

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN



**BÁO CÁO MÔN HỌC  
MẠNG MÁY TÍNH**

**PROJECT SOCKET PROGRAMMING  
VIDEO STREAMING WITH RTSP AND RTP**

**Giảng viên hướng dẫn:**

Ths. Chung Thùy Linh

Ths. Huỳnh Thụy Bảo Trân

**Nhóm 2.1:**

Nguyễn Minh Trọng	24120233
Thái Hoàng	24120316
Trần Vĩnh Kiệt	24120360

**Tp. Hồ Chí Minh, ngày 5 tháng 12 năm 2025**

## BẢNG ĐÁNH GIÁ CÁC THÀNH VIÊN NHÓM 2.1

STT	MSSV	HỌ VÀ TÊN	NỘI DUNG	ĐÓNG GÓP	XÁC NHẬN
1	24120233	Nguyễn Minh Trọng	Hoàn thiện phần logic và nội dung của phần Client	100%	
2	24120316	Thái Hoàng	Hoàn thiện phần HD Video Streaming và Client-Side Caching	100%	
3	24120360	Trần Vĩnh Kiệt	Hoàn thiện phần logic và nội dung của phần Server	100%	

## MỤC LỤC

<b>I. GIỚI THIỆU .....</b>	<b>3</b>
1. Yêu cầu báo cáo.....	3
2. Tóm tắt nội dung đồ án.....	3
3. Tóm tắt hệ thống.....	3
4. Mục tiêu thiết kế hệ thống.....	3
<b>II. TRÌNH TỰ HOẠT ĐỘNG .....</b>	<b>4</b>
1. Flow hoạt động.....	4
1.1. Khởi tạo kết nối ( <i>Handshake</i> ).....	4
1.2. Trạng thái INIT -> READY ( <i>Lệnh SETUP</i> ).....	4
1.3. Trạng thái READY -> PLAYING ( <i>Lệnh PLAY</i> ).....	5
1.4. Trạng thái PLAYING -> READY ( <i>Lệnh PAUSE</i> ).....	5
1.5. Kết thúc phiên ( <i>Lệnh TEARDOWN</i> ).....	6
2. Sơ đồ hoạt động của hệ thống .....	6
<b>III. VẤN ĐỀ GẶP PHẢI VÀ CÁCH GIẢI QUYẾT .....</b>	<b>7</b>
1. Mục đích.....	7
2. Các vấn đề và giải pháp.....	7
a. Vấn đề vỡ hình và mất gói tin với Video HD (1080p) .....	7
b. Vấn đề tràn bộ đệm mạng ( <i>Network Buffer Overflow</i> ) .....	7
c. Vấn đề Video bị giật ( <i>Jitter</i> ) và Bất đồng bộ .....	8
d. Phát triển thêm cơ chế Flow Control thông minh ( <i>Hysteresis</i> ).....	8
<b>IV. TỔNG KẾT YÊU CẦU .....</b>	<b>8</b>
Bảng tổng kết yêu cầu: .....	8
<b>V. PHỤ LỤC.....</b>	<b>9</b>
Tài liệu tham khảo .....	9

# I. GIỚI THIỆU

## 1. Yêu cầu báo cáo

- Overview of your system design (Tổng quan thiết kế hệ thống).
- Diagrams (architecture/state machine) (Sơ đồ kiến trúc/trạng thái).
- Screenshots of a successful session (Ảnh chụp màn hình phiên chạy thành công).
- Problems encountered and how you solved them (Các vấn đề gặp phải và cách giải quyết).
- Summary of how each requirement was fulfilled (Tổng kết mức độ hoàn thành yêu cầu).

## 2. Tóm tắt nội dung đề án

- Xây dựng một ứng dụng Video Streaming Client-Server hoàn chỉnh.
- Hiện thực giao thức RTSP (Real-Time Streaming Protocol) ở phía Client để điều khiển trạng thái phát video.
- Hiện thực đóng gói gói tin RTP (Real-time Transport Protocol) ở phía Server để truyền tải dữ liệu video.
- Đảm bảo khả năng truyền tải video chất lượng cao (HD) và cơ chế bộ đệm (Caching) để video chạy mượt mà.

## 3. Tóm tắt hệ thống

- Ngôn ngữ lập trình: Python 3.
- Thư viện giao diện (GUI): Tkinter (Python standard library).
- Thư viện xử lý ảnh: Pillow (PIL).
- Giao thức mạng:
  - TCP (Control Channel): Dùng cho RTSP.
  - UDP (Data Channel): Dùng cho RTP.
- Các module chính:
  - ServerWorker.py: Xử lý logic phía Server, phân mảnh gói tin và gửi RTP.
  - Client.py: Xử lý logic phía Client, gửi lệnh RTSP, nhận RTP, Caching và hiển thị.
  - RtpPacket.py: Đóng gói và giải mã Header của gói tin RTP.

## 4. Mục tiêu thiết kế hệ thống

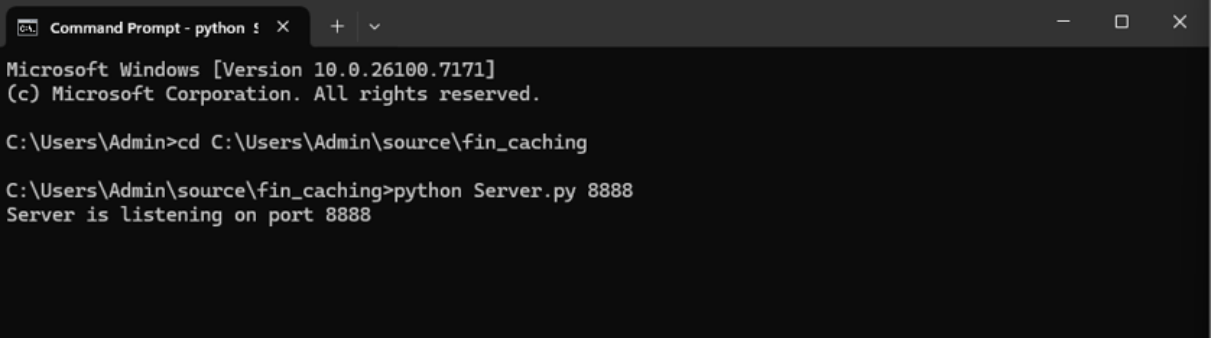
- Tối ưu hóa quá trình trao đổi dữ liệu giữa Client và Server để giảm độ trễ (latency).
- Đảm bảo video không bị vỡ hình khi truyền tải dung lượng lớn (Fragmentation).
- Giữ nguyên các chức năng RTSP cơ bản (SETUP, PLAY, PAUSE, TEARDOWN) và bổ sung tính năng tua (Seeking), hiển thị thời gian (OSD).

## II. TRÌNH TỰ HOẠT ĐỘNG

### 1. Flow hoạt động

#### 1.1. Khởi tạo kết nối (Handshake)

- Người dùng chạy Server.py với tham số cổng (ví dụ: 1025).



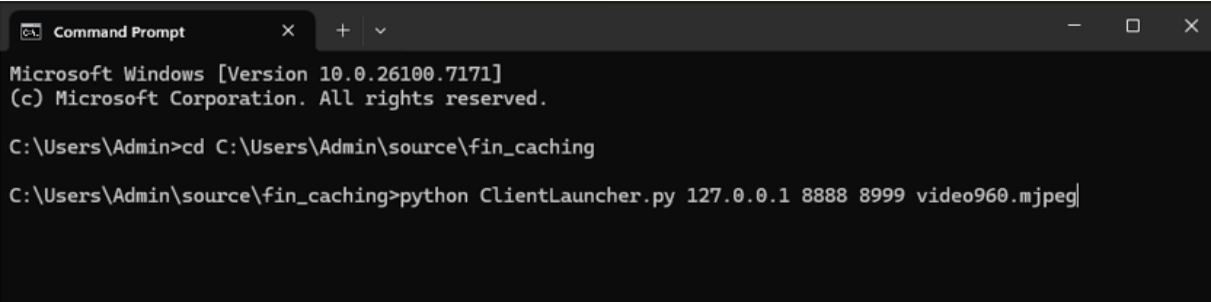
```

Microsoft Windows [Version 10.0.26100.7171]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin>cd C:\Users\Admin\source\fin_caching

C:\Users\Admin\source\fin_caching>python Server.py 8888
Server is listening on port 8888
  
```

- Người dùng chạy ClientLauncher.py, nhập địa chỉ IP Server, cổng RTSP, cổng RTP và tên file video.
- Client kết nối TCP tới Server.



```

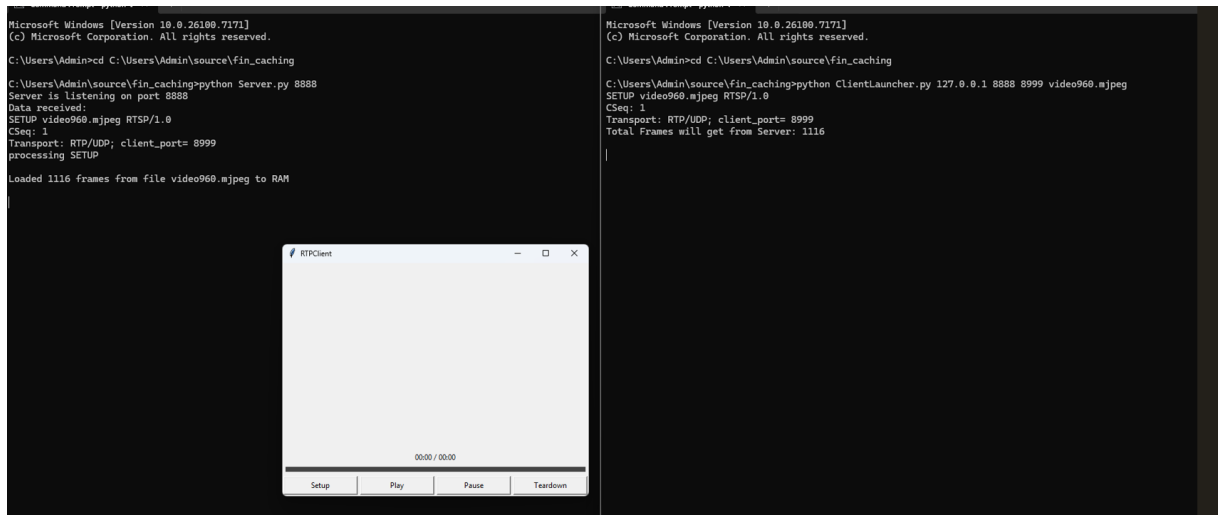
Microsoft Windows [Version 10.0.26100.7171]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin>cd C:\Users\Admin\source\fin_caching

C:\Users\Admin\source\fin_caching>python ClientLauncher.py 127.0.0.1 8888 8999 video960.mjpeg
  
```

#### 1.2. Trạng thái INIT -> READY (Lệnh SETUP)

- Client gửi lệnh SETUP <filename> RTSP/1.0 kèm theo cổng nhận dữ liệu RTP (client\_port).
- Server khởi tạo luồng đọc video (VideoStream), tạo Session ID ngẫu nhiên.
- Server phản hồi 200 OK kèm Session ID và tổng số khung hình (TotalFrames).

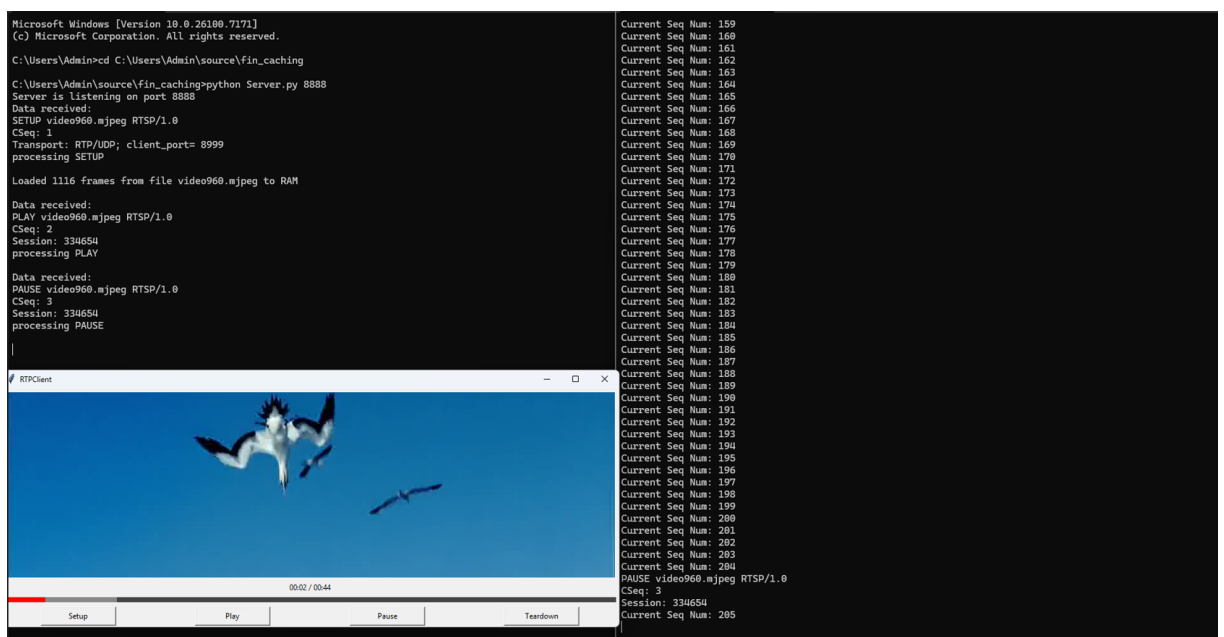


### 1.3. Trạng thái READY -> PLAYING (Lệnh PLAY)

- Client gửi lệnh PLAY.
- Server bắt đầu luồng gửi dữ liệu (sendRtp). Server đọc file MJPEG, cắt nhỏ các frame lớn (Fragmentation) thành các gói tin RTP (Payload < 1400 bytes) và gửi qua UDP tới Client.
- Client nhận dữ liệu vào bộ đệm (Queue), chờ đủ ngưỡng (Buffering) rồi bắt đầu hiển thị lên giao diện.

### 1.4. Trạng thái PLAYING -> READY (Lệnh PAUSE)

- Client gửi lệnh PAUSE.
- Server nhận lệnh, đặt cờ event để tạm dừng vòng lặp gửi RTP. Video trên Client dừng lại nhưng kết nối vẫn giữ.



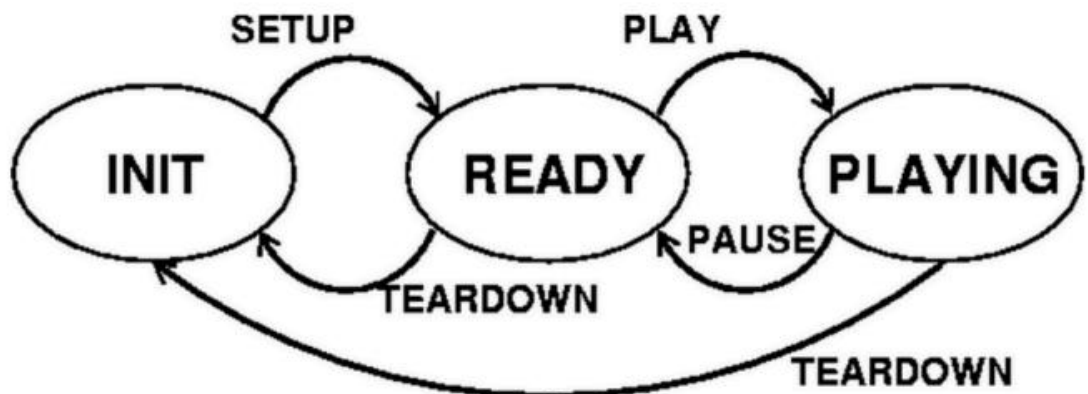
## 1.5. Kết thúc phiên (Lệnh TEARDOWN)

- Client gửi TEARDOWN.
- Server đóng kết nối socket, giải phóng tài nguyên và reset trạng thái về INIT.

```
(c) Microsoft Corporation. All rights reserved.
C:\Users\Admin>cd C:\Users\Admin\source\fin_caching
C:\Users\Admin\source\fin_caching>python Server.py 8888
Server is listening on port 8888
Data received:
SETUP video960.mjpeg RTSP/1.0
CSeq: 1
Transport: RTP/UDP; client_port= 8999
processing SETUP
Loaded 1116 frames from file video960.mjpeg to RAM
Data received:
PLAY video960.mjpeg RTSP/1.0
CSeq: 2
Session: 334654
processing PLAY
Data received:
PAUSE video960.mjpeg RTSP/1.0
CSeq: 3
Session: 334654
processing PAUSE
Data received:
PLAY video960.mjpeg RTSP/1.0
CSeq: 4
Session: 334654
processing PLAY
Data received:
PAUSE video960.mjpeg RTSP/1.0
CSeq: 5
Session: 334654
processing PAUSE
Data received:
PLAY video960.mjpeg RTSP/1.0
CSeq: 6
Session: 334654
processing PLAY
Data received:
TEARDOWN video960.mjpeg RTSP/1.0
CSeq: 7
Session: 334654
processing TEARDOWN

Current Seq Num: 1039
Current Seq Num: 1040
Current Seq Num: 1041
Current Seq Num: 1042
Current Seq Num: 1043
Current Seq Num: 1044
Current Seq Num: 1045
Current Seq Num: 1046
Current Seq Num: 1047
Current Seq Num: 1048
Current Seq Num: 1049
Current Seq Num: 1050
Current Seq Num: 1051
Current Seq Num: 1052
Current Seq Num: 1053
Current Seq Num: 1054
Current Seq Num: 1055
Current Seq Num: 1056
Current Seq Num: 1057
Current Seq Num: 1058
Current Seq Num: 1059
Current Seq Num: 1060
Current Seq Num: 1061
Current Seq Num: 1062
Current Seq Num: 1063
Current Seq Num: 1064
Current Seq Num: 1065
Current Seq Num: 1066
Current Seq Num: 1067
Current Seq Num: 1068
Current Seq Num: 1069
Current Seq Num: 1070
Current Seq Num: 1071
Current Seq Num: 1072
Current Seq Num: 1073
Current Seq Num: 1074
Current Seq Num: 1075
Current Seq Num: 1076
Current Seq Num: 1077
Current Seq Num: 1078
Current Seq Num: 1079
Current Seq Num: 1080
Current Seq Num: 1081
Current Seq Num: 1082
Current Seq Num: 1083
Current Seq Num: 1084
TEARDOWN video960.mjpeg RTSP/1.0
CSeq: 7
Session: 334654
C:\Users\Admin\source\fin_caching>
```

## 2. Sơ đồ hoạt động của hệ thống



Hệ thống hoạt động dựa trên máy trạng thái (Finite State Machine) của giao thức RTSP phía Client như mô tả trong. Chu trình bao gồm 3 trạng thái chính:

- **INIT (Khởi tạo):** Trạng thái mặc định khi Client vừa khởi động, chưa thiết lập phiên làm việc nào với Server.
- **READY (Sẵn sàng):** Trạng thái sau khi Client gửi lệnh SETUP thành công. Lúc này, phiên làm việc (Session ID) đã được khởi tạo và Server đã sẵn sàng để gửi dữ liệu.
- **PLAYING (Đang phát):** Trạng thái khi Client gửi lệnh PLAY. Server liên tục gửi các gói tin RTP video và Client thực hiện giải mã, hiển thị.

Quá trình chuyển đổi giữa các trạng thái được thực hiện thông qua các lệnh điều khiển: **SETUP** (INIT → READY), **PLAY** (READY → PLAYING), **PAUSE** (PLAYING → READY), và **TEARDOWN** (kết thúc phiên, quay về INIT).

---

### III. VẤN ĐỀ GẶP PHẢI VÀ CÁCH GIẢI QUYẾT

#### 1. Mục đích

- Ghi nhận các khó khăn kỹ thuật trong quá trình làm và triển khai hệ thống Streaming, nêu ra giải pháp cụ thể mà nhóm đã dùng để khắc phục các giới hạn của giao thức UDP và xử lý video HD.

#### 2. Các vấn đề và giải pháp

##### a. Vấn đề vỡ hình và mất gói tin với Video HD (1080p)

- **Nguyên nhân:** Các frame video HD (MJPEG) có kích thước rất lớn (có thể lên tới 200KB), vượt quá kích thước truyền tải tối đa (MTU) của gói tin UDP (~1500 bytes). Nếu gửi nguyên frame, router sẽ phân mảnh hoặc loại bỏ gói tin, dẫn đến Client nhận thiếu dữ liệu và không thể giải mã ảnh.
- **Giải pháp:**
  - Triển khai thuật toán Phân mảnh tầng ứng dụng (Application-level Fragmentation) tại Server.
  - Chia frame lớn thành các gói nhỏ (MAX\_PAYLOAD\_SIZE = 1400 bytes).
  - Sử dụng bit Marker trong RTP Header để đánh dấu gói tin cuối cùng của một frame. Client sẽ gom các gói lại cho đến khi gặp Marker mới hiển thị.

##### b. Vấn đề tràn bộ đệm mạng (Network Buffer Overflow)

- **Nguyên nhân:** Khi Server gửi một loạt các mảnh của frame 1080p (khoảng 100-200 gói tin) cùng một lúc với tốc độ CPU tối đa, bộ đệm nhận (Receive Buffer) của socket phía Client bị tràn ngay lập tức, dẫn đến mất gói tin hàng loạt.
- **Giải pháp:**
  - Server: Thêm cơ chế Flow Control đơn giản bằng cách `time.sleep(0.002)` sau mỗi lần gửi gói tin RTP để "nhường" băng thông.
  - Client: Tăng kích thước bộ đệm socket hệ thống lên 20MB bằng lệnh `setsockopt(socket.SOL_SOCKET, socket.SO_RCVBUF, 20*1024*1024)`.



### c. Vấn đề Video bị giật (Jitter) và Bất đồng bộ

- **Nguyên nhân:** Do UDP không đảm bảo thứ tự và tốc độ đến không đều, video hiển thị trực tiếp sẽ bị giật cục.
- **Giải pháp:**
  - Đánh số cho các gói được phân ra và gửi qua Client, Client nhận và sắp xếp lại các gói theo timestamp (FrameNum).
  - Kiểm tra xem frame có bị mất gói trước khi thêm vào buffer.
  - Hiện thực Client-Side Caching (Bộ đệm phía Client).
  - Sử dụng một hàng đợi (Queue) để lưu trữ tạm thời các frame đã nhận.
  - Áp dụng ngưỡng Pre-buffering (chỉ phát khi đã tải được 80 frames) để đảm bảo trải nghiệm mượt mà.

### d. Phát triển thêm cơ chế Flow Control thông minh (Hysteresis)

Thay vì chỉ Pause khi buffer đầy, nhóm áp dụng cơ chế Hysteresis để điều khiển luồng:

- **Ngưỡng trên (High Watermark):** Khi Buffer đạt 130/150 frame, Client chủ động gửi PAUSE để tránh tràn bộ nhớ.
- **Ngưỡng dưới (Low Watermark):** Khi Buffer tiêu thụ xuống còn 50 frame, Client tự động gửi PLAY để nạp tiếp dữ liệu. Cơ chế này giúp server không bị ngắt/bật liên tục (flapping), giúp video mượt mà hơn so với việc đợi buffer cạn sạch mới tải lại.

---

## IV. TỔNG KẾT YÊU CẦU

### Bảng tổng kết yêu cầu:

Yêu cầu	Đáp ứng
Implement RTSP Protocol	Client và Server giao tiếp hoàn chỉnh qua 4 lệnh: SETUP, PLAY, PAUSE, TEARDOWN. Quản lý phiên bằng Session ID và CSeq đúng chuẩn RFC 2326.
RTP Packetization	Đóng gói đúng cấu trúc RTP Header (12 bytes) bao gồm Version, Sequence Number, Timestamp, SSRC và Payload Type (MJPEG).
HD Video Streaming (Advanced)	Hỗ trợ video 720p/1080p thông qua thuật toán phân mảnh gói tin (Fragmentation) và cơ chế gom gói tin (Reassembly) tại Client.

<b>Yêu cầu</b>	<b>Đáp ứng</b>
Client-Side Caching (Advanced)	Đã cài đặt bộ đệm (Frame Buffer) giúp video chạy mượt mà, khử jitter. Tự động xử lý tình huống mạng chậm hoặc máy xử lý không kịp (Frame Skipping).
Giao diện & Tính năng thêm	Giao diện Tkinter trực quan, hỗ trợ vẽ thời gian thực (OSD) lên video và tua (Seeking) đến vị trí bất kỳ.

---

## **V. PHỤ LỤC**

### **Tài liệu tham khảo**

- RFC 2326 - Real Time Streaming Protocol (RTSP).
- RFC 1889 - RTP: A Transport Protocol for Real-Time Applications.
- RFC 2435 - RTP Payload Format for JPEG-compressed Video.
- Tài liệu môn học Mạng máy tính - Khoa CNTT, ĐH KHTN.