

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC BÁCH KHOA  
KHOA KHOA HỌC - KỸ THUẬT MÁY TÍNH



MẠNG MÁY TÍNH

---

---

# Assignment 1

---

---

**GVHD :** Nguyễn Hồng Nam  
**Thành viên :** Đoàn Tấn Lộc - 1812962  
Đặng Hữu Nam - 1720034  
Nguyễn Văn Khang - 1812554  
Nguyễn Duy Kiên - 1812704

TP. HỒ CHÍ MINH, THÁNG 11/2020



## Mục lục

<b>1</b>	<b>Giới thiệu tổng quát</b>	<b>2</b>
1.1	Yêu cầu bài toán . . . . .	2
1.2	Giao diện ứng dụng . . . . .	2
1.3	Các chức năng . . . . .	3
<b>2</b>	<b>Giao thức sử dụng</b>	<b>3</b>
2.1	Real-Time Streaming Protocol (RTSP) . . . . .	3
2.2	Real-time Transfer Protocol (RTP) . . . . .	3
2.3	RTP Packet Header . . . . .	3
<b>3</b>	<b>State Diagram của Client</b>	<b>4</b>
<b>4</b>	<b>Hàm và chức năng</b>	<b>5</b>
<b>5</b>	<b>Class Diagram</b>	<b>6</b>
<b>6</b>	<b>Yêu cầu - phản hồi của Client - Server</b>	<b>7</b>
<b>7</b>	<b>Hướng dẫn sử dụng</b>	<b>10</b>
<b>8</b>	<b>Đánh giá kết quả hiện thực</b>	<b>12</b>
<b>9</b>	<b>Tham khảo</b>	<b>13</b>

# 1 Giới thiệu tổng quát

## 1.1 Yêu cầu bài toán

Tạo ra một ứng dụng streaming video sử dụng mô hình Client-Server.

Sử dụng giao thức Real-Time Streaming Protocol (RTSP) trên client và Real-time Transfer Protocol (RTP) packetization trên server.

## 1.2 Giao diện ứng dụng

Gồm khu vực hiển thị video và một số nút chức năng:

- Setup
- Play
- Pause
- Teardown



### 1.3 Các chức năng

- Setup
  - Gửi yêu cầu SETUP để kết nối với server.
- Play
  - Gửi yêu cầu PLAY đến cho server để phát video.
- Pause
  - Gửi yêu cầu PAUSE đến cho server để dừng phát video.
- Teardown
  - Gửi yêu cầu TEARDOWN đến cho server để hủy "session" và đóng kết nối với server.

## 2 Giao thức sử dụng

### 2.1 Real-Time Streaming Protocol (RTSP)

**Real Time Streaming Protocol (RTSP)** – Giao thức truyền tin thời gian thực là một giao thức điều khiển truyền thông mạng ở tầng ứng dụng được thiết kế để sử dụng trong các hệ thống giải trí và truyền thông để điều khiển máy chủ chứa các dữ liệu truyền tin đa phương tiện (streaming media). Giao thức này được sử dụng để thiết lập và điều khiển các phiên truyền thông giữa các trạm cuối. Các máy khách của các máy chủ truyền thông ban ra các lệnh kiểu VCR, chẳng hạn như chơi, thâu và tạm dừng, để điều khiển thời gian thực của các phương tiện truyền tin trực tuyến từ máy chủ tới máy khách (Video On Demand) hoặc từ máy khách đến máy chủ (Voice Recording).

### 2.2 Real-time Transfer Protocol (RTP)

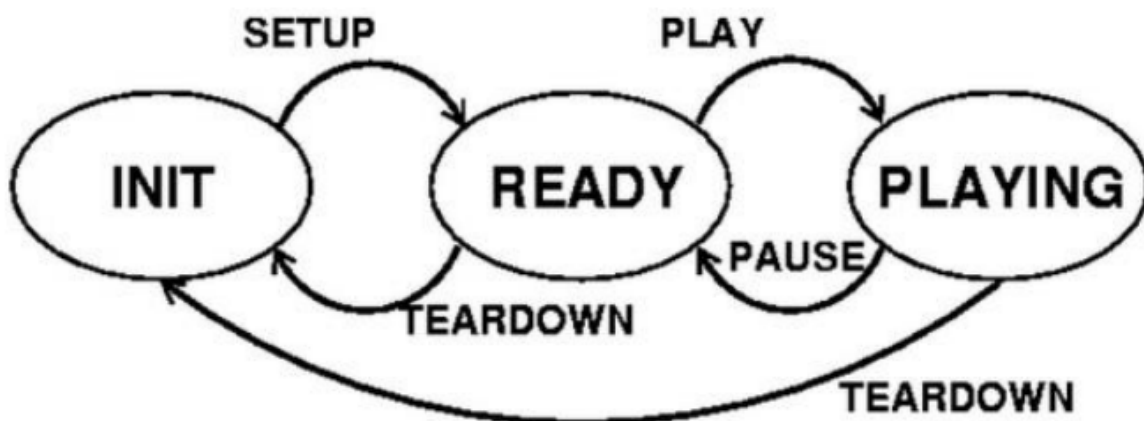
**Real time transfer protocol (RTP)** – Giao thức truyền tải thời gian thực, là một giao thức mạng để chuyển tập tin, video, âm thanh qua mạng IP. RTP được sử dụng rộng rãi trong các hệ thống truyền thông và giải trí liên quan đến các streaming media như gọi điện thoại, các ứng dụng hội nghị truyền hình, các dịch vụ truyền hình và các tính năng push-to-talk dựa trên nền web.

### 2.3 RTP Packet Header

RTP packet header																																	
Offsets	Octet	0						1								2								3									
Octet	Bit <sup>[a]</sup>	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Version		P	X	CC		M	PT								Sequence number																
4	32	Timestamp																															
8	64	SSRC identifier																															
12	96	CSRC identifiers																															
		...																															
12+4×CC	96+32×CC	Profile-specific extension header ID															Extension header length																
16+4×CC	128+32×CC	Extension header																															
		...																															

- Version (V), độ rộng 2 bit
- Padding (P), độ rộng 1 bit
- Extension (X), độ rộng 1 bit
- Number of contributing sources (CC), độ rộng 4 bit
- Marker (M), độ rộng 1 bit
- Payload type (PT), độ rộng 7 bit
- Sequence number, độ rộng 16 bit
- Timestamp, độ rộng 32 bit
- Source identifier (SSRC), độ rộng 32 bit

### 3 State Diagram của Client



- Client có 3 trạng thái: INIT, READY và PLAYING.
- Các tín hiệu chuyển trạng thái là các nút: SETUP, PLAY, PAUSE và TEARDOWN.

## 4 Hàm và chức năng

### Client.py:

- `__init__(self, master, serveraddr, serverport, rtpport, filename)` -> Constructor của class client.
- `createWidgets(self)` -> Tạo các GUI cho button.
- `setupMovie(self)` -> Xử lý nút nhấn SETUP.
- `exitClient(self)` -> Xử lý nút nhấn TEARDOWN.
- `pauseMovie(self)` -> Xử lý nút nhấn PAUSE.
- `playMovie(self)` -> Xử lý nút nhấn PLAY.
- `listenRtp(self)` -> Nhận các gói tin RTP.
- `writeFrame(self, data)` -> Tạo một file để lưu frame hiện tại sau khi PAUSE.
- `updateMovie(self, imageFile)` -> Cập nhật điểm PAUSE mới.
- `connectToServer(self)` -> Kết nối với server và mở RTSP/TCP session.
- `sendRtpRequest(self, requestCode)` -> Xử lý yêu cầu và gửi cho server.
- `recvRtpReply(self)` -> Nhận phản hồi từ server.
- `parseRtpReply(self, data)` -> Phân tích phản hồi của server để chuyển trạng thái của client.
- `openRtpPort(self)` -> Mở socket RTP sử dụng RTP port được cung cấp bởi client.

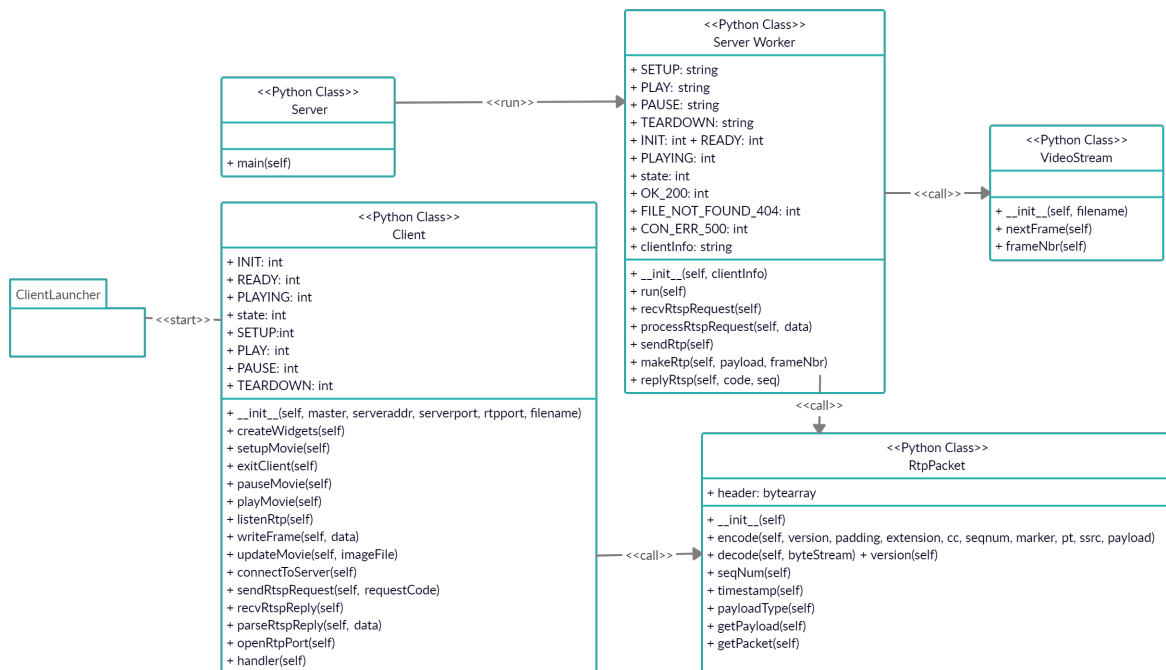
### RtpPacket.py:

- `encode(self, version, padding, extension, cc, seqnum, marker, pt, ssrc, payload)` -> Điền các thông số và dữ liệu vào trường RTP header.
- `decode(self, byteStream)` -> Giải mã gói RTP.
- `version(self)` -> Trả về phiên bản của RTP.
- `seqNum(self)` -> Trả về số frame.
- `timestamp(self)` -> Trả về timestamp.
- `payloadType(self)` -> Trả về kiểu payload.
- `getPayload(self)` -> Trả về payload.
- `getPacket(self)` -> Trả về gói RTP.

## ServerWorker.py:

- `recvRtspRequest(self)` -> Nhận yêu cầu RTSP từ client.
- `processRtspRequest(self, data)` -> Phân tích yêu cầu của client.
- `sendRtp(self)` -> Gửi gói RTP qua UDP.
- `makeRtp(self, payload, frameNbr)` -> Tạo một gói RTP.
- `replyRtsp(self, code, seq)` -> Gửi phản hồi RTSP cho client.

## 5 Class Diagram



## 6 Yêu cầu - phản hồi của Client - Server

- **SETUP**

– Client:

```
Data sent:
SETUP movie.Mjpeg RTSP/1.0
CSeq: 1
Transport: RTP/UDP; client_port= 20480
```

– Server:

```
processing SETUP

RTSP/1.0 200 OK
CSeq: 1
Session: 994243
```

- **PLAY**

– Client:

```
Data sent:
PLAY movie.Mjpeg RTSP/1.0
CSeq: 2
Session: 647160
Current Seq Num: 1
Current Seq Num: 2
Current Seq Num: 3
```



– Server:

```
processing PLAY  
  
RTSP/1.0 200 OK  
CSeq: 2  
Session: 647160
```

- **PAUSE**

– Client:

```
Data sent:  
PAUSE movie.Mjpeg RTSP/1.0  
CSeq: 3  
Session: 647160
```

– Server:

```
processing PAUSE  
  
RTSP/1.0 200 OK  
CSeq: 3  
Session: 647160
```



- TEARDOWN

– Client:

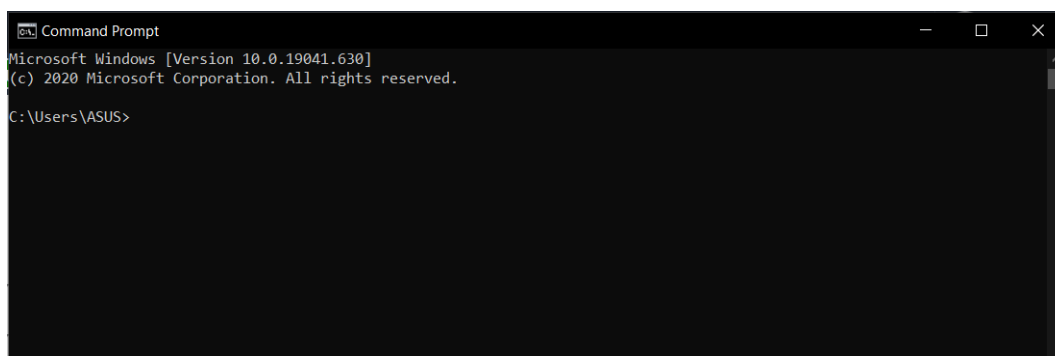
```
Data sent:  
TEARDOWN movie.Mjpeg RTSP/1.0  
CSeq: 5  
Session: 647160
```

– Server:

```
processing TEARDOWN  
  
RTSP/1.0 200 OK  
CSeq: 5  
Session: 647160
```

## 7 Hướng dẫn sử dụng

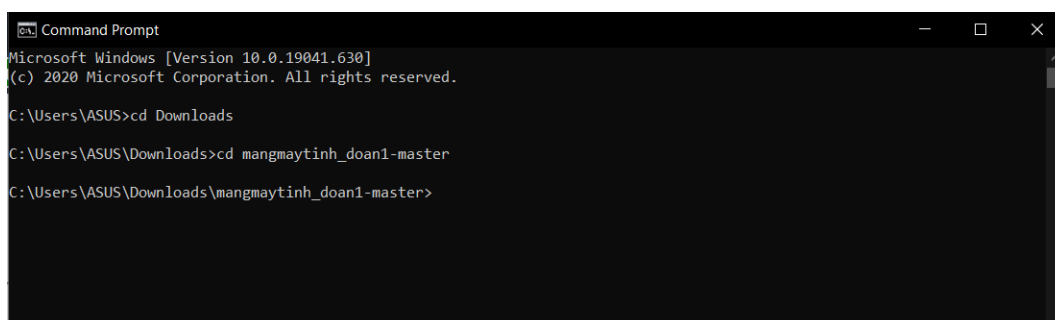
- Khởi động Server
  - Mở terminal



```
Command Prompt
Microsoft Windows [Version 10.0.19041.630]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\ASUS>
```

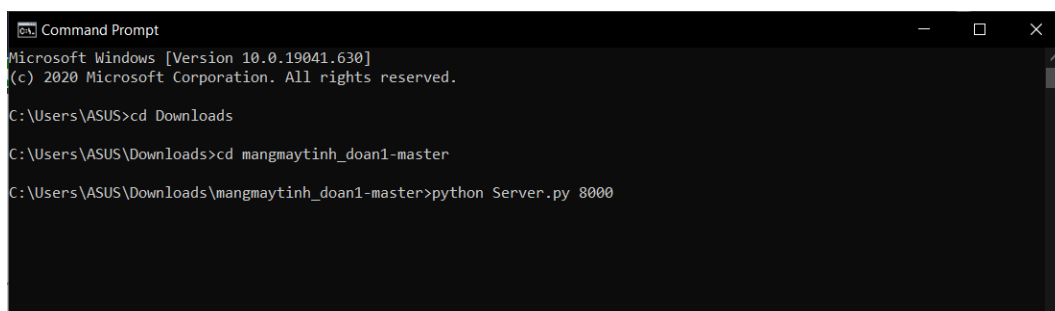
- cd vào thư mục chứa source code



```
Command Prompt
Microsoft Windows [Version 10.0.19041.630]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\ASUS>cd Downloads
C:\Users\ASUS\Downloads>cd mangmaytinh_doan1-master
C:\Users\ASUS\Downloads\mangmaytinh_doan1-master>
```

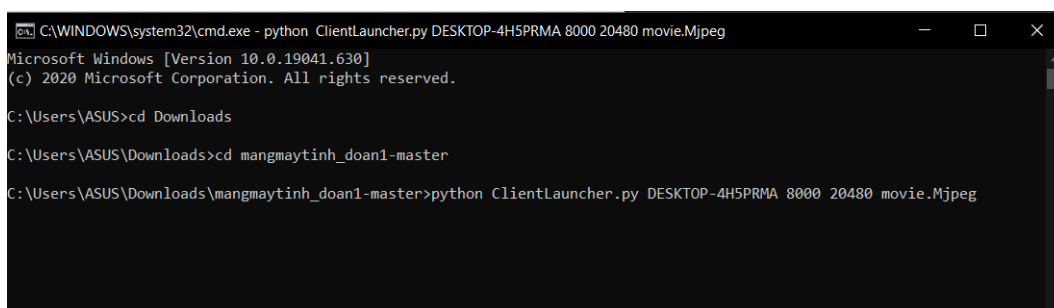
- Chạy lệnh "*python Server.py server\_port*", lưu ý chọn port lớn hơn 1024.



```
Command Prompt
Microsoft Windows [Version 10.0.19041.630]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\ASUS>cd Downloads
C:\Users\ASUS\Downloads>cd mangmaytinh_doan1-master
C:\Users\ASUS\Downloads\mangmaytinh_doan1-master>python Server.py 8000
```

- Khởi động Client
  - Mở Terminal.
  - cd vào thư mục chứa source code.
  - Chạy lệnh "`python ClientLauncher.py server_host server_port RTP_port video_file`"
    - \* `server_host` là hostname (gõ lệnh `hostname` trong terminal để kiểm tra hostname).
    - \* `server_port` trùng với port Server.
    - \* `RTP_port` là port RTP trong bài assignment này là 20480.
    - \* `video_file` gói video cần gửi đi.



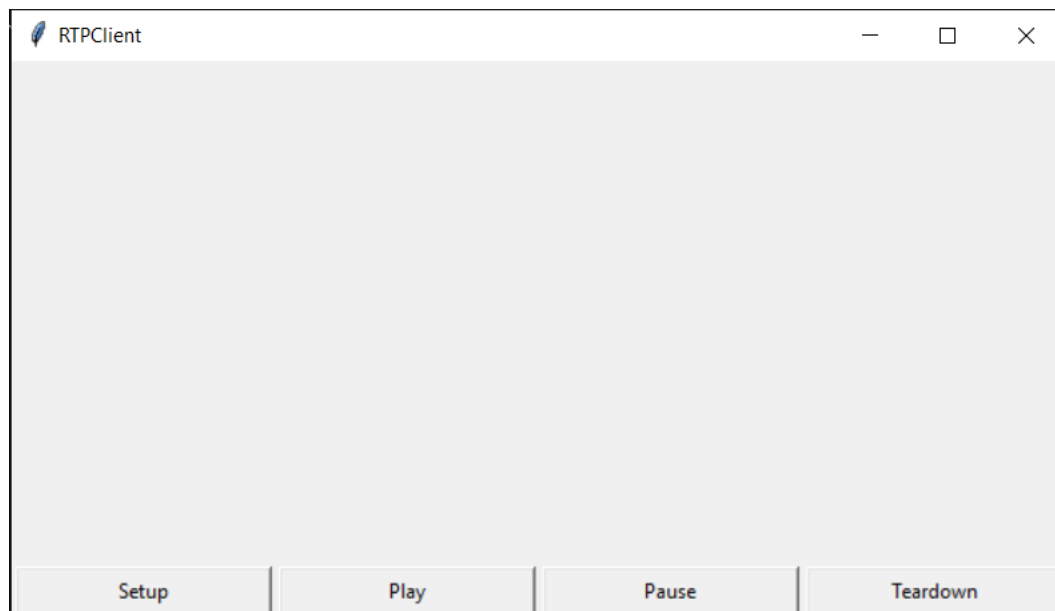
```
C:\WINDOWS\system32\cmd.exe - python ClientLauncher.py DESKTOP-4H5PRMA 8000 20480 movie.Mjpeg
Microsoft Windows [Version 10.0.19041.630]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\ASUS>cd Downloads

C:\Users\ASUS\Downloads>cd mangmaytinh_doan1-master

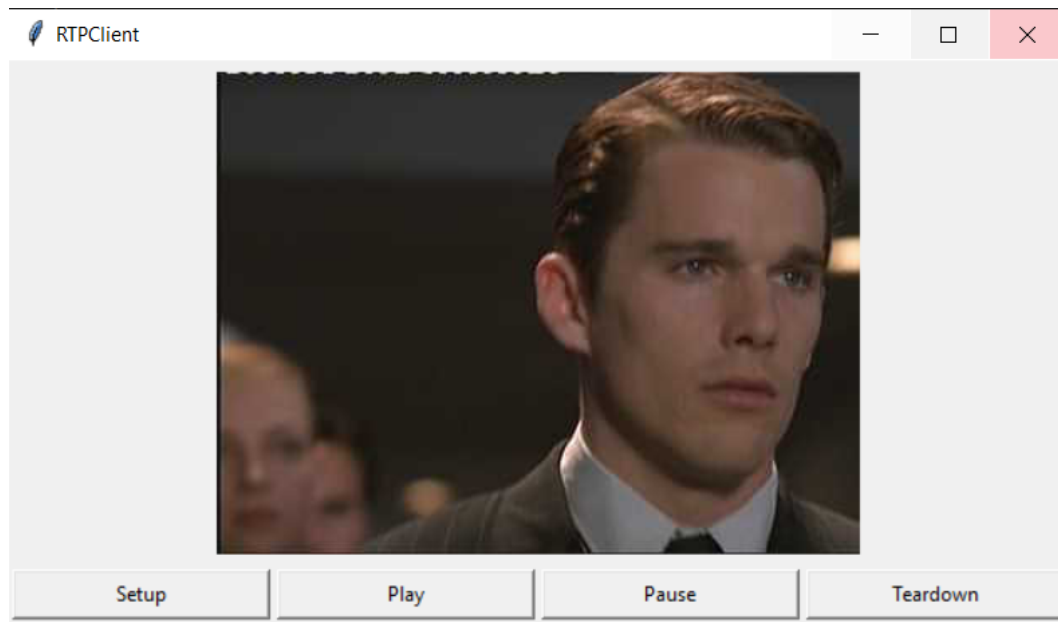
C:\Users\ASUS\Downloads\mangmaytinh_doan1-master>python ClientLauncher.py DESKTOP-4H5PRMA 8000 20480 movie.Mjpeg
```

- Giao diện xuất hiện



## 8 Đánh giá kết quả hiện thực

- Các chức năng hoạt động ổn định.
- Tốc độ phản hồi giữa Client - Server nhanh.
- Video được phát ổn định, không có hiện tượng giật lag.
- Không bị mất gói trong quá truyền gói.





## 9 Tham khảo

- <https://github.com/gabrieljablonski/rtsp-rtp-stream>
- [https://vi.wikipedia.org/wiki/Real\\_Time\\_Streaming\\_Protocol](https://vi.wikipedia.org/wiki/Real_Time_Streaming_Protocol)
- [https://en.wikipedia.org/wiki/Real-time\\_Transport\\_Protocol](https://en.wikipedia.org/wiki/Real-time_Transport_Protocol)
- <https://github.com/anki24/Video-Streaming-with-RTSP-and-RTP>