***TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI***

------o0o------

****

Báo cáo môn Project KTMT

**Đề tài: Xây dựng ứng dụng VoIP qua Internet bằng Java**

**Demo: Local Area Network**

***Giảng viên hướng dẫn:* Thầy Nguyễn Linh Giang**

|  |  |  |
| --- | --- | --- |
|  | **Họ và tên** | **MSSV** |

**Phạm Thành Phong 20173297**

*Hà Nội, tháng 7 năm 2021*

**I. Mô tả bài toán**

Xây dựng và triển khai ứng dụng VoIP hay còn gọi là Voice IP, nghĩa là âm thanh được truyền qua giao thức internet.

**II. Chức năng hệ thống**

Voice over Internet Protocol (VoIP) là công nghệ đáng ngạc nhiên, cho phép bạn thực hiện cuộc gọi qua Internet và giảm được chi phí gọi điện thoại truyền thống, giao thức này cho phép bạn sử dụng kết nối Internet để thực hiện các cuộc gọi. Thay vì phụ thuộc vào đường dây tương tự như hệ thống điện thoại truyền thống, VoIP sử dụng kỹ thuật số và yêu cầu kết nối băng thông tốc độ cao như DSL hoặc cáp. Có rất nhiều nhà cung cấp khác nhau cung cấp VoIP và nhiều dịch vụ khác. Ứng dụng chung nhất của VoIP cho sử dụng cá nhân hoặc gia đình là các dịch vụ điện thoại dựa trên Internet có chuyển mạch điện thoại. Với ứng dụng này, bạn vẫn cần có một số điện thoại, vẫn phải quay số để thực hiện cuộc gọi như sử dụng thông thường. Người mà bạn gọi đến sẽ không thấy có sự khác nhau so với các cuộc gọi từ hệ thống điện thoại truyền thống. Nhiều nhà cung cấp dịch vụ cũng cho ra những tính năng để sử dụng adapter VoIP của bạn ở bất kỳ đâu có kết nối Internet tốc độ cao, cho phép bạn mang nó đi theo khi đi công tác hoặc đi du lịch.

**III. Cơ sở lý thuyết và triển khai**

1. **Các công cụ sử dụng:**
   1. **Ngôn ngữ lập trình: Java**

Java là một trong những ngôn ngữ lập trình hướng đối tượng. Nó được sử dụng trong phát triển phần mềm, trang web, game hay ứng dụng trên các thiết bị di động.

Java được tạo ra với tiêu chí “Viết (code) một lần, thực thi khắp nơi” (Write Once, Run Anywhere  – WORA). Chương trình phần mềm viết bằng Java có thể chạy trên mọi nền tảng (platform) khác nhau thông qua một môi trường thực thi với điều kiện có môi trường thực thi thích hợp hỗ trợ nền tảng đó

* 1. **Thư viện lập trình JMF**

JMF là một API cho phép tích hợp đa phương tiện vào ứng dụng của developer

**1.3 Real Time Protocol**

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.

RTP chạy trên giao thức UDP (User Diagram Protocol), RTP được sử dụng kết hợp với RTP Control Protocol (RTCP). Trong khi RTP mang các luồng truyền thông (ví dụ: âm thanh và video), RTCP được sử dụng để giám sát số liệu truyền tải và chất lượng dịch vụ (QoS) và đồng bộ hóa nhiều luồng. RTP là một trong những cơ sở kỹ thuật của Voice over IP và trong ngữ cảnh này thường được sử dụng kết hợp với một giao thức báo hiệu như Session Initiation Protocol (SIP) thiết lập kết nối qua mạng

**2. Cơ sở lý thuyết**

**2.1. Real Time Protocol (RTP)**

**2.1.1. Khái niệm:**

RTP là giao thức cung cấp dịch vụ phân phối từ đầu đến cuối cho dữ liệu với thời gian thực các đặc điểm, chẳng hạn như âm thanh và video tương tác. Các dịch vụ đó bao gồm xác định loại trọng tải, đánh số thứ tự, đánh dấu thời gian và giám sát giao hàng. Các ứng dụng thường chạy RTP trên UDP để sử dụng các dịch vụ ghép kênh và tổng kiểm tra của nó; cả hai giao thức đóng góp các phần của chức năng giao thức truyền tải.

Tuy nhiên, RTP có thể được sử dụng với mạng cơ sở phù hợp khác hoặc giao thức vận chuyển. RTP hỗ trợ truyền dữ liệu tới nhiều điểm đến bằng cách sử dụng phân phối đa hướng nếu được cung cấp bởi mạng cơ bản.

**2.2.2 .Lợi ích của RTP :**

- Cuộc họp online

- Họp video, audio

- Mixers và Translator

- Layered Encoding

**2.2.3. Một số định nghĩa**

- RTP Payload: Dữ liệu được vận chuyển bởi RTP trong một gói

- RTP Packet: Một gói dữ liệu bao gồm tiêu đề RTP cố định, một danh sách các nguồn đóng góp có thể trống và RTP Payload.

- RTCP: Gói tin điều khiển

- RTP Session: Một liên kết giữa một nhóm người tham gia giao tiếp với RTP. Một người tham gia có thể tham gia vào nhiều phiên RTP cùng một lúc. Trong một phiên đa phương tiện, mỗi phương tiện thường được mang trong một phiên RTP riêng biệt với các gói RTCP của riêng nó trừ khi bản thân mã hóa ghép nhiều phương tiện vào một luồng dữ liệu duy nhất. Người tham gia phân biệt nhiều phiên RTP bằng cách tiếp nhận các phiên khác nhau bằng cách sử dụng các cặp địa chỉ truyền tải đích khác nhau, trong đó một cặp địa chỉ truyền tải bao gồm một địa chỉ mạng cộng với một cặp cổng cho RTP và RTCP.

- Nguồn đồng bộ hóa (SSRC): Nguồn của một luồng gói RTP, được xác định bằng mã định danh SSRC số 32-bit có trong tiêu đề RTP để không phụ thuộc vào địa chỉ mạng. Tất cả các gói từ nguồn đồng bộ hóa tạo thành một phần của cùng một không gian thời gian và số thứ tự, vì vậy bộ thu nhóm các gói theo nguồn đồng bộ để phát lại. Ví dụ về nguồn đồng bộ hóa bao gồm người gửi luồng gói tin bắt nguồn từ nguồn tín hiệu như micrô hoặc máy ảnh hoặc bộ trộn RTP. Nguồn đồng bộ hóa có thể thay đổi định dạng dữ liệu của nó, ví dụ: mã hóa âm thanh, theo thời gian. Số nhận dạng SSRC là một giá trị được chọn ngẫu nhiên có nghĩa là duy nhất trên toàn cầu trong một phiên RTP cụ thể

- Nguồn đóng góp (CSRC): Nguồn của một luồng gói RTP đã đóng góp vào luồng kết hợp do bộ trộn RTP tạo ra. Bộ trộn sẽ chèn một danh sách các mã định danh SSRC của các nguồn đã góp phần tạo ra một gói cụ thể vào tiêu đề RTP của gói đó. Danh sách này được gọi là danh sách CSRC.

**2.2.Java Media Framework**

JMF1.0 là một API cho phép tích hợp phương tiện truyền thông (media) vào ứng dụng của bạn và applet. Nó cho phép Java developer lập trình ứng dụng có phương tiện truyền thông theo thời gian. JMF2.0 mở rộng framework để cung cấp khả năng thu và lưu trữ media data, quản lý quá trình được thực thi trong khi playback và thực thi quá trình xử lý tùy ý ở media data stream

### Ưu điểm:

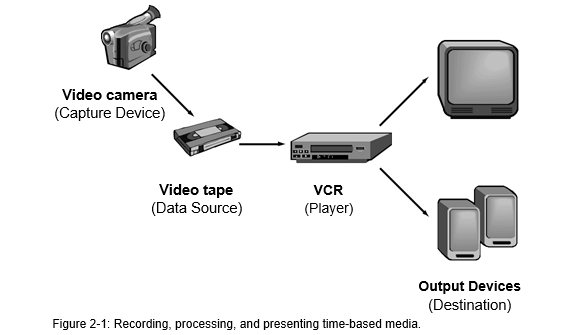
* Dễ dàng lập trình
* Hỗ trợ thu media data
* Cho phép phát triển ứng dụng media stream và conferencing bằng Java
* Cho phép các nhà phát triển và các công nghệ cao để cải tiến các giải pháp tùy ý dựa trên API sẵn có và dễ dàng tích hợp các cộng cụ mới vào framework
* Cho phép truy cập đến media data nguyên bản
* Cho phép quá trình phát triển một cách tùy biến
* Duy trì khả năng tương thích với các phiên bản cũ

### Mô hình xử lý time-base cơ bản

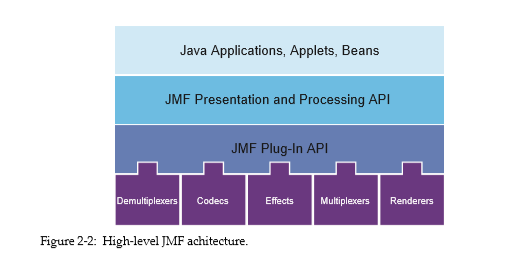
### 

### Kiến trúc JMF cấp cao

Đây là mô hình cơ bản cho quá trình xử lý time-base cơ bản:



JMF sử dụng cùng một mô hình cơ bản này. Datasource đóng gói luồng phương tiện giống như một băng video và player cung cấp các cơ chế xử lý và điều khiển tương tự như VCR. Phát và ghi lại âm thanh và video bằng JMF yêu cầu các thiết bị đầu vào và đầu ra thích hợp như micrô, máy ảnh, loa và màn hình

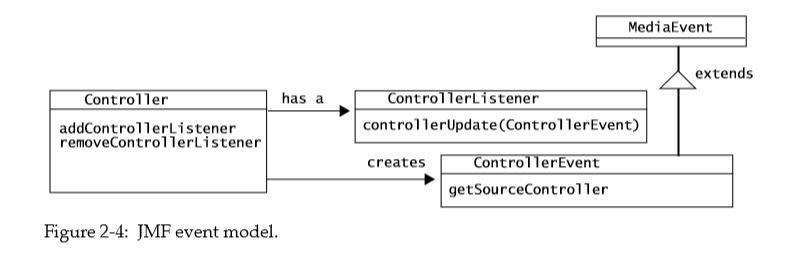


### Các định nghĩa cơ bản

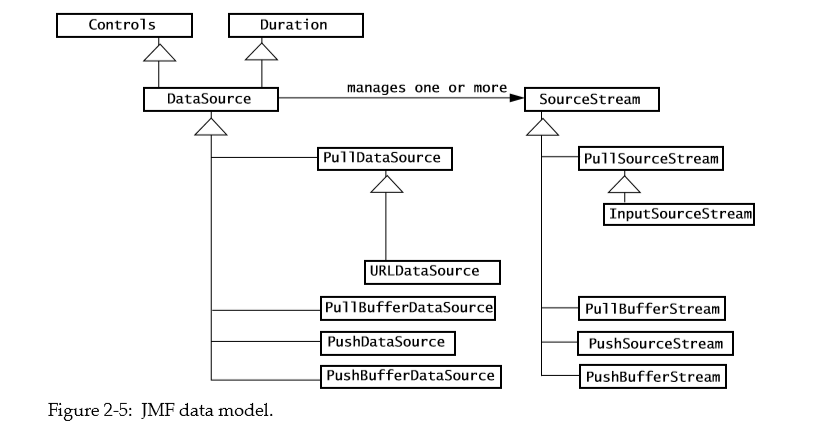
* Time Model: JMF giữ thời gian chính xác đến nano giây. Một điểm cụ thể trong thời gian được đại diện điển hình bởi đối tượng Time

### Manager: quản lý các interface và implementation, gồm 4 Manager: ManagerHandle, PackageManager, CaptureDeviceManager, PluginManager với các chức năng quản lý khác nhau

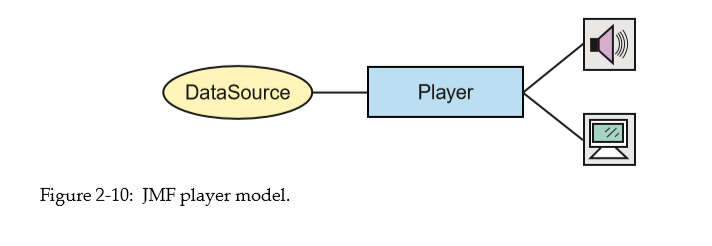
* Event Model: Bất cứ khi nào một đối tượng JMF cần báo cáo về các điều kiện hiện tại, nó sẽ đăng một MediaEvent. MediaEvent được phân lớp để xác định nhiều loại sự kiện cụ thể.



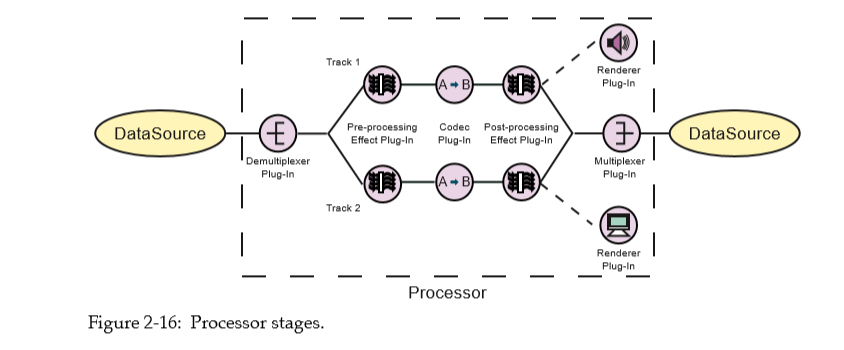
* Data Source: Trình phát đa phương tiện JMF thường sử dụng DataSource để quản lý việc truyền tải nội dung đa phương tiện. Một DataSource bao gồm cả vị trí của phương tiện và giao thức và phần mềm được sử dụng để phân phối phương tiện. Sau khi lấy được, nguồn không thể được sử dụng lại để phân phối các phương tiện khác.



* Các đối tượng DataSink hoặc Multiplexer đọc phương tiện từ DataSource và viết nó ra một đích chẳng hạn như một file để có thể thực hiện giao diện StreamWriterControl. Điều khiển này cho phép người dùng giới hạn kích thước của luồng được tạo.
* Controller: quá trình trình bày được mô hình hóa bởi Controller. Controller xác định trạng thái cơ bản và cơ chế điều khiển cho một đối tượng Controller, trình bày hoặc chụp phương tiện dựa trên thời gian. Nó loại bỏ các giai đoạn mà một bộ điều khiển phương tiện đi qua và cung cấp một cơ chế để kiểm tra sự chuyển đổi giữa các giai đoạn đó. Một số hoạt động phải được thực hiện trước khi dữ liệu phương tiện có thể được trình bày có thể tốn thời gian, vì vậy JMF cho phép điều khiển theo chương trình khi chúng xảy ra
* Player: Player xử lý luồng dữ liệu phương tiện đầu vào và hiển thị nó vào thời điểm chính xác. Một DataSource được sử dụng để cung cấp luồng phương tiện đầu vào đến Player. Điểm đến hiển thị phụ thuộc vào loại phương tiện được gửi trước.



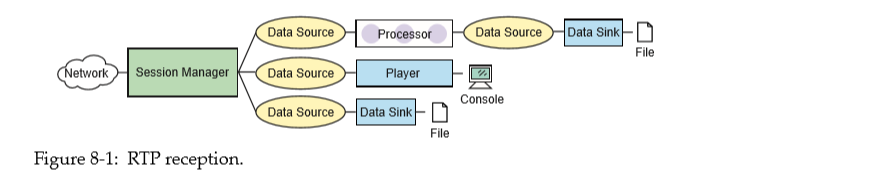
* Processor: Processor cũng có thể được sử dụng để trình bày dữ liệu phương tiện. Bộ xử lý chỉ là một loại Player chuyên dụng cung cấp quyền kiểm soát những gì xử lý được thực hiện trên luồng phương tiện đầu vào. Processor hỗ trợ tất cả các điều khiển bản trình bày giống như một Player.



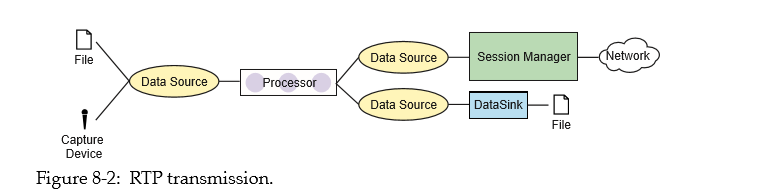
* DataSink: DataSink được sử dụng để đọc dữ liệu media từ DataSource và render media đến một số đích. Một DataSink cụ thể có thể ghi dữ liệu vào một file, ghi dữ liệu qua mạng hoặc hoạt động như một chương trình phát RTP.
* Session: RTP Session là một liên kết giữa một tập hợp các ứng dụng giao tiếp với RTP. Một session được xác định bởi một địa chỉ mạng và một cặp cổng. Một cổng được sử dụng cho dữ liệu phương tiện và cổng kia được sử dụng cho dữ liệu điều khiển (RTCP).

### Jmf-Rtp api

Jmf có hỗ trợ rtp và được đóng gói trong javax.media.rtp



**Mô hình nhận luồng rtp**



**Và có thể nhận luồng rtp theo mô hình**

### 3. Thiết kế hệ thống

### 

### Sơ đồ hệ thống

### Hệ thống xây dựng với mục đích truyền tải, lưu trữ dữ liệu media qua internet, các client target một địa chỉ Ip đích, hệ thống sẽ tạo kết nối keep connect cho phiên truyền tải, dữ liệu truyền qua internet sẽ được nén lại và giải nén ở đầu bên kia

### Sơ đồ hoàn chỉnh

### 4. Thiết kế phần mềm

### 4.1 Thiết kế luồng gửi (Send Stream)

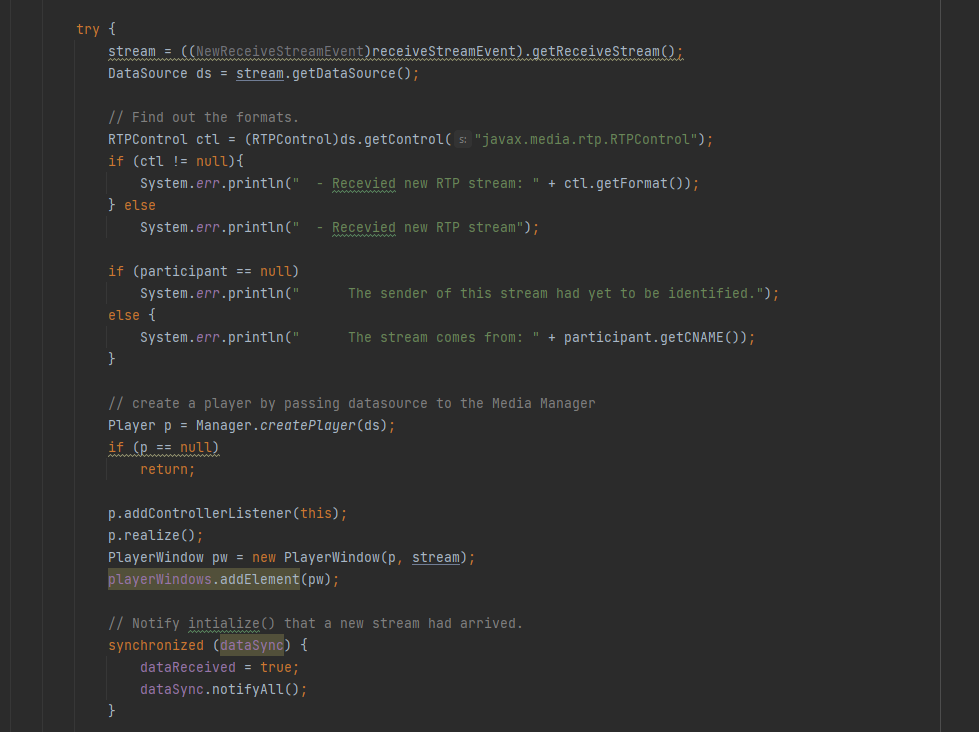
### 

Ban đầu ta sẽ cần lấy thiết bị phần cứng (ở đây là microphone và camera)

Hàm determindDataOutput() sẽ config Processor và dữ liệu thu được từ micro và camera

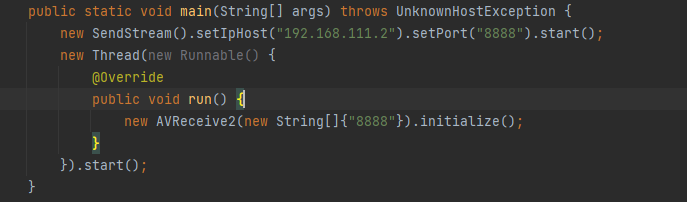
Sau đó ta cần tạo các thành phần theo mô hình luồng gửi RTP

### 4.2 Thiết kế luồng nhận (Receive Stream)



Khi có sự kiện một participant kết nối đến, ta sẽ tạo một receiveStream ứng với luồng đó, đồng thời tạo DataSource và Player để phát dữ liệu thu nhận được

### 4.3 Chương trình main



Với mỗi ứng dụng, ta cần tạo hai luồng chạy song song, một luồng gửi để truyền dữ liệu đến đích cụ thể (hoặc là server chung), một luồng chạy nhận dữ liệu thu được rồi phát ra. Ở đây luồng gửi tạo kết nối đến địa chỉ đích 192.168.111.2 tại cổng 8888, đồng thời tạo cổng nghe ở 8888

### 5. Kết quả

### 

Chương trình chạy thành công và có thể bắt đầu thực hiện giao tiếp qua internet

### 6. Hạn chế

### Do máy tính cá nhân của em không có camera nên em không thể thiết lập chuyển video, chỉ có thể lấy được microphone

### 

Hệ thống chỉ là bước đầu phát triển, nên sau này nếu được mở rộng ra sẽ mong muốn có một server điều phối hoạt động của hệ thống như việc làm trung gian thực hiện tạo kết nối giữa các end point