

BÁO CÁO BÀI TẬP LỚN 1

Môn Mạng Máy Tính

DANH SÁCH THÀNH VIÊN NHÓM 1	
La Quốc Anh (C)	1710465
Cao Văn Tín	1814334
Nguyễn Hồ Phước Lộc	1812969
Phạm Bùi Hải Thanh	1449529

Giảng viên hướng dẫn: Nguyễn Hồng Nam

Đại Học Quốc Gia Thành Phố Hồ Chí Minh
Trường Đại Học Bách Khoa



Mục Lục

0. Bảng Phân Chia Công Việc	2
1. Requirements Analysis	3
2. Function Description	4
3. Class Diagram	6
4. A Summative Evaluation of Results Achieved	7
5. User Manual	10
6. Source Code and References	11

0. Bảng Phân Chia Công Việc

La Quốc Anh	Chia việc Hiện thực code Evaluation of Results Achieved User Manual Tổng hợp và làm báo cáo
Cao Văn Tín	Requirements Analysis
Nguyễn Hồ Phước Lộc	Class Diagram
Phạm Bùi Hải Thanh	Function Description

1. Requirements Analysis

1.1. Introduction

Truyền thông đa phương tiện đang dần trở nên rất quan trọng khi ta đang chứng kiến sự phát triển khủng khiếp của các ứng dụng này trên Internet.

Công nghệ xem trực tiếp giúp ta có thể xem video mà không cần tải toàn bộ video.

RTSP là một giao thức sẽ được dùng để giao tiếp giữa máy khách (client) và máy chủ (server)

1.2. Giao thức truyền tin thời gian thực (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, dừng 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).

1.3. Giao Thức Truyền Tải Thời Gian Thực (RTP)

Real time transfer protocol (RTP), tiếng Việt gọi là **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

1.4. Phân Tích Các Module

Client, ClientLauncher

Hai lớp này sẽ hiện thực giao diện người dùng và các chức năng của máy khách

ServerWorker, Server

Hai lớp này sẽ hiện thực chức năng của máy chủ, phản hồi lại các lệnh RTSP và phát video

RtpPacket

Lớp này được dùng để xử lý các gói RTP (gồm encode, decode)

VideoStream

Lớp này được dùng để đọc video từ file nạp vào

1.5. Mục Tiêu

- Hiện thực giao thức RTSP ở máy khách
- Hiện thực đóng gói RTP ở máy chủ

2. Function Description

Client:

- createWidget: tạo giao diện chương trình, gồm các nút bấm và nơi hiển thị video
- setupMovie: gửi lệnh RTSP SETUP tới máy chủ
- exitClient: gửi lệnh RTSP TEARDOWN tới máy chủ, và thoát chương trình
- pauseMovie: gửi lệnh RTSP PAUSE tới máy chủ để dừng video
- playMovie: gửi lệnh RTSP PLAY tới máy chủ để bắt đầu phát video
- listenRtp: nhận và decode các gói RTP, cập nhật lại frame number hiện tại
- writeFrame: viết dataframe của video vào file ảnh
- updateMovie: hiển thị file ảnh lên giao diện người dùng
- connectToServer: kết nối với máy chủ
- sendRtspRequest: gửi lệnh RTSP đến máy chủ
- recvRtspReply: nhận phản hồi RTSP từ máy chủ, decode dữ liệu và gửi data đến hàm đọc, đồng thời tắt chương trình nếu lệnh là TEARDOWN
- parseRtspReply: xử lý dữ liệu nhận được từ dữ liệu đã được decode
- openRtpPort: tạo cổng kết nối giữa máy khách và máy chủ
- handler: xử lý khi người dùng bấm vào nút [X] ở góc trên

RtpPacket:

- encode: encode các thông số vào header theo đúng định dạng, update lại header và payload
- decode: tách gói RTP thành 2 phần header và payload
- version: trả về phiên bản của gói RTP
- seqNum: trả về sequence number của gói
- timestamp: trả về timestamp
- payloadType: trả về kiểu của payload
- getPayload: trả về payload
- getPacket: trả về packet (gồm header và payload)

ServerWorker:

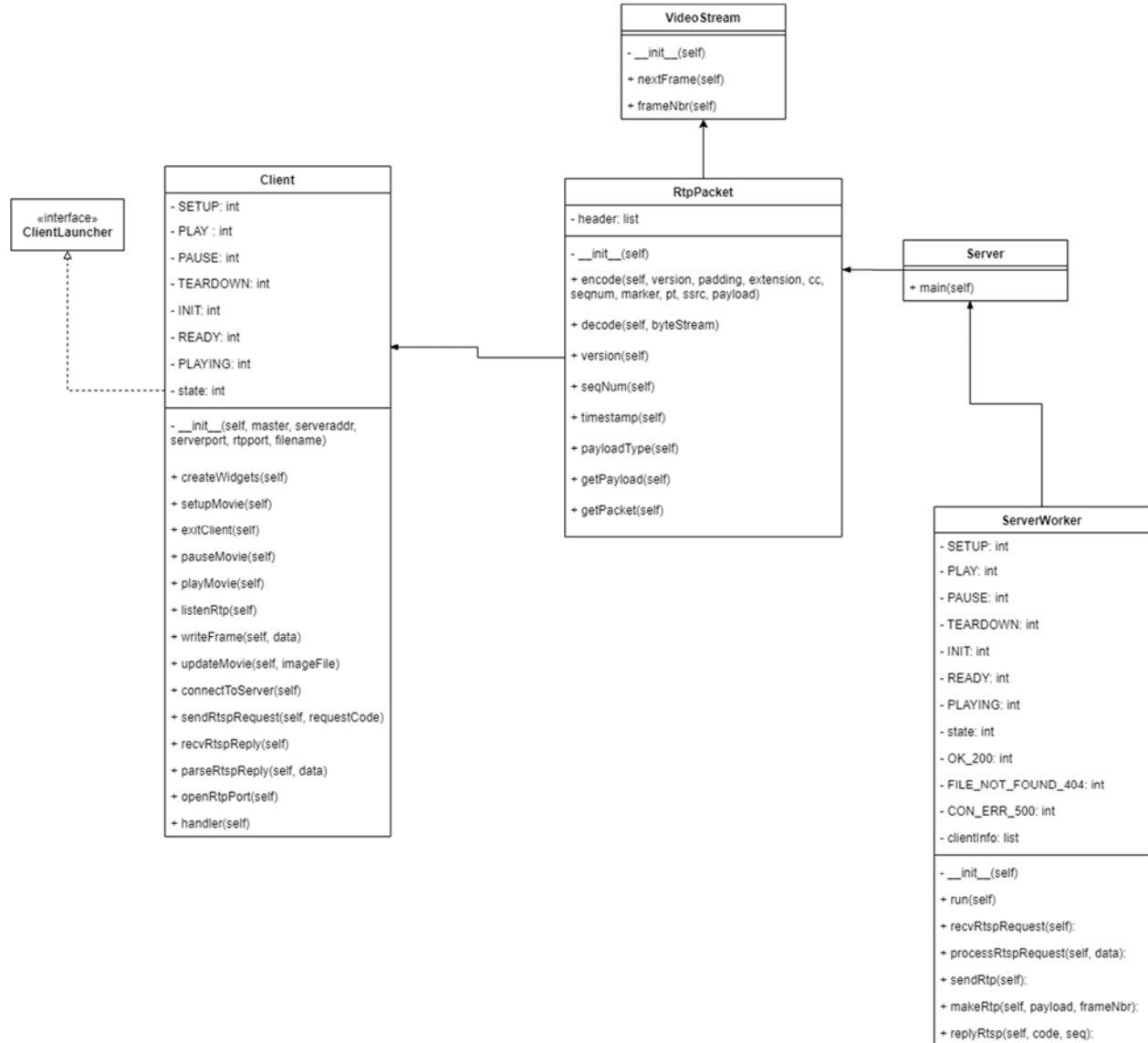
- run: chạy server
- recvRtspRequest: chờ nhận request từ máy khách, decode dữ liệu và gửi đi xử lý

- processRtspRequest: xử lý dữ liệu request
- sentRtp: gửi các gói RTP đến máy khách
- makeRtp: dùng encode để đóng gói header và payload, gửi đến máy khách. Trả về gói RTP vừa được tạo
- replyRtsp: phản hồi thông tin về máy khách thông qua terminal ("200 OK" nếu thành công, "404 NOT FOUND" khi không tìm thấy file, "500 CONNECTION ERROR" nếu bị lỗi kết nối)

VideoStream:

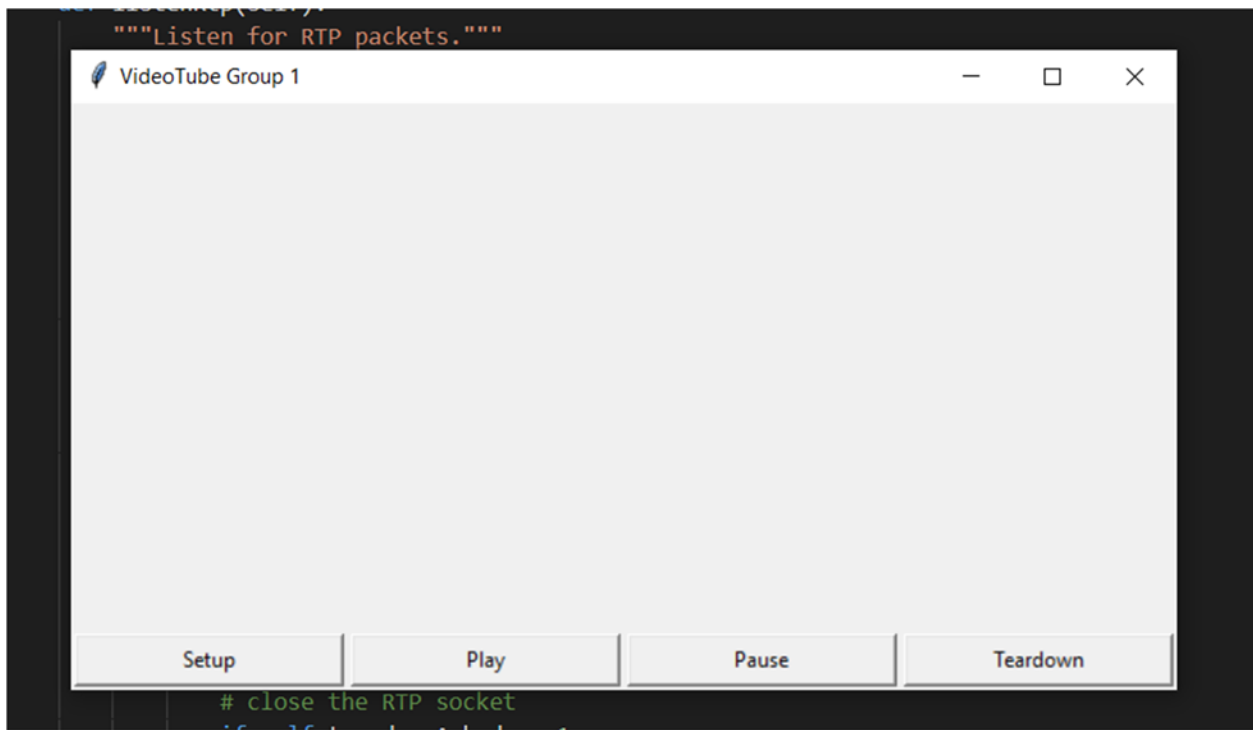
- nextFrame: trả về frame tiếp theo của Video
- frameNbr: trả về frame number hiện tại

3. Class Diagram



4. A Summative Evaluation of Results Achieved

Giao diện:



- Khi bấm nút Setup:

Server terminal:

```
Data received:
SETUP movie.Mjpeg RTSP/1.0
CSeq: 1
Transport: RTP/UDP; client_port= 5000
processing SETUP
```

Client terminal:

```
Data sent:
SETUP movie.Mjpeg RTSP/1.0
CSeq: 1
Transport: RTP/UDP; client_port= 5000
```


- Khi bấm nút Play

Server terminal:

```
Data received:
PLAY movie.Mjpeg RTSP/1.0
CSeq: 2
Session: 304624
processing PLAY
```

Client terminal:

```
Data sent:
PLAY movie.Mjpeg RTSP/1.0
CSeq: 2
Session: 137289
Current Seq Num: 1
Current Seq Num: 2
Current Seq Num: 3
Current Seq Num: 4
Current Seq Num: 5
Current Seq Num: 6
```

- Khi bấm nút Pause

Server terminal:

```
Data received:
PAUSE movie.Mjpeg RTSP/1.0
CSeq: 3
Session: 137289
processing PAUSE
```

Client terminal:

```
Data sent:
PAUSE movie.Mjpeg RTSP/1.0
CSeq: 3
Session: 137289
□
```

- Khi bấm nút Teardown

Server terminal:

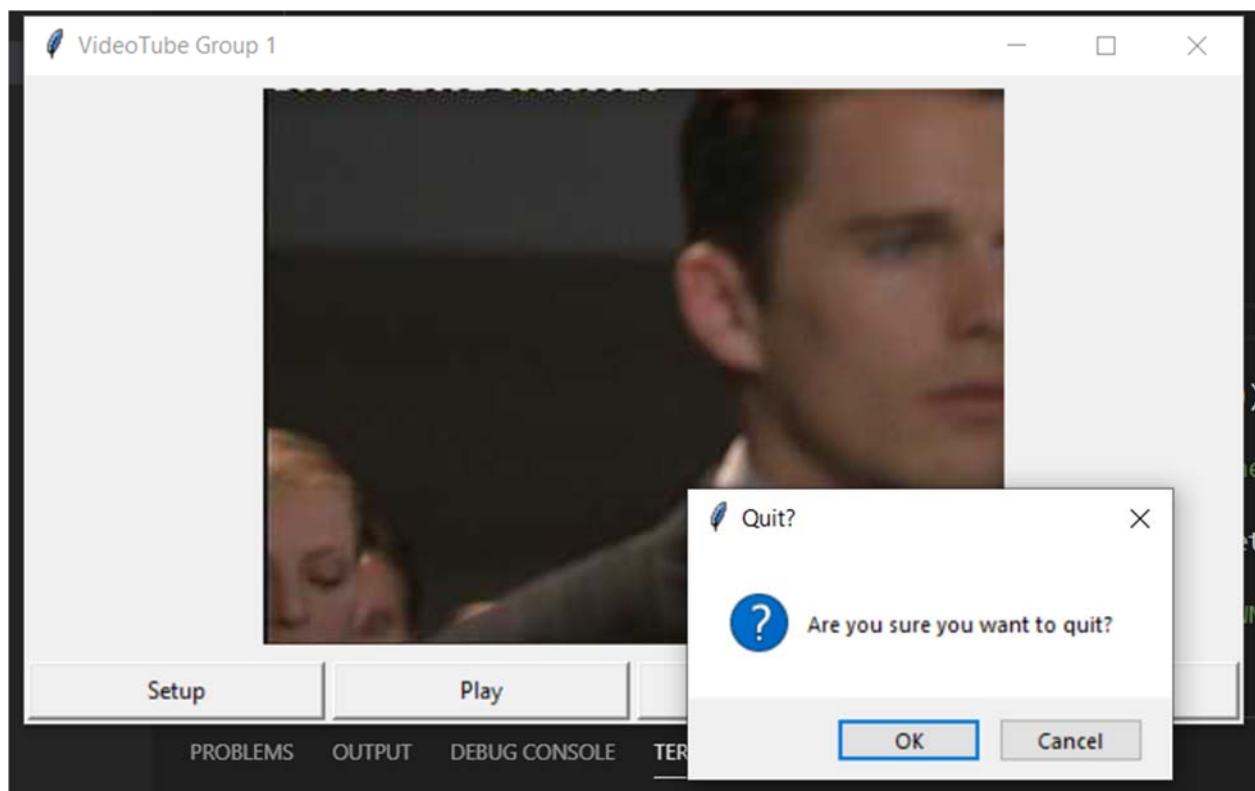
```
Data received:  
TEARDOWN movie.Mjpeg RTSP/1.0  
CSeq: 12  
Session: 137289  
processing TEARDOWN
```

Client terminal:

```
Data sent:  
TEARDOWN movie.Mjpeg RTSP/1.0  
CSeq: 12  
Session: 137289
```

- Khi bấm nút [X] bên góc trên:

Client sẽ gửi lệnh Pause và hiển thị Popup:



Bấm OK, Client sẽ gửi lệnh Teardown đến server và thoát chương trình

Bấm Cancel, Client sẽ gửi lệnh Play đến server

* Khi bấm Teardown hoặc [X] trước khi Play, xuất hiện lỗi do chưa có file ảnh nào được lưu vào cache

```
Exception in Tkinter callback
Traceback (most recent call last):
  File "C:\Users\anh.lq\AppData\Local\Programs\Python\Python38-32\lib\tkinter\_init_.py", line 1883, in __call__
    return self.func(*args)
  File "C:\Users\anh.lq\Desktop\Students\Client.py", line 77, in exitClient
    os.remove(CACHE_FILE_NAME + str(self.sessionId) + CACHE_FILE_EXT) # Delete the cache image from video
FileNotFoundError: [WinError 2] The system cannot find the file specified: 'cache-0.jpg'
```

5. User Manual

1. Chạy server:

```
python Server.py <server_port>
```

với <server_port> là cổng nghe của server

Ví dụ: `python Server.py 8000`

2. Chạy client:

```
python ClientLauncher <server_host> <server_port> <RTP_port> <video_file>
```

với <server_host> là tên của máy đang chạy server, <server_port> là cổng mà server đang nghe, <RTP_port> là cổng để nhận gói RTP, <video_file> là tên file video muốn gửi lệnh

Ví dụ: `python ClientLauncher.py TTG-023B 8000 5000 movie.Mjpeg`

3. Dừng chương trình

Chương trình có 4 chức năng chính:

- Setup: Khởi tạo video muốn xem
- Play: Chạy video
- Pause: Tạm dừng video, bấm Play để resume
- Teardown: Kết thúc video và thoát chương trình

6. Source Code and References

- Source code: https://github.com/Quoc-Anh-17/computernetworking_assignment1
- References:
 - RTP (https://vi.wikipedia.org/wiki/Real-time_Transport_Protocol)
 - RTSP (https://vi.wikipedia.org/wiki/Real_Time_Streaming_Protocol)
 - Socket Programming in Python (<https://realpython.com/python-sockets>)