James Cross
Michael Patel
ECE 577
Lab Hw 2

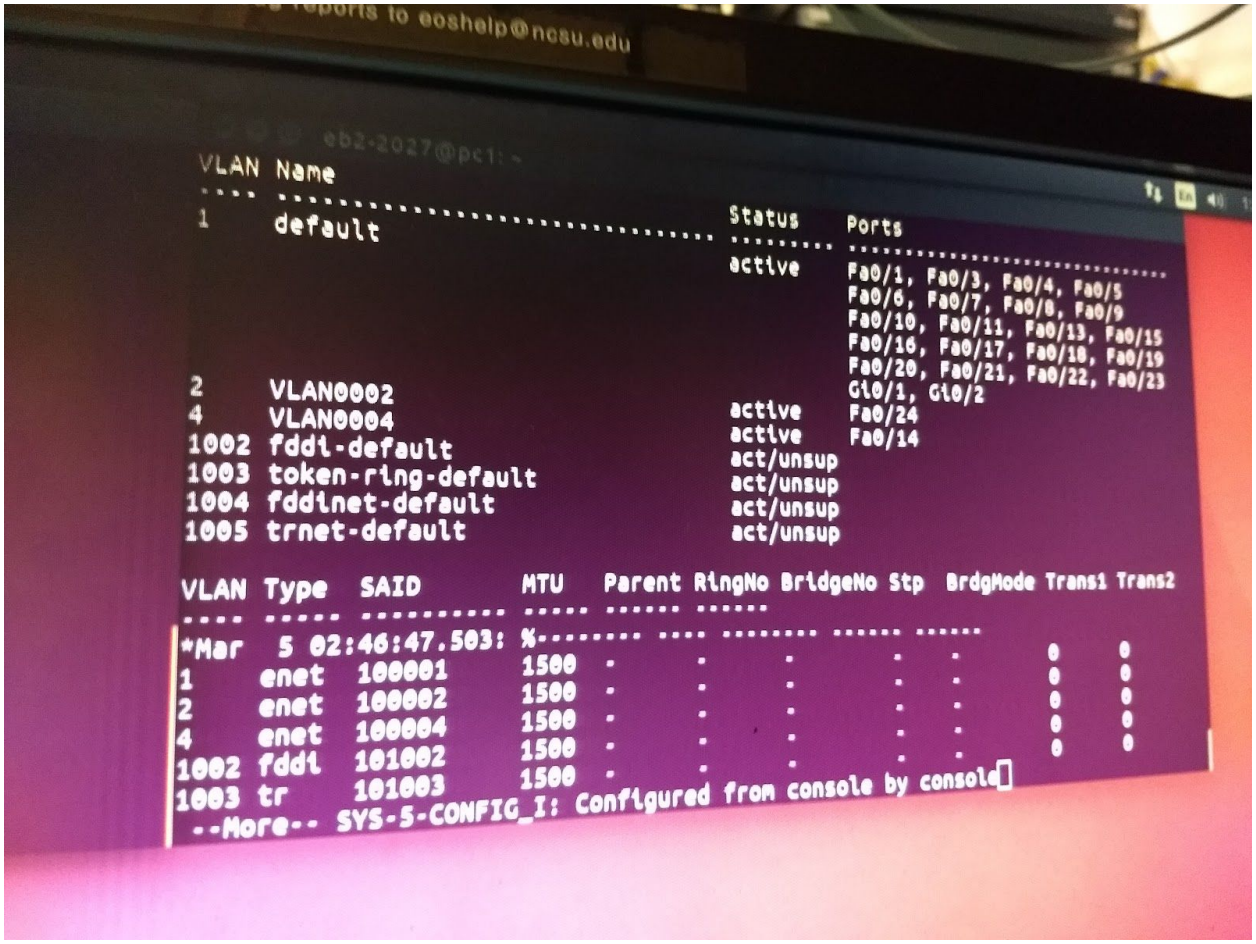
# Problem 1 (Q1-4)

1. **Objective**: To configure vlan and trunk ports, and to verify L2 vlan connectivity over a trunk
2. **System Under Observation**: Multiple tests will be conducted at different switch ports, so the system under observation is 2 switches (S1 and S2), 1 trunk port, 2 vlans (vlan1 and vlan2) on each switch. A “ping test” will be conducted at hosts in vlan2.
3. **Input**
   a. **Verification Problem**: Once topology has been configured, ping command will be used to verify connectivity/verify network segmentation
4. **Expected Output:**
   a. **Verification Problem:**
      i. Q1: Expected output is that host B on vlan2-S2 will receive ping request, and that host A on vlan2-S1 receives ping reply
      ii. Q2: Expected output is that host B on vlan4-S1 will receive ping request that host A on vlan4-S1 will receive reply. A mirrored trunk port on S2 will verify that no vlan4 traffic crosses trunk.
      iii. Q3: Expected output is that node <> on S2 will receive ping request, and node <> on S1 will receive ping reply. Vlan1 and Vlan2 on both S1 and S2 will not receive ping request. This verifies that trunk is carrying default/native traffic, but not interfering with vlan traffic.
5. **Actual Output**

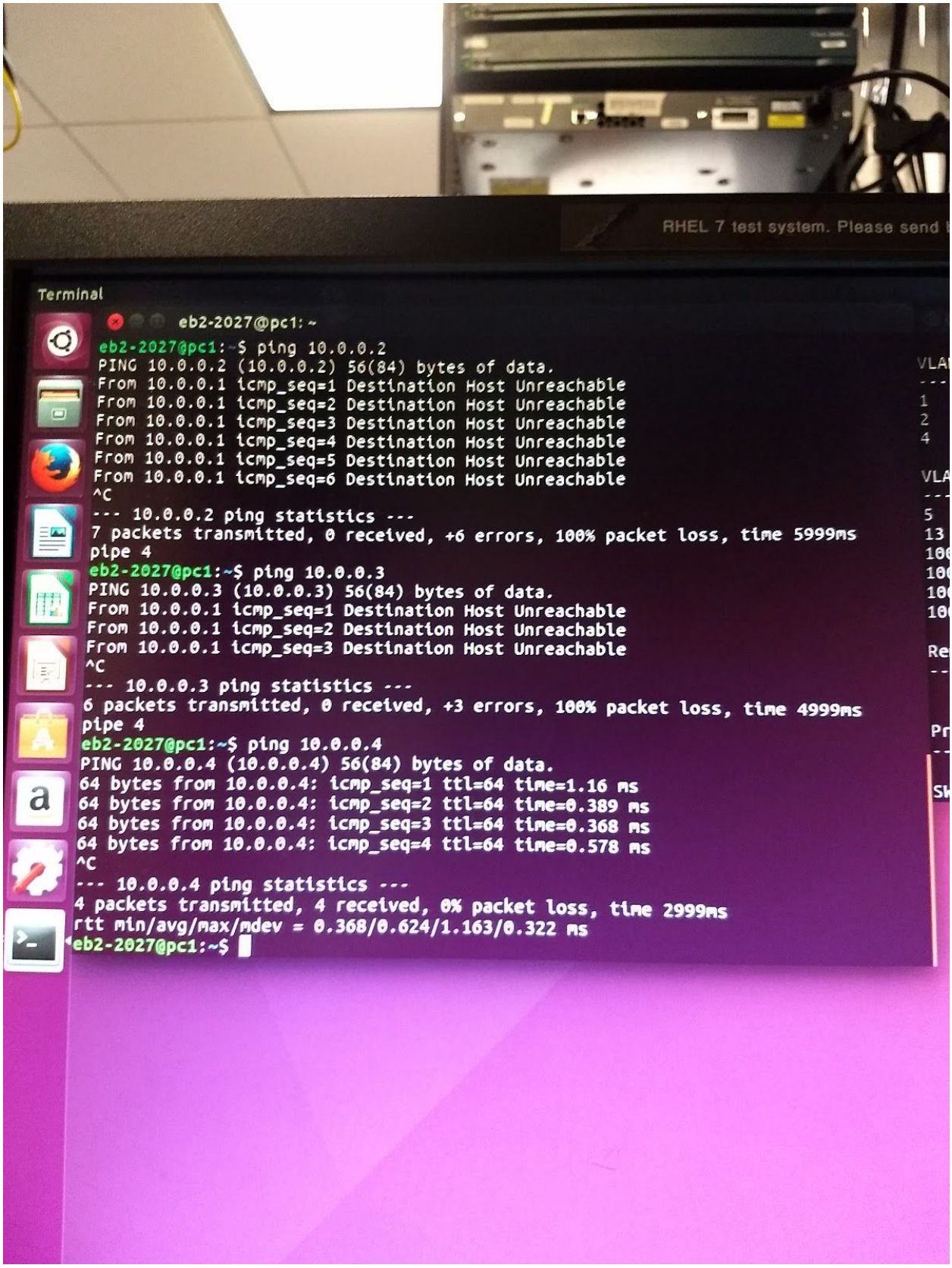
Q1.1 successful ping over trunk

```
BOOL := [1|0]
FEATURES := ecn
eb2-2027@pc1:~$ sudo ip addr add 10.0.0.1/24 dev enp0s25
sudo: unable to resolve host pc1: Connection timed out
eb2-2027@pc1:~$ ip route
10.0.0.0/24 dev enp0s25  proto kernel  scope link  src 10.0.0.1
169.254.0.0/16 dev enp0s25  scope link  metric 1000
eb2-2027@pc1:~$ 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=1.16 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.390 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.421 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.376 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.365 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.387 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.591 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.183 ms
^C
--- 10.0.0.2 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7000ms
rtt min/avg/max/mdev = 0.183/0.484/1.165/0.278 ms
eb2-2027@pc1:~$
```

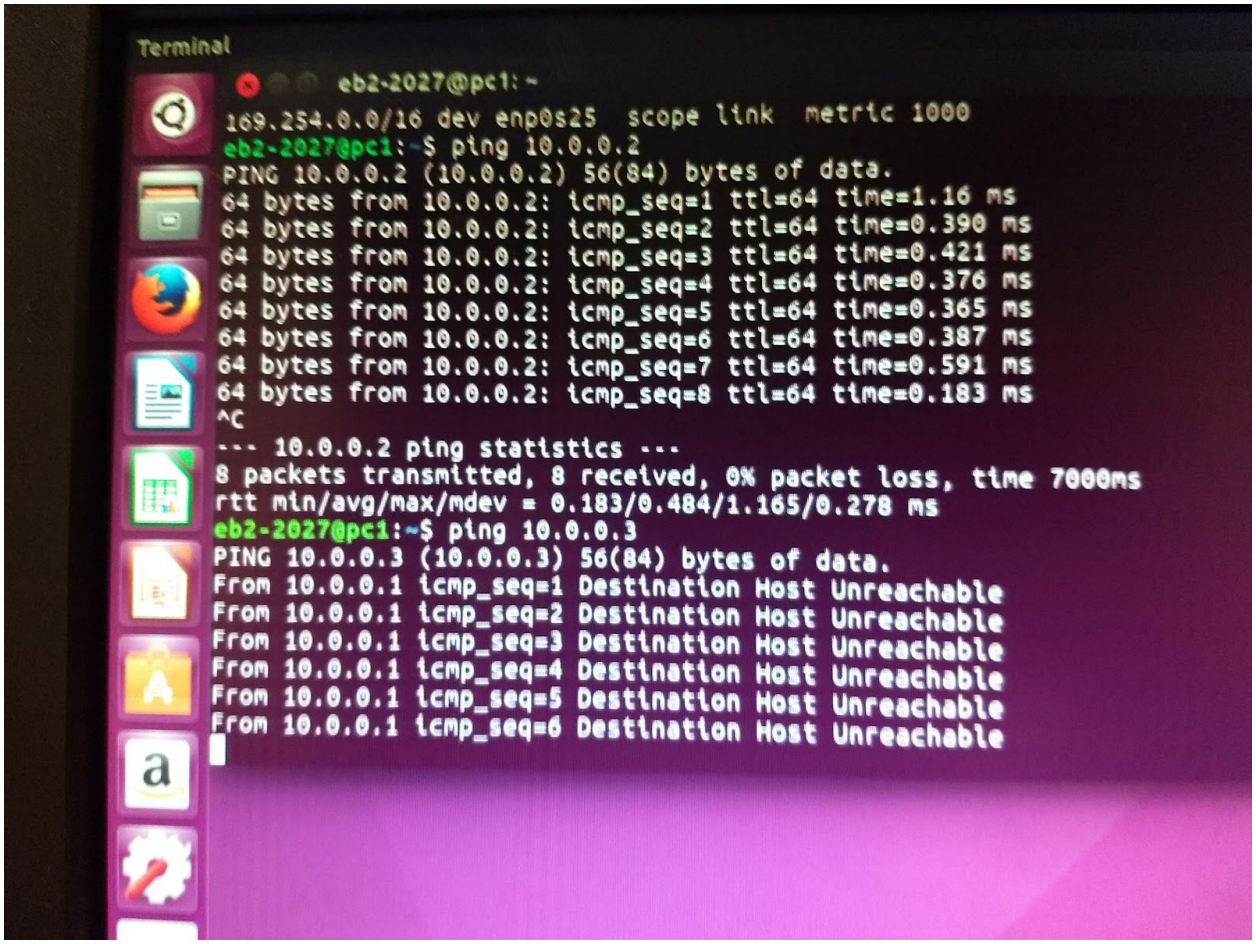
Q1.2



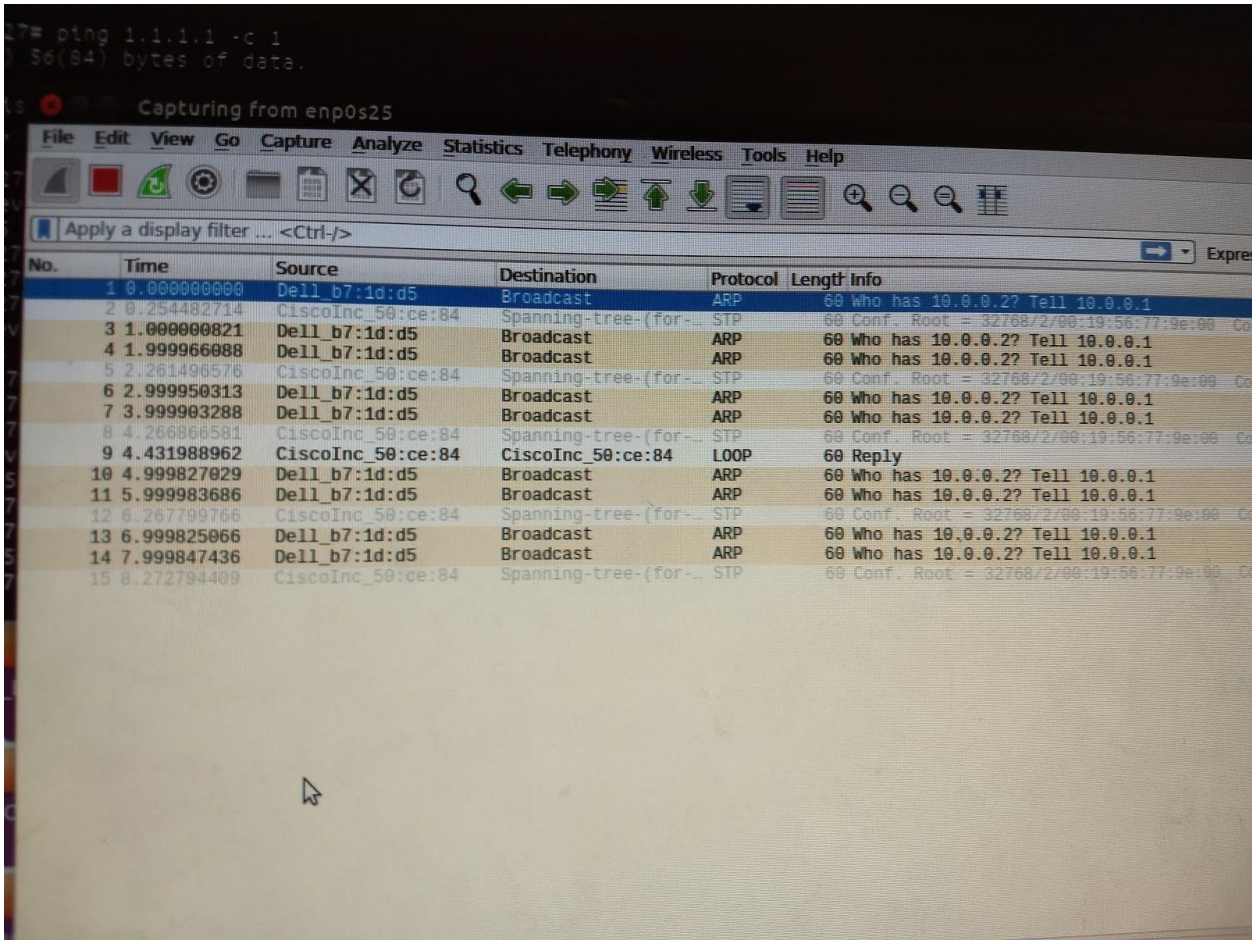

Created vlan2 and vlan4




ping from host 1 in vlan2 to host 2 in vlan2 successful, not successful for hosts in different vlans

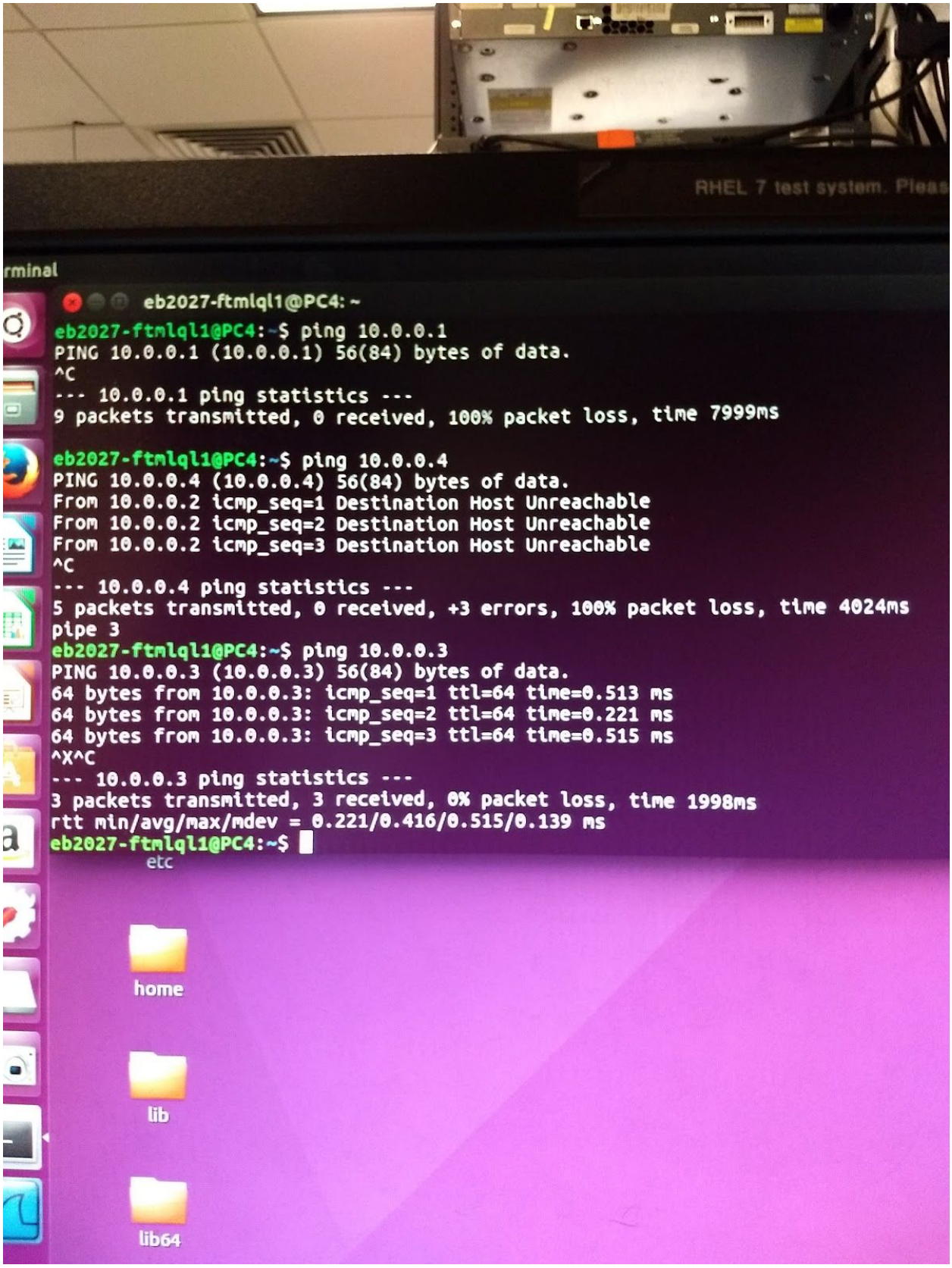



Hosts in different vlans

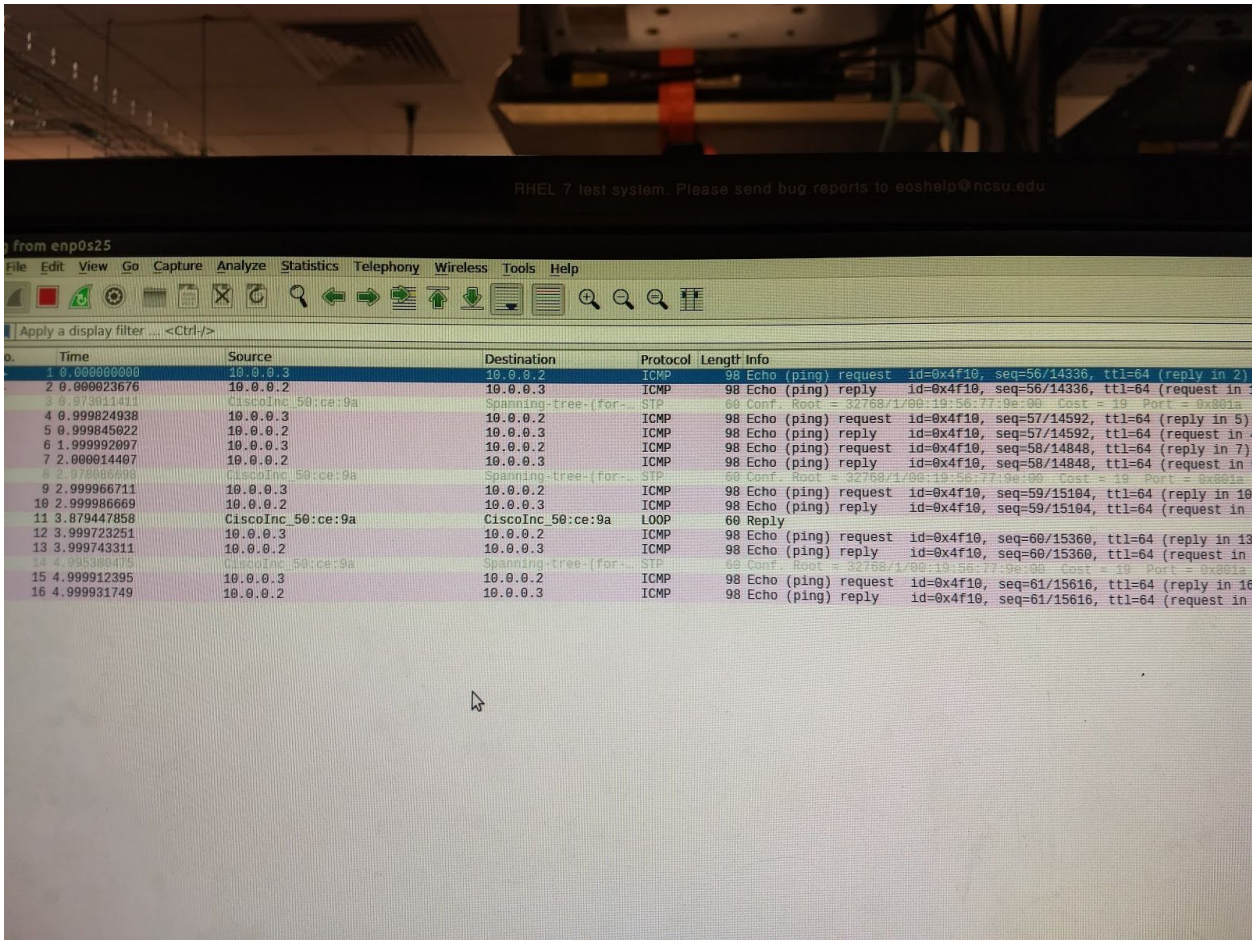



Wireshark of broadcasted ARPs - different vlans

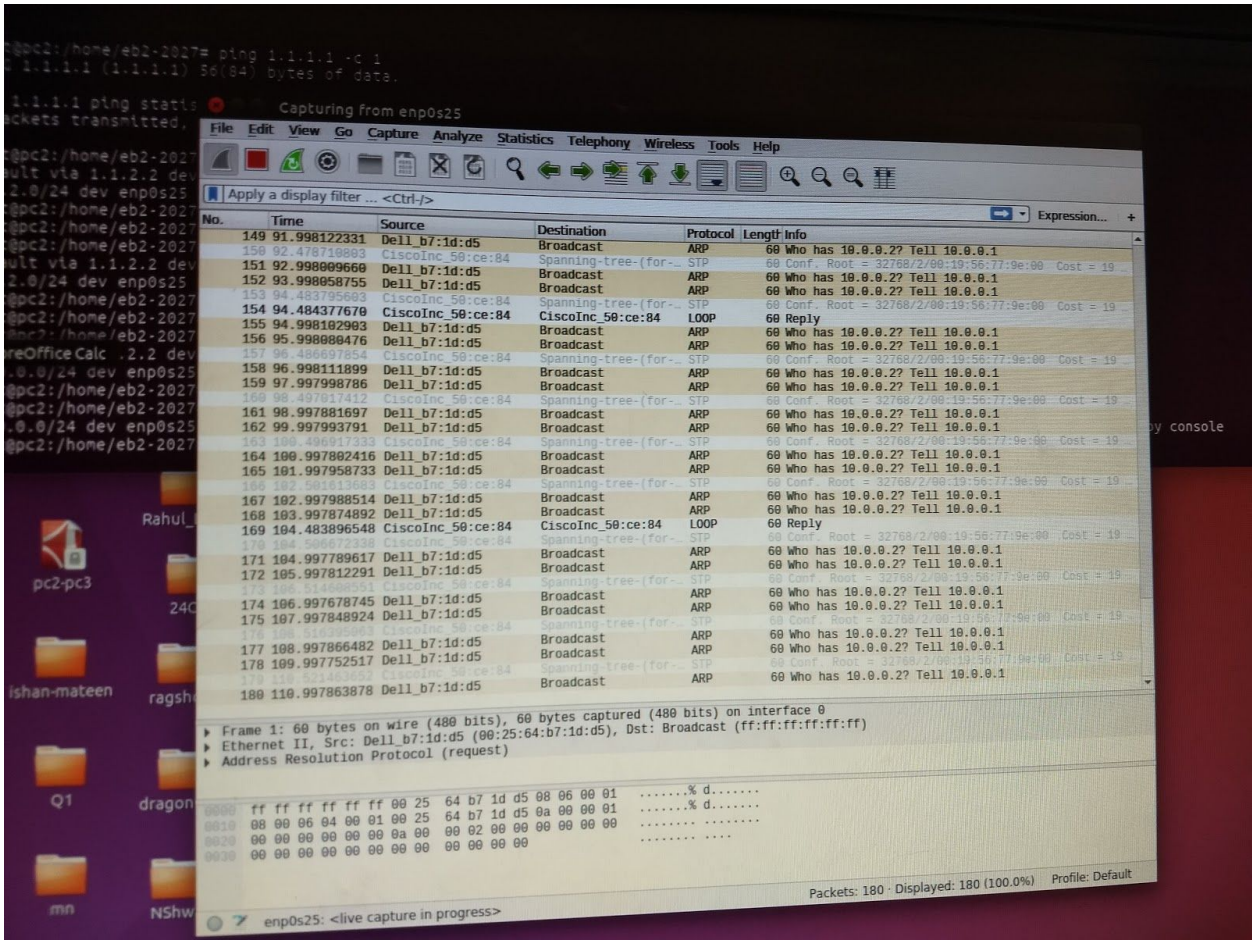
Q1.3 and Q1.4




vlan1 connectivity




Wireshark of successful ping over vlan1




Wireshark of broadcast ARPs

---

[IEEE 802.1q standards]
Approximately 60 different experiments

# Problem 1.2 (Q5 experiment 1)

1. **Objective**: To configure vlan and trunk port, and to verify L2 connectivity
2. **System Under Observation:** Multiple tests will be conducted at different switch ports, so the system under observation is 2 switches (S1 and S2), 1 trunk port, 2 vlans (vlan1 and vlan2) on each switch. However, the specific portion to be observed is at the trunk ports to examine vlan tagging done by trunk port.
3. **Input**
    a. **Verification Problem**: Ping test while running Wireshark to capture packets in order to examine 4-byte tag inserted by trunk. The 4-byte tag should correspond to 4096 vlans possible.
4. **Expected Output:**

    a. **Verification Problem**: Using Wireshark captures, expand packet header to identify 4-byte tag used as a vlan tag placed by the trunk port. Verify that 4-byte tag is within range of 4096 possible.
5. Actual Output

---

# Problem 1 (Q5 experiment 2)

1. **Objective**: To configure vlan and trunk port, and to verify L2 connectivity
2. **System Under Observation:** Multiple tests will be conducted at different switch ports, so the system under observation is 2 switches (S1 and S2), 1 trunk port, 2 vlans (vlan1 and vlan2) on each switch. However, the specific site of observation is at the receiving end of the trunk link.
3. **Input**
    a. **Verification Problem**: Conduct a ping test that will cross through trunk link, so that the trunk port tags the packet. Use Wireshark to perform packet capture, so as to examine headers.
4. **Expected Output:**
    a. **Verification Problem**: Using Wireshark captures, expand packet header to observe that vlan tag has been removed at receiving end of trunk link (to ensure that packet will be forwarded to correct vlan thereafter)
5. Actual Output

---

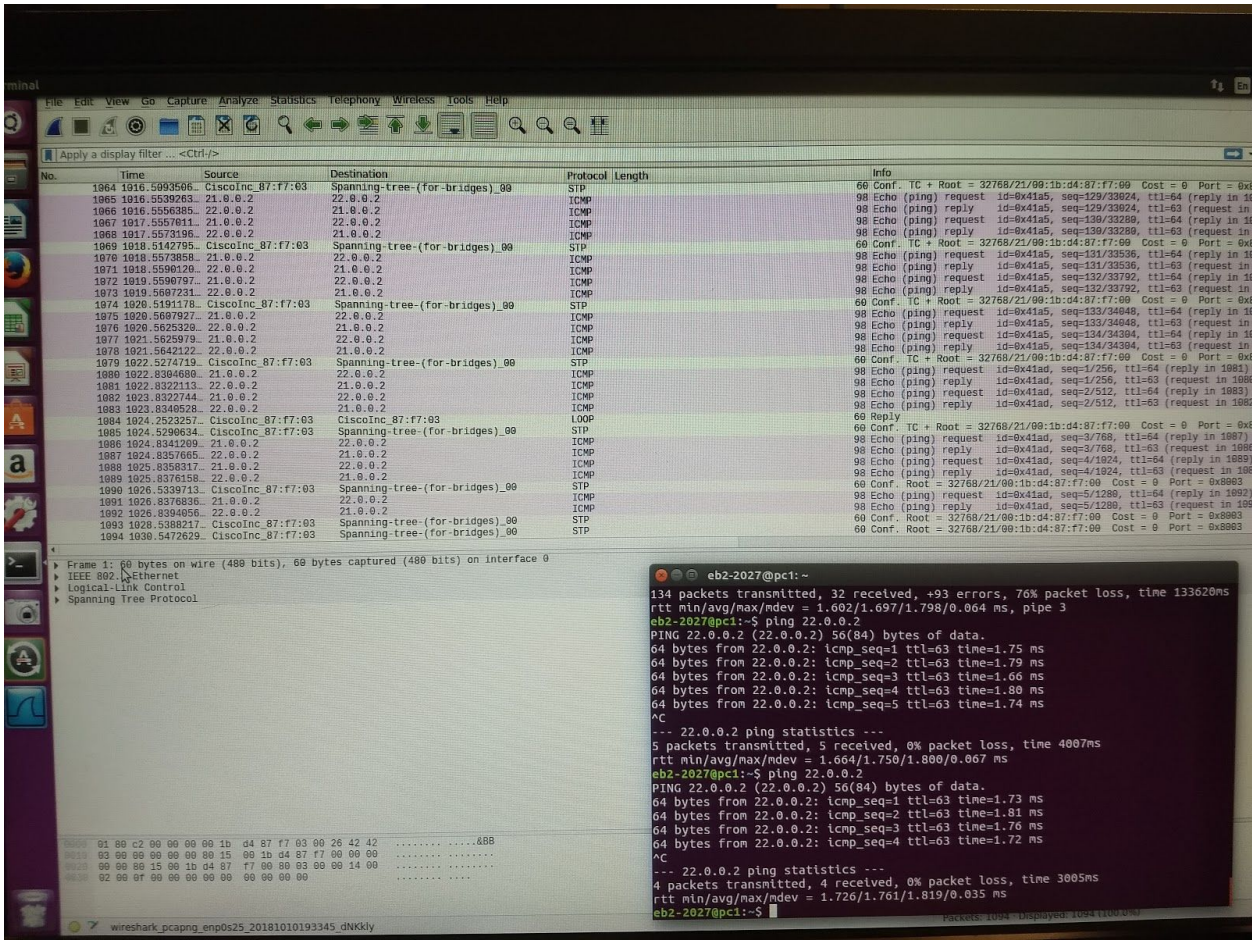
# Problem 1 (Q5 experiment 3)

1. **Objective**: To configure vlan and trunk port, and to verify L2 connectivity
2. **System Under Observation:** Multiple tests will be conducted at different switch ports, so the system under observation is 2 switches (S1 and S2), 1 trunk port, 2 vlans (vlan1 and vlan2) on each switch. However, site of observation will be at receiving end of trunk link.
3. **Input**
    a. **Verification Problem**: Conduct a ping test with Wireshark capturing running to verify that the trunk port does not tag packets on the native vlan.
4. **Expected Output:**
    a. **Verification Problem**: Using Wireshark captures, expand packet header to observe that the packets have not been vlan tagged by the trunk port since packets on the native vlan should not be tagged.
5. Actual Output

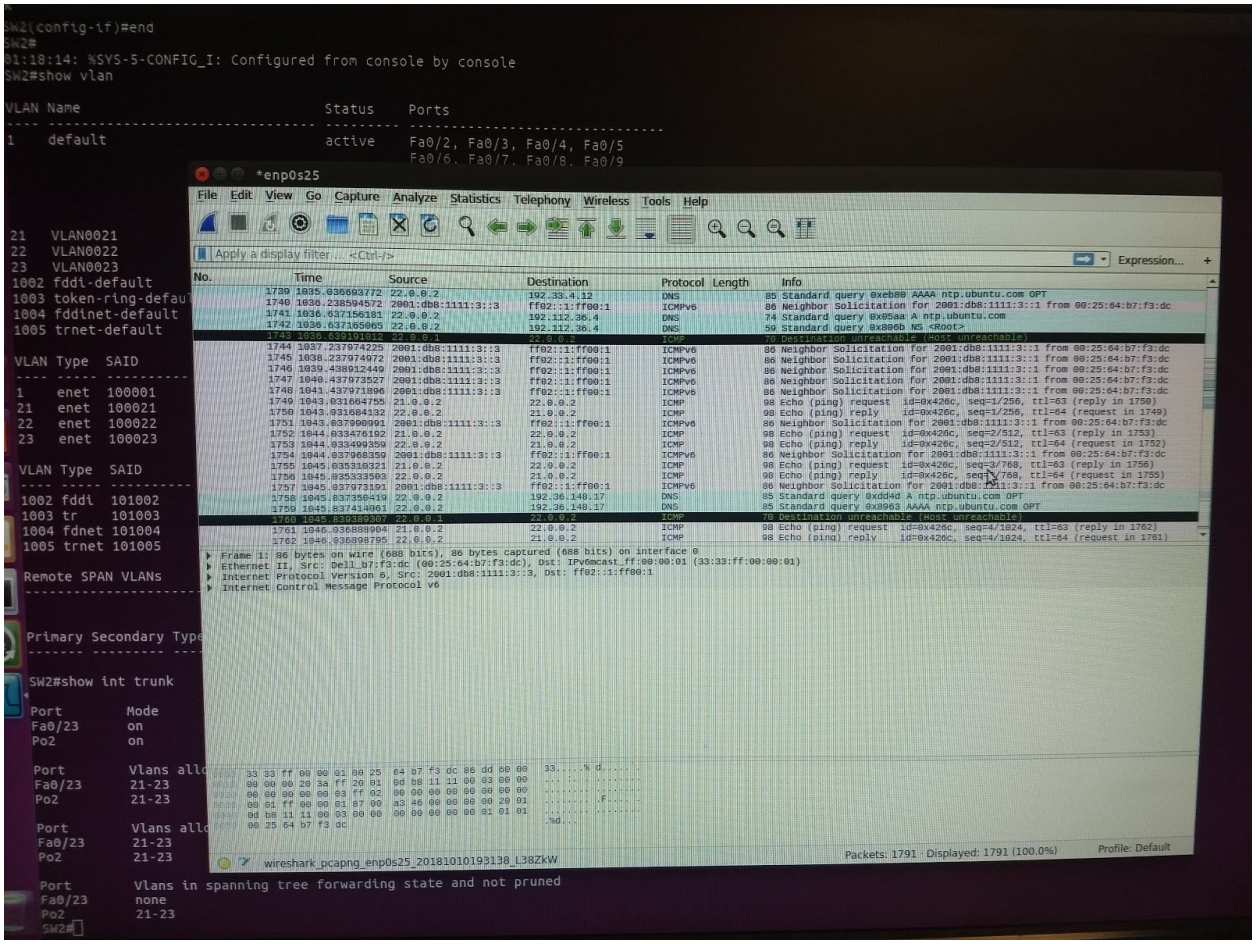
---

# Problem 2

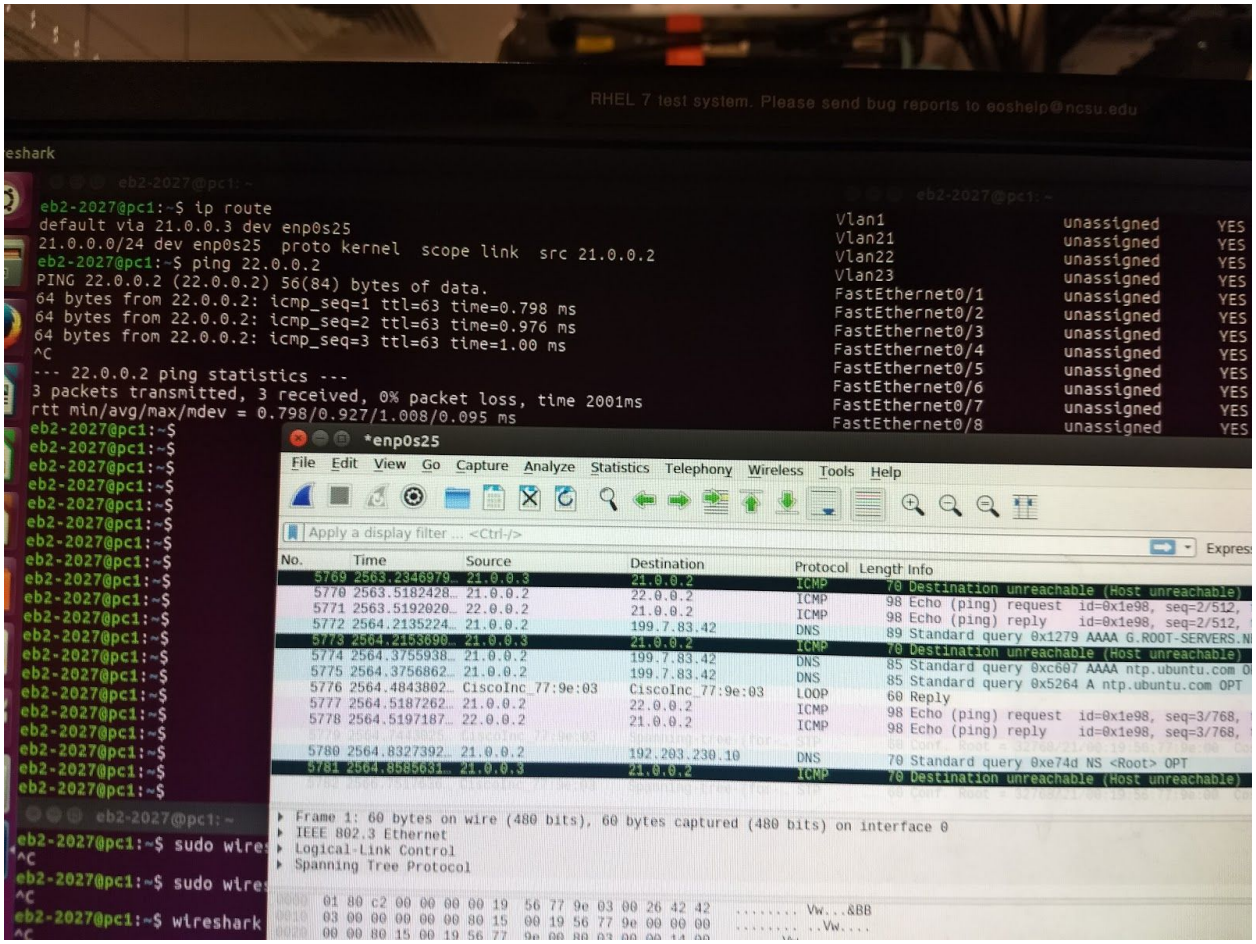
1. **Objective**: To perform link aggregation and perform vlan management using gateways on switches or routers
2. **System Under Observation**: The system under observation includes 2 switches (SwA and SwB) connected via 2 trunk lines, 1 router connected to a switch, and 3 named vlans (vlan21, vlan22, vlan23) connected to the switch-not-connected to the router. Each vlan has 2 hosts each.
3. **Input**
    a. **Verification Problem**: Once topology is configured, the ping command will be used to perform several different task verifications.
4. **Expected Output:**
    a. **Verification Problem**:
        i. Q5: The following ports will have the following packet captures:
            1. Host A's port on vlan21-SwA will see 4 packet captures (ARP request to SwB gateway, ARP response from SwB gateway, ping request to Host B, ping reply from Host B)
            2. Host B's port on vlan22-SwA will see 4 packet captures (ARP request from SwB gateway, ARP response to SwB gateway, ping echo request from Host A via SwB gateway, ping reply to Host A via SwB gateway)
            3. SwA and SwB trunk ports will see 4 packet captures (ARP request to SwB gateway, ARP reply from SwB gateway, ping request and ping reply to SwB gateway)
            4. The trunk ports of SwB will receive tagged packets while the SVI virtual ports will not receive tagged packets.
        ii. Q6: The following ports will have the following packet captures:
            1. Host A's port on vlan21-SwA will see 4 packet captures (ARP request to Router gateway, ARP response from Router gateway, ping request to Host B, ping reply from Host B)
            2. Host B's port on vlan22-SwA will see 4 packet captures (ARP request from Router gateway, ARP response to Router gateway, ping echo request from Host A via Router gateway, ping reply to Host A via Rotuer gateway)
            3. SwA and SwB trunk ports will see 4 packet captures (ARP request to Router gateway, ARP reply from Router gateway, ping request and ping reply to Router gateway)
5. **Actual Output**



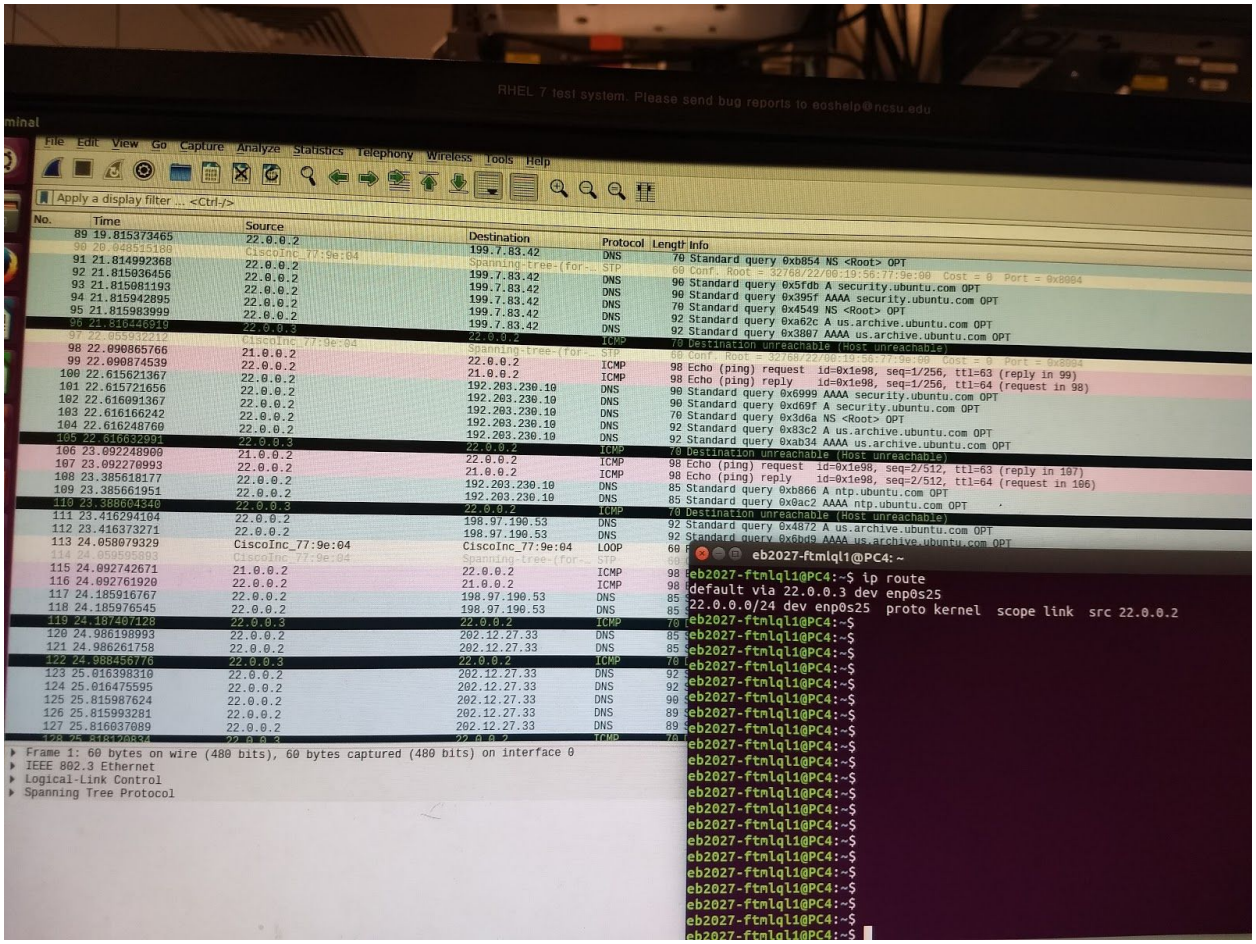

Wireshark and ping successful of host1 in vlan21 to gateways in SwB



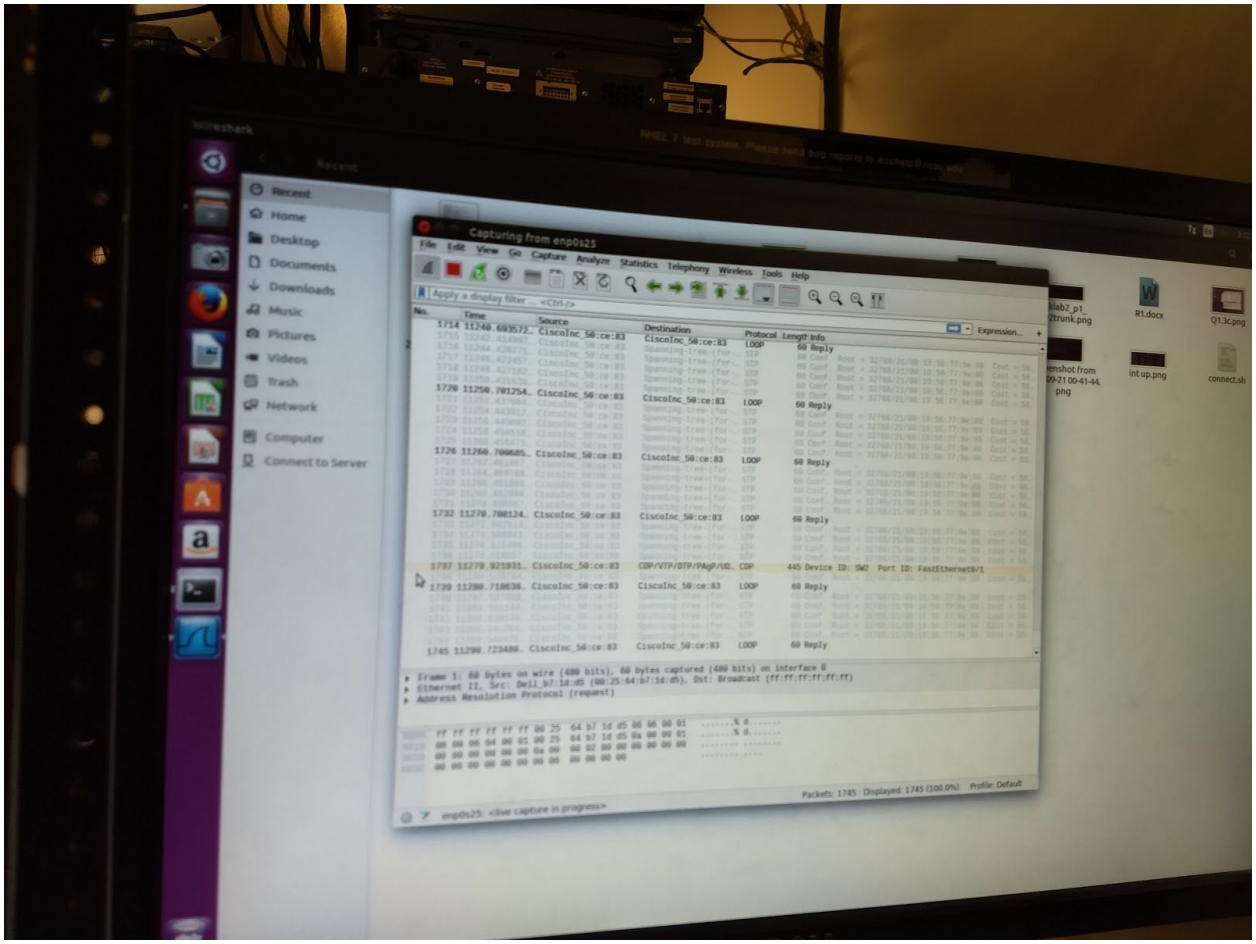

Wireshark of successful connectivity




Wireshark and ping successful from host in vlan21 to host in vlan22




Wireshark of successful ping




Wireshark of vlan23 not receiving ping traffic

---

# Problem 3

1. **Objective**: To query and return L2 topology information/network structure via remote automation and ssh
2. **System Under Observation**: The system under observation includes 2 switches (SwA and SwB) connected via 2 trunk lines, 1 router connected to a switch, and 3 named vlans (vlan21, vlan22, vlan23) connected to the switch-not-connected to the router. Each vlan has 2 hosts each.
3. **Input**
   a. **Measurement Problem**: python p3.py <IP address 1> <IP address 2> <IP address n>, using “show cdp neighbors detail” Cisco CLI command to retrieve desired information
4. **Expected Output:**
   a. **Measurement Problem:** For Python script, console print out of adjacency list

## 5. Actual Output

```
PS C:\Users\micha\Desktop> python .\p3.py 169.254.231.151 123.123.123.1

Invalid IP given: 123.123.123.1

#######################
{'169.254.231.151': {'Neighbor 1: Sw1.abc.com': {'Local Node Interface': 'FastEthernet0/5', 'Adjacent Node Interface': 'FastEthernet0/
7', 'Local VLAN': '1', 'Adjacent VLAN': '1'}, 'Neighbor 2: Sw1.abc.com': {'Local Node Interface': 'FastEthernet0/11', 'Adjacent Node I
nterface': 'FastEthernet0/11', 'Local VLAN': '20', 'Adjacent VLAN': '20'}}}
#######################
PS C:\Users\micha\Desktop>
```