VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY
UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

COMPUTER NETWORKS (CO3094)

Assignment

Real-Time Streaming Protocol (RTSP) and Real-time Transfer Protocol (RTP)

Advisor: Nguyễn Lê Duy Lai
Students: Phan Văn Sỹ - 1852721
Phạm Văn Minh Toàn - 1953028
Tiêu Viết Trọng Nghĩa - 1852611
Nguyễn Trung Nguyên - 1852619

HO CHI MINH CITY, NOVEMBER 2021

Contents

1 Member list & Workload **2**

2 Requirements analysis **2**
2.1 Functional requirements 2
2.1.1 System-side 2
2.1.2 User-side 2
2.2 Non-functional requirements 2

3 Description of the functions' tasks **2**

4 Class diagram **4**

5 Result **5**

6 User manual **5**

1 Member list & Workload

No.	Fullname	Student ID	Percentage of work
1	Phan Văn Sỹ	1852721	25%
2	Phạm Văn Minh Toàn	1953028	25%
3	Nguyễn Trung Nguyên	1852619	25%
4	Tiêu Viết Trọng Nghĩa	1852611	25%

2 Requirements analysis

2.1 Functional requirements

2.1.1 System-side

- The system can stream video.
- The system can communicate with user via RTSP/RTP protocol.

2.1.2 User-side

- User can connect to the server via the terminal.
- User can play video from server, pause and teardown.
- User can view the basic parameters of the video such as the time of the video.

2.2 Non-functional requirements

- The extension of the video must be .Mjpeg
- Response time from server less than or equal 0.5s

3 Description of the functions' tasks

Function	Description
run(self)	Run the server
processRtspRequest(self, data)	Process the RTSP request
sendRtp(self)	Send RTP packets to client
makeRtp(self, payload, frameNbr)	Make RTP packet
replyRtsp(self, code, seq)	Reply to the Client
__init__(self, clientInfo)	Constructor

Function description of ServerWorker class

Function	Description
__init__(self)	Constructor
encode(self,version,padding,extension,cc,seqnum, market, pt, ssrc, payload)	Encode RTP packet into bytearray
decode(self, byteStream)	Decode the RTP packet
version(self)	Return RTP version
seqNum(self)	Return sequence frame number
timestamp(self)	Return timestamp
payloadType(self)	Return payload type
getPayload(self)	Return payload
getPacket(self)	Return RTP packet

Function description of RTP clas

Function	Description
init(self, master, serverAddr, serverPort, rtpPort, filename)	Constructor
createWidgets(self)	Init GUI
setupMovie(self)	Event handler for SETUP button
exitClient(self)	Event handler for TEARDOWN button
pauseMovie(self)	Event handler for PAUSE button
playMovie(self)	Event handler for PLAY button
listenRtp(self)	Listen to RTP port and handle any incoming packets
writeFrame(self, data)	Write the receive frame to a image file
updateMovie(self, imageFile)	UPdate the image file to the GUI
connectToServer(self)	Connect to the server
sendRtspRequest(self, requestCode)	Send RTSP request to server with request code
recvRtspRequest(self)	Listen to RTSP reply from the server
parseRtspReply(self, data)	Parse the RTSP reply from the server
openRtpPort(self)	Open RTP socket to receive RTP packets
handler(self)	Handle exceptions when closing the GUI

Function description of Client class

Function	Description
__init__(self, fileName)	Constructor
nextFrame(self)	Get next frame
frameNbr(self)	Get frame number

Function description of VideoStream class

4 Class diagram

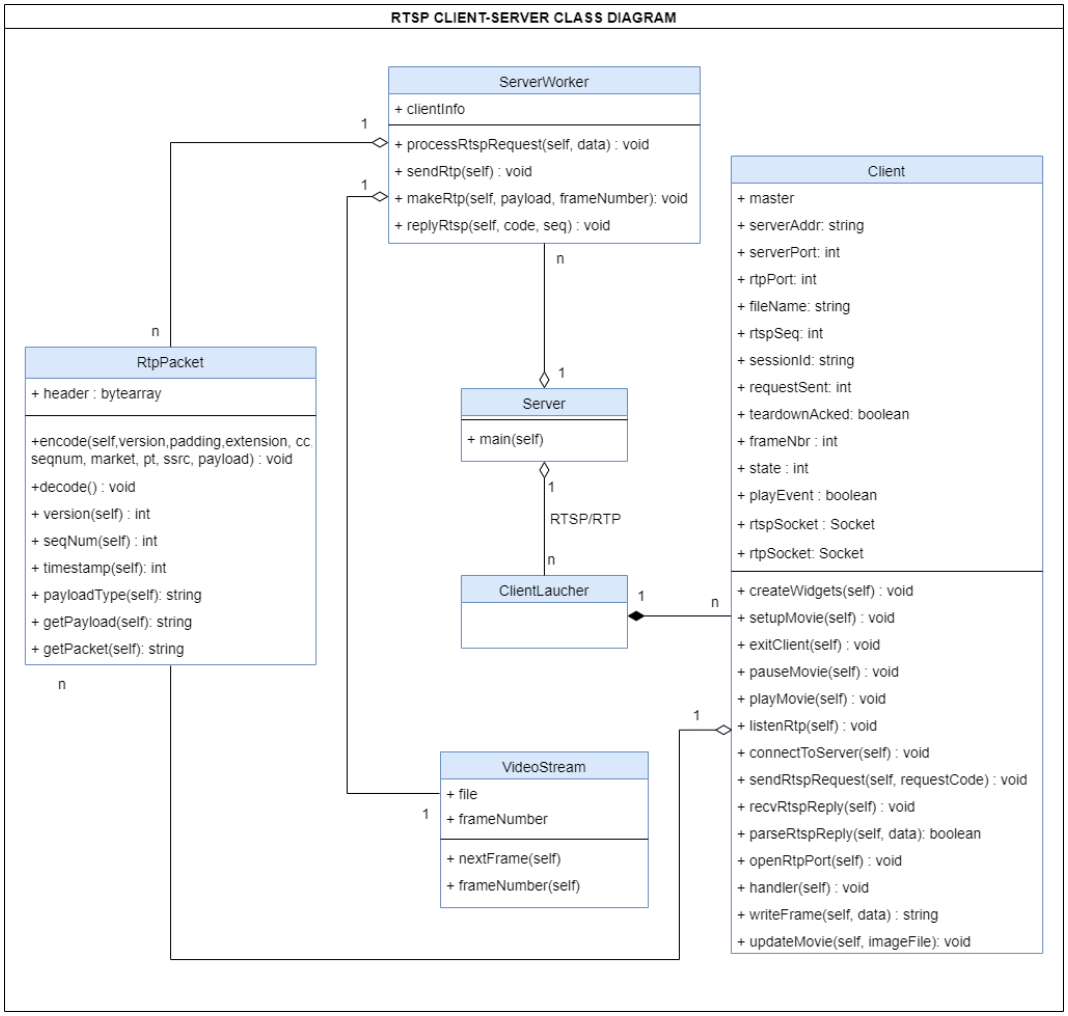


Figure 2: Class diagram

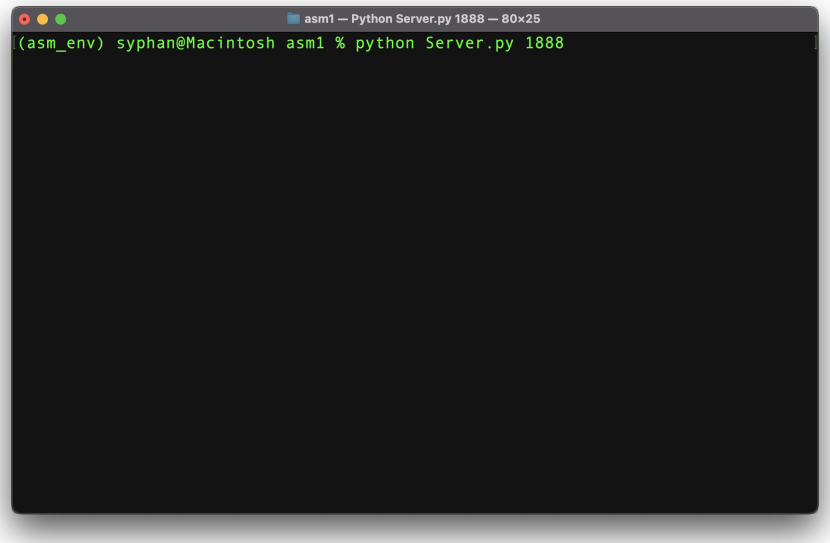
5 Result

- Completing the RTSP protocol at the client
- Complete RTP protocol at server

6 User manual

- Step 01: We must run the server first: run the terminal in the directory containing the file Server.py

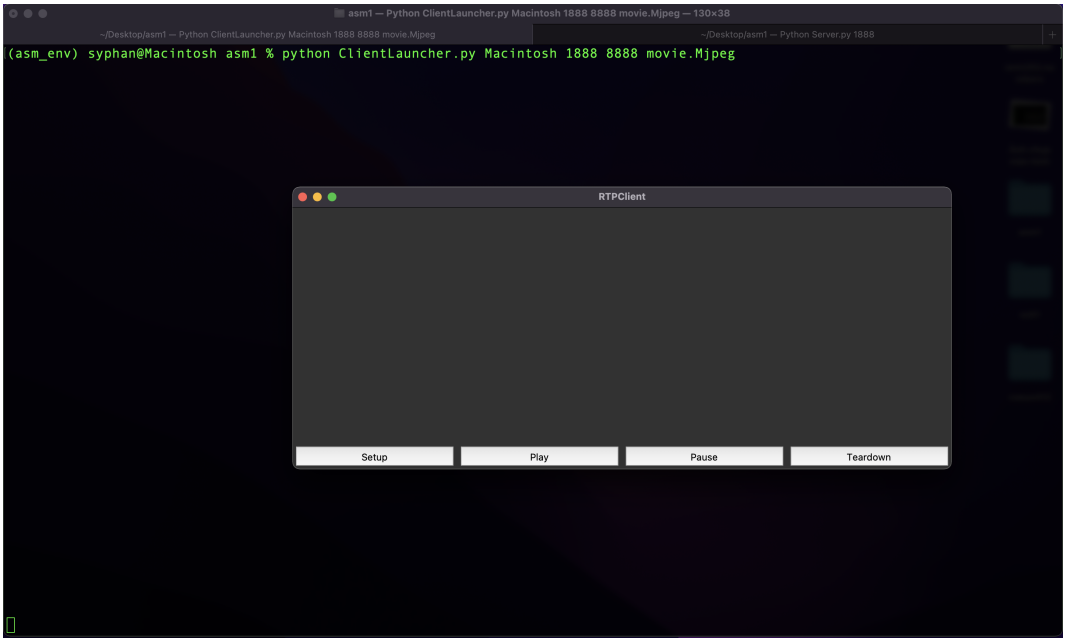
  Run command in form: python Server.py 1888 (Where port_server should greater than 1024).



- Step 02: We open a new terminal in the folder containing the ClientLauncher.py file to connect to the Server we opened in step 01.

  Run command in form: python ClientLauncher.py Macintosh 1888 8888 movie.Mjpeg

  - «host_name»: is the IP of the Server on the computer you are using, here is "192.168.2.7"
  - «port_server»: is the port initialized in step 1, here is 1200
  - «port_RTP»: for example, we choose 5008
  - «name_video»: the name of the video, here is movie.Mjpeg



- Step 03: Click **Setup** to create RTP stream and press **Play** to watch video, **Pause** to stop and **Teardown** to finish.