

ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
_____ * _____



BÁO CÁO PROJECT II

Đề tài: Tìm hiểu và thử nghiệm công nghệ
WebRTC (truyền audio, video)

Học kỳ : 20222
Mã lớp : 727529
Giáo viên hướng dẫn : TS. Đỗ Tiến Dũng
Sinh viên thực hiện :

Họ tên	Mã số sinh viên	Email	Lớp
Trần Anh Tuấn	20204860	tuân.ta204860@sis.hust.edu.vn	IT2-03-K65

Hà Nội, tháng 6 năm 2023

Mục lục

I. Giới thiệu đề tài	3
II. Kiến thức liên quan	3
1. Giao tiếp thời gian thực - RTC	3
2. Kết nối ngang hàng - P2P	3
3. NAT (Network Address Translation):	3
4. Các kỹ thuật để vượt NAT sử dụng trong WebRTC.....	4
III. Thiết kế	5
IV. Kết quả đạt được	6
V. Hướng phát triển	7

I. Giới thiệu đề tài

WebRTC là một dự án mã nguồn mở được bắt đầu xây dựng vào năm 2011 bởi Google. Công nghệ này bao gồm tập các chuẩn giao thức và API được tạo ra nhằm giúp các trình duyệt có thể tương tác với nhau trong thời gian thực thông qua kết nối ngang hàng.

II. Kiến thức liên quan

1. Giao tiếp thời gian thực - RTC

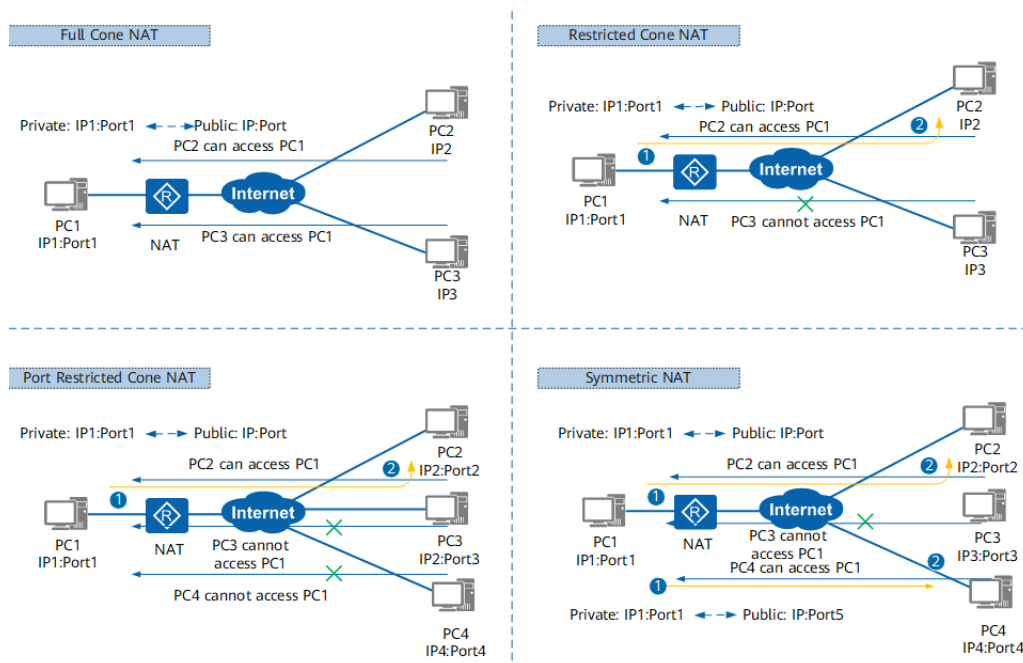
- Việc trao đổi thông tin giữa hai bên độ trễ tối thiểu và gần như không có độ trễ đường truyền.

2. Kết nối ngang hàng - P2P

- Các máy tính kết nối trực tiếp với nhau thông qua Internet, chia sẻ dữ liệu mà không cần máy chủ trung tâm

3. NAT (Network Address Translation):

- Dữ liệu chuyển tiếp từ mạng LAN (sử dụng địa chỉ cục bộ) sang mạng Internet (sử dụng địa chỉ công cộng) và ngược lại cần được chuyển đổi địa chỉ.
- NAT gây ra khó khăn trong việc thiết lập liên kết P2P khi cả 2 máy đều nằm sau NAT do các thiết bị NAT che giấu địa chỉ nội bộ của các máy tính.
- Để có thể kết nối với nhau, 2 máy cần phải biết địa chỉ công cộng của nhau. Tuy nhiên các máy tính nội bộ không thể tự biết địa chỉ công cộng của mình nên cũng không thể trao đổi cho nhau biết.
- Các kỹ thuật NAT phổ biến bao gồm: Full-cone NAT, Restricted-cone NAT, Port-restricted NAT, Symmetric NAT.



4. Các kỹ thuật để vượt NAT sử dụng trong WebRTC

- STUN (Session Traversal Utilities for NAT)
 - Một máy chủ nằm ngoài Internet (không nằm sau NAT)
 - Kỹ thuật sử dụng STUN còn được gọi là Hole-punching, hoạt động được với Full-cone NAT, Restricted-cone NAT, Port-restricted NAT
 - Nhận yêu cầu và trả về địa chỉ công cộng của máy tính
 - Chỉ cần sử dụng ở giai đoạn thiết lập kết nối
 - Khi sử dụng Symmetric NAT, máy sẽ được ánh xạ sang một địa chỉ riêng biệt để trao đổi với STUN server nên không thể sử dụng địa chỉ được trả về để yêu cầu thiết lập liên kết với các máy khác
- TURN (Traversal Using Relay NAT)
 - Server trung gian chuyển tiếp dữ liệu giữa 2 máy sử dụng trong trường hợp không thể thiết lập kết nối trực tiếp giữa 2 máy
 - Sử dụng cần kết hợp với cả STUN server hoặc phải kiêm vai trò của STUN server để lấy địa chỉ công cộng.
- SDP (Session Description Protocol) – Giao thức mô tả phiên
 - Là một chuẩn dùng để mô tả thông tin cho một phiên kết nối.
 - Thông tin của phiên (session) bao gồm trong đó có cả thông tin về thời gian (time) và phương tiện (media).
 - SDP Signaling: là quá trình chuyển được gói tin SDP sang máy còn lại, không phân biệt cách thức

- Session description

```
v= (protocol version number, currently only 0)
o= (originator and session identifier : username, id, version number, network address)
s= (session name : mandatory with at least one UTF-8-encoded character)
i= (session title or short information)
u= (URI of description)
e= (zero or more email address with optional name of contacts)
p= (zero or more phone number with optional name of contacts)
c= (connection information-not required if included in all media)
b= (zero or more bandwidth information lines)
One or more time descriptions ("t=" and "r=" lines; see below)
z= (time zone adjustments)
k= (encryption key)
a= (zero or more session attribute lines)
Zero or more Media descriptions (each one starting by an "m=" line; see below)
```

- Time description (mandatory)

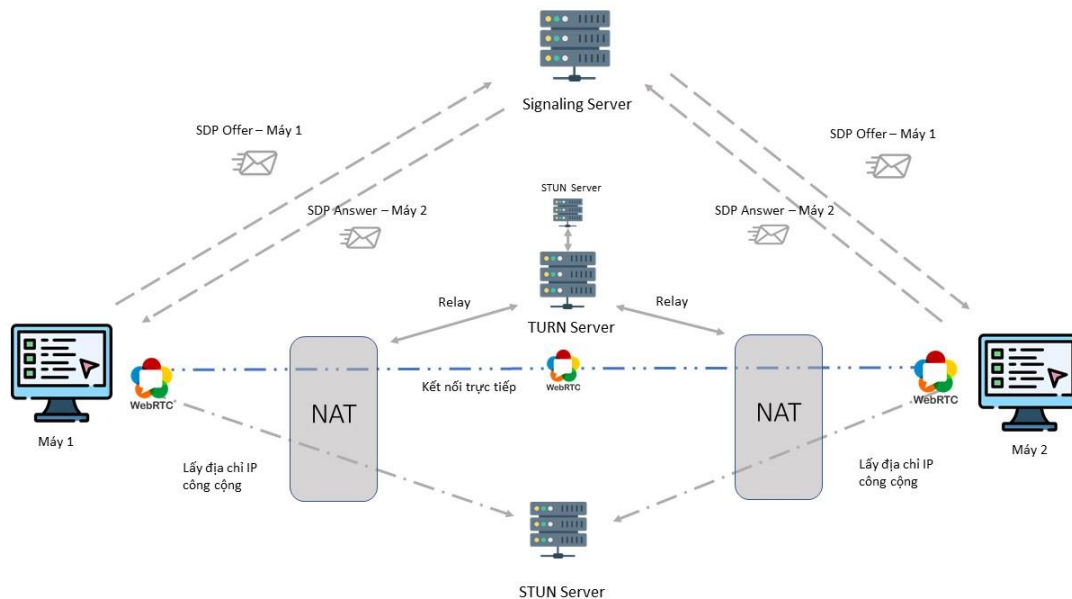
```
t= (time the session is active)
r= (zero or more repeat times)
```

- Media description (optional)

```
m= (media name and transport address)
i= (media title or information field)
c= (connection information - optional if included at session level)
b= (zero or more bandwidth information lines)
k= (encryption key)
a= (zero or more media attribute lines - overriding the Session attribute lines)
```

- ICE (Interactive Connectivity Establishment)
 - Giao thức có chức năng tìm tất cả các ứng viên (bao gồm các máy chủ STUN, máy chủ TURN và các địa chỉ IP nội bộ) để có thể thiết lập được đường đi tốt nhất giữa 2 máy
 - Các thông tin này được chuyển đến máy còn lại thông qua SDP

III. Thiết kế

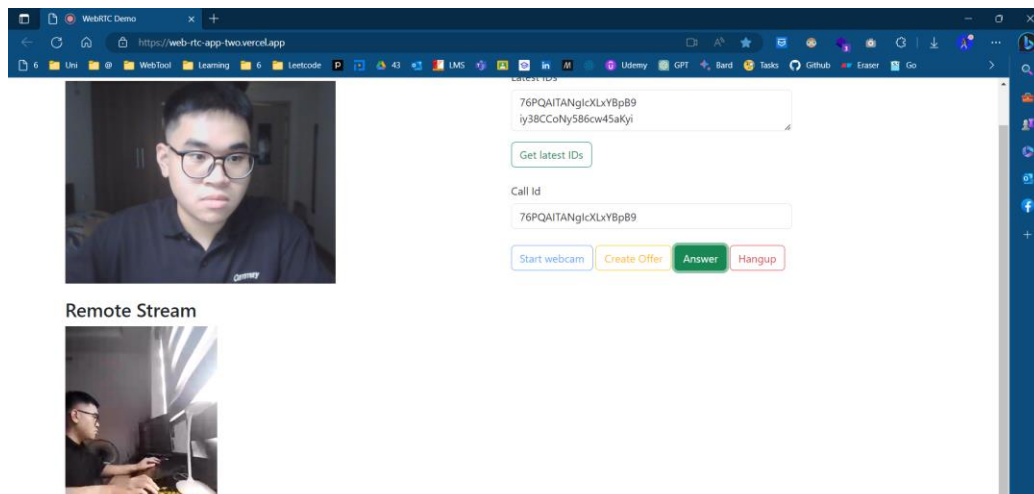
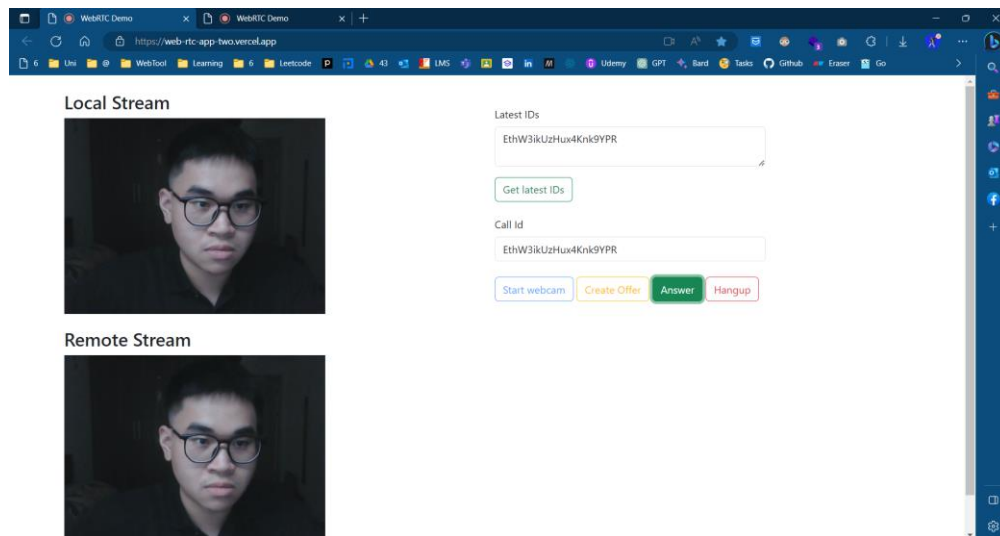


- Ứng dụng chạy trên trình duyệt web ở máy của người dùng
- Sử dụng Google Firebase để làm Signaling Server
- Sử dụng các máy chủ STUN và TURN của [Metered](#) ([Metered - Dashboard](#))
- Quá trình kết nối 2 máy sử dụng WebRTC APIs:
 - Mỗi máy có 2 đối tượng **MediaStream()** là **localStream** và **remoteStream**, đối tượng **localStream** lấy máy ảnh và mic của máy gọi làm nguồn
 - Sử dụng **RTCPeerConnection()** để khởi tạo một đối tượng kết nối, thu thập các ICE candidate và lưu vào Firebase mỗi khi có sự kiện **onicecandidate()**
 - Đối tượng này sẽ thêm các dòng truyền video và audio của máy còn lại vào đối tượng **remoteStream** khi gặp sự kiện **ontrack()**
 - Máy gọi tạo gói tin SDP Offer dựa vào các ICE candidate đã thu được bằng **createOffer()**, lưu vào Firebase, cập nhật thông tin máy bằng **setLocalDescription()**
 - Đợi gói tin SDP Answer, cập nhật thông tin kết nối của máy còn lại bằng **setRemoteDescription()**
 - Đợi ICE Candidate của máy trả lời được đưa lên cơ sở dữ liệu và thêm vào danh sách ICE Candidate bằng **addIceCandidate()**
 - Máy trả lời sau khi khởi tạo đối tượng kết nối, thu thập các ICE candidate thì sẽ đọc tin SDP Offer từ cơ sở dữ liệu, cập nhật thông tin của máy gọi bằng **setRemoteDescription()** và tạo gói tin SDP Answer bằng **createAnswer()**, lưu vào Firebase, sau đó cập nhật thông tin kết nối của máy bằng **setLocalDescription()**
 - Máy trả lời tiếp tục lắng nghe các ICE candidate của máy gọi thay đổi và cập nhật vào đối tượng kết nối của mình

- Lúc này nếu có đường kết nối khả thi, 2 máy sẽ có thể truyền được dữ liệu cho nhau

IV. Kết quả đạt được

- Tìm hiểu được các thành phần cơ bản của công nghệ WebRTC: các APIs trong WebRTC, cách WebRTC kết nối các máy nằm sau NAT.
- Cài đặt được một chương trình sử dụng WebRTC APIs: [Mã nguồn](#)
 - Kết nối được các máy trên cùng mạng nội bộ và các máy khác mạng
 - Hai máy đã có thể truyền được video và âm thanh cho nhau



V. Hướng phát triển

- Bổ sung các tính năng giúp người dùng chỉ định người cần gọi, tính năng nhắn tin, chia sẻ màn hình
- Tự cài đặt các máy chủ STUN, TURN và máy chủ báo hiệu riêng
- Chỉnh sửa giao diện thân thiện với người dùng hơn
- Nghiên cứu thêm về cách mã hóa các gói tin của WebRTC

Nguồn tham khảo:

<https://info.supPORT.huawei.com/info-finder/encyclopedia/en/NAT.html>

https://en.wikipedia.org/wiki/Session_Description_Protocol

<https://webrtc.github.io/webrtc-org/architecture/>

https://youtu.be/WmR9IMUD_CY