Ho Chi Minh City University of Technology

Faculty of Computer Science and Engineering



**Mạng Máy Tính**

**Report**

**ASSIGNMENT 1- Streaming Video**

Instructor: Hoàng Nguyễn Minh Đức

Student:

Mai Nguyễn Minh Hiếu - 1812171

--

--

Mục Lục

[**1.Phân tích yêu cầu**](#_lrwodg4205ac) **2**

[**2. Mô tả các hàm chức năng**](#_ycqh14n64tw5) **6**

[**3. Mô hình Class diagram**](#_gvhs2cdu1czw) **6**

[**4. Đánh giá tổng hợp các kết quả đạt được**](#_gwqwprq7u1is) **6**

[**5. Hiện thực và Source Code**](#_1z8hiu8yyrwd) **6**

# 

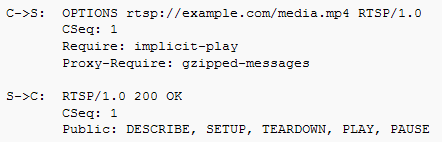
# 1.Phân tích yêu cầu

streaming video là quá trình chia nhỏ file video thành các frame, rồi lần lượt gửi từng frame tới một bộ đệm trên máy tính của người xem và hiển thị nội dung frame đó. Và quá trình này tuân thủ chặt chẽ về ràng buộc theo thời gian, nói khác là tuân thủ chặt chẽ theo giao thức RTSP, RTP và RTCP.

Giao thức RTSP

Để thực hiện kỹ thuật streaming video theo giao thức RTSP nhất thiết máy client phải gửi lên máy server ( streaming server) những request sau và phải theo một trình tự nhất định.

Đầu tiên, máy client sẻ gửi yêu cầu OPTIONS kèm với đường link trỏ tới file video cần xem tới máy server, để máy server chấp nhận đường link này.



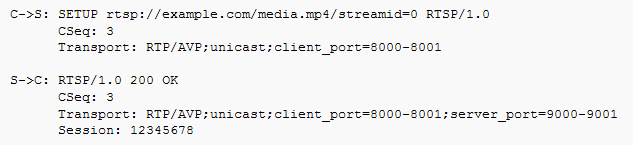
**OPTIONS Request**

Nếu máy server trả về mã chấp nhận đường link trên thì máy client tiếp tục gửi yêu cầu DESCRIBE tới máy server để máy server phân tích đường link. Một yêu cầu DESCRIBE bao gồm một đường link RTSP có dạng (rtsp:// ) và kiểu dữ liệu đáp trả từ phía server. Cổng mặc định được sử dụng cho giao thức RTSP là 554 và cổng này được sử dụng cho cả giao thức của tầng giao vận UDP và TCP. Thông điệp đáp lại từ máy server cho yêu cầu DESCRIBE của máy client bao gồm bản tin miêu tả chi tiết phiên giao dịch( Session Description Protocol – SDP). Ngoài ra trong thông điệp trả về từ máy server còn liệt kê các đường link thích hợp hơn tới file video cần chơi khi mà trong file video đó có trộn lẫn giữa phụ đề và âm thanh. Và điều quan trọng nhất ở trong bản tin miêu tả phiên giao dịch này là streamid của luồng video và streamid của luồng âm thanh khi mà đoạn video đó có lồng âm thanh vào trong các frame.



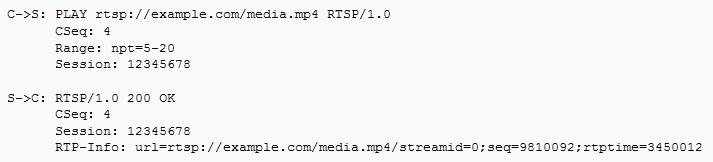
**DESCRIBE Request**

Sau khi máy client nhận được thông điệp đáp trả từ máy server sau yêu cầu DESCRIPTION thì máy client sẽ tiếp tục gửi tiếp yêu cầu SETUP tới máy server. Một yêu cầu SETUP sẽ chỉ ra cách mà một dòng dữ liệu ( single media stream ) bắt buộc phải được truyền đi như thế nào. Và yêu cầu SETUP bắt buộc phải được hoàn thành trước khi một yêu cầu PLAY được gửi từ máy client. Yêu cầu SETUP bao gồm một đường link tới file video cần streaming và một thông tin đặc tả cho phần giao vận. Đặc tả này bao gồm 2 cổng trong đó có một cổng cục bộ trên máy client dành cho việc nhận cac gói tin RTP (audio và video) và cổng còn lại dùng để nhận các gói tin RTCP ( meta information ). Máy server sẽ đáp trả lại bằng các xác nhận các tham số đã được lựa chọn, và điền vào các phần còn thiếu ví dụ như máy server có thể chọn lại cổng của mình. Mỗi luồng dữ liệu sẽ được cấu hình cụ thể sau khi yêu cầu SETUP được hoàn tất trước khi máy client gửi yêu cầu PLAY.



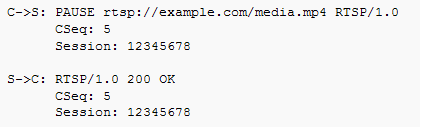
**SETUP Request**

Sau khi hoàn tất yêu cầu SETUP, cấu hình được các luồng dữ liệu để chuẩn bị streaming, máy client sẽ gửi yêu cầu PLAY để thực hiện truyền các frame dữ liệu thật sự từ máy server tới máy client , và các frame dữ liệu này sẽ được lưu trong một bộ đệm của máy client, các frame này sẽ được giải mã ( decode ), rồi được hiển thị bởi trình chơi file video và âm thanh ( VLC). Yêu cầu PLAY bao gồm một đường dẫn trỏ tới file video cần phát giống như các yêu cầu trước đó. Đường link này có thể là đường tổng hợp ( để phát các luồng dữ liệu) hoặc là môt đường link đơn lẻ ( chỉ phát một luồng dữ liệu duy nhất ). Trong yêu cầu PLAY, máy client cũng sẽ chỉ ra một dải ( range) chỉ rõ một cách cụ thể số hiệu frame bắt đầu được gửi và số hiệu frame kết thúc, Nếu như không chỉ rõ tham số này, thì toàn bộ các frame sẽ được gửi tới máy client. Và nếu như luồng dữ liệu có bị tạm dừng ( pause) thì luồng dữ liệu này cũng sẽ được phục hồi ở frame mà nó tạm dừng truyền.



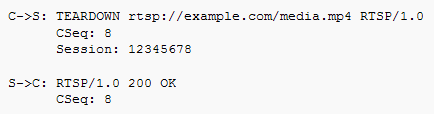
**PLAY Request**

Trong quá trình streaming video, nếu như người dùng muốn tạm dừng quá trình streaming thì sẽ gửi yêu cầu PAUSE tới máy server, yêu cầu này sẽ làm tạm dừng một hay nhiều luồng dữ liệu đang truyền các frame về máy client. Máy server sẽ tạm dừng gửi các frame dữ liệu tới máy client.



**PAUSE Request**

Trong quá trình streaming video, nếu như người dùng muốn dừng hẳn quá trình streaming thì sẽ gửi yêu cầu TEARDOWN để dừng truyền và kết thúc một phiên giao dịch của giao thức RTSP. Máy server sẽ đáp trả lại thông điệp xác nhận cho yêu cầu TEARDOWN và sẽ dừng gửi các frame tới máy client.



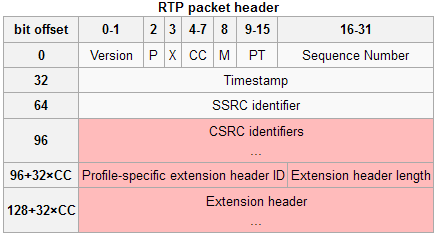
**TEARDOWN Request**

Giao thức RTP

Các giao thức thành phần: Đặc tả RTP gồm 2 giao thức con là RTP và RTCP

Một phiên RTP được thiết lập cho mỗi luồng dữ liệu. Một phiên bao gồm một địa chỉ IP với một cặp cổng của giao thức RTP và RTCP. Ví dụ, các luồng video và audio sẽ có các phiên RTP khác nhau, bên nhận sẽ nhận một cách riêng biệt giữa dữ liệu video và audio thông qua 2 cổng khác nhau cho 2 giao thức RTP và RTCP. Thường thì số hiệu cổng của RTP là một số chẵn trong khoảng 1024 tới 65535 và cổng của RTCP là một số lẻ kế tiếp.

Hình vẽ dưới đây là hình ảnh của một header của gói tin RTP



**Header của RTP Packet**

Kích thước nhỏ nhất của một header của gói tin RTP là 12 bytes. Sau phần header chính, là phần header mở rộng và không cần thiết phải có phần header này. Chi tiết các trường trong một header như sau:

* Version ( 2 bits): Cho biết phiên bản của giao thức này. Phiên bản hiện tại là phiên bản 2.
* P (Padding) (1 bit) : Cho biết số các byte mở rộng cần thêm vào cuối của gói tin RTP. Ví dụ trong trường hợp ta muốn sử dụng các thuật toán mã hóa, ta có thể thêm vào một số byte vào phần kết thúc của gói tin để tiến hành mã hóa frame trên đường truyền.
* X ( Extension) ( 1bit): Cho biết có thêm phần header mở rộng vào sau phần header chính hay không.
* CC (CSRC Count) ( 4 bit) : Chứa con số định danh CSRC cho biết kích thước cố định của header.
* M ( Marker) ( 1 bit) : Cho biết mức của ứng dụng và được định nghĩa bởi một profile. Nếu được thiết lập, có nghĩa là dữ liệu hiện tại đã được tính toán chi phí một cách thích hợp
* PT (Payload Type) ( 7 bit) : Cho biết định dạng của file video. Đây là một đặc tả được định nghĩa bởi một profile RTP.
* Sequence Number (16 bits) : số hiệu của frame. Và sẽ được tăng lên 1 đơn vị cho mỗi gói tin RTP trước khi gửi và được sử dụng bởi bên nhận để dò ra các gói bị lạc và có thể phục hồi lại gói có số thứ tự đó.
* Timestamp ( 32 bits): Được sử dụng thông báo cho bên nhận biết để phát lại frame này trong khoảng thời gian thích hợp.

SSRC ( 32 bits): Định danh cho nguồn streaming. Mỗi nguồn cho phép streaming video sẽ định danh bởi một phiên RTP duy nhất.

# 2. Mô tả các hàm chức năng

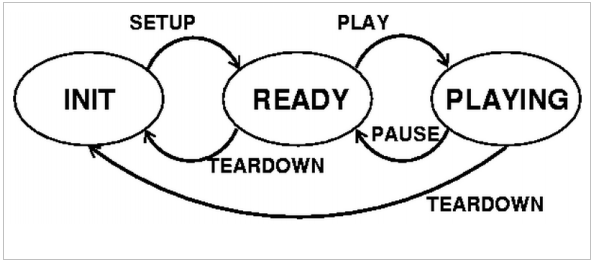
##

# 3. Mô hình Class diagram

##

# 4. Đánh giá tổng hợp các kết quả đạt được

hoàn thành được các mục được mô tả bao gồm các trạng thái được mô tả trong hình:



# 5. Hiện thực và Source Code

**Chương trình bao gồm 5 tệp:**

Client.py

ClientLauncher.py

RtpPacket.py

Server.py

ServerWorker.py

Những gì sẽ làm là:

**1.Run Server.py trên Server Terminal để khởi động máy chủ:**

Ví dụ: python Server.py server\_port

server\_port là cổng mà máy chủ của bạn lắng nghe các kết nối RTSP đến

# chúng ta có thể cung cấp cho nó giá trị 1025

# Cổng RTSP tiêu chuẩn là 554

# Trong dự án này, chúng tôi sẽ đặt giá trị> 1024

**2. Chạy ClientLauncher.py trên Client Terminal để bắt đầu một ứng dụng khách:**

Ví dụ: python ClientLauncher.py server\_host server\_port PRT\_port video\_file

server\_host là địa chỉ IP của máy cục bộ (chúng ta có thể sử dụng “ 127.0.0.1 ” )

server\_port là cổng mà máy chủ đang nghe (ở đây là “ 1025 ” )

RTP\_port là cổng nhận gói RTP (ở đây là “ 5008 ” )

video\_file là tên của tệp video mà chúng tôi muốn phát (ở đây là “ video.mjpeg ” )

**RTSP**

Giao thức phát trực tuyến thời gian thực

Đối với hệ thống giải trí và truyền thông để điều khiển các máy chủ phương tiện truyền trực tuyến

Thiết lập và kiểm soát các phiên truyền thông giữa các điểm cuối

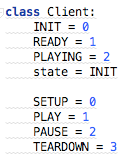
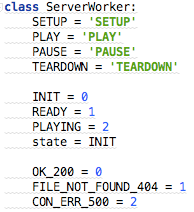
Nó sử dụng TCP

**RTP**

Giao thức vận tải thời gian thực

Giao thức mạng để phân phối âm thanh và video qua Mạng IP

Nó sử dụng UDP



Những gì sẽ được gửi từ máy khách đến máy chủ thông qua Giao thức RTSP là các lệnh như : SETUP , PLAY , PAUSE , TEARDOWN

Các lệnh này sẽ cho phía máy chủ biết hành động tiếp theo mà nó sẽ hoàn thành.

Những gì sẽ được trả lời từ máy chủ đến máy khách thông qua Giao thức RTSP là các tham số là : OK\_200

FILE\_NOT\_FOUND\_404

CON\_ERR\_500

Để cho khách hàng biết nếu máy chủ nhận được các lệnh của nó một cách chính xác

Sau khi khách hàng nhận được trả lời từ server, nó sẽ thay đổi trạng thái của nó để :

READY

PLAYING

Nếu lệnh SETUP được gửi từ máy khách đến máy chủ



Các “ SETUP ” RTSP gói sẽ bao gồm:

1. Lệnh SETUP

2.Tên tệp video sẽ phát

3.RTSP Số thứ tự gói bắt đầu từ 1

4. loại giao thức: RTSP / 1.0 RTP

5. Giao thức truyền: UDP

6. Cổng RTP để truyền dòng video

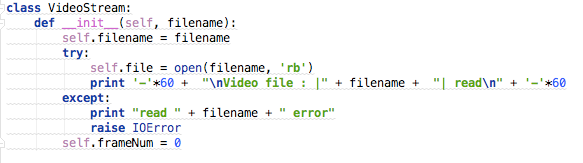


Khi Server side nhận lệnh “ SETUP ”, nó sẽ:

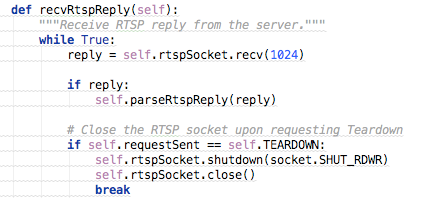
1. Gán cho khách hàng một Specific Session Number ngẫu nhiên

2.Nếu có gì đó sai với lệnh này hoặc trạng thái của máy chủ, nó sẽ trả lời ERROR trở lại máy khách

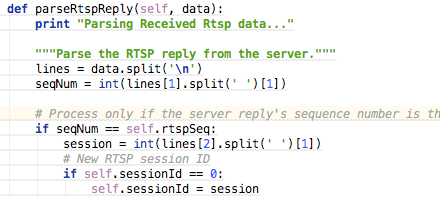
3.Nếu lệnh đúng,nó sẽ trả lời lại OK\_200 cho khách hàng và đặt STATE thành READY



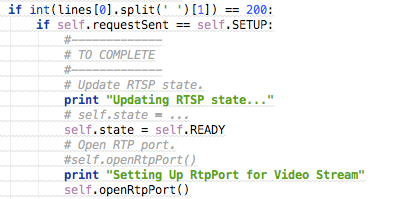
Máy chủ sẽ mở tệp video được chỉ định trong SETUP Packet và Khởi tạo số khung hình video của nó thành 0

Phía Client sẽ lặp lại để nhận RTSP Reply của Server

Sau đó, phân tích cú pháp gói RTSP Relpy: lấy Session Number



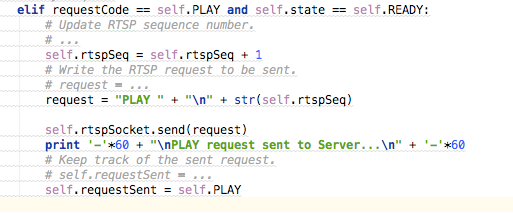
Và nếu Gói Trả lời phản hồi cho lệnh SETUP Khách hàng sẽ đặt STATE là READY



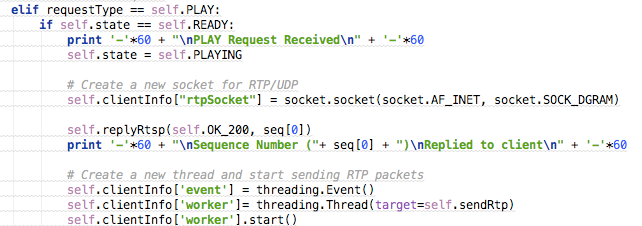
Sau đó, mở một Cổng Rtp để nhận video stream



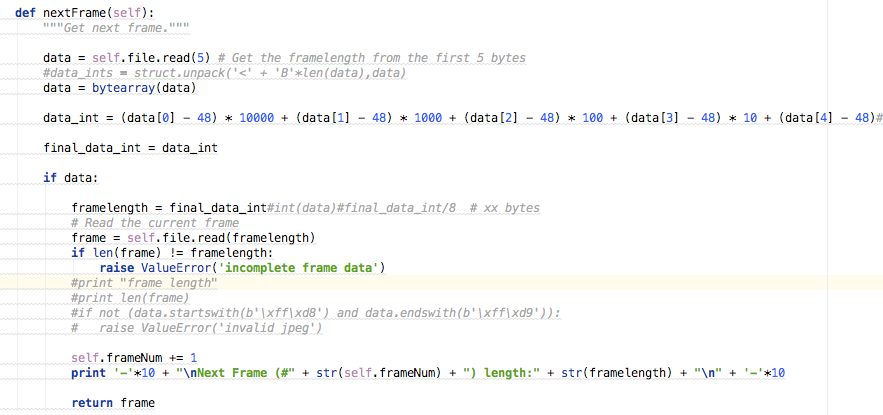
Sau đó.Nếu lệnh PLAY RTSP được gửi từ máy khách đến máy chủ:



Máy chủ sẽ tạo một Socket để truyền RTP qua UDP và bắt đầu một bước để gửi gói video stream

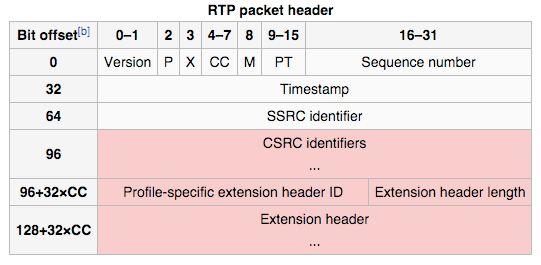


VideoStream.py sẽ giúp chia nhỏ tệp video thành từng khung riêng biệt và đưa từng khung vào gói dữ liệu RTP

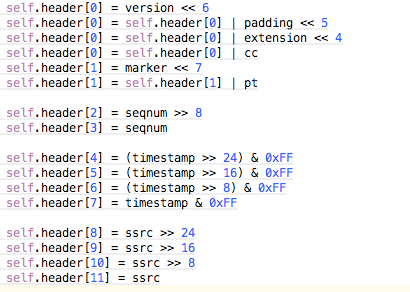


Mỗi gói dữ liệu cũng sẽ được mã hóa với một tiêu đề, tiêu đề sẽ bao gồm

RTP-version filed , Padding , extension , Contributing source , Marker , Type Field , Sequence Number , Timestamp,SSRC



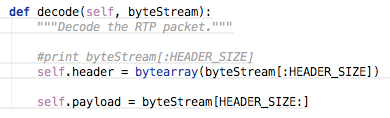
chúng đã được chèn vào Gói RTP thông qua các thao tác bitwise



Cuối cùng, Gói RTP sẽ bao gồm một tiêu đề và một khung video được gửi đến Cổng RTP ở phía máy khách:



Sau đó, Khách hàng giải mã Gói RTP để lấy tiêu đề và khung video, tổ chức lại các khung và hiển thị trên giao diện người dùng



Nếu lệnh PAUSE được gửi từ máy khách đến máy chủ, nó sẽ ngăn máy chủ gửi các khung hình video đến máy khách

Nếu một lệnh TEARDOWN được gửi từ máy khách đến máy chủ, nó cũng sẽ ngăn máy chủ gửi các khung hình video đến máy khách và đóng cả thiết bị đầu cuối của máy khách