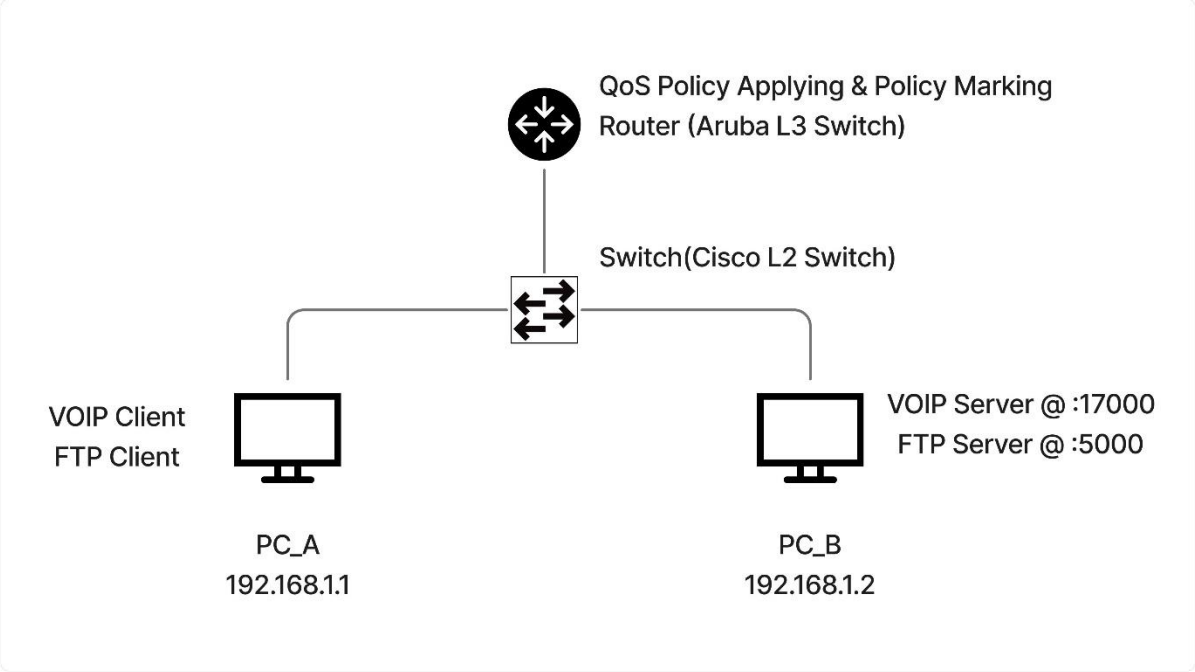
CO513: Advanced Computer Communication Networks - Lab 03

E/19/309, E/19/349, E/19/372, E/19/409,
E/19/413, E/19/426, E/19/ 432, E/19/443,
E/19/446, E/19/452, E/19/455

16/05/2025

Network Configuration and Setup Summary



We initially began the lab with the recommended network topology consisting of two PCs (Client A and Client B), one router, and one switch. However, we faced ongoing challenges in finding the correct Aruba version and its commands for the router, which was also running an older version of the firmware. These issues complicated the configuration process and diverted focus from the main objective of the lab. As a result, we were instructed to remove the switch from the topology in order to simplify the setup. This adjustment did not impact the core QoS policy implementation or testing, as the switch was not essential for the classification and marking processes carried out on the router.

Equipment used:

- Routing: Aruba HP 2920 Layer 3 Switch
  Firmware Version: 15 (https://www.hpe.com/psnow/doc/c04111401?ver=15)
- Switching: Cisco Layer 2 Switch

Lab Tasks:

1. Generate Traffic

For traffic generation, we used iPerf3 to simulate both FTP (for Bulk Data) and VoIP (for Voice Data) traffic. The corresponding iPerf3 servers were also configured on PC-B to receive the traffic by executing the command “iperf3 -s”.

The following are the traffic generation snapshots (iperf3 clients)

a. VOIP Traffic

```
root@LAPTOP-LAL35MFR:/mnt/c/Users/Nipul# iperf3 -c 192.168.1.2 -u -b 512k -t 300 --tos 184 -p 17000
Connecting to host 192.168.1.2, port 17000
[  5] local 172.26.99.113 port 58233 connected to 192.168.1.2 port 17000
[ ID] Interval           Transfer     Bitrate         Total Datagrams
[  5]   0.00-1.00   sec  63.6 KBytes   521 Kbits/sec  45
[  5]   1.00-2.00   sec  62.2 KBytes   510 Kbits/sec  44
[  5]   2.00-3.00   sec  62.2 KBytes   510 Kbits/sec  44
[  5]   3.00-4.00   sec  62.2 KBytes   510 Kbits/sec  44
[  5]   4.00-5.00   sec  62.2 KBytes   510 Kbits/sec  44
[  5]   5.00-6.00   sec  63.6 KBytes   521 Kbits/sec  45
[  5]   6.00-7.00   sec  62.2 KBytes   510 Kbits/sec  44
[  5]   7.00-8.00   sec  62.2 KBytes   510 Kbits/sec  44
[  5]   8.00-9.00   sec  62.2 KBytes   510 Kbits/sec  44
[  5]   9.00-10.00  sec  62.2 KBytes   510 Kbits/sec  44
[  5]  10.00-11.00  sec  63.6 KBytes   521 Kbits/sec  45
[  5]  11.00-12.00  sec  62.2 KBytes   510 Kbits/sec  44
```

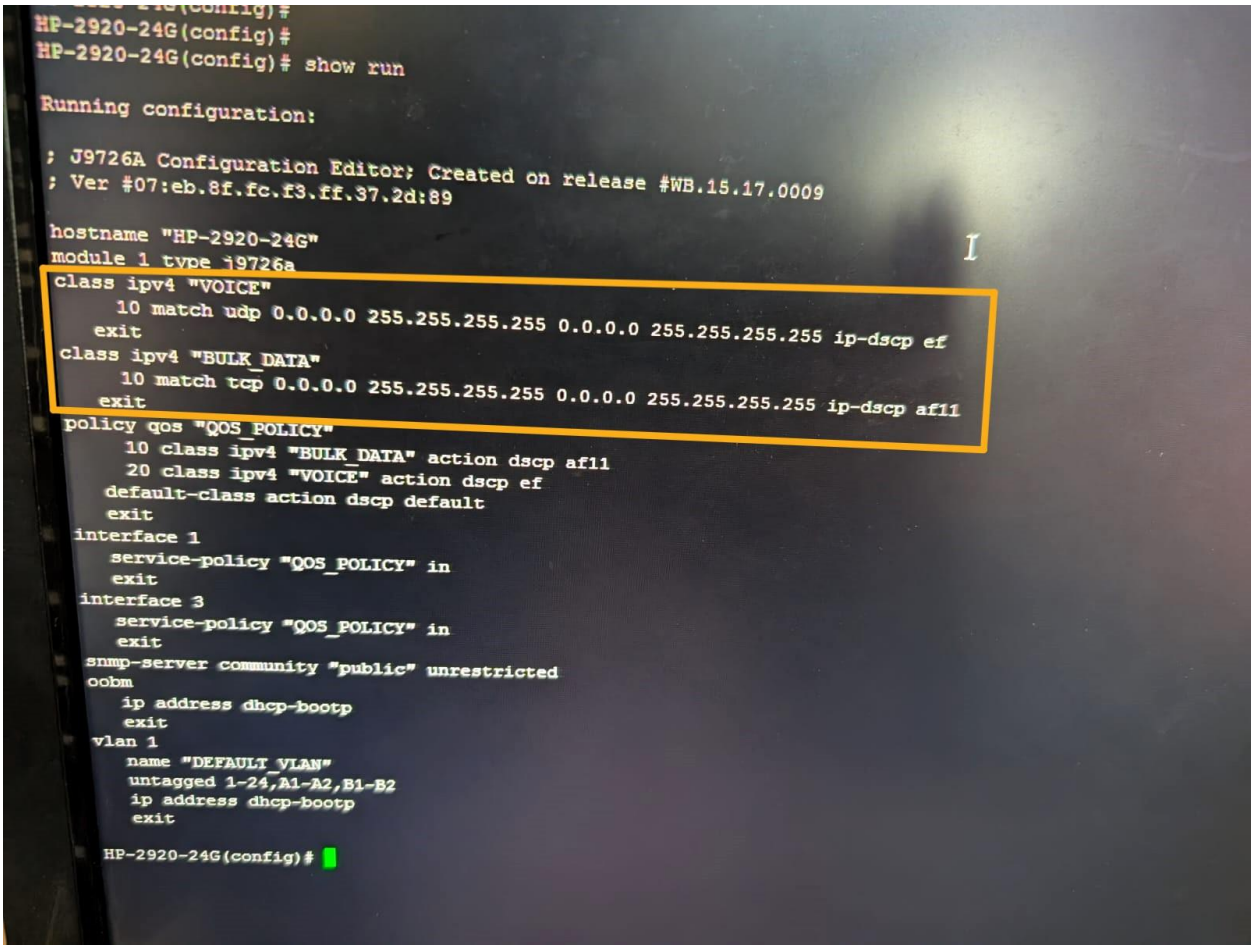
This command sends UDP traffic from PC-A to PC-B at a rate of 512 Kbps for 300 seconds, representing a typical VoIP stream. The --tos 184 option sets the ToS byte to 184, which maps to DSCP EF (Expedited Forwarding)—a class used for latency-sensitive traffic such as VoIP. Port 17000 was chosen as it falls within the standard RTP range which is commonly used for VOIP.

b. Bulk Data Traffic

```
root@LAPTOP-LAL35MFR:/mnt/c/Users/Nipul# iperf3 -c 192.168.1.2 -b 5M -t 300 --tos 40 -p 5000
Connecting to host 192.168.1.2, port 5000
[  5] local 172.26.99.113 port 53746 connected to 192.168.1.2 port 5000
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-1.00   sec   618 KBytes  5.06 Mbits/sec    0   65.0 KBytes
[  5]   1.00-2.00   sec   640 KBytes  5.24 Mbits/sec    0   93.3 KBytes
[  5]   2.00-3.00   sec   640 KBytes  5.24 Mbits/sec    0    113 KBytes
[  5]   3.00-4.00   sec   640 KBytes  5.24 Mbits/sec    0    124 KBytes
[  5]   4.00-5.00   sec   640 KBytes  5.24 Mbits/sec    0    132 KBytes
[  5]   5.00-6.00   sec   512 KBytes  4.19 Mbits/sec    0    132 KBytes
[  5]   6.00-7.00   sec   640 KBytes  5.24 Mbits/sec    0    132 KBytes
[  5]   7.00-8.00   sec   640 KBytes  5.24 Mbits/sec    0    132 KBytes
[  5]   8.00-9.00   sec   640 KBytes  5.24 Mbits/sec    0    132 KBytes
[  5]   9.00-10.00  sec   512 KBytes  4.19 Mbits/sec    0    132 KBytes
[  5]  10.00-11.00  sec   640 KBytes  5.24 Mbits/sec    0    132 KBytes
[  5]  11.00-12.00  sec   640 KBytes  5.24 Mbits/sec    0    132 KBytes
[  5]  12.00-13.00  sec   640 KBytes  5.24 Mbits/sec    0    132 KBytes
[  5]  13.00-14.00  sec   512 KBytes  4.19 Mbits/sec    0    132 KBytes
```

The above command initiates a TCP connection from PC-A to the iPerf3 server on PC-B at IP address 192.168.1.2, sending data at a rate of 5 Mbps for 300 seconds. The --tos 40 option sets the Type of Service (ToS) byte to 40, which corresponds to DSCP AF11(Assured Forwarding class 1), commonly used for bulk data transfers like FTP. The traffic is directed through port 5000.

2. Configure Traffic Classification (Router)

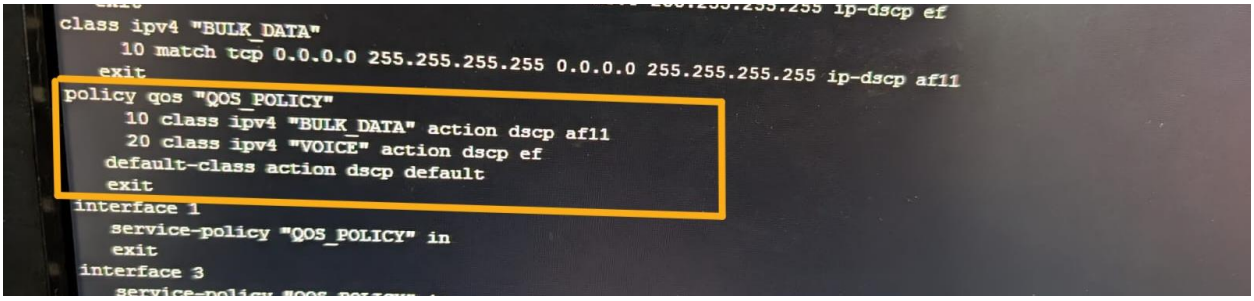


For traffic classification, we used Method A: Protocol-based Class Maps, and created two IPv4 classes on the router: "VOICE" to match all UDP traffic marked with DSCP EF (used for VoIP), and "BULK DATA" to match all TCP traffic marked with DSCP AF11 (used for FTP and similar bulk data transfers). Each class uses a wildcard mask of any-any to match all source and destination IP addresses.

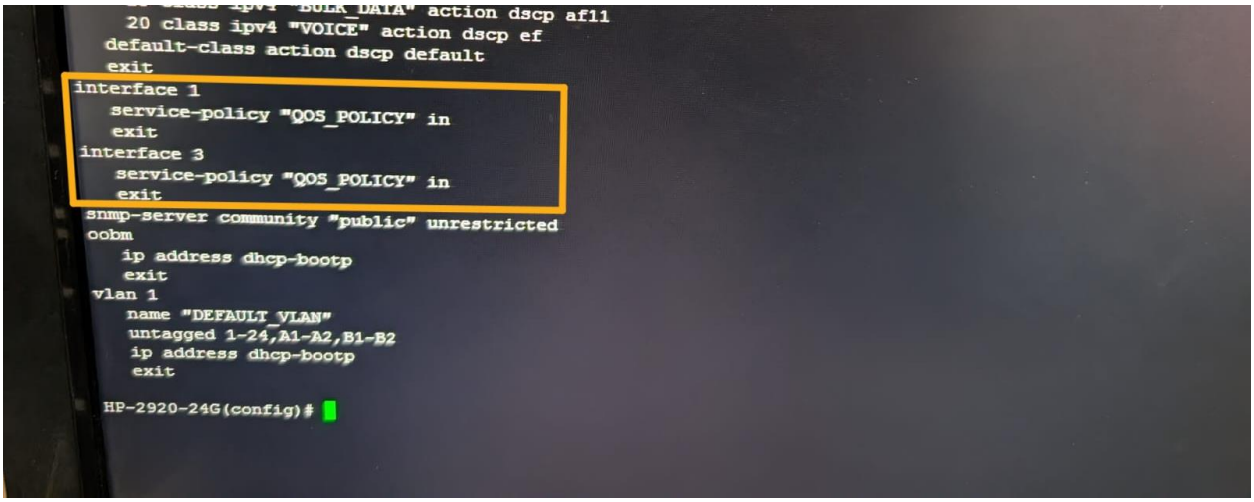
3. Create a Policy Map

To create the QoS policy, we defined a policy map named "QOS_POLICY" that binds the previously defined traffic classes to specific actions: "BULK_DATA" traffic is marked with DSCP AF11, "VOICE" traffic with DSCP EF, and all other unmatched traffic is assigned the default DSCP value.

The highlighted part of the following snapshot is our Policy Map configuration.



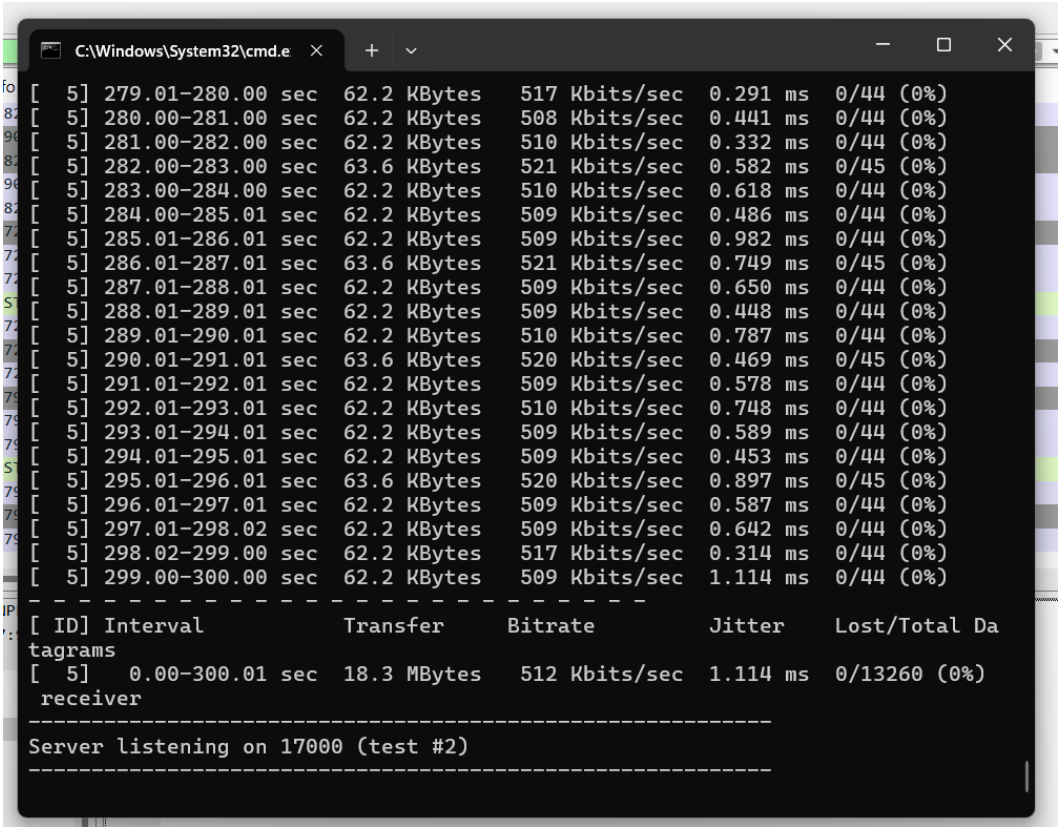
4. Apply the Policy to an Interface

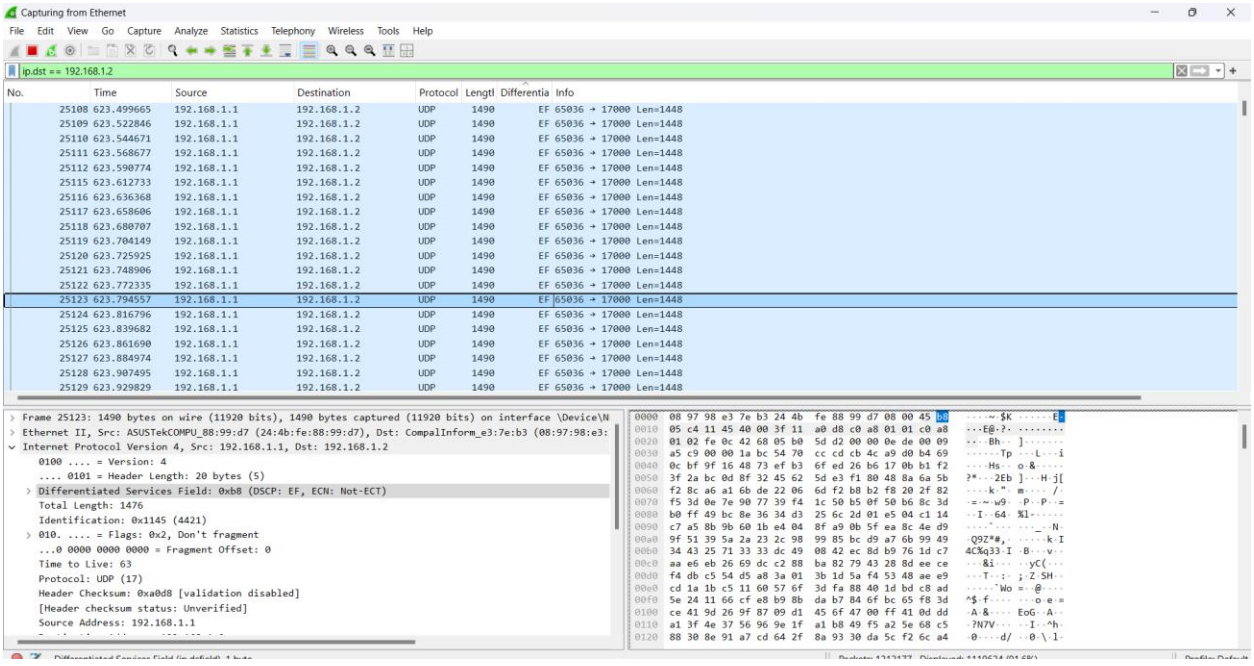


We applied the "QOS_POLICY" in the inbound direction on both interface 1 and interface 3 because the Aruba Layer 3 switch (version 15) we used only supports applying QoS policies in the inbound direction, unlike Cisco devices which typically support both inbound and outbound directions for service policies.

5. Verification and Testing

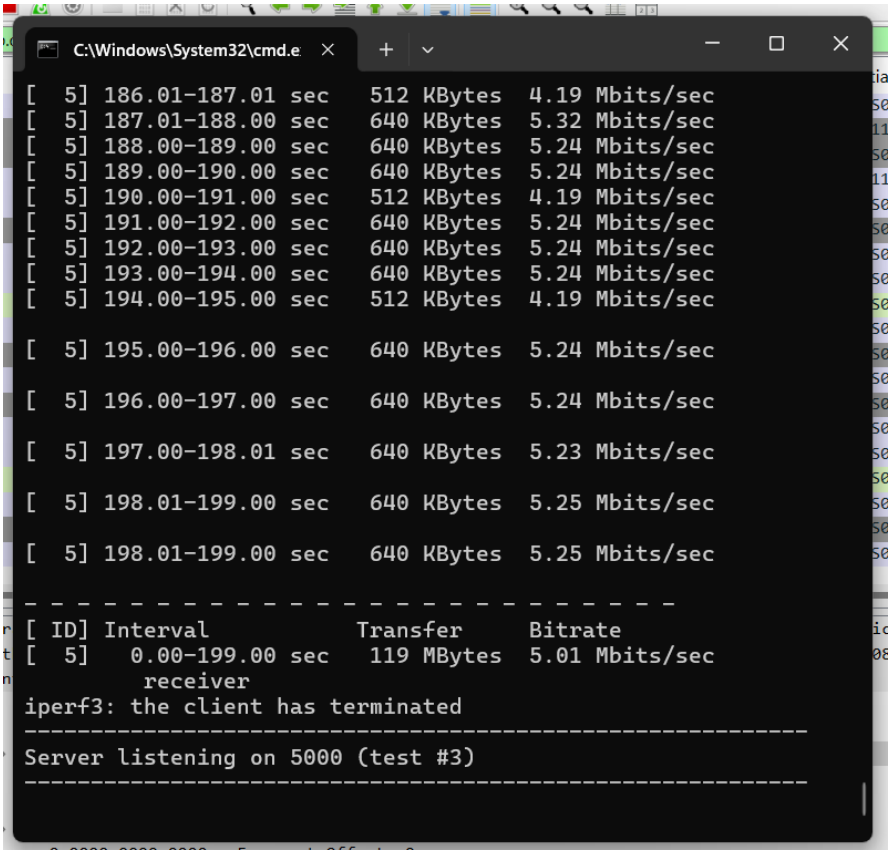
a. Verification of the VOIP traffic received from the server side

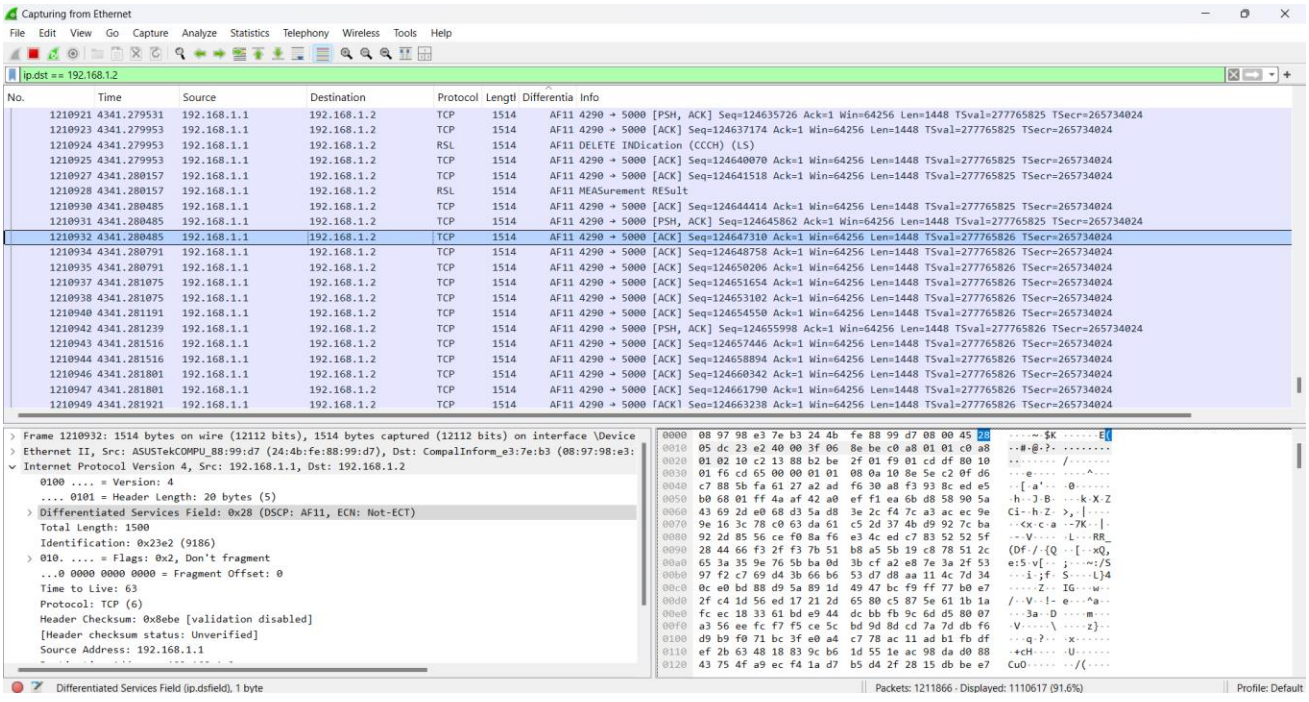




It can be observed from the above wireshark output, that the traffic marked as EF (VOIP) was successfully received by the server.

b. Verification of the BULK DATA traffic received from the server side

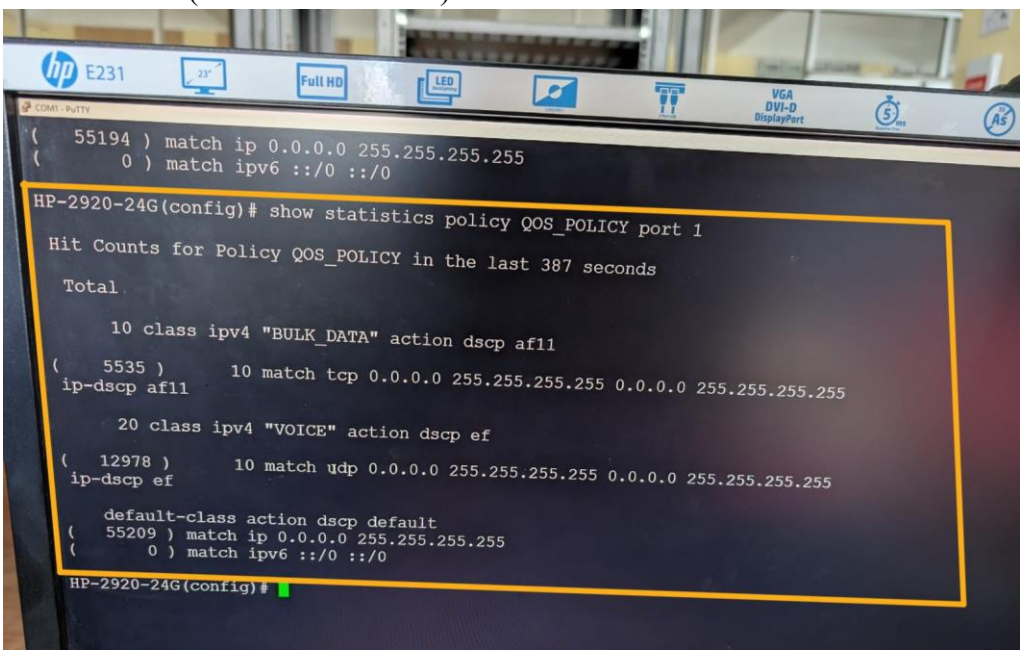




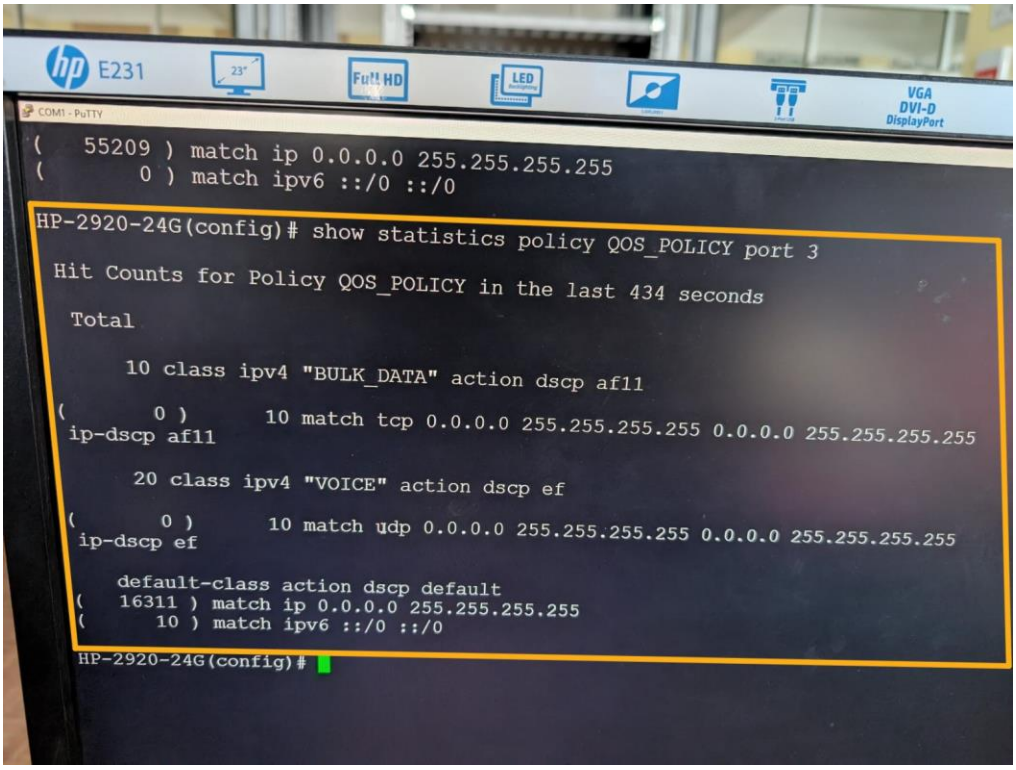
It can be observed from the above wireshark output, that the traffic marked as AF11 (BULK_DATA) was successfully received by the server.

c. Overall Verification through the Router using live statistics

At interface 1 (connected to client)



At interface 3 (connected to the server)

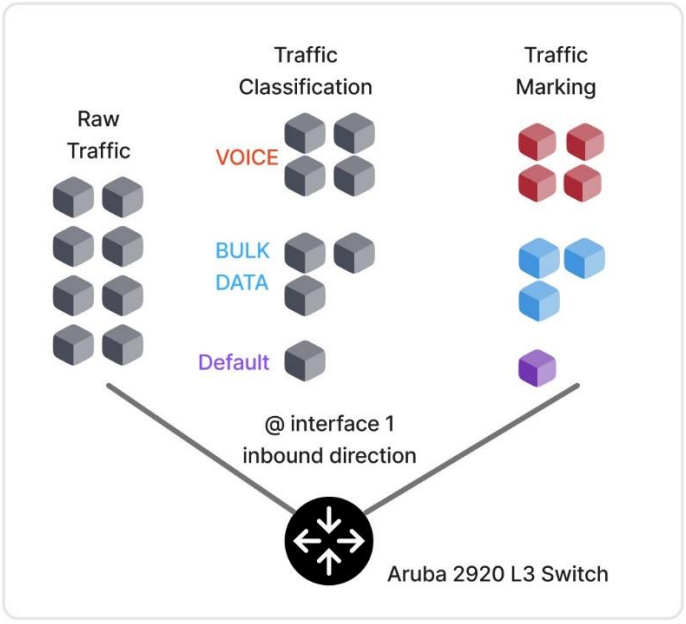


The above output confirms that QoS classification and marking occurred successfully on the client-facing interface (Interface 1). However, no matching was observed on the server-facing interface (Interface 3), because QoS policies were applied in the inbound direction only, as Aruba HP switches do not support applying QoS on outbound interfaces.

Reflection Questions:

1. How does traffic classification differ from traffic marking?

Traffic classification is the process of identifying and categorizing packets based on attributes like protocol, port, IP or DSCP value. In contrast to that traffic marking is assigning specific QoS values (like DSCP) to those classified packets. The following figure depicts the high level behavior of the QoS classification and marking that was observed in this lab experiment.



2. Why is classification a prerequisite before applying other QoS mechanisms like queuing or policing?

Classification is essential because it tells the router how to identify different types of traffic. Without first identifying the traffic, applying mechanisms like queuing, policing, or prioritization would not make sense and would be meaningless. Even if these mechanisms were somehow applied blindly to individual packets without classification, it would lead to inefficient resource usage and inconsistent treatment of traffic, ultimately defeating the purpose of QoS.

3. How are RTP or VoIP packets identified in a live network?

The Aruba L3 switch used in this lab had the capability to identify RTP or VoIP packets using two main methods: by matching UDP port ranges (specifically the range 16384–32767, commonly used by RTP for VoIP traffic) and by inspecting the DSCP value, such as EF (Expedited Forwarding), which is typically used to mark real-time voice traffic. However, in our configuration, we used DSCP-based matching to classify the traffic, relying on previously marked packets to apply QoS policies.

4. What challenges arise when classifying traffic that is encrypted (e.g., HTTPS or VPN)? How might network devices handle such cases?

In the context of encrypted traffic such as HTTPS or VPN, matching using DSCP values is often not possible because encryption may hide or strip QoS markings, or intermediate devices may overwrite them. As a result, network devices cannot rely solely on DSCP for classification and must instead use alternative methods like matching based on port numbers, or employ deep packet inspection (DPI) if available.