

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC BÁCH KHOA  
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



## MẠNG MÁY TÍNH

---

Báo cáo Bài tập lớn 1

## STREAMING VIDEO

---

GVHD:	Th.S Nguyễn Hồng Nam	
SV thực hiện:	Nguyễn Phi Thông	1814205
	Nguyễn Văn Thuận	1814220
	Lộc Quốc Huy	1812369
	Lê Văn Thi	1613298

Tp. Hồ Chí Minh, Tháng 11/2020



# Mục lục

<b>1</b>	<b>Phân tích yêu cầu</b>	<b>2</b>
1.1	Mục tiêu . . . . .	2
1.2	Yêu cầu . . . . .	2
1.2.1	Client . . . . .	2
1.2.2	Server . . . . .	2
<b>2</b>	<b>Function description</b>	<b>3</b>
2.1	Client.py . . . . .	3
2.2	RtpPacket.py . . . . .	3
2.3	ServerWorker.py . . . . .	4
2.4	VideoStream.py . . . . .	4
<b>3</b>	<b>Class diagram</b>	<b>5</b>
<b>4</b>	<b>Extend</b>	<b>5</b>
<b>5</b>	<b>Hướng dẫn sử dụng</b>	<b>6</b>
<b>6</b>	<b>Đánh giá kết quả</b>	<b>10</b>

# 1 Phân tích yêu cầu

## 1.1 Mục tiêu

Mục tiêu của bài tập lớn là chúng ta sẽ hiện thực streaming video server và client sử dụng Real-Time Streaming Protocol (RTSP) để giao tiếp và Real-Time Transfer Protocol (RTP) để gửi dữ liệu.

## 1.2 Yêu cầu

### 1.2.1 Client

Hiện thực RTSP protocol. Chúng ta cần hoàn thành các function mà được gọi khi người dùng click vào các button trên giao diện người dùng. Khi client bắt đầu, RTSP socket sẽ được mở đến server và sử dụng socket này cho việc gửi các yêu cầu RTSP. Các button:

- **SETUP**  
Command này sẽ thiết lập các session và tham số truyền tải.
- **PLAY**  
Command này sẽ playback cho client.
- **PAUSE**  
Command này sẽ tạm dừng việc playback.
- **TEARDOWN**  
Command này sẽ kết thúc session và đóng kết nối.

### 1.2.2 Server

Chúng ta tạo ra packet, set các fields trong header của packet và copy payload vào trong packet. Khi server nhận được yêu cầu PLAY từ client, server sẽ đọc một video frame từ file và tạo ra một đối tượng RtpPacket là RTP-encapsulation của video frame và gửi frame này đến client qua UDP sau mỗi 50ms. Để đóng gói, chúng ta sẽ gọi hàm encode của class RtpPacket. Nhiệm vụ của chúng ta là hiện thực cái hàm này theo những bước sau:

- Set RTP-version ( $V$ ) = 2.
- Set padding ( $P$ ), extension ( $X$ ), số contributing sources ( $CC$ ), marker ( $M$ ), tất cả đều bằng 0.
- Set payload type ( $PT$ ) = 26.
- Set sequence number.
- Set timestamp.
- Set sources identifier.

## 2 Function description

### 2.1 Client.py

- createWidgets: tạo giao diện cho client với 4 button: SETUP, PLAY, PAUSE, TEARDOWN.
- setupMovie: gửi yêu cầu SETUP đến server.
- exitClient: gửi yêu cầu TEARDOWN đến server và đóng giao diện người dùng.
- pauseMovie: gửi yêu cầu PAUSE đến server.
- playMovie: gửi yêu cầu PLAY đến server.
- listenRtp: nhận data từ server để decode và update lại số frame.
- writeFrame: viết các frame nhận được vào 1 file và trả về image file.
- updateMovie: update lại image file cũng như video frame trên giao diện.
- connectToServer: kết nối với server và bắt đầu session RTSP/TCP mới.
- sendRtpRequest: gửi các request đến server.
- recvRtpReply: nhận phản hồi từ server.
- parseRtpReply: xử lý phản hồi từ server.
- openRtpPort: mở socket.
- handler: xử lý event khi người dùng nhấn nút (x) để thoát.

### 2.2 RtpPacket.py

- encode: set các field trong header của RTP packet.
- decode: tách RTP packet thành header và payload.
- version: trả về version của RTP.
- seqNum: trả về sequence number.
- timestamp: trả về timestamp.
- payloadType: trả về kiểu của payload.
- getPayLoad: trả về payload.
- getPacket: trả về RTP packet.

## 2.3 ServerWorker.py

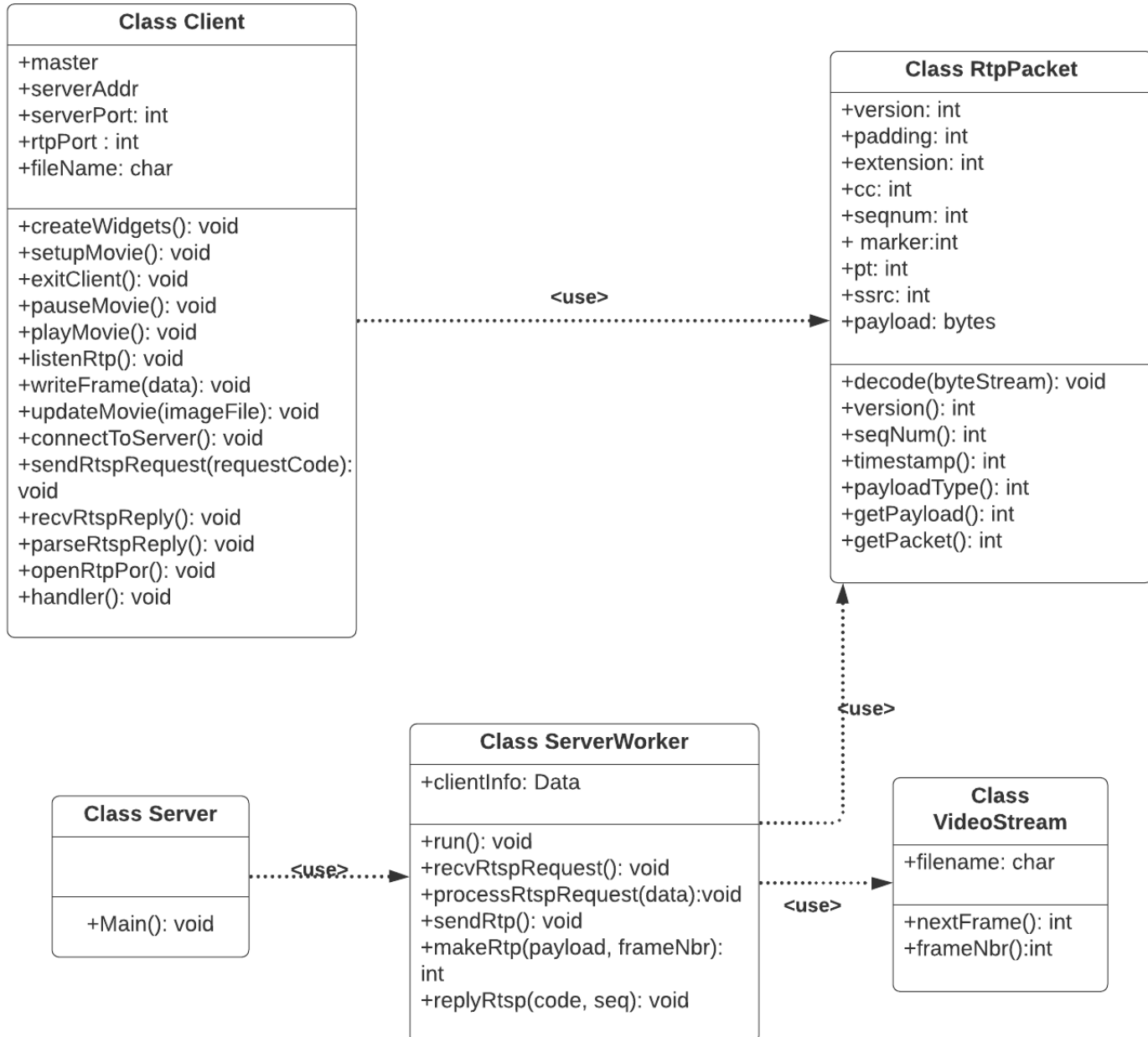
- run: bắt đầu server.
- recvRtspRequest: nhận RTSP request từ client.
- processRtspRequest: xử lý RTSP request từ client.
- sendRtp: gửi RTP packet qua UDP.
- makeRtp: RTP-packetize dữ liệu video.
- replyRtsp: gửi phản hồi RTSP đến

## 2.4 VideoStream.py

- nextFrame: get frame kế tiếp.
- frameNbr: trả về fram number.

### 3 Class diagram

#### Class diagram Assignment NetWork



### 4 Extend

- Tính toán RTP packet loss rate: Khi cho thực hiện các lệnh chạy trên cùng một máy cho server và client thì không nhận thấy được sự mất gói trong quá trình phát video. Từ đó, kết luận rằng RTP packet loss rate = 0 trong trường hợp thực hiện chạy các lệnh như trên.
- Tính toán video data rate: Sử dụng công thức tính dựa vào kích thước file và thời gian của video. Với kích thước file là 4170 KB và thời gian video đo được là 32 giây thì kết quả thu được là 1.0425 mbps.

## 5 Hướng dẫn sử dụng

1. Bắt đầu server bằng cách chạy file `Server.py` bằng lệnh sau:

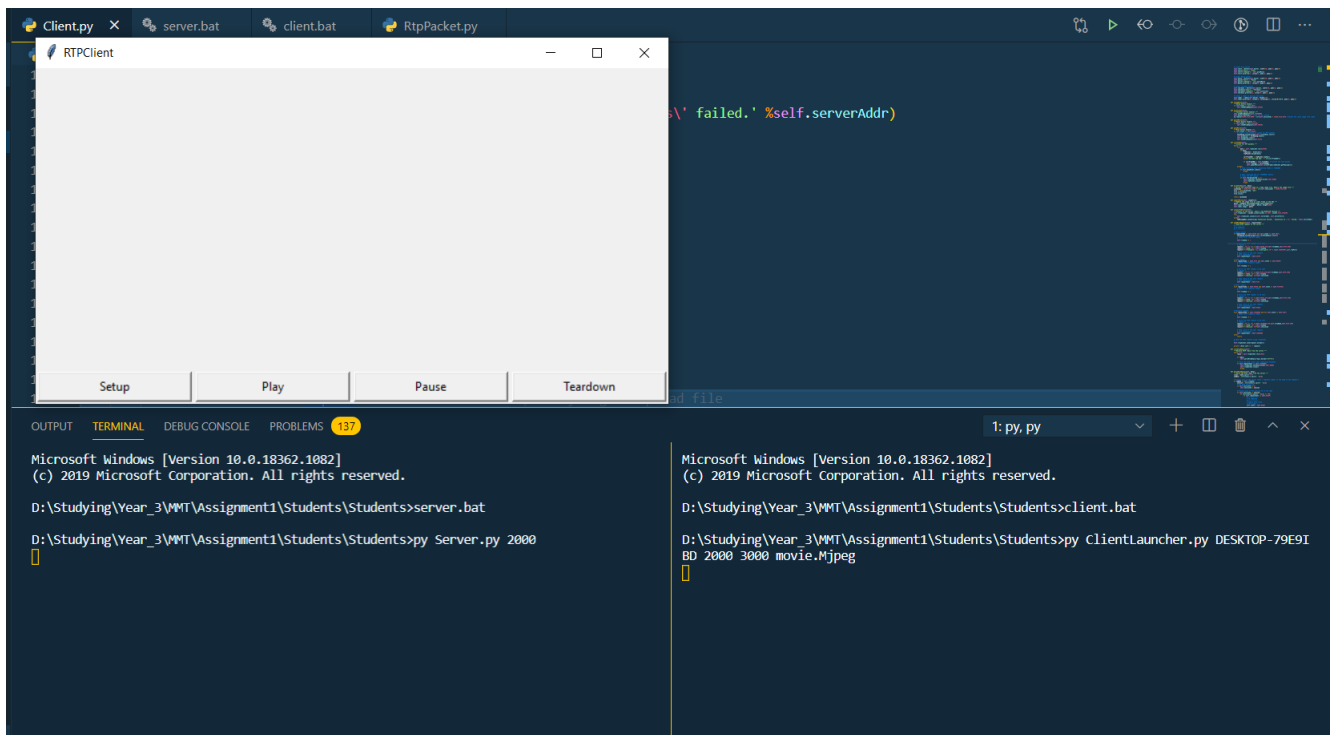
**py Server.py server\_port**

Ta nên chọn `server_port` lớn hơn 1024, ví dụ ta chọn `server_port = 2000` trong phần hướng dẫn này.

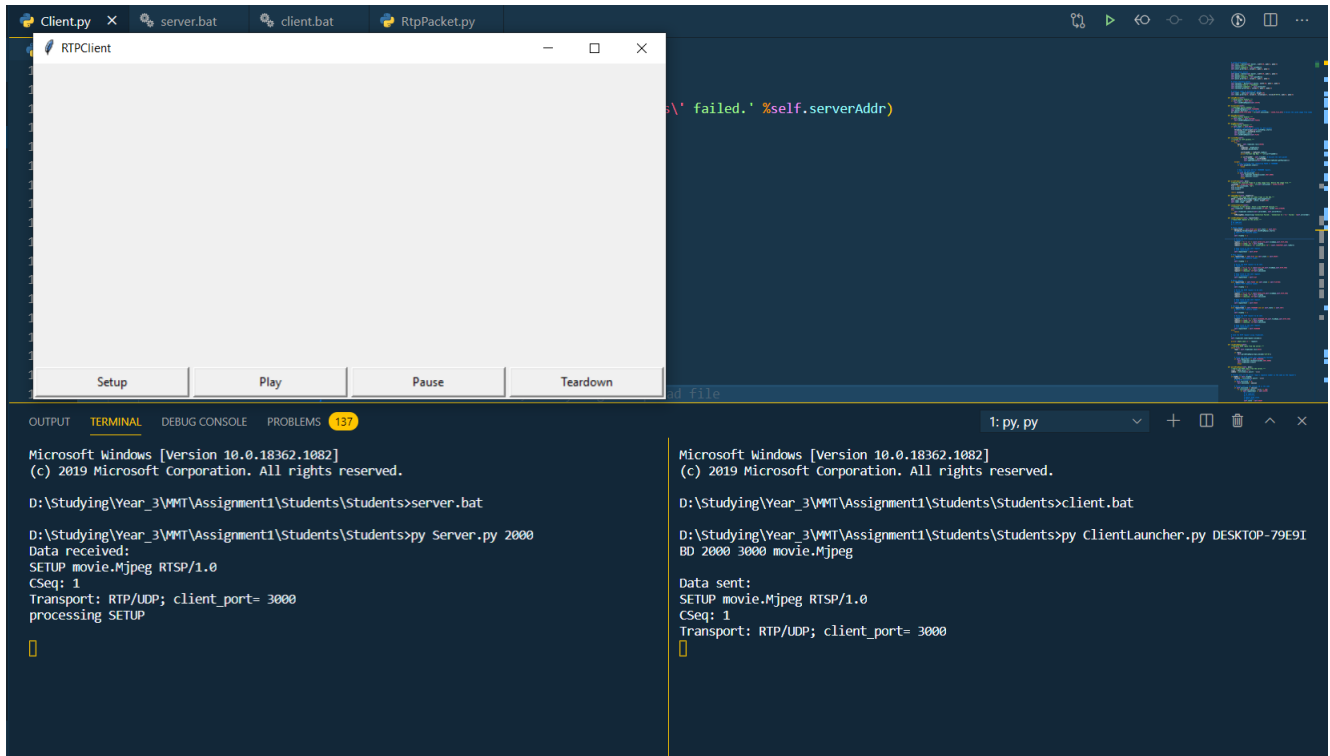
2. Sau đó, ta bắt đầu bên client bằng cách chạy file `ClientLauncher.py` bằng lệnh sau:

**py ClientLaucher.py server\_host server\_port RTP\_port name\_video**

Sau khi chạy lệnh trên, ta sẽ thấy giao diện xuất hiện, và có 4 button *Setup*, *Play*, *Pause*, *TearDown*. Ngoài ra, khung terminal bên trái là bắt đầu server và bên phải là bắt đầu bên client.



3. Khi client click vào button **Setup**



Bên phía client sẽ gửi yêu cầu lên server với data gồm có:

- Command SETUP
- Tên video
- Version RTSP
- RTSP sequence number
- Giao thức truyền
- client\_port

Bên phía server sẽ nhận được tất cả data mà client gửi và hiện ra trong terminal cùng với dòng thông báo *processing SETUP*

#### 4. Khi client click vào button **Play**

**Server:**

```
Data received:
SETUP Movie.Mjpeg RTSP/1.0
CSeq: 1
Transport: RTP/UDP; client_port= 5008
processing SETUP

Data received:
PLAY Movie.Mjpeg RTSP/1.0
CSeq: 2
Session: 428824
processing PLAY
```

**Client:**



```
C:\Users\Win 10\Desktop\network>python ClientLauncher.py 127.0.0.1 1025 5008 Movie.Mjpeg

Data sent:
SETUP Movie.Mjpeg RTSP/1.0
CSeq: 1
Transport: RTP/UDP; client_port= 5008

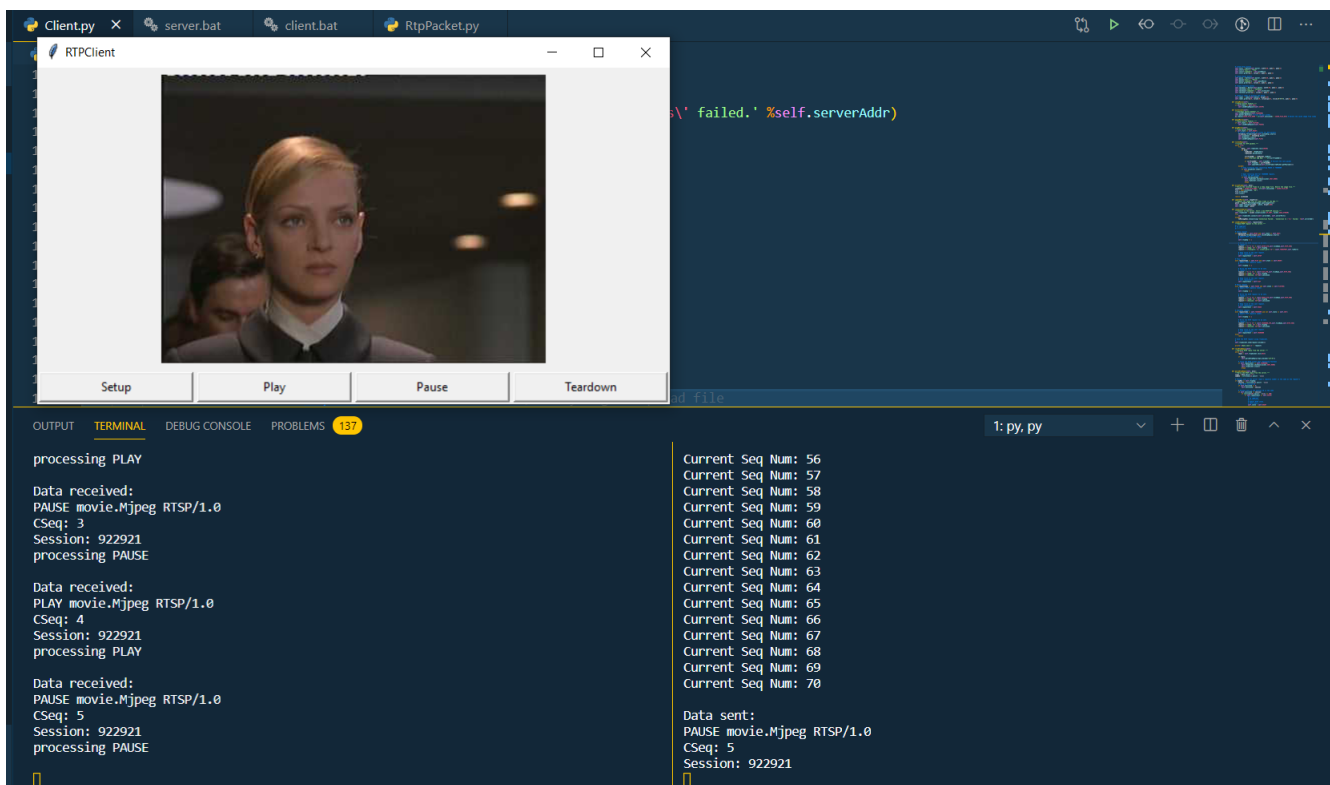
Data sent:
PLAY Movie.Mjpeg RTSP/1.0
CSeq: 2
Session: 428824
Current Seq Num: 1
Current Seq Num: 2
Current Seq Num: 3
Current Seq Num: 4
Current Seq Num: 5
Current Seq Num: 6
Current Seq Num: 7
Current Seq Num: 8
Current Seq Num: 9
Current Seq Num: 10
Current Seq Num: 11
Current Seq Num: 12
```

Bên phía client sẽ gửi yêu cầu lên server với data gồm có:

- Command PLAY
- Tên video
- Version RTSP
- RTSP sequence number
- Session
- Chuỗi Current Seq Num

Bên phía server sẽ nhận được tất cả data mà client gửi và hiện ra trong terminal cùng với dòng thông báo *processing PLAY*

##### 5. Khi client click vào button **Pause**



The screenshot shows a Python application with a video player window titled 'RTPClient' displaying a video of a woman. Below the video player are four buttons: 'Setup', 'Play', 'Pause', and 'Teardown'. The 'Pause' button is highlighted. To the right of the video player is a code editor showing Python code. Below the video player and code editor is a terminal window with the following output:

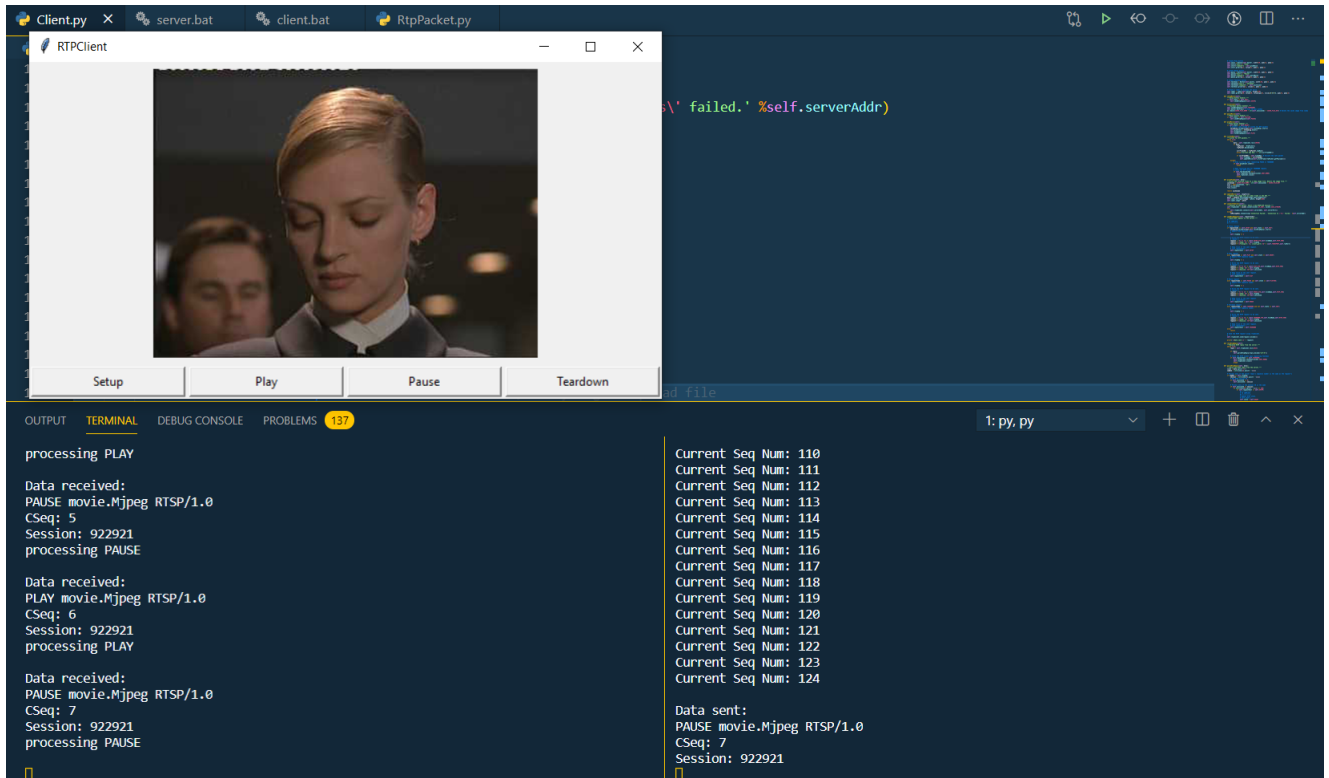
```
processing PLAY
Data received:
PAUSE movie.Mjpeg RTSP/1.0
CSeq: 3
Session: 922921
processing PAUSE
Data received:
PLAY movie.Mjpeg RTSP/1.0
CSeq: 4
Session: 922921
processing PLAY
Data received:
PAUSE movie.Mjpeg RTSP/1.0
CSeq: 5
Session: 922921
processing PAUSE
```

On the right side of the terminal, there is a list of sequence numbers:

```
Current Seq Num: 56
Current Seq Num: 57
Current Seq Num: 58
Current Seq Num: 59
Current Seq Num: 60
Current Seq Num: 61
Current Seq Num: 62
Current Seq Num: 63
Current Seq Num: 64
Current Seq Num: 65
Current Seq Num: 66
Current Seq Num: 67
Current Seq Num: 68
Current Seq Num: 69
Current Seq Num: 70
```

At the bottom right of the terminal, there is a message:

```
Data sent:
PAUSE movie.Mjpeg RTSP/1.0
CSeq: 5
Session: 922921
```



Bên phía client sẽ gửi yêu cầu lên server với data gồm có:

- Command PAUSE
- Tên video
- Version RTSP
- RTSP sequence number
- Session

Bên phía server sẽ nhận được tất cả data mà client gửi và hiện ra trong terminal cùng với dòng thông báo *processing PAUSE*

#### 6. Khi client click vào button **Teardown**

The image shows a Visual Studio Code editor with a Python file named `Client.py` open. The script implements an RTSP client. It has a `sendRtspRequest` method that connects to a server and sends requests. The terminal output shows the client successfully connecting and sending several RTSP requests (PLAY, PAUSE, TEARDOWN) and receiving responses.

```
Client.py > Client > sendRtspRequest
148     self.rtspSocket.connect((self.serverAddr, self.serverPort))
149 except:
150     tkMessageBox.showwarning('Connection Failed', 'Connection to \'%s\' failed.' % self.serverAddr)
151
152 def sendRtspRequest(self, requestCode):
153     """Send RTSP request to the server."""
154     #-----
155     # TO COMPLETE
156     #-----
157
158     # Setup request
159     if requestCode == self.SETUP and self.state == self.INIT:
160         threading.Thread(target=self.recvRtspReply).start()
161         # Update RTSP sequence number.
162         # ...
163         self.rtspSeq += 1
164
165     # Write the RTSP request to be sent.
```

Terminal Output:

```
processing PAUSE
Data received:
PLAY movie.Mjpeg RTSP/1.0
CSeq: 6
Session: 922921
processing PLAY
Data received:
PAUSE movie.Mjpeg RTSP/1.0
CSeq: 7
Session: 922921
processing PAUSE
Data received:
TEARDOWN movie.Mjpeg RTSP/1.0
CSeq: 8
Session: 922921
processing TEARDOWN
Current Seq Num: 116
Current Seq Num: 117
Current Seq Num: 118
Current Seq Num: 119
Current Seq Num: 120
Current Seq Num: 121
Current Seq Num: 122
Current Seq Num: 123
Current Seq Num: 124
Data sent:
PAUSE movie.Mjpeg RTSP/1.0
CSeq: 7
Session: 922921
Data sent:
TEARDOWN movie.Mjpeg RTSP/1.0
CSeq: 8
Session: 922921
D:\Studying\Year_3\MMT\Assignment1\Students\Students>
```

Bên phía client sẽ gửi yêu cầu lên server với data gồm có:

- Command TEARDOWN
- Tên video
- Version RTSP
- RTSP sequence number
- Session

Bên phía server sẽ nhận được tất cả data mà client gửi và hiện ra trong terminal cùng với dòng thông báo *processing TEARDOWN*

## 6 Đánh giá kết quả

- Về cơ bản, nhóm em đã hoàn thành được phần bài làm. Code được các event cho các button PLAY, SETUP, TEARDOWN, PAUSE.
- Chương trình chạy khá ổn. Không xuất hiện bug trong quá trình chạy.

MSSV	Học và tên	Công việc
1814205	Nguyễn Phi Thông	Phân tích yêu cầu, function description, user manual, báo cáo
1814220	Nguyễn Văn Thuần	Extend, user manual
1812369	Lộc Quốc Huy	class diagram
1613289	Lê Văn Thi	Function description