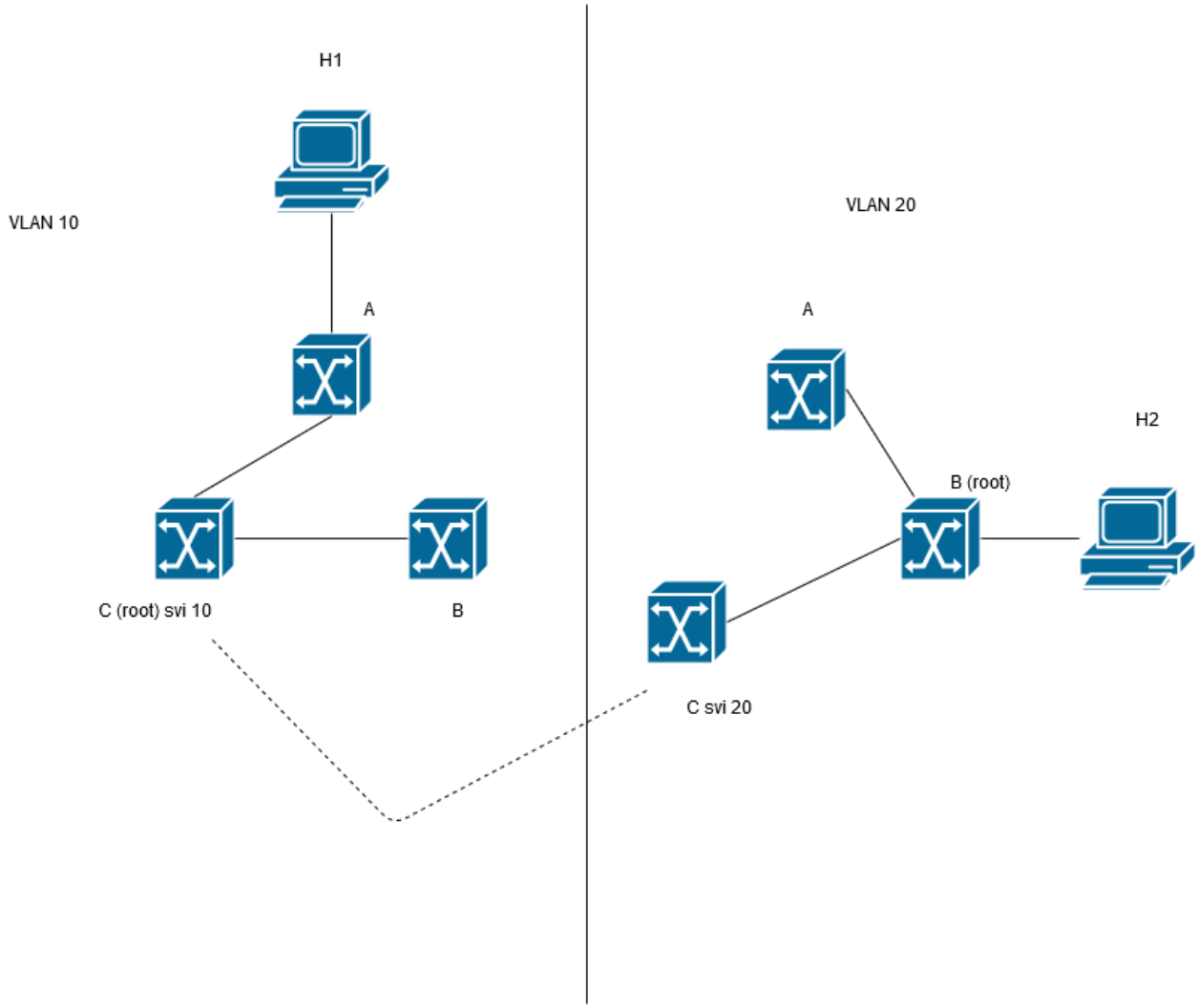
1. **Objective**: To investigate STP operations in a L2 network with loops and multiple vlans
2. **System Under Observation**: vlan 10, vlan 20 are two different systems of observation that will be under the operation of STP in assigning root bridge, root ports, designated ports, and blocking ports
3. **Input**
   a. **Verification Problem:**
      i. 5.1 To show that switch C is the root bridge of vlan 10, run command “show spanning-tree” and make sure that switch C has lowest priority number of all switches on vlan 10 with command “spanning-tree vlan 10 priority 4096”
      ii. 5.2 To show that switch C is not the root bridge of vlan 20, run “command show spanning-tree” and make sure that switch C has higher priority numbers than other switches on vlan 20
   b. **Measurement Problem**:

    i. 5.3 To measure the number of ping packets lost before STP converges again for host 1 on switch A on vlan 10 to host 2 on switch B on vlan 10, ping from H1 to H2.
    ii. 5.4 To measure the number of ping packets lost before STP converges again for host 1 on switch A on vlan 10 to host 2 on switch B on vlan 10, ping from H1 to H2.

4. **Expected Output**:
    a. **Verification Problem**:
        i. 5.1 Expected output is that root bridge for vlan 10 will be switch C
        ii. 5.2 Expected output is that root bridge for vlan 20 will not be switch C
    b. Measurement Problem:
        i. 5.3 Pings will be initially successful from host 1 to host 2. Then, a link will fail, and pings will stop until STP converges again after sufficient time.
        ii. 5.4 Pings will be initially successful from host 1 to host 2. Then, a link will fail, and pings will stop until STP converges again after sufficient time.
5. Actual Output

5.3 50 ping packets
5.4 32 ping packets
For vlan 10, the link break is not connected to the root directly, so that when it fails, it must wait until the max age timer expires. For vlan 20, the broken link was directly connected so that when it fails, it is immediately recognized. Thus, for vlan 10, 50 ping packets were lost before STP convergence, while for vlan 20, only 32 ping packets were lost

5.5 vlan 10 - 31 s
5.5 vlan 20 - 50s
5.5 vlan 10, 20 - 31s
The difference in convergence time has to do with whether a link was directly connected, so that when it fails it is immediate, or if the link happened on a non-direct link, such that it had to wait until the max age timer expired. For vlan 10, the link break is not connected to the root directly, so that when it fails, it must wait until the max age timer expires. For vlan 20, the broken link was directly connected so that when it fails, it is immediately recognized.


H1
VLAN 10
A
C (root) svi 10
B
VLAN 20
A
H2
B (root)
C svi 20

```
Middle_Switch(config)#spanning-tree vlan 10 priority ?
  <0-61440>  bridge priority in increments of 4096

Middle_Switch(config)#spanning-tree vlan 10 priority 4096
Middle_Switch(config)#end
Middle_Switch#
05:02:53: %SYS-5-CONFIG_I: Configured from console by consolshow sp
Middle_Switch#show spanning-tree

VLAN0010
  Spanning tree enabled protocol ieee
  Root ID    Priority    4106
             Address     0018.19a0.d980
             This bridge is the root
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    4106   (priority 4096 sys-id-ext 10)
             Address     0018.19a0.d980
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time 300

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- -------------------------------
Fa0/1            Desg FWD 19        128.3    P2p
Fa0/3            Desg FWD 19        128.5    P2p


VLAN0020
  Spanning tree enabled protocol ieee
  Root ID    Priority    32788
             Address     0015.6285.aa80
             Cost        19
             Port        5 (FastEthernet0/3)
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32788  (priority 32768 sys-id-ext 20)
             Address     0018.19a0.d980
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time 300

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- -------------------------------
Fa0/1            Altn BLK 19        128.3    P2p
Fa0/3            Root FWD 19        128.5    P2p

Middle_Switch#
```

```
eb2027-xia2@eb2027xia2-OptiPlex-780: ~
64 bytes from 10.0.0.2: icmp_seq=42 ttl=64 time=0.585 ms
64 bytes from 10.0.0.2: icmp_seq=43 ttl=64 time=0.603 ms
64 bytes from 10.0.0.2: icmp_seq=44 ttl=64 time=0.559 ms
64 bytes from 10.0.0.2: icmp_seq=45 ttl=64 time=0.578 ms
64 bytes from 10.0.0.2: icmp_seq=46 ttl=64 time=0.601 ms
64 bytes from 10.0.0.2: icmp_seq=47 ttl=64 time=0.603 ms
64 bytes from 10.0.0.2: icmp_seq=48 ttl=64 time=0.583 ms
64 bytes from 10.0.0.2: icmp_seq=49 ttl=64 time=0.604 ms
64 bytes from 10.0.0.2: icmp_seq=50 ttl=64 time=0.522 ms
64 bytes from 10.0.0.2: icmp_seq=51 ttl=64 time=0.583 ms
64 bytes from 10.0.0.2: icmp_seq=52 ttl=64 time=0.637 ms
From 10.0.0.1 icmp_seq=99 Destination Host Unreachable
From 10.0.0.1 icmp_seq=100 Destination Host Unreachable
From 10.0.0.1 icmp_seq=101 Destination Host Unreachable
64 bytes from 10.0.0.2: icmp_seq=102 ttl=64 time=1.14 ms
64 bytes from 10.0.0.2: icmp_seq=103 ttl=64 time=0.586 ms
64 bytes from 10.0.0.2: icmp_seq=104 ttl=64 time=0.537 ms
64 bytes from 10.0.0.2: icmp_seq=105 ttl=64 time=0.546 ms
64 bytes from 10.0.0.2: icmp_seq=106 ttl=64 time=0.574 ms
^C
--- 10.0.0.2 ping statistics ---
106 packets transmitted, 57 received, +3 errors, 46% packet loss, time 104998ms
rtt min/avg/max/mdev = 0.522/0.601/1.142/0.076 ms, pipe 4
eb2027-xia2@eb2027xia2-OptiPlex-780:~$
```

```
^C
--- 10.0.0.2 ping statistics ---
106 packets transmitted, 57 received, +3 errors, 46% packet loss, time 1
rtt min/avg/max/mdev = 0.522/0.601/1.142/0.076 ms, pipe 4
eb2027-xla2@eb2027xla2-OptiPlex-780:~$ ping 10.0.0.2\
> ^C
eb2027-xla2@eb2027xla2-OptiPlex-780:~$ ^C
eb2027-xla2@eb2027xla2-OptiPlex-780:~$ ping 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.600 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.584 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.577 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.584 ms
64 bytes from 10.0.0.2: icmp_seq=36 ttl=64 time=1.10 ms
64 bytes from 10.0.0.2: icmp_seq=37 ttl=64 time=0.548 ms
64 bytes from 10.0.0.2: icmp_seq=38 ttl=64 time=0.578 ms
64 bytes from 10.0.0.2: icmp_seq=39 ttl=64 time=0.544 ms
64 bytes from 10.0.0.2: icmp_seq=40 ttl=64 time=0.553 ms
64 bytes from 10.0.0.2: icmp_seq=41 ttl=64 time=0.568 ms
64 bytes from 10.0.0.2: icmp_seq=42 ttl=64 time=0.554 ms
64 bytes from 10.0.0.2: icmp_seq=43 ttl=64 time=0.542 ms
```

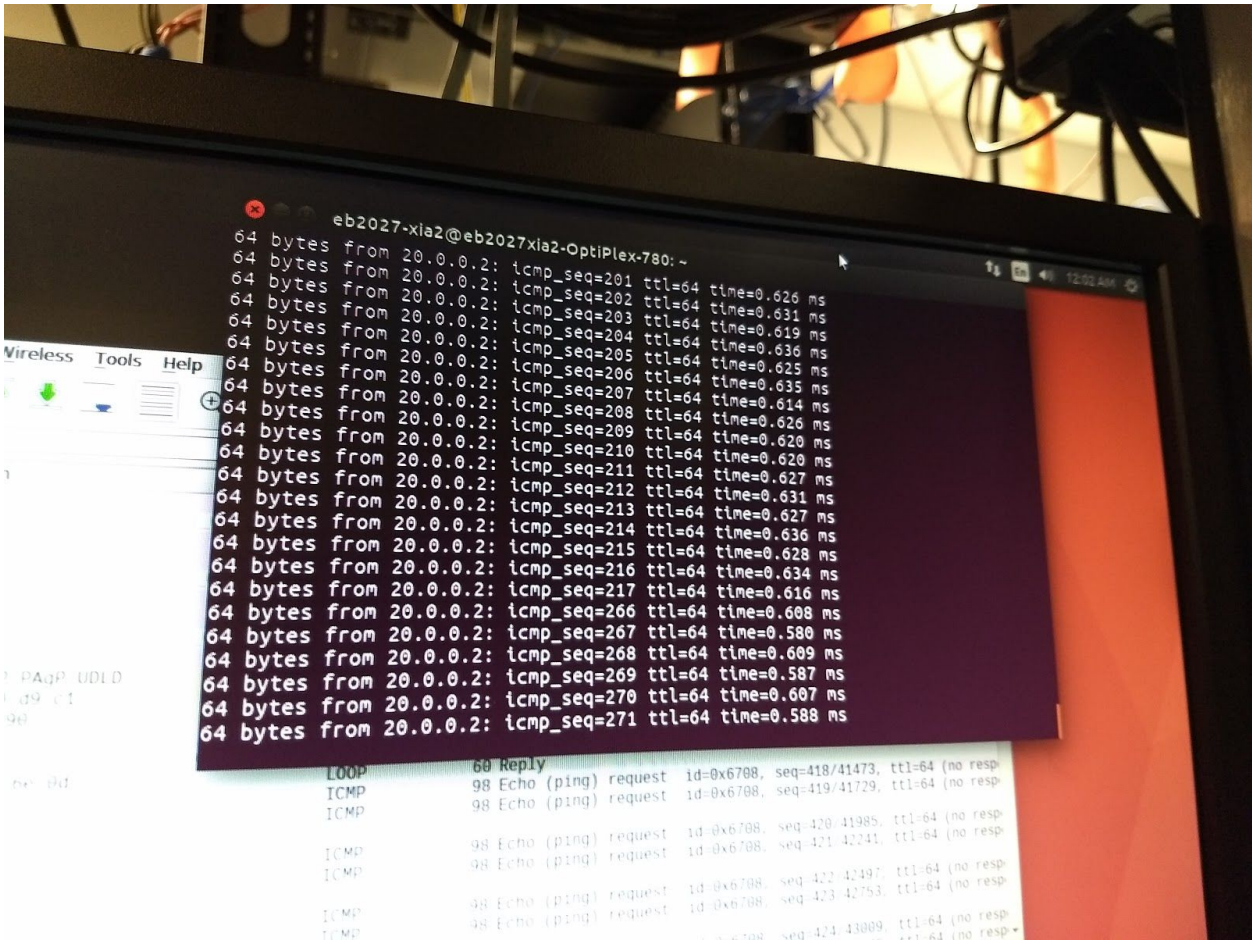
SSH
SSH
SSH

```
64 bytes from 20.0.0.2: icmp_seq=73 ttl=64 time=0.641 ms
64 bytes from 20.0.0.2: icmp_seq=74 ttl=64 time=0.638 ms
64 bytes from 20.0.0.2: icmp_seq=75 ttl=64 time=0.619 ms
64 bytes from 20.0.0.2: icmp_seq=76 ttl=64 time=0.637 ms
64 bytes from 20.0.0.2: icmp_seq=77 ttl=64 time=0.619 ms
64 bytes from 20.0.0.2: icmp_seq=78 ttl=64 time=0.631 ms
64 bytes from 20.0.0.2: icmp_seq=79 ttl=64 time=0.625 ms
64 bytes from 20.0.0.2: icmp_seq=80 ttl=64 time=0.623 ms
64 bytes from 20.0.0.2: icmp_seq=81 ttl=64 time=0.641 ms
64 bytes from 20.0.0.2: icmp_seq=82 ttl=64 time=0.631 ms
64 bytes from 20.0.0.2: icmp_seq=83 ttl=64 time=0.639 ms
64 bytes from 20.0.0.2: icmp_seq=84 ttl=64 time=0.636 ms
64 bytes from 20.0.0.2: icmp_seq=85 ttl=64 time=0.639 ms
64 bytes from 20.0.0.2: icmp_seq=86 ttl=64 time=0.636 ms
64 bytes from 20.0.0.2: icmp_seq=87 ttl=64 time=0.617 ms
64 bytes from 20.0.0.2: icmp_seq=88 ttl=64 time=0.628 ms
64 bytes from 20.0.0.2: icmp_seq=89 ttl=64 time=0.624 ms
64 bytes from 20.0.0.2: icmp_seq=90 ttl=64 time=0.624 ms
64 bytes from 20.0.0.2: icmp_seq=91 ttl=64 time=0.643 ms
64 bytes from 20.0.0.2: icmp_seq=122 ttl=64 time=0.618 ms
64 bytes from 20.0.0.2: icmp_seq=123 ttl=64 time=0.612 ms
64 bytes from 20.0.0.2: icmp_seq=124 ttl=64 time=0.604 ms
64 bytes from 20.0.0.2: icmp_seq=125 ttl=64 time=0.601 ms
```

Help

LOOP	60	Reply
ICMP	98	Echo (ping) request id=0x6708, seq=418/41473, ttl=64 (no resp
ICMP	98	Echo (ping) request id=0x6708, seq=419/41729, ttl=64 (no resp
ICMP	98	Echo (ping) request id=0x6708, seq=420/41985, ttl=64 (no resp
ICMP	98	Echo (ping) request id=0x6708, seq=421/42241, ttl=64 (no resp
ICMP	98	Echo (ping) request id=0x6708, seq=422/42497, ttl=64 (no resp

Router 4
SSH
A
SSH
B
SSH
C


eb2027-xia2@eb2027xia2-OptiPlex-780: ~
64 bytes from 20.0.0.2: icmp_seq=201 ttl=64 time=0.626 ms
64 bytes from 20.0.0.2: icmp_seq=202 ttl=64 time=0.631 ms
64 bytes from 20.0.0.2: icmp_seq=203 ttl=64 time=0.619 ms
64 bytes from 20.0.0.2: icmp_seq=204 ttl=64 time=0.636 ms
64 bytes from 20.0.0.2: icmp_seq=205 ttl=64 time=0.625 ms
64 bytes from 20.0.0.2: icmp_seq=206 ttl=64 time=0.635 ms
64 bytes from 20.0.0.2: icmp_seq=207 ttl=64 time=0.614 ms
64 bytes from 20.0.0.2: icmp_seq=208 ttl=64 time=0.626 ms
64 bytes from 20.0.0.2: icmp_seq=209 ttl=64 time=0.620 ms
64 bytes from 20.0.0.2: icmp_seq=210 ttl=64 time=0.620 ms
64 bytes from 20.0.0.2: icmp_seq=211 ttl=64 time=0.627 ms
64 bytes from 20.0.0.2: icmp_seq=212 ttl=64 time=0.631 ms
64 bytes from 20.0.0.2: icmp_seq=213 ttl=64 time=0.627 ms
64 bytes from 20.0.0.2: icmp_seq=214 ttl=64 time=0.636 ms
64 bytes from 20.0.0.2: icmp_seq=215 ttl=64 time=0.628 ms
64 bytes from 20.0.0.2: icmp_seq=216 ttl=64 time=0.634 ms
64 bytes from 20.0.0.2: icmp_seq=217 ttl=64 time=0.616 ms
64 bytes from 20.0.0.2: icmp_seq=266 ttl=64 time=0.608 ms
64 bytes from 20.0.0.2: icmp_seq=267 ttl=64 time=0.580 ms
64 bytes from 20.0.0.2: icmp_seq=268 ttl=64 time=0.609 ms
64 bytes from 20.0.0.2: icmp_seq=269 ttl=64 time=0.587 ms
64 bytes from 20.0.0.2: icmp_seq=270 ttl=64 time=0.607 ms
64 bytes from 20.0.0.2: icmp_seq=271 ttl=64 time=0.588 ms
Wireless Tools Help
LOOP 60 Reply
ICMP 98 Echo (ping) request id=0x6708, seq=418/41473, ttl=64 (no resp
ICMP 98 Echo (ping) request id=0x6708, seq=419/41729, ttl=64 (no resp