Exploring Traffic Management in DiffServ Domains: QoS Scheduling, Bandwidth Sharing, and Performance Metrics

Suvendu Barai*
Communication Systems and Networks
Technische Hochschule Köln
Köln, Germany
suvendu.barai@smail.th-koeln.de

Felix Hendrik Josef Hagenbrock*
Technical Computer Science
Technische Hochschule Köln
Köln, Germany
felix_hendrik_josef.hagenbrock@smail.th-koeln.de

***Abstract*—In this paper, we explore the implementation and analysis of Differentiated Services (DiffServ) in a multi-service enterprise network with a focus on Quality of Service (QoS) mechanisms. Through a series of lab-based milestones, we address critical aspects of traffic prioritization, bandwidth allocation, and congestion control for Voice over IP (VoIP) services and best-effort traffic. Key tasks include setting up the networks, the creation and validation of Access Control Lists (ACLs) for SIP, RTP, and iPerf traffic, the configuration of Low-Latency Queuing (LLQ) for traffic management and the implementation of policing and shaping mechanisms. Real-time performance metrics, such as throughput and latency, are analyzed using tools like Wireshark and iPerf to evaluate the effectiveness of the QoS configurations. The results demonstrate significant improvements in traffic handling and user experience, particularly for VoIP traffic under constrained bandwidth conditions, highlighting the importance of QoS in modern enterprise networks.**

***Index Terms*—Quality of Service (QoS), Differentiated Services (DiffServ), Voice over IP (VoIP), Traffic Prioritization, Bandwidth Allocation, Access Control Lists (ACLs), Low-Latency Queuing (LLQ), Network Performance Analysis**

I. Introduction

The increasing demand for real-time communication services such as Voice over IP (VoIP) and video conferencing has highlighted the critical need for effective Quality of Service (QoS) mechanisms in modern networks [1], [2], [3]. As network traffic grows in volume and complexity, ensuring reliable and efficient delivery of delay-sensitive traffic alongside best-effort traffic presents a significant challenge [4], [5], [6], [7]. Differentiated Services (DiffServ) provides a scalable and practical approach to address these challenges by enabling traffic classification and prioritization based on application requirements [8], [9].

In this paper, we explore the implementation of DiffServ mechanisms in a simulated network environment to optimize traffic management and ensure QoS for VoIP services [10], [11], [12]. By leveraging traffic classification techniques, Access Control Lists (ACLs), the study investigates the prioritization of VoIP traffic over less time-sensitive data, such as iPerf-generated UDP traffic [13], [14], [15]. The lab tasks were designed to replicate real-world network scenarios, emphasizing the effective allocation of bandwidth.

The experiments included the configuration of class maps, policy maps, and ingress and egress traffic policies to implement low-latency queuing (LLQ) for VoIP traffic with priority over best-effort traffic [16], [17], [18], [19], [20]. By analyzing the interaction between VoIP and background traffic under constrained bandwidth conditions, this work demonstrates the importance of applying traffic management principles in ensuring the quality of real-time communications [21], [22].

This study provides valuable insights into the technical implementation of DiffServ and its impact on network performance, focusing on achieving a balance between resource allocation and service quality [23], [24], [25].

II. Objectives

The objective of this study is to document the configuration, implementation, and analysis of a VoIP-enabled enterprise network with QoS capabilities. The lab work is structured into three progressive tasks to explore various aspects of VoIP communication, traffic analysis, and QoS optimization.

- **Task 1: Building a LAN with Internet Access and VoIP Service**
  - Set up and configure a basic VoIP network using an Asterisk server. This will be called LAN-A.
  - Assign and verify IP addresses, subnet masks, and gateways for LAN devices to ensure connectivity.
  - Test and validate SIP registration, RTP traffic flow, and VoIP communication between hardphones.

- **Task 2: Network Extension and Best-Effort (BE) Traffic Analysis**
  - Extend the network to include LAN-B and LAN-C, connected via a serial link emulating a low-bandwidth WAN.
  - Configure static routing and implement IP subnet addressing for the extended topology.
  - Analyze the interaction of VoIP and best-effort traffic using iPerf for load generation and Wireshark for packet capture, measuring throughput and bandwidth sharing behavior whilst having constrained bandwith on the serial link and without prioritization of VoIP traffic.
- **Task 3: DiffServ Domain Implementation and QoS Traffic Analysis**
  - Define service classes and implement DiffServ-based QoS policies using DSCP markings for VoIP and best-effort traffic.
  - Apply class maps, policy maps, and ACLs to prioritize VoIP traffic over best-effort traffic.
  - Evaluate QoE (Quality of Experience) for VoIP calls under varying network load conditions, ensuring minimal latency and guaranteed bandwidth for voice traffic.
  - Analyze the interaction between VoIP and best-effort traffic again with implemented prioritization.

III. Configuration Details

A. Network Topology Overview

The lab setup consisted of a DiffServ-enabled VoIP enterprise network designed with three interconnected local subnets (LAN A, LAN B, and LAN C). The topology utilized the following devices and configurations:

- Routers (R1, R2): Configured with DiffServ QoS mechanisms, ACLs, and policy maps for traffic classification and prioritization.
- Switches (S1, S2): Used to connect endpoints within each LAN.
- **Asterisk Server**: Acted as a SIP proxy and server for call setup, codec negotiation, and media relay between endpoints.
- VoIP Phones: Deployed for SIP-based voice communication.
- Traffic Generators (iPerf): Used for simulating best-effort traffic.
- Monitor PC: Deployed for capturing and analyzing traffic with Wireshark.

1) LAN A Subnet Addressing:

- LAN A uses the private IP address space `10.6.0.0/24`.
- IP addresses for LAN A devices:
- Router R1's outside interface (`g0/0/0`) uses a public IP address:

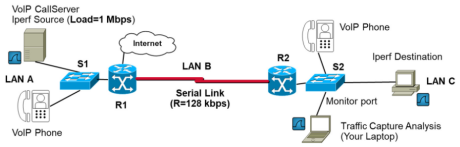

Fig. 1. Topology

Device	Subnet Address	Subnet Mask
Default Gateway	`10.6.0.1`	`255.255.255.0`
Asterisk/Iperf Server	`10.6.0.5`	`255.255.255.0`
VoIP Phone	`10.6.0.4`	`255.255.255.0`

TABLE I
LAN A Device Subnet Addresses

Interface	Subnet Address	Subnet Mask
Router R1 Outside	`139.6.19.33`	`255.255.255.224`

TABLE II
Router R1 Public IP Address

2) LAN B Subnet Addressing:

- LAN B uses the private IP address space `10.6.1.0/24`.
- Serial interface addresses:

LAN B	Interface No.	IP Address	Subnet Mask
R1 Serial Interface	`s0/1/0`	`10.6.1.1`	`255.255.255.0`
R2 Serial Interface	`s0/1/0`	`10.6.1.2`	`255.255.255.0`

TABLE III
LAN B Serial Interface Addresses

3) LAN C Subnet Addressing:

- LAN C uses the private IP address space `10.6.2.0/24`.
- Device IP addresses in LAN C:

Device	Subnet Address	Subnet Mask
Default Gateway (R2)	`10.6.2.1`	`255.255.255.0`
VoIP Phone	`10.6.2.2`	`255.255.255.0`
Iperf Destination	`10.6.2.3`	`255.255.255.0`

TABLE IV
LAN C Device Subnet Addresses

B. DiffServ Domain Implementation

1) Service Definition: The network shall run 2 network services:

- VoIP (or voice and video) including signaling traffic and media data (sample) traffic. The VoIP service uses a PCM codec for sampling and SIP signaling. The VoIP service shall become expedited forwarding (EF) service.
- Best effort (BE) service for any other traffic.

```
[general]
context=default
allowguest=0
allow=all
videosupport=yes

[6001]
type=friend
context=local
host=dynamic
secret=6001
allow=all
allow=h264
canreinvite=no
videosupport=yes
video=all
directmedia=no

[6002]
type=friend
context=local
host=dynamic
secret=6002
allow=all
allow=h264
canreinvite=no
videosupport=yes
video=all
directmedia=no
```

Fig. 2. sip.conf for Asterisk

```
[general]
static=yes
writeprotect=no

[local]
exten => 6001,1,Answer()              ; Answer the call
exten => 6001,2,Set(VIDEO_CALL=1)       ; Explicitly enable video
exten => 6001,n,Dial(SIP/6001)        ; Dial SIP extension 6001
exten => 6001,1,Hangup()              ; Hang up after the call

exten => 6002,1,Answer()              ; Answer the call
exten => 6002,2,Set(VIDEO_CALL=1)       ; Explicitly enable video
exten => 6002,n,Dial(SIP/6002)        ; Dial SIP extension 6002
exten => 6002,1,Hangup()              ; Hang up after the call
```

Fig. 3. extensions.conf for Asterisk

2) Traffic Matching Using ACLs:

- **ACLs:**

R1 ACL:
access-list 101 permit udp any any range 10000 32767
access-list 102 permit udp any any eq 5060
access-list 103 permit udp any any eq 5001

R2 ACL:
access-list 101 permit udp any any range 10000 32767
access-list 102 permit udp any any eq 5060
access-list 103 permit udp any any eq 5001

3) Class Maps:

R1 Class Maps:
class-map match-any Premium
match access-group 101
match access-group 102
class-map match-any Default
match access-group 103
R2 Class Maps:
class-map match-any Premium
match access-group 101
match access-group 102
class-map match-any Default
match access-group 103

4) Policy Maps:

R1 Policy Maps:
policy-map DSCPVALUES
class Premium
set ip dscp 46
class Default
set ip dscp 0

policy-map VOIP
class Premium
priority 100

R2 Policy Maps:
policy-map DSCPVALUES
class Premium
set ip dscp 46
class Default
set ip dscp 0

policy-map VOIP
class Premium
priority 100

5) Ingress and Egress Policy Applications:

R1 Serial Interface (s0/1/0):
interface s0/1/0
service-policy input DSCPVALUES
service-policy output VOIP
R2 Serial Interface (s0/1/0):
interface s0/1/0
service-policy input DSCPVALUES
service-policy output VOIP

C. QoS Scheduling

Service	Bandwidth (kbps)
VoIP RTP bandwidth	90
VoIP signaling bandwidth	9
Total required bandwidth	100

TABLE V
BANDWIDTH ALLOCATION FOR VOIP SERVICES

The final policy ensured that VoIP traffic was dequeued with low latency using Low-Latency Queuing (LLQ).

D. Traffic Monitoring and Analysis

1) Wireshark Filters:

- RTP and SIP Traffic: `udp.port == 5060 || rtp`

- iPerf Traffic: `udp.port == 5001`

Because the VoIP service uses UDP encapsulation, we generated UDP traffic in parallel to observe the bandwidth sharing behavior in a best effort (BE) scenario. The following steps were performed:

- The serial link bandwidth was set to **128 kbps**.
- A **load of 1 Mbps UDP traffic** was generated using the traffic load generator for a **duration of 10 seconds**.

The command used to generate the traffic was:

```
iperf -c 10.6.2.3 -u -b 1M -t
10
```

This command specifies:

- `-c 10.6.2.3`: The IP address of the destination.
- `-u`: Use UDP protocol for the traffic.
- `-b 1M`: Bandwidth for the generated traffic (1 Mbps).
- `-t 10`: Duration of the traffic generation (10 seconds).

2) Observed Data Link Layer (DLL) Throughput: The QoS implementation provides priority for VoIP traffic.

Link Bandwidth (kbps)	VoIP (kbps)	Load Traffic (kbps)
128	80 - 90 (oscillating)	40 - 50 (oscillating)

TABLE VI
OBSERVED THROUGHPUT OF VOIP AND LOAD TRAFFIC

IV. EVALUATION RESULTS

A. Milestone 1

a) Activities::

- Configured IP addresses, subnet masks, and default gateways for devices in LAN A (10.6.0.0/24).
- Connected two VoIP hardphones within the LAN A network.
- Registered the hardphones with the Asterisk server and performed call setup between the devices.

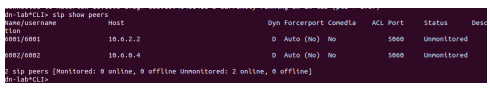

Fig. 4. Phones Registered in Asterisk

b) Evaluation Metrics::

- Correctness of IP addressing.
- Successful registration of hardphones with the Asterisk server.
- Establishment of call between the hardphones.

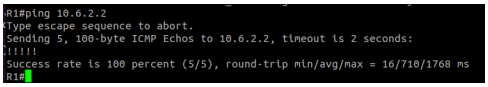

Fig. 5. Ping from Lan A to Lan C

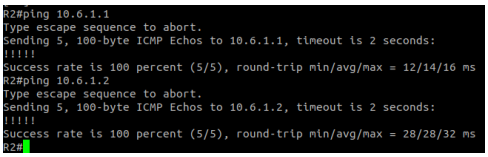

Fig. 6. Ping for Serial Interfaces Between Both Routers

c) Results::

- All devices were successfully configured with IP addresses and verified for connectivity using `ping`.
- VoIP hardphones were successfully registered with the Asterisk server.
- Calls between the hardphones were established successfully with clear audio quality.

B. Milestone 2: Monitoring and Analyzing VoIP Traffic

Activities:

- **Setting up LAN-B and LAN-C:**
  - Establish another network LAN-C with its own hardphone and iPerf Source configured similar to LAN-A
  - Establish network LAN-B connecting LAN-A and LAN-C via serial link configured to 128 kbps bandwith (changed to lower bandwiths in some measurements)

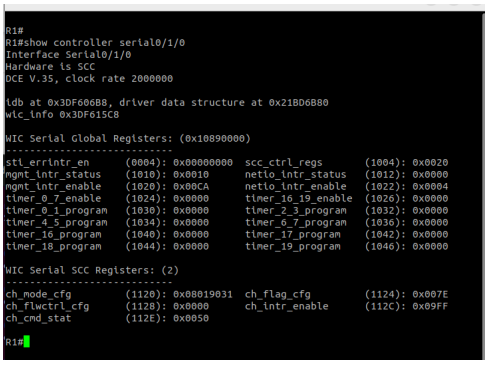

Fig. 7. Setting up Clock Rate in R1 DCE interface

- **Configuration of Monitor Port:**
  - Configured SPAN port Fa0/24 on Switch S2 to mirror traffic for monitoring and analysis.
  - Connected a monitoring PC to the SPAN port and verified traffic capture for VoIP RTP and SIP signaling using Wireshark.
  - Ensured the Monitor PC captured both transmitted and received traffic from LAN A to LAN C (10.6.0.0 to 10.6.2.0).

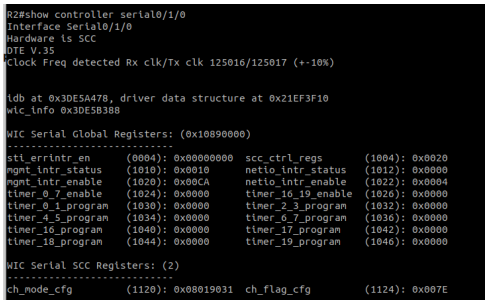

Fig. 8. R2 DTE interface

- **Traffic Capture and Analysis:**
  - Captured SIP signaling data to inspect call setup procedures, including registration, authentication mechanisms, and Session Description Protocol (SDP) exchanges.
  - Captured and analyzed RTP streams to evaluate media transmission quality under different bandwidth conditions.
  - Filtered VoIP and iPerf data in Wireshark for selective protocol analysis using advanced filtering rules (`rtp | sip` for VoIP traffic and `udp.port == 5001` for iPerf traffic).
- **Bandwidth Configuration and QoS Behavior:**
  - Generated a load of 1 Mbps using iPerf to simulate a Best Effort (BE) traffic scenario alongside VoIP traffic with configured 128 kbps bandwith.

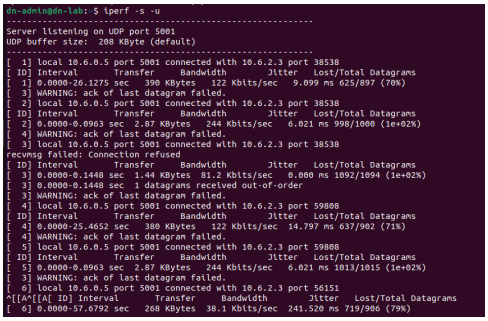

Fig. 9. Iperf Load Generation Receiver Test

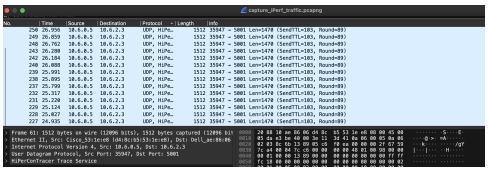

Fig. 10. Captured Iperf Traffic with Wireshark

  - Observed bandwidth sharing between iPerf and VoIP traffic under constrained bandwidth conditions.
- **Codec Selection and QoE Evaluation:**
  - Selected PCMA (G.711) and G726/32 codecs for comparison of throughput and Quality of Experience (QoE).
  - Measured application layer throughput and data link layer throughput for both codecs.
  - Adjusted serial link bandwidth to observe the effects of different bandwidth allocations (32, 64, 128 kbps) on QoE and packet delivery.

Evaluation Metrics:

- **Traffic Capture Effectiveness:** Verified accurate capture of SIP and RTP streams across LAN A and LAN C using Wireshark.
- **Analysis of Call Signaling:** Assessed the effectiveness of SIP signaling processes such as registration, call setup, and codec negotiation.
- **Measurement of Throughput:**
  - Application Layer Throughput: Measured 64 kbps for PCMA and 32 kbps for G726/32.
  - Data Link Layer Throughput: Identified a discrepancy due to protocol overheads (UDP, IP, RTP, Ethernet), resulting in an average throughput of ∼85 kbps for PCMA.
- **Bandwidth Sharing Efficiency:** Observed fair distribution of bandwidth based on packet ratios, with VoIP traffic consuming ∼7.5% and iPerf traffic ∼92.5% of the overall bandwidth.

Results:

- **Monitor Port Configuration:** Successfully configured SPAN port on S2 and validated traffic capture using the Monitor PC without data loss or packet duplication.
- **Traffic Analysis:**
  - SIP registration processes captured accurately, including detailed authentication and SDP parameter exchanges.
  - RTP streams monitored showed consistent throughput values aligned with codec specifications.
- **QoS Behavior:**
  - Since there are no QoS Mechanisms implemented yet, VoIP traffic is treated the same as Best Effort
  - VoIP traffic throughput ranged between 9.4 kbps and 10 kbps, while iPerf traffic fluctuated between 119 kbps and 125 kbps. (Overall 128 kbps bandwith)
- **Quality of Experience (QoE):**
  - PCMA provided the best voice quality with minimal distortion or delay (10/10). On lower bandwiths 64 kbps and 32 kbps the voice quality deteriorated quickly and delay rose up
  - G726/32 demonstrated acceptable quality under higher bandwidth but exhibited delays at 32 kbps (6/10). Had a higher voice quality than PCMA on 64 kbps.

Challenges:

- **SPAN Port Configuration Issues:** The initial configuration of the SPAN port for traffic monitoring led to packet duplication errors. This was due to misconfigured source and destination ports on Switch S2, resulting in overlapping traffic being mirrored to the MonitorPC. This required reassigning the SPAN source and destination ports and verifying the configurations to ensure accurate traffic captures.
- **Incorrect iPerf Traffic Direction:** During the iPerf bandwidth sharing analysis, the UDP traffic was inadvertently generated in the wrong direction (LAN C to A). This caused inaccurate measurements of the bandwidth sharing behavior between iPerf and VoIP traffic. This was fixed for the final measurements.
- **Uplink Interference:** Forgetting to disable the uplink to the Internet during QoS measurements introduced external traffic, leading to inaccurate bandwidth and delay results. This necessitated re-running experiments under controlled network conditions.
- **Codec Selection Constraints:** Although PCMA (G.711 a-law) and PCMU (G.711 μ-law) have identical throughput values, PCMA had to be paired with PCMU for testing due to because the hardphones were not compatible with other codecs like G.722, iLBC or Speex.
- **Bandwidth Congestion During Dual Traffic Testing:** When generating simultaneous VoIP and iPerf traffic, network congestion became evident, leading to inconsistent throughput values. Adjustments were required to stabilize the 128 kbps serial link bandwidth and accurately capture traffic flow dynamics for both best-effort and prioritized services.
- **QoE Analysis Complexity:** During the Quality of Experience (QoE) evaluation, accurately assessing the perceptual delay and distortion in VoIP calls under different bandwidth conditions proved challenging. Subjective factors, such as user perception of delay and voice clarity, required meticulous documentation and iterative testing to validate QoE metrics.

Milestone 3: DiffServ Domain Implementation and QoS Traffic Analysis

Activities:

- Configured the network with DiffServ to differentiate and prioritize VoIP and best-effort traffic. Assigned DSCP (Differentiated Services Code Point) values of EF (Expedited Forwarding) to VoIP traffic and BE (Best Effort) to other traffic.
- Defined access control lists (ACLs) to match traffic flows based on IP addresses, source/destination ports, and protocols for VoIP (SIP and RTP) and iPerf traffic.
- Created class-maps and policy-maps on routers R1 and R2 to apply DiffServ configurations, ensuring proper DSCP tagging and prioritization.
- Implemented QoS scheduling by guaranteeing a bandwidth of 100 kbps for VoIP traffic (90 kbps for RTP and 10 kbps for SIP).
- Conducted traffic analysis using Wireshark to capture throughput and packet-level insights for VoIP and iPerf traffic.
- Analyzed the effects of traffic prioritization by monitoring throughput and delay under concurrent traffic conditions.

Evaluation Metrics:

- Effectiveness of DiffServ-based prioritization for VoIP traffic.
- Bandwidth allocation and utilization for VoIP and iPerf traffic.
- Measured throughput and delay for VoIP traffic under prioritized scenarios.
- Overall network performance and Quality of Experience under mixed traffic conditions.

Results:

- Successfully applied DiffServ configurations on both R1 and R2. Traffic flows were differentiated and prioritized correctly based on DSCP values.
- Under concurrent load VoIP traffic (DSCP EF) received guaranteed bandwidth of 80–90 kbps, with reduced delay. iPerf traffic (DSCP BE) observed oscillating bandwidth between 40–50 kbps.
- Through Wireshark analysis, VoIP traffic exhibited consistent and smooth performance even under load conditions. Best-effort traffic experienced fluctuations due to lower priority.
- QoE (Quality of Experience) for VoIP calls improved significantly with perceptual clarity and reduced latency compared to scenarios without DiffServ prioritization.

Challenges:

- **RTP and Iperf Source Port Ranges:** The RTP source ports were dynamically assigned in the range of 10000–13000, differing from the typical expected range, which made creating accurate ACLs for traffic matching complex. Similarly, Iperf's source port was inconsistent, further complicating traffic policies required for QoS prioritization.
- **SIP and QoE Issues in Wireshark:** SIP packets were initially not visible in Wireshark during calls from LAN A to LAN C, making it difficult to diagnose call setup issues. Additionally, the participant in LAN C faced degraded audio quality

and delays compared to LAN A, due to a mistake in a policy-map in the DiffServ domain.

- **Ingress and Egress Configuration:** Misunderstanding the distinction between ingress and egress interfaces initially led to policy maps being applied incorrectly. This resulted in misclassification of traffic until interface roles were clarified.
- **Erroneous Statement:** Initially, we configured all class-maps with the match-all statement, causing issues as traffic had to match all criteria simultaneously, resulting in misclassification. Changing the configuration to match-any resolved the issue by allowing traffic to match any one of the specified criteria, ensuring proper classification.

V. Discussion

A. MS1

Question1: Why is it important to disable DNS lookup on a router when configuring it?
Disabling DNS lookup on a router prevents the router from trying to resolve mistyped or incorrect command entries as domain names, which can cause delays and unnecessary DNS queries, thereby simplifying and speeding up the command-line interface operations.

Question2: What is the reason for canreinvite=yes setting in Asterisk's SIP configuration (e.g., pjsip.conf or sip.conf)
The canreinvite=yes (or its equivalent directmedia=yes in newer versions of Asterisk) setting in Asterisk's SIP configuration allows the server to enable direct media path between two endpoints in a SIP call after the call setup is complete. This means that once the call is established, Asterisk instructs the two endpoints (e.g., SIP phones or clients) to send their audio (RTP) streams directly to each other, bypassing Asterisk itself.

Question3: Why Asterisk is needed in signaling (SIP) during the call establishment phase?

- **Call Setup and Routing**
- Asterisk acts as a SIP proxy and server, managing the `INVITE`, `ACK`, and other SIP messages during call establishment.
- It identifies and routes calls between endpoints, especially in cases where endpoints are on different networks or subnets.
- **Codec Negotiation**
- Asterisk helps negotiate codecs between endpoints by acting as an intermediary, ensuring both devices agree on a compatible codec for media transmission.
- **NAT Traversal**
- In networks where endpoints are behind NAT, Asterisk ensures the SIP signaling and media paths are correctly routed through the NAT.
- It uses its public IP and RTP port ranges to facilitate communication between endpoints that cannot directly "see" each other.

Question4: How Asterisk helps in finding RTP ports in hard phones?

- **Port Negotiation During Call Setup:** During the SIP handshake (e.g., in the `SIP INVITE` and `200 OK` messages), Asterisk identifies and exchanges the `RTP` port numbers that endpoints (hard phones) intend to use for media (audio or video) streams. Hard phones include their intended `RTP` port numbers in the `SDP (Session Description Protocol)` portion of the SIP messages, and Asterisk coordinates this exchange.
- **RTP Port Allocation:** If `canreinvite=no` (or `directmedia=no`), Asterisk remains in the media path and allocates an `RTP` port from its configured range (e.g., `10000-20000` by default in `rtp.conf`). Asterisk then relays the media between the two endpoints using these allocated `RTP` ports.

Question5: What is the meaning of the host=dynamic command in sip.conf?

- The `host=dynamic` command in `sip.conf` specifies that the SIP endpoint (e.g., a phone or client) will register with Asterisk dynamically.
- Asterisk does not rely on a static IP address for the endpoint; instead, it waits for the endpoint to send a `REGISTER` request and dynamically learns its IP address and port.

Question6: What happens if we don't mention 20 here? `exten => 3001,n,Dial(PJSIP/3001,20)`
In the Asterisk dialplan, the `Dial` application is used to connect calls to specified endpoints. The syntax is:

```
exten => 3001,n,Dial(PJSIP/3001,20)
```

Here, the second parameter `20` represents the timeout in seconds. If this timeout is omitted:

```
exten => 3001,n,Dial(PJSIP/3001)
```

the `Dial` application will wait indefinitely for the called party to answer. This means the system will continue to attempt the call until one of the following occurs:

- The called party answers.
- The caller hangs up.
- All dialed channels return a busy or error condition.

Specifying a timeout ensures that the call attempt does not persist indefinitely, allowing the dialplan to proceed to subsequent steps if the call is not answered within the designated time frame.

Question7: How session timers are handled in Asterisk? brief and precise for overleaf

- Session timers are managed through SIP INVITE and re-INVITE messages, ensuring that active sessions are maintained or terminated if no activity is detected.

- Configured using the `session-timers` setting in `sip.conf` or `pjsip.conf`, where values can be `accept`, `refuse`, or `originate`.
- Regular re-INVITEs refresh session state, and a lack of re-INVITE signals session timeout, prompting call termination.

Question8: How the NAT is working here? What's the meaning of overload here in `ip nat inside source list 1 interface g0/0/0 overload`

- NAT (Network Address Translation) translates private IP addresses in the internal network to a single public IP address for communication with external networks.
- The `overload` keyword in `ip nat inside source list 1 interface g0/0/0 overload` enables PAT (Port Address Translation), allowing multiple devices to share a single public IP address by mapping each session to a unique port number.

Question9: Why PAT is needed? How unique port numbers are assigned in PAT?

- **Why PAT is Needed:** PAT (Port Address Translation) allows multiple devices in a private network to share a single public IP address by distinguishing each device's connection through unique port numbers. This conserves public IPv4 addresses.
- **Port Assignment:** In PAT, unique port numbers are dynamically assigned by the NAT router for each session. The router maintains a translation table mapping internal private IP addresses and ports to the public IP address and unique port numbers.

Question10: Why wildcard masks are important? `access-list 1 permit 10.6.0.0 0.0.255.255`

Wildcard masks are crucial in access control lists (ACLs) as they define which bits of an IP address should be considered when applying rules. In the command `R1(config)# access-list 1 permit 10.6.0.0 0.0.255.255`, the wildcard mask `0.0.255.255` indicates that the first two octets (10.6) must match exactly, while the last two octets can vary, allowing access to all devices in the subnet `10.6.0.0/16`.

Question11: `[transport-udp] type=transport protocol=udp bind=0.0.0.0` what does it mean 0.0.0.0?

The IP address `0.0.0.0` in the `bind` directive means that the transport layer will listen on all available network interfaces of the server. This ensures that the Asterisk instance accepts incoming UDP traffic on any IP address configured on the server.

Question12: What happens if we want to use a different transport protocol rather than UDP? Like maybe TCP or TLS?

To use a different transport protocol, such as `TCP` or `TLS`, you need to configure the corresponding transport section in the `pjsip.conf` file. For example, for `TCP`:

```
[transport-tcp]
type=transport
protocol=tcp
bind=0.0.0.0
```

For `TLS`, additional settings are required for encryption, such as specifying the `cert file` and `priv key file`:

```
[transport-tls]
type=transport
protocol=tls
bind=0.0.0.0
cert_file=/path/to/cert.pem
priv_key_file=/path/to/key.pem
```

This allows Asterisk to handle SIP traffic using the specified protocol.

B. MS2

Question1: What are the reasons a V.35 cable (or similar high-speed serial cables) is used for connecting two routers, instead of a standard Ethernet cable typically used for connecting a router to a switch?

- **Wide Area Network (WAN) Connectivity:** V.35 cables are specifically designed for high-speed serial connections over WAN links, such as leased lines or Frame Relay, where Ethernet is not natively supported.
- **Physical Layer Compatibility:** V.35 cables support synchronous communication, which is commonly used in WAN environments, while Ethernet primarily supports asynchronous communication for LAN setups.

Question2: WAN connections rely on clock synchronization between devices. What do you understand by that?

In serial communication over a WAN link, one device (usually the Data Communications Equipment, or DCE) is responsible for providing a clock signal to synchronize data transfer, clock rate 125000 or 125 Kbps, while the device at the DTE (Data Terminal Equipment) end follows it.

Question3: Why is there a difference between application layer throughput and data link layer throughput for VoIP codecs, and what overheads contribute to this discrepancy? Why the Data Link Layer throughput is slightly less in Wireshark than theoretical computation?

The difference arises due to the overhead added by protocols such as UDP, IPv4, RTP, and Ethernet. Each layer adds headers to the data, which increases the total packet size. For example, an RTP packet

Codec	Data Link Layer Throughput
PCMU	A to C = 85.6 kbps, C to A = 85.6 kbps
PCMA	A to C = 85.6 kbps, C to A = 85.6 kbps
G726/32	A to C = 53.6 kbps, C to A = 53.6 kbps

TABLE VII
DATA LINK LAYER THROUGHPUT FOR DIFFERENT CODECS IN WIRESHARK

includes 12 bytes of RTP headers, 8 bytes for UDP, 20 bytes for IPv4, and 18 bytes for Ethernet, significantly increasing the total size over the application payload.

- Total RTP Packets: 576
- Duration: 11.50 seconds
- Payload Size per Packet:
  - G.711 (PCMA) Payload Size: 160 Bytes (20 ms of audio)
- Total Bytes:

$$\text{Total Bytes} = 576 \times 160 = 92160\text{Bytes}$$

- Throughput:

$$\text{Throughput (kbps)} = \frac{\text{Total Bytes} \times 8}{\text{Duration} \times 1000}$$

$$\text{Throughput (kbps)} = \frac{92160 \times 8}{11.50 \times 1000} = 64.14\text{kbps}$$

- Data Link Layer Throughput Calculation:
  - Ethernet Header: 14 bytes
  - IP Header: 20 bytes
  - UDP Header: 8 bytes
  - RTP Header: 12 bytes
  - Frame Check Sequence: 4 bytes
  - Total Overhead per Packet:

$$14 + 20 + 8 + 12 + 4 = 58\text{Bytes}$$

- Total Frame Size:

$$\text{Total Frame Size} = \text{Payload} + \text{Overhead}$$

$$\text{Total Frame Size} = 160 + 58 = 218\text{Bytes}$$

- Total Bytes (Data Link Layer):

$$\text{Total Bytes} = \text{RTP Packets} \times \text{Total Frame Size}$$

$$\text{Total Bytes} = 576 \times 218 = 125568\text{Bytes}$$

- Data Link Layer Throughput:

$$\text{Throughput (kbps)} = \frac{\text{Total Bytes} \times 8}{\text{Duration} \times 1000}$$

$$\text{Throughput (kbps)} = \frac{125568 \times 8}{11.50 \times 1000} = 87.35\text{kbps}$$

This is the real Data Link Layer throughput theoretically. It is even a bit more higher than the previous because wireshark doesn't include extra bytes used for error checking.

Question4: Why is the data link layer throughput for G726/32 significantly lower than for PCMU and PCMA despite using the same sampling rate?

Codec	Sampling Rate	Application Layer Throughput
PCMU	8 kHz	64 kbps
G726/32	8 kHz	32 kbps
PCMA	8 kHz	64 kbps

TABLE VIII
CODEC SAMPLING RATES AND APPLICATION LAYER THROUGHPUT

The G726/32 codec has a lower bitrate (32 kbps) compared to PCMU and PCMA (64 kbps) at the application layer, leading to a reduced payload size per packet. Although the encapsulation overhead from RTP, UDP, IPv4, and Ethernet headers remains constant for all codecs, the smaller payload of G726/32 results in a higher proportion of overhead relative to the total packet size. This significantly impacts the data link layer throughput, reducing the efficiency of data transmission for G726/32 compared to PCMU and PCMA. This is evident in the data link layer throughput measurements, where G726/32 records only 53.6 kbps, while PCMU and PCMA achieve 85.6 kbps.

Question5: What factors influence QoE when using different link bandwidths for VoIP calls, and how does bandwidth impact delay and distortion?

Codec	Link Bandwidth (in kbps)	QoE Description	Rate and Ranking
PCMA	32	Voice is understandable but distorted, lower quality and high delay, almost 10 sec	2/10 (4th)
PCMA	64	The voice quality was okay like 128 kbps but the delay was a lot higher, 4 to 5 sec	5/10 (3rd)
PCMA	128	We can understand perfectly, there was no perceived delay	10/10 (1st)
G726/32	32	Voice quality was the worst, high distortion but delay was shorter than the PCMA 32	1/10 (5th)
G726/32	64	The voice quality was like 128 kbps	6/10 (2nd)
G726/32	128	We can understand but a little slow and delayed, the voice was not clear	6/10 (2nd)

TABLE IX
QOE COMPARISON FOR VOIP CODECS WITH DIFFERENT BANDWIDTHS

QoE is influenced by available bandwidth, codec selection, and network conditions. At lower bandwidths (e.g., 32 kbps), delays and distortions increase due to congestion and packet queuing.

Higher bandwidths (e.g., 128 kbps) reduce delays and improve audio clarity, as more data can be transmitted simultaneously without bottlenecks.

Question6: Why does PCMA achieve the best Quality of Experience (QoE) at 128 kbps with no perceived delay, while G726/32 achieves a similar rating at the same bandwidth but with slower and unclear voice?

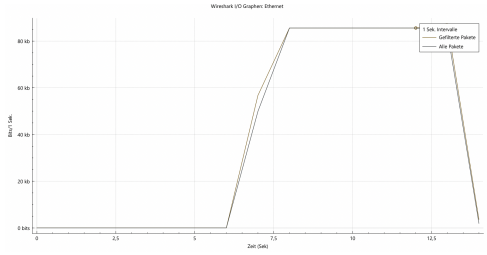

Fig. 11. I/O Graph for PCMA codec

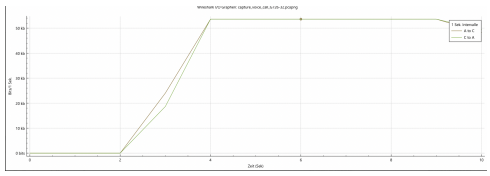

Fig. 12. I/O Graph for G726/32 codec

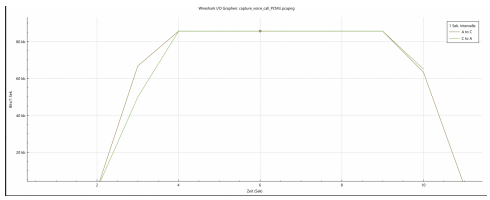

Fig. 13. I/O Graph for PCMU codec

PCMA operates as a G.711 codec with uncompressed audio, ensuring optimal voice quality and minimal latency at higher bandwidths. G726/32, being a compressed ADPCM codec, introduces slight delays and compression artifacts even at 128 kbps. The difference stems from PCMA's lossless nature compared to G726/32's lossy compression algorithm, which prioritizes bandwidth efficiency over audio fidelity.

Question7: Why does PCMA at 64 kbps exhibit higher delay compared to G726/32 at the same bandwidth, despite having better audio clarity?

PCMA requires higher processing overhead due to its uncompressed audio format, leading to increased transmission delay at constrained bandwidths like 64 kbps. Conversely, G726/32 uses a compression algorithm optimized for low-bandwidth usage, reducing transmission delays but at the cost of introducing distortions. This trade-off results in PCMA delivering clearer audio but experiencing higher delays under similar bandwidth conditions.

Question8: How does the UDP traffic generated by iPerf affect the bandwidth sharing in the best effort (BE) scenario when the serial link is limited to 128 kbps?

In a best effort (BE) scenario, the serial link's bandwidth of 128 kbps is shared proportionally based on the traffic load of each flow. UDP traffic generated by iPerf, being a high-bandwidth stream, consumes 92.5% of the available bandwidth (115.625 kbps), leaving only 7.5% (9.375 kbps) for VoIP traffic. This distribution occurs because BE service does not guarantee bandwidth and prioritizes fairness per packet, causing higher traffic flows to dominate.

Shared fairly on a per-packet basis so every traffic should get as much percentage of the bandwidth as it has of the overall traffic. This means higher traffic percentages get more bandwidth percentage, but all with no guarantees.

Calculation of Expected Ratio of Load Traffic to VoIP Traffic

1) Total Traffic Bandwidth:

$$\text{Total Bandwidth} = \text{iPerf Traffic} + \text{VoIP Traffic}$$

$$\text{Total Bandwidth} = 1081 + 87.2 = 1168.2\,\text{Kbps}$$

2) Bandwidth Percentage:

$$\text{iPerf} = \frac{1081}{1168.2} \times 100 = 92.5\%$$

$$\text{VoIP} = \frac{87.2}{1168.2} \times 100 = 7.5\%$$

3) Bandwidth Allocation on a Serial Link of 125 Kbps:

$$\text{VoIP} = 7.5\% \times 125\,\text{Kbps} = 9.375\,\text{Kbps}$$

$$\text{iPerf} = 92.5\% \times 125\,\text{Kbps} = 115.625\,\text{Kbps}$$

Results:

- VoIP Traffic will use **9.375 Kbps**.
- iPerf Traffic will use **115.625 Kbps**.

Question9: Why does the measured Data Link Layer throughput for UDP traffic in Wireshark (125.7–137.8 kbps) differ from the iPerf Application Layer throughput (122 kbps)?

iPerf Load Generation (LAN A)	Wireshark Data Link Layer Throughput (LAN C, Mirror Port)	iPerf Application Layer Throughput (LAN C, Mirror Port)
1 Mbps	125.7 - 137.8 kbps	122 Kbps

TABLE X
IPERF THROUGHPUT AND WIRESHARK DATA COMPARISON

The difference arises from the overhead added by Data Link Layer encapsulation, which includes Ethernet headers, IPv4 headers, and UDP headers. These

additional bytes per packet increase the throughput measured at the Data Link Layer compared to the Application Layer. The range (125.7–137.8 kbps) also reflects variability in traffic conditions, retries, or additional signaling captured by Wireshark.

Question10: What is the significance of filtering traffic in Wireshark using `udp.port == 5001` when analyzing iPerf traffic?

Filtering with `udp.port == 5001` isolates iPerf-generated traffic by matching packets using the specific UDP port configured for the iPerf stream.

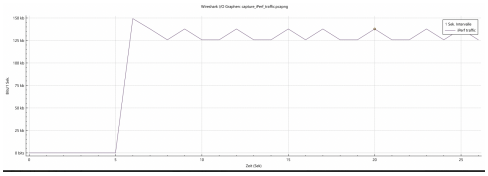

Fig. 14. Filtering Iperf traffic from I/O Graph

This ensures that only relevant traffic is captured and analyzed, preventing interference from other flows or background traffic and enabling precise throughput measurement for iPerf traffic.

Question11: How does the simultaneous generation of VoIP and iPerf traffic impact the QoS of the VoIP call, and what does the captured data reveal about priority handling?

Simultaneous traffic generation shows that VoIP traffic suffers reduced throughput (9.4 kbps) due to iPerf traffic dominating the link. Since BE service does not prioritize traffic types, VoIP packets compete equally with iPerf packets, leading to degraded voice quality and higher delay. This demonstrates the need for QoS mechanisms like DiffServ to prioritize VoIP traffic for consistent performance.

Link Bandwidth	iPerf DLL Throughput	VoIP DLL Throughput
128 kbps	ca. 119 kbps	9.4 kbps

TABLE XI
DATA LINK LAYER THROUGHPUT (DLL) FOR IPERF AND VOIP TRAFFIC

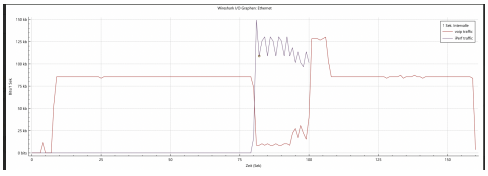

Fig. 15. Data Link Layer (DLL) Throughput for Iperf vs VoIP

Question12: What role does packet overhead (e.g., RTP, UDP, IP, and Ethernet headers) play in reducing the effective bandwidth for VoIP traffic in a Best Effort (BE) scenario?

Packet overhead, including headers from RTP, UDP, IP, and Ethernet layers, consumes a portion of the available bandwidth, reducing the effective bandwidth for payload data. For VoIP traffic, where packets are small and frequent, the relative impact of overhead is significant, as each packet carries a fixed-size header. In a BE scenario, this overhead further limits VoIP throughput, amplifying the disparity between VoIP and high-bandwidth streams like iPerf, which utilize larger packets and experience proportionally less impact from overhead.

C. MS3

Question1: What is the purpose of assigning DSCP values in the policy map for DiffServ domain implementation?

Service	Class Name	DSCP Class	DSCP Value
VOIP (SIP and RTP)	Premium	EF	46
Best Effort	Default	BE	0

TABLE XII
SERVICE CLASSES AND DSCP MAPPING

IP Flow	DSCP	Source Address	Source Port	Destination Address	Destination Port
SIP	46	10.6.0.5	5060	10.6.2.2	5060
RTP	46	10.6.0.5	16384 - 32767	10.6.2.2	50040
Iperf	0	10.6.0.5	51115	10.6.2.3	5001

TABLE XIII
TRAFFIC MATCHING TABLE FOR IP FLOWS

DSCP values are used to mark packets with specific priority levels, enabling routers to apply differentiated treatment to traffic classes like VoIP (EF) and Best Effort (BE). This ensures VoIP traffic receives higher priority for minimal delay and jitter.

Question2: Why is it necessary to use different access lists for VoIP signaling (SIP) and media traffic (RTP) in DiffServ configuration?

SIP and RTP have different traffic characteristics and port ranges. SIP uses fixed ports for signaling (e.g., 5060), while RTP utilizes dynamically assigned port ranges. Separate ACLs ensure precise traffic identification and prioritization.

Question3: How does the ”class-map match-any” command improve traffic classification compared to ”class-map match-all”?

”Match-any” allows a packet to match any one condition in the class map, making it suitable

for diverse traffic flows. ”Match-all” requires all conditions to be satisfied, which can incorrectly exclude valid traffic when multiple criteria overlap.

Question4: What is the function of the ingress and egress service policies configured on serial interfaces?

The ingress policy marks packets with DSCP values based on the defined class maps, while the egress policy ensures that priority traffic (e.g., VoIP) is queued and forwarded with minimal latency on the outgoing interface.

Question5: Why is it necessary to configure precise ACLs (Access Control Lists) for SIP, RTP, and iPerf traffic in DiffServ domains, and how do source port ranges impact their effectiveness?

Configuring precise ACLs ensures that specific traffic flows, such as SIP (port 5060), RTP (dynamic port range 16384–32767), and iPerf (variable source port), are correctly classified and matched to their respective QoS classes. The effectiveness of ACLs depends on accurately capturing dynamic port ranges (e.g., RTP) and adapting to traffic with changing source ports (e.g., iPerf). Misconfigured or overly broad ACLs can result in incorrect traffic classification, reducing QoS enforcement accuracy.

Question6: What challenges arise when monitoring traffic from a GigabitEthernet port (1 Gbps) using a FastEthernet monitoring port (100 Mbps), and how can these be mitigated?

The significant speed mismatch between the GigabitEthernet port and FastEthernet monitoring port can result in packet loss or incomplete captures, as the monitoring port cannot handle the full traffic load. This issue can be mitigated by limiting the traffic captured to specific flows using filters or employing a monitoring port with equivalent bandwidth capacity.

Question7: What is the role of service-policy commands in ingress and egress configurations, and how are they applied to enforce QoS in DiffServ networks?

The service-policy command links a policy-map to an interface, enabling the application of QoS rules. In ingress configurations, policies like service-policy input DSCPVALUES mark traffic with appropriate DSCP values, while in egress configurations, policies like service-policy output VOIP enforce prioritization and bandwidth allocation, ensuring traffic is handled per QoS requirements. These policies are essential for maintaining traffic differentiation and efficient bandwidth utilization.

Question8: How does the ”policy-map VOIP” on router R1 ensure minimum latency and guaranteed bandwidth for VoIP traffic?

The ”policy-map VOIP” prioritizes VoIP traffic using the ”priority” command, which allocates a fixed amount of bandwidth (100 kbps in this case) to VoIP packets, ensuring low latency and guaranteed throughput even under high load. This ensures VoIP traffic is not delayed or dropped due to congestion.

Question9: What challenges can arise when capturing iPerf and VoIP traffic simultaneously in Wireshark, and how does filtering help resolve these issues?

Simultaneous traffic capture may result in overlapping or redundant data, especially when analyzing high-bandwidth iPerf traffic and low-latency VoIP traffic. Applying filters like ”udp.port == 5001” or ”rtp” ensures focused analysis by isolating specific traffic flows, reducing noise and improving measurement accuracy.

Question10: In the ”QoE scheduling” step, why is 10% of the total bandwidth allocated for signaling, and how does it impact VoIP performance?

Signaling traffic, such as SIP messages, requires a separate allocation to ensure reliable call setup and control, which is critical for VoIP functionality. Allocating 10% (9 kbps) ensures signaling traffic is not affected by other flows, maintaining call stability and preventing dropped connections.

Question11: Why does the perceptual QoE improve significantly when the bandwidth for PCMA is increased from 32 kbps to 128 kbps?

At higher bandwidths (128 kbps), there is sufficient capacity for encoding and transmitting high-quality audio data without significant packet loss or jitter. This minimizes delays and distortions, resulting in a clearer and more stable audio experience for VoIP calls.

Question12: What is the significance of assigning DSCP values (e.g., EF for VoIP and BE for iPerf) in DiffServ domain implementation, and how does it affect traffic prioritization?

DSCP values classify traffic into different service classes. EF (Expedited Forwarding) ensures VoIP traffic receives high priority with minimal delay and jitter, while BE (Best Effort) allows non-critical iPerf traffic to utilize leftover bandwidth. This differentiation ensures QoS for latency-sensitive applications like VoIP.

Question13: Why does the VoIP traffic data link layer throughput oscillate between 80-90 kbps, while the iPerf load traffic oscillates between 40-50 kbps, and how does the QoS configuration prioritize VoIP traffic in this scenario?

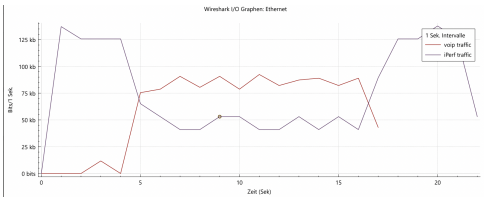

Fig. 16. I/O Graph for Iperf vs VoIP Traffic with QoS

The oscillation in throughput is caused by the dynamic allocation of the 128 kbps serial link

bandwidth, where QoS prioritizes VoIP traffic (classified under Expedited Forwarding with DSCP EF) to ensure low latency and consistent performance. VoIP traffic is allocated higher priority in the queue, receiving the majority of the bandwidth (80-90 kbps), while the remaining bandwidth (40-50 kbps) is shared with iPerf load traffic classified under Best Effort (DSCP BE). This demonstrates the effective prioritization by QoS to meet VoIP's real-time requirements.

Question14: How does Low-Latency Queuing (LLQ) ensure VoIP traffic maintains quality during network congestion, and what configuration steps enable this behavior in the observed QoS implementation?

Codec	PHB Guaranteed Bandwidth (kbps)	QoE Description
PCMA	100	We hear each other perfectly with no latency. SIP is also working perfectly which it did not before (receiving the call).

TABLE XIV
PERCEPTUAL QOE IMPRESSION

LLQ ensures VoIP traffic maintains quality by prioritizing its packets in the queuing process, minimizing delay and jitter critical for real-time communication. This is achieved by classifying VoIP packets with Access Control Lists (ACLs) and assigning them a high-priority queue in the policy-maps. The observed consistent throughput of 80-90 kbps for VoIP traffic confirms the effectiveness of this configuration, as it allows VoIP traffic to preempt lower-priority bulk traffic like iPerf during congestion.

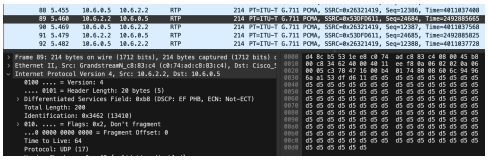

Fig. 17. DSCP values for QoS

Question15: What could cause SIP traffic to initially not appear in Wireshark captures during a VoIP call setup, and how does ensuring proper QoS configurations resolve this issue?

SIP traffic may not appear in Wireshark captures if the traffic is being filtered, misclassified, or dropped due to misconfigured ACLs or policy-maps on the routers. Additionally, NAT or firewall issues might block or obscure SIP signaling. By configuring QoS to prioritize SIP packets (e.g., using DSCP 46 for EF traffic) and ensuring ACLs explicitly permit SIP traffic on the correct ports (such as 5060 for SIP signaling), the signaling becomes properly prioritized and visible in packet captures. This also ensures the SIP messages are routed correctly, facilitating seamless call setup.

REFERENCES

[1] Cisco, "Quality of service for voice over ip," Cisco Networking White Paper, n.d., [Online]. Available: https://www.cisco.com. [Online]. Available: https://www.cisco.com

[2] M. H. Miraz, S. A. Molvi, M. A. Ganie, M. Ali, and A. H. Hussein, "Simulation and analysis of quality of service (qos) parameters of voice over ip (voip) traffic through heterogeneous networks," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 7, 2017. [Online]. Available: http://dx.doi.org/10.14569/IJACSA.2017.080732

[3] E. Kim and S.-G. Kang, "Qos support based on intserv/diffserv for sip-based applications," in *2004 IEEE/IFIP Network Operations and Management Symposium (IEEE Cat. No.04CH37507)*, vol. 2, 2004, pp. 131–144 Vol.2.

[4] ——, "Qos support based on intserv/diffserv for sip-based applications," vol. 2, 05 2004, pp. 131 – 144 Vol.2.

[5] S. Park, "Indirect diffserv qos for sip in broadband access networks," in *Agent and Multi-Agent Systems: Technologies and Applications*, N. T. Nguyen, A. Grzech, R. J. Howlett, and L. C. Jain, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 859–866.

[6] M. O. Ortega, G. C. Altamirano, and M. F. Abad, "Evaluation of the voice quality and qos in real calls using different voice over ip codecs," in *2018 IEEE Colombian Conference on Communications and Computing (COLCOM)*, 2018, pp. 1–6.

[7] M. Ortega and J. Ortega Ortega, "An assessment of voice quality and qos in real voip calls using multiple voice codecs," 03 2024, pp. 79–88.

[8] S. Daoud and Y. Qu, "A comprehensive study of dscp markings' impact on voip qos in hfc networks," *International Journal of Computer Networks Communications*, vol. 11, pp. 1–19, 09 2019.

[9] A. Lazzez and T. Slimani, "Deployment of voip technology: Qos concerns," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 2, pp. 3514–3521, 09 2013.

[10] E. Schooler, J. Rosenberg, H. Schulzrinne, A. Johnston, G. Camarillo, J. Peterson, R. Sparks, and M. J. Handley, "SIP: Session Initiation Protocol," RFC 3261, Jul. 2002. [Online]. Available: https://www.rfc-editor.org/info/rfc3261

[11] International Telecommunication Union, "Recommendation G.711: Pulse Code Modulation (PCM) of Voice Frequencies," ITU-T, Standard, 1988, (Accessed: [Insert Date]). [Online]. Available: https://www.itu.int/rec/T-REC-G.711

[12] ——, "Recommendation G.726: Adaptive Differential Pulse Code Modulation (ADPCM)," ITU-T, Standard, 1990, (Accessed: [Insert Date]). [Online]. Available: https://www.itu.int/rec/T-REC-G.726

[13] W. Zhang, Y. Chang, Y. Liu, and Y. Tian, "Perceived qos assessment for voip networks," in *2013 15th IEEE International Conference on Communication Technology*, 2013, pp. 707–711.

[14] I. Project, "Iperf: Network bandwidth measurement tool," n.d., [Online]. Available: https://iperf.fr. [Online]. Available: https://iperf.fr

[15] ITU-T, "Recommendation g.114: One-way transmission time," 2003, [Online]. Available: https://www.itu.int/rec/T-REC-G.114. [Online]. Available: https://www.itu.int/rec/T-REC-G.114

[16] J. F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach*, 8th ed. Boston, MA, USA: Pearson, 2020.

[17] F. Yihunie and E. Abdelfattah, "Simulation and analysis of quality of service (qos) of voice over ip (voip) through local area networks," 05 2019.

[18] W. Goralski, "Traffic engineering and qos for ip networks," 2008, focuses on DiffServ queuing methods such as LLQ for managing VoIP and bulk traffic.

[19] R. Mandeville, *Traffic Management for IP-Based Networks*. Artech House, 2012, discusses real-world implementation of policy maps and class maps in ensuring traffic prioritization.

[20] Z. Zhou and J. Li, "Qos-aware traffic shaping for voip traffic under bandwidth constraints," *Springer Journal of Network and Systems Management*, 2018, [Online]. Available: https://link.springer.com. [Online]. Available: https://link.springer.com

[21] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated services," *IETF RFC 2475*, 1998, [Online]. Available: https://www.rfc-editor.org/rfc/rfc2475. [Online]. Available: https://www.rfc-editor.org/rfc/rfc2475

[22] P. Ferguson and G. Huston, "Quality of service: Delivering qos on the internet and in corporate networks," *Cisco Systems White Paper*, 1998, [Online]. Available: https://www.cisco.com. [Online]. Available: https://www.cisco.com

[23] K. Malik and A. Riaz, "Performance analysis of qos in voip traffic over differentiated services networks," *International Journal of Computer Applications*, vol. 128, no. 7, pp. 25–30, 2015, [Online]. Available: https://www.ijcaonline.org. [Online]. Available: https://www.ijcaonline.org

[24] S. Salsano, "Diffserv resource management and allocation strategies in ip networks," *IEEE Communications Magazine*, vol. 38, no. 5, pp. 136–142, 2000, [Online]. Available: https://ieeexplore.ieee.org. [Online]. Available: https://ieeexplore.ieee.org

[25] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao, "Overview and principles of internet traffic engineering," *IETF RFC 3272*, 2002, [Online]. Available: https://www.rfc-editor.org/rfc/rfc3272. [Online]. Available: https://www.rfc-editor.org/rfc/rfc3272