

[Lập trình mạng](#) >

# Tìm hiểu kỹ thuật video streaming

## 1. Giới thiệu

Streaming video là một kỹ thuật được sử dụng khá phổ biến trong các ứng dụng mạng. Rất nhiều các địa chỉ ứng dụng rộng rãi trong thực tế như: các phần mềm (media player, web browser, ...) trên các máy khách truy cập và xem video từ các máy chủ theo mô hình server/client; các ứng dụng hội họp trực tuyến, đào tạo từ xa; giám sát, điều khiển từ xa qua hình ảnh thời gian thực, v.v... Trong đó có nhiều ứng dụng sử dụng hệ thống nhúng triển khai kỹ thuật này.

Bài viết này trình bày tóm tắt cơ bản về kỹ thuật streaming video, các giao thức cần thiết và giới thiệu một thư viện lập trình để triển khai kỹ thuật này. Từ đó phục vụ cho phần xây dựng một ứng dụng minh họa streaming video trên hệ nhúng KIT Linux FriendlyArm micro 2440.

Streaming video sử dụng cách thức phát lại các đoạn video được lưu trữ trên các máy tính trên mạng tới người dùng đầu cuối muốn xem đoạn video mà không cần tải đoạn video đó về trên máy tính. Về bản chất, 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. Với đặc tính như vậy thì streaming video là kỹ thuật cũng khá phức tạp để triển khai. Nhưng với những lợi ích mà kỹ thuật streaming video đem lại, chúng ta hoàn toàn có thể triển khai được kỹ thuật này trên thực tế.

## 2. Kỹ thuật streaming video

### Các bước thực hiện kỹ thuật streaming video:

- Phần mềm máy khách (media player, web browser, ...) cần kết nối được và xác định file video trên máy streaming server muốn xem.
- Yêu cầu streaming file video đó sẽ được gửi tới streaming server để tìm file video đó.
- Chương trình thực hiện streaming chạy trên máy streaming server sẽ chia file video thành các frame rồi gửi các frame đó tới máy yêu cầu sử dụng các giao thức ràng buộc về thời gian (RTSP, RTP, RTCP).
- Khi các frame về máy khách, sẽ được lưu trữ trong vùng đệm và nội dung các frame sẽ được giải mã (decode) và hiển thị thông qua các chương trình chơi video (ví dụ VLC)



Hình 1. Streaming video

Một số khái niệm được sử dụng trong streaming video:

- **Streaming video** (luồng video) thực chất là quá trình truyền các frame của file video tới người nhận.
- **Demand streaming** (stream theo yêu cầu) là quá trình streaming một file video có sẵn (đã được lưu trên ổ cứng) tới người nhận.
- **Live streaming** (stream từ một nguồn tạo video) là quá trình streaming trực tiếp từ các frame video được tạo ra từ các thiết bị thu nhận video (như camera) tới người nhận.
- **H.264, VP8** là các thuật toán mã hóa cho các luồng video.
- **Bitstream** là khái niệm ám chỉ một luồng video từ máy chủ streaming tới máy khách nhận các frame video dựa vào giao thức MMS hay RTP.
- **Codec**: thuật ngữ ám chỉ chung cho các thuật toán mã hóa đường truyền trong quá trình streaming audio hay video.
- **RTSP (Real Time Streaming Protocol)** là giao thức mạng điều khiển quá trình streaming video hay streaming audio.
- **RTP (Real-time Transport Protocol)** là giao thức chuẩn định dạng cho gói tin (packet) video hay audio được truyền trên mạng.

### 3. Giao thức RTSP

RTSP (Real Time Streaming Protocol) là một giao thức điều khiển trên mạng được thiết kế để sử dụng giao tiếp giữa máy client và máy streaming server. Giao thức này được sử dụng để thiết

lập và điều khiển phiên giao dịch giữa các máy tính (end points).

Về hình thức giao thức RTSP cũng có nét tương đồng với giao thức HTTP, RTSP định nghĩa một bộ các tín hiệu điều khiển tuần tự, phục vụ cho việc điều khiển quá trình playback. Trong khi giao thức HTTP là giao thức không có trạng thái thì RTSP là giao thức có xác định trạng thái. Một định danh được sử dụng khi cần thiết để theo dõi các phiên giao dịch hiện tại của quá trình streaming video gọi là số hiệu session. Cũng giống như HTTP, RTSP sử dụng TCP là giao thức để duy trì một kết nối đầu cuối tới đầu cuối và các thông điệp điều khiển của RTSP được gửi bởi máy client tới máy server. Nó cũng thực hiện điều khiển lại các đáp trả từ máy server tới máy client. Cổng mặc định được sử dụng bởi giao thức này là 554.

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 sẽ 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.

```
C->S:  OPTIONS rtsp://example.com/media.mp4 RTSP/1.0
      CSeq: 1
      Require: implicit-play
      Proxy-Require: gzipped-messages

S->C:  RTSP/1.0 200 OK
      CSeq: 1
      Public: DESCRIBE, SETUP, TEARDOWN, PLAY, PAUSE
```

## Hình 2: 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.

```

C->S: DESCRIBE rtsp://example.com/media.mp4 RTSP/1.0
      CSeq: 2

S->C: RTSP/1.0 200 OK
      CSeq: 2
      Content-Base: rtsp://example.com/media.mp4
      Content-Type: application/sdp
      Content-Length: 460

      m=video 0 RTP/AVP 96
      a=control:streamid=0
      a=range:npt=0-7.741000
      a=length:npt=7.741000
      a=rtpmap:96 MP4V-ES/5544
      a=mimetype:string;"video/MP4V-ES"
      a=AvgBitRate:integer;304018
      a=StreamName:string;"hinted video track"
      m=audio 0 RTP/AVP 97
      a=control:streamid=1
      a=range:npt=0-7.712000
      a=length:npt=7.712000
      a=rtpmap:97 mpeg4-generic/32000/2
      a=mimetype:string;"audio/mpeg4-generic"
      a=AvgBitRate:integer;65790
      a=StreamName:string;"hinted audio track"

```

**Hình 3: 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 các 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.

```

C->S: SETUP rtsp://example.com/media.mp4/streamid=0 RTSP/1.0
      CSeq: 3
      Transport: RTP/AVP;unicast;client_port=8000-8001

S->C: RTSP/1.0 200 OK
      CSeq: 3
      Transport: RTP/AVP;unicast;client_port=8000-8001;server_port=9000-9001
      Session: 12345678

```

**Hình 4: 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 một 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.

```
C->S: PLAY rtsp://example.com/media.mp4 RTSP/1.0
      CSeq: 4
      Range: npt=5-20
      Session: 12345678

S->C: RTSP/1.0 200 OK
      CSeq: 4
      Session: 12345678
      RTP-Info: url=rtsp://example.com/media.mp4/streamid=0;seq=9810092;rtptime=3450012
```

### Hình 5: 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.

```
C->S: PAUSE rtsp://example.com/media.mp4 RTSP/1.0
      CSeq: 5
      Session: 12345678

S->C: RTSP/1.0 200 OK
      CSeq: 5
      Session: 12345678
```

### Hình 6: 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.

```
C->S: TEARDOWN rtsp://example.com/media.mp4 RTSP/1.0
      CSeq: 8
      Session: 12345678

S->C: RTSP/1.0 200 OK
      CSeq: 8
```

### Hình 7: TEARDOWN Request

## 4. Giao thức RTP

RTP (Real-time Transport Protocol) định dạng một gói tin RTP được dùng để truyền trên luồng dữ liệu video hay audio dựa trên địa chỉ IP. RTP được sử dụng trong phiên giao dịch giữa các hệ thống giải trí hoặc giao tiếp mà có triển khai kỹ thuật streaming video như là telephony, ứng dụng hội họp từ xa, hệ thống giám sát bằng hình ảnh dựa trên IP.

RTP được sử dụng kết hợp với giao thức RTCP ( RTP Control Protocol ). Trong đó, RTP được sử dụng để đóng gói các frame dữ liệu ( audio và video) để truyền trên luồng dữ liệu thì RTCP được sử dụng để giám sát chất lượng của dịch vụ (QoS) hoặc để thống kê theo các tiêu chí trong quá trình truyền tải. Thường thì giao thức RTP sử dụng cổng có số hiệu chẵn còn giao thức RTCP sử dụng cổng có số hiệu lẻ.

RTP được thiết kế cho quá trình streaming theo thời gian thực từ theo kiểu điểm tới điểm. Giao thức này cung cấp tiện ích để dò ra những gói tin RTP đã quá hạn. Trên thực tế, gói tin RTP sử dụng địa chỉ IP trên mạng để định danh các máy tính gửi và nhận. RTP cũng hỗ trợ truyền dữ liệu tới nhiều điểm đích thông qua địa chỉ IP multicast.

RTP được phát triển bởi tổ chức Audio / Video Transport của tổ chức tiêu chuẩn IETF. RTP được sử dụng kết hợp với các giao thức khác như H.323 và giao thức RTSP. Chuẩn RTP định nghĩa một cặp giao thức làm việc với nhau đó là RTP và RTCP. RTP được sử dụng để truyền tải dữ liệu đa phương tiện và giao thức RTCP được sử dụng để gửi các thông tin điều khiển với các tham số QoS.

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

Giao thức truyền, RTP, quy định cách thức truyền dữ liệu theo thời gian thực. Thông tin được cung cấp bởi giao thức này bao gồm thời gian đồng bộ (timestamps), số thứ tự gói tin (phục vụ cho việc tìm gói tin bị lạc ) và chi phí cho việc mã hóa định dạng dữ liệu.

Giao thức điều khiển, RTCP được sử dụng cho việc kiểm tra chất lượng (QoS) luồng dữ liệu và thực hiện đồng bộ giữa các luồng dữ liệu. So với RTP, thì băng thông của RTCP sẽ nhỏ hơn, vào cỡ 5%.

Một giao thức cho phép miêu tả dữ liệu đa phương tiện nhưng không bắt buộc phải kèm theo là giao thức miêu tả phiên ( Session Description Protocol – SDP).

Phiên ( Session ): 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

RTP packet header							
bit offset	0-1	2	3	4-7	8	9-15	16-31
0	Version	P	X	CC	M	PT	Sequence Number
32	Timestamp						
64	SSRC identifier						
96	CSRC identifiers ...						
96+32×CC	Profile-specific extension header ID					Extension header length	
128+32×CC	Extension header ...						

**Hình 8: 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.

## 5. Thư viện mã nguồn mở LIVE555



LIVE555 là một tập các mã nguồn được viết bằng C++ và được gom nhóm thành thư viện. LIVE555 là thư viện được dùng để viết các chương trình streaming video, audio và các chương trình chơi video và audio theo kiểu streaming. Thư viện này hỗ trợ các chuẩn giao thức mở như RTP/RTCP và RTSP phục vụ cho kỹ thuật streaming. Hơn thế nữa, LIVE555 có thể quản lý được các định dạng video như H.264, MPEG, VP8 and DV và các định dạng audio như MPEG, AMR, AC-3 và Vorbis. Mặt khác thư viện này cũng được thiết kế để có khả năng mở rộng hỗ trợ thêm các định dạng video và audio khác nữa một cách dễ dàng. LIVE555 có thể sử dụng để triển khai các ứng dụng RTSP server và RTSP client. Có rất nhiều dịch vụ streaming video hay streaming audio sử dụng LIVE555 để cung cấp dịch vụ của mình. Mặt khác, cũng có rất nhiều chương trình chơi nhạc, phát video sử dụng thư viện này để triển khai RTSP client như là VLC media player hay Mplayer. LIVE555 được phát triển bởi công ty Live Networks.

LIVE555 bao gồm 4 thư mục con:

- ¢ UsageEnvironment: Thư mục này bao gồm các lớp UsageEnvironment và lớp TaskScheduler được sử dụng cho việc lập lịch xử lý các sự kiện, gán các chương trình xử lý sự kiện cho các sự kiện bất đồng bộ và xuất ra các thông điệp cảnh báo hoặc các thông điệp lỗi. Ngoài ra, lớp HashTable định nghĩa một giao diện cho một bảng băm tổng hợp. Các lớp này hầu hết là các lớp cơ sở ảo, các hàm ảo bắt buộc phải được định nghĩa ở các lớp kế thừa.
- ¢ Groupsock: Thư mục này chứa các lớp đóng gói lại các giao diện mạng và sockets. Các lớp này đóng gói một socket được dùng cho việc gửi ( hay nhận) các gói tin multicast.
- ¢ liveMedia: Thư mục này có chứa các lớp tạo thành một cấu trúc phân cấp lớp mà lớp gốc là lớp Medium – lớp này định nghĩa các kiểu streaming media và các thuật toán mã hóa codecs.
- ¢ BasicUsageEnvironment: Thư mục này chứa các lớp con của các lớp cơ sở ảo trong thư mục UsageEnvironment. Các lớp này được sử dụng rất đơn giản cho các ứng dụng console. Các thao tác đọc sự kiện hay delay đều được sử lý bởi một vòng lặp select().

**[Xem bài viết tiếp theo minh họa kỹ thuật streaming video sử dụng lập trình Qt trên hệ nhúng Linux FriendlyArm Tiny6410.](#)**