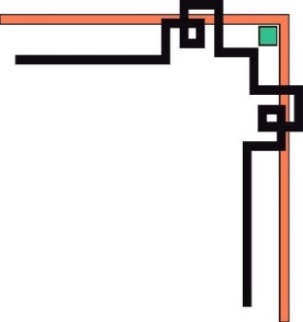
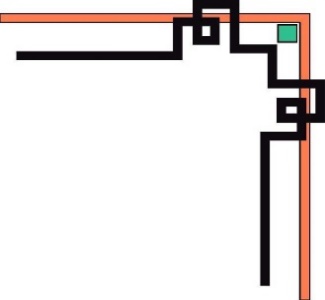
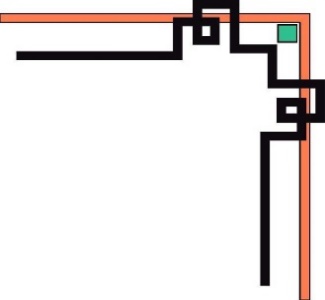
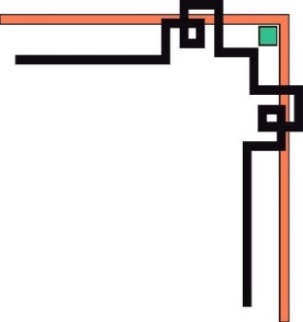
**ĐẠI HỌC ĐÀ NẴNG**



**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**VÀ TRUYỀN THÔNG VIỆT - HÀN**

**KHOA KĨ THUẬT MÁY TÍNH**

**🙢🙠🕮🙢🙠**

****

**BÁO CÁO ĐỒ ÁN LẬP TRÌNH MẠNG**

**ĐỀ TÀI: SỬ DỤNG KĨ THUẬT WEBRTC PHÁT TRIỂN ỨNG DỤNG TRUYỂN DỮ LIỆU THEO MÔ HÌNH PEER-TO-PEER**

Giáo viên hướng dẫn : THS. NGUYỄN THANH CẨM

Sinh viên thực hiện : lê đức mạnh

Lớp : 20CE

Mã sinh viên : 20CE008

*Đà Nẵng, ngày 30 tháng 10 năm 2022*

**ĐẠI HỌC ĐÀ NẴNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**VÀ TRUYỀN THÔNG VIỆT - HÀN**

**KHOA KĨ THUẬT MÁY TÍNH**

****

**BÁO CÁO ĐỒ ÁN LẬP TRÌNH MẠNG**

**ĐỀ TÀI: SỬ DỤNG KĨ THUẬT WEBRTC PHÁT TRIỂN ỨNG DỤNG TRUYỂN DỮ LIỆU THEO MÔ HÌNH PEER-TO-PEER**

Giáo viên hướng dẫn : THS. NGUYỄN THANH CẨM

Sinh viên thực hiện : lê đức mạnh

Lớp : 20CE

Mã sinh viên : 20CE008

*Đà Nẵng, ngày 30 tháng 10 năm 2022*

**LỜI NÓI ĐẦU**

Dưới sự phát triển không ngừng của ngành công nghệ thông tin và trong xu thể tiến đến cuộc cách mạng công nghiệp 4.0. Để đáp ứng nhu cầu việc đơn giản hóa trong việc giao tiếp, một giải pháp hoàn thiện đem đến cho con người lợi ích về truyền thông và liên lạc. Công nghệ WebRTC là một công nghệ đang rất phổ biến hiện nay. WebRTC được sử dụng để truyền tải video, âm thanh, gửi dữ liệu theo thời gian thực của hai thiết bị mà không qua bên thứ ba hay cài đặt thêm plugin, phần mềm. Hơn nữa, nó cũng được ứng dụng để tạo ra các tựa game mà người dùng chỉ cần sử dụng trình duyệt có WebRTC để chơi chứ không cần cài đặt rườm rà.

Được sự quan tâm giúp đỡ của các thầy cô giáo viên hướng dẫn là ThS. Nguyễn Thanh Cẩm, chúng em xin cảm ơn Khoa Khoa Học Máy Tính nói chung và xin cảm ơn thầy Cẩm nói riêng đã tạo điều kiện cho chúng em có cơ hội tìm hiểu cũng như hoàn thành bản tiểu luận, báo cáo về WebRTC.

**LỜI CẢM ƠN**

Em xin gửi lời cảm ơn sâu sắc đến các thầy cô trường Đại học Công nghệ thông tin và Truyền thông Việt-Hàn đã hết lòng truyền đạt cho chúng em những kiến thức bổ ích trong suốt thời gian qua, cũng như đã tạo điều kiện cho chúng em được phát huy ý tưởng.

Đặc biệt, chúng em xin gửi lời cảm ơn đến thầy ThS. Nguyễn Thanh Cẩm đã tận tình hướng, truyền đạt kiến thức và chỉ bảo cho chúng em trong suốt thời gian thực hiện đề tài.

Mặc dù em đã rất cố gắng, song chắc chắn báo cáo sẽ còn rất nhiều thiếu sót, em rất mong nhận được sự hướng dẫn thêm của quý thầy cô, các anh chị và các bạn.

Đà nẵng, ngày…. tháng …. năm 2022

Sinh viên thực hiện

Lê Đức Mạnh

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

Đà nẵng, ngày…. tháng …. năm 2022

Giảng viên hướng dẫn

ThS. Nguyễn Thanh Cẩm

**MỤC LỤC**

[**CHƯƠNG 1 : CƠ SỞ LÍ THUYẾT** 7](#_Toc121263311)

[**1.** **Sơ lược lập trình mạng** 7](#_Toc121263312)

[**2.** **Sơ lược về WEBRTC và quá trình phát triển** 7](#_Toc121263313)

[ WebRTC 7](#_Toc121263314)

[ Quá trình phát triển 7](#_Toc121263315)

[**3.** **Sơ lược về mạng ngang hàng** 8](#_Toc121263316)

[**a)** **Giới thiệu** 8](#_Toc121263317)

[**b)** **Định nghĩa** 8](#_Toc121263318)

[**c)** **So sánh mô hình Client-Server và mô hình Peer-to-Peer** 9](#_Toc121263319)

[**CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG** 10](#_Toc121263320)

[**1.** **Mô hình tổng quan của WEBRTC:** 10](#_Toc121263321)

[a) Các giao thức được sử dụng trong WEBRTC 10](#_Toc121263322)

[b) Các thành phần API của WEBRTC 13](#_Toc121263323)

[**2.** **Biểu đồ usecase** 15](#_Toc121263324)

[**3.** **Biểu đồ hoạt động** 15](#_Toc121263325)

[**CHƯƠNG 3. TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ** 16](#_Toc121263326)

[**1.** **Giới thiệu về website dùng để chia sẻ file** 16](#_Toc121263327)

[**2.** **Một số thành phần của ứng dụng** 17](#_Toc121263328)

[a) Kéo thả file 17](#_Toc121263329)

[**KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN** 20](#_Toc121263330)

[**1.** **Kết luận:** 20](#_Toc121263331)

[**2.** **Hạn chế** 20](#_Toc121263332)

[**3.** **Hướng phát triển** 20](#_Toc121263333)

[**TÀI LIỆU THAM KHẢO:** 21](#_Toc121263334)

**DANH SÁCH HÌNH ẢNH :**

[Hình 1 - Mô hình mạng ngang hàng 8](#_Toc121295537)

[Hình 2 - Mô hình tổng quan của WEBRTC 10](#_Toc121295538)

[Hình 3 - Minh họa Traversal Using Relays 11](#_Toc121295539)

[Hình 4 - Minh họa Session Traversal Utilities for NAT 12](#_Toc121295540)

[Hình 5 - Minh họa việc cấp quyền truy cập MediaStream APIs 13](#_Toc121295541)

[Hình 6 - Mô phỏng việc truyền dữ liệu sử dụng DataChannel 14](#_Toc121295542)

[Hình 7 - Biểu đồ usecase 15](#_Toc121295543)

[Hình 8 - Biểu đồ hoạt động hệ thống chia sẻ file 16](#_Toc121295544)

[Hình 9 - Trang chủ chia sẻ file 17](#_Toc121295545)

[Hình 10 - Drop và kéo thả file 17](#_Toc121295546)

[Hình 11 - Đường dẫn room 18](#_Toc121295547)

**CHƯƠNG 1 : CƠ SỞ LÍ THUYẾT**

1. **Sơ lược lập trình mạng**

Lập trình mạng là một công việc nhằm phát triển các ứng dụng doanh nghiệp hoạt động trên mạng Internet. Theo đó, người lập trình cần viết ra một chương trình mạng để chạy những chương trình trên các máy tính khác nhau, giúp các nút mạng ấy chia sẻ dữ liệu, truyền tin an toàn, hiệu quả. Sử dụng các công cụ lập trình để tạo ra một chương trình giúp các máy tính giao tiếp với nhau bao gồm ngôn ngữ lập trình mạng, kiến thức về phân tích thiết kế hệ thống, kiến thức hệ thống mạng, mô hình xây dựng chương trình ứng dụng mạng, kiến thức về cơ sở dữ liệu.

1. **Sơ lược về WEBRTC và quá trình phát triển**
2. WebRTC

WebRTC đã quá phổ biến trong lập trình các ứng dụng thời gian thực và video call. Lần này mình sẽ viết một bộ series về WebRTC. WebRTC (Web Real-Time Communications) là một tập hợp các hàm lập trình dùng cho việc liên lạc thời gian thực bằng video, âm thanh cũng như các loại dữ liệu khác. WebRTC có thể giúp chúng ta gọi điện video ngay trong trình duyệt mà không cần đăng kí tài khoản, cũng không cần cài thêm plugin gì phức tạp, ngoài ra chúng còn được xài để phát triển game chơi trực tiếp trong trình duyệt và rất nhiều loại ứng dụng khác.

1. Quá trình phát triển

WebRTC bắt đầu từ lúc Google muốn xây dựng một chuẩn nhằm thực hiện thời gian thực trên tất cả các trình duyệt.Vì vậy trong năm 2010 Google đa mua lại Global IP Solutions (GIPS), đây là một công ty phần mềm về VoIP và video hội họp (công ty này cũng phát triển nhiều thành phần được yêu cầu cho RTC (Real Time Communication) như codecs và các công nghệ echo cancellation ).Google đưa GIPS thành một mã nguồn mở và bắt đầu tham gia vào IETF và W3C. Tháng 5/2011 Google bắt đầu đưa ra một dự án mã nguồn mở cho phép thực hiện cái ứng dụng thời gian thực cho trình duyệt được biết tới như WebRTC.Nhưng công ty hỗ trợ WebRTC tích cực nhất gồm có Google, Mozilla, Opera. Sự phát triển của WebRTC qua các năm gần đây trên các trình duyệt có thể kể đến như:

* 27/10/2011: WebRTC bắt đầu được công bố.
* 11/2011: WebRTC bắt đầu hỗ phần trên Chrome 23.
* 1/2013: WebRTC hỗ trợ trên firefox.
* 7/2013: Phiên bản beta của Chrome 29 trên Android hỗ trợ WebRTC.
* 10/2013: WebRTC bắt đầu hỗ trợ trên phiên bản Opera beta.
* 2/2015: WebRTC 1.0 working draft chính thức công bố.

1. **Sơ lược về mạng ngang hàng**
2. Giới thiệu

Mạng ngang hàng (P2P) hay còn gọi là Peer-To-Peer bắt đầu xuất hiện từ năm 1999 và đã thu hút sự quan tâm của giới công nghệ thông tin trong những năm gần đây. Đặc biệt, việc áp dụng các mô hình P2P trong việc xây dựng những ứng dụng chia sẻ file (file sharing), video call, điện thoại trên nền tảng Intern(Internet-based telephony) dã đạt được nhiều thành công. Hiện nay, các ứng dụng P2P chiếm khoảng 50% (thậm chí 75%) băng thông trên Internet. Các ứng dụng của kiểu mạng này như là: Napster,Skype, BitTorrent, v.v..

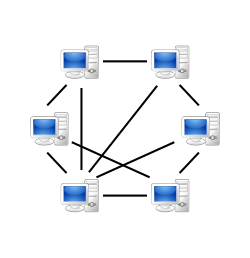
1. Định nghĩa

* Mạng ngang hàng là một kiểu mạng được thiết kế cho các thiết bị trong đó

có chức năng và khả năng của các thiết bị là như nhau.

* Mạng P2P không có khái niệm trạm (client) hay máy chủ (server) mà chỉ

có khái niệm các nốt (peers) đóng vai trò như cả client và server.



Hình 1 - Mô hình mạng ngang hàng

1. So sánh mô hình Client-Server và mô hình Peer-to-Peer

* Ưu điểm

|  |  |
| --- | --- |
| Clien-Server | Peer-to-Peer |
| * Tốc độ truy cập nhanh * Khả năng mở rộng cao. * Hoạt động với bất kì loại ứng * dụng nào. * Sử dụng được với các ứng dụng * chia sẻ CSDL. * Đáng tin cậy (có server riêng). * Mức độ an toàn cao nhất. | * Không cần Server riêng, các nốt   chia sẻ tài nguyên. Khi mạng càng mở rộng thì khả năng hoạt độngcủa hệ thống càng tốt.   * Rẻ * Dễ cài đặt và bảo trì. * Thuận lợi cho việc chia sẻ file, * máy in, CD\_ROM v.v... |

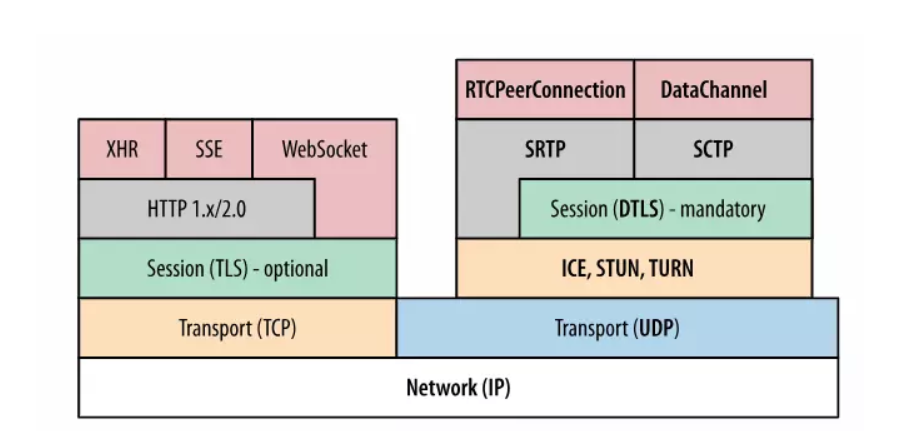
* Nhược điểm

|  |  |
| --- | --- |
| Clien-Server | Peer-to-Peer |
| * Cần server riêng * Đắt * Phức tạp trong việc bảo trì | * Chậm * Không tốt cho các ứng dụng cơ sở dữ liệu * Kém tin cậy |

**CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG**

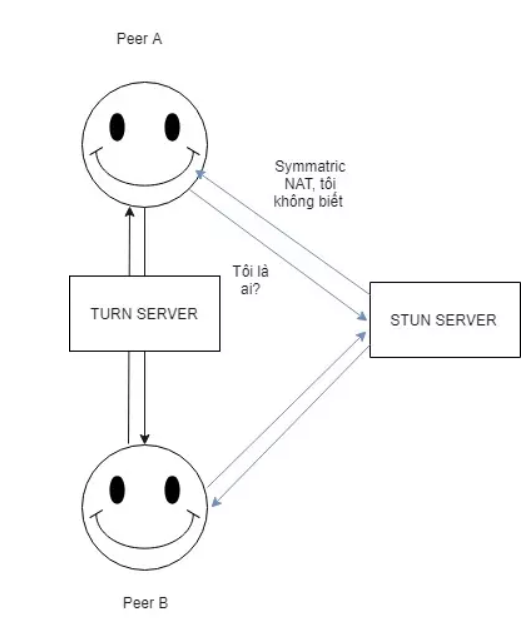
1. **Mô hình tổng quan của WEBRTC:**
2. Các giao thức được sử dụng trong WEBRTC

Do các đặc điểm cần thời gian thực cao hơn tính tin cậy, giao thức UDP được sử dụng trong WebRTC là giao thức vận chuyển.Nhưng để thỏa mãn yêu cầu của trình duyệt phải hỗ trợ giao thức và dịch vụ ở lớp khác nữa.Về cơ bản các giao thức chính sử dụng trong WebRTC được thể hiện ở hình dưới:



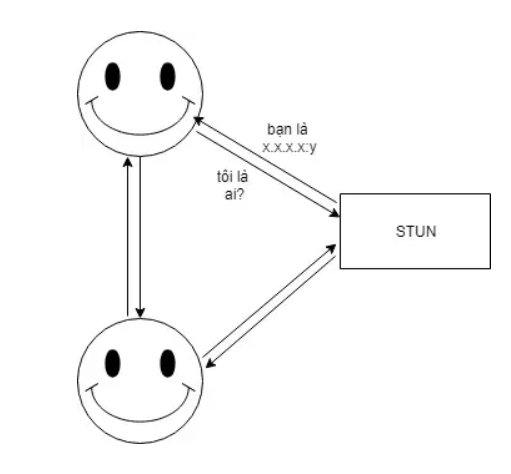
Hình 2 - Mô hình tổng quan của WEBRTC

* **SDP**
* Session Description Protocol là một chuẩn mô tả các thông số của mỗi kết nối như là độ phân giải, định dạng, codecs, mã hóa… Điều này làm cho mỗi peer có thể hiểu nhau khi dữ liệu được truyền. Đây thực chất là meta-data miêu tả nội dung chứ không phải là dữ liệu media.
* **ICE**
* Interactive Connectivity Establishment là một chuẩn được sử dụng để thiết lập kết nối giữa các các peer trên internet.Mặc dù WebRTC là kết nối trực tiếp Peer-to-Peer, nhưng thực tế nó vẫn gặp phải vấn đề NAT (Network Address Translation) gây khó khăn khi kết nối.ICE sẽ cố gắng thiết lập kết nối bằng cách tìm đường đi tốt nhất cho kết nối.Bằng cách thử hết các khả năng song song nhau, ICE có thể chọn ra được lựa chọn hiệu quả nhất cho kết nối. Đầu tiên ICE cố gắng kết nối sử dụng địa chỉ host lấy được từ hệ điều hành hoặc card mạng. Nếu thất bại nó sẽ lấy địa chỉ IP public thông qua STUN server.Nếu vẫn thất bại, lưu lượng được gửi thông qua TURN server. Các cách để kết nối này được gọi là “candidate”, cách thức trao đổi sẽ được mô tả ở các phần bên sau.
* **TURN**
* Traversal Using Relays around NAT (TURN) . Được xây dựng nhằm vượt qua symmetric NAT bằng cách mở một kết nối tới TURN server và đáp lại tất cả thông tin thông qua server này (dữ liệu audio/video/data streaming giữa các peer, không phải signaling data).



Hình 3 - Minh họa Traversal Using Relays

* **STUN**
* NAT cung cấp một địa chỉ IP để sử dụng trong mạng nội bộ. Nhưng địa chỉ này không được sử dụng bên ngoài mạng. Không biết địa chỉ công khai sẽ không có cách nào để hai Peer có thể tương tác. Để giải quyết vấn đề đó WebRTC sử dụng STUN (Session Traversal Utilities for NAT) [8]. STUN server tồn tại trên mạng internet và chỉ có nhiệm vụ duy nhất là kiểm tra địa chỉ IP và cổng của yêu cầu vừa đến và gửi trở lại IP và cổng đó.Các ứng dụng sử dụng STUN server để cung cấp IP và cổng công khai từ internet. Từ đó một WebRTC peer có thể tự lấy được địa chỉ IP và cổng công khai và đưa nó cho các peer khác thông qua cơ chế signaling.Hình bên dưới mô tả về cách làm việc của STUN server:



Hình 4 - Minh họa Session Traversal Utilities for NAT

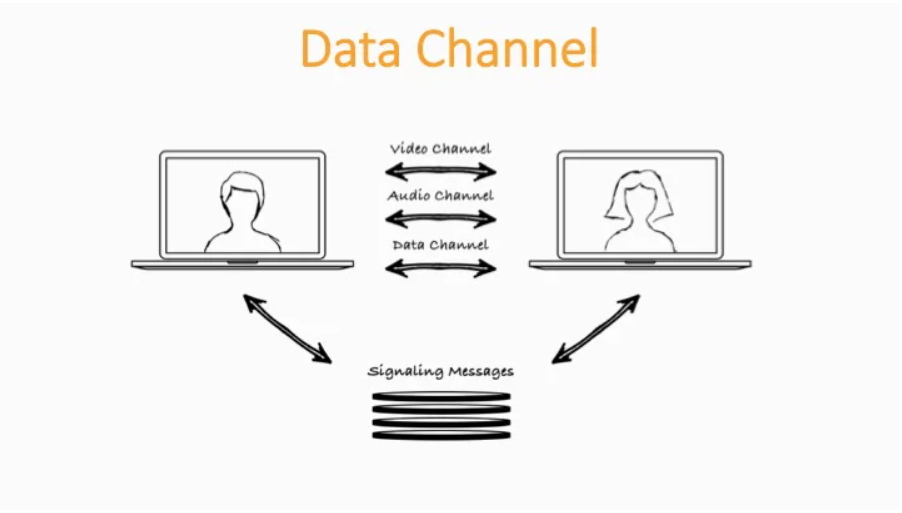
1. Các thành phần API của WEBRTC

* **MediaStream APIs**
* Để bắt đầu một cuộc Video call, trước tiên chúng ta cần có quyền truy cập vào webcam (để thu hình video), micro (để thu audio) . Và để làm được điều này WebRTC cung cấp cho chúng ta MediaStream API . Nó được thiết kế để dễ dàng truy cập các media stream của máy tính hoặc điện thoại chúng ta đang sử dụng . Thông thường chúng được gọi qua navigator.mediaDevices object . Từ object này chúng ta có thể liệt kê tất cả các thiết bị được kết nối, lắng nghe các thay đổi của thiết bị (khi thiết bị được kết nối hoặc ngắt kết nối) và sử dụng một thiết bị để truy xuất Media Stream (xem bên dưới).



Hình 5 - Minh họa việc cấp quyền truy cập MediaStream APIs

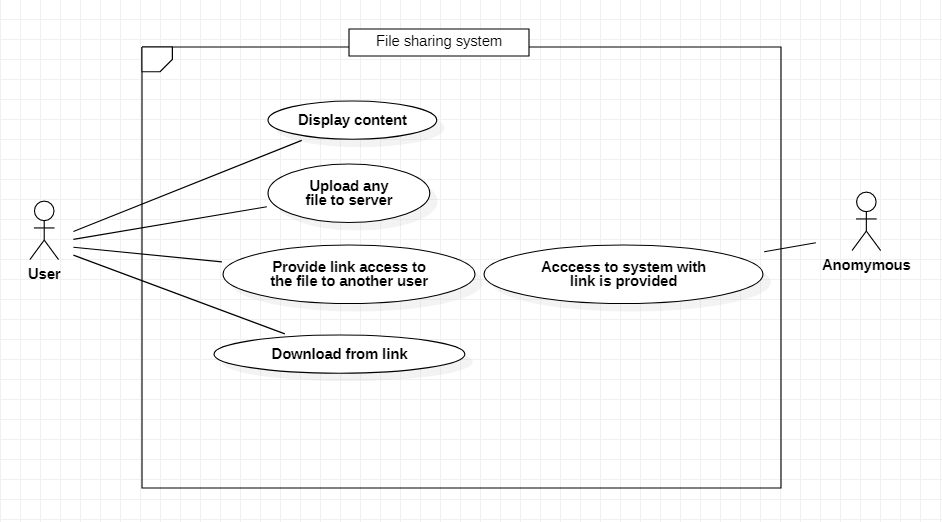
* Việc gọi function getUserMedia sẽ xin user (những người dùng của hệ thống) xin phép được truy cập vào camera và micro của thiết bị.
* **RTCDataChannel**
* RTCDataChannel là một kênh hai chiều để gửi dữ liệu bất kỳ qua kết nối WebRTC. Nó hoạt động tương tự web sockets,nhưng dữ liệu được truyền tải trực tiếp giữa hai trình duyệt, và nó cho phép trao đổi thông tin nhanh và đáng tin cậy (giống như giữa UDP và TCP).



Hình 6 - Mô phỏng việc truyền dữ liệu sử dụng DataChannel

* Chúng ta có thể sử dụng Data Channels cho bất cứ điều gì bạn muốn bằng cách sử dụng WebSocket. Điểm khác biệt là bất cứ điều gì bạn gửi thông qua DataChannel sẽ được mã hóa. Bạn sẽ cần bảo mật nhiều hơn máy chủ WebServer và ít lưu lượng truy cập hơn.
* **WebRTC PeerConnection**
* PeerConnection là một phần của trong công nghệ WebRTC . Nó có chức năng tạo ra kết nối hai máy tính khác nhau thông qua giao thức peer-to-peer (ngang hàng). Chúng ta có thể truyền dữ liệu dạng video, audio hoặc binany data bất kì (thông qua RTCDataChannel). Để có thể kết nối 2 máy tính với nhau thì chúng cần cung cấp thông tin về cấu hình ICE Server (Interactive Connectivity Establishment). Nó bao một gồm STUN và TURN server và trách nhiệm của các server này là cung cấp các ICE candidates cho các máy tính khi truyền dữ liệu từ peer gốc cho đến peer địa chỉ. Việc truyền các ICE candidates này thường được gọi là Signaling.
* Về phía bạn - người gọi (caller), chúng ta cần:
* Khởi tạo một RTCPeerConnection object`
* Khời tạo RTCSessionDescription từ RTCPeerConnection thông qua cách gọi hàm createOffer()
* RTCSessionDescription là sự đại diện của máy caller
* Cuối cùng, chúng ta cần set một nơi để lắng nghe khi bạn gửi yêu cầu call đến partner - parthner đồng ý - gửi tín hiệu đồng ý đến bạn.

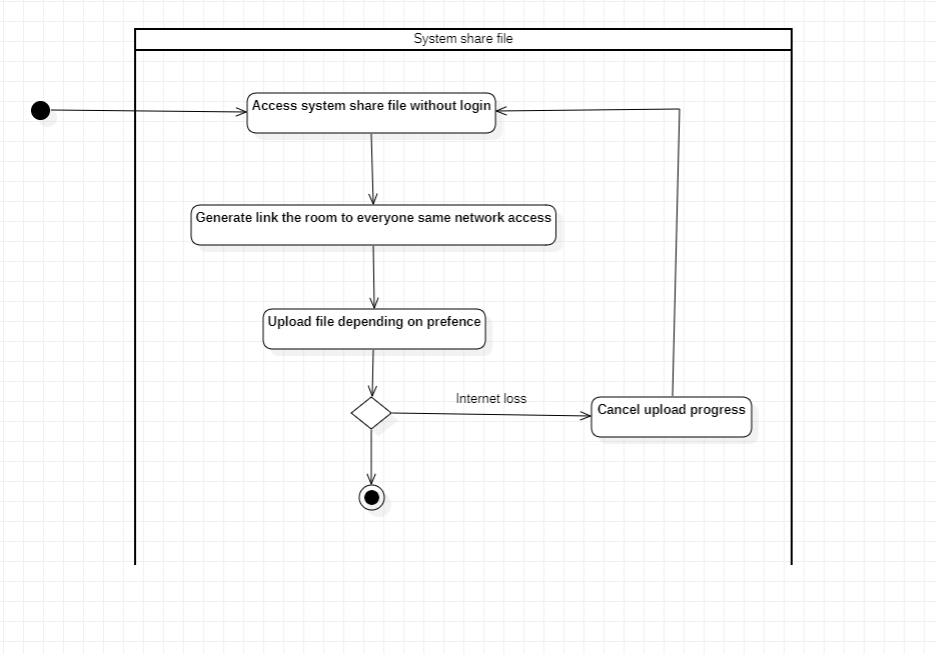
1. **Biểu đồ usecase**



Hình 7 - Biểu đồ usecase

* Tóm tắt:
* Người muốn chia sẻ tập tin không cần phải đăng nhập, sau khi truy cập vào hệ thống, user có thể thực hiện upload file cần chia sẻ lên hệ thống. Sau khi upload xong thì sẽ có link chia sẻ để mọi người có thể tải về.

1. **Biểu đồ hoạt động**

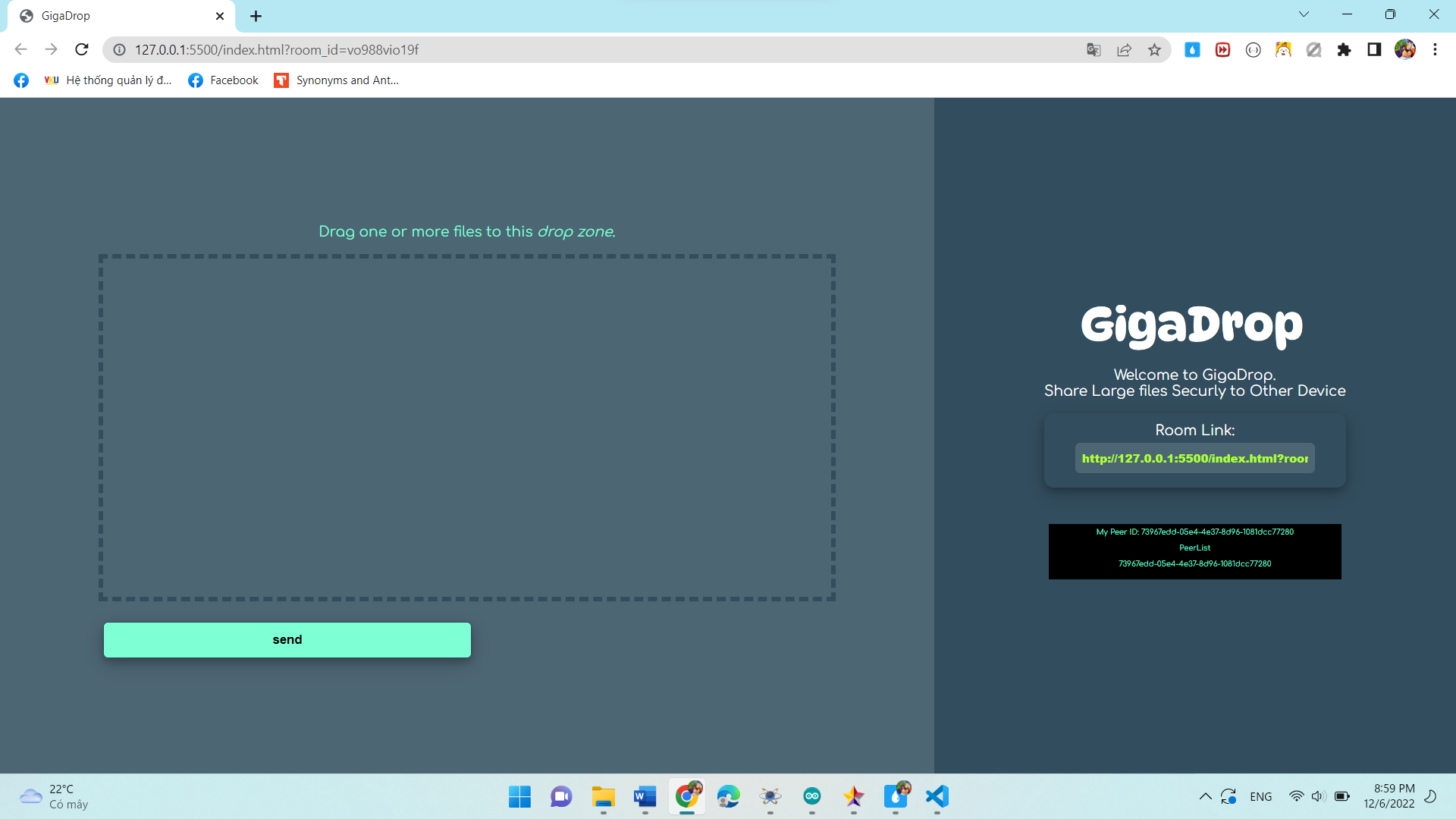


Hình 8 - Biểu đồ hoạt động hệ thống chia sẻ file

**CHƯƠNG 3. TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ**

1. **Giới thiệu về website dùng để chia sẻ file**

Sau một thời gian tìm hiểu và học hỏi từ những kiến thức đã được học, em đã xây dựng lên website chia sẻ file với người khác. Trang web sử dụng công nghệ mạng ngang hàng P2P (peer-to-peer network) vì không cần phải tải dữ liệu lên bất cứ server lưu trữ nào nên có thể truyền tải dữ liệu một cách an toàn giữa các máy ngang hàng với nhau. Trang web có thể chia sẻ file cho máy tính và các thiết bị di động. Hiểu đơn giản là giống như bạn copy-paste file từ máy này qua máy kia, không cần up lên server.



Hình 9 - Trang chủ chia sẻ file

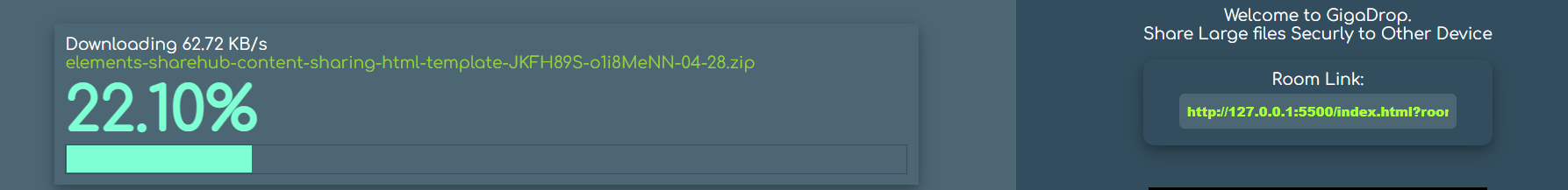
1. **Một số thành phần của ứng dụng**
2. Kéo thả file



Hình 10 - Drop và kéo thả file

* Người dùng có thể click vào vùng trống để chọn file cần upload, hoặc kéo một file bất kì, sau đó thực hiện thao tác thả.

1. Truy cập đường dẫn room



Hình 11 - Đường dẫn room

* Room được tạo ra khi có một peer truy cập vào hệ thống, mỗi peer khác có thể truy cập vào hệ thống chia sẻ file thông qua đường dẫn “ room link”
* Thanh progress bar hiển thị quá trình upload file cho người dùng dễ dành theo dõi và quan sát

1. Xem danh sách peer online và tải file được chia sẻ



* Danh sách các peer online được hiển thị, mỗi peer sẽ có một tên riêng biệt do hệ thống tự động tạo ra
* Các file sau khi được peer chia sẻ sẽ hiện thị với tên tương ứng do người gửi upload và sẽ xuất hiện khi quá trình chia sẻ hoàn tất. Bất cứ các peer trong mạng đều có thể tải về một cách dễ dàng.

# **KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

1. **Kết luận:**

Qua quá trình học hỏi và nghiên cứu đồ án, em đã tiếp thu được nhiều kiến thức như sử dụng thư viện PeerJS cung cấp giải pháp và các hàm có sẵn cho các lập trình viên phát triển ứng dụng peer-to-peer, qua đó nâng cao kỹ năng lập trình sử dụng ngôn ngữ JavaScript. . Song với đó, trình độ làm báo cáo, slide thuyết trình và thiết kế hệ thống và hình ảnh ngày càng tốt, nhanh chóng và hiệu quả hơn

1. **Hạn chế**

Mặc dù đã hệ thống đã hoàn thành, nhưng nó vẫn còn rất nhiều mặt hạn chế và vấn đề tồn tại. Trong thời gian tới thì em sẽ cố gắng khắc phục các hạn chế, những vấn đề chưa được tốt, và bổ sung thêm những chức năng mới cho hệ thống.

1. **Hướng phát triển**

* Xây dựng ứng dụng trên nền tảng di động
* Cải thiện trải nghiệm người dùng bằng cách nâng cấp giao diện website
* Phát triển thêm tính năng gọi video sử dụng kĩ thuật WebRTC

# **TÀI LIỆU THAM KHẢO:**

* Mạng internet.
* Từ trang https://www.wikipedia.org/