MSCS-631: Python: Lab7: Video Streaming

Sandesh Pokharel

University of the Cumberlands

MSCS-631 Advanced Computer Networks

Dr. Charles Lively

February 23, 2025

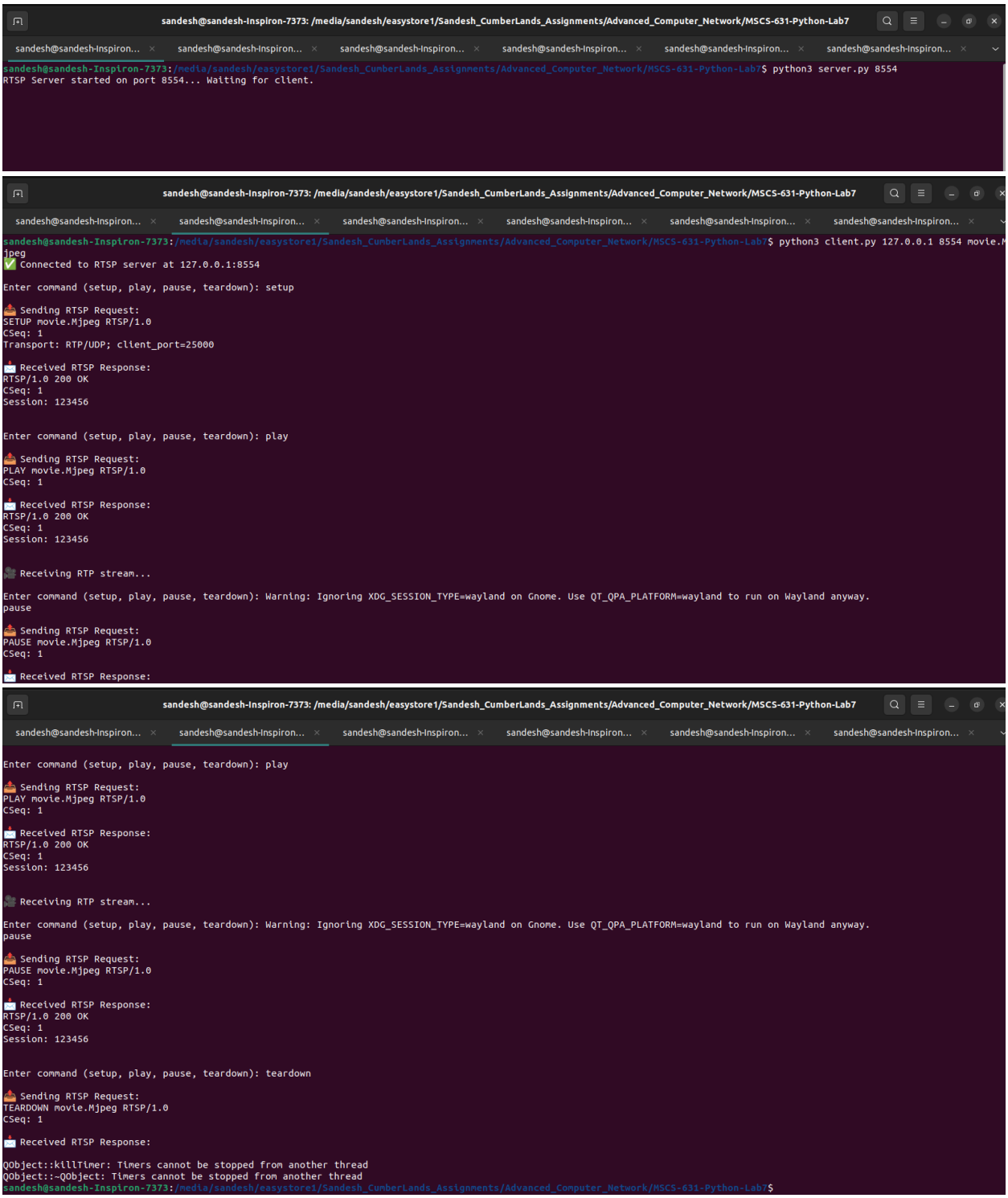
Introduction

This report details the implementation and analysis of an RTSP video streaming system. The objective was to set up an RTSP server and client, ensure correct RTP packet handling, and analyze streaming performance. The system was tested by streaming a video file over RTP and verifying network packet transmission.

Implementation details

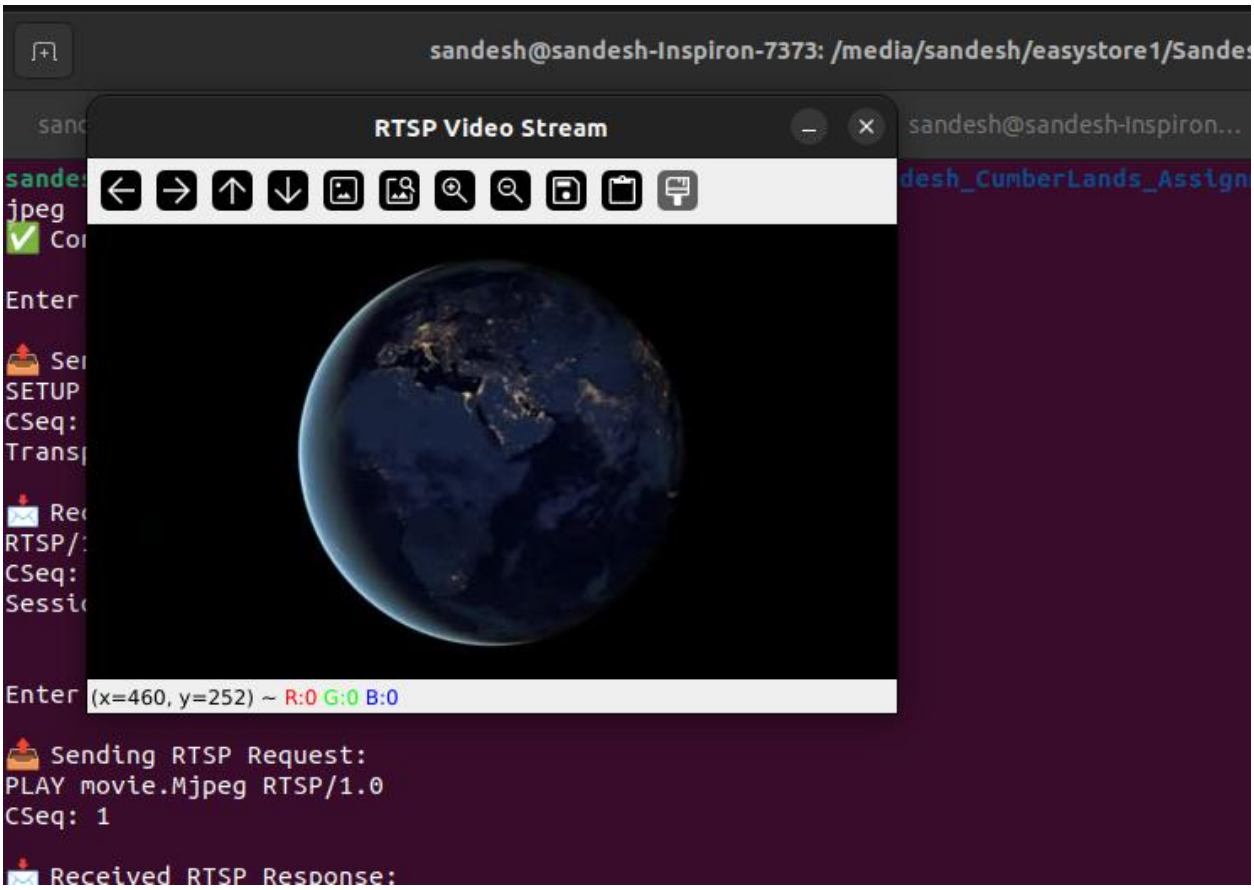
RTSP Server and Client Setup

The RTSP server was developed to handle client requests, process RTSP commands, and stream a video file over RTP. The client implemented the necessary logic to send RTSP requests, receive RTP packets, and play the video using OpenCV.



Video Playback

The client processes RTP packets and decodes video frames using OpenCV, displaying the video stream in a separate window.

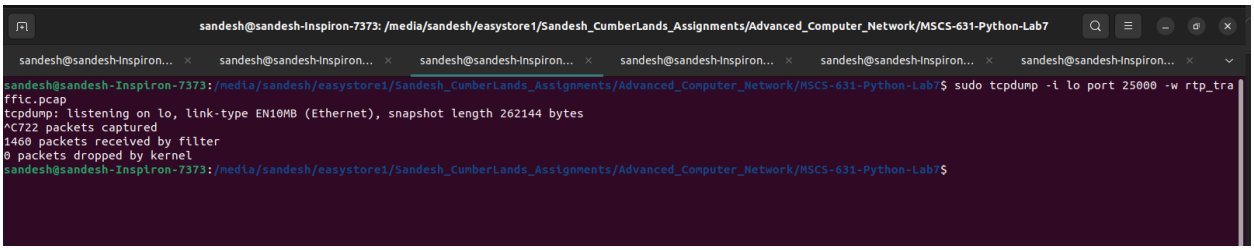


RTP Packet Capture and Analysis

Capturing RTP Packets (Packet loss)

To analyze the RTP packets, tcpdump was used to capture the network traffic on the specified port.

sudo tcpdump -i lo port 25000 -w rtp_traffic.pcap



Observed zero packet loss during streaming.

Verifying RTP Stream in Tshark

tshark -r rtp_traffic.pcap -q -z io,stat,10

```
sandesh@sandesh-Inspiron-7373:/media/sandesh/easystore1/Sandesh_CumberLands_Assignments/Advanced_Computer_Network/MSCS-631-Python-Lab7$ tshark -r rtp_traffic.pcap -q -z io,stat
,10

==============================
| IO Statistics              |
|                            |
| Duration: 12.589455 secs   |
| Interval: 10 secs          |
|                            |
| Col 1: Frames and bytes    |
|----------------------------|
|          |1                |
| Interval | Frames |  Bytes |
|----------------------------|
|  0 <> 10 |    270 | 2940528 |
| 10 <> Dur|     71 |  756990 |
==============================
sandesh@sandesh-Inspiron-7373:/media/sandesh/easystore1/Sandesh_CumberLands_Assignments/Advanced_Computer_Network/MSCS-631-Python-Lab7$
```

Video Date Rate Calculation

To calculate the video data rate, we used:

tshark -r rtp_traffic.pcap -q -z io,stat,10


```
sandesh@sandesh-Inspiron-7373:/media/sandesh/easystore1/Sandesh_CumberLands_Assignments/Advanced_Computer_Network/MSCS-631-Python-Lab7$ tshark -r rtp_traffic.pcap -q -z io,stat
,10

==============================
| IO Statistics              |
|                            |
| Duration: 12.589455 secs   |
| Interval: 10 secs          |
|                            |
| Col 1: Frames and bytes    |
|----------------------------|
|          |1                |
| Interval | Frames |  Bytes |
|----------------------------|
|  0 <> 10 |    270 | 2940528 |
| 10 <> Dur|     71 |  756990 |
==============================
sandesh@sandesh-Inspiron-7373:/media/sandesh/easystore1/Sandesh_CumberLands_Assignments/Advanced_Computer_Network/MSCS-631-Python-Lab7$
```


Formula Used:

Video Data Rate(bps) = (Total Bytes * 8)/Duration(Seconds)

Calculating Video Data Rate:

Total Bytes Transferred: 3,697,518 bytes

Total Duration: 12.59 seconds

Computed Video Data Rate: ~2.35 Mbps

Challenges Faced and Solution

Challenges:

- Video was not displaying initially.
- RTP packets were not properly processed.
- OpenCV dependencies were missing.
- tshark initially did not detect RTP packets.

Solutions Implemented:

- Debugged RTP packet structure to correctly extract and decode frames.
- Fixed OpenCV integration and ensured frames were properly reconstructed.
- Verified RTP stream capture using tcpdump and tshark.

Conclusion

This project successfully implemented an RTSP server-client system that streams video using RTP. We verified RTP packet transmission, analyzed the video data rate, and ensured correct playback.

Key takeaways:

- Understanding the RTSP protocol and RTP streaming.
- Debugging and analyzing RTP streams with packet capture tools.
- Measuring video performance metrics like data rate and packet loss.