

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC BÁCH KHOA**



**BÁO CÁO BÀI TẬP LỚN**

**MẠNG VIỄN THÔNG**

**Đề tài:**

**TÌM HIỂU VỀ PHẦN MỀM OPENSIPS VÀ XÂY DỰNG TỔNG ĐÀI VOIP**

**Learning about OpenSIPS software and Building VOIP switchboard**

**LỚP L01 - NHÓM 04 - HK242**

**GVHD: ĐÌNH QUỐC HÙNG**

**Thành viên trong nhóm:**

<b>MSSV</b>	<b>Họ và tên</b>	<b>Đánh giá</b>
2213758	Văn Đắc Phong Trực	35%
1910834	Trịnh Công Bắc	15%
2210329	Phan Lê Bình	15%
2213802	Tô Anh Tuấn	35%

**Thành phố Hồ Chí Minh, ngày .. tháng .. năm 2025**

## MỤC LỤC

<b>1. TỔNG QUAN VỀ PHẦN MỀM OPENSIPS .....</b>	<b>4</b>
<b>1.1. Giới thiệu chung về phần mềm OpenSIPS.....</b>	<b>4</b>
<b>1.2. Đặc điểm nổi bật của phần mềm OpenSIPS .....</b>	<b>6</b>
<b>1.3. Các tính năng chính của phần mềm OpenSIPS.....</b>	<b>7</b>
<b>1.4. Lợi ích của phần mềm OpenSIPS .....</b>	<b>7</b>
<b>1.5. Phân tích xử lý trong phần mềm OpenSIPS .....</b>	<b>8</b>
<b>1.5.1. Kỹ thuật xử lý tín hiệu điều khiển trong OpenSIPS .....</b>	<b>9</b>
<b>Phân tích cú pháp và xử lý thông điệp SIP .....</b>	<b>9</b>
<b>Định tuyến tín hiệu .....</b>	<b>10</b>
<b>Tối ưu hóa thời gian thực.....</b>	<b>10</b>
<b>Tích hợp với các công cụ xử lý tín hiệu truyền thông .....</b>	<b>11</b>
<b>Kết hợp với giao thức RTP: .....</b>	<b>11</b>
<b>Tích hợp với máy chủ truyền thông.....</b>	<b>11</b>
<b>Phân tích chất lượng dịch vụ (QoS) .....</b>	<b>12</b>
<b>1.5.2. Các mô-đun liên quan đến phân tích tín hiệu trong OpenSIPS .....</b>	<b>13</b>
<b>Mô-đun rtpproxy và rtpengine .....</b>	<b>13</b>
<b>Mô-đun dialog .....</b>	<b>13</b>
<b>Mô-đun statistics .....</b>	<b>14</b>
<b>Mô-đun event.....</b>	<b>14</b>
<b>1.5.3. Hạn chế và tiềm năng của OpenSIPS trong xử lý tín hiệu .....</b>	<b>14</b>
<b>1.6. Ứng dụng của phần mềm OpenSIPS .....</b>	<b>15</b>
<b>2. CÀI ĐẶT PHẦN MỀM OPENSIPS.....</b>	<b>16</b>
<b>3. GIỚI THIỆU VỀ VOIP.....</b>	<b>19</b>

<b>3.1. Tổng quan lý thuyết về VOIP.....</b>	<b>19</b>
<b>3.2. Đặc điểm nổi bật của VOIP .....</b>	<b>20</b>
<b>3.3. Kỹ thuật truyền dữ liệu VOIP.....</b>	<b>21</b>
<b>3.4. Ứng dụng .....</b>	<b>22</b>
<b>4. XÂY DỰNG TỔNG ĐÀI VOIP.....</b>	<b>22</b>
<b>4.1. Các phương thức có thể dùng để xây dựng tổng đài VOIP (phần mềm, kỹ thuật,...).....</b>	<b>22</b>
<b>4.1.1. VOIP hoạt động như thế nào? .....</b>	<b>22</b>
<b>4.1.2. Quá trình số hóa tín hiệu analog: .....</b>	<b>23</b>
<b>4.1.3. Các kỹ thuật sử dụng trong quá trình số hóa:.....</b>	<b>24</b>
<b>4.2. Giao thức SIP.....</b>	<b>24</b>
<b>4.2.1. Giới thiệu về SIP.....</b>	<b>24</b>
<b>4.2.2. Các thành phần trong SIP.....</b>	<b>25</b>
<b>4.2.3. Các bản tin trong SIP .....</b>	<b>26</b>
<b>4.2.4. Phương thức hoạt động.....</b>	<b>26</b>
<b>4.2.5. Tính năng của SIP .....</b>	<b>26</b>
<b>4.2.6. Các giao thức của SIP .....</b>	<b>27</b>
<b>4.3. Xây dựng tổng đài VOIP trên phần mềm OpenSIPS. ....</b>	<b>28</b>
<b>4.4. Demo sản phẩm.....</b>	<b>33</b>
<b>5. KẾT LUẬN CHUNG .....</b>	<b>49</b>

## MỤC LỤC HÌNH ẢNH

<i>Figure 1: Các module HTTPD</i> .....	33
<i>Figure 2: Socket UDP và TCP</i> .....	33
<i>Figure 3: Tạo domain</i> .....	34
<i>Figure 4: Server được reload</i> .....	34
<i>Figure 5: User tuan</i> .....	35
<i>Figure 6: User truc</i> .....	35
<i>Figure 7: Thông tin các user</i> .....	36
<i>Figure 8: Đăng nhập các user vào Zoiper5</i> .....	36
<i>Figure 9: Blink Sip Client</i> .....	37
<i>Figure 10: Tương tự với user còn lại</i> .....	38
<i>Figure 11: Active các user</i> .....	38
<i>Figure 12: Check status các user</i> .....	39
<i>Figure 13: Kết nối chéo trên Linphone</i> .....	39
<i>Figure 14: Kiểm tra lại trạng thái kết nối</i> .....	40
<i>Figure 15: Kết nối thành công</i> .....	41
<i>Figure 16: Capture các gói tin trên tổng đài</i> .....	41
<i>Figure 17: Flowdata của gói thoại</i> .....	43
<i>Figure 18: Sơ đồ quy trình chuyển gói tin với Cancel</i> .....	43
<i>Figure 19: Quy trình chuyển gói tin khi nhắc máy</i> .....	44
<i>Figure 20: Thiết lập cài đặt PSTN</i> .....	45
<i>Figure 21: Cấu hình thông tin cho thiết bị trong mạng PSTN</i> .....	45
<i>Figure 22: Các thiết bị sau thiết lập</i> .....	46
<i>Figure 23: Capture các gói tin giữa 2 gateways</i> .....	46
<i>Figure 24: Các gói tin cụ thể của call flow</i> .....	47
<i>Figure 25: Nhiều thành phần được cài đặt</i> .....	48

## 1. TỔNG QUAN VỀ PHẦN MỀM OPENSIPS

### 1.1. Giới thiệu chung về phần mềm OpenSIPS



OpenSIPS là một phiên bản mã nguồn mở hoàn chỉnh của một SIP server. OpenSIPS hơn một SIP proxy/router vì nó bao gồm tính năng của lớp ứng dụng. OpenSIPS như một SIP server, là thành phần cốt lõi của mọi giải pháp VoIP dựa trên nền SIP. Là một thiết bị định tuyến có khả năng mềm dẻo và tùy biến, OpenSIPS tích hợp voice, video, IM (Instant Message) và các dịch vụ Presence với một phương pháp có hiệu quả cao, nhờ được thiết kế kiểu module nên dễ dàng nâng cấp. OpenSIPS mang lại các tính năng tin cậy và hiệu suất cao. OpenSIPS là một trong những SIP server nhanh nhất, là một giải pháp thông suốt cho các tổ chức kinh doanh hoặc lớp vận chuyển.

OpenSIPS là một nền tảng mã nguồn mở được thiết kế để xử lý các giao thức SIP trong các hệ thống truyền thông thời gian thực. Nó hoạt động như một máy chủ proxy SIP, bộ định tuyến (router), bộ cân bằng tải (load balancer), hoặc thậm chí là một máy chủ đăng ký (registrar), tùy thuộc vào cấu hình và nhu cầu triển khai. OpenSIPS không trực tiếp xử lý các luồng truyền thông (media streams) như âm thanh hoặc video, mà

tập trung vào tín hiệu điều khiển, tức là các thông điệp SIP dùng để thiết lập, duy trì và kết thúc các phiên truyền thông.

Trong bối cảnh VoIP, tín hiệu điều khiển (signaling) và tín hiệu truyền thông (media) là hai thành phần chính. Tín hiệu điều khiển bao gồm các thông điệp SIP như INVITE, ACK, BYE, REGISTER, hoặc OPTIONS, được sử dụng để quản lý các phiên liên lạc. Tín hiệu truyền thông, thường được truyền qua giao thức RTP (Real-time Transport Protocol), chứa dữ liệu thực tế như âm thanh hoặc video. OpenSIPS chủ yếu xử lý tín hiệu điều khiển, nhưng nó có thể tương tác với các hệ thống xử lý tín hiệu truyền thông thông qua các mô-đun và tích hợp.

Vì vậy, khi nói đến "phân tích và xử lý tín hiệu" trong OpenSIPS, chúng ta cần hiểu rằng khái niệm này chủ yếu liên quan đến việc phân tích cú pháp, định tuyến, và tối ưu hóa các gói tin SIP, thay vì xử lý tín hiệu số truyền thống như trong các ứng dụng DSP (Digital Signal Processing). Tuy nhiên, một số khía cạnh của OpenSIPS, chẳng hạn như giám sát hiệu suất mạng hoặc tích hợp với các công cụ xử lý âm thanh, có thể gián tiếp liên quan đến các kỹ thuật phân tích tín hiệu.

OpenSIPS bao gồm sự đa dạng hợp nhất, sự đa dạng xuất phát từ nhiều người tham gia vào việc phát triển dự án và do tính phức tạp và các tính năng phong phú của OpenSIPS (nó tích hợp và kết hợp chặt chẽ rất nhiều chức năng và tính năng, đứng trong Top thiết bị định tuyến SIP). Để đạt được độ ổn định và tin cậy cao, thì cần phải hợp nhất tính đa dạng này.

Dự án OpenSIPS là sự tiếp tục của dự án OpenSER. Kế thừa OpenSER, tính mở rộng với cộng đồng và phát triển lên (liên tục phát triển, tăng cường và mở rộng code). OpenSIPS tiếp tục và nâng cấp OpenSER lên.

## 1.2. Đặc điểm nổi bật của phần mềm OpenSIPS

OpenSIPS bao gồm hơn 70 module, mỗi module có một chức năng riêng. Với giao diện module PLUG and PLAY, OpenSIPS dễ dàng mở rộng và nâng cấp các tính năng của nó mà không ảnh hưởng tới phần lõi, đồng thời càng làm tăng sự ổn định của hệ thống lõi. Số lượng và tính năng của các module được thay đổi theo từng phiên bản để tăng sự ổn định và tính mềm dẻo trong OpenSIPS.

Dưới đây là một số module trong phần mềm:

Tên module	Mô tả
CPL-C	Module Call Processing Language
ENUM	Module tra cứu Enum
LCR	Module định tuyến chi phí ít nhất
LOAD_BALANCER	Module cân bằng tải (cho các cuộc gọi)
MI_FIFO	Hỗ trợ FIFO cho giao diện quản lý
NAT_TRAVERSAL	Module thực hiện chức năng Nat traversal
PERL	Thực hiện chức năng Perl
XCAP_CLIENT	Hỗ trợ XCAP cho client
XMPP	Cho phép trao đổi tin nhắn tức thời giữa SIP Client và XMPP(Jabber) Client

### **1.3. Các tính năng chính của phần mềm OpenSIPS**

OpenSIP hoạt động chủ yếu như một máy chủ proxy SIP, có thể định tuyến các yêu cầu SIP giữa các thiết bị đầu cuối, chẳng hạn như điện thoại VoIP, máy chủ SIP khác hoặc các hệ thống truyền thông.

Phần mềm cũng hỗ trợ tính năng registrar SIP, cho phép các thiết bị SIP đăng ký với OpenSIP và thông báo vị trí của mình để nhận cuộc gọi.

OpenSIP có khả năng chuyển hướng cuộc gọi SIP đến các máy chủ hoặc hệ thống khác khi cần thiết, giúp phân phối tải hoặc điều phối cuộc gọi hiệu quả.

OpenSIP có thể được tích hợp với các máy chủ media (như Asterisk hoặc FreeSWITCH) để xử lý các cuộc gọi âm thanh và video.

Tính năng presence cho phép theo dõi trạng thái người dùng, như người dùng có đang bận, online, hay offline, giúp cải thiện hiệu quả giao tiếp trong hệ thống.

### **1.4. Lợi ích của phần mềm OpenSIPS**

Hiệu suất cao: OpenSIP có khả năng xử lý hàng triệu yêu cầu mỗi ngày, giúp các hệ thống VoIP lớn có thể hoạt động liên tục mà không gặp sự cố về hiệu suất.

Mở rộng và tùy chỉnh: Với cấu trúc module, OpenSIP có thể được mở rộng và tùy chỉnh dễ dàng để đáp ứng nhu cầu cụ thể của hệ thống. Các tính năng có thể được thêm vào qua các module như bảo mật, thanh toán, hoặc tích hợp với các hệ thống khác.

Hỗ trợ đa nền tảng: OpenSIP có thể hoạt động trên nhiều hệ điều hành khác nhau, bao gồm Linux, Windows (qua WSL), và macOS, giúp dễ dàng triển khai trên môi trường đa dạng.

Tính bảo mật cao: OpenSIP hỗ trợ các giao thức bảo mật như TLS và SRTP để mã hóa và bảo vệ các cuộc gọi VoIP khỏi các cuộc tấn công mạng.



## 1.5. Phân tích xử lý trong phần mềm OpenSIPS

OpenSIPS là một phần mềm mã nguồn mở nổi tiếng, được sử dụng rộng rãi như một máy chủ proxy SIP (Session Initiation Protocol) đa năng trong các hệ thống Voice over IP (VoIP), nhắn tin tức thời, hội nghị truyền hình và các ứng dụng liên quan đến giao thức SIP. Với vai trò trung tâm trong việc quản lý tín hiệu điều khiển (signaling) trong mạng VoIP, OpenSIPS đóng một vai trò quan trọng trong việc đảm bảo hiệu suất, độ tin cậy và khả năng mở rộng của các dịch vụ truyền thông thời gian thực. Tuy nhiên, câu hỏi đặt ra là liệu OpenSIPS có áp dụng các kỹ thuật phân tích và xử lý tín hiệu (signal processing), vốn thường được hiểu là các phương pháp xử lý tín hiệu số như âm thanh, hình ảnh hoặc dữ liệu analog, hay không. Trong luận văn này, chúng ta sẽ phân tích chi tiết cách OpenSIPS xử lý tín hiệu trong bối cảnh SIP, các kỹ thuật liên quan đến phân tích dữ liệu mạng, và vai trò của nó khi tích hợp với các hệ thống xử lý tín hiệu khác.

OpenSIPS là một nền tảng mã nguồn mở được thiết kế để xử lý các giao thức SIP trong các hệ thống truyền thông thời gian thực. Nó hoạt động như một máy chủ proxy SIP, bộ định tuyến (router), bộ cân bằng tải (load balancer), hoặc thậm chí là một máy chủ đăng ký (registrar), tùy thuộc vào cấu hình và nhu cầu triển khai. OpenSIPS không trực tiếp xử lý các luồng truyền thông (media streams) như âm thanh hoặc video, mà tập trung vào tín hiệu điều khiển, tức là các thông điệp SIP dùng để thiết lập, duy trì và kết thúc các phiên truyền thông.

Trong bối cảnh VoIP, tín hiệu điều khiển (signaling) và tín hiệu truyền thông (media) là hai thành phần chính. Tín hiệu điều khiển bao gồm các thông điệp SIP như INVITE, ACK, BYE, REGISTER, hoặc OPTIONS, được sử dụng để quản lý các phiên liên lạc. Tín hiệu truyền thông, thường được truyền qua giao thức RTP (Real-time Transport Protocol), chứa dữ liệu thực tế như âm thanh hoặc video. OpenSIPS chủ yếu xử lý tín hiệu điều khiển, nhưng nó có thể tương tác với các hệ thống xử lý tín hiệu truyền thông thông qua các mô-đun và tích hợp.

Vì vậy, khi nói đến "phân tích và xử lý tín hiệu" trong OpenSIPS, chúng ta cần hiểu rằng khái niệm này chủ yếu liên quan đến việc phân tích cú pháp, định tuyến, và tối ưu hóa các gói tin SIP, thay vì xử lý tín hiệu số truyền thống như trong các ứng dụng DSP (Digital Signal Processing). Tuy nhiên, một số khía cạnh của OpenSIPS, chẳng hạn như giám sát hiệu suất mạng hoặc tích hợp với các công cụ xử lý âm thanh, có thể gián tiếp liên quan đến các kỹ thuật phân tích tín hiệu.

### **1.5.1. Kỹ thuật xử lý tín hiệu điều khiển trong OpenSIPS**

#### **Phân tích cú pháp và xử lý thông điệp SIP**

OpenSIPS hoạt động như một bộ xử lý tín hiệu điều khiển, trong đó "tín hiệu" là các thông điệp SIP. Các thông điệp này được phân tích cú pháp (parsed) để trích xuất các thông tin quan trọng như:

- *Tiêu đề SIP (SIP headers)*: Bao gồm thông tin như To, From, Call-ID, CSeq, và Via, được sử dụng để xác định nguồn, đích, và trạng thái của phiên liên lạc.
- *Thân thông điệp (message body)*: Thường chứa thông tin SDP (Session Description Protocol), mô tả các tham số truyền thông như codec, địa chỉ IP, và cổng.

Quá trình phân tích cú pháp trong OpenSIPS sử dụng các thuật toán hiệu quả để đảm bảo xử lý nhanh chóng các thông điệp SIP trong thời gian thực. Điều này đòi hỏi các kỹ thuật tối ưu hóa, chẳng hạn như:

- *Tìm kiếm chuỗi (string matching)*: Để xác định các trường tiêu đề hoặc các tham số cụ thể trong thông điệp.
- *Quản lý bộ nhớ đệm (buffer management)*: Để xử lý khối lượng lớn các gói tin mà không gây ra độ trễ.

## **Định tuyến tín hiệu**

Sau khi phân tích, OpenSIPS áp dụng các thuật toán định tuyến để quyết định cách chuyển tiếp thông điệp SIP. Các kỹ thuật định tuyến này có thể được xem như một dạng xử lý tín hiệu trạng thái, trong đó trạng thái của hệ thống (ví dụ: vị trí người dùng, tải máy chủ) được phân tích để đưa ra quyết định. Các phương pháp định tuyến bao gồm:

- *Định tuyến dựa trên bảng tra cứu (lookup-based routing)*: Sử dụng cơ sở dữ liệu hoặc bảng định tuyến để ánh xạ URI của người dùng đến địa chỉ đích.
- *Định tuyến động (dynamic routing)*: Dựa trên các tiêu chí thời gian thực như tải máy chủ, chất lượng mạng, hoặc chính sách QoS.
- *Định tuyến dựa trên chính sách (policy-based routing)*: Áp dụng các quy tắc do quản trị viên định nghĩa, ví dụ: chặn các cuộc gọi từ một số nguồn cụ thể.

## **Tối ưu hóa thời gian thực**

OpenSIPS được thiết kế để xử lý tín hiệu trong môi trường thời gian thực, nơi độ trễ thấp là yếu tố quan trọng. Các kỹ thuật tối ưu hóa bao gồm:

- *Xử lý song song (parallel processing)*: OpenSIPS sử dụng kiến trúc đa luồng (multi-threading) để xử lý đồng thời nhiều thông điệp SIP.
- *Tối ưu hóa tài nguyên*: Giảm thiểu việc sử dụng CPU và bộ nhớ thông qua các cấu trúc dữ liệu hiệu quả như bảng băm (hash tables) hoặc cây tìm kiếm (search trees).

Những kỹ thuật này, mặc dù không phải là xử lý tín hiệu số theo nghĩa truyền thống, nhưng có sự tương đồng với các phương pháp xử lý tín hiệu thời gian thực, nơi dữ liệu cần được phân tích và phản hồi ngay lập tức.

## **Tích hợp với các công cụ xử lý tín hiệu truyền thông**

Mặc dù OpenSIPS không trực tiếp xử lý tín hiệu truyền thông như âm thanh hoặc video, nó thường được triển khai trong các hệ thống tích hợp với các công cụ chuyên về xử lý tín hiệu, chẳng hạn như máy chủ truyền thông (media servers) hoặc bộ chuyển mã (transcoders). Các công cụ này thực hiện các tác vụ xử lý tín hiệu số, và OpenSIPS đóng vai trò như một bộ điều phối tín hiệu điều khiển để hỗ trợ các tác vụ đó.

### **Kết hợp với giao thức RTP:**

OpenSIPS thường làm việc với các máy chủ RTP như RTPproxy hoặc RTPengine để quản lý luồng truyền thông. Các máy chủ này thực hiện các chức năng như:

- *NAT traversal*: Đảm bảo luồng RTP có thể vượt qua các tường lửa hoặc bộ định tuyến NAT.
- *Chuyển đổi codec*: Chuyển đổi giữa các định dạng mã hóa âm thanh hoặc video (ví dụ: từ G.711 sang Opus).
- *Giám sát chất lượng*: Thu thập dữ liệu về độ trễ, jitter, hoặc mất gói tin để đánh giá hiệu suất truyền thông.

OpenSIPS hỗ trợ các máy chủ RTP bằng cách cung cấp thông tin SDP từ các thông điệp SIP, giúp thiết lập các tham số truyền thông chính xác. Mặc dù bản thân OpenSIPS không thực hiện các thuật toán xử lý tín hiệu như mã hóa/giải mã, nhưng nó đảm bảo rằng các luồng truyền thông được thiết lập và định tuyến đúng cách.

### **Tích hợp với máy chủ truyền thông**

Các máy chủ truyền thông như Asterisk hoặc FreeSWITCH thường được sử dụng cùng với OpenSIPS trong các hệ thống VoIP lớn. Những máy chủ này thực hiện các tác vụ xử lý tín hiệu cụ thể, bao gồm:

- *Mã hóa và giải mã (codec processing)*: Sử dụng các thuật toán nén tín hiệu như G.729 hoặc Opus để giảm băng thông mà vẫn duy trì chất lượng âm thanh.
- *Loại bỏ tiếng vang (echo cancellation)*: Áp dụng các bộ lọc tín hiệu để loại bỏ hiện tượng tiếng vang trong cuộc gọi.
- *Giảm nhiễu (noise reduction)*: Sử dụng các kỹ thuật xử lý tín hiệu số như bộ lọc thông thấp (low-pass filter) hoặc thông cao (high-pass filter) để cải thiện chất lượng âm thanh.
- *Phát hiện giọng nói (voice activity detection - VAD)*: Xác định các khoảng lặng trong cuộc gọi để tối ưu hóa băng thông.

OpenSIPS hỗ trợ các máy chủ này bằng cách định tuyến các thông điệp SIP đến các điểm cuối phù hợp và quản lý trạng thái cuộc gọi. Ví dụ, trong một hệ thống hội nghị truyền hình, OpenSIPS có thể phân phối các thông điệp INVITE đến nhiều người tham gia, trong khi máy chủ truyền thông xử lý việc trộn tín hiệu video (video mixing) hoặc âm thanh (audio mixing).

### **Phân tích chất lượng dịch vụ (QoS)**

OpenSIPS có thể thu thập dữ liệu về hiệu suất mạng, chẳng hạn như độ trễ, jitter, hoặc tỷ lệ mất gói tin, thông qua các mô-đun như rtpmproxy hoặc rtpe. Những dữ liệu này có thể được sử dụng để:

- Tối ưu hóa định tuyến truyền thông, ví dụ: chọn máy chủ RTP gần nhất với người dùng.
- Phát hiện các vấn đề về chất lượng cuộc gọi, chẳng hạn như gián đoạn hoặc méo tiếng.

- Cung cấp thông tin cho các công cụ phân tích tín hiệu bên ngoài để cải thiện hiệu suất hệ thống.

Mặc dù OpenSIPS không thực hiện các thuật toán phân tích tín hiệu như biến đổi Fourier (Fourier Transform) hoặc lọc tín hiệu (signal filtering), nhưng nó cung cấp nền tảng để các công cụ khác thực hiện những tác vụ này.

### 1.5.2. Các mô-đun liên quan đến phân tích tín hiệu trong OpenSIPS

OpenSIPS cung cấp nhiều mô-đun hỗ trợ các chức năng liên quan đến phân tích và xử lý dữ liệu mạng, có thể được xem là một dạng xử lý tín hiệu trong bối cảnh VoIP. Một số mô-đun quan trọng bao gồm:

#### Mô-đun rtpproxy và rtpengine

Những mô-đun này quản lý luồng RTP và cung cấp các chức năng như:

- *Điều chỉnh SDP*: Sửa đổi thông tin SDP trong thông điệp SIP để đảm bảo luồng truyền thông được định tuyến chính xác.
- *Giám sát hiệu suất*: Thu thập dữ liệu về chất lượng truyền thông, chẳng hạn như độ trễ hoặc mất gói tin.
- *Chuyển đổi codec*: Hỗ trợ chọn codec phù hợp giữa các điểm cuối.

Mặc dù các mô-đun này không trực tiếp thực hiện xử lý tín hiệu số, chúng đóng vai trò quan trọng trong việc đảm bảo rằng tín hiệu truyền thông được truyền tải hiệu quả.

#### Mô-đun dialog

Mô-đun dialog theo dõi trạng thái của các phiên liên lạc (dialog state), bao gồm thông tin như thời gian bắt đầu, thời lượng cuộc gọi, và trạng thái kết thúc. Dữ liệu này có thể được sử dụng để:

- Phân tích hiệu suất hệ thống, chẳng hạn như số lượng cuộc gọi đồng thời hoặc tỷ lệ lỗi.
- Phát hiện các mẫu bất thường, ví dụ: các cuộc gọi bị gián đoạn do vấn đề mạng.

### **Mô-đun statistics**

Mô-đun này thu thập và phân tích dữ liệu về hoạt động của OpenSIPS, chẳng hạn như:

- Số lượng thông điệp SIP được xử lý mỗi giây.
- Thời gian phản hồi trung bình của hệ thống.
- Tỷ lệ lỗi hoặc thất bại trong định tuyến.

Những số liệu này có thể được sử dụng để tối ưu hóa hiệu suất hệ thống và gián tiếp hỗ trợ các ứng dụng xử lý tín hiệu bằng cách đảm bảo rằng tín hiệu điều khiển được truyền tải đáng tin cậy.

### **Mô-đun event**

Mô-đun event cho phép OpenSIPS phản hồi các sự kiện cụ thể, chẳng hạn như nhận được một thông điệp SIP hoặc phát hiện một lỗi mạng. Cơ chế này tương tự như xử lý tín hiệu dựa trên sự kiện (event-driven signal processing), nơi hệ thống phản ứng với các kích thích thời gian thực.

## **1.5.3. Hạn chế và tiềm năng của OpenSIPS trong xử lý tín hiệu**

### **Hạn chế:**

Không phải nền tảng DSP: OpenSIPS không được thiết kế để thực hiện các tác vụ xử lý tín hiệu số như mã hóa âm thanh, lọc tín hiệu, hoặc phân tích phổ

(spectral analysis). Những tác vụ này thường được thực hiện bởi các công cụ chuyên dụng như Asterisk, FreeSWITCH, hoặc phần cứng DSP.

Tập trung vào tín hiệu điều khiển: Vai trò chính của OpenSIPS là xử lý tín hiệu điều khiển (control plane), không phải luồng truyền thông (media plane). Do đó, các kỹ thuật xử lý tín hiệu mà nó hỗ trợ chủ yếu liên quan đến phân tích và định tuyến gói tin SIP.

Phụ thuộc vào tích hợp: Để thực hiện các tác vụ xử lý tín hiệu, OpenSIPS cần được tích hợp với các hệ thống khác, điều này có thể làm tăng độ phức tạp của triển khai.

### **Tiềm năng:**

Hỗ trợ hệ thống thời gian thực: OpenSIPS được tối ưu hóa cho xử lý thời gian thực, điều này làm cho nó trở thành một nền tảng lý tưởng để hỗ trợ các ứng dụng yêu cầu phân xử lý tín hiệu nhanh chóng.

Tích hợp với AI/ML: Các hệ thống hiện đại có thể tích hợp OpenSIPS với các công cụ học máy để phân tích dữ liệu cuộc gọi, phát hiện gian lận, hoặc tối ưu hóa định tuyến. Những ứng dụng này có thể sử dụng các kỹ thuật xử lý tín hiệu để phân tích mẫu dữ liệu thời gian thực.

Khả năng mở rộng: Với kiến trúc mô-đun và khả năng xử lý khối lượng lớn thông điệp SIP, OpenSIPS có thể được sử dụng trong các hệ thống phân tích tín hiệu quy mô lớn, chẳng hạn như giám sát mạng hoặc phân tích chất lượng dịch vụ.

## **1.6. Ứng dụng của phần mềm OpenSIPS**

Trong các dịch vụ VoIP: OpenSIPS được sử dụng rộng rãi trong các hệ thống gọi điện qua Internet (VoIP), giúp giảm chi phí và cải thiện hiệu quả truyền thông.



Trong các trung tâm cuộc gọi (Call Centers): OpenSIPS có thể xử lý số lượng lớn cuộc gọi đồng thời, làm cho nó trở thành một công cụ lý tưởng cho các trung tâm cuộc gọi lớn.

Hệ thống tổng đài IP (IP PBX): OpenSIPS có thể được sử dụng để xây dựng hệ thống tổng đài VoIP cho các doanh nghiệp, kết nối các điện thoại VoIP và các thiết bị khác.

## 2. CÀI ĐẶT PHẦN MỀM OPENSIPS

### *Trên hệ điều hành Linux (Ubuntu)*

Để cài đặt OpenSIPS trên Linux, bạn có thể làm theo các bước sau:

Cập nhật hệ thống:

```
sudo apt update
```

```
sudo apt upgrade
```

Cài đặt các phụ thuộc:

```
sudo apt install build-essential git cmake libpcap-dev libssl-dev libsctp-dev  
libcurl4-openssl-dev libxml2-dev libsqlite3-dev libevent-dev libjson-c-dev  
libmysqlclient-dev
```

```
sudo su
```

```
apt update
```

```
apt install wget gnupg2
```

Thêm OpenSIPS cho repository của Ubuntu

```
curl https://apt.opensips.org/opensips- org. gpg -o/ usr/ share/ keyrings/  
opensips -org.gpg
```

```
echo "deb[signed-by =/usr/share/ keyrings/ opensips - org .gpg  
]https://apt.opensips.org focal 3.5-releases" >/etc/apt/sources.list.d/opensips.list
```

```
curl https://apt.opensips.org/opensips-org.gpg -o /usr/ share/ keyrings  
/opensips-org.gpg
```

```
echo "deb [signed-by=/usr/share/keyrings/opensips-org.gpg] https://  
apt.opensips. org focal cli-nightly" > /etc/apt/sources.list.d/opensips-cli.list
```

Cài đặt MySQLserver

```
apt install mysql-server
```

```
sudo systemctl status MySQL
```

Tạo database

```
sudo mysql -u root -p
```

```
UNINSTALL COMPONENT 'file://component_validate_password';
```

```
Exit
```

Lệnh tạo các bảng

```
sudo opensips-cli -x database create
```

Bật tự động khởi động OpenSIPS

```
sudo systemctl enable opensips
```

```
sudo systemctl start opensips
```

Cài đặt OpenSIPS Control Panel

```
sudo apt install -y apache2 php php-mysql php-cli php-xml php-curl git unzip
```

```
sudo nano /etc/apache2/sites-available/opensips-cp.conf
```

Dẫn liên kết vào:

```
<VirtualHost *:80>

    DocumentRoot /var/www/html/opensips-cp

    <Directory /var/www/html/opensips-cp>

        Options Indexes FollowSymLinks

        AllowOverride All

        Require all granted

    </Directory>

</VirtualHost>
```

Tải OpenSIPS-CP GUI về và cấp quyền

```
cd /var/www/html

sudo git clone https://github.com/OpenSIPS/opensips-cp.git

sudo chown -R www-data: www-data opensips-cp

mysql -Dopensips -p < config/db_schema.mysql

cp config/tools/system/smonitor/opensips_stats_cron /etc/cron.d/

sudo a2ensite opensips-cp

sudo systemctl restart apache2
```

Tải mã nguồn OpenSIPS:

```
git clone https://github.com/OpenSIP/OpenSIP.git

cd OpenSIP
```

Biên dịch và cài đặt:

```
make clean
```

*make*

*sudo make install*

Cấu hình và khởi động OpenSIPS: Sau khi cài đặt, bạn có thể chỉnh sửa file cấu hình *opensip.cfg* và khởi động dịch vụ OpenSIPS.

### ***Trên hệ điều hành Windows (Thông qua WSL)***

Để cài đặt trên Windows, bạn có thể sử dụng Windows Subsystem for Linux (WSL) và làm theo các bước tương tự như trên Linux.

## **3. GIỚI THIỆU VỀ VOIP**

### **3.1. Tổng quan lý thuyết về VOIP**

VoIP, hay Voice over Internet Protocol (Giọng nói qua Giao thức Internet), là một công nghệ cho phép truyền tải tín hiệu thoại (âm thanh) qua mạng Internet hoặc các mạng dựa trên giao thức IP thay vì sử dụng hệ thống điện thoại truyền thống (PSTN - Public Switched Telephone Network). Công nghệ này chuyển đổi tín hiệu giọng nói analog thành các gói dữ liệu số, sau đó truyền qua mạng IP và được tái tạo lại thành âm thanh ở phía người nhận.

VoIP cho phép tạo cuộc gọi dùng kết nối băng thông rộng thay vì dùng đường dây điện thoại tương tự (analog). Nhiều dịch vụ VoIP có thể chỉ cho phép bạn gọi người khác dùng cùng loại dịch vụ, tuy nhiên cũng có những dịch vụ cho phép gọi những người khác dùng số điện thoại như số nội bộ, đường dài, di động, quốc tế. Trong khi cũng có những dịch vụ chỉ làm việc qua máy tính, cũng có vài dịch vụ dùng điện thoại truyền thống qua một bộ điều hợp (adaptor). Nguyên tắc hoạt động của VoIP bao gồm việc số hoá tín hiệu tiếng nói, thực hiện việc nén tín hiệu số, chia nhỏ các gói nếu cần và truyền gói tin này qua mạng, tới nơi nhận các gói tin này được ráp lại theo đúng thứ tự của bản tin, giải mã tín hiệu tương tự phục hồi lại tiếng nói ban đầu.

VoIP tận dụng hạ tầng mạng dữ liệu sẵn có, giúp giảm chi phí và tăng tính linh hoạt so với các hệ thống điện thoại truyền thống.

OpenSIPS là một phần mềm mạnh mẽ và linh hoạt cho các hệ thống VoIP quy mô lớn. Với khả năng mở rộng, hiệu suất cao, và hỗ trợ đa nền tảng, OpenSIPS là một công cụ quan trọng cho các nhà cung cấp dịch vụ viễn thông, doanh nghiệp và các tổ chức muốn triển khai hệ thống viễn thông của riêng mình.

Với các tính năng như SIP Proxy, Register, Media Server Integration và hỗ trợ bảo mật, OpenSIPS đáp ứng tốt nhu cầu của các hệ thống viễn thông hiện đại.

### **3.2. Đặc điểm nổi bật của VOIP**

#### **Ưu điểm:**

*Thứ nhất*, không cần xây dựng hạ tầng riêng như mạng điện thoại truyền thống, VoIP sử dụng mạng Internet sẵn có, giúp giảm chi phí cuộc gọi, đặc biệt là đường dài hoặc quốc tế.

*Thứ hai*, người dùng (user) có thể truy cập và sử dụng, kết nối tới đầu cuối người nhận từ bất cứ vị trí địa lý nào miễn là nơi đó có phủ sóng Internet, hỗ trợ trên nhiều thiết bị, nền tảng như điện thoại thông minh, Tablet, Laptop hay Desktop.

*Thứ ba*, với sự đa dạng dịch vụ được cung cấp từ các nhà phát triển phần mềm cũng như được tích hợp đầy đủ trên mọi nền tảng điều hành của các thiết bị sử dụng mạng, VoIP cũng có thể cho phép liên lạc kết hợp thoại và video, tin nhắn hoặc gửi dữ liệu trong cùng một hệ thống hoặc ứng dụng.

*Thứ tư*, vì không chiếm diện tích vật lý lớn mà chỉ sử dụng kho dữ liệu trên nền tảng ảo (hay điện toán đám mây – Cloud Computing) nên việc mở rộng hạ tầng nhằm tăng lượng người dùng sẽ không khó khăn và không tốn nhiều chi phí để phát triển.

#### **Nhược điểm:**

*Thứ nhất*, phụ thuộc vào Internet, vì thế nên chất lượng các dịch vụ sẽ phụ thuộc vào tốc độ và độ ổn định của kết nối mạng. Nếu mạng chậm hoặc không ổn định, kết nối có thể bị nhiễu, méo hoặc mất kết nối.

*Thứ hai*, dữ liệu được đăng ký và lưu trữ trên các máy chủ (server) ảo nên nếu bảo mật không tốt sẽ dễ dàng bị các đối tượng tội phạm mạng tấn công nhằm đánh cắp dữ liệu thông tin các cá nhân phục vụ cho việc mua bán thông tin trái phép và hàng loạt các hoạt động phạm pháp khác. Điều này ảnh hưởng không nhỏ đến các vấn đề bảo mật thông tin cá nhân, doanh nghiệp.

*Thứ ba*, tuy tốc độ truyền dữ liệu hay truy cập Internet cao nhưng sẽ không tránh khỏi hiện tượng delay, giật do lượng bit truyền dẫn lớn và bị lost bit dẫn đến retransmission hoặc mất sound, frame do sự cố bất kỳ trên server.

*Thứ tư*, không giống điện thoại truyền thống vẫn có thể hoạt động được ngay cả khi mất điện, VoIP cần cung cấp điện liên tục cho toàn bộ các thiết bị sử dụng liên quan như thiết bị đầu cuối, tổng đài hoặc cả hệ thống máy chủ (server).

### **3.3. Kỹ thuật truyền dữ liệu VOIP**

Quá trình hoạt động của VoIP có thể được chia thành các bước cơ bản sau:

*Chuyển đổi tín hiệu (Analog-to-Digital Conversion)*: Giọng nói của người dùng được thu qua micro và chuyển từ dạng sóng analog sang dữ liệu số bằng các codec (như G.711, G.729).

*Nén dữ liệu*: Để tiết kiệm băng thông, dữ liệu âm thanh thường được nén trước khi truyền.

*Đóng gói dữ liệu*: Dữ liệu âm thanh số được chia thành các gói tin (packets) theo giao thức IP.

*Truyền tải qua mạng:* Các gói tin này được gửi qua mạng Internet hoặc mạng nội bộ (LAN/WAN) đến đích.

*Giải mã và tái tạo:* Ở phía người nhận, các gói tin được sắp xếp lại, giải nén và chuyển đổi ngược về dạng âm thanh analog để phát qua loa.

Các giao thức chính hỗ trợ VoIP bao gồm:

*SIP (Session Initiation Protocol):* Quản lý phiên kết nối giữa các thiết bị.

*RTP (Real-time Transport Protocol):* Truyền dữ liệu âm thanh theo thời gian thực.

*H.323:* Một tiêu chuẩn khác cho việc truyền thông đa phương tiện qua mạng IP.

### **3.4. Ứng dụng**

VoIP được ứng dụng rộng rãi trong nhiều lĩnh vực:

*Doanh nghiệp:* Các hệ thống tổng đài IP (IP PBX) thay thế tổng đài truyền thống, hỗ trợ hội nghị thoại/video, chăm sóc khách hàng.

*Cá nhân:* Các ứng dụng như Skype, WhatsApp, Zoom sử dụng VoIP để cung cấp dịch vụ gọi thoại miễn phí hoặc chi phí thấp.

*Trung tâm liên lạc (Call Center):* VoIP giúp tối ưu hóa chi phí và quản lý cuộc gọi hiệu quả.

*Giáo dục và y tế:* Hỗ trợ học trực tuyến, hội chẩn từ xa qua kết hợp thoại và video.

## **4. XÂY DỰNG TỔNG ĐÀI VOIP**

**4.1. Các phương thức có thể dùng để xây dựng tổng đài VOIP (phần mềm, kỹ thuật,...).**

**4.1.1. VOIP hoạt động như thế nào?**

Trong VoIP khi nói vào ống nghe hay microphone, giọng nói sẽ tạo ra tín hiệu điện từ, đó là những tín hiệu analog. Tín hiệu analog được chuyển sang tín hiệu số dùng thuật toán đặc biệt để chuyển đổi. Những thiết bị khác nhau có cách chuyển đổi khác nhau như VoIP phone hay softphone, nếu dùng điện thoại analog thông thường thì cần một Telephony Adapter (TA). Sau đó giọng nói được số hóa sẽ được đóng vào gói tin và gửi trên mạng IP.

#### **Các bước cơ bản để thực hiện một cuộc gọi trong VoIP:**

*Thứ nhất:* Xác định địa điểm cần gọi đến (mã quốc gia, mã tỉnh,...) và bấm số cần gọi đến.

*Thứ hai:* Các kết nối giữa người gọi và người nhận sẽ được thiết lập.

*Thứ ba:* Khi nói vào ống nghe hay microphone, giọng nói sẽ tạo ra tín hiệu điện từ, đó là những tín hiệu analog. Tín hiệu analog được chuyển sang tín hiệu số dùng thuật toán đặc biệt để chuyển đổi. Sau đó giọng nói được số hóa sẽ được đóng thành gói tin và gửi trên mạng IP. Trong suốt tiến trình một giao thức như SIP hay H323 sẽ được dùng để điều khiển (control) cuộc gọi như là thiết lập, quay số, ngắt kết nối,... và RTP thì được dùng cho tính năng đảm bảo độ tin cậy và duy trì chất lượng dịch vụ trong quá trình truyền.

*Thứ tư:* Dữ liệu sẽ được truyền tải qua kết nối được thiết lập lúc đầu.

*Thứ năm:* Dữ liệu chứa âm thanh mà bạn nói sẽ được chuyển hóa trở lại thành âm thanh mà người nghe hiểu được và âm thanh bạn nói ra sẽ được phát ra bên phía người nhận.

#### **4.1.2. Quá trình số hóa tín hiệu analog:**

Biểu diễn tín hiệu analog thành dạng số (digital) là công việc khó khăn. Vì bản thân dạng âm thanh như giọng nói con người ở dạng analog do đó phải cần một số lượng lớn các giá trị digital để biểu diễn biên độ (amplitude), tần số, và pha (phase), chuyển đổi những giá trị đó thành dạng số nhị phân (0 & 1) là rất khó khăn.



Vì vậy, để thực hiện sự chuyển đổi này chúng ta cần phải dùng đến thiết bị được gọi là codec (coder-decoder) hay là thiết bị mã hóa và giải mã. Tín hiệu analog được đặt vào đầu vào của thiết bị này và được chuyển thành các chuỗi số nhị phân ở đầu ra. Sau đó quá trình này thực hiện trở lại bằng cách chuyển đổi chuỗi số nhị phân thành dạng analog ở đầu cuối.

Có 4 bước liên quan đến quá trình số hóa một tín hiệu analog:

1. Lấy mẫu (Sampling)
2. Lượng tử hóa (Quantization)
3. Mã hóa (Encoding)
4. Nén giọng nói (Voice Compression).

#### **4.1.3. Các kỹ thuật sử dụng trong quá trình số hóa:**

**Multiplexing:** Ghép kênh là qui trình chuyển một số tín hiệu đồng thời qua một phương tiện truyền dẫn.

**TDM** (Time Division Multiplexing): Ghép kênh phân chia theo thời gian. Phân phối khoảng thời gian xác định vào mỗi kênh, mỗi kênh chiếm đường truyền cao tốc trong suốt một khoảng thời gian theo định kì.

**FDM** (Frequency Division Multiplexing): Ghép kênh phân chia theo tần số. Mỗi kênh được phân phối theo một băng tần xác định, thông thường có bề rộng 4Khz cho dịch vụ thoại.

**PCM** (Pulse Code Modulation): Điều chế theo mã: là phương pháp thông dụng nhất chuyển đổi các tín hiệu analog sang dạng digital ( và ngược lại) để có thể vận chuyển qua một hệ thống truyền dẫn số hay các quá trình xử lý số.

## **4.2. Giao thức SIP.**

### **4.2.1. Giới thiệu về SIP**

*Thứ nhất:* **SIP** (Session Initiation Protocol) là giao thức báo hiệu điều khiển lớp ứng dụng được dùng để thiết lập, duy trì, kết thúc các phiên truyền thông đa phương tiện (multimedia). Các phiên multimedia bao gồm thoại Internet, hội nghị và các ứng dụng tương tự có liên quan đến các phương tiện truyền đạt (media) như âm thanh, hình ảnh và dữ liệu.

*Thứ hai:* **SIP** sử dụng các bản tin mời (INVITE) để thiết lập các phiên và mang các thông tin mô tả mang phiên truyền dẫn. SIP hỗ trợ các phiên đơn bá (unicast) và quảng bá (multicast) tương ứng các cuộc gọi điểm tới điểm và các cuộc gọi đa điểm.

*Thứ ba:* **SIP** là một giao thức dạng văn bản, rất công khai và linh hoạt. Được thiết kế tương thích tương thích với các giao thức khác như TCP, UDP, IP,... để cung cấp một lĩnh vực rộng hơn cho dịch vụ VoIP.

#### **4.2.2. Các thành phần trong SIP.**

SIP gồm hai thành phần lớn là SIP client (là thiết bị hỗ trợ giao thức SIP) và SIP server (là thiết bị trong mạng xử lý các bản tin SIP). Trong SIP có 5 thành phần quan trọng là:

*Thứ nhất:* User Agents (UA) là các đầu cuối trong mạng SIP, nó đại diện cho phía người sử dụng để khởi tạo một yêu cầu tới SIP server hoặc User Agent server.

*Thứ hai:* Proxy server làm nhiệm vụ chuyển tiếp các SIP request tới các nơi khác trong mạng. Chức năng chính của nó là định tuyến cho các bản tin đến đích.

*Thứ ba:* Redirect server là user agent server nhận các bản tin request từ các user agent client và trả về bản tin return để thông báo thiết bị là chuyển hướng bản tin tới địa chỉ khác – tự liên lạc thông qua địa chỉ trả về.

*Thứ tư:* Registrar server là server nhận bản tin SIP Register yêu cầu cập nhật thông tin mà user agent cung cấp từ bản tin Register.

*Thứ năm:* Location Server là lưu lượng thông tin, trạng thái hiện tại của người dùng trong mạng SIP.

#### **4.2.3. Các bản tin trong SIP**

- *INVITE*: bắt đầu thiết lập cuộc gọi bằng cách gửi bản tin mời đầu cuối khác tham gia
- *ACK*: bản tin này khẳng định máy trạm đã nhận được các bản tin trả lời bản tin INVITE
- *BYE*: bắt đầu kết thúc cuộc gọi
- *CANCEL*: hủy yêu cầu nằm trong hàng đợi
- *REGISTER*: thiết bị đầu cuối của SIP sử dụng bản tin này để đăng ký với máy chủ đăng ký
- *OPTION*: sử dụng để xác định năng lực của máy chủ
- *INFO*: sử dụng để tải các thông tin như âm báo
- *REQUEST*: cho phép user agent và proxy có thể xác định người dùng, khởi tạo, sửa đổi, hủy một phiên.
- *RETURN*: được gửi bởi user agent server hoặc SIP server để trả lời cho một bản tin request trước đó.

#### **4.2.4. Phương thức hoạt động**

Hoạt động của máy chủ ủy quyền (Proxy Server): SIP Client [userA@yahoo.com](mailto:userA@yahoo.com) gửi bản tin INVITE cho [userB@hotmail.com](mailto:userB@hotmail.com) để mời tham gia cuộc gọi.

Hoạt động của máy chủ chuyển đổi địa chỉ (Redirect Server).

#### **4.2.5. Tính năng của SIP**

*Thiết lập một phiên:* SIP sử dụng bản tin INVITE để yêu cầu thiết lập một phiên truyền thông.

*Đơn giản và có khả năng mở rộng:* SIP có rất ít bản tin, không có chức năng thừa nhưng SIP có thể sử dụng để thiết lập nhưng phiên kết nối phức tạp như hội nghị... Các phần mềm của máy chủ ủy quyền, máy chủ đăng ký, máy chủ chuyển đổi địa chỉ,...có thể chạy trên các máy chủ khác nhau và việc cài đặt thêm máy chủ hoàn toàn không ảnh hưởng đến máy chủ đã có.

*Hỗ trợ tối đa sự di động của đầu cuối:* do máy chủ ủy quyền, máy chủ đăng ký và máy chủ chuyển đổi địa chỉ hệ thống luôn nắm được địa điểm chính xác của thuê bao. Ví dụ thuê bao với địa chỉ *ptit@vnpt.com.vn* có thể nhận được cuộc gọi thoại hay thông điệp ở bất cứ địa điểm nào qua bất cứ đầu cuối nào như máy tính để bàn, máy xách tay, điện thoại SIP... Với SIP rất nhiều dịch vụ di động mới được hỗ trợ.

*Định vị người sử dụng:* những người sử dụng đầu cuối sẽ luôn di động và địa chỉ IP của họ là không cố định, các đầu cuối có thể đăng ký với một SIP server thông qua bản tin REGISTER, SIP server sẽ lưu lại địa chỉ IP của đầu cuối đăng ký. Khi có một yêu cầu thiết lập cuộc gọi tới SIP server, SIP server sẽ tìm địa chỉ của người được gọi và forward bản tin INVITE tới người được gọi.

#### **4.2.6. Các giao thức của SIP**

*UDP (User Datagram Protocol):* là giao thức tầng vận chuyển không có điều khiển tắc nghẽn. Nó được dùng để vận chuyển bản tin SIP vì đơn giản và thích hợp với các ứng dụng thời gian thực.

*TCP (Transmission Control Protocol):* là giao thức ở tầng vận chuyển do có điều khiển tắc nghẽn, hơn nữa có thể vận chuyển nhiều gói tin có kích thước bất kỳ.

*SDP (Session Description Protocol)*: được sử dụng để mô tả các thông số media cho một cuộc gọi, các thông số này là các thông tin về băng thông, các chuẩn hóa audio, video và một số thông tin khác.

#### **4.3. Xây dựng tổng đài VOIP trên phần mềm OpenSIPS.**

OpenSIPS là một SIP proxy server mạnh mẽ, chuyên xử lý và quản lý các yêu cầu SIP trong hệ thống VoIP. Để giúp bạn hiểu rõ hơn về quá trình này, dưới đây là từng bước chi tiết về cách OpenSIP tiếp nhận, xử lý và quản lý cuộc gọi

##### **Các thành phần chính trong hệ thống tổng đài VoIP sử dụng OpenSIP**

*SIP Proxy/Server*: OpenSIPS sẽ nhận và xử lý các yêu cầu SIP (như *INVITE*, *REGISTER*, *BYE*) từ các thiết bị đầu cuối VoIP và các máy chủ SIP khác. Nó sẽ đảm nhận vai trò định tuyến và chuyển tiếp các yêu cầu này tới các máy chủ hoặc thiết bị khác trong hệ thống.

*SIP Endpoints (Điện thoại IP, Softphone)*: Các thiết bị đầu cuối VoIP (như điện thoại IP, softphone) sẽ gửi các yêu cầu SIP đến OpenSIP để thực hiện cuộc gọi hoặc đăng ký.

*SIP Trunking*: OpenSIP có thể tích hợp với các dịch vụ SIP Trunk để kết nối hệ thống tổng đài VoIP với mạng điện thoại truyền thống (PSTN). Điều này cho phép thực hiện và nhận các cuộc gọi ngoại vi.

*Backend Servers (Asterisk, FreePBX)*: Mặc dù OpenSIP chủ yếu chịu trách nhiệm xử lý SIP routing, nó có thể tích hợp với các phần mềm tổng đài khác như *Asterisk* hoặc *FreePBX* để cung cấp các tính năng bổ sung như IVR, voicemail, call queuing, và quản lý cuộc gọi.

##### **Quá trình hoạt động và xử lý cuộc gọi của OpenSIPS**

Khi một yêu cầu SIP được gửi đến OpenSIPS (chẳng hạn như một cuộc gọi VoIP), OpenSIP thực hiện quá trình xử lý theo các bước sau:

### **Bước 1: Nhận yêu cầu SIP từ các thiết bị đầu cuối**

Quá trình bắt đầu khi một thiết bị đầu cuối SIP (như điện thoại IP, softphone) gửi yêu cầu đến OpenSIPS. Đây có thể là các yêu cầu SIP cơ bản như INVITE, REGISTER, BYE, ACK. Những yêu cầu này là các tín hiệu điều khiển cuộc gọi hoặc trạng thái đăng ký giữa các thiết bị.

*Yêu cầu INVITE:* Khi một người dùng muốn bắt đầu cuộc gọi, thiết bị đầu cuối của họ gửi yêu cầu INVITE đến OpenSIPS. Yêu cầu này chứa thông tin về thiết bị gửi và đích cuộc gọi.

*Yêu cầu REGISTER:* Thiết bị đầu cuối đăng ký với OpenSIPS để nhận cuộc gọi SIP trong tương lai. Mỗi thiết bị có một SIP URI (Uniform Resource Identifier) duy nhất để nhận cuộc gọi.

*Yêu cầu BYE:* Được gửi khi một cuộc gọi kết thúc. Điều này yêu cầu OpenSIPS và các thiết bị đầu cuối ngắt kết nối cuộc gọi. OpenSIPS sẽ lắng nghe các yêu cầu SIP trên một cổng cụ thể, chẳng hạn như cổng 5060 cho UDP. Khi có yêu cầu SIP đến, OpenSIPS sẽ tiếp nhận và xử lý nó.

### **Bước 2: Xử lý yêu cầu SIP trong OpenSIPS**

Sau khi nhận yêu cầu SIP từ thiết bị đầu cuối, OpenSIPS sẽ xử lý yêu cầu này dựa trên các thông tin trong yêu cầu và cấu hình của hệ thống.

#### ***Xử lý yêu cầu REGISTER (Đăng ký SIP)***

Yêu cầu REGISTER là một phần quan trọng trong việc quản lý các thiết bị đầu cuối trong hệ thống VoIP. Khi một thiết bị gửi yêu cầu REGISTER đến OpenSIPS, mục đích là để thông báo địa chỉ IP của nó, từ đó OpenSIPS có thể nhận biết vị trí của thiết bị và định tuyến các cuộc gọi đến thiết bị đó.

OpenSIPS sẽ lưu thông tin về SIP URI và địa chỉ IP của thiết bị đầu cuối trong cơ sở dữ liệu. Điều này cho phép OpenSIPS biết được thiết bị này đang ở đâu và có thể gửi cuộc gọi đến đúng địa chỉ IP khi có cuộc gọi đến thiết bị này.

```
if (is_method("REGISTER")) {  
    save("location"); # Lưu thông tin đăng ký của thiết bị vào cơ sở dữ liệu  
}
```

Sau khi OpenSIP xử lý yêu cầu đăng ký thành công, nó sẽ trả về phản hồi *200 OK*, thông báo rằng thiết bị đã đăng ký thành công.

### ***Xử lý yêu cầu INVITE (Bắt đầu cuộc gọi)***

Yêu cầu INVITE là yêu cầu bắt đầu một cuộc gọi VoIP. Khi một người dùng muốn gọi điện cho người khác, yêu cầu INVITE sẽ được gửi từ thiết bị đầu cuối của họ đến OpenSIPS. Đây là một trong những yêu cầu SIP quan trọng nhất trong quá trình xây dựng hệ thống tổng đài VoIP.

*Kiểm tra tính hợp lệ của cuộc gọi:* OpenSIP kiểm tra các tham số trong yêu cầu INVITE, xác nhận tính hợp lệ của cuộc gọi và tiến hành định tuyến cuộc gọi đến đích.

*Định tuyến cuộc gọi:* Dựa vào cấu hình định tuyến SIP trong tệp *opensips.cfg*, OpenSIPS sẽ xác định đích của cuộc gọi (máy nhánh, máy chủ khác, hoặc hệ thống tổng đài như Asterisk). OpenSIPS sẽ tiếp tục chuyển tiếp yêu cầu INVITE tới đúng đích.

```
if (is_method("INVITE")) {  
    if (!t_relay()) {  
        sl_reply_error(); # Chuyển tiếp yêu cầu INVITE đến máy đích  
    }  
}
```

Khi yêu cầu INVITE được gửi đến máy đích và cuộc gọi được chấp nhận, End-devices sẽ phản hồi bằng *180 Ringing* (điện thoại đang đổ chuông) hoặc *200 OK* (cuộc gọi đã được kết nối).

### ***Xử lý thông điệp ACK (Xác nhận cuộc gọi)***

Khi *200 OK* đã được gửi từ thiết bị đích (người nhận cuộc gọi), yêu cầu *ACK* sẽ được gửi từ thiết bị gọi (người gọi) để xác nhận rằng cuộc gọi đã được kết nối thành công. OpenSIP chỉ đóng vai trò trong việc định tuyến các thông điệp *ACK*.

*Chuyển tiếp thông điệp ACK*: OpenSIP sẽ chuyển tiếp thông điệp *ACK* đến thiết bị đích để hoàn thành quá trình thiết lập cuộc gọi.

### ***Xử lý cuộc gọi thực tế (Media-RTP)***

Sau khi cuộc gọi đã được thiết lập, OpenSIPS không tham gia trực tiếp vào việc truyền tải âm thanh (media) giữa các thiết bị đầu cuối. Thay vào đó, *RTP (Real-time Transport Protocol)* sẽ được sử dụng để truyền tải dữ liệu âm thanh giữa các thiết bị. OpenSIPS chỉ chịu trách nhiệm xử lý tín hiệu SIP.

Mặc dù OpenSIP không chịu trách nhiệm trực tiếp về phần âm thanh, nó sẽ đảm bảo rằng các thiết bị được kết nối qua RTP để truyền tải dữ liệu âm thanh một cách chính xác.

### ***Kết thúc cuộc gọi (BYE Request)***

Khi một bên muốn kết thúc cuộc gọi, yêu cầu *BYE* sẽ được gửi. OpenSIPS sẽ nhận và xử lý yêu cầu này, sau đó thông báo cho thiết bị đầu cuối rằng cuộc gọi đã kết thúc.

*Xử lý yêu cầu BYE*: Khi OpenSIP nhận được yêu cầu *BYE*, nó sẽ ngắt kết nối cuộc gọi và gửi phản hồi *200 OK* để xác nhận kết thúc cuộc gọi.

```
if (is_method("BYE")) {  
    sl_reply_ok(); # Xác nhận kết thúc cuộc gọi  
}
```



### **Bước 3: Định tuyến SIP và các phương thức định tuyến**

Định tuyến SIP trong OpenSIP là một quá trình quan trọng giúp phân phối các yêu cầu SIP tới đúng đích. OpenSIP cung cấp khả năng định tuyến linh hoạt và có thể cấu hình các quy tắc khác nhau dựa trên yêu cầu của hệ thống.

#### ***Định tuyến theo SIP URI***

OpenSIPS có thể được cấu hình để định tuyến các yêu cầu SIP đến các điểm đích cụ thể dựa trên SIP URI (ví dụ: *sip:username@domain.com*).

#### ***Định tuyến theo các quy tắc nâng cao***

Ngoài việc định tuyến đơn giản, OpenSIPS có thể được cấu hình để định tuyến dựa trên các yếu tố khác nhau như:

*Số điện thoại:* Các cuộc gọi có thể được định tuyến dựa trên mã vùng hoặc số điện thoại.

*Trạng thái thiết bị:* OpenSIP có thể được cấu hình để kiểm tra trạng thái của các thiết bị SIP (online/offline) trước khi định tuyến cuộc gọi.

*Load balancing (Cân bằng tải):* OpenSIP có thể phân phối các cuộc gọi SIP đến các máy chủ khác nhau để giảm tải cho một máy chủ duy nhất.

*Failover (Chuyển hướng khi thất bại):* Khi một máy chủ SIP gặp sự cố, OpenSIP có thể tự động chuyển cuộc gọi đến một máy chủ dự phòng.

### **Bước 4: Quản lý cuộc gọi và thông tin**

*Cập nhật trạng thái:* Trong suốt quá trình cuộc gọi, OpenSIPS sẽ liên tục theo dõi và cập nhật trạng thái cuộc gọi (đang kết nối, đang đổ chuông, đã kết thúc).

*Ghi lại cuộc gọi:* Nếu yêu cầu ghi âm cuộc gọi được cấu hình, OpenSIPS sẽ chuyển tiếp yêu cầu ghi âm và lưu trữ âm thanh cuộc gọi vào cơ sở dữ liệu hoặc hệ thống lưu trữ.

## **Bước 5: Kết thúc cuộc gọi**

*BYE Request:* Khi một trong các bên muốn kết thúc cuộc gọi, yêu cầu BYE sẽ được gửi. OpenSIPS sẽ nhận và xử lý yêu cầu này, kết thúc cuộc gọi và thông báo cho các thiết bị đầu cuối về việc ngắt kết nối.

### **4.4. Demo sản phẩm**

Để có thể tiến hành làm mọi thứ với OPENSIPS thì điều đầu tiên là ta cần tạo ra tập lệnh (script) cấu hình cho OpenSIPS để OpenSIPS biết cách xử lý các thông điệp SIP. Điều này quyết định cách thức kết nối, quản lý cuộc gọi, định tuyến, bảo mật, và tương tác với các dịch vụ mạng.

Đầu tiên ta tạo module HTTPD với địa chỉ ip là 127.0.0.11

```
#### HTTPD module
loadmodule "httpd.so"
modparam("httpd", "port", 8888)
modparam("https", "ip", "127.0.0.1")
```

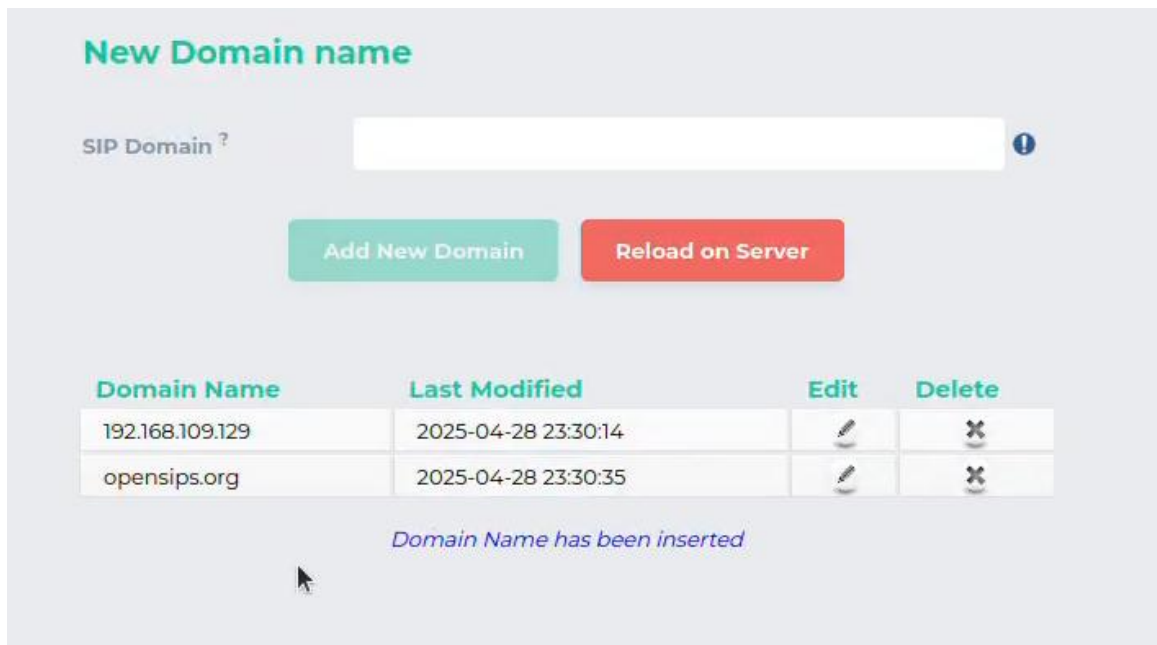
*Figure 1: Các module HTTPD*

Sau đó, ta tiến hành cài đặt socket của UDP và TCP

```
socket=udp:192.168.109.129:5060 # CUSTOMIZE ME
socket=tcp:192.168.109.129:5060 # CUSTOMIZE ME
```

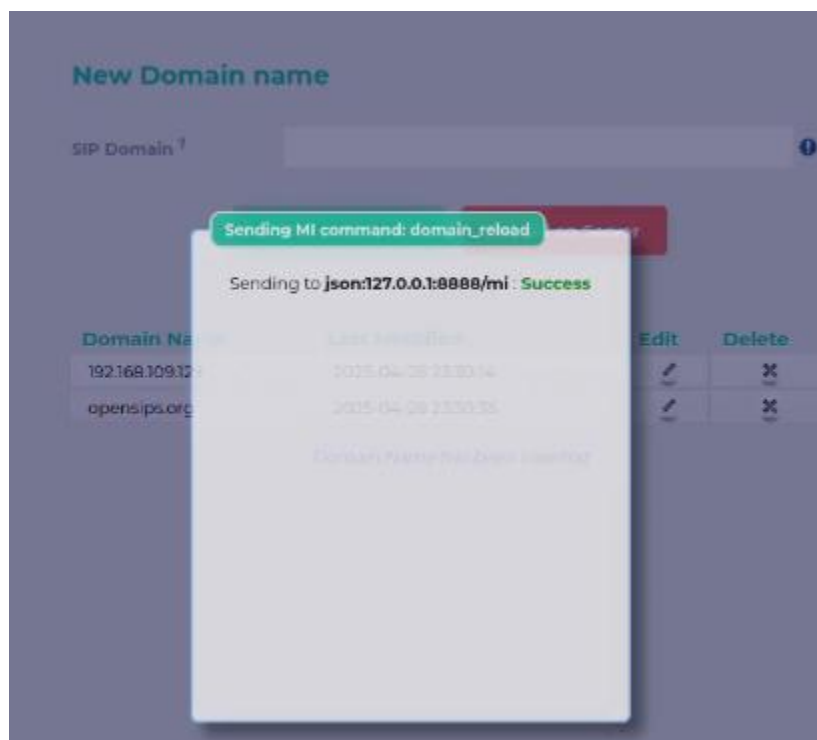
*Figure 2: Socket UDP và TCP*

Tạo 2 địa chỉ SIP Domain



*Figure 3: Tạo domain*

Sau đó thì tiến hành reload sever

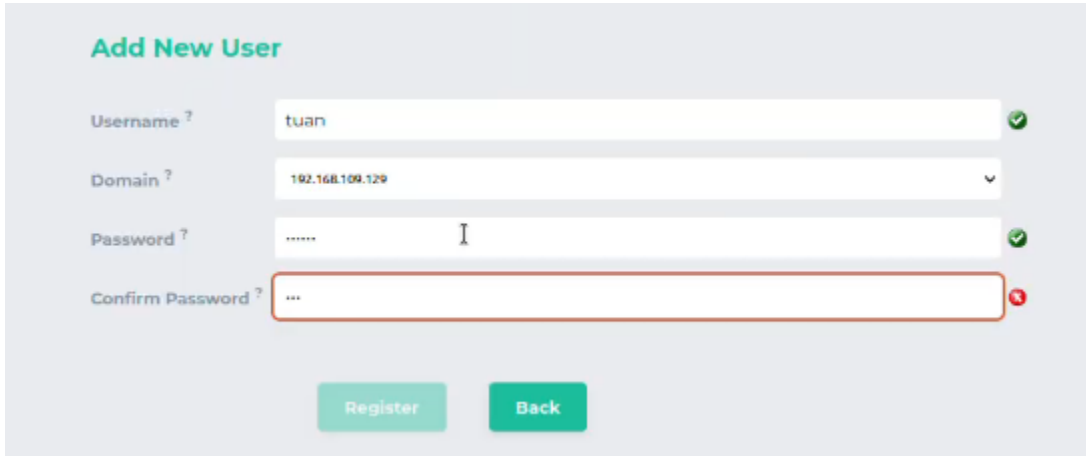


*Figure 4: Server được reload*

Và sau khi hoàn thành việc tạo ra tập lệnh (script) cấu hình cho OpenSIPS, chúng ta bắt đầu tiến hành các bước để tạo ra thực hiện cuộc gọi đầu tiên giữa hai thiết bị (clients) thông qua OpenSIPS làm máy chủ trung gian (SIP Proxy Server).

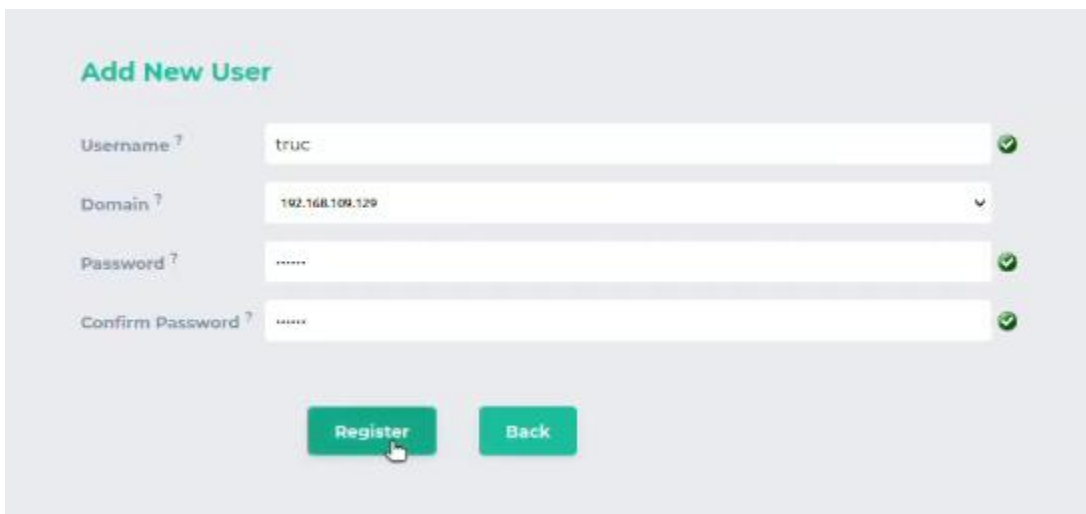
Đầu tiên, chúng ta cần đăng ký SIP (SIP Registration):

User đầu tiên đặt tên là “tuan”

The image shows a web form titled "Add New User". It contains four input fields: "Username" with the value "tuan", "Domain" with a dropdown menu showing "192.168.109.129", "Password" with masked characters "\*\*\*\*\*", and "Confirm Password" with masked characters "\*\*\*". Each field has a small green checkmark icon to its right, except for the "Confirm Password" field which has a red X icon. Below the fields are two buttons: "Register" and "Back".

*Figure 5: User tuan*

User thứ 2 đặt tên là “truc”

The image shows the same "Add New User" web form. The "Username" field now contains the value "truc". The "Domain" dropdown still shows "192.168.109.129". The "Password" and "Confirm Password" fields are masked with "\*\*\*\*\*". All four fields now have a small green checkmark icon to their right. The "Register" and "Back" buttons are still present at the bottom.

*Figure 6: User truc*

Ta có thể thấy các thông tin của các user như sau:

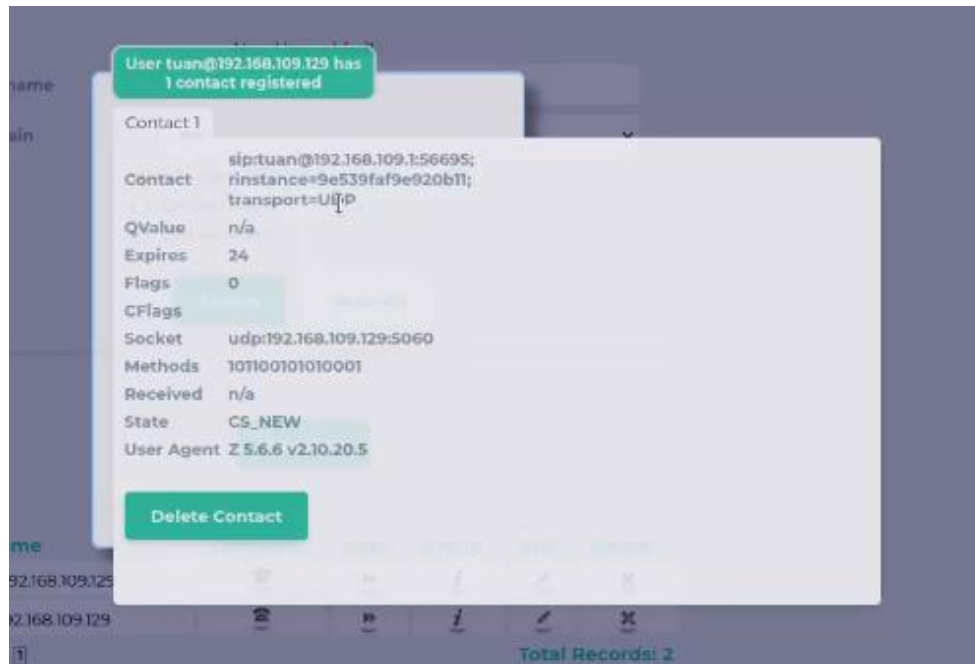


Figure 7: Thông tin các user

Sau khi có các user rồi thì việc tiếp theo chúng ta cần làm là đăng nhập 2 user vào Zoiper5. Sau đó chúng ta vào Account => Advanced => set up Don't use Stun

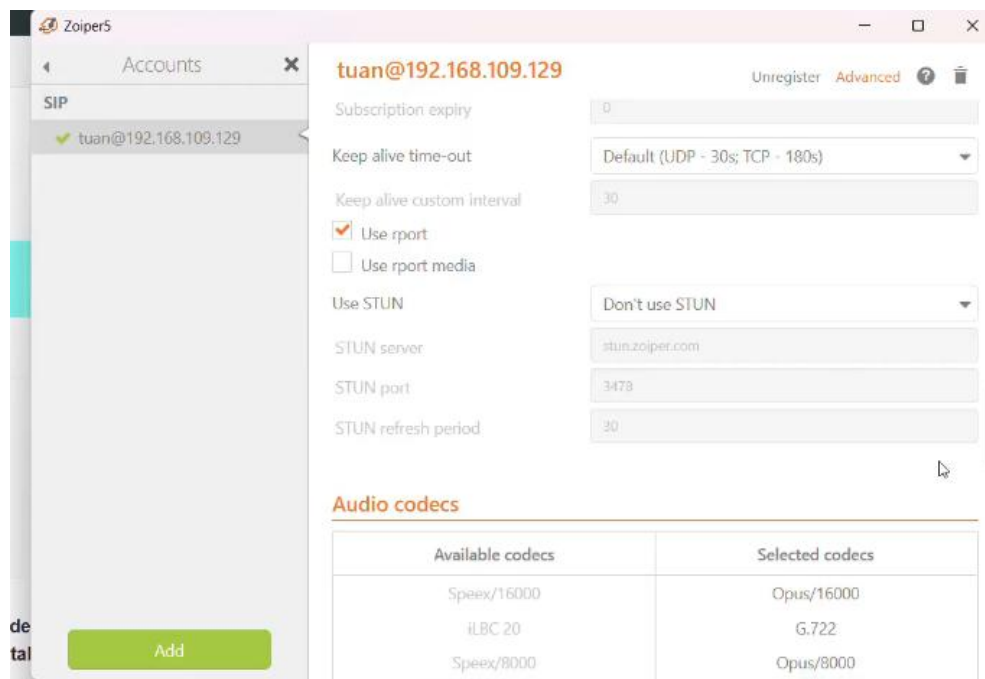
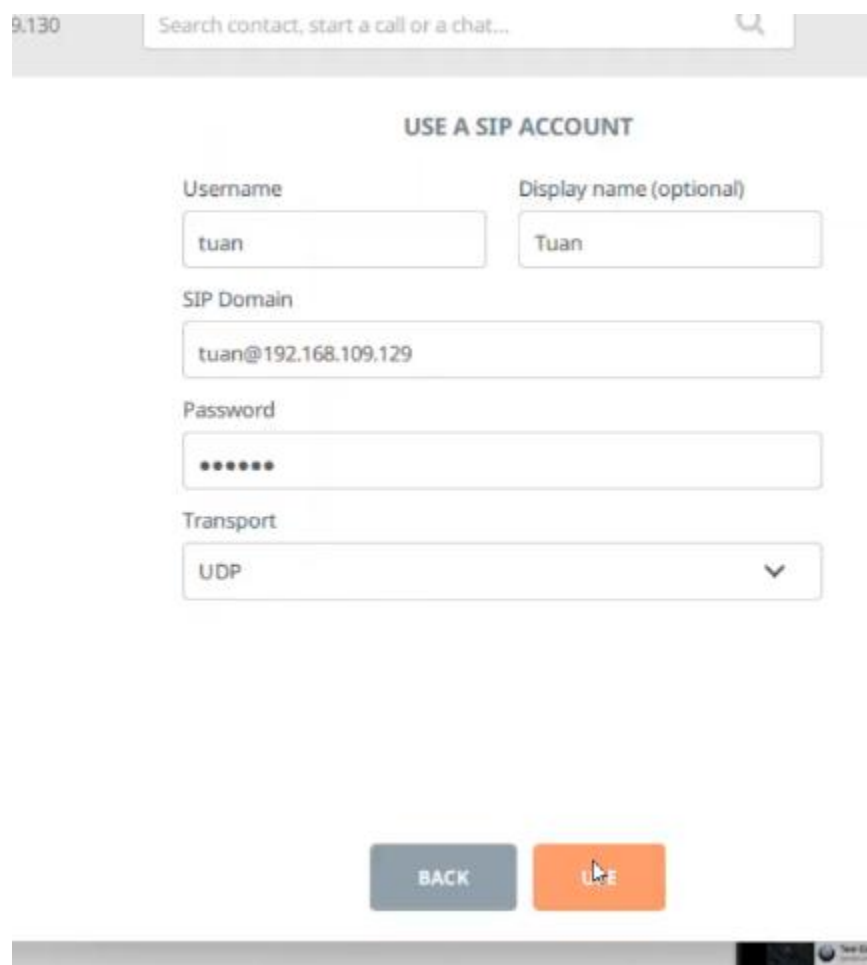


Figure 8: Đăng nhập các user vào Zoiper5

Khi dùng trong mạng kết nối thế giới (global) thì STUN sẽ rất tốt những dùng trong các mạng nội bộ như OPENSIPS thì STUN không phải ý kiến hay vì mọi địa chỉ IP đều là **private IP** và có thể định tuyến trực tiếp. STUN lúc này có thể gây lỗi định tuyến, khi nó cố gắng lấy địa chỉ IP public không tồn tại.

Sau khi cài đặt xong ta tiến hành Delete contact của cả 2 user trên OPENSIPS rồi Unregister và Register tài khoản lại một lần nữa trên Zoiper5.

Tiếp theo, ta tiến hành cài Blink Sip Client. Mở Linphone và tiến hành điền các thông tin user.



9.130 Search contact, start a call or a chat...

**USE A SIP ACCOUNT**

Username:

Display name (optional):

SIP Domain:

Password:

Transport:

*Figure 9: Blink Sip Client*

**USE A SIP ACCOUNT**

Username	Display name (optional)
<input type="text" value="truc"/>	<input type="text" value="Truc"/>

SIP Domain

Password

Transport

*Figure 10: Tương tự với user còn lại*

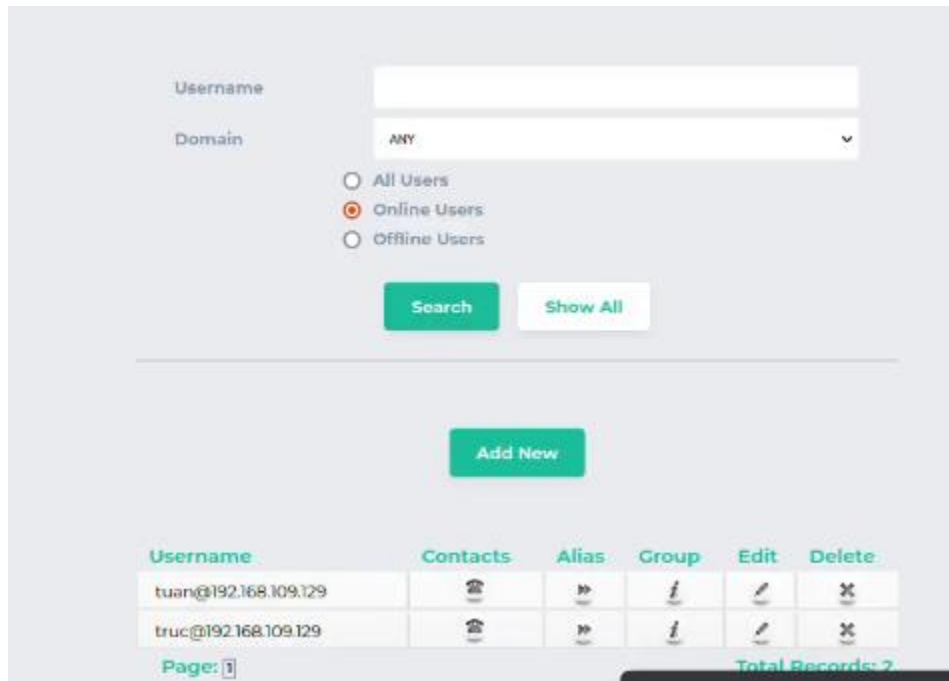
Presence status

Active account

<input type="checkbox"/>	"user2" <sip:user2@192.168.109.130>
<input checked="" type="checkbox"/>	"Tuan" <sip:tuan@192.168.109.129>
<input checked="" type="checkbox"/>	"Truc" <sip:truc@192.168.109.129>

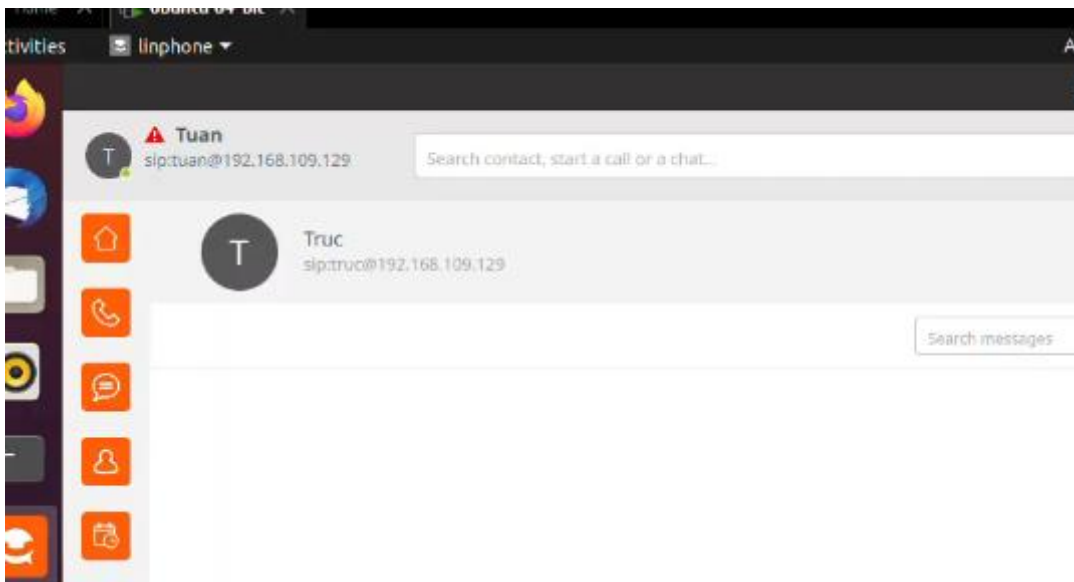
*Figure 11: Active các user*

Trong OpenSIPS ta sẽ check lại các Online user xem 2 tài khoản đã online chưa



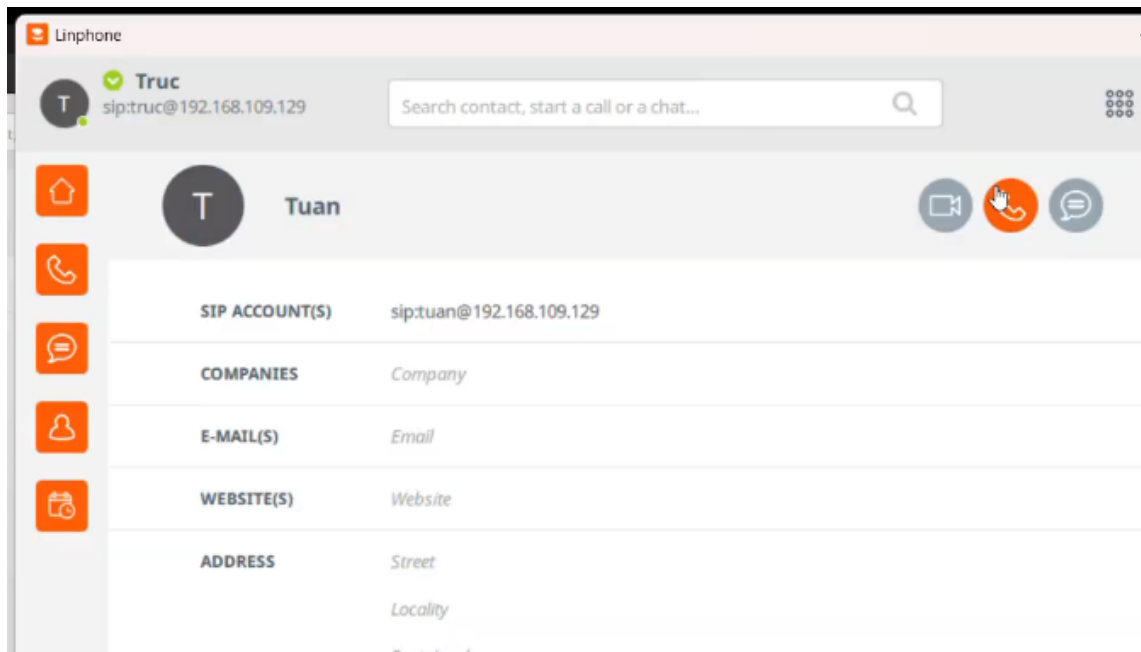
*Figure 12: Check status các user*

Sau đó, ta tiến hành thêm data của 2 users chéo vào tài khoản Linphone của cả 2.



*Figure 13: Kết nối chéo trên Linphone*





*Figure 14: Kiểm tra lại trạng thái kết nối*

Sau khi thêm xong thì ta sẽ tiến hành cuộc gọi để kiểm tra.

User “truc” sẽ tiến hành gọi user “tuan” trong liên kết Linphone. Sau đó tiến hành kiểm tra độ ổn định của đường truyền và giao tiếp giữa 2 users.

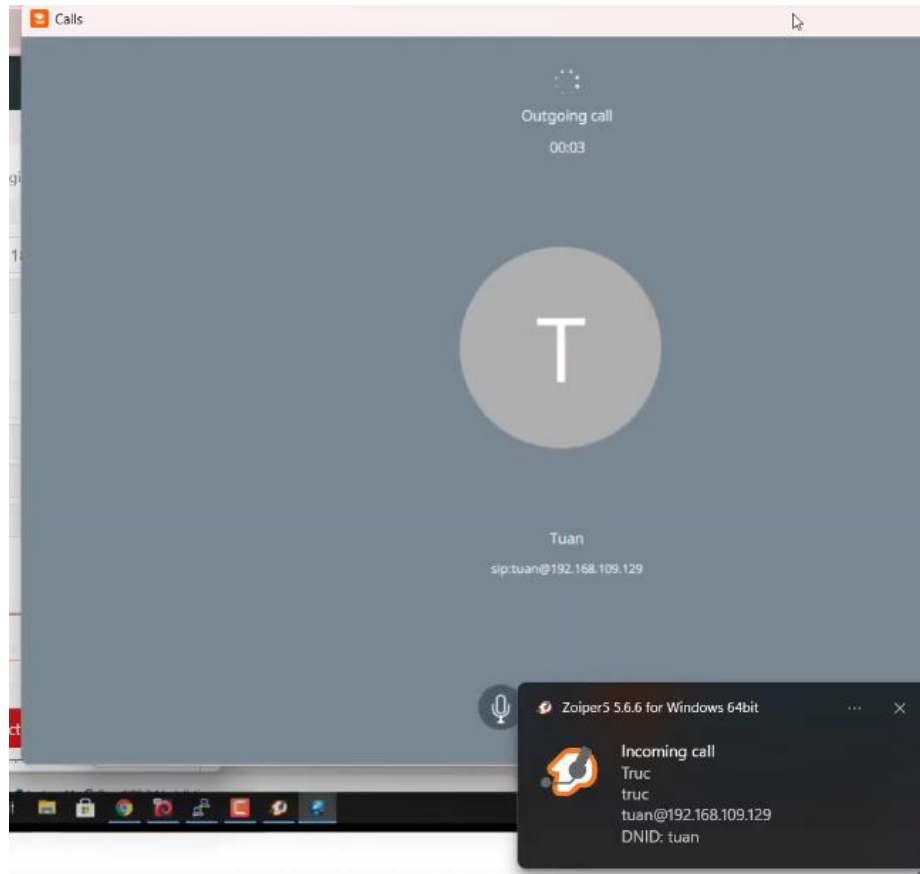


Figure 15: Kết nối thành công

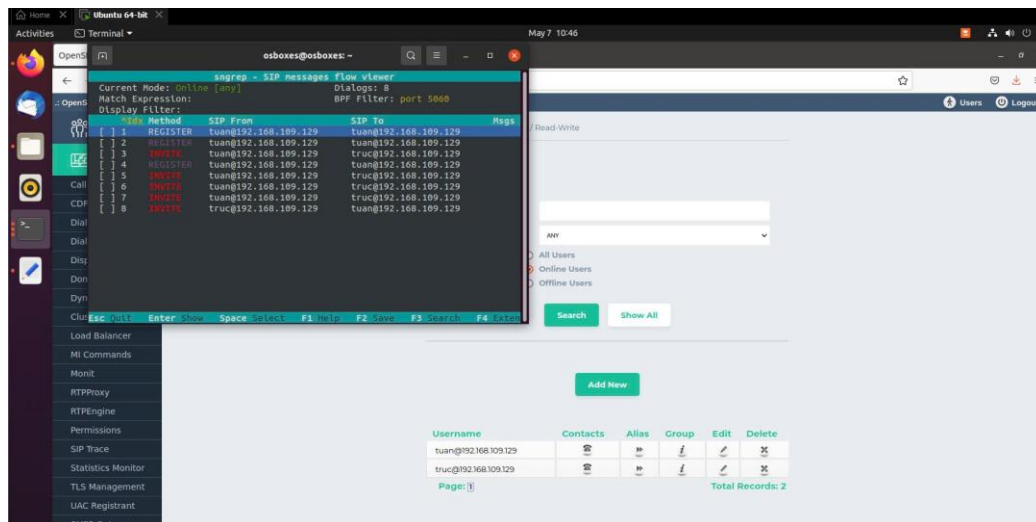


Figure 16: Capture các gói tin trên tổng đài

Phần này hiển thị kết quả capture gói tin SIP theo thời gian thực, với các cột thông tin sau:

**Current Mode:** Đang ở chế độ "SIP Messages" (hiển thị các bản tin SIP).

**Match Expression:** Bộ lọc hiện tại là port 5060, nghĩa là chỉ capture lưu lượng trên cổng SIP mặc định (5060).

**SIP From:** Địa chỉ nguồn của các bản tin SIP (ví dụ: [tuan@192.168.109.120](mailto:tuan@192.168.109.120)).

**SIP To:** Địa chỉ đích của các bản tin SIP (ví dụ: [tuan@192.168.109.120](mailto:tuan@192.168.109.120) hoặc [truc@192.168.109.120](mailto:truc@192.168.109.120)).

**Call-ID:** Định danh duy nhất cho mỗi cuộc gọi (ví dụ: ZW2I7t...).

**Method:** Phương thức SIP được sử dụng (ví dụ: REGISTER, INVITE, 200 OK).

**Status:** Mã trạng thái hoặc thông tin bổ sung (ví dụ: 200 OK cho phản hồi thành công).

### **Chi tiết các dòng log:**

Các bản tin được liệt kê theo thứ tự thời gian, với các cuộc gọi từ cùng một Call-ID (như ZW2I7t...) cho thấy một phiên SIP đang hoạt động.

Ví dụ:

- REGISTER từ [tuan@192.168.109.120](mailto:tuan@192.168.109.120) đến 192.168.109.120 cho thấy quá trình đăng ký người dùng với máy chủ SIP.
- 200 OK là phản hồi thành công từ máy chủ.
- Nhiều bản tin liên quan đến [truc@192.168.109.120](mailto:truc@192.168.109.120) có thể chỉ một cuộc gọi hoặc phiên khác.

Các bản tin REGISTER và 200 OK cho thấy quá trình đăng ký người dùng thành công, trong khi phân quản lý người dùng hỗ trợ theo dõi danh sách tham gia.

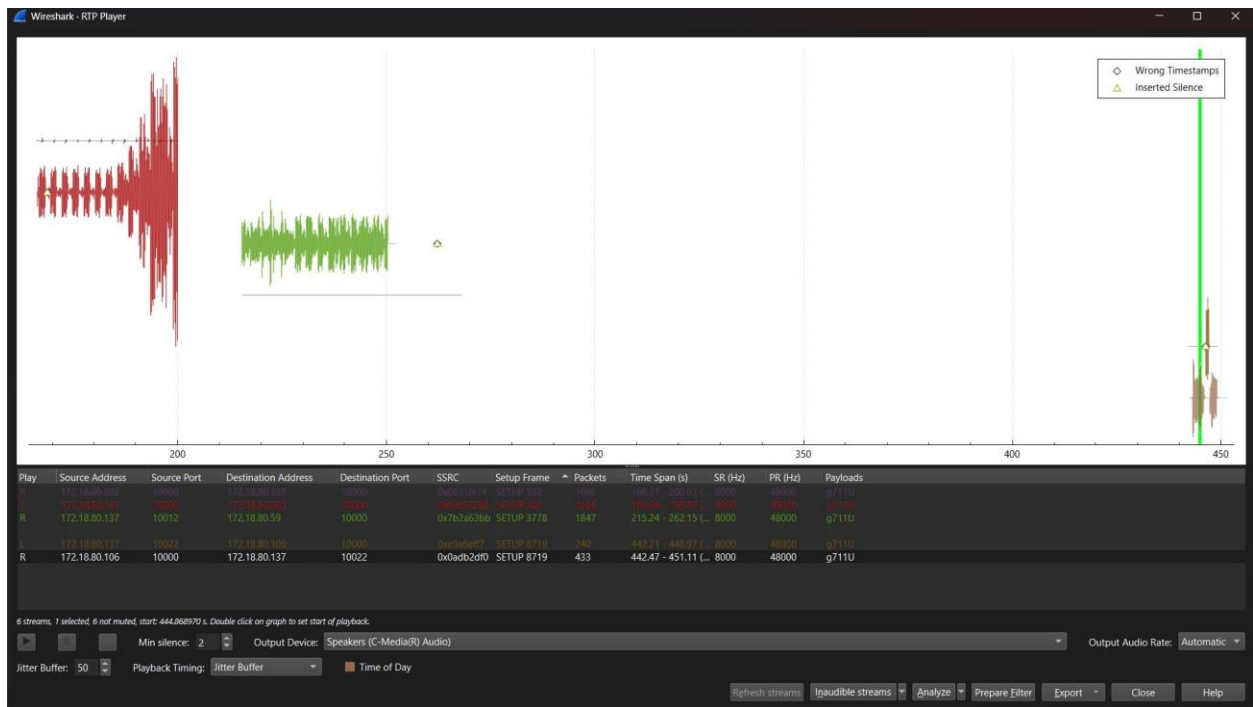


Figure 17: Flowdata của gói thoại

Bên cạnh việc có thể capture các gói tin và thể hiện chúng, một phần mềm capture các gói tin qua kết nối mạng có thể dùng là Wireshark. Phần mềm này có thể capture nhiều hơn các gói tin là flowdata (voice flow) từ các mạng kết nối. Từ đó ta có thể dễ dàng xem được các luồng dữ liệu thoại được gửi và nhận giữa các users.



Figure 18: Sơ đồ quy trình chuyển gói tin với Cancel

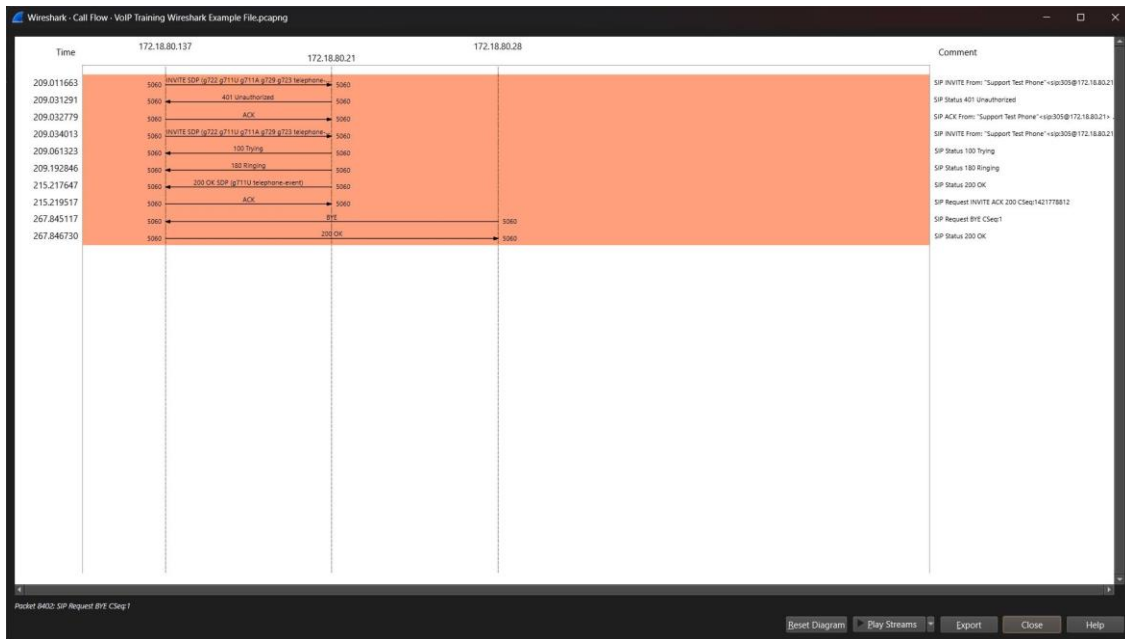


Figure 19: Quy trình chuyển gói tin khi nhắc máy

Hình ảnh ta lấy mẫu hiển thị một luồng cuộc gọi VoIP (SIP Flow) được capture bằng Wireshark, mô tả quá trình thiết lập và kết thúc một cuộc gọi giữa hai thiết bị có địa chỉ IP 172.16.80.127 và 172.16.80.21. Luồng bắt đầu lúc 17:26:46 với bản tin INVITE từ 172.16.80.127 gửi đến 172.16.80.21, tiếp theo là các phản hồi như 100 Trying và 180 Ringing, cho thấy thiết bị đích đang rung chuông. Sau đó, 172.16.80.21 gửi 200 OK để chấp nhận cuộc gọi, kèm theo SDP (Session Description Protocol) để thiết lập luồng RTP (truyền tải âm thanh). Các bản tin ACK xác nhận kết nối, và cuộc gọi kết thúc bằng bản tin BYE từ 172.16.80.127, được 172.16.80.21 xác nhận với 200 OK. Phần bình luận bên phải ghi chú trạng thái cuộc gọi, như "SIP INVITE from 'Support Phone'" và "SIP STATUS 200 OK", hỗ trợ việc phân tích chi tiết.

Bên cạnh việc không chỉ thực hiện các kênh thoại VoIP trong mạng local thông qua intranet, ta còn có thể thực hiện các luồng kết nối với các thiết bị thoại trong mạng công cộng PSTN. Bằng các cách thiết lập sau:

```
opensips@opensips:~$
}

if ($rU==NULL) {
    # request with no Username is ROR
    send_reply(404,"Address Incomplete");
    exit;
}

# apply DB based aliases
alias_db_lookup("dbaliases");

# apply transformations from dialplan table
dp_translate( 0, "$rU", $rU);

if ($rU=~"^[1-9]{0-9}+$") {
    strip(1)
    if (!do_routing(0)) {
        send_reply(500,"No PSTN Route found");
        exit;
    }

    route(relay);
    exit;
}

# do lookup with method filtering
if (!lookup("location","m")) {
    if (!db_does_uri_exist("$ru","subscriber")) {
        send_reply(420,"Bad Extension");
        exit;
    }
}

t_reply(404, "Not Found");
```

Figure 20: Thiết lập cài đặt PSTN

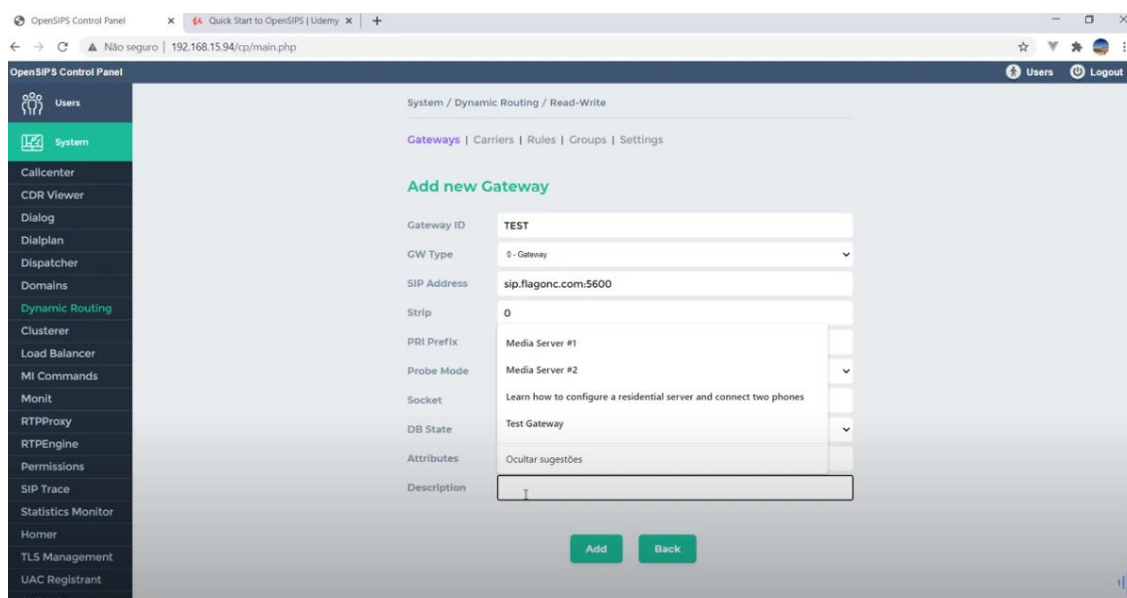


Figure 21: Cấu hình thông tin cho thiết bị trong mạng PSTN

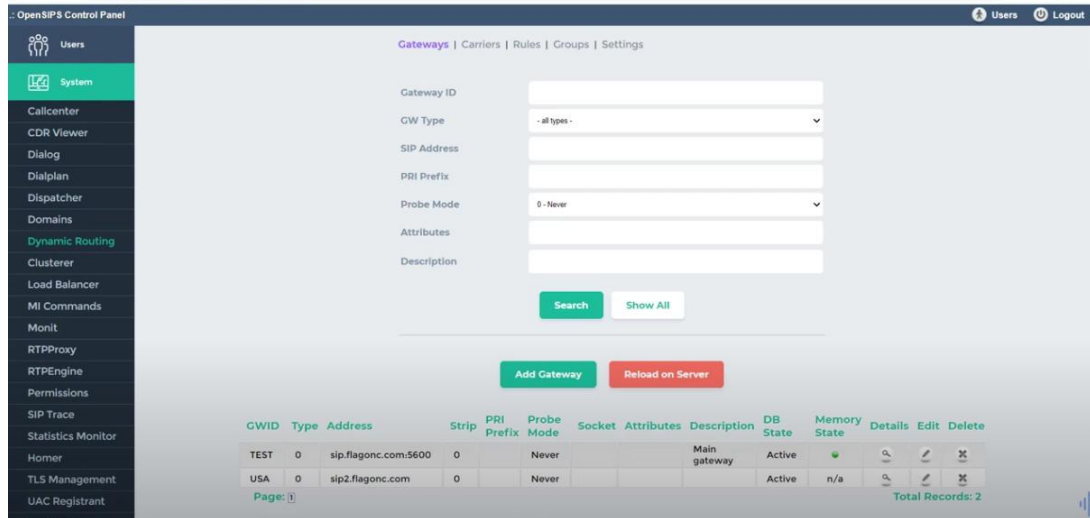


Figure 22: Các thiết bị sau thiết lập

Sau khi thiết lập các thông tin cho các thiết bị thoại cho mạng: Cả hai gateway đều trở đến cùng một domain (flagcon.com), được giả lập là một nhà cung cấp SIP hoặc máy chủ trung gian. Gateway TEST sử dụng cổng cụ thể (5060), trong khi USA không chỉ định cổng (mặc định 5060). Cả hai gateway đều đang hoạt động, cho thấy chúng sẵn sàng xử lý lưu lượng cuộc gọi. Giao diện cho phép thêm gateway mới (Add Gateway), tải lại cấu hình (Reload on Server), và chỉnh sửa/xóa gateway hiện có.

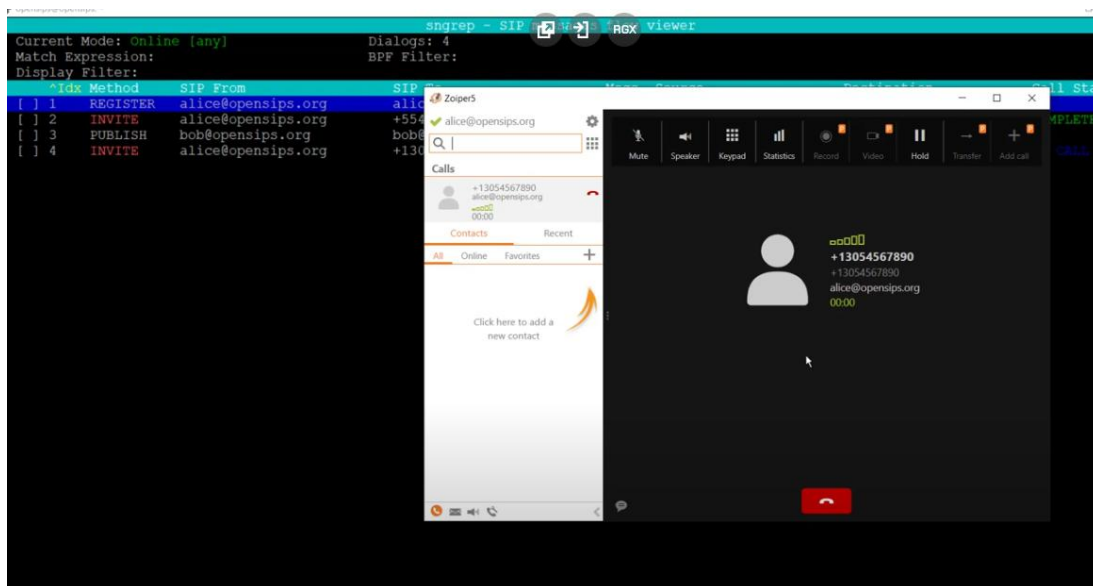


Figure 23: Capture các gói tin giữa 2 gateways

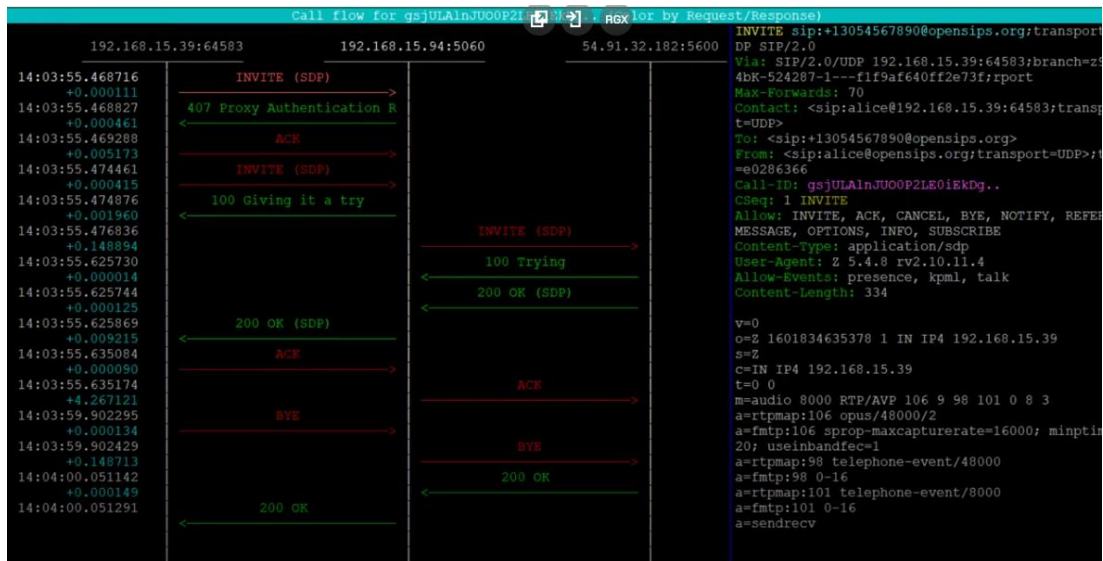
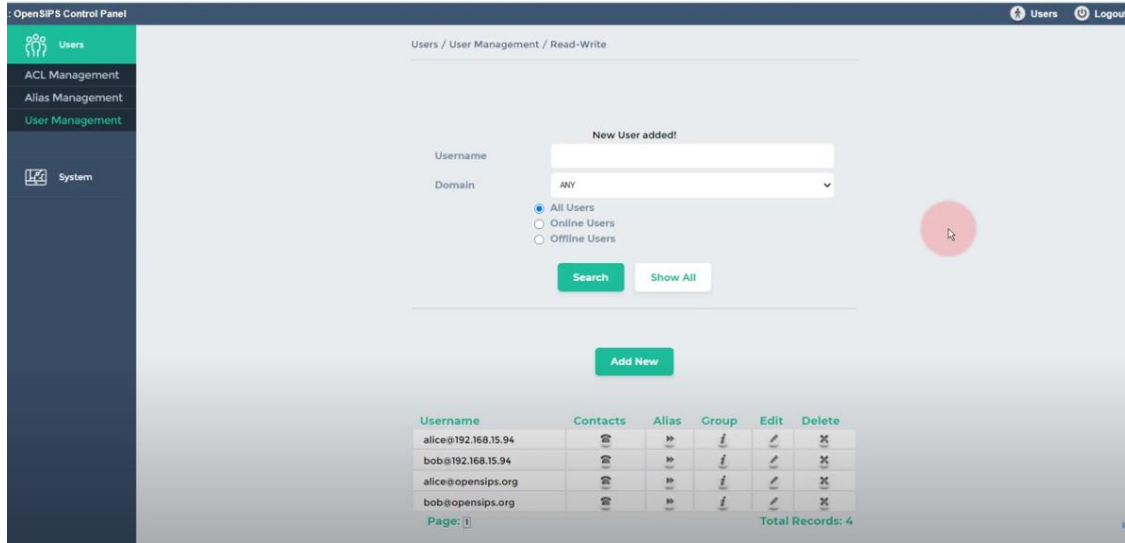


Figure 24: Các gói tin cụ thể của call flow

SIP Call Flow được capture và phân tích, minh họa quá trình thiết lập và kết thúc một cuộc gọi giữa hai địa chỉ IP: 192.168.15.39:64583 và 54.91.32.182:5600, với OpenSIPS (opensips.org) đóng vai trò trung gian. Luồng bắt đầu lúc 14:03:55 với bản tin INVITE (chứa SDP) từ 192.168.15.39, tiếp theo là phản hồi 407 Proxy Authentication Required, dẫn đến việc gửi ACK và INVITE lặp lại sau khi xác thực. Máy chủ OpenSIPS phản hồi 100 Trying, sau đó gửi INVITE đến 54.91.32.182, nhận lại 100 Trying và 200 OK (chứa SDP) từ đích. Cuộc gọi được xác nhận bằng ACK, và sau một thời gian, bản tin BYE được gửi từ 192.168.15.39 để kết thúc, nhận phản hồi 200 OK. Bên phải hình ảnh hiển thị chi tiết bản tin INVITE, bao gồm các header như From, To, Contact, và nội dung SDP với thông tin codec (Opus/48000) và luồng RTP, cung cấp cái nhìn toàn diện về quá trình giao tiếp SIP.

Ngoài ra, ta còn có thể cấu hình sâu hơn với việc cài đặt miền (domain) và biệt danh (Aliases) nhằm xác minh được khả năng nhận diện đa dạng và dễ dàng đăng ký các user của OpenSIPS hay giao thức SIP.





*Figure 25: Nhiều thành phần được cài đặt*

Về mô phỏng tuy đã thành công và các kết quả nhận được là khá ổn cho đến hiện tại, các đường truyền được đảm bảo ổn định, ít nhiễu, độ trễ chuyển tiếp thoại giữa các kênh thoại chưa nhiều. Tuy nhiên trong một số trường hợp, do vấn đề máy chủ không đủ mạnh nên không thể cùng lúc thực hiện quản lý được nhiều cuộc gọi. Bên cạnh đó, ta chưa nhận thấy được rõ ràng lỗi có thể xảy ra khi có thiết bị thoại thứ ba xen vào cuộc trò chuyện đang được thiết lập giữa 2 users trước đó. Ngoài ra, do vấn đề chưa có sự bảo mật cao trên đường truyền và các gateway nên sự xâm nhập bất hợp pháp có thể dễ dàng gây nên tổn thất hoặc mất mát dữ liệu của người dùng. Do đó, hệ thống này cần thời gian để có thể nghiên cứu và phát triển sâu hơn để có thể đảm bảo tính trọn vẹn và an toàn bảo mật thông tin tuyệt đối cho người dùng.

## 5. KẾT LUẬN CHUNG

VoIP là một bước tiến quan trọng trong công nghệ truyền thông, mang lại sự tiện lợi và tiết kiệm chi phí đáng kể so với hệ thống điện thoại truyền thống. Quá trình mô phỏng tổng đài VoIP sử dụng OpenSIPS đã chứng minh hiệu quả của nền tảng này trong việc quản lý và định tuyến các cuộc gọi VoIP. OpenSIPS, với vai trò là một SIP Proxy Server mạnh mẽ, đã xử lý thành công các bản tin SIP, từ việc đăng ký người dùng (REGISTER) đến thiết lập và kết thúc cuộc gọi (INVITE, BYE), đảm bảo tính ổn định và hiệu suất cao. Việc tích hợp các công cụ như sngrep và Wireshark đã hỗ trợ đắc lực trong việc capture và phân tích lưu lượng VoIP, giúp phát hiện và khắc phục sự cố nhanh chóng. Mô phỏng này không chỉ khẳng định khả năng của OpenSIPS trong việc xây dựng một hệ thống tổng đài VoIP linh hoạt, mà còn mở ra tiềm năng ứng dụng thực tiễn trong các hệ thống viễn thông quy mô lớn. Tuy nhiên, để tối ưu hơn, cần cân nhắc các yếu tố như bảo mật (TLS/SRTP) và hiệu suất khi triển khai trên lưu lượng cao. Tuy nhiên, để tận dụng tối đa lợi ích của VoIP, cần đảm bảo chất lượng mạng tốt và các biện pháp bảo mật phù hợp. Với sự phát triển không ngừng của Internet và công nghệ, VoIP hứa hẹn sẽ tiếp tục đóng vai trò quan trọng trong tương lai của truyền thông toàn cầu.