CNP FISAC 2: Wireshark

RTP

Group Number: 3

Subject Code: ICT2226

Subject Name: CNP

Date: 05/04/2025

Names:

Yashvardhan Tomar (230953420)

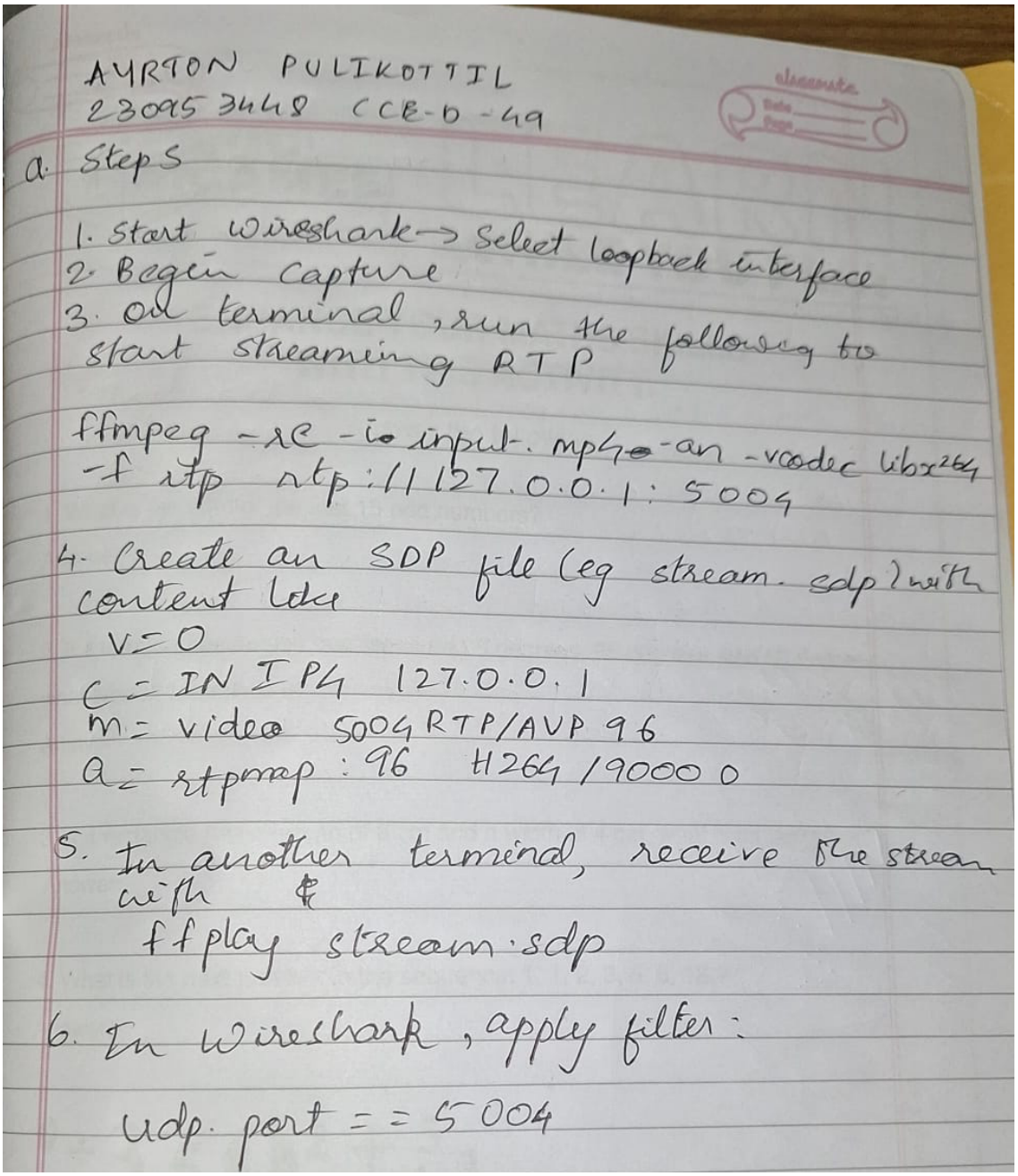
Hisham Adil (230953434)

Ayrton Joseph (230953448)

Priyansh Nandan (230953450)

Section: CCE-D

Q1) Write the Steps to generate and capture RTP packets using Wireshark.



AYRTON PULIKOTTIL
23095 3448 CCE-D-49

a. Steps

1. Start Wireshark → Select loopback interface
2. Begin capture
3. On terminal, run the following to start streaming RTP

ffmpeg -re -i input.mp4 -an -vcodec libx264 -f rtp rtp://127.0.0.1:5004

4. Create an SDP file (eg stream.sdp) with content like

v=0
c=IN IP4 127.0.0.1
m=video 5004 RTP/AVP 96
a=rtpmap:96 H264/90000

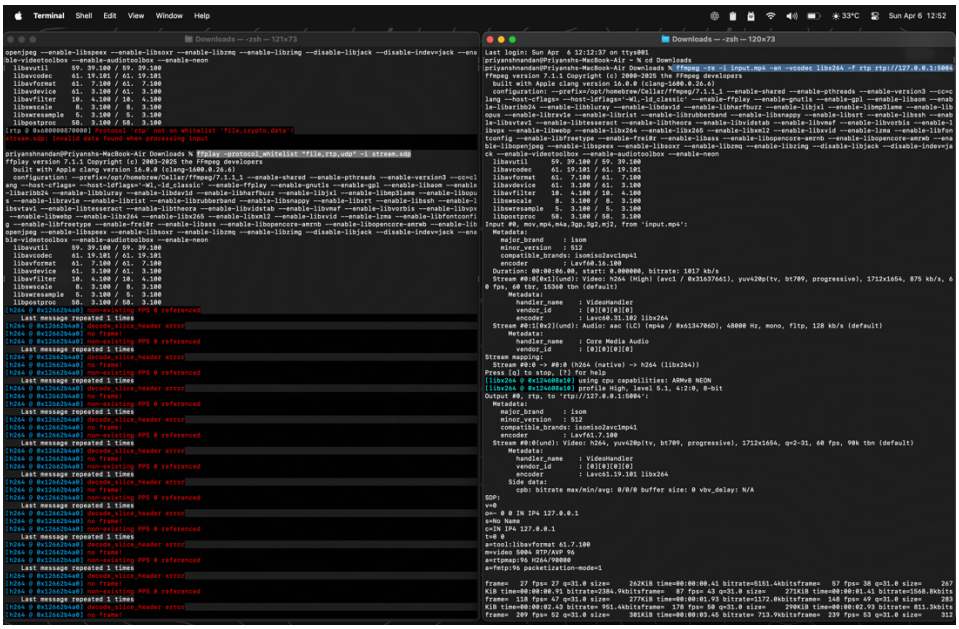
5. In another terminal, receive the stream with

ffplay stream.sdp

6. In Wireshark, apply filter:

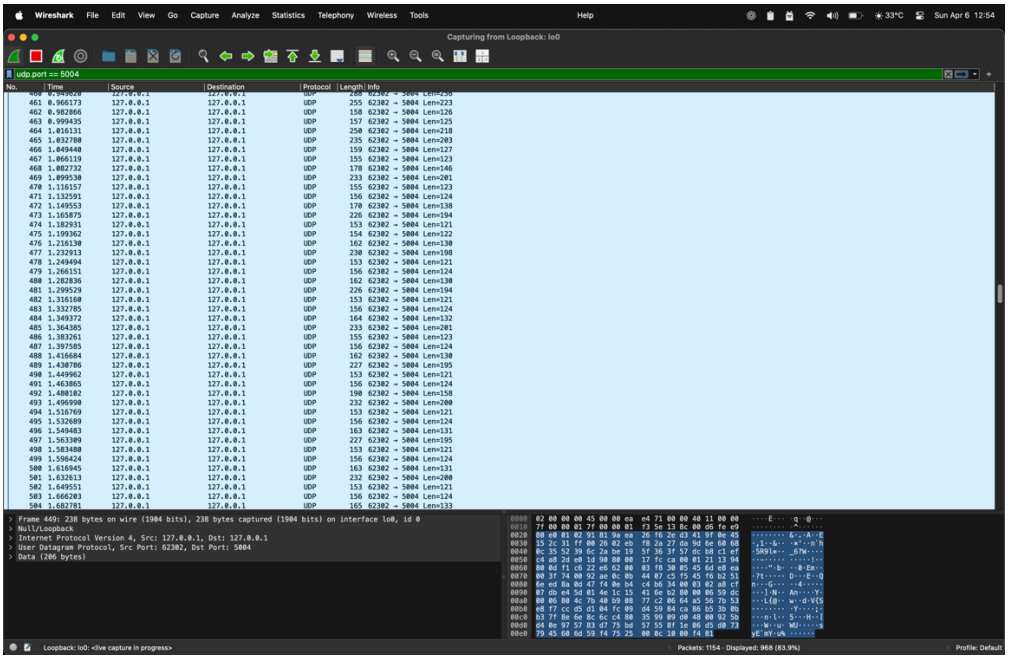
udp.port==5004

Generating and directing the packets:

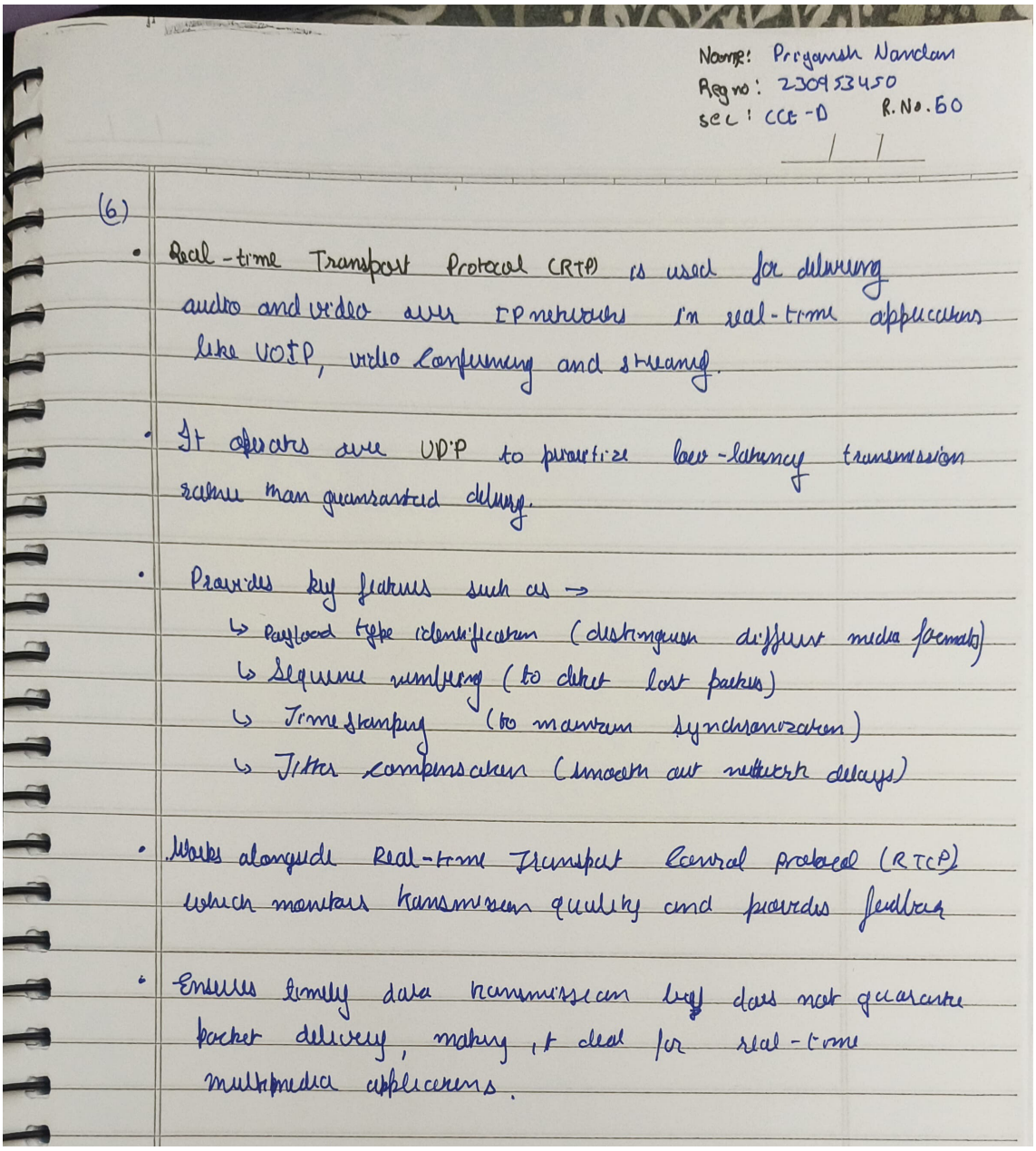


```
priyanshnandan@Priyanshs-MacBook-Air Downloads % cat stream.sdp
v=0
o=- 0 0 IN IP4 127.0.0.1
s=No Name
c=IN IP4 127.0.0.1
t=0 0
m=video 5004 RTP/AVP 96
a=rtpmap:96 H264/90000
```

Wireshark capture



Q2) Write a short note on RTP.



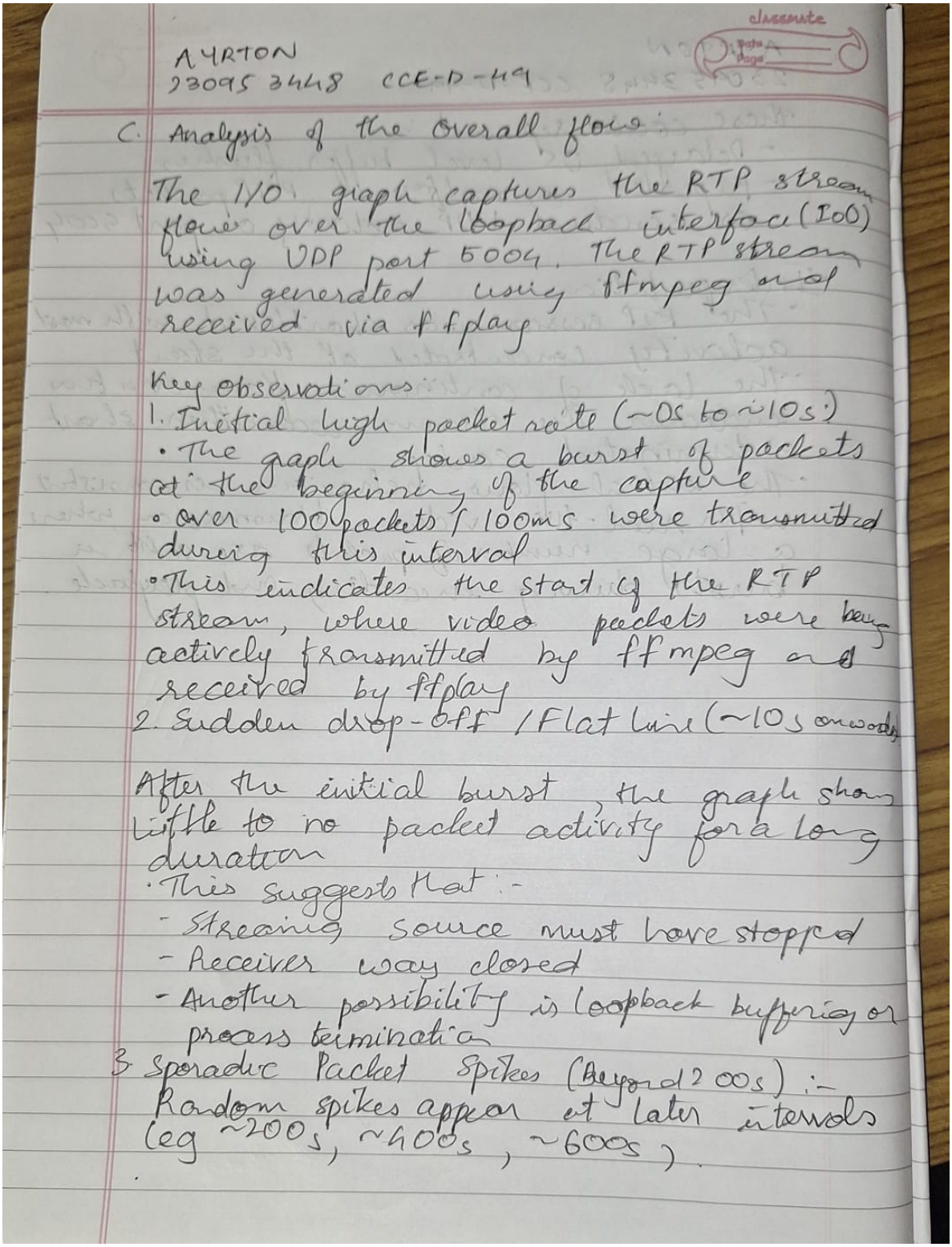
Name: Priyanshu Nandan
Reg no: 23095345O
sec: CCE-D R.No.60

(6)

- Real-time Transport Protocol (RTP) is used for delivering audio and video over IP networks in real-time applications like VoIP, video conferencing and streaming.
- It operates over UDP to prioritize low-latency transmission rather than guaranteed delivery.
- Provides key features such as →
  - ↳ Payload type identification (distinguish different media formats)
  - ↳ Sequence numbering (to detect lost packets)
  - ↳ Time stamping (to maintain synchronization)
  - ↳ Jitter compensation (smooth out network delays)
- Works alongside Real-time Transport Control protocol (RTCP) which monitors transmission quality and provides feedback
- Ensures timely data transmission but does not guarantee packet delivery, making it ideal for real-time multimedia applications.

Q3) Show the flow IO graph and try to analyze the flow (overflow).

Analysis:



AYRTON
23095 3448 CCE-D-49

C. Analysis of the Overall flow:

The I/O graph captures the RTP stream flows over the loopback interface (lo0) using UDP port 5004. The RTP stream was generated using ffmpeg and received via ffplay

Key observations:
1. Initial high packet rate (~0s to ~10s):
- The graph shows a burst of packets at the beginning of the capture
- over 100 packets / 100ms were transmitted during this interval
- This indicates the start of the RTP stream, where video packets were being actively transmitted by ffmpeg and received by ffplay
2. Sudden drop-off / Flat line (~10s onwards)

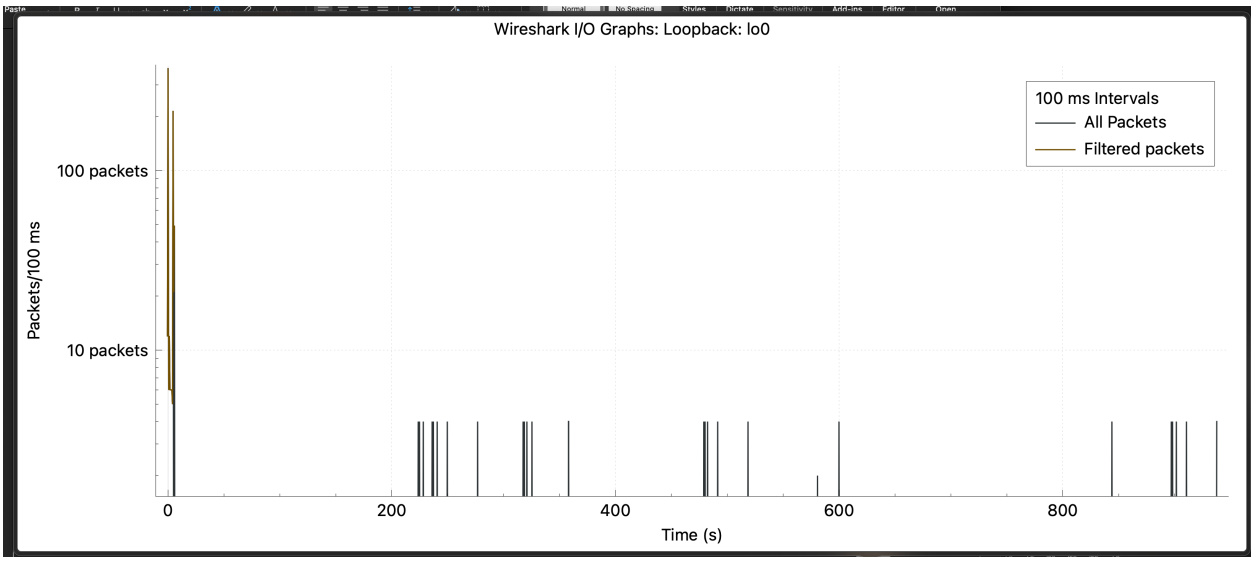
After the initial burst, the graph shows little to no packet activity for a long duration
- This suggests that:-
  - Streaming source must have stopped
  - Receiver was closed
  - Another possibility is loopback buffering or process termination
3. Sporadic Packet Spikes (Beyond 200s):-
Random spikes appear at later intervals (eg ~200s, ~400s, ~600s).

AYRTON
23095 3448 CCE-D-49

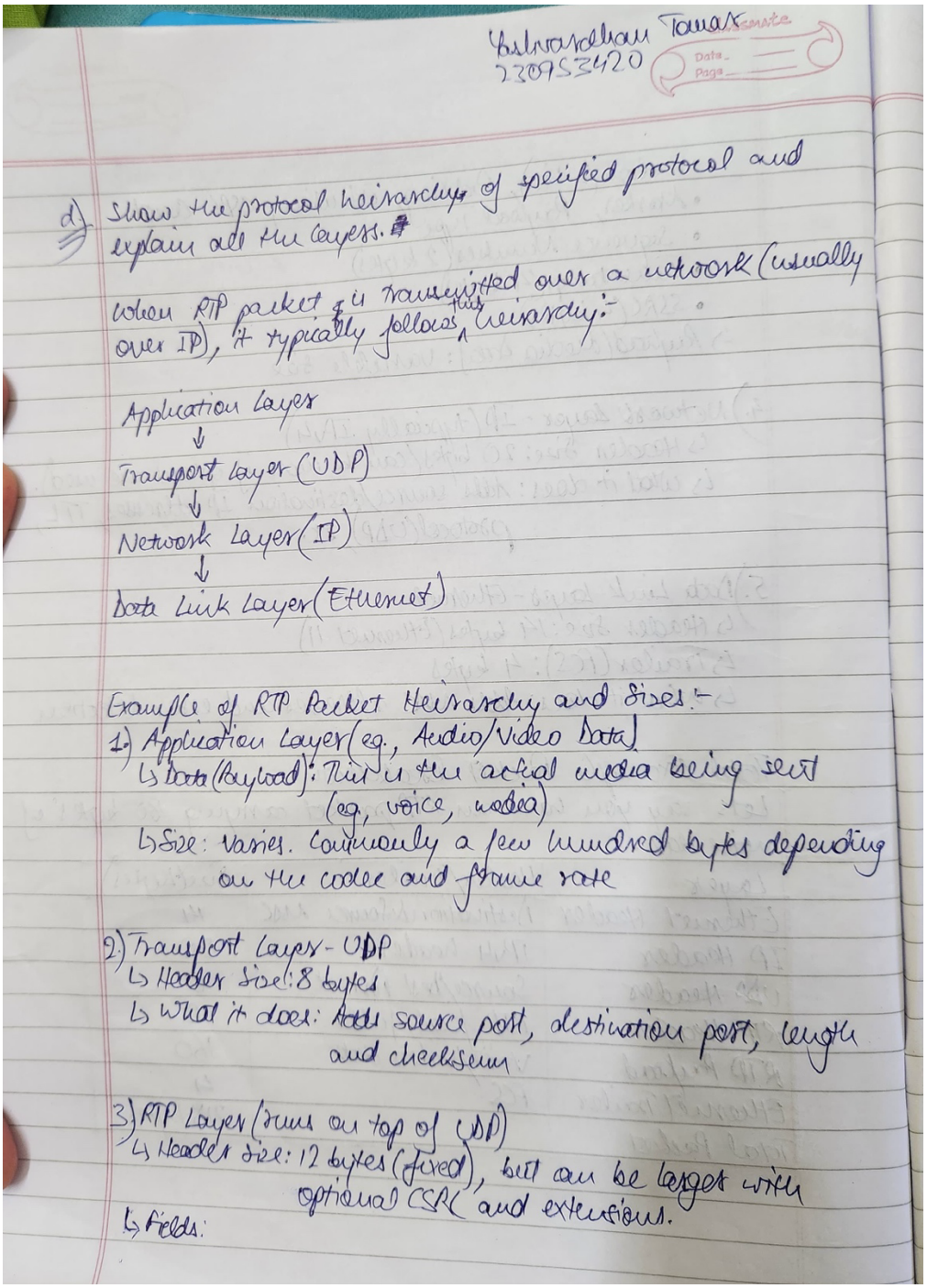
These could represent:
- Delayed OS-level buffer flushes
- Repeated or malformed RTP packets
- Or residual UDP activity on port 5004

- Interpretation
- The RTP session was short-lived with most activity concentrated at the start
- The lack of continuous flow over time indicates the stream was either short or terminated early
- The initial flow behaviour is consistent with real-time video transmission, where a large number of packets are sent in bursts during encoding and playback.

Flow IO graph:



Q4) Show the Protocol hierarchy of RTP and explain all the layers in it.



Lakshmidhar Tomar
230953420

4) Show the protocol heirarchy of specified protocol and explain all the layers.

When RTP packet is transmitted over a network (usually over IP), it typically follows this heirarchy:-

Application Layer
↓
Transport Layer (UDP)
↓
Network Layer (IP)
↓
Data Link Layer (Ethernet)

Example of RTP Packet Heirarchy and Sizes:-

1) Application Layer (eg., Audio/Video Data)
   ↳ Data (Payload): This is the actual media being sent (eg, voice, video)
   ↳ Size: Varies. Commonly a few hundred bytes depending on the codec and frame rate

2) Transport Layer - UDP
   ↳ Header Size: 8 bytes
   ↳ What it does: Adds source port, destination port, length and checksum

3) RTP Layer (runs on top of UDP)
   ↳ Header Size: 12 bytes (fixed), but can be larger with optional CSRC and extensions.
   ↳ Fields:

- Version (2 bits), Padding, Extension, CSRC count
- Marker, Payload Type
- Sequence Number (2 bytes)
- Timestamp (4 bytes)
- SSRC (4 bytes)

↳ Payload (media data): Variable size

4.) Network Layer - IP (typically IPv4)
↳ Header Size: 20 bytes (can be more if options are used).
↳ What it does: Adds source/destination IP addresses, TTL, protocol (UDP), etc.

5.) Data Link Layer - Ethernet
↳ Header Size: 14 bytes (Ethernet II)
↳ Trailer (FCS): 4 bytes
↳ What it does: Adds MAC addresses and error detection

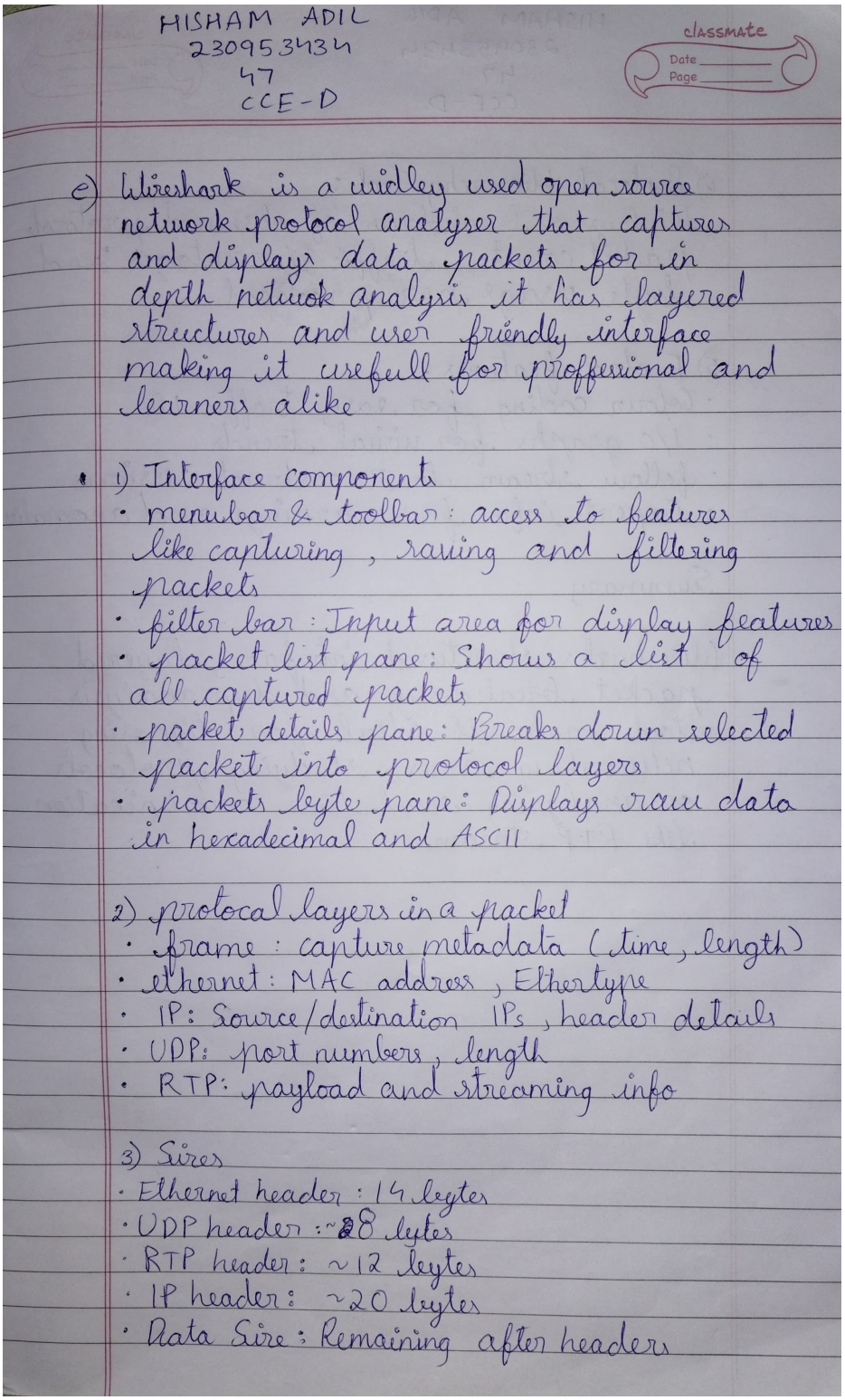
Total Example Packet Breakdown
Lets say you have an RTP packet carrying 160 bytes of voice data:

Layer	Header/Trailer	Size (Bytes)
Ethernet Header	Destination & Source MAC	14
IP Header	IPv4 header	20
UDP Header	Source/Dest Port etc.	8
RTP Header	RTP fixed fields	12
RTP Payload	Voice/Video data	160
Ethernet Trailer	FCS	4
Total Packet		218 bytes

Summary of RTP Protocol Hierarchy

Layer	Protocol	Header Size	Purpose
Application	RTP Payload	Variable	Actual media data
Transport	UDP	8 bytes	Multiplexing, no guarantee
RTP Sub-layer	RTP	12+ bytes	Sequencing, timing, sync
Network	IP	20 bytes	Logical addressing
Data Link	Ethernet	14+4 bytes	Physical addressing, error detection

Q5) Explain the structure of Wireshark.



HISHAM ADIL
23095343447
CCE-D

classmate
Date
Page

e) Wireshark is a widely used open source network protocol analyzer that captures and displays data packets for in depth network analysis it has layered structures and user friendly interface making it usefull for proffesional and learners alike

- 1) Interface components
  - menubar & toolbar: access to features like capturing, saving and filtering packets
  - filter bar: Input area for display features
  - packet list pane: Shows a list of all captured packets
  - packet details pane: Breaks down selected packet into protocol layers
  - packets bytes pane: Displays raw data in hexadecimal and ASCII

2) protocal layers in a packet
- frame: capture metadata (time, length)
- ethernet: MAC address, Ethertype
- IP: Source/destination IPs, header details
- UDP: port numbers, length
- RTP: payload and streaming info

3) Sizes
- Ethernet header: 14 bytes
- UDP header: ~8 bytes
- RTP header: ~12 bytes
- IP header: ~20 bytes
- Data Size: Remaining after headers

HISHAM ADIL
230953434
47
CCE-D

4) Protocal Hierarchy Tool:
Displays a breakdown of observed protocols packet counts, traffic percentages and byte usage per protocol

5) Extra features:
- Colour coding for easy traffic ID
- I/O graphs for visual trends
- follow stream to reconstruct sessions
- Expert info for warnings and anomalies

Summary

Wireshark's structured interface, layered packet breakdown and deep analysis tools make it ideal for diagnosing networking issues, studying protocols or analyzing real time communication like RTP streams