

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**  
**KHOA CÔNG NGHỆ PHẦN MỀM**

**NGUYỄN QUANG SÁNG**  
**TRẦN THÁI BÌNH**

**KHÓA LUẬN TỐT NGHIỆP**  
**NGHIÊN CỨU VÀ XÂY DỰNG CÁC GIẢI PHÁP**  
**MOBILE VIDEO CONFERENCE VÀ ỨNG DỤNG**

**KỸ SƯ NGÀNH CÔNG NGHỆ THÔNG TIN**

**TP. HỒ CHÍ MINH, 2015**

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**  
**KHOA CÔNG NGHỆ PHẦN MỀM**

**NGUYỄN QUANG SÁNG – 10520200**

**TRẦN THÁI BÌNH – 10520518**

**KHÓA LUẬN TỐT NGHIỆP**  
**NGHIÊN CỨU VÀ XÂY DỰNG CÁC GIẢI PHÁP**  
**MOBILE VIDEO CONFERENCE VÀ ỨNG DỤNG**

**KỸ SƯ NGÀNH CÔNG NGHỆ THÔNG TIN**

**GIẢNG VIÊN HƯỚNG DẪN**  
**PGS.TS. VŨ THANH NGUYỄN**  
**ThS. HUỖNH TUẤN ANH**

**TP. HỒ CHÍ MINH, 201**

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC  
CÔNG NGHỆ THÔNG TIN

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM  
Độc Lập - Tự Do - Hạnh Phúc

TP. HCM, ngày.....tháng.....năm.....

**NHẬN XÉT KHÓA LUẬN TỐT NGHIỆP**  
**(CỦA CÁN BỘ HƯỚNG DẪN)**

**Tên khóa luận:**

**NGHIÊN CỨU VÀ XÂY DỰNG CÁC GIẢI PHÁP MOBILE VIDEO  
CONFERENCE VÀ ỨNG DỤNG**

**Nhóm SV thực hiện:**

NGUYỄN QUANG SÁNG  
TRẦN THÁI BÌNH

10520200  
10520518

**Cán bộ hướng dẫn:**

ThS HUỖNH TUẦN ANH

**Đánh giá Khóa luận**

1. Về cuốn báo cáo:

Số trang	_____	Số chương	_____
Số bảng số liệu	_____	Số hình vẽ	_____
Số tài liệu tham khảo	_____	Sản phẩm	_____

Một số nhận xét về hình thức cuốn báo cáo:

.....  
.....  
.....

2. Về nội dung nghiên cứu:

.....  
.....  
.....

3. Về chương trình ứng dụng:

.....

.....

.....

4. Về thái độ làm việc của sinh viên:

.....

.....

.....

**Đánh giá chung:**

.....

.....

.....

**Điểm từng sinh viên:**

NGUYỄN QUANG SÁNG:...../10

TRẦN THÁI BÌNH:...../10

**Người nhận xét**

ThS Huỳnh Tuấn Anh

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

TRƯỜNG ĐẠI HỌC

Độc Lập - Tự Do - Hạnh Phúc

CÔNG NGHỆ THÔNG TIN

TP. HCM, ngày.....tháng.....năm.....

## NHẬN XÉT KHÓA LUẬN TỐT NGHIỆP

(CỦA CÁN BỘ PHẢN BIỆN)

**Tên khóa luận:**

**NGHIÊN CỨU VÀ XÂY DỰNG CÁC GIẢI PHÁP MOBILE VIDEO  
CONFERENCE VÀ ỨNG DỤNG**

**Nhóm SV thực hiện:**

**Cán bộ phản biện:**

NGUYỄN QUANG SÁNG

10520200

TRẦN THÁI BÌNH

10520518

**Đánh giá Khóa luận**

1. Về cuốn báo cáo:

Số trang	_____	Số chương	_____
Số bảng số liệu	_____	Số hình vẽ	_____
Số tài liệu tham khảo	_____	Sản phẩm	_____

Một số nhận xét về hình thức cuốn báo cáo:

.....

.....

.....

2. Về nội dung nghiên cứu:

.....

.....

.....

3. Về chương trình ứng dụng:

.....

.....

.....

4. Về thái độ làm việc của sinh viên:

.....

.....

.....

**Đánh giá chung:**

.....

.....

.....

**Điểm từng sinh viên:**

NGUYỄN QUANG SÁNG:...../10

TRẦN THÁI BÌNH:...../10

**Người nhận xét**

# LỜI CẢM ƠN

Trong khoảng thời gian nỗ lực thực hiện luận văn “Nghiên cứu và xây dựng các giải pháp mobile video conference và ứng dụng”, chúng em đã nhận được sự giúp đỡ tận tình từ phía nhà trường, thầy cô, gia đình và bạn bè.

Để đáp lại sự giúp đỡ tận tình ấy, chúng con xin được gửi lời cảm ơn tới ba mẹ, người thân đã luôn động viên và tạo mọi điều kiện trong suốt quá trình làm việc và học tập của chúng con, để chúng con có cơ hội hoàn thành luận văn này.

Tiếp theo đó chúng em xin được gửi tới các thầy cô trường Đại học Công Nghệ Thông tin đã tận tình truyền đạt những kiến thức quý báu cho chúng em. Đặc biệt chúng em xin được bày tỏ lòng cảm ơn sâu sắc nhất đến hai thầy Vũ Thanh Nguyên và thầy Huỳnh Tuấn Anh, những người đã tận tình giúp đỡ và theo sát chúng em trong suốt quá trình làm luận văn tốt nghiệp.

Chúng tôi cũng xin cảm ơn bạn bè đã, đang động viên và giúp đỡ chúng em trong suốt quá trình học tập và thực hiện luận văn này.

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

TRƯỜNG ĐẠI HỌC

Độc Lập - Tự Do - Hạnh Phúc

CÔNG NGHỆ THÔNG TIN

## ĐỀ CƯƠNG CHI TIẾT

<b>TÊN ĐỀ TÀI: NGHIÊN CỨU CÁC GIẢI PHÁP VIDEO CONFERENCE TRÊN THIẾT BỊ DI ĐỘNG</b>
<b>Cán bộ hướng dẫn: PGS.TS Vũ Thanh Nguyên, ThS Huỳnh Tuấn Anh</b>
<b>Thời gian thực hiện: Từ ngày 01/09/2014 đến ngày 05/01/2015</b>
<b>Sinh viên thực hiện:</b>  <b>Nguyễn Quang Sáng - 10520200</b>  <b>Trần Thái Bình - 10520518</b>
<b>Nội dung đề tài:</b> <ul style="list-style-type: none"><li>➤ Mục tiêu<ul style="list-style-type: none"><li>○ Nghiên cứu và đề xuất mô hình kiến trúc của dịch vụ Mobile Video Conference dựa trên mạng đồng đẳng Peer to Peer.</li><li>○ Đề xuất mô hình hoạt động của dịch vụ.</li><li>○ Nghiên cứu mô hình mạng Peer to Peer và ứng dụng vào trong chương trình.</li><li>○ Nghiên cứu giao thức thiết lập phiên cuộc gọi Session Initiation Protocol (SIP).</li><li>○ Nghiên cứu cách thức truyền tải dữ liệu thời gian thực được sử dụng trong chương trình.</li></ul></li></ul>



- Tìm hiểu các chuẩn nén audio/video hiện nay, đưa ra chọn lựa phù hợp với mục đích của chương trình.
- Nghiên cứu các điều kiện hoạt động có thể xảy ra từ đó đưa ra các kịch bản kiểm thử khác nhau (về số lượng người dùng, về thiết bị truy cập dịch vụ...)
- Phạm vi
  - Nghiên cứu và xây dựng chương trình thử nghiệm trong mạng LAN cục bộ sử dụng modem Wifi để tạo lập mô hình mạng Peer to Peer.
  - Chương trình ứng dụng được xây dựng trên nền tảng di động Android.
- Đối tượng
  - Thiết kế mô hình kiến trúc của chương trình.
  - Mô hình mạng Peer to Peer và cách thức xây dựng nên một mạng Peer to Peer.
  - Giao thức SIP và cách thức ứng dụng của giao thức SIP vào trong chương trình.
  - Các giao thức truyền dữ liệu video và audio như RTSP, RTP, RTCP và cách thức ứng dụng vào trong chương trình.
- Phương pháp thực hiện
  - Đề xuất mô hình thiết kế của hệ thống.
  - Nghiên cứu các thành phần liên quan đến việc triển khai.
  - Tìm hiểu các codec video sử dụng trong truyền tải dữ liệu.
  - Tìm hiểu nền tảng di động Android, đưa ra thiết kế hệ thống cho phù hợp.
  - Phát triển ứng dụng chạy trên nền tảng di động Android.
  - Tiến hành kiểm thử hệ thống và đưa ra cải tiến phù hợp.
- Kết quả mong đợi
  - Nắm được cách thức nghiên cứu và hoàn thành một đề tài khóa luận.
  - Nắm kiến thức về mạng Peer to Peer.
  - Nắm kiến thức về SIP.

- Nắm kiến thức về streaming media.
- Hoàn thành chương trình thử nghiệm.
- Thực thi được cuộc gọi giữa hai máy Android.

**Kế hoạch thực hiện:**

- Tìm hiểu: thời gian 01/09/2014 đến 01/10/2014.
  - Nguyễn Quang Sáng: tìm hiểu mô hình mạng Peer to Peer, cách thức truyền tải dữ liệu qua mạng, các giao thức truyền tải dữ liệu qua mạng.
  - Trần Thái Bình: tìm hiểu giao thức thiết lập cuộc gọi Session Initiation Protocol và cách thức ứng dụng vào trong chương trình.
- Thực thi, triển khai: thời gian từ 01/10/2014 đến 01/12/2014.
  - Nguyễn Quang Sáng: thực thi và triển khai module Peer to Peer network và module truyền dữ liệu qua mạng.
  - Trần Thái Bình: thực thi module SIP.
  - Ghép nối các thành phần chương trình lại với nhau và tạo nên một chương trình hoàn chỉnh.
- Đánh giá kết quả, viết báo cáo khóa luận: thời gian từ 01/12/2014 đến 05/01/2015.
  - Kiểm tra lại chương trình, cập nhật và sửa lỗi.
  - Viết báo cáo khóa luận.

<p><b>Xác nhận của CBHD</b></p>          <p><b>ThS Huỳnh Tuấn Anh</b></p>	<p><b>TP. HCM, ngày 05 tháng 01 năm 2015</b></p> <p><b>Sinh viên</b></p>
---	--

## MỤC LỤC

CHƯƠNG 1. TỔNG QUAN .....	4
1.1 Tổng quan tình hình nghiên cứu:.....	4
1.2 Tính khoa học và tính mới của đề tài: .....	6
1.3 Mục tiêu, đối tượng và phạm vi nghiên cứu của đề tài: .....	7
1.4 Nội dung, phương pháp dự định nghiên cứu:.....	8
CHƯƠNG 2. NGHIÊN CỨU VÀ THIẾT KẾ HỆ THỐNG DỊCH VỤ MOBILE VIDEO CONFERENCING. ....	9
2.1 Mô hình thiết kế hệ thống và giới thiệu các thành phần: .....	9
2.1.1. Mô hình thiết kế hệ thống:.....	9
2.1.2. Giới thiệu các thành phần có trong hệ thống: .....	10
2.2 Thành phần P2P network.....	11
2.2.1. Mô hình mạng Peer To Peer .....	11
2.2.2. Những thách thức cần giải quyết khi tạo nên một mạng P2P .....	13
2.2.3. Ý tưởng giải quyết vấn đề trong đề tài .....	14
2.2.4. Các thành phần trong mạng P2P.....	18
2.2.5. Phương thức giao tiếp giữa các thành phần trong mạng .....	20
2.2.6. Các pha giao tiếp giữa các thành phần trong mạng .....	25
2.2.7. Ưu điểm và nhược điểm của mô hình.....	32
2.3. Thành phần SIP .....	33
2.3.1. Giới thiệu SIP .....	33
2.3.2. So sánh H323 và SIP và lý do chọn SIP .....	33
2.3.3. Mục tiêu của SIP.....	41
2.3.4. Các thành phần trong SIP .....	41
2.3.5. Các bản tin SIP .....	43
2.3.6. Các giao dịch SIP.....	47
2.3.7. Các hội thoại SIP .....	48
2.3.8. Các giao thức SIP .....	51
2.4. Thành phần Media .....	55

2.4.1	Giao thức RTSP .....	55
2.4.2.	Giao thức RTP .....	59
2.4.3.	Giao thức RTCP.....	64
CHƯƠNG 3. TRIỂN KHAI VÀ THỬ NGHIỆM HỆ THỐNG.....		68
3.1	Cấu hình hệ thống triển khai và kiểm thử .....	68
CHƯƠNG 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....		70
4.1.	Kết luận.....	70
4.2.	Hướng phát triển.....	70

## DANH MỤC HÌNH ẢNH

Hình 2.1: Mô hình thiết kế chương trình .....	9
Hình 2.2: Mô hình mạng Peer-to-Peer (p2p) .....	11
Hình 2.3: Mô hình mạng P2P.....	13
Hình 2.4: Phiên thông điệp.....	22
Hình 2.5: Thông điệp JOIN.....	26
Hình 2.6: Thông điệp UPDATE.....	27
Hình 2.7: Thông điệp LEAVE .....	28
Hình 2.8: Thông điệp NODE ADD .....	29
Hình 2.9: Thông điệp NODE UPDATE .....	30
Hình 2.10: Thông điệp NODE LEAVE .....	31
Hình 2.11: PING thành công.....	32
Hình 2.12: PING không thành công.....	32
Hình 2.13: Bản tin INVITE.....	44
Hình 2.14: Bản tin 200 OK .....	45
Hình 2.15: Phiên giao dịch SIP .....	48
Hình 2.16: Hội thoại trong SIP .....	49
Hình 2.17: Phiên giao dịch hai SIP Proxy .....	50
Hình 2.18: Phiên giao dịch Registrar .....	52
Hình 2.19: Phiên giao dịch SIP Proxy .....	53
Hình 2.20: Phiên kết thúc giao dịch với SIP Proxy .....	54
Hình 2.21: Phiên kết thúc giao dịch SIP .....	54
Hình 2.22: RTSP kết nối client và server.....	56
Hình 2.23: Yêu cầu OPTIONS.....	57
Hình 2.24: Yêu cầu DESCRIBE .....	57
Hình 2.25: Yêu cầu SETUP .....	58
Hình 2.26: Yêu cầu PLAY .....	58
Hình 2.27: Yêu cầu PAUSE.....	59
Hình 2.28: Yêu cầu TEARDOWN.....	59
Hình 2.29: Hoạt động của RTCP .....	65
Hình 2.30: Mẫu gói tin SR của RTCP .....	67

## **DANH MỤC BẢNG**

Bảng 2.1: So sánh H.323 và SIP .....	40
Bảng 2.2: Mã hóa gói tin RTP trong IP .....	62
Bảng 2.3: Kiến trúc gói dữ liệu .....	63
Bảng 3.1: Bảng so sánh thử nghiệm H.264 và H.263 .....	69

## DANH MỤC TỪ VIẾT TẮT

VIẾT TẮT	THUẬT NGỮ TIẾNG VIỆT	THUẬT NGỮ TIẾNG ANH
CODEC	Thành phần mã hóa/giải mã	Coder-Decoder
HTTH	Hội thảo truyền hình	Video conferencing
IETF		Internet Engineering Task Force
IP		Internet Protocol
H.323	<i>Tên riêng</i>	
SIP	Giao thức khởi tạo phiên	Session Initiation Protocol
LAN	Mạng cục bộ	Local Area Network
NAT		Network Address Translation
PSTN		Public Switch Telephon Network
RAS		Registration, Admission, Status
RTCP		Real-Time Transport Control Protocol
RTSP		Real-Time Streaming Protocol
RTP		Real-time Transport Protocol
SDP		Session Description Protocol
Terminal	Trạm đầu cuối	
UA		User Agent
URI		Uniform Resource Indentifiers
VoIP		Voice over IP



## **TÓM TẮT KHÓA LUẬN**

Luận văn gồm 4 chương có nội dung khái quát như sau:

- Chương 1: Tổng quan – Nêu khái quát về đề tài, lý do chọn đề tài.
- Chương 2: Nghiên cứu và thiết kế hệ thống dịch vụ Mobile Video Conference – Phần này đi sâu vào các nghiên cứu về dịch vụ Mobile Video Conference, các thành phần có trong hệ thống và cách thức thực thi các thành phần đó trong chương trình.
- Chương 3: Triển khai và thử nghiệm hệ thống – Mô tả cách thức triển khai hệ thống và đánh giá hoạt động của hệ thống.
- Chương 4: Kết luận và hướng phát triển – Đánh giá những mặt đã làm được, những hạn chế và hướng phát triển của đề tài.

## MỞ ĐẦU

Sự phát triển vượt bậc của Internet và ứng dụng của nó đã được chứng minh qua những thay đổi mạnh mẽ trong cuộc sống và có lẽ khả năng ứng dụng của mạng Internet chỉ phụ thuộc vào sự tưởng tượng của con người. Khả năng mang dữ liệu với độ lớn không giới hạn, mang đến đích một cách chính xác và cùng với sự phát triển của hạ tầng vật lý đã đưa đến khả năng mang nhiều dữ liệu hơn, định tuyến nhanh hơn và thời gian trễ giữa điểm gửi và điểm nhận ngày càng rút ngắn đi đến mức hầu như rất ít. Những đặc tính này đã làm cho Internet đã được ứng dụng vào rất nhiều lĩnh vực nhưng đặc biệt là truyền dữ liệu thời gian thực được sử dụng trong các ứng dụng như truyền âm thanh, hình ảnh trực tiếp, thực hiện cuộc gọi thoại – hình ảnh giữa các thiết bị số có kết nối Internet – Voice over Internet Protocol (VoIP).

Thời kỳ bùng nổ của điện thoại di động thông minh đánh dấu một bước tiến lớn trong lĩnh vực thiết bị số. Thiết bị di động thông minh đã đưa những ứng dụng thời đại Internet đến với từng cá nhân, truy cập mạng mọi lúc, mọi nơi. Và ứng dụng VoIP đã có đủ những điều kiện thuận lợi để phát triển: di động thông minh có thể lập trình ứng dụng được, Internet mọi lúc, mọi nơi và là thiết bị luôn luôn được người sử dụng mang theo bên mình dùng để liên lạc. Với sự phổ biến của Internet như hiện nay trên các thiết bị số và chi phí của Internet ngày càng rẻ cùng với xu hướng sử dụng những thiết bị cá nhân ngày càng mạnh mẽ thì VoIP có một môi trường ứng dụng thật sự rộng lớn và có tiềm năng.

Qua đó cho thấy ứng dụng hội thảo truyền hình là vô cùng to lớn và có thể được áp dụng rộng rãi trong nhiều lĩnh vực khác nhau. Việc ứng dụng hội thảo truyền hình trên thiết bị di động là nhu cầu cấp bách và rất cần thiết cho sự phát triển của xã hội. Tại Việt Nam, nhu cầu sử dụng dịch vụ hội thảo truyền hình đã và đang được mở rộng, nhiều nhà cung cấp dịch vụ lớn của thế giới cũng đã có mặt tại Việt Nam. Hội thảo truyền hình trên thiết bị di động là lĩnh vực còn khá mới và chưa phát triển nhiều. Tuy nhiên với những

lợi ích to lớn như tiết kiệm chi phí, tiếp cận nhanh chóng, dễ dàng hơn trong giao tiếp thảo luận, việc phát triển nghiên cứu và triển khai hệ thống quản lý và cung cấp dịch vụ hội thảo truyền hình trên thiết bị di động thí điểm sẽ mang lại nhiều kết quả to lớn trong nhiều lĩnh vực đời sống.

Trong thời gian thực hiện nghiên cứu đề tài, nhóm em đã nhận được rất nhiều sự chỉ dẫn, ý kiến của thầy Vũ Thanh Nguyên và thầy Huỳnh Tuấn Anh để nhóm em có thể hoàn thành đề tài này một cách tốt nhất. Em chân thành cảm ơn hai thầy.

## **CHƯƠNG 1. TỔNG QUAN**

### **1.1 Tổng quan tình hình nghiên cứu:**

#### **1.1.1 Ngoài nước:**

Hội thảo truyền hình là một dịch vụ ngày càng được áp dụng nhiều cho phép nhiều người hội thảo từ xa với sự trung chuyển hình ảnh và âm thanh từ một người đến những người còn lại. Sự phát triển của công nghệ kỹ thuật thúc đẩy sự phát triển của Internet làm cho mạng máy tính tốc độ ngày càng một cao hơn, dữ liệu được chuyển đi ngày càng một nhiều hơn và chính xác hơn. Cùng với đó là những cách thức truyền tải dữ liệu mới ra đời làm cho dữ liệu được xử lý và truyền đi hiệu quả hơn, đã tạo điều kiện thuận lợi và thúc đẩy dịch vụ này ngày một phát triển trên môi trường mạng.

Trong những năm gần đây, thế giới công nghệ đã chứng kiến một cuộc cách mạng về sự phát triển và sử dụng thiết bị số di động. Các thiết bị di động đã thay đổi cách thức con người sử dụng Internet, giờ đây mọi người có thể truy cập Internet tốc độ cao mọi lúc, mọi nơi. Và hơn thế nữa, những ứng dụng trước kia chỉ có thể sử dụng trên máy tính cá nhân thì nay thiết bị di động có thể thực hiện được. Sự phổ biến của thiết bị số di động như điện thoại thông minh, máy tính bảng,... và thời gian sử dụng hằng ngày của các thiết bị này rất nhiều cùng với nhu cầu liên lạc thường xuyên. Những điều kiện thuận lợi này làm cho lĩnh vực Mobile Video Conference là lĩnh vực nhận được nhiều sự quan tâm trong đời sống hằng ngày. Ứng dụng của Mobile Video Conference là vô cùng to lớn và có thể áp dụng rộng rãi trong nhiều lĩnh vực sau:

- Giáo dục: Mobile Video Conference cho phép trường học có thể mở rộng đối tượng giảng dạy: từ các lớp học trong bậc tiểu học và trung học, đến bậc đại học và sau đại học. Với khả năng hỗ trợ người học theo dõi và tham gia các lớp học cũng như các hoạt động tương tác nhóm ở bất kỳ nơi đâu, thông qua bất cứ thiết bị di động nào, một trường học hoặc bất cứ cơ quan giáo

dục nào cũng có thể hỗ trợ đến mức tối đa cho các học viên của mình, điều này đặc biệt có lợi đối với trẻ em học tại nhà hoặc có hạn chế về thể chất cũng như đối với người học vốn đang đi làm. Mobile Video Conferencing còn mở ra một thị trường học viên đáng kể, góp phần tăng thu nhập cho cơ quan giáo dục. Việc này đã được thực hiện tại Một số trường đại học Old Dominion University, University of Amsterdam, Niigata Seiryo University, Jacobs University, đã sử dụng phần mềm LifeSize ClearSea.

- Y tế: Sử dụng Mobile Video Conferencing, bác sĩ chuyên khoa có thể trực tiếp hỗ trợ các trường hợp phức tạp mà các bác sĩ tổng quát hay chuyên khoa khác gặp phải từ bất cứ nơi đâu trên thế giới. Bằng cách này, ta có thể giảm chi phí cho người bệnh và tăng khả năng phục vụ của các cơ sở y tế. Ví dụ như tại: University of Kentucky triển khai mobile video conferencing để cho phép các bác sĩ của UK HealthCare tiếp cận với bệnh nhân dễ dàng qua sử dụng phần mềm Polycom RealPresence Mobile.
- Xây dựng: Mobile Video Conferencing cho phép đội ngũ xây dựng hợp tác, liên lạc nhịp nhàng với nhau cũng như với các nhà cung cấp vật tư mà không cần rời khỏi công trường. Với khả năng tương tác thời gian thực, việc trao đổi thông tin giữa các bên liên quan với nhau – giữa các bên liên quan với khách hàng càng hiệu quả hơn.
- Thương mại: Các doanh nhân là những người thường xuyên phải di chuyển cũng như có những đối tác quan trọng ở khắp nơi trên thế giới. Mobile Video Conferencing là giải pháp cho phép doanh nhân có thể thực hiện các cuộc hội thảo hoặc đàm phán, ký kết với đối tác trên phạm vi toàn cầu tại bất kỳ đâu và bất kỳ lúc nào.
- Sản xuất: Khi nền sản xuất công nghiệp mở rộng ra phạm vi toàn cầu, việc phân tán nguồn nhân lực ở nhiều nơi trên thế giới hoàn toàn có thể xảy ra: Đội ngũ cung cấp vật tư ở một vùng, đội ngũ thiết kế và kỹ sư lại ở vùng

khác, thậm chí đội ngũ trực tiếp sản xuất cũng có thể ở nơi khác. Với khả năng hỗ trợ tương tác của mình, video conferencing cho phép các nhóm liên quan làm việc với nhau mà không gặp sự gián đoạn nào. Điều này là cực kỳ quan trọng đối với một công ty có phạm vi toàn cầu. Giảm thiểu chi phí, thời gian đi lại thông qua giải pháp tương tác trực tuyến cũng là một cách nâng cao hiệu suất làm việc và giảm chi phí phát sinh.

- Quân sự: Sử dụng Mobile Video Conferencing, những người lính trên chiến trường có thể truyền hình ảnh trực tiếp về cho chỉ huy để nhận được những quyết định mang tính sống còn trong cuộc chiến.

### **1.1.2 Trong nước:**

Tại Việt Nam, nhu cầu sử dụng dịch vụ Video Conferencing đã và đang được mở rộng, nhiều nhà cung cấp dịch vụ lớn của thế giới cũng đã có mặt tại Việt Nam. Hội thảo truyền hình trên thiết bị di động còn khá mới và chưa phát triển nhiều. Chưa có nhiều công trình nghiên cứu về lĩnh vực hội thảo truyền hình trên thiết bị di động được thực hiện tại Việt Nam trong những năm qua. Tuy nhiên với những lợi ích to lớn như tiết kiệm chi phí, tiếp cận nhanh chóng, dễ dàng hơn trong giao tiếp thảo luận, việc phát triển nghiên cứu và triển khai thí điểm hệ thống quản lý và cung cấp dịch vụ hội thảo truyền hình trên thiết bị di động sẽ mang lại nhiều kết quả to lớn trong nhiều lĩnh vực đời sống.

Đề tài luận văn của nhóm em có liên quan đến luận văn:

- Huỳnh Tuấn Anh “NGHIÊN CỨU VÀ XÂY DỰNG CÁC GIẢI PHÁP MOBILE VIDEO CONFERENCE VÀ ỨNG DỤNG”

## **1.2 Tính khoa học và tính mới của đề tài:**

Xây dựng hội thảo truyền hình trên thiết bị di động góp phần tạo nên sự dễ dàng trong việc giao tiếp thông tin, linh hoạt về địa điểm và thời gian. Hệ thống Mobile Video Conference cung cấp nhiều ưu điểm:

- Hệ thống được cấu hình đơn giản cho người dùng cuối.
- Khả năng giao tiếp nhanh chóng, không giới hạn thời gian.
- Có khả năng đáp ứng cuộc gọi chất lượng cao trong môi trường mạng tương đối ổn định.
- Dễ triển khai.

Thời điểm hiện tại, vẫn còn nhiều vấn đề cần được đầu tư và nghiên cứu triển khai một hệ thống hội thảo trực tiếp trên thiết bị di động. Do đó, nhóm đề tài đề xuất trong đề tài này tập trung nghiên cứu các vấn đề chính sau:

- Vấn đề xây dựng mạng Peer to Peer có khả năng tự mở rộng cao.
- Nghiên cứu giao thức khởi tạo phiên giao tiếp để thực hiện cuộc gọi.
- Vấn đề truyền tải dữ liệu thời gian thực trong chương trình.
- Nghiên cứu các thuật toán nén/giải nén video, sử dụng thuật toán nén và giải nén video hiệu quả để sử dụng trong chương trình.

### **1.3 Mục tiêu, đối tượng và phạm vi nghiên cứu của đề tài:**

#### **1.3.1 Nghiên cứu và thiết kế hệ thống Peer to Peer Mobile Video Conference:**

- Nghiên cứu và đề xuất mô hình kiến trúc của dịch vụ dựa trên mạng đồng đẳng Peer to Peer.
- Đề xuất mô hình hoạt động của dịch vụ.
- Nghiên cứu mô hình mạng Peer to Peer và ứng dụng vào trong chương trình.
- Nghiên cứu giao thức thiết lập phiên cuộc gọi SIP.
- Nghiên cứu cách thức truyền tải dữ liệu thời gian thực để sử dụng trong chương trình.
- Tìm hiểu các chuẩn nén dữ liệu audio/video hiện nay, đưa ra lựa chọn phù hợp với mục đích của chương trình.
- Nghiên cứu các điều kiện hoạt động có thể xảy ra từ đó đưa ra các kịch bản kiểm thử khác nhau (về số lượng người dùng, về thiết bị truy cập dịch vụ,...)

### **1.3.2 Xây dựng kiến trúc hạ tầng căn bản của dịch vụ:**

- Dựa trên mô hình kiến trúc chung, triển khai cài đặt và thiết lập cấu hình cho các thiết bị phần cứng.
- Triển khai thành phần quản lý băng thông và phân phối gói tin, giảm thiểu độ trễ.
- Triển khai cài đặt các thành phần bảo mật thông tin cho hệ thống.

### **1.3.3 Xây dựng ứng dụng chạy trên nền tảng di động Android để sử dụng dịch vụ:**

- Phát triển ứng dụng chạy trên nền tảng di động phổ biến Android và sẽ mở rộng chương trình sang hệ điều hành iOS với mục đích phục vụ cho số đông người sử dụng.
- Các ứng dụng client có thể giúp người dùng tham gia hội thảo trực tuyến, tham gia cầu truyền hình qua thiết bị di động cũng như chia sẻ các sự kiện trực tiếp thông qua Mobile Video Conference.

## **1.4 Nội dung, phương pháp dự định nghiên cứu:**

### **1.4.1 Nghiên cứu và thiết kế hệ thống dịch vụ Mobile Video Conference:**

- Đề xuất mô hình thiết kế của hệ thống.
- Nghiên cứu các thành phần liên quan đến việc triển khai.
- Tìm hiểu các codec video sử dụng trong truyền tải dữ liệu.
- Tìm hiểu nền tảng di động Android, đưa ra thiết kế hệ thống cho phù hợp.

### **1.4.2 Xây dựng ứng dụng chạy trên nền tảng di động Android để sử dụng dịch vụ và tiến hành kiểm thử:**

- Phát triển ứng dụng chạy trên nền tảng di động Android.
- Tiến hành kiểm thử hệ thống và đưa ra các cải tiến phù hợp.



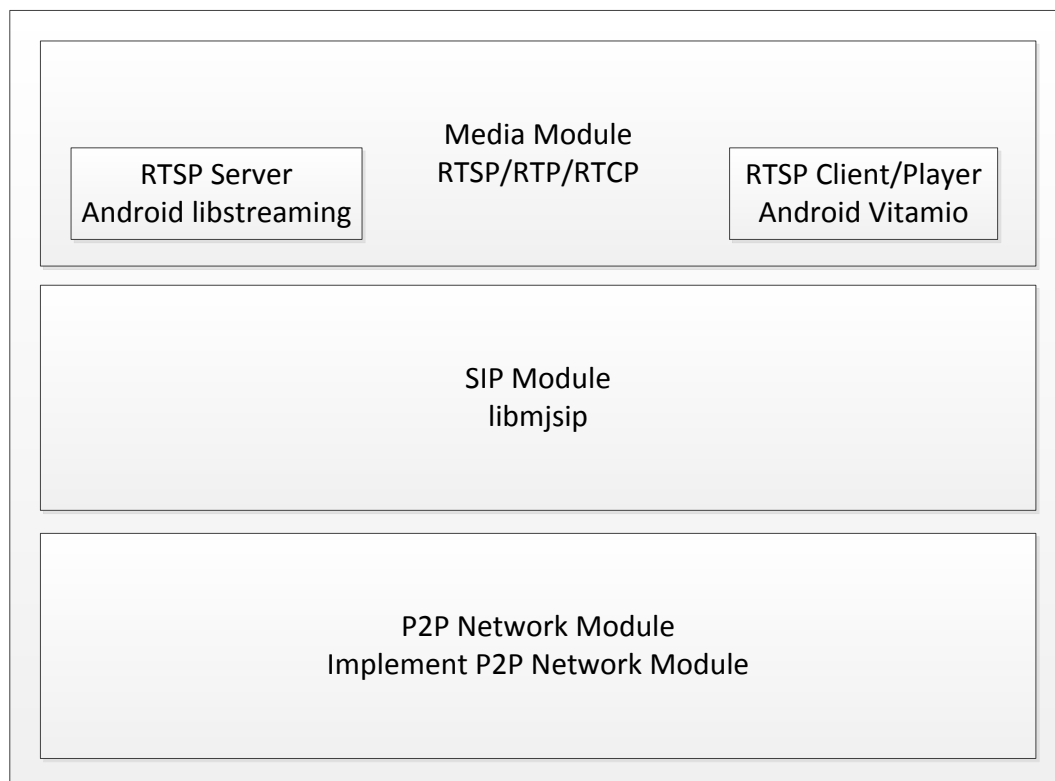
## CHƯƠNG 2.     NGHIÊN CỨU VÀ THIẾT KẾ HỆ THỐNG DỊCH VỤ MOBILE VIDEO CONFERENCING.

### 2.1   Mô hình thiết kế hệ thống và giới thiệu các thành phần:

#### 2.1.1.   Mô hình thiết kế hệ thống:

Hệ thống được thiết kế theo mô hình stack-layer, mỗi layer sẽ chịu trách nhiệm về một tác vụ cốt lõi trong hệ thống, những layer bên dưới sẽ cung cấp API cho những layer bên trên sử dụng. Sự kết hợp giữa các layer sẽ tạo nên chương trình hoàn chỉnh.

Thiết kế của chương trình như sau:



Hình 2.1: Mô hình thiết kế chương trình

Chương trình gồm có 3 module: P2P Network Module, SIP Module, Media Module. Mỗi module trong chương trình đảm trách một nhiệm vụ cốt lõi và là nền

tăng để các module khác xây dựng lên trên đó và sử dụng các thành phần của module bên dưới cung cấp.

### **2.1.2. Giới thiệu các thành phần có trong hệ thống:**

#### **1. Module P2P Network:**

- Module này chịu trách nhiệm cung cấp cơ chế liên kết vào mạng ngang hàng, cung cấp cơ chế giao tiếp giữa các node trong mạng với nhau.
- Module sử dụng mô hình mạng p2p để xây dựng và thực thi.

#### **2. Module SIP:**

- Module này đảm trách vai trò thiết lập và thực hiện cuộc gọi giữa các Peer Node.
- Module SIP sử dụng kết quả của module P2P Network liên kết các node và cung cấp phương thức giao tiếp giữa các node để thực thi giao thức SIP giữa các Peer Node với nhau.

#### **3. Module Media:**

- Module Media đảm trách vai trò truyền dữ liệu hình ảnh và âm thanh sau khi module SIP thực hiện cuộc gọi thành công và cuộc gọi bắt đầu diễn ra.
- Sau khi module SIP thiết lập và thực hiện cuộc gọi thành công. Quá trình chuyển giao dữ liệu giữa các SIP node sẽ được diễn ra. Dữ liệu này bao gồm âm thanh và hình ảnh (cuộc gọi thoại). Module Media lúc này sẽ thực thi cả chức năng server truyền tải dữ liệu cho SIP node còn lại, cũng đồng thời hiển thị hình ảnh và cuộc gọi từ SIP node kia truyền đến.
- Module Media bao gồm một RTSP Server đảm trách tác vụ truyền dữ liệu đến SIP node còn lại trong phiên gọi, và một RTSP client/player sẽ nhận dữ liệu media từ SIP node còn lại và hiển thị node bên này.

## 2.2 Thành phần P2P network

### 2.2.1. Mô hình mạng Peer To Peer

Mô hình mạng Peer-to-Peer còn gọi là mạng đồng đẳng, là một mạng máy tính trong đó hoạt động của mạng chủ yếu dựa vào khả năng tính toán và băng thông của các máy tham gia chứ không tập trung vào một số nhỏ các máy chủ trung tâm như các mạng thông thường. Mạng đồng đẳng thường được sử dụng để kết nối các máy tính một cách trực tiếp với nhau thay vì các máy tính tương tác với một máy chủ cố định làm nhiệm vụ cung cấp dịch vụ như trong mô hình mạng truyền thống client-server. Một mạng đồng đẳng đúng nghĩa không có khái niệm mô hình khách chủ, tất cả các máy tham gia trong mạng đều bình đẳng và được gọi là một **peer node**, đóng vai trò đồng thời là máy chủ cung cấp dịch vụ cho những peer node khác, và là máy khách yêu cầu dịch vụ từ những peer node khác.



Hình 2.2: Mô hình mạng Peer-to-Peer (p2p)

Mạng đồng đẳng có rất nhiều ứng dụng bởi khả năng có thể mở rộng một cách dễ dàng và vô hạn của nó. Càng được mở rộng thì tài nguyên của mạng càng nhiều, và khả năng cung cấp tài nguyên của mạng càng lớn. Cách thức liên lạc trực tiếp của mạng P2P làm cho sự liên lạc trở nên bảo mật và riêng tư bởi vì kết nối chỉ diễn ra giữa hai peer node, đồng thời chi phí cho chuyển giao dữ liệu giữa hai peer

node ít tốn chi phí hơn so với mô hình client-server thông thường. Khái niệm đồng đẳng ngày nay được tiến hóa vào nhiều mục đích sử dụng khác nhau, không chỉ để trao đổi tệp mà còn khái quát hóa thành trao đổi thông tin giữa người với người, đặc biệt trong những tình huống hợp tác giữa một nhóm người trong cộng đồng. Ứng dụng thường gặp nhất là chia sẻ tập tin hoặc để truyền dữ liệu thời gian thực như điện thoại VoIP.

Trong khi hệ thống P2P trước đó đã được sử dụng trong nhiều ứng dụng, kiến trúc của mạng P2P chính thức được phổ biến bởi hệ thống chia sẻ file Napster vào năm 1999. Hệ thống P2P cho phép hàng triệu người sử dụng Internet kết nối trực tiếp, tạo nhóm và hợp tác với nhau để trở thành cỗ máy tìm kiếm người dùng được tạo ra bởi số lượng user tham gia, máy chủ ảo và hệ thống chia sẻ file. Trong mô hình này sử dụng mô hình máy chủ - máy khách cho một số tác vụ và mô hình đồng đẳng cho những tác vụ khác, những mô hình mạng kiểu này thuộc thể hệ mạng P2P thứ nhất. Trong khi đó, các mạng khác như Gnutella hay Freenet (thể hệ thứ 2) sử dụng mô hình đồng đẳng cho tất cả các tác vụ, nên các mạng này thường được xem như là mạng đồng đẳng đúng nghĩa (thực ra Gnutella vẫn sử dụng một số máy chủ để giúp các máy trong mạng tìm kiếm địa chỉ IP của nhau). Do đó mạng đồng đẳng tùy thuộc vào cấu trúc có thể chia ra làm 2 loại:

a. Mô hình mạng đồng đẳng thuần túy

Mô hình mạng này có những đặc điểm đặc trưng như sau:

- Các peer node vừa có vai trò là máy khách, vừa là máy chủ.
- Không có máy chủ trung tâm quản lý mạng.
- Không có máy định tuyến (bộ định tuyến) trung tâm và các peer node có khả năng định tuyến

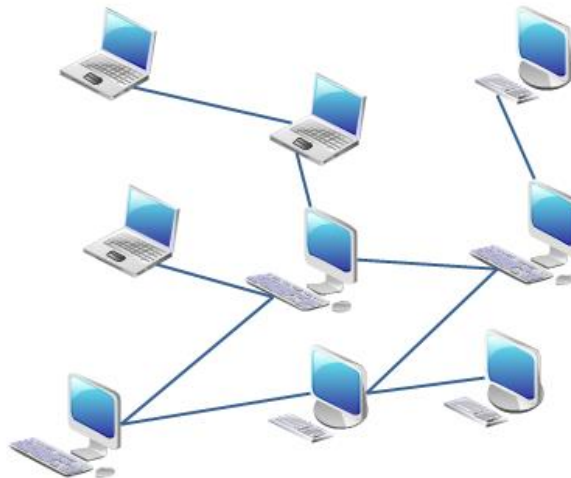
Trong mô hình này các peer node khi tham gia vào mạng cần đều có khả năng tìm vị trí của một peer node khác trong mạng dựa vào một thuật toán có được sử dụng để đánh địa chỉ và tìm thông tin về những peer node khác.

*b. Mạng đồng đẳng lại:*

- Có một máy chủ trung tâm dùng để lưu trữ dữ thông tin của các máy trạm và trả lời các thông tin truy vấn này. Máy chủ trung tâm này có mục đích chính là xử lý và trả lời các thông tin truy vấn từ các nút mạng khác như tham gia mạng, rời mạng, cập nhật thông tin của nút mạng. Nút mạng này được gọi là Bootstrap Node. Vì sự cần thiết và tác dụng của Bootstrap Node nên Bootstrap Node luôn luôn phải được chạy trước tiên và chạy liên tục trong quá trình hoạt động của mạng.
- Sử dụng các trạm định tuyến để xác định địa chỉ IP của các máy trạm.

**2.2.2. Những thách thức cần giải quyết khi tạo nên một mạng P2P**

Thách thức của việc thiết kế một phần mềm hoạt động cho mạng P2P, đó là làm thế nào để các peer có thể giao tiếp được với nhau. Hình bên dưới minh họa một cấu trúc mạng không có tổ chức. Tất cả các peer đều kết nối với Internet, nhưng làm cách nào để những peer này có thể biết địa chỉ của nhau, cũng như là thế nào để chắc chắn rằng một peer giao tiếp đúng với peer mà nó cần?



Hình 2.3: Mô hình mạng P2P

Vấn đề cốt lõi trong một mạng, mà cụ thể ở đây là mạng P2P là làm sao để những Peer Node có thể tìm thấy và giao tiếp chính xác với Peer Node khác mà nó muốn cần phải được giải quyết. Để những peer trong mạng P2P có thể giao tiếp được với nhau, cần giải quyết hai vấn đề sau:

- Nhận dạng Peer trong mạng: một Peer Node trong mạng phải được phân biệt với các Peer Node khác và mang một định danh duy nhất. Khi đó một Peer Node muốn liên lạc với Peer Node khác sẽ sử dụng ID để liên lạc, đảm bảo được điều kiện liên lạc chính xác node mà nó muốn.
- Tìm vị trí của Peer trong mạng: Một liên kết tồn tại giữa hai nút mạng khi một nút mạng biết vị trí của nút mạng kia. Do đó cần phải đưa ra giải pháp làm sao cho Peer Node này có thể chứa hoặc tìm kiếm được thông tin của những Peer Node khác nằm trong mạng, và sử dụng thông tin có được này tạo kết nối và liên lạc với những Peer Node còn lại.

### **2.2.3. Ý tưởng giải quyết vấn đề trong đề tài**

Trong ngữ cảnh nghiên cứu của đề tài này, nghiên cứu sẽ tập trung vào việc implement một phần mềm tạo sự tương tác giữa các peer trong cùng một mạng LAN nhỏ và số lượng peer node tham gia không lớn (khoảng 50 – 100 peer). Trong ngữ cảnh này, từng vấn đề nêu trên sẽ được giải quyết theo những ý tưởng sau.

#### **a. Giải quyết vấn đề thứ nhất – Định danh Peer Node**

- Mỗi Peer Node tham gia vào mạng phải được gán một định danh duy nhất và định danh này sẽ được biết đến bởi các Peer Node khác. Điều này giúp một Peer Node khi muốn liên lạc với một Peer Node khác trong mạng thì chỉ cần gọi định danh của Peer Node đó ra và sử dụng thông tin của Peer Node đó để thực hiện kết nối, vì định danh là duy nhất nên đảm bảo rằng Peer Node liên lạc chính xác với node mà nó muốn.

- Trong phạm vi đề tài này là thực thi một thư viện phần mềm hiện thực hóa mô hình P2P trong cùng một mạng LAN nên để đơn giản thì chương trình sẽ lấy luôn địa chỉ IP của thiết bị trong mạng để làm định danh cho Peer Node. Vì địa chỉ IP của một thiết bị trong một mạng là duy nhất nên điều này thỏa mãn điều kiện thứ nhất là ID của Peer Node phải là duy nhất.
- Trong trường hợp Peer Node thay đổi IP thì IP trong một mạng LAN cũng là duy nhất, do đó khi trường hợp này xảy ra thì điều kiện trên vẫn được đảm bảo.

*b. Vấn đề thứ hai – Tìm vị trí của một Peer Node trong mạng:*

Để giải quyết vấn đề này thì chương trình sử dụng cấu trúc mạng đồng đẳng lai với mô hình Bootstrap Node đã nêu ở trên. Bootstrap Node có nhiệm vụ lưu trữ thông tin của tất cả các Peer Node trong mạng. Bootstrap Node sẽ xử lý và trả lời các yêu cầu thông tin truy vấn với các Peer Node khi Peer Node phát sinh các sự kiện liên quan đến liên kết mạng. Đồng thời Bootstrap Node cũng thực hiện gửi các tác vụ đến Peer Node khi cần thiết để đảm bảo mạng hoạt động ổn định và các Peer Node luôn luôn có thông tin cập nhật về tất cả những Peer Node khác trong mạng.

Ý tưởng cơ bản của mô hình này như sau: một mạng P2P được hình thành bởi các nút mạng và liên kết nút mạng giữa chúng. Một Peer Node được gọi là liên kết với một Peer Node khác trong mạng khi Peer Node đó có chứa thông tin về định danh và vị trí của Peer Node khác. Quy luật này được áp dụng cho tất cả những Peer Node tham gia vào mạng, đó là Peer Node sẽ có tất cả các thông tin về các Peer Node khác có trong mạng. Vấn đề cần giải quyết ở đây là làm sao để Peer Node có thể có được những thông tin này, và phải đảm bảo rằng những thông tin này luôn được cập nhật liên tục khi có một Peer Node tham gia mạng hoặc một Peer Node rời mạng.

- Khi mới tham gia vào mạng, một Peer Node không thể biết chính xác nó phải liên kết với nút mạng nào để có thể có được những thông tin này vì địa chỉ IP tự phát sinh khi kết nối vào mạng LAN. Điều này được giải quyết khi sử dụng Bootstrap Node có địa chỉ IP cố định, cung cấp dịch vụ xử lý yêu cầu truy vấn thông tin Peer Node có trong mạng. Bootstrap Node được chạy trước tiên, cung cấp dịch vụ và hoạt động cho đến khi mạng dừng lại để liên tục xử lý việc truy vấn thông tin cho những Peer Node tham gia vào mạng, cập nhật thông tin khi có một Peer Node rời mạng hoặc cập nhật lại thông tin của node đó. Mỗi khi có một Peer Node tham gia vào mạng, địa chỉ IP Bootstrap Node sẽ được Peer Node này sử dụng để gửi yêu cầu tham gia vào mạng (thông điệp JOIN) đến dịch vụ mà Bootstrap Node cung cấp đồng thời Peer Node này cũng cung cấp một dịch vụ lắng nghe sự kiện thay đổi của mạng từ Bootstrap Node.
- Sau khi nhận được yêu cầu từ Peer Node, Bootstrap Node sẽ xử lý thông tin về Peer Node gửi tới, lấy thông tin của Peer Node đó và chèn vào ***bảng định tuyến (routing table)***, bao gồm tất cả những thông tin về Peer Node có trong mạng, đến đây thông tin Peer Node đã có trong bảng định tuyến. Tiếp đến, Bootstrap Node sẽ gửi bảng định tuyến này cho Peer Node vừa phát sinh yêu cầu qua dịch vụ mà Peer Node đã mở để lắng nghe sự kiện thay đổi của mạng, Peer Node sẽ ghi nhận lại tất cả những thông tin này và tạo ra một bảng định tuyến ***clone*** từ Bootstrap Node. Đến lúc này Peer Node đã chính thức trở thành một node trong mạng P2P và Peer Node sẽ sử dụng bảng định tuyến để thực hiện kết nối và tương tác với các Peer Node khác trong mạng và cung cấp các chức năng chính của chương trình.
- Tuy rằng phiên giao tiếp của Peer Node gửi yêu cầu tham gia mạng đến lúc này đã xong (nếu chỉ xét Peer Node). Nhưng đến lúc này công việc Bootstrap Node vẫn chưa dừng lại, nó tiếp tục gửi broadcast đến tất cả



những Peer Node có trong bảng định tuyến thông điệp một Peer Node vừa được thêm vào mạng (thông điệp NODE\_ADD) và yêu cầu các Peer Node phải cập nhật lại bảng định tuyến của riêng mình cùng với thông tin mà Bootstrap Node kèm theo. Sau khi thực hiện công việc này xong thì yêu cầu tham gia mạng gửi từ Peer Node trên Bootstrap Node mới thật sự kết thúc.

- Khi một Peer Node muốn rời mạng, nó cũng thực hiện quy trình giống như trên nhưng nó sẽ gửi yêu cầu khác (thông điệp LEAVE) đến Bootstrap Node. Khi Bootstrap Node nhận được thông điệp LEAVE từ Peer Node, nó sẽ thực hiện truy vấn trong bảng định tuyến, tìm thông tin về Peer Node được gửi lên và xóa thông tin đó khỏi bảng định tuyến. Sau đó Bootstrap Node sẽ gửi broadcast đến tất cả các Peer Node trong mạng thông điệp NODE\_LEAVE để tất cả các Peer Node có thể cập nhật lại bảng định tuyến của mình. Sau khi tác vụ này diễn ra thì Peer Node sẽ không còn là một node thành viên của mạng nữa vì mọi thông tin liên lạc với Peer Node đã không còn trong bảng định tuyến của các Peer Node khác nữa.
- Khi một Peer Node thay đổi thông tin của mình (thay đổi IP) thì Peer Node sẽ phát sinh yêu cầu đến Bootstrap Node (thông điệp UPDATE) và thực hiện quy trình giống như trên. Sau khi Bootstrap Node nhận được thông điệp này, nó sẽ truy vấn trong bảng định tuyến của mình thông tin của Peer Node vừa được gửi lên và cập nhật lại bảng định tuyến. Sau đó, Bootstrap Node sẽ phát sinh broadcast đến tất cả những Peer Node có trong mạng yêu cầu cập nhật Peer Node (thông điệp NODE\_UPDATE). Các Peer Node sẽ cập nhật lại bảng định tuyến của riêng mình.

Ý tưởng trên đã thực thi được yêu cầu thứ hai là làm thế nào để một Peer Node trong mạng có thể tìm thấy được Peer Node khác. Ý tưởng này cung cấp

một giải pháp đảm bảo các Peer Node luôn luôn có được thông tin cập nhật về trạng thái của tất cả các Peer Node trong mạng. Từ đó khi một Peer Node muốn liên lạc với một Peer Node khác, với bảng định tuyến nó có thể tạo kết nối và liên lạc một cách dễ dàng.

Ý tưởng này tạo nên một mô hình mạng đồng đẳng lai với Bootstrap Node là một máy chủ dữ liệu trung tâm chuyên xử lý và thực hiện các thông tin truy vấn từ các Peer Node trong mạng, đảm bảo rằng dữ liệu của tất cả các thành viên trong mạng đều được cập nhật mới nhất đến bảng định tuyến và thông tin này cũng được cập nhật đến tất cả bảng định tuyến của các Peer Node trong mạng. Bootstrap Node ngoài đóng vai trò xử lý và cung cấp thông tin về các Peer Node trong mạng thì không còn vai trò gì khác nữa. Trong mô hình mạng kiểu này, Bootstrap Node chỉ đóng vai trò xử lý, phân phối dữ liệu, cập nhật thông tin về tình trạng mạng P2P cho các Peer Node để khởi tạo và duy trì hoạt động của mạng. Còn các tác vụ chính thì Peer Node sau khi gia nhập vào mạng sẽ sử dụng mạng P2P để thực hiện tác vụ như chia sẻ file hoặc và VoIP.

#### **2.2.4. Các thành phần trong mạng P2P**

Phần ý tưởng đã nêu rõ hai thực thể chính trong mô hình được thực thi, đó là Bootstrap Node và Peer Node. Sự tương tác giữa các Bootstrap Node và Peer Node tạo cơ sở hình thành và mở rộng mạng các node. Sau khi đã hình thành nên mạng lưới, các Peer Node sẽ kết nối các Peer Node khác mà nó muốn tương tác để thực hiện tác vụ khác. Phần trên cũng đã nêu bật rõ vai trò cơ bản của Bootstrap Node và Peer Node trong mạng cùng những yêu cầu cơ bản của những thực thể này. Phần này sẽ đi vào sâu hơn cấu trúc và các yêu cầu cần thiết phải có của Bootstrap Node và Peer Node.

- a. Bootstrap Node: Như đã nêu ở trên, Bootstrap Node là máy chủ trung tâm đảm trách vai trò tiếp nhận yêu cầu truy vấn dữ liệu, phân phối và cập nhật

thông tin tất cả các node trong mạng đến tất cả các Peer Node, đảm bảo tình trạng mới nhất của mạng đến với các Peer Node. Để làm được điều này thì Bootstrap Node phải:

- Có cơ chế lưu trữ, cập nhật thông tin của tất cả các Peer Node có trong mạng khi có yêu cầu tham gia mạng, update thông tin, hoặc rời mạng của một Peer Node nào đó. Cơ chế này được thực thi bằng cách Bootstrap Node tạo ra một bảng định tuyến và lưu tất cả các thông của các Peer Node còn active (còn hoạt động và chưa rời khỏi mạng). Khi có yêu cầu phát sinh từ bất kỳ một Peer Node nào trong mạng, thông tin của Peer Node sẽ được cập nhật vào bảng định tuyến này. Trong chương trình bảng định tuyến này được thực thi bằng file text định dạng JSON, có tên là “*list\_peer.json*”.
  - Cung cấp dịch vụ lắng nghe và xử lý các yêu cầu từ các Peer Node như tham gia, cập nhật thông tin và rời mạng. Trong chương trình này, Bootstrap Node sẽ lắng nghe và xử lý yêu cầu ở port 6868. Mỗi khi một Peer Node muốn phát sinh yêu cầu thì nó sẽ phát sinh vào cổng này và Bootstrap Node sẽ xử lý.
  - Cơ chế ***broadcast*** đến tất cả các Peer Node có trong mạng về một sự kiện nào xảy ra với bất kỳ Peer Node trong mạng. Trong chương trình, trong pha giao tiếp giữa Peer Node và Bootstrap Node, sau khi xử lý yêu cầu của Peer Node, Bootstrap Node sẽ phát sinh một thông điệp tương đương với yêu đã được xử lý cùng thông tin đến tất cả các Peer Node trong mạng.
- b. Peer Node: là thành phần cốt lõi tạo nên mạng P2P và đảm trách vai trò thực thi mục đích chính của chương trình. Để trở thành một node trong mạng thì cần phải đảm bảo những thành phần sau đây:

- Thực thi phương thức giao tiếp với Bootstrap Node để thực hiện phương thức giao tiếp đã được quy định trong chương trình (sẽ được diễn tả ở phần kế tiếp). Phương thức đó bao gồm các thực thi về yêu cầu tham gia (JOIN), yêu cầu cập nhật (UPDATE), yêu cầu rời mạng (LEAVE).
- Cơ chế quản lý thông tin của các Peer Node khác trong mạng nhận được bảng định tuyến từ Bootstrap Node sau khi thực hiện yêu cầu JOIN.
- Cung cấp dịch vụ lắng nghe sự thay đổi trạng thái của mạng khi bất kỳ Peer Node nào trong mạng thay đổi thông tin và Bootstrap Node gửi broadcast đến. Trong chương trình của đề tài, Peer Node sẽ tạo một dịch vụ xử lý yêu cầu của Bootstrap Node ở port 8686.

#### 2.2.5. Phương thức giao tiếp giữa các thành phần trong mạng

Các node trong mạng cần phải giao tiếp được với nhau để có thể thông báo tình trạng và dữ liệu cần trao đổi, cụ thể ở đây là sự trao đổi thông tin giữa Bootstrap Node và Peer Node. Cách thức giao tiếp giữa các node trong chương trình được sử dụng bằng thông điệp. Mỗi một thông điệp mang một yêu cầu tác vụ và thông tin thông điệp mà hai bên hiểu được và dựa vào tác vụ đó bên nhận sẽ có cách thức đáp ứng tương ứng.

Một quá trình bắt đầu gửi thông điệp đến khi kết thúc gửi thông điệp giữa hai node được gọi là một *phiên giao tiếp*.

Một phiên giao tiếp bao gồm những pha:

- Khi muốn giao tiếp, client node sẽ phát sinh một thông điệp yêu cầu server node bắt đầu một phiên giao tiếp.
- Server node chấp nhận và gửi lại thông điệp chấp nhận phiên giao tiếp.
- Tiếp đến, client node sẽ gửi thông điệp đến server node. Server node sẽ nhận thông điệp và giữ lại.
- Client node gửi yêu cầu kết thúc session giao tiếp.

- Server node chấp nhận và gửi lại thông điệp chấp nhận kết thúc phiên giao tiếp.

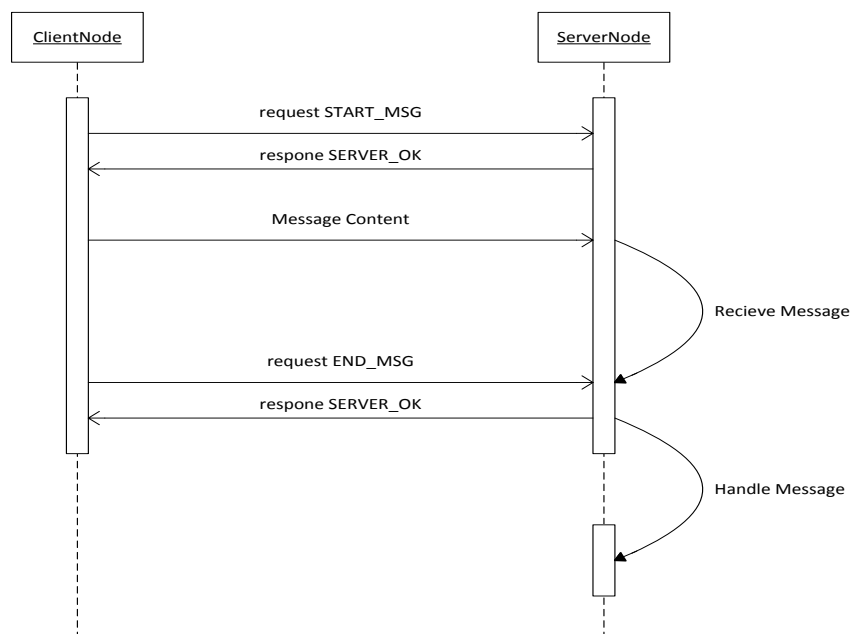
Sau khi phiên giao tiếp kết thúc, server node sẽ xử lý thông điệp. Nếu có phản hồi lại client node thì server node sẽ tạo ra một phiên giao tiếp với node mà nó cần và quy trình lặp lại như quy trình vừa trên và server node lúc này sẽ đóng vai trò là client node yêu cầu phiên giao tiếp.

a. Cách thức gửi thông điệp giữa các node trong mạng:

- Mỗi khi một node muốn giao tiếp với một node khác, nó sẽ thiết lập một phiên gửi message tới node mà nó muốn giao tiếp. Node nhận thông điệp sẽ đóng vai trò là server node và node gửi thông điệp sẽ đóng vai trò là client node. Quá trình thiết lập phiên gửi thông điệp sẽ được diễn ra bởi client node và server node bằng các request và response.
- **Request** là một yêu cầu được gửi từ client node lên server node thông báo cho server node biết rằng client node yêu cầu một tác vụ giao tiếp với server node. Request trong chương trình được xây dựng trên định dạng text, nghĩa là request này là một chuỗi từ mà client node gửi lên cho server node một cách riêng biệt, khi server node gặp những từ này thì theo quy định server node sẽ có những cách đáp ứng phù hợp. Có các loại request sau:
  - **START\_MSG** (“start\_msg”): gửi yêu cầu tác vụ thông báo cho server node biết rằng client node bắt đầu phiên giao dịch gửi message. Sau khi nhận được request này, server node sẽ chấp nhận và gửi response **SERVER\_OK** về client. Sau đó client sẽ gửi message và server sẽ chấp nhận message.
  - **END\_MSG** (“end\_msg”): gửi yêu cầu tác vụ thông báo cho server node biết rằng client node đã gửi xong message và muốn kết thúc phiên gửi message. Sau khi nhận được request này, server node gửi response thông

báo SERVER\_OK cho client node. Tiếp đó, server node sẽ xử lý thông điệp mà client node vừa gửi.

- Sau khi phiên giao tiếp thông điệp giữa server node và client node được thiết lập, client node sẽ dùng request sẽ tiếp tục gửi nội dung của thông điệp lên server node.
- Response là một đáp trả từ server node đến client node thông báo rằng server node đã chấp nhận request của client node. Response cũng được xây dựng dựa trên định dạng text giống như request, khác ở đây là cách thức sử dụng của reponse khác với request.
  - SERVER\_OK (“ok”): thông điệp đáp trả từ server node tới client node rằng yêu cầu của client node đã được chấp nhận bởi server node.
- Hình dưới đây mô tả một phiên giao tiếp thông điệp đầy đủ giữa các node, node gửi request yêu cầu thiết lập phiên gửi thông điệp là client node và node đáp ứng yêu cầu là server node.



Hình 2.4: Phiên thông điệp

b. Thông điệp (message):

Thành phần quan trọng trong phiên giao tiếp và trao đổi thông tin giữa các node chính là thông điệp. Thông điệp là một yêu cầu tác vụ và thông tin về tác vụ đó được tạo ra theo một quy định từ trước được hiểu bởi tất cả các node trong mạng, và các node sẽ dựa trên đó để xử lý các thông điệp. Trong chương trình thông điệp được gửi sau khi phiên giao tiếp được thiết lập.

Thông điệp trong chương trình có định dạng json. Thông điệp gồm 2 thành phần: tác vụ thông điệp (msg\_type) và nội dung thông điệp (msg\_data). Cấu trúc thông điệp như sau:

```
{“msg_type” : “loại thông điệp”, “msg_data” : “nội dung thông điệp”}
```

c. Các loại message (msg\_type):

Loại message được mô tả bởi phần đầu của thông điệp – phần “msg\_type”, mô tả tác vụ được yêu cầu của client node lên server node. Trong đặc tả của chương trình sử dụng các loại thông điệp sau:

- Thông điệp JOIN (“join”): thông điệp này thông báo yêu cầu tác vụ một Peer Node muốn tham gia vào mạng. Khi một Peer Node muốn tham gia vào mạng, nó sẽ gửi thông điệp này đến Bootstrap Node thông báo rằng nó muốn tham gia vào mạng.
- Thông điệp LEAVE (“leave”): thông điệp này thông báo yêu cầu tác vụ một Peer Node muốn rời khỏi mạng đến Bootstrap Node.
- Thông điệp UPDATE (“update”): trong chương trình sau khi một Peer Node trở thành một node trong mạng thì sẽ sử dụng địa chỉ IP của mình để làm định danh và liên lạc. Do đó nếu thông tin Peer Node có thay đổi thì Peer Node sẽ sử dụng thông điệp này để thông báo cho Bootstrap Node.
- Thông điệp TRANSFER\_LIST (“transfer\_list”): thông điệp này được phát sinh bởi Bootstrap Node đến Peer Node sau khi nhận được thông điệp JOIN từ Peer Node đó. Thông điệp này thông báo với Peer Node rằng Bootstrap

Node đã xử lý xong thông điệp JOIN của Peer Node đó, đã thêm Peer Node đó vào bảng định tuyến và nội dung thông điệp mang danh sách bảng định tuyến.

- Thông điệp NODE\_ADD (“node\_add”): thông điệp này được Bootstrap Node gửi đến cho tất cả các Peer Node trong mạng thông báo rằng có một node vừa được thêm vào mạng. Và nội dung của thông điệp chính là thông tin node vừa thêm. Sau khi nhận được thông điệp, các Peer Node sẽ xử lý và thêm thông tin Peer Node vào trong bảng định tuyến của mình.
- Thông điệp NODE\_UPDATE (“node\_update”): thông điệp này được Bootstrap Node gửi đến cho tất cả các Peer Node trong mạng thông báo rằng có một node vừa được cập nhật thông tin. Và nội dung thông điệp là thông tin node vừa update. Các Peer Node sẽ tìm trong bảng định tuyến của mình và cập nhật lại thông tin.
- Thông điệp NODE\_LEAVE (“node\_leave”): thông điệp này được Bootstrap Node gửi đến cho tất cả các Peer Node trong mạng thông báo rằng có một node vừa rời khỏi mạng. Nội dung thông điệp là thông tin node vừa rời đi. Các Peer Node sẽ tìm trong bảng định tuyến của mình và xóa thông tin node đó đi.
- Thông điệp PING (“ping”): thông điệp này được phát sinh từ một node đến một node kiểm tra Peer Node có đang hoạt động hay không.

d. Nội dung message (msg\_data):

Nội dung của thông điệp được mô tả bởi phần sau của thông điệp – phần “msg\_data”. Nội dung thông điệp bao gồm thông tin cần được gửi bởi client node đến server node. Trong chương trình nội dung thông điệp là danh sách thông tin các Peer Node được lưu dưới định dạng JSON. Mỗi Peer Node đều có thông tin của riêng mình gọi là thông tin Peer Node. Trong chương trình thông tin Peer Node gồm có địa chỉ IP của Peer Node và tên người dùng (user name)



người dùng đặt cho node. Tên người dùng này được sử dụng trong địa chỉ SIP. Nội dung của thông điệp giống như sau:

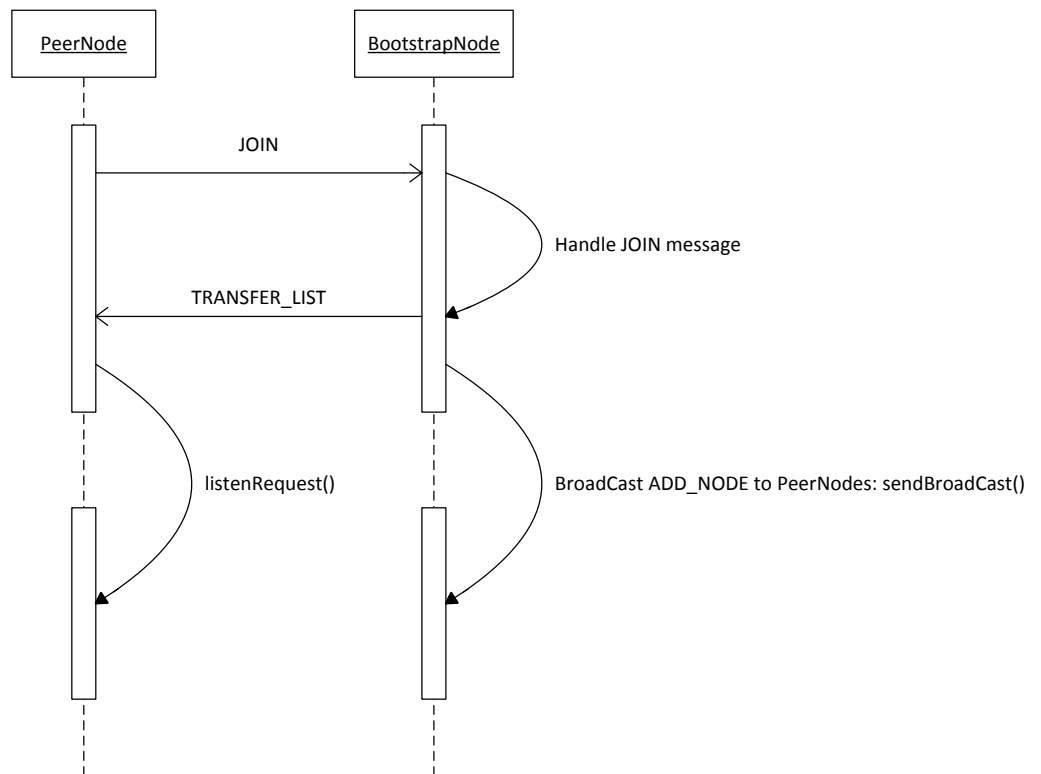
```
{"list_peer":[{"address":"192.168.0.104","username":"SangNguyen"}]}
```

#### **2.2.6. Các pha giao tiếp giữa các thành phần trong mạng**

Phần bên trên đã mô tả các message được dùng để giao tiếp giữa các node trong mạng. Phần này sẽ trình bày rõ chi tiết tác vụ của từng message.

a. Message JOIN – TRANSFER\_LIST: trong quá trình giao tiếp thông điệp này, Peer Node muốn tham gia vào mạng sẽ gửi thông điệp JOIN đến Bootstrap Node. Quá trình cụ thể như sau:

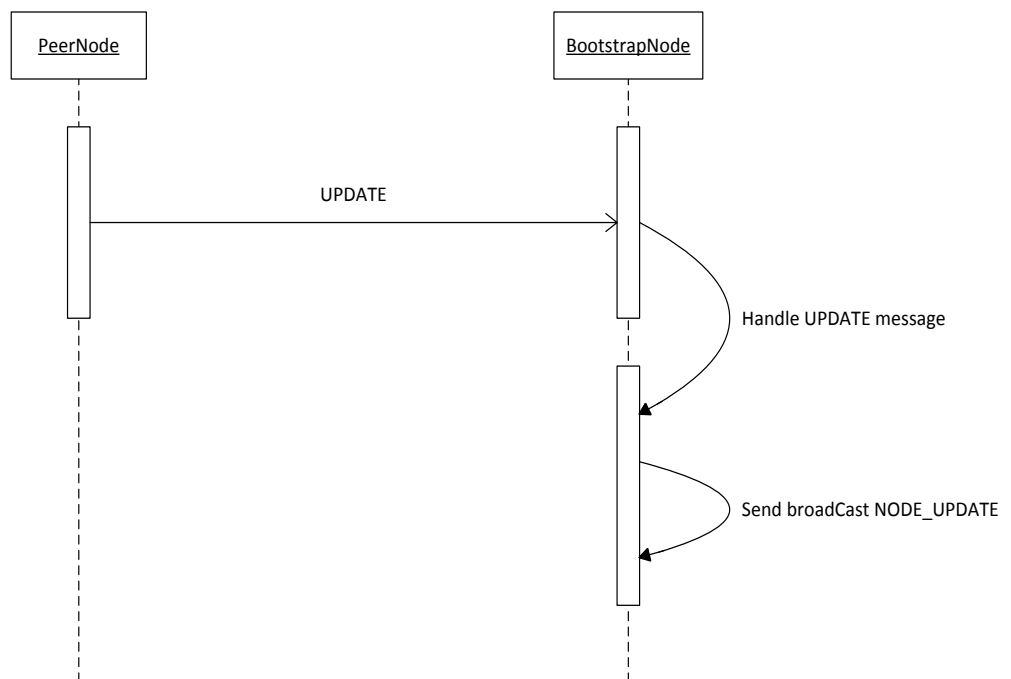
- Peer Node muốn tham gia vào mạng gửi message JOIN đến Bootstrap Node.
- Bootstrap Node xử lý thông điệp, lấy danh sách thông tin Peer Node trong nội dung thông điệp, thêm thông tin vào trong bảng định tuyến.
- Bootstrap Node phát sinh thông điệp TRANSFER\_LIST đến Peer Node kèm theo bảng định tuyến vào nội dung thông điệp.
- Peer Node nhận thông điệp trên, xử lý thông điệp và lấy bảng định tuyến được gửi từ Bootstrap Node, lưu vào trong dữ liệu của mình. Đến lúc này Peer Node đã trở thành một node trong mạng.
- Đến lúc này phiên giao tiếp giữa Peer Node gửi thông điệp JOIN với Bootstrap Node đã kết thúc.
- Bootstrap Node sẽ broadcast message NODE\_ADD đến tất cả các node trong mạng để thông báo rằng có một Peer Node vừa tham gia vào mạng cùng với nội dung thông điệp là thông tin Peer Node đó.
- Các Peer Node nhận được broad cast NODE\_ADD từ Bootstrap Node sẽ lấy thông tin Peer Node vừa được thêm vào mạng thêm vào bảng định tuyến của mình.



Hình 2.5: Thông điệp JOIN

- b. Message UPDATE: thông điệp này được phát sinh giữa Peer Node và Bootstrap Node để thông báo rằng thông tin Peer Node được cập nhật. Các hoạt động cụ thể như sau:
- Peer Node gửi message UPDATE kèm theo nội dung là thông tin Peer Node cập nhật đến Bootstrap Node.
  - Bootstrap Node nhận message và lấy thông tin của Peer Node cập nhật ra từ nội dung message. Sau đó Bootstrap Node truy vấn trong bảng định tuyến của mình tìm thông tin Peer Node được cập nhật và cập nhật thông tin.

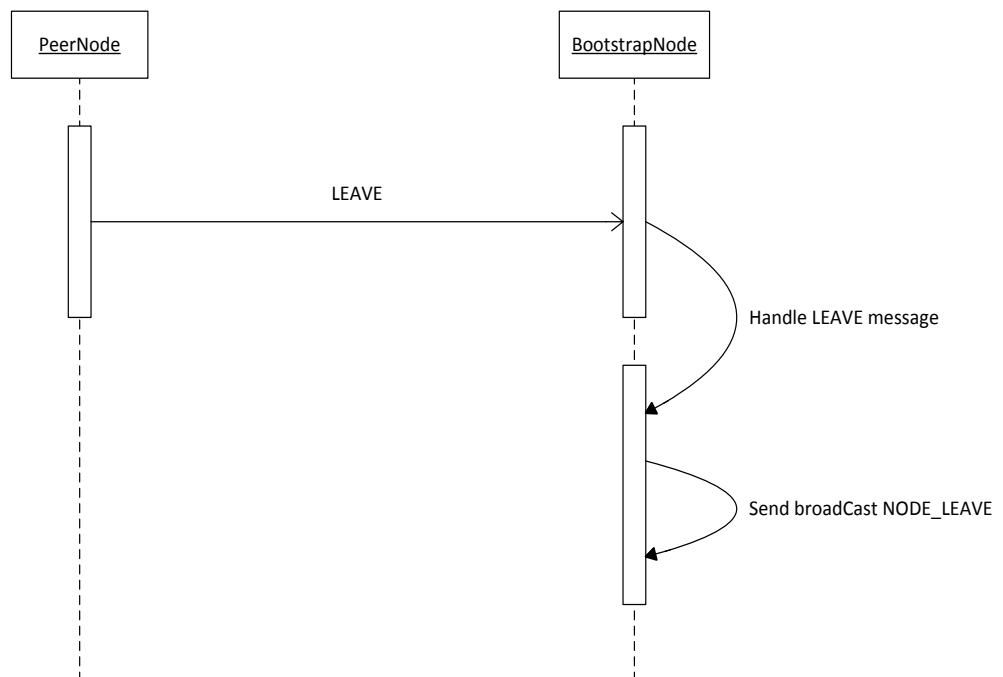
- Bootstrap Node broadcast message NODE\_UPDATE đến tất cả các Peer Node trong mạng kèm theo thông tin của Peer Node được cập nhật.
- Các Peer Node nhận được broadcast sẽ xử lý thông tin, lấy thông tin của Peer Node được cập nhật, truy vấn trong bảng định tuyến của mình thông tin Peer Node đó và cập nhật.
- Đến đây thông tin Peer Node đã được cập nhật.



Hình 2.6: Thông điệp UPDATE

- c. Message LEAVE: thông điệp này được phát sinh giữa Peer Node đến Bootstrap Node khi Peer Node muốn rời khỏi mạng. Các hoạt động diễn ra như sau:
- Peer Node gửi message LEAVE kèm theo nội dung là thông tin Peer Node đến Bootstrap Node.

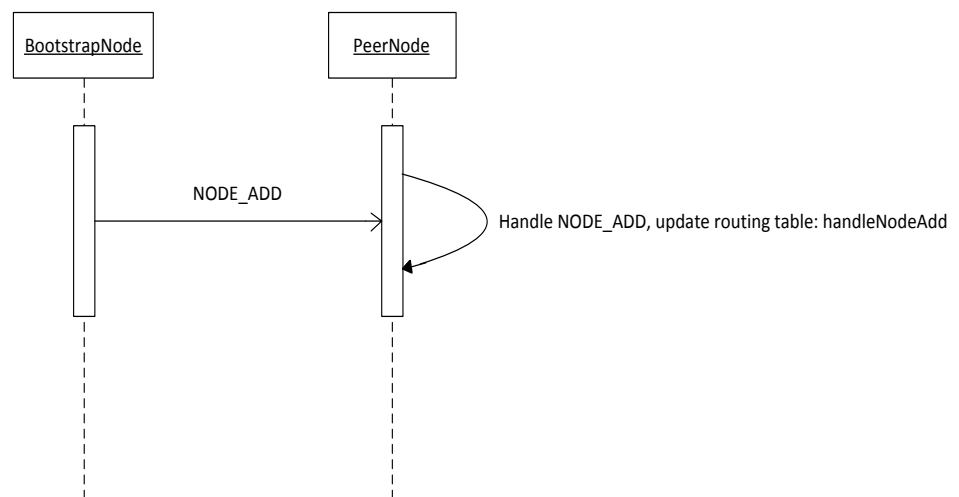
- Bootstrap Node nhận message và lấy thông tin của Peer Node từ nội dung message. Sau đó Bootstrap Node sẽ truy vấn trong bảng định tuyến của mình để tìm thông tin Peer Node muốn rời mạng. Sau khi tìm thấy Bootstrap Node sẽ xóa thông tin Peer Node ra khỏi bảng định tuyến.
- Bootstrap Node broadcast message NODE\_LEAVE đến tất cả Peer Node trong mạng kèm theo thông tin của Peer Node muốn rời khỏi mạng.
- Các Peer Node nhận được broadcast sẽ xử lý thông tin, lấy thông tin của Peer Node muốn rời mạng, truy vấn trong bảng định tuyến của mình thông tin Peer Node và xóa khỏi bảng định tuyến.
- Đến lúc này Peer Node đã thực sự rời khỏi mạng.



Hình 2.7: Thông điệp LEAVE

d. Message NODE\_ADD: thông điệp này được gửi từ Bootstrap Node đến Peer Node trong quá trình broadcast message sau khi một Peer Node tham gia vào mạng. Các hoạt động diễn ra như sau:

- Bootstrap Node sẽ gửi message NODE\_ADD cùng thông tin Peer Node tham gia vào mạng đến Peer Node.
- Peer Node nhận được message này sẽ lấy thông tin Peer Node trong thông điệp ra, thêm vào bảng định tuyến của mình.

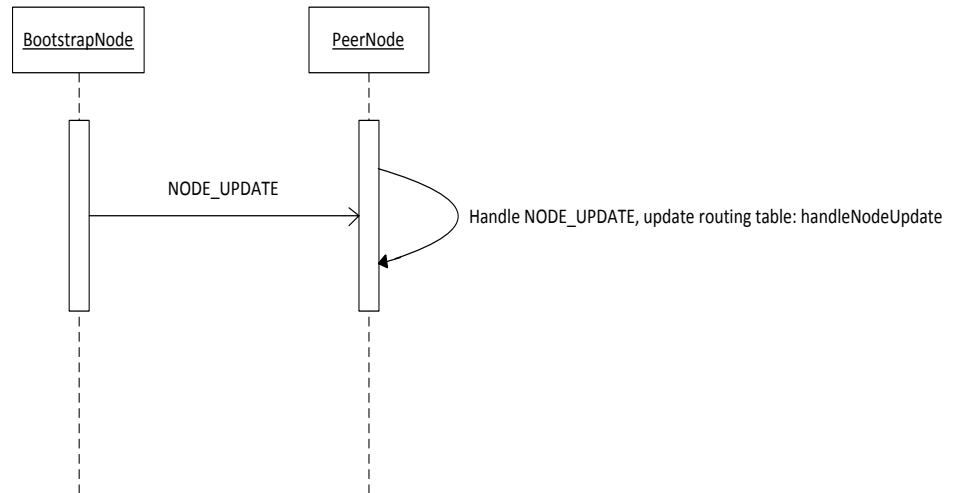


Hình 2.8: Thông điệp NODE ADD

e. Message NODE\_UPDATE: thông điệp này được gửi từ Bootstrap Node đến Peer Node trong quá trình broadcast message sau khi một Peer Node muốn cập nhật thông tin của mình. Các hành động tương tự như broadcast message NODE\_ADD.

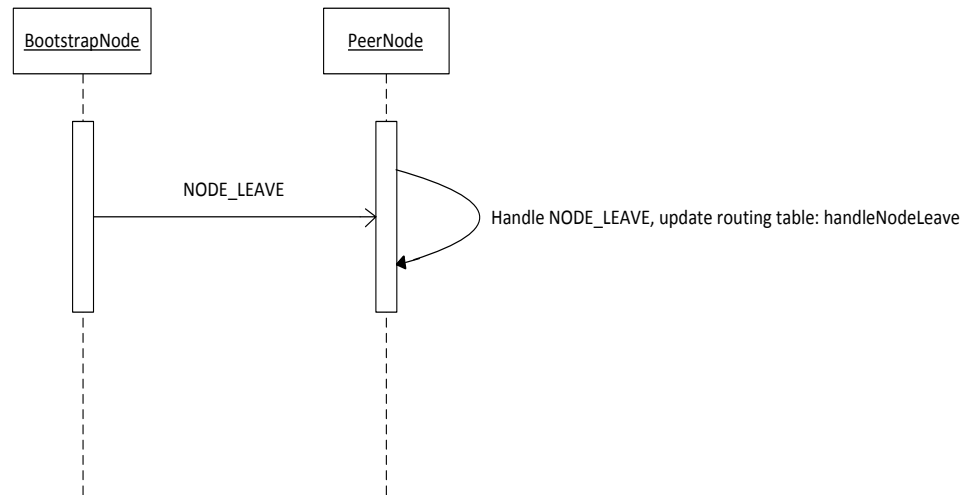
- Bootstrap Node sẽ gửi message NODE\_UPDATE cùng thông tin Peer Node cập nhật đến Peer Node.

- Peer Node sẽ xử lý message và sẽ lấy thông tin muốn cập nhật trong message, truy vấn thông tin Peer Node cập nhật trong bảng định tuyến và cập nhật thông tin Peer Node đó.



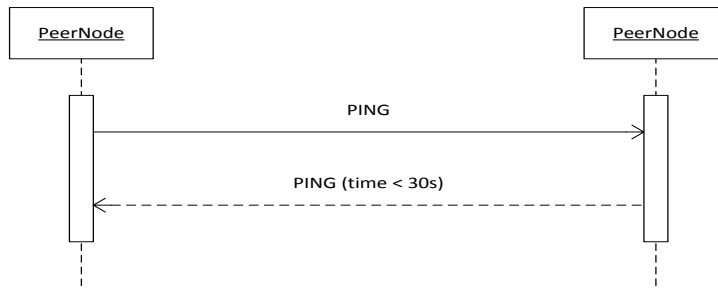
Hình 2.9: Thông điệp NODE UPDATE

- f. Message NODE LEAVE: thông điệp này được gửi từ Bootstrap Node đến Peer Node trong quá trình broadcast message sau khi một Peer Node muốn rời khỏi mạng. Các hành động tương tự như broadcast message NODE\_ADD.
- Bootstrap Node sẽ gửi message NODE\_LEAVE cùng thông tin Peer Node muốn rời khỏi mạng đến Peer Node.
  - Peer Node sẽ xử lý message và sẽ lấy thông tin Peer Node muốn rời khỏi mạng trong message, truy vấn thông tin Peer Node muốn rời khỏi mạng trong bảng định tuyến và xóa thông tin Peer Node đó.

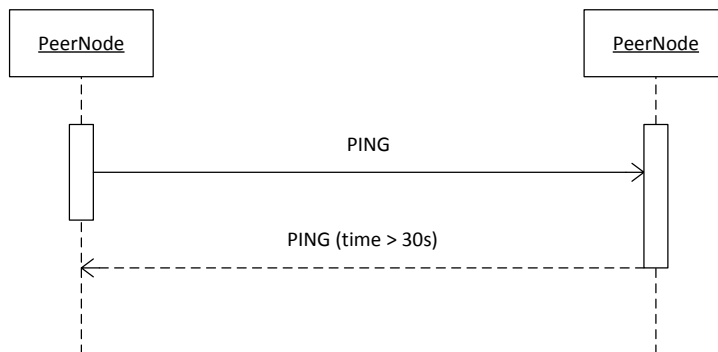


Hình 2.10: Thông điệp NODE LEAVE

- g. Message PING: thông điệp PING được gửi giữa các Peer Node với nhau để kiểm tra tình trạng của Peer Node. Quy trình diễn ra như sau:
- Peer Node muốn kiểm tra Peer Node khác phát sinh message PING.
  - Peer Node phát message chờ Peer Node nhận message trong 30s.
  - Nếu có sự hồi đáp từ Peer Node nhận message gửi message PING lại cho Peer Node phát message thì quá trình ping thành công.
  - Nếu trong thời gian chờ mà Peer Node nhận message không có phản hồi lại cho Peer Node phát message thì quá trình ping sẽ có kết quả là time out.
  - Các hình sequence dưới đây mô tả về hai quá trạng thái ping của Peer Node.



Hình 2.11: PING thành công



Hình 2.12: PING không thành công

## 2.2.7. Ưu điểm và nhược điểm của mô hình

### a. Ưu điểm

- Mô hình dễ thực thi vì có thiết kế đơn giản.
- Cách thức giao tiếp thông điệp dễ dàng, đơn giản.
- Một node tham gia vào mạng thì sẽ có thể liên lạc ngay tức thời với các node khác trong mạng vì Peer Node đã sao chép bảng định tuyến từ Bootstrap Node.
- Các tình trạng của mạng ngay lập tức được cập nhật tới các Peer Node thông qua Bootstrap Node nên Peer Node luôn luôn biết được tình trạng mạng hiện thời.



*b. Nhược điểm*

- Vì phải sử dụng Bootstrap Node cho tác vụ quản lý mạng nên vẫn phải phụ thuộc vào Bootstrap Node. Nếu Bootstrap Node không hoạt động hoặc hoạt động không ổn định thì mạng sẽ bị ảnh hưởng theo.
- Khi mạng trở nên lớn với nhiều Peer Node tham gia hơn thì cơ chế sử dụng bảng định tuyến và update bảng định tuyến không khả thi vì công việc này chiếm rất nhiều tài nguyên xử lý cũng như băng thông của mạng.

*c. Hướng phát triển và cách khắc phục*

- Để giải quyết vấn đề không phụ thuộc vào Bootstrap Node nữa, mạng sẽ phát triển thành mạng p2p đúng nghĩa, nghĩa là không có nút xử lý trung tâm nữa. Thay vào đó các Peer Node đều có khả năng tự định tuyến.
- Để thực thi mô hình này mô hình sẽ phát triển và sử dụng kỹ thuật distributed hash table (DHT). Với mô hình này, các Peer Node sẽ có khả năng tự định tuyến và mỗi tác vụ sẽ sử dụng tài nguyên mạng thấp hơn.

## **2.3. Thành phần SIP**

### **2.3.1. Giới thiệu SIP**

SIP (Session Initiation Protocol – Giao thức khởi tạo phiên) là một giao thức điều khiển lớp ứng dụng mà có thể thiết lập, chuyển đổi và kết thúc các phiên đa phương tiện như các cuộc gọi điện thoại Internet.

SIP (Session Initiation Protocol) là một giao thức được thiết kế bởi Henning Schulzrinne và Mark Handley vào năm 1996. Sau này được phát triển và chuẩn hóa trong RFC 3261 dưới sự bảo hộ của IETF ( Internet Engineering Task Force).

### **2.3.2. So sánh H323 và SIP và lý do chọn SIP**

H.323 là giao thức hoàn thiện hơn trong hai giao thức, nhưng do thiếu linh hoạt nên có thể phát sinh nhiều vấn đề. SIP là giao thức nhỏ gọn hơn, nhưng lại có khả

năng tích hợp dễ dàng với các ứng dụng Internet. Thật khó nói rằng cuối cùng giao thức nào sẽ thống trị, các phân tích khách quan dưới đây sẽ giúp quyết định giao thức nào là phù hợp hơn đối với mỗi loại hình ứng dụng.

	<b>H.323</b>	<b>SIP</b>
Kiến trúc	H.323 phủ hầu hết các dịch vụ, như trao đổi năng lực giữa 2 terminal, điều khiển hội thảo, signaling, QoS, đăng ký, tìm kiếm dịch vụ, ...	SIP là modular do nó phủ hầu hết các chức năng signaling căn bản, định vị người dùng và đăng ký. Các đặc tính khác thể hiện trong các giao thức khác.
Các thành phần	Terminal/Gateway	UA
Gatekeeper	Servers	
Giao thức kế thừa	RAS/Q.931	HTTP/SMTP
H.245	SDP	
Chức năng điều khiển cuộc gọi		
Call Transfer	Có	Có
Call Forwarding	Có	Có
Call Holding	Có	Có
Call Parking/Pickup	Có	Có
Call Waiting	Có	Có
Message Waiting Indication	Có	Không
Name Identification	Có	Không
Call Completion on Busy Subscriber	Có	Có

	<b>H.323</b>	<b>SIP</b>
Call Offer	Có	Không
Call Intrusion	Có	Không
<b>Các tính năng cao cấp</b>		
Multicast Signaling	Có, các yêu cầu vị trí (LRQ) và tự động phát hiện gatekeeper (GRQ).	Có, thông qua thông điệp INVITE.
Third-party Call Control	Có, chức năng tạm ngưng (pause) và định tuyến lại được định nghĩa trong H.323. Các chức năng phức tạp hơn được định nghĩa trong chuỗi tiêu chuẩn H.450x.	Có trong SIP, như đã mô tả trong Internet Drafts.
Conference	Có	Có
Click for Dial	Có	Có
<b>Scalability</b>		
Large Number of Domains	H.323 ban đầu chỉ hỗ trợ cho LAN, không cho phép định vị trên WAN. Sau này khái niệm Zone được đưa thêm nhằm cho phép định vị trên WAN. Các thủ tục được định nghĩa cho chức năng “user location” qua các zones. Từ định nghĩa	SIP có hỗ trợ việc định vị qua mạng WAN. Khi có nhiều server tham gia vào việc thiết lập một cuộc gọi, SIP sử dụng thuật toán loop detection tương tự như thuật toán đã được sử dụng trong BGP, vốn có thể được hiện theo lối

	<b>H.323</b>	<b>SIP</b>
	truyền thông giữa các domain cho phép phân giải địa chỉ, ủy quyền truy cập và báo cáo sử dụng giữa các domain. Trong tìm kiếm qua nhiều domain, vẫn chưa có cách đơn giản giúp phát hiện lặp lại (loop detection). Thực hiện việc phát hiện lặp lại có thể được thực hiện thông qua sử dụng mục PathValue, nhưng lại đưa đến nhiều vấn đề phát sinh liên quan đến tính scalability.	stateless, do đó tránh được các vấn đề liên quan đến tính scalability. Các Server đăng ký và định hướng lại được thiết kế để cung cấp các tính năng “user location”
Large Number of Calls	Điều khiển cuộc gọi của H.323 có thể được cài đặt theo lối stateless. Có thể sử dụng một gateway (được định nghĩa theo H.225) nhằm hỗ trợ cho gatekeeper trong việc thực hiện cân bằng tải qua nhiều gateway.	Chức năng điều khiển cuộc gọi cũng có thể cài đặt theo lối stateless. SIP cho phép UA và server theo tỷ lệ n:n. SIP sử dụng CPU ít hơn cho việc phát sinh các thông điệp signaling; do đó một server có thể xử lý nhiều giao tác. SIP đã đặc tả một phương pháp cân bằng tải dựa trên cơ chế

	<b>H.323</b>	<b>SIP</b>
		chuyển đổi mẫu tin SRV của DNS.
Connection state	Stateful hoặc Stateless	Stateful hoặc Stateless. Một cuộc gọi the SIP không chỉ độc lập với sự tồn tại của kết nối ở tầng vận chuyển, mà còn thay thế các tín hiệu kết thúc cuộc gọi.
Internationalization	Có, H.323 sử dụng Unicode cho các thông tin dạng văn bản.	Có, SIP sử dụng Unicode (ISO 10646-1), được mã hóa như UTF-8, cho tất cả các chuỗi văn bản. SIP cung cấp khả năng chỉ định ngôn ngữ và các tùy chọn về ngôn ngữ.
Security	Định nghĩa các cơ chế an ninh và các phương tiện để thương lượng thông qua H.235, cho phép sử dụng SSL cho an ninh ở tầng vận chuyển.	SIP cung cấp chức năng chứng thực người gọi và người được gọi thông qua cơ chế HTTP. Các nội dung chứng thực được mã hóa nhờ SSL/TSL, ngoài ra SIP có thể sử dụng bất kỳ cơ chế an ninh nào ở tầng vận

	<b>H.323</b>	<b>SIP</b>
		chuyển hoặc cơ chế an ninh kiểu như HTTP (như SSH, hoặc S-HTTP).
Interoperability among Versions	Tương tích hoàn toàn với các phiên bản H.323 trước .	Trong SIP, phiên bản mới hơn có thể vứt bỏ những tính năng trước đó không còn phù hợp. Cách tiếp cận này giữ cho kích thước mã lệnh được nhỏ gọn và giảm thiểu độ phức tạp giao thức, nhưng lại đánh mất khả năng tương thích với các phiên bản trước.
Implementation Interoperability	H.323 có cung cấp một hướng dẫn cho người thi công, tài liệu này vốn làm rõ các tiêu chuẩn và các giúp đỡ theo hướng cho phép cộng tác giữa các thi công khác nhau.	Cho đến nay SIP vẫn chưa cung cấp một hướng dẫn thống nhất việc thi công.
Codecs	H.323 cho phép tích hợp hầu hết các Codec, không chỉ là các codecs của ITU-T.	SIP cho phép các codec IANA Registered.

	<b>H.323</b>	<b>SIP</b>
Call Forking	H.323 gatekeeper có thể kiểm soát call signaling và có thể tách chia gọi tới bất kỳ một thiết bị nào, một cách đồng thời.	SIP proxy có thể kiểm soát call signaling và có thể tách chia gọi tới bất kỳ một thiết bị nào, một cách đồng thời.
Transport Protocol	Cung cấp cả hai: TCP và UDP.	Cung cấp cả hai: TCP và UDP.
Message Encoding	H.323 mã hóa các thông điệp theo dạng thức nhị phân nhỏ gọn, thích hợp cho các kết nối băng tần hẹp và rộng.	Các thông điệp SIP được mã hóa theo dạng bảng mã ASCII, thích hợp cho con người đọc.
Addressing	Cơ chế định vị bằng địa chỉ linh hoạt, bao gồm các URL và số E.164.	SIP chỉ hiểu các địa chỉ kiểu URL.
PSTN Interworking	H.323 vay mượn từ giao thức PSTN cổ điển, nghĩa là Q.931 và do đó rất thích hợp với việc tích hợp với PSTN. Tuy nhiên, H.323 không làm việc với các công nghệ chuyển mạch vòng của PSTN như SIP,	SIP không có sự tương đồng nào với PSTN

	<b>H.323</b>	<b>SIP</b>
	H.323 chỉ làm việc với mạng chuyên mạch gói.	
Loop Detection	Có. Gatekeeper có thể phát hiện vòng loop bằng cách tìm xem trong mục CallIdentifier và destinationAddress trong các thông điệp xử lý cuộc gọi. Nếu tổ hợp của những mục này khớp với cuộc gọi đang có, nghĩa là có một vòng loop.	Có, thông điệp VIA của SIP cho phép làm điều này. Tuy nhiên, hơi phức tạp. Có thể sử dụng Max-Forward để giới hạn số hop, và do đó giới hạn loop.
Minimum Ports for VoIP Call	5 (Call signaling, 2 RTP, and 2 RTCP.)	5 (Call signaling, 2 RTP, and 2 RTCP.)
Video and Data Conferencing	H.323 hỗ trợ đầy đủ cho cả hội thảo truyền hình lẫn dữ liệu. Các thủ tục được xây dựng nhằm cho phép kiểm soát tiến trình hội thảo cũng như đồng bộ hóa giữa cử động môi và âm thoại trong video stream.	SIP có hỗ trợ một cách có giới hạn cho video và không hỗ trợ cho data conferencing như T.120. SIP cũng không có giao thức điều khiển hội nghị và không có cơ chế để đồng bộ giữa cử động môi và video stream.

Bảng 2.1: So sánh H.323 và SIP



Từ việc so sánh H.323 và SIP ở trên. Nhóm em nhận thấy SIP là một giao thức nhỏ đơn giản, dễ hiểu. Quan trọng hơn là SIP đáp ứng đầy đủ các chức năng cơ bản của signaling. Vì thế nó đáp ứng đủ 2 yếu tố nhóm cần đó là:

- Đơn giản, dễ hiểu về giao thức
  - Đáp ứng cơ bản về chức năng cuộc gọi
- ⇒ Qua những lý do trên, nhóm đã quyết định chọn giao thức SIP.

### 2.3.3. Mục tiêu của SIP

Mục tiêu chính của SIP là truyền thông giữa các thiết bị đa phương tiện. SIP là cho việc truyền thông trở nên khả thi nhờ áp dụng hai giao thức RTPP/RTCP và SDP.

Giao thức RTP được sử dụng để vận chuyển dữ liệu voice theo thời gian thực, trong khi giao thức SDP được sử dụng để thương lượng giữa các bên tham gia về năng lực, kiểu mã hóa, ...

SIP được thiết kế phù hợp với mô hình Internet. Nó là một giao thức báo hiệu điểm-điểm, trong đó toàn bộ logic được lưu trong các thiết bị cuối (ngoại trừ định tuyến những thông điệp SIP). Trạng thái cũng được lưu trong thiết bị cuối.

Do đó, SIP là một giao thức điều khiển tầng ứng dụng, một giao thức signaling cho Internet Telephony. SIP có thể thiết lập các phiên theo từng tính năng như audio/videoconferencing, trò chơi trực tuyến, và chuyển cuộc gọi, được triển khai trên các mạng IP do đó cho phép các nhà cung cấp dịch vụ tích hợp các dịch vụ điện thoại IP với dịch vụ Web, e-mail, chat. Nó dựa trên HTTP (Hyper Text Transfer Protocol) và SMTP (Simple Mail Transfer).

### 2.3.4 Các thành phần trong SIP

a. User Agent(UA):

UA là thiết bị đầu cuối trong mạng SIP. UA có thể là một máy tính cài phần mềm SIP, có thể là điện thoại SIP, điện thoại di động, PDA...

UA thường được đề cập tới là UA Server (UAS) và UA Client (UAC). UAS và UAC chỉ là các thực thể logic, mỗi UA đều chứa 1 UAS và UAC.

UAC là dùng gửi yêu cầu và nhận trả lời. Còn UAS thì nhận yêu cầu và gửi trả lời.

*b. Proxy Server:*

Proxy Server là thiết bị trung gian giữ các UA. Proxy Server chuyển các yêu cầu và trả lời giữa các UA.

Có 2 loại Proxy Server là Proxy Server trạng thái (stateful) và Proxy Server không trạng thái (stateless).

- Proxy Server không trạng thái: Proxy Server không trạng thái đơn giản chỉ là một bộ chuyển tiếp bản tin. Nó chuyển tiếp các bản tin độc lập với nhau. Mặc dù các bản tin được sắp xếp vào các giao dịch nhưng nó không quan tâm đến giao dịch. Proxy Server không trạng thái truyền nhanh hơn Proxy Server trạng thái. Một trong những hạn chế của Proxy Server không trạng thái là nó không có khả năng truyền lại các bản tin và thực hiện các định tuyến phức tạp hơn.
- Proxy Server trạng thái: Proxy Server trạng thái phức tạp hơn. Khi nhận được một yêu cầu, Proxy Server tạo ra một trạng thái, trạng thái này được duy trì cho tới khi kết thúc phiên giao dịch. Một số giao dịch, đặc biệt là giao dịch được tạo bởi “INVITE” có thể dài hơi lâu. Bởi vì máy chủ phải quản lý trạng thái trong suốt thời gian giao dịch nên thực thi của nó bị giới hạn. Khả năng liên kết các bản tin SIP trong các giao dịch làm cho Proxy Server có một số tính năng thú vị. Proxy Server có thể thực hiện việc nhận 1 bản tin và gửi 2 hay nhiều bản tin khác. Proxy Server có thể thực hiện việc truyền lại các bản tin bởi vì từ trạng thái của giao dịch nó biết

được là đã nhận được cùng bản tin đó chưa. Proxy Server có thể thực hiện các phương pháp phức tạp để tìm kiếm một người sử dụng. Hầu hết các SIP Proxy hiện nay là trạng thái vì cấu hình của chúng thường phức tạp.

c. Registrar Server:

Registrar Server là một thiết bị nhận các yêu cầu đăng ký và lưu trữ thông tin của người dùng.

d. Redirect Server:

Redirect Server là một thiết bị nhận bản tin yêu cầu và gửi lại bản tin trả lời chứa danh sách vị trí của một người dùng.

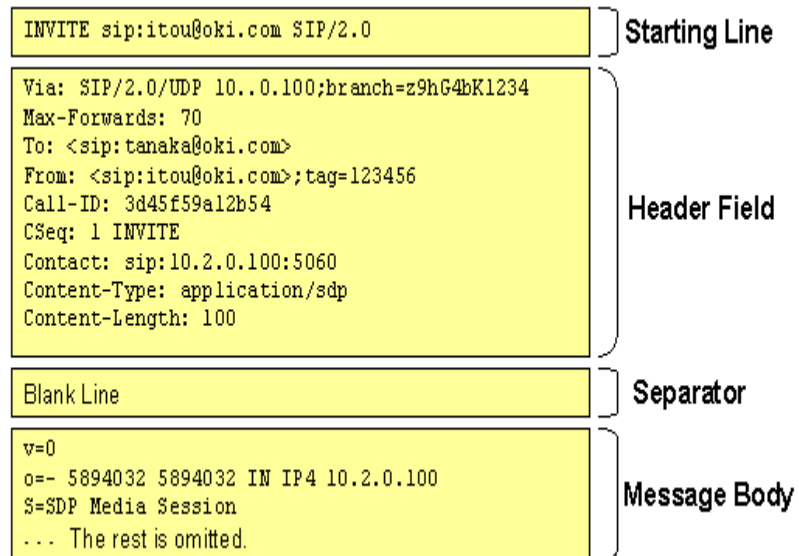
### 2.3.5. Các bản tin SIP

Truyền thông sử dụng SIP bao gồm một dãy các bản tin. Các bản tin có thể được truyền độc lập bởi mạng. Thông thường mỗi bản tin được truyền trong một dữ liệu UDP. Mỗi bản tin bao gồm phần dòng đầu tiên, phần đầu đề và phần thân bản tin. Phần dòng đầu tiên chỉ ra loại của bản tin. Có 2 loại bản tin SIP. Bản tin yêu cầu được sử dụng để khởi tạo một số hành động hoặc là báo cho người nhận yêu cầu nào đó. Bản tin trả lời để xác nhận một yêu cầu đã được nhận và được xử lý và chứa trạng thái của việc xử lý.

a. Các bản yêu cầu:

- **INVITE:** bản tin này dùng để thiết lập một phiên.

Ví dụ:



Hình 2.13: Bản tin INVITE

- Phần đầu dòng cho ta biết là bản tin INVITE. Sau đó là địa chỉ của bước truyền tiếp theo của văn bản tin.
- Phần header:
  - o Trường đầu đề “Via” được sử dụng để ghi lại đường đi của bản tin yêu cầu. Sau đó nó được sử dụng để định tuyến bản tin trả lời theo chính xác cùng một đường đi. Bản tin INVITE chỉ chứa một trường “Via” là được tạo ra bởi UA Client
  - o Trường đề “From” và “To” là địa chỉ của người gọi và người bị gọi
  - o Trường “Call-ID” để nhận dạng bản tin trong cùng một cuộc gọi. Các bản tin này có cùng một “Call-ID”. Trường “CSeq” dùng để chỉ thứ tự của các yêu cầu.
  - o Trường “Contact” chứa địa chỉ IP và cổng mà người gửi đang đợi các yêu cầu từ người bị gọi.
  - o Đầu đề của bản tin được phân cách với phần thân bởi một dòng trống.

- Phần thân của bản tin INVITE chứa mô tả kiểu dữ liệu được chấp nhận bởi người gửi và được mã hóa trong SDP.
- **ACK**: bản tin này công nhận bản tin trả lời cuối cùng cho INVITE. Thiết lập một phiên sử dụng bắt tay 3 bên. Việc này tốn thêm thời gian trước khi bị gọi chấp nhận hoặc từ chối cuộc gọi. Bị gọi sẽ gửi lại trả lời cuối cùng theo chu kỳ cho đến khi nhận được bản tin ACK.
- **BYE**: bản tin này dùng để kết thúc một phiên đa phương tiện. Bên nào muốn kết thúc phiên thì gửi BYE cho bên kia.
- **CANCEL**: dùng để hủy bỏ một phiên không được thiết lập đầy đủ.
- **REGISTER**: dùng để máy chủ registrar biết vị trí hiện tại của user. Thông tin về địa chỉ IP và cổng hiện tại của một user chứa trong bản tin.
- **REGISTER**: Máy chủ registrar lấy thông tin này và đưa vào cơ sở dữ liệu vị trí. Cơ sở dữ liệu này sau đó được sử dụng bởi các proxy server để định tuyến cuộc gọi tới user. Việc đăng ký là hạn chế thời gian và cần phải được làm tươi theo chu kỳ.

*b. Các bản tin phức tạp:*

Khi một UA hoặc proxy server nhận được một yêu cầu thì nó gửi lại một phức tạp. Tất cả yêu cầu phải được phức tạp trừ yêu cầu ACK. **Ví dụ:**

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP petr-pc.mydom.org:5060;branch=z9hG4bKfw19b;received=100.101.102.103
To: Ivan Ivanov <sip:ivan@domain.ru>;tag=a53e42
From: Petr <sip:petr@mydom.org>;tag=76341
Call-ID: j2qu348ek2328ws
CSeq: 1 INVITE
Contact: <sip:ivan@myhost.domain.ru>
Content-Type: application/sdp
Content-Length: 155

v=0
o=Ivan 2890844528 2890844528 IN IP4 myhost.domain.ru
s=Phone Call
c=IN IP4 200.201.202.203
t=0 0
m=audio 60000 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

Hình 2.14: Bản tin 200 OK

- Dòng đầu tiên của bản tin phúc đáp chứa phiên bản của giao thức, mã phúc đáp và lý do.
- Mã phúc đáp là một số nguyên từ 100 đến 699 và chỉ loại phúc đáp. Có 6 lớp của phúc đáp:
  - **1xx** là phúc đáp tạm thời. Một phúc đáp tạm thời là phúc đáp mà báo cho bên nhận biết là yêu cầu tương ứng đã nhận được nhưng kết quả của xử lý là không biết. Phúc đáp tạm thời được gửi chỉ khi việc xử lý không hoàn thành ngay lập tức. Người gửi phải dừng việc gửi lại yêu cầu trên. Thông thường các Proxy Server gửi phúc đáp với mã là 100 khi chúng bắt đầu xử lý một INVITE và các UA gửi phúc đáp với mã là 180 (Ring).
  - **2xx** là các phúc đáp xác thực cuối cùng. Phúc đáp cuối cùng đồng thời kết thúc giao dịch. Phúc đáp với mã từ 200 đến 299 là các phúc đáp xác thực có ý nghĩa là yêu cầu đã được xử lý thành công và được chấp nhận. Ví dụ phúc đáp 200 OK được gửi khi một user chấp nhận lời mời tới một phiên (yêu cầu INVITE).
  - **3xx** dùng để định hướng lại một người gọi. Một phúc đáp định hướng lại cho thông tin về vị trí mới của user hoặc một dịch vụ mà người gọi có thể sử dụng. Phúc đáp định hướng lại thường được gửi bởi Proxy Server. Khi một Proxy nhận một yêu cầu và không muốn hoặc không thể xử lý vì bất cứ lý do gì, nó sẽ gửi một phúc đáp định hướng lại tới người gọi và đặt vị trí khác vào trong phúc đáp mà người gọi có thể muốn thử lại. Nó có thể là vị trí của một Proxy khác hoặc là vị trí hiện tại của bị gọi (từ cơ sở dữ liệu vị trí của registrar). Chủ gọi sau đó có thể gửi lại yêu cầu tới vị trí mới. Các phúc đáp 3xx là cuối cùng.

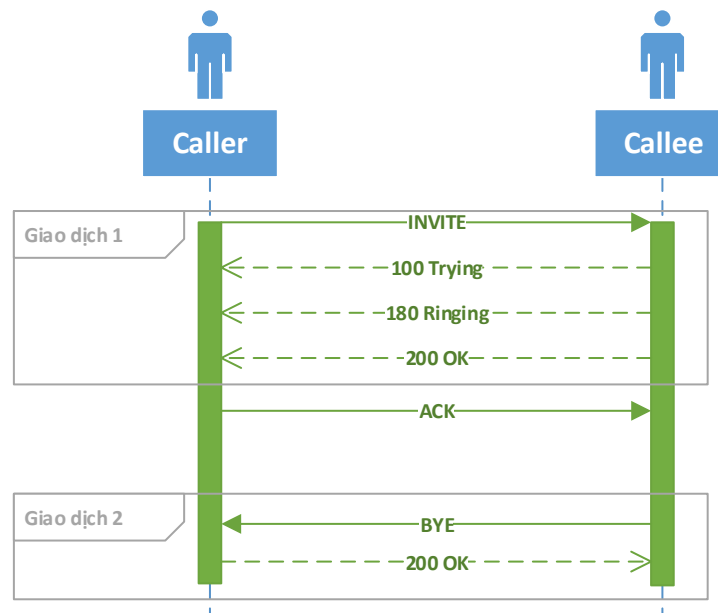
- **4xx** là các phúc đáp cuối cùng từ chối. Một phúc đáp 4xx có nghĩa rằng vấn đề ở phía chủ gọi. Yêu cầu không thể xử lý vì nó chứa cú pháp sai hoặc nó không thể được thực hiện ở Server.
- **5xx** là vấn đề nằm ở phía Server. Yêu cầu có thể hợp lệ nhưng server lỗi không thực hiện được.
- **6xx** là yêu cầu không thể được đáp ứng ở bất kỳ Server nào. Phúc đáp này thường được gửi bởi server mà có thông tin cuối cùng về một user cụ thể. Các UA thường gửi bản tin 603 từ chối trả lời khi user đó không muốn tham dự vào phiên.

#### **2.3.6. Các giao dịch SIP**

SIP được gọi là giao thức giao dịch. Vì các bản tin SIP được gửi một cách độc lập qua mạng và được sắp xếp vào các giao dịch bởi các UA và các Proxy Server.

Một giao dịch là một chuỗi các bản tin SIP được trao đổi giữa các phần tử mạng SIP. Một giao dịch bao gồm một yêu cầu và tất cả phúc đáp cho yêu cầu đó. Đó bao gồm không hoặc nhiều phúc đáp tạm thời và một hoặc nhiều bản tin cuối cùng.

Nếu một giao dịch được khởi tạo bởi một yêu cầu INVITE thì giao dịch đó cũng bao gồm ACK nhưng chỉ khi phúc đáp cuối cùng không phải là phúc đáp 2xx. Nếu phúc đáp cuối cùng là 2xx thì ACK không phải là một phần của giao dịch.



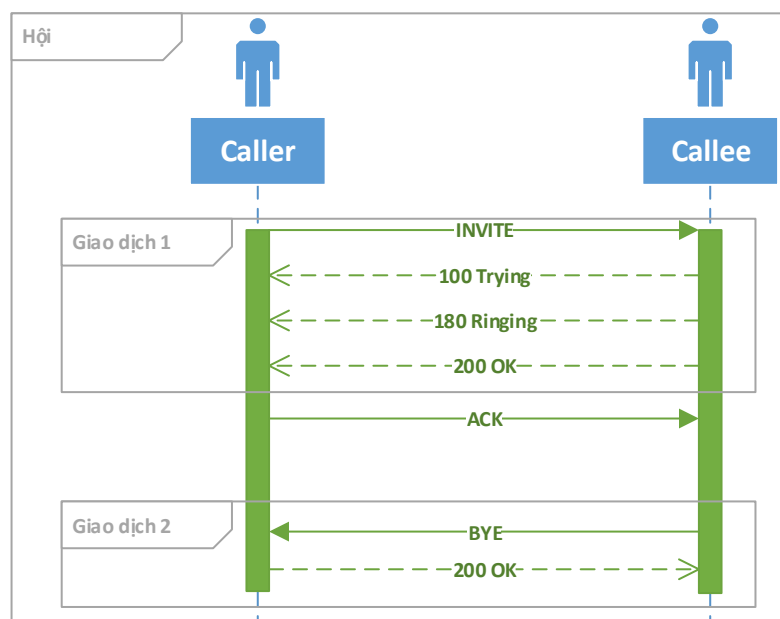
Hình 2.15: Phiên giao dịch SIP

### 2.3.7. Các hội thoại SIP

Một hội thoại tương ứng với một quan hệ SIP đồng đẳng (peer-to-peer) giữa hai UA. Các hội thoại làm cho việc sắp xếp thứ tự và định tuyến các bản tin giữa các điểm cuối SIP được thuận tiện hơn.

Các hội thoại được định danh sử dụng các trường Call-ID, From và To. Các bản tin có ba trường này giống nhau thì cùng trong một hội thoại. Một hội thoại là một dãy các giao dịch. Hình dưới đây sẽ chỉ ra các bản tin trong hội thoại.





Hình 2.16: Hội thoại trong SIP

Một số bản tin thiết lập một hội thoại, một số thì không. Điều này cho phép biểu diễn rõ ràng mối quan hệ của các bản tin đồng thời để gửi các bản tin mà không liên quan tới các bản tin khác ngoài hội thoại. Điều này thực hiện dễ dàng bởi vì UA không phải giữ trạng thái hội thoại.

Ví dụ bản tin INVITE thiết lập một hội thoại thì nó sẽ được kèm theo sau yêu cầu BYE. Nhưng nếu một UA gửi một yêu cầu MESSAGE thì yêu cầu này không thiết lập hội thoại. Các bản tin sau sẽ được gửi độc lập với bản tin trước.

a. Định tuyến hội thoại:

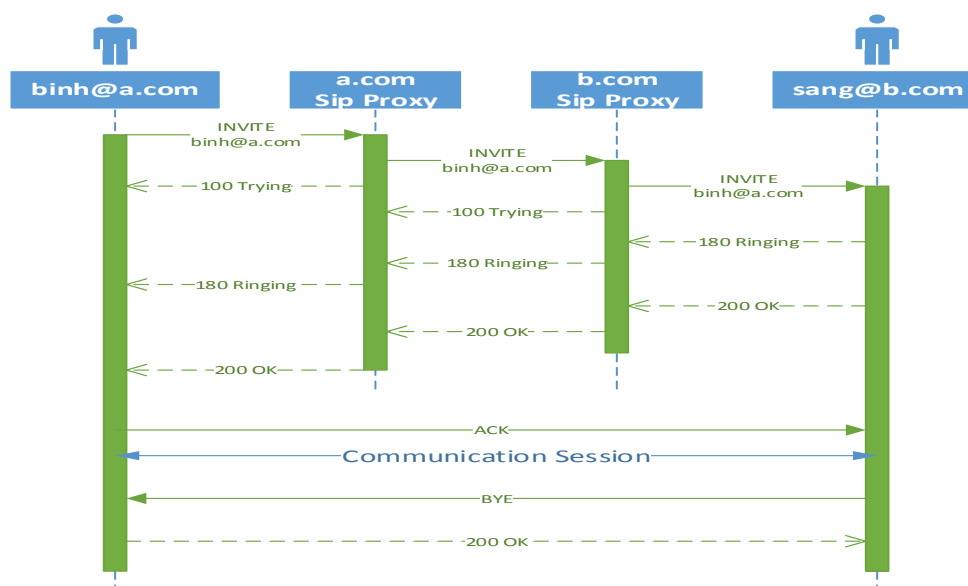
Chúng ta biết rằng các hội thoại cũng được sử dụng để định tuyến các bản tin giữa các UA.

Giả sử rằng user [sip:binh@a.com](mailto:sip:binh@a.com) muốn nói chuyện với user [sip:sang@b.com](mailto:sip:sang@b.com). Chủ gọi biết địa chỉ SIP gọi nhưng địa chỉ này không nói bất cứ điều gì về vị trí hiện tại của user. Vì vậy yêu cầu INVITE được gửi tới một proxy server.

Yêu cầu sau đó gửi từ proxy tới khi nào đến một proxy mà biết vị trí hiện tại của người nghe. Quy trình này được gọi là định tuyến. Khi yêu cầu đã đến được người nghe, người nghe sẽ gửi lại phúc đáp cho người gọi. Người nghe đồng thời cũng đưa đầu đề “Contact” vào trong phúc đáp và sẽ chứa địa chỉ hiện tại của user. Yêu cầu gốc cũng chứa trường đầu đề “Contact”. Vì thế hai UA biết vị trí hiện tại của nhau.

Bởi vì các UA biết vị trí của nhau nên không cần thiết phải gửi yêu cầu qua các Proxy nữa, chúng có thể được gửi trực tiếp từ UA đến UA. Đó chính là hội thoại làm cho định tuyến thuận tiện.

Các bản tin trong cùng một hội thoại sau đó sẽ được gửi trực tiếp từ UA đến UA. Điều này giúp cải thiện hiệu năng bởi vì các Proxy không xem tất cả các bản tin trong cùng một hội thoại, chúng thường được sử dụng để định tuyến yêu cầu đầu tiên mà thiết lập hội thoại. Các bản tin trực tiếp đồng thời cũng được truyền với độ trễ nhỏ hơn nhiều bởi vì một Proxy thường hiện các phép toán logic định tuyến phức tạp.



Hình 2.17: Phiên giao dịch hai SIP Proxy

*b. Những dạng hội thoại:*

Chúng ta biết rằng sự nhận dạng hội thoại bao gồm 3 phần: Call-ID, From tag và To tag.

Call-ID được gọi là phần nhận dạng cuộc gọi. Call-ID phải là một chuỗi ký tự duy nhất nhận dạng cuộc gọi. Một cuộc gọi bao gồm một hay nhiều hội thoại. Đa số UA có thể phức đáp cho một yêu cầu khi một Proxy phân nhánh yêu cầu đó. Mỗi UA gửi một phức đáp 2xx thiết lập một hội thoại riêng rẽ với chủ gọi. Tất cả cá hội thoại này là một phần của cùng một cuộc gọi và có cùng tham số “Call-ID”.

From tag: được tạo ra bởi chủ cuộc gọi và nhận dạng duy nhất hội thoại trong UA của người gọi.

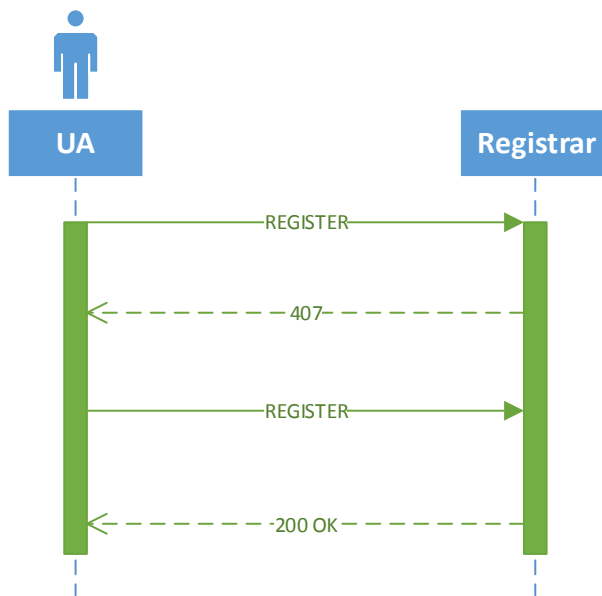
To tag: được tạo ra bởi người nghe và nhận dạng duy nhất hội trong UA của người nghe.

Sự nhận dạng hội thoại phân cấp này là cần thiết bởi vì một lời mời cuộc gọi có thể tạo ra vài hội thoại và người gọi phải có khả năng phân biệt chúng.

### **2.3.8 Các giao thức SIP**

*a. Đăng ký:*

Người sử dụng phải đăng ký với một registrar để những người sử dụng khác có thể liên lạc đến được. Một sự đăng ký bao gồm một bản tin “REGISTER” và một phức đáp “200 OK” từ registrar nếu sự đăng ký là thành công. Sự đăng ký thường phải xác thực do đó một phức đáp “407” có thể được gửi nếu người sử dụng không cung cấp sự tin cậy hợp lệ.



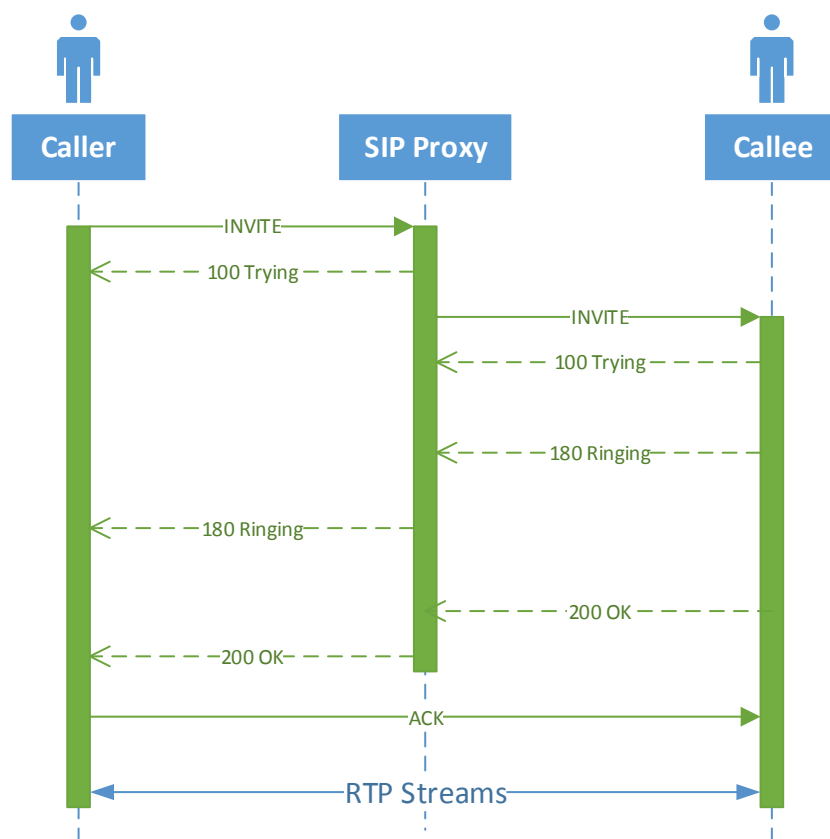
Hình 2.18: Phiên giao dịch Registrar

*b. Khởi tạo phiên:*

Một sự khởi tạo phiên bao gồm một yêu cầu “INVITE” mà thường là gửi tới một Proxy. Proxy gửi ngay một phúc đáp “100 Trying” để ngừng việc gửi lại và chuyển yêu cầu sau này.

Tất cả phúc đáp tạm thời được tạo bởi người nghe được gửi lại cho người gọi. Khi người nghe đổ chuông nó gửi phúc đáp cho người gọi bản tin “180 Ringing”.

Khi người nghe nhắc máy nó gửi bản tin “200 OK” và được gửi lại cho đến khi nhận được một “ACK” từ người gọi. Phiên được thiết lập từ thời điểm này.



Hình 2.19: Phiên giao dịch SIP Proxy

c. Kết thúc phiên:

Kết thúc phiên được thực hiện bằng cách gửi bản tin “BYE”. Bản tin “BYE” được gửi trực tiếp từ một callee đến caller trừ khi một Proxy trên đường đi của yêu cầu “INVITE” chỉ ra rằng nó phải đi theo bằng cách sử dụng định tuyến bản ghi.

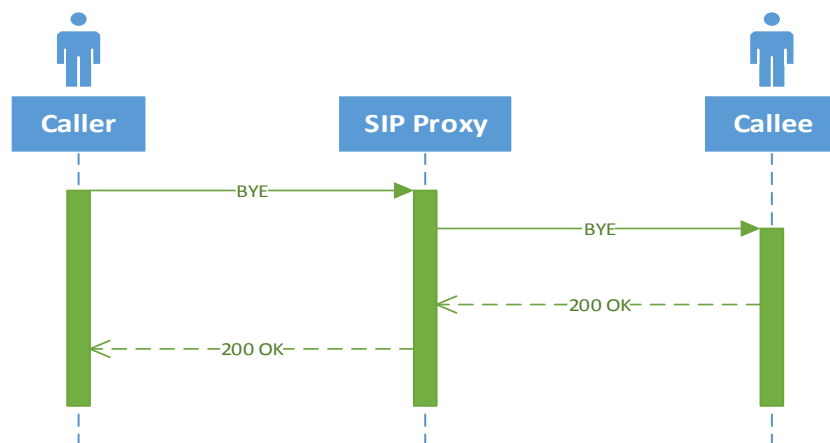
Bên muốn kết thúc phải gửi một yêu cầu “BYE” tới bên kia. Bên kia sẽ gửi lại một phúc đáp “200 OK” để xác nhận yêu cầu “BYE” và phiên chấm dứt.

d. Định tuyến bản ghi:

Tất cả yêu cầu trong một hội thoại mặc định được gửi trực tiếp từ một caller đến callee. Chỉ những yêu cầu ngoài một hội thoại là đi qua cá proxy. Cách này làm cho mạng SIP mềm dẻo hơn bởi vì chỉ có một số nhỏ bản tin đến proxy.

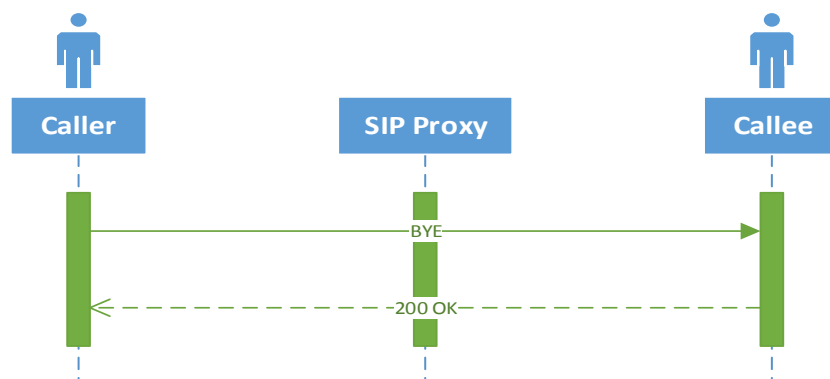
Có những tình huống mà Proxy cần lưu lại đường đi của tất cả bản tin. Ví dụ Proxy điều khiển hộp NAT hoặc Proxy thực hiện tính cước cần phải lưu lại đường đi của yêu cầu “BYE”.

Trường hợp có định tuyến bản ghi:



Hình 2.20: Phiên kết thúc giao dịch với SIP Proxy

Trường hợp không có định tuyến bản ghi:



Hình 2.21: Phiên kết thúc giao dịch SIP

## 2.4. Thành phần Media

### 2.4.1 Giao thức RTSP

#### a. Giới thiệu:

RTSP (Real Time Streaming Protocol) là một giao thức mức ứng dụng, thường được dùng để điều khiển việc truyền dữ liệu với các đặc tính thời gian thực. Nó thiết lập và điều khiển một hay nhiều stream của continuous media (như audio và video) được đồng bộ theo thời gian. Thực tế, bản thân RTSP không truyền continuous media mà nó hoạt động như một “network remote control” (điều khiển mạng từ xa) cho các dịch vụ đa phương tiện.

Không có khái niệm RTSP connection (kết nối RTSP); thay vào đó sever sẽ duy trì một session được đánh dấu bởi một định danh. Một RTSP session không thể bị ràng buộc với một kết nối ở mức giao vận như kết nối TCP. Trong suốt một RTSP session, một RTSP client có thể mở và đóng nhiều kết nối truyền thông tin cậy như TCP hoặc không tin cậy như UDP tới sever để đưa ra các RTSP request.

RTSP được xây dựng trên đỉnh của các giao thức chuẩn Internet như: UDP, TCP/IP, RTP, RTCP, SCP và IP multicast. Giao thức này cung cấp, phương tiện cho việc chọn kênh truyền (như UDP, TCP ...) và phương tiện cho việc chọn cơ chế truyền như dựa trên RTP ...). Stream được điều khiển bởi RTSP có thể được sử dụng RTP nhưng hoạt động của RTSP không phụ thuộc vào cơ chế được sử dụng để truyền continuous media.

RTSP hỗ trợ các thao tác sau:

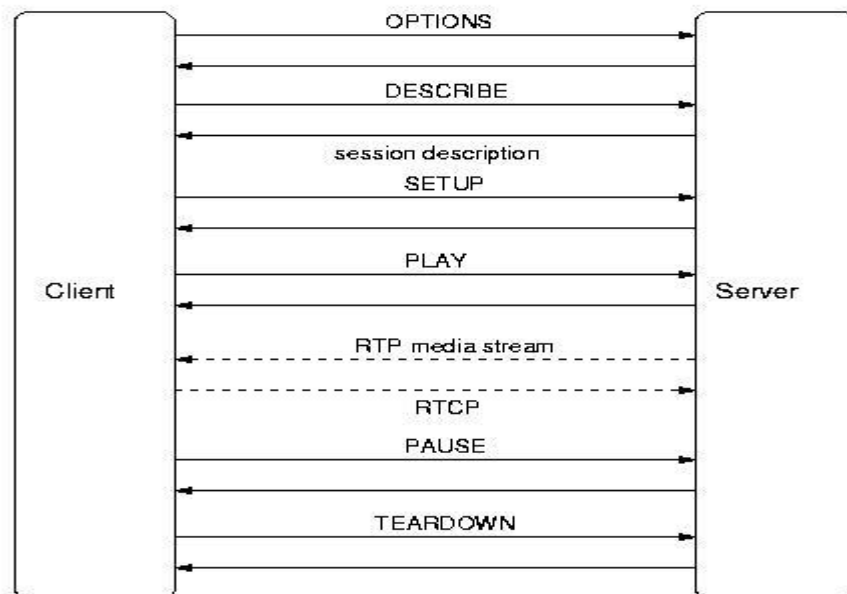
- Lấy media từ media sever.
- Mời một media sever vào conference.
- Thêm media vào một presentation đã có sẵn

RTSP được phát triển bởi RealNetworks, Netscape và Columbia University, với bản dữ thảo đầu tiên từ IETF năm 1996. Nó được chuẩn hóa bởi Multiparty Multimedia Session Control Working Group (MMUSIC WG) thuộc Internet Engineering Task Force (IETF) và xuất bản trong RFC 2326 năm 1998. RTSP 2.0 hiện đang được phát triển như một sự thay thế cho RTSP 1.0.

*b. Các thông điệp của giao thức:*

RTSP tương tự như HTTP, tuy nhiên RTSP có thể kiểm soát được trình tự, nó hữu ích trong việc kiểm soát multimedia playback. HTTP là giao thức phi trạng thái trong khi RTSP là giao thức trạng thái đầy đủ. Giống như HTTP, RTSP sử dụng TCP để duy trì một kết nối end-to-end. RTSP hầu hết các thông điệp được gửi từ client tới server, một số lệnh server tới client. Định nghĩa transport layer với port number là 554 cho cả 2 TCP và UDP.

Những yêu cầu cơ bản của RTSP :



Hình 2.22: RTSP kết nối client và server



- **OPTIONS** : Yêu cầu OPTIONS được client gửi server và server sẽ phản hồi chấp nhận yêu cầu.

```
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 2.23: Yêu cầu OPTIONS

- **DESCRIBE** : Client gửi yêu cầu DESCRIBE bao gồm RTSP URL (rtsp://...) tới server, và server gửi lại các dữ liệu xử lý cần xử lý tới client. Thông tin phản hồi là định dạng Session Description Protocol (SDP)

```
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 2.24: Yêu cầu DESCRIBE

- **SETUP:** Client sẽ gửi yêu cầu SETUP để thiết lập một đường truyền thông duy nhất. Yêu cầu này phải trước yêu cầu PLAY. Yêu cầu sẽ bao gồm media stream URL và miêu tả transport. Nó còn chứa thông tin port nhận dữ liệu RTP (audio và video) và RTCP. Sau đó server sẽ phản hồi về client, xác nhận các yêu cầu của client và phản hồi cổng máy chủ chọn.

```
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 2.25: Yêu cầu SETUP

- **PLAY:** Khi server nhận yêu cầu PLAY từ client, lúc đó server sẽ mở một hoặc tất cả media stream. Các yêu cầu có thể được stack bởi gửi nhiều yêu cầu PLAY. Các URL có thể tổng hợp hoặc một URL media stream. Một phạm vi trong đoạn media có thể được xác định. Nếu không xác định thì stream chơi từ đầu tới cuối media.

```
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 2.26: Yêu cầu PLAY

- PAUSE: Khi client gửi yêu cầu PAUSE thì việc stream media dừng lại. Để bắt đầu stream lại client phải gửi yêu cầu PLAY. Một yêu cầu PAUSE có chứa thời gian xác định, nếu không có thì thời gian dừng sẽ là vô hạn.

```
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 2.27: Yêu cầu PAUSE

- TEARDOWN: là yêu cầu client gửi để kết thúc session và dừng tất cả các stream media. Qua đó sẽ giải phóng tất cả dữ liệu liên quan đến phiên server.

```
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 2.28: Yêu cầu TEARDOWN

## 2.4.2. Giao thức RTP

### a. Giới thiệu:

Phương thức thông thường để truyền tải dữ liệu dạng audio hay video cùng các dữ liệu đính kèm là khung truyền RTP. RTP nhằm mục tiêu cung cấp các dịch vụ truyền tải hình ảnh thời gian thực như audio/video thông qua mạng IP.

Các dịch vụ đó bao gồm khôi phục dữ liệu sau khoảng thời gian xác định (timing recovery), tìm và sửa lỗi (loss detection and correction), định dạng khung truyền và nguồn (payload and source identification), tiếp nhận phản hồi về chất lượng dịch vụ (reception quality feedback), đồng bộ dữ liệu (synchronization) và

quản lý thành viên ( membership management). RTP được thiết kế để dùng trong trao đổi quảng bá (multicast conferences) dùng cơ chế lightweight sessions.

Nó tỏ ra hữu ích trong hàng loạt các ứng dụng: hội nghị truyền hình dùng cơ chế H.323, webcasting, truyền hình, hệ thống thoại có dây và không dây. Bằng việc dùng giao thức này mỗi phiên truyền tải được gửi đến hàng nghìn người.

RTP được coi là chuẩn đóng gói để truyền tải dữ liệu dạng audio và video qua mạng Internet. Nó được phát triển bởi Audio-Video Transport Working Group thuộc IETF (Internet Engineering Task Force) và được công bố lần đầu vào năm 1996 với tên gọi là RFC 1889. Sau này là RFC 3550 năm 2003.

RTP thường được dùng kết hợp với RTCP. Trong khi RTP truyền dữ liệu. RTCP được dùng để điều khiển trạng thái truyền và chất lượng dịch vụ QoS. Khi được dùng kết hợp RTP thường được sắp xếp nhận trên cổng chặn, RTCP trên cổng lẻ.

RTP ban đầu được tạo ra là giao thức multicast nhưng nó vẫn chấp nhận trong nhiều ứng dụng unicast. Trong truyền thông dạng host-to-host, RTP và RTCP thường dùng UDP qua các giao thức lớp vận chuyển.

Số hiệu cổng: RTP không thao tác trên các cổng mặc định như TCP và UDP. Tuy nhiên nó thường dùng các cổng có số hiệu chẵn, có giá trị từ 16384-32767, dải địa chỉ này đang dần được mở rộng. Việc này gây khó khăn trong trường hợp có firewall và NATs. Để giải quyết vấn đề này, nó cần khởi tạo STUN server.

*b. Các thành phần của giao thức:*

*i. Sender:*

Bên gửi có nhiệm vụ lưu trữ và biến đổi dữ liệu dạng nghe nhìn để truyền tải. RTP tham gia vào quá trình sửa lỗi. Và điều phối tránh tắc nghẽn bằng cách thêm vào dòng dữ liệu truyền tải thông tin phản hồi của bên nhận.

Các khung truyền được nạp vào các gói RTP và sẵn sàng gửi. Nếu khung truyền quá lớn, nó sẽ được chia nhỏ thành nhiều gói RTP. Ngược lại, nó sẽ

được tập hợp lại thành một gói RTP. Tùy thuộc vào lược đồ sửa lỗi được dùng, 1 channel coder được dùng để tạo ra các gói sửa lỗi hoặc các gói lặp lại trước khi truyền.

Sau khi các gói RTP được gửi bộ đệm dữ liệu tương ứng với nó được giải phóng. Tuy vậy bên gửi phải lưu trữ dữ liệu trong một khoảng thời gian sau khi gói tin tương ứng được gửi.

Bên gửi có trách nhiệm tạo các bản báo cáo trạng thái định kỳ về dòng dữ liệu để đồng bộ dữ liệu. Nó cũng nhận thông tin phản hồi từ các bên tham gia và dùng thông tin đó để tham gia vào quá trình truyền.

ii. Receiver:

Bên nhận có trách nhiệm thu nhận các gói RTP từ mạng, sửa lỗi, recovering the timing, cắt bớt dữ liệu và hiển thị kết quả cho người dùng. Nó gửi phản hồi chất lượng cho bên gửi. Cơ sở dữ liệu của các bên được duy trì trong một phiên truyền.

c. Đặc điểm của giao thức:

RTP không có sẵn các cơ chế để đảm bảo việc truyền theo thời gian hay các kỹ thuật về QoS mà dựa vào các dịch vụ ở lớp dưới để thực hiện những khả năng này. RTP không đảm bảo an toàn hay hay thứ tự các packet khi truyền, số thứ tự trong RTP packet cho phép bên nhận sắp xếp lại các packet thứ tự khi truyền của người gửi. Ngoài ra số thứ tự cũng có thể được tận dụng để xác định vị trí thích hợp của một packet, ví dụ trong các việc giải mã video, mà không cần phải giải mã các packet theo thứ tự.

Các gói tin truyền trên mạng Internet có độ trễ (jitter) không đoán được. Nhưng các ứng dụng đa phương tiện yêu cầu một thời gian thích hợp khi truyền các gói dữ liệu và phát lại. RTP cung cấp các cơ chế bảo đảm thời gian, số thứ tự và các cơ chế khác liên quan đến thời gian.

Bằng các cơ chế này RTP cung cấp sự truyền tải dữ liệu thời gian thực giữa các đầu cuối qua mạng. Tem thời gian (Time-stamping) là thành phần quan trọng nhất trong các ứng dụng thời gian thực. Người gửi thiết lập các “tem thời gian” tang dần theo thời gian với mọi gói. Sau khi nhận được gói dữ liệu, bên thu sử dụng các “tem thời gian” này nhằm khôi phục thời gian gốc để chạy các ứng dụng này với tốc độ thích hợp. Ngoài ra nó được sử dụng để đồng bộ các dòng dữ liệu khác nhau (chẳng hạn như giữa hình và tiếng). Tuy nhiên RTP không thực hiện đồng bộ mà các ứng dụng phía trên sẽ thực hiện sự đồng bộ này. Bộ phận dạng tải xác định kiểu định dạng của tải tin cũng như các phương cách mã hóa nén. Từ các bộ phận định dạng này, các ứng dụng phía thu biết cách phân tích và chạy dữ liệu tải tin. Tại một thời điểm bất kỳ trong quá trình truyền tin, các bộ phát RTP chỉ có thể gửi một dạng của tải tin dạng tải tin có thể thay đổi trong thời gian truyền (thay đổi để thích ứng với sự tắc nghẽn của mạng). Một chức năng khác mà RTP có là xác định nguồn. Nó cho phép các ứng dụng thu biết được dữ liệu từ đâu. Ví dụ thoại hội nghị, từ thông tin nhân dạng nguồn một người sử dụng có thể biết được ai đang nói.

	UDP Header	RTP Header	RTP Payload
--	------------	------------	-------------

Bảng 2.2: Mã hóa gói tin RTP trong IP

Các cơ chế hoạt động nêu trên được thực hiện thông qua mã đầu của RTP. Các mã hóa gói tin RTP trong các gói tin IP được mô tả trên hình RTP nằm ở phía trên UDP, sử dụng các chức năng ghép kênh và kiểm tra của UDP. UDP và TCP là hai giao thức sử dụng chủ yếu trên Internet. TCP cung cấp kết nối định hướng và các dòng thông tin với độ tin cậy thấp với hai trạm chủ. Sở dĩ UDP được sử dụng làm thủ tục truyền tải cho RTP là bởi hai lý do:

- Thứ nhất, RTP được thiết kế chủ yếu cho việc truyền tin đa đối tượng, các kết nối có định hướng, có báo nhận không đáp ứng tốt điều này.

- Thứ hai, đối với dữ liệu thời gian thực, độ tin cậy không quan trọng bằng truyền đúng theo thời gian. Hơn nữa sự tin cậy trong TCP là do cơ chế báo phát lại không thích hợp cho RTP. Ví dụ khi mạng bị tắc nghẽn một số gói dữ liệu sẽ bị mất, chất lượng dịch vụ thấp nhưng vẫn chấp nhận được. Nếu thực hiện việc phát lại sẽ gây ra độ trễ lớn, chất lượng thấp gây ra sự tắc nghẽn của mạng.

d. Kiến trúc gói dữ liệu:

bit offset	0 - 1	2	3		4 - 7	8	9 - 15	16 – 31
0	Ver	P	X		CC	M	PT	Sequence Number
32	Timestamp							
64	SSRC identifiers							
96	CSRC identifiers (optional) ...							
96 + (CCx32)	Extension header (optional)							
96 + (CCx32) + (Xx((EHL+1) x32))	Data							

Bảng 2.3: Kiến trúc gói dữ liệu

RTP header có kích thước tối thiểu là 12 byte:

- Ver (2 bit): Cho biết version của giao thức. Hiện tại đang là version 2.
- P (Padding 1 bit): Được dùng để cho biết có byte mở rộng thêm vào cuối gói RTP
- X (Extension 1 bit): Cho biết sự có mặt của Extension header và payload data.
- CC (CSRC Count 4 bit): Tổng số CSRS đã được định nghĩa thêm vào header.

- M (Marker 1 bit): Được dùng trong các mức ứng dụng và được định nghĩa bởi 1 profile. Nếu nó được khởi tạo dữ liệu có liên quan đặc biệt đến ứng dụng
- PT (Payload Type): Cho biết khuôn dạng payload và xác định giới hạn của nó trong ứng dụng
- Sequence Number: số thứ tự tăng dần trong mỗi ứng dụng RTP đã được gửi và được dùng để biên nhận sửa lỗi và lưu trữ lại số thứ tự gói tin.
- Timestamp: phản ánh sampling instant của dữ liệu trong gói RTP. Nó tăng dần và tuyến tính để đồng bộ dữ liệu và tính jitter. Số liệu đó phải đủ để đồng bộ chính xác và tính được jitter, thông thường 1 tick trên 1 khung truyền video là không đủ để tính. Tần số của đồng hồ phụ thuộc vào khung dữ liệu. Giá trị mặc định khi khởi tạo là 0.
- SSRC (32 bit): Đồng bộ nguồn dữ liệu xác định duy nhất một dòng dữ liệu vào.
- CSRC: Đếm số các nguồn vào từ nhiều nguồn khác nhau.
- Extension header: 32 bit đầu bao gồm 16 bit định danh chương trình và 16 bit xác định độ dài phần mở rộng (EHL – Extension Header Length), thêm 32 bit mở rộng.

### 2.4.3. Giao thức RTCP

#### a. Giới thiệu:

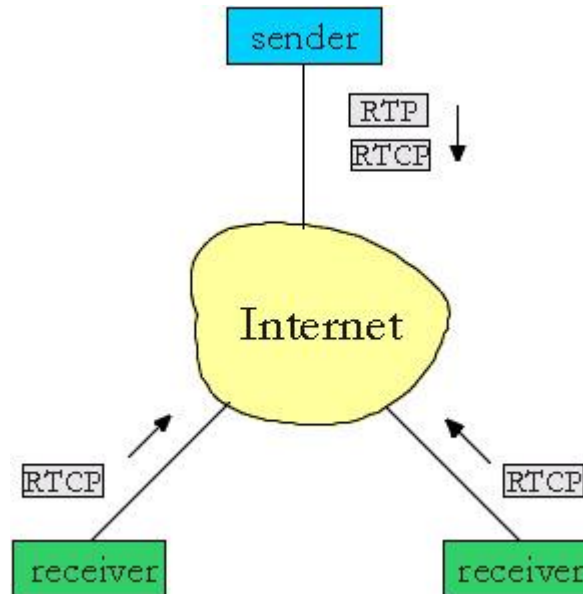
RTCP (Real-time Transport Control Protocol) được định nghĩa trong RFC 3550, là một giao thức ứng dụng trên mạng lưới đa phương tiện, RTCP được sử dụng hỗ trợ cho RTP cung cấp thông tin phản hồi về chất lượng truyền dữ liệu.

#### b. Chức năng và hoạt động của RTCP:

Giao thức này dùng để điều khiển gói mang tin trong phiên truyền của mỗi thành viên, được phân phối theo cùng cơ chế như các gói mang tin. Các giao



thức lớp dưới phải đảm bảo các gói mang dữ liệu RTP và các gói mang thông tin điều khiển RTCP được truyền trên 2 cổng UDP khác nhau. Thường thì gói tin mang thông tin điều khiển đi qua cổng lẻ, các gói dữ liệu đi qua cổng chẵn.



Hình 2.29: Hoạt động của RTCP

Các chức năng mà giao thức RTCP cung cấp:

- Giám sát chất lượng và điều khiển tắt nghẽn – Đây là chức năng cơ bản của RTCP. Nó cung cấp thông tin phản hồi tới một ứng dụng về chất lượng phân phối dữ liệu. Thông tin điều khiển này rất hữu ích cho các bộ phát, bộ thu và giám sát. Bộ phát có thể điều chỉnh cách thức truyền dữ liệu dựa trên các thông báo phản hồi của bộ thu. Bộ thu có thể xác định được tắt nghẽn là cục bộ, từng phần hay toàn bộ. Người quản lý mạng có thể đánh giá được hiệu suất mạng.
- Xác định nguồn – Trong các gói RTP, các nguồn được xác định bởi các số ngẫu nhiên có độ dài 32 bit. Các số này không thuận tiện đối với người sử dụng. RTCP cung cấp thông tin nhận dạng nguồn cụ thể hơn ở dạng

văn bản. Nó có thể bao gồm tên người sử dụng, số điện thoại, địa chỉ e-mail và các thông tin khác.

- Đồng bộ môi trường – Các thông báo của bộ phát RTCP chứa thông tin để xác định thời gian RTP tương ứng. Chúng có thể được sử dụng để đồng bộ giữa âm thanh với hình ảnh.
- Điều chỉnh thông tin điều khiển – Các gói RTCP được gửi theo chu kỳ giữ những người tham dự. Khi số lượng người tham dự tăng lên, cần phải cân bằng giữa việc nhận thông tin điều khiển mới nhất và hạn chế lưu lượng điều khiển. Để hỗ trợ một nhóm người sử dụng lớn, RTCP phải giảm lưu lượng điều khiển rất lớn đến từ các tài nguyên khác của mạng. RTP chỉ cho phép tối đa 5% lưu lượng cho điều khiển toàn bộ lưu lượng của phiên làm việc. Điều này được thực hiện bằng các điều chỉnh tốc độ phát của RTCP theo số lượng người tham dự

c. Các loại gói tin:

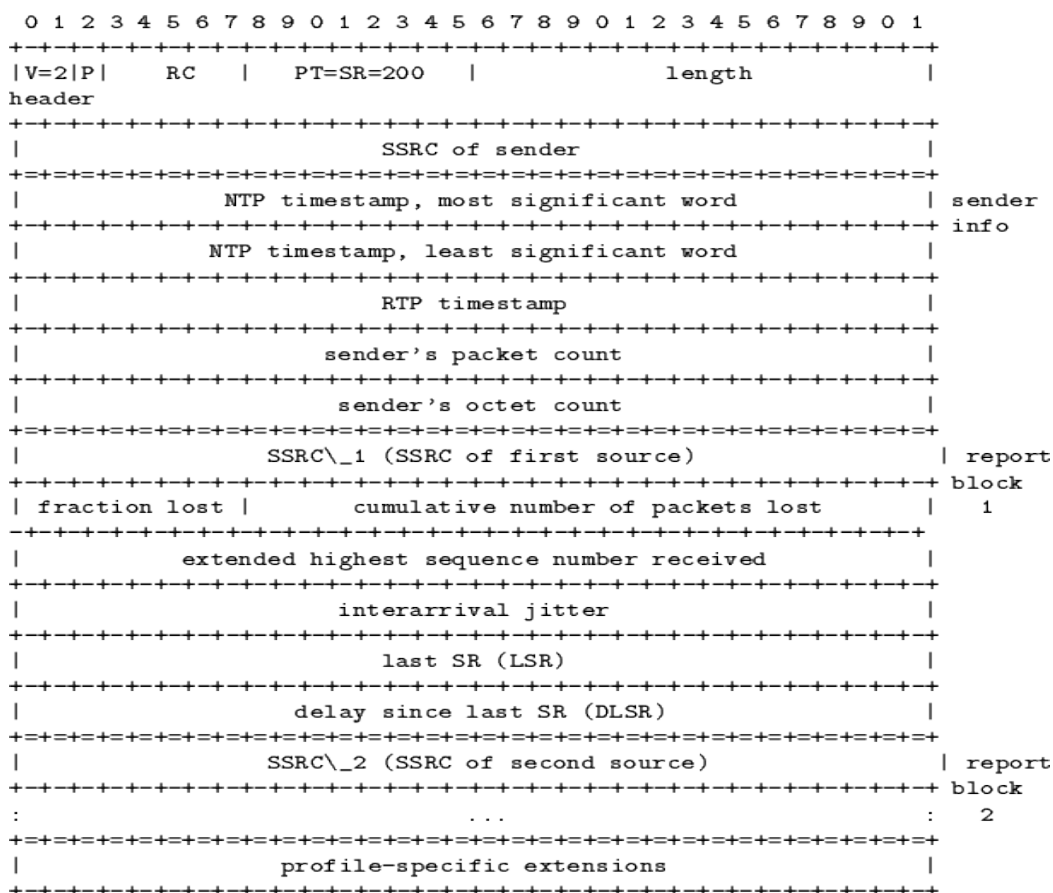
Chúng ta sẽ đề cập đến nhiều loại gói tin RTCP tương ứng với những loại thông tin điều khiển khác nhau:

- SR (Sender report): Bản tin phía gửi, dùng để thông tin về trạng thái truyền và nhận của phía gửi tin.
- RR (Receiver report): Bản tin người nhận, dùng để thông tin về trạng thái nhận của phía nhận tin.
- SDES (Source description items): Thông tin mô tả các về nguồn gửi, bao gồm cả CNAME.
- BYE: Dùng khi thành viên nào đó thoát khỏi hội thảo.
- APP (Application specific functions): Xác định các chức năng phụ thuộc vào từng ứng dụng cụ thể.

Mỗi gói tin RTCP được bắt đầu với phần tiêu đề cố định tương tự như của gói tin RTP, tiếp theo là các thành phần cấu trúc, chúng có thể có chiều dài biến đổi tùy thuộc vào loại của gói tin, nhưng phải giới hạn trong 32-bit.

Sự chuẩn hóa và các trường có chiều dài cố định trong mỗi gói RTCP làm cho chúng có thể được lưu trong ngăn xếp. Nhiều gói RTCP có thể được kết nối với nhau thành gói ghép (compound packet), mà không cần sử dụng dấu phân cách giữa các gói RTCP, khi đưa chúng xuống đóng gói ở các lớp dưới. Ví dụ như trong các gói UDP.

Mỗi gói RTCP trong gói ghép có thể được xử lý một cách độc lập, không cần dựa vào thứ tự hoặc sự kết hợp giữa các gói.



Hình 2.30: Mẫu gói tin SR của RTCP

### CHƯƠNG 3. TRIỂN KHAI VÀ THỬ NGHIỆM HỆ THỐNG

Vì hạn chế về khả năng nghiên cứu quy mô lớn và công nghệ nghiên cứu chương trình của đề tài không có điều kiện nên chương trình không thể thử nghiệm quy mô lớn trên mạng Internet. Vì vậy chương trình của đề tài chú trọng xây dựng một mô hình và hướng đi mới là xây dựng và thử nghiệm chương trình thoại P2P và ứng dụng trên môi trường mạng LAN Wifi cục bộ.

#### 3.1 Cấu hình hệ thống triển khai và kiểm thử

a. Cấu hình thử nghiệm:

- Hai thiết bị Android API lớn hơn 4.0 có cài đặt chương trình của đề tài.
- Video Code: H263 và H264.
- Audio Codec: AAC và AMR.
- Thiết bị Wifi: TPLink 300Mb.
- Một máy tính kết nối vào mạng đóng vai trò Bootstrap Node.

b. Mục tiêu của thử nghiệm:

- Đánh giá hiệu quả giữa hai codec video là H264 và H323 trong vấn đề chất lượng truyền dẫn của hình ảnh và giữa hai codec audio là AAC và AMR trong vấn đề chất lượng truyền dẫn âm thanh.
- Đánh giá một số điểm yếu trong thiết kế sẽ được giải quyết trong giai đoạn thử nghiệm.
- Phân tích chất lượng video và audio tại các thời điểm khác nhau trong quá trình truyền dữ liệu giữa các codec.

c. Thử nghiệm hệ thống:

- Trong hoàn cảnh này, thử nghiệm sẽ đưa ra đánh giá về chất lượng hình ảnh của mỗi codec để kiểm tra chất lượng hình ảnh truyền đi với điều kiện đường truyền là tốt nhất. Điều này nhằm đánh giá thử chương trình xây dựng có tiềm năng truyền dẫn video trên mạng Internet khi xây dựng chương trình lớn hay không?

Đồng thời nó cũng đánh giá với tốc độ mạng như thế nào thì sử dụng codec nào cho phù hợp.

- Độ trễ khi một node tham gia vào mạng: gần như không có.
- Tốc độ cập nhật khi một node tham gia vào mạng và rời mạng: gần như không có.
- Phát sinh cuộc gọi: delay 0.5s.
- Tín hiệu phản hồi và chấp nhận bởi thực thể thực hiện cuộc gọi: 0.5s.

Resolution	Tỉ lệ mất gói tin		Độ trễ		Băng thông nhận tối đa	
320x240	0%	0%	Không có	Không có	2500kb	2500kb
640x480	1%	2%	0.5s	1s	2300kb	2300kb
1280x720	2%	3%	1s	1.5s	2100kb	2000kb



H.264



H.263

Bảng 3.1: Bảng so sánh thử nghiệm H.264 và H.263

## **CHƯƠNG 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

### **4.1. Kết luận**

Tác giả đã thực hiện trong luận văn:

- Tìm hiểu và nghiên cứu mạng đồng đẳng Peer to Peer, một mô hình mạng mới. Thực thi một mô hình mạng của mô hình mạng đồng đẳng này.
- Tìm hiểu và nghiên cứu các mô hình hệ thống Mobile Video Conference. Nghiên cứu giao thức các giao thức thiết lập cuộc gọi thông qua mạng Internet, cụ thể là giao thức SIP.
- Tìm hiểu và nghiên cứu các giao thức streaming video và audio thông qua mạng Internet như RTSP, RTP, RTCP. Đồng thời tìm hiểu cách thức thực thi streaming gửi và nhận dữ liệu trên nền tảng di động Android.
- Nghiên cứu và khảo sát các giải pháp phần cứng, phần mềm đồng thời đề xuất mô hình kiến trúc dịch vụ hội thảo truyền hình trên thiết bị di động.
- Nghiên cứu một số phương pháp đánh giá chất lượng video và một số thuật toán liên quan đến bài toán quản lý băng thông và giảm thiểu độ trễ.
- Triển khai mô hình hội thảo truyền hình, phát triển ứng dụng client chạy trên các nền tảng mobile tham gia hội thảo trực tuyến thông qua thiết bị di động.

### **4.2. Hướng phát triển**

Hiện tại chương trình đang ở mức độ thử nghiệm nên chỉ có thể xây dựng được mạng P2P trong mạng LAN Wifi. Hướng phát triển của chương trình sẽ là xây dựng module P2P tốt hơn, đưa mạng thành mạng P2P đúng nghĩa và có khả năng NAT (Network Address Translation).

Cải tiến cài đặt giao thức SIP trong chương trình để mở rộng khả năng chương trình sang hội nghị video và cách thức giao tiếp dễ dàng hơn.

Cải tiến cách thức truyền tải dữ liệu, tăng hiệu quả truyền tải dữ liệu video và audio.

## DANH MỤC TÀI LIỆU THAM KHẢO

### Tiếng Việt:

- 1) <http://www.sinhvienit.net>
- 2) <http://www.quantrimang.com>

### Tiếng Anh:

- 1) <http://www.cisco.com>
- 2) <http://www.wikipedia.org>
- 3) <http://www.vitamio.org>
- 4) <http://www.stackoverflow.com>
- 5) Voice and Video Conferencing Fundamentals, Scott Firestone, Thiya Ramlingam, and Steve Fry, Cisco Press, 2007.
- 6) RADVISION Ltd, An Overview of H.323 – SIP Interworking, Technical Report, 2001
- 7) Michael Gough and al, Video Conferencing over IP: Configure, secure and Troobleshoot, Syngress Publishing, Inc, 2006