ĐẠI HỌC QUỐC GIA TP.HCM

TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA KHOA HỌC & KỸ THUẬT MÁY TÍNH



BÁO CÁO THỰC TẬP TỐT NGHIỆP

NGHIÊN CỨU PHÁT TRIỂN

HỆ THỐNG HỖ TRỢ  
CHƯƠNG TRÌNH ĐÀO TẠO TỪ XA

SỬ DỤNG CÔNG NGHỆ WEBRTC/HTML5

GVHD : TS. BÙI HOÀI THẮNG

GVPB : Th.S NGUYỄN VĂN ĐOÀN

======🙡🙣======

SVTH : NGUYỄN MẬU QUANG VŨ - 50802669

PHẠM NGUYÊN TRÌNH - 50802353

*Tháng 06/2012*

LỜI CAM ĐOAN

Chúng tôi xin cam đoan toàn bộ phần nghiên cứu và trình bày đồ án được chúng tôi thực hiện dưới sự hướng dẫn của thầy Bùi Hoài Thắng. Ngoài các tài liệu tham khảo được liệt kê, chúng tôi không sao chép từ các nguồn tài liệu hoặc các công trình nghiên cứu khác. Nếu có bất kỳ sai phạm nào, chúng tôi xin hoàn toàn chịu trách nhiệm trước Ban chủ nhiệm Khoa và Ban giám hiệu nhà trường.

TP.Hồ Chí Minh, tháng 06 năm 2012

LỜI CẢM ƠN

Sau năm tháng thực hiện đề tài, bên cạnh nỗ lực cá nhân, chúng tôi nhận được sự hỗ trợ tận tâm từ thầy và các anh chị trong ngành. Điều này giúp chúng tôi định hướng nghiên cứu, theo sát tiến độ đề ra và hoàn thành những phần nghiên cứu quan trọng.

Chúng tôi xin được gửi lời cảm ơn chân thành đến thầy Bùi Hoài Thắng, giảng viên hướng dẫn trực tiếp đề tài. Thầy là người đã định hướng nghiên cứu và đóng góp những ý kiến quý báu cho đề tài. Chúng tôi xin cảm ơn anh Nguyễn Hoàng Nguyên và anh Đặng Xuân Hiếu đã tận tình giúp đỡ, luôn theo dõi sát sao và góp ý thẳng thắn trong suốt quá trình thực hiện. Và gửi lời cảm ơn đến quý thầy cô và khoa Khoa Học & Kỹ Thuật Máy Tính đã tận tình giảng dạy và tạo điều kiện thuận lợi cho chúng tôi học tập, nghiên cứu và thực hiện đề tài này.

Những lời cảm ơn sau cùng dành cho gia đình và bạn bè, những người đã cho tôi sự động viên, quan tâm và giúp đỡ kịp thời nhất để vượt qua những khó khăn trong cuộc sống và trong quá trình hoàn thành luận văn.

TP.Hồ Chí Minh, tháng 06 năm 2012

TÓM TẮT TÀI LIỆU

Mục đích chính của tài liệu này trình bày về kết quả thu được sau giai đoạn I thực hiện đề tài: Nghiên cứu phát triển hệ thống hỗ trợ chương trình đào tạo từ xa sử dụng công nghệ WebRTC/HTML5. Tài liệu trình bày về công nghệ truyền tải nội dung real-time media (video và voice/audio) trên nền web và khả năng ứng dụng công nghệ xây dựng hệ thống đào tạo từ xa. Công nghệ trọng tâm được trình bày là WebRTC - tên gọi được đặt ra lần đầu bởi Google năm 2011. Công nghệ hoạt động trên nền trình duyệt và là một chuẩn của HTML5, đây là một công nghệ đầy tiềm năng và còn nhiều vấn đề phải giải quyết trước khi được chính thức thương mại hóa, dự kiến vào năm 2013. Phần đầu của tài liệu trình bày các đặc điểm kỹ thuật và tiềm năng ứng dụng của công nghệ; phần sau phân tích về khả năng sử dụng công nghệ xây dựng hệ thống đào tạo từ xa.

MỤC LỤC

**Nội dung Trang**

[LỜI CAM ĐOAN i](#_Toc327184952)

[LỜI CẢM ƠN ii](#_Toc327184953)

[TÓM TẮT TÀI LIỆU iii](#_Toc327184954)

[MỤC LỤC iv](#_Toc327184955)

[MỤC LỤC HÌNH ix](#_Toc327184956)

[KIẾN THỨC CẦN CÓ x](#_Toc327184957)

[TRẠNG THÁI x](#_Toc327184958)

[TÓM TẮT TÀI LIỆU xi](#_Toc327184959)

[PHẦN I: TỔNG QUAN 1](#_Toc327184960)

[1. Giới thiệu đề tài 2](#_Toc327184961)

[1.1. Tóm tắt tình hình công nghệ thế giới 2](#_Toc327184962)

[1.2. Mục đích đề tài 2](#_Toc327184963)

[1.3. Phạm vi đề tài 3](#_Toc327184964)

[1.4. Các vấn đề cần giải quyết: 3](#_Toc327184965)

[2. Đồ án và hướng phát triển luận văn: 4](#_Toc327184966)

[2.1. Mục đích đồ án: 4](#_Toc327184967)

[2.2. Phạm vi đồ án: 4](#_Toc327184968)

[2.3. Phát triển luận văn: 4](#_Toc327184969)

[PHẦN II: CƠ SỞ LÝ THUYẾT 5](#_Toc327184970)

[1. Khái quát mô hình lý thuyết 6](#_Toc327184971)

[2. WebRTC 6](#_Toc327184972)

[2.1. WebRTC là gì ? 6](#_Toc327184973)

[2.1.1. RTC 7](#_Toc327184974)

[2.1.2. WebRTC Web API 7](#_Toc327184975)

[2.1.3. RTCWeb 7](#_Toc327184976)

[2.1.4. HTML5 7](#_Toc327184977)

[2.1.5. JavaScript 8](#_Toc327184978)

[2.1.6. ROAP - RTCWeb Offer/Answer Protocol 8](#_Toc327184979)

[2.1.7. JSEP – JavaScript Establishment Protocol 8](#_Toc327184980)

[2.1.8. Trình duyệt 8](#_Toc327184981)

[2.1.9. Audio codec và Video codec 8](#_Toc327184982)

[2.2. Tình trạng phát triển hiện tại 8](#_Toc327184983)

[2.3. Kiến trúc tổng quan WebRTC 9](#_Toc327184984)

[2.4. WebRTC có thể làm được gì ? 10](#_Toc327184985)

[2.4.1. Giao tiếp trong tương lai 10](#_Toc327184986)

[2.4.2. Ứng dụng hội thoại trực tuyến 11](#_Toc327184987)

[2.4.3. Ứng dụng hội thoại trực tuyến giữa trình duyệt và điện thoại 11](#_Toc327184988)

[2.4.4. Ứng dụng hội thoại trực tuyến khía cạnh doanh nghiệp. 11](#_Toc327184989)

[2.4.5. Ứng dụng hội thoại trực tuyến nhiều bên 12](#_Toc327184990)

[2.4.6. Ứng dụng truyền tải media 13](#_Toc327184991)

[2.4.7. Ứng dụng thu hình trực tiếp hỗ trợ ghi hình và phát lại 15](#_Toc327184992)

[2.4.8. Mô hình liên lạc giữa các nhà cung cấp dịch vụ khác nhau 16](#_Toc327184993)

[2.4.9. On-cloud speech / video recognization 16](#_Toc327184994)

[2.4.10. On-cloud real-time game play, real-time rendering. 16](#_Toc327184995)

[2.5. Chuẩn RTCWeb 17](#_Toc327184996)

[2.5.1. Kiến trúc và nhóm chức năng 17](#_Toc327184997)

[2.5.2. Audio codec 19](#_Toc327184998)

[2.5.3. Video codec 19](#_Toc327184999)

[2.6. Kiến trúc WebRTC API 20](#_Toc327185000)

[2.7. Mô hình hoạt động của WebRTC Native API 22](#_Toc327185001)

[2.7.1.1. Block Diagram 22](#_Toc327185002)

[2.7.1.2. Bắt đầu cuộc gọi video 23](#_Toc327185003)

[2.7.1.3. Nhận cuộc gọi video 24](#_Toc327185004)

[2.7.1.4. Kết thúc cuộc gọi video 25](#_Toc327185005)

[2.8. Các công nghệ liên quan 25](#_Toc327185006)

[2.8.1. HTML 25](#_Toc327185007)

[2.8.2. HTML5 26](#_Toc327185008)

[2.8.3. CSS 26](#_Toc327185009)

[2.8.4. JavaScript 26](#_Toc327185010)

[2.8.5. - XMLHttpRequest: 26](#_Toc327185011)

[2.8.6. - Media Capture API: 27](#_Toc327185012)

[2.8.7. Push Notification 27](#_Toc327185013)

[2.8.8. WebSocket 27](#_Toc327185014)

[2.8.9. Tag <video> và <audio> 27](#_Toc327185015)

[2.8.10. Canvas 27](#_Toc327185016)

[2.8.11. Offline Application 27](#_Toc327185017)

[3. Lưu trữ và truyền tải video: 27](#_Toc327185018)

[3.1. Giới thiệu: 27](#_Toc327185019)

[3.1.1. Audio & Video là gì ? 27](#_Toc327185020)

[3.1.2. Streaming Video 28](#_Toc327185021)

[3.1.2.1. Streaming Video là gì 28](#_Toc327185022)

[3.1.2.2. Quá trình Video Streaming 28](#_Toc327185023)

[3.1.2.3. Video Streaming: 29](#_Toc327185024)

[3.1.2.4. Kiến trúc hệ thống Streaming 29](#_Toc327185025)

[3.2. Các phương pháp lưu trữ và truyền tải video 30](#_Toc327185026)

[3.2.1. Tập tin chứa dữ liệu dữ liệu đa phương tiện: 30](#_Toc327185027)

[3.2.2. Thuật toán mã hóa (Codec = Compressor-Decompressor): 31](#_Toc327185028)

[3.2.2.1. Codec là gì? 31](#_Toc327185029)

[3.2.2.2. Tại sao phải sử dụng Codec? 32](#_Toc327185030)

[3.2.2.3. Các loại codec video: 32](#_Toc327185031)

[3.2.2.4. Các codec audio: 36](#_Toc327185032)

[3.3. Truyền tải video/audio thông qua WebRTC: 40](#_Toc327185033)

[3.3.1. VP8: 40](#_Toc327185034)

[3.3.2. Opus 45](#_Toc327185035)

[4. Các công nghệ network VoIP: 46](#_Toc327185036)

[4.1. SIP: 46](#_Toc327185037)

[4.2. NAT / TURN / STUN / ICE: 47](#_Toc327185038)

[PHẦN III: ỨNG DỤNG WEBRTC TRONG GIẢNG DẠY TRỰC TUYẾN 49](#_Toc327185039)

[1. Đào tạo từ xa 50](#_Toc327185040)

[1.1. Giới thiệu 50](#_Toc327185041)

[1.1.1. Lịch sử đào tạo từ xa. 50](#_Toc327185042)

[1.1.2. Xu hướng công nghệ 50](#_Toc327185043)

[1.2. Đặc điểm của đào tạo từ xa 51](#_Toc327185044)

[1.2.1. Thế mạnh của đào tạo từ xa 51](#_Toc327185045)

[1.2.2. Đảm bảo chất lượng đào tạo từ xa 52](#_Toc327185046)

[1.3. Ví dụ mô hình đào tạo từ xa 52](#_Toc327185047)

[1.3.1. Đào tạo từ xa ở trường ĐH Bách Khoa TP. HCM 52](#_Toc327185048)

[1.3.2. Udacity (www.udacity.com) 53](#_Toc327185049)

[2. Đào tạo từ xa từ góc nhìn kỹ thuật 54](#_Toc327185050)

[2.1. Xác thực học viên 54](#_Toc327185051)

[2.2. Trình chiếu video và slide bài giảng 54](#_Toc327185052)

[2.3. Giảng dạy và thảo luận trực tuyến 54](#_Toc327185053)

[2.4. Làm bài tập, thi và theo dõi quá trình học tập 55](#_Toc327185054)

[2.5. Phòng thí nghiệm ảo 56](#_Toc327185055)

[2.6. Chịu được lượng lớn người dùng truy cập đồng thời 56](#_Toc327185056)

[3. Sử dụng WebRTC/HTML5 trong hệ thống đào tạo từ xa 56](#_Toc327185057)

[3.1. Trình chiếu video và slide bài giảng 56](#_Toc327185058)

[3.2. Giảng dạy và thảo luận trực tuyến 57](#_Toc327185059)

[3.3. Phòng thí nghiệm ảo 58](#_Toc327185060)

[PHẦN IV: KẾT QUẢ 59](#_Toc327185061)

[1. Những phần việc đã làm được và chưa làm được: 60](#_Toc327185062)

[2. Các khó khăn, thuận lợi: 60](#_Toc327185063)

[3. Demo: 61](#_Toc327185064)

[3.1. Demo native WebRTC trên Linux (Ubuntu): 61](#_Toc327185065)

[3.1.1. Server: 61](#_Toc327185066)

[3.1.2. Client: 61](#_Toc327185067)

[3.2. Demo native WebRTC trên Windows: 62](#_Toc327185068)

[3.2.1. Server: 62](#_Toc327185069)

[3.2.2. Client: 63](#_Toc327185070)

[3.3. Demo giao tiếp WebRTC giữa Brower, Linux và Windows: 63](#_Toc327185071)

[3.3.1. Client trên Windows: 64](#_Toc327185072)

[3.3.2. Client trên Ubuntu: 64](#_Toc327185073)

[3.3.3. Web loopback client: 65](#_Toc327185074)

[3.4. Demo nhận video từ camera và xử lý bằng HTML5/JavaScript 65](#_Toc327185075)

[PHẦN V: PHỤ LỤC 66](#_Toc327185076)

[TÀI LIỆU THAM KHẢO 67](#_Toc327185077)

MỤC LỤC HÌNH

**Nội dung Trang**

[Hình 1: Mô hình giao tiếp RTC đơn giản 11](#_Toc327185529)

[Hình 2: Kiến trúc API của WebRTC 20](#_Toc327185530)

[Hình 3: Block Diagram 22](#_Toc327185531)

[Hình 4: Bắt đầu cuộc gọi 23](#_Toc327185532)

[Hình 5: Nhận cuộc gọi 24](#_Toc327185533)

[Hình 6: Kết thúc cuộc gọi 25](#_Toc327185534)

KIẾN THỨC CẦN CÓ

Để có thể theo dõi nội dung được trình bày trong tài liệu một cách hiệu quả, người đọc cần chuẩn bị những kiến thức sau :

* Kiến thức cơ sở chuyên ngành Khoa học Máy tính.
* Kiến thức cơ sở về web application và HTML/HTML5.
* Kiến thức cơ sở về các công nghệ đàm thoại trực tuyến VoIP.

Trong khi trình bày, chúng tôi giả định rằng người đọc đã có các kiến thức trên và chỉ giới thiệu tổng quan về các kiến thức cơ sở liên quan mà không đi sâu trình bày nhằm giảm tải các nội dung không quan trọng trong tài liệu. Người đọc có thể tìm hiểu về công nghệ liên quan qua các nguồn tham khảo ở cuối tài liệu.

TRẠNG THÁI

Nội dung tài liệu được thực hiện bởi các sinh viên khóa 2008, khoa Khoa học và Kỹ thuật Máy tính, trường ĐH Bách Khoa TP. HCM. Tài liệu hoàn thành vào tháng 06 năm 2012. Tài liệu sử dụng các nguồn tham khảo đang trong quá trình phát triển và có thể thay đổi vào bất kỳ lúc nào. Các trích dẫn từ tài liệu vui lòng ghi rõ nguồn và chú thích “đang trong quá trình phát triển” (trạng thái “working in progress”).

TÓM TẮT TÀI LIỆU

Tài liệu trình bày kết quả thực hiện giai đoạn I của đề tài. Nhiệm vụ chính của giai đoạn này là nghiên cứu các công nghệ truyền tải và lưu trữ nội dung real-time media và phân tích khả năng ứng dụng xây dựng hệ thống đào tạo từ xa.

Tài liệu được chia làm bốn phần chính

* 1. Tổng quan

Phần này trình bày tổng quan về đề tài nghiên cứu và các giai đoạn thực hiện: (I) thực tập tốt nghiệp và (II) luận văn tốt nghiệp. Giúp người đọc nắm được nội dung đề tài và nội dung được thực hiện trong từng giai đoạn.

* 1. Cơ sở lý thuyết

Phần này tập trung trình bày về lý thuyết các công nghệ có khả năng được sử dụng trong hệ thống đào tạo từ xa. Tập trung vào công nghệ trọng tâm là WebRTC và các công nghệ xoay quanh. Phần này bao gồm

1. Khái quát mô hình lý thuyết

Cung cấp cho người đọc cái nhìn bao quát về các công nghệ được trình bày, mối liên hệ giữa chúng và tại sao các công nghệ này được chọn để nghiên cứu.

1. Công nghệ WebRTC

Công nghệ đàm thoại video trực tiếp giữa các trình duyệt được phát triển và chuẩn hóa bởi hai tổ chức quốc tế IETF và W3C, có khả năng trở thành phương thức giao tiếp chủ đạo trong tương lai gần. Phần này giúp người đọc hình dung được công nghệ WebRTC, tình trạng phát triển hiện tại, các tiềm năng ứng dụng trong thực tiễn, mô hình hoạt động và mô tả kỹ thuật.

1. Công nghệ lưu trữ và truyền tải media

Phần đầu trình bày về các công nghệ lưu trữ và truyền tải media hiện nay và xu hướng tương lai. Phần sau phân tích đặc điểm kỹ thuật và thế mạnh của hai codec sử dụng trong WebRTC là Opus (audio codec) và VP8 (video codec).

1. Các công nghệ liên quan

Ngoài các công nghệ trọng tâm, muốn xây dựng hệ thống nhà phát triển cần có kiến thức về các công nghệ liên quan khác bao gồm cả HTML5, đàm thoại trực tuyến VoIP và các công nghệ network có liên quan. Nhiều trong số này là các công nghệ mới được ra mắt trong vài năm gần đây hoặc vẫn đang được phát triển. Phần này trình bày tóm tắt về các công nghệ trên và giới thiệu phương pháp tìm hiểu. Các công nghệ thông dụng như C/C++, PHP hay cơ sở dữ liệu mặc dù cần thiết để xây dựng hệ thống nhưng không được trình bày trong tài liệu này với giả định rằng nhà phát triển đã có kiến thức và kinh nghiệm sử dụng chúng.

* 1. Sử dụng WebRTC trong đào tạo từ xa

Phần này trình bày ngữ cảnh chương trình đào tạo từ xa và giải pháp ứng dụng WebRTC cùng các công nghệ liên quan để xây dựng hệ thống. Phần này gồm

1. Đào tạo từ xa

Giới thiệu và phân tích các đặc điểm của mô hình đào tạo từ xa. Trình bày hai ví dụ về chương trình đào tạo từ xa ở Việt Nam và Thế Giới.

1. Đào tạo từ xa từ góc nhìn kỹ thuật

Trình bày một số vấn đề kỹ thuật gặp phải trong quá trình đào tạo từ xa.

1. Sử dụng WebRTC/HTML5 trong hệ thống đào tạo từ xa

Trình bày các vấn đề kỹ thuật có thể sử dụng WebRTC/HTML5 để cải tiến. Phần này trình bày giải pháp sẵn có và so sánh với giải pháp sử dụng WebRTC/HTML5 để thấy được lý do nên chuyển sang công nghệ mới.

* 1. Kết quả

Trình bày về kết quả thu được sau quá trình thực hiện đồ án và minh họa một số demo thể hiện công nghệ WebRTC/HTML5.

Phụ lục

1. Chú thích thuật ngữ
2. Tài liệu tham khảo

PHẦN I:  
TỔNG QUAN

1. Giới thiệu đề tài
   1. Tóm tắt tình hình công nghệ thế giới

Ngày nay, việc truyền tải và trình chiếu video trên internet trở nên phổ biến. Một phần lớn băng thông internet phục vụ cho việc truyền tải video với sự góp mặt của các website như youtube.com, clip.vn. Đàm thoại trực tuyến có video và âm thanh (video conference) trở nên phổ biến và dần trở thành thói quen của người dùng internet, với các ứng dụng như Microsoft Skype, Apple FaceTime, Google Hangout, Facebook Video Calling... Các doanh nghiệp, chính phủ sử dụng đàm thoại trực tuyến trong các cuộc họp trực tuyến nhằm tiết kiệm chi phí di chuyển và tăng số lượng cũng như hiệu quả họp. Trong tương lai, hội thoại trực tuyến sẽ trở thành một phương pháp giao tiếp chủ đạo của con người.

HTML5 là một tập hợp các chuẩn mới và công nghệ mới trên nền web. Công nghệ WebRTC cho phép việc đàm thoại trực tuyến được thực hiện ngay trong trình duyệt. Công nghệ cung cấp phương pháp thu hình/âm thanh trên thiết bị và truyền tải qua internet. Sử dụng công nghệ WebRTC/HTML5 giúp ứng dụng web có thể tương thích với hầu hết nền tảng desktop và di động, hầu hết hệ điều hành, trình duyệt và các thiết bị di động tương lai, miễn là nền tảng đó hỗ trợ chuẩn HTML5.

Trước đây, Flash là một plugin có mặt trên hầu hết các trình duyệt desktop nhằm thể hiện nội dung media (video, âm thanh và các hoạt cảnh) trên trình duyệt. Hiện nay, với sự phát triển của HTML5, Flash đã không còn là lựa chọn bắt buộc trên các trình duyệt và trên môi trường di động, Flash không còn được hỗ trợ và phát triển. Thẻ <video>, <audio> cung cấp cho trình duyệt khả năng phát tập tin media trên website không cần đến Flash. Còn công nghệ WebRTC mở rộng khả năng của trình duyệt có thể truyền tải nội dung real-time media.

Công nghệ được giới thiệu rộng rãi lần đầu tiên trên Google Chrome vào tháng 1/2012, được hiện thực trên Firefox nightly build vào tháng 4/2012. Các chuẩn vẫn đang được phát triển và hy vọng đến năm 2013 sẽ có các bản hiện thực chuẩn.

* 1. Mục đích đề tài

Hiện nay những hệ thống giảng dạy trực tuyến phục vụ chương trình đào tạo từ xa của các tổ chức đào tạo ngày càng trở nên phổ biến, vừa mang lại sự linh hoạt về thời gian, địa điểm cho cả người dạy lẫn người học, vừa tiết kiệm hàng loạt các chi phí mà mô hình giảng dạy truyền thống phải chi trả.

Một số trường đại học trực tuyến trên thế giới đã đưa ra chương trình đào tạo trực tuyến dành cho tất cả mọi người trên internet hoàn toàn miễn phí

Mục tiêu của đề tài là nghiên cứu và đưa ra giải pháp phát triển hệ thống giảng dạy trực tuyến phục vụ chương trình đào tạo từ xa, ứng dụng công nghệ WebRTC trên nền C++ và HTML5 để trình chiếu video trực tuyến với tốc độ đáp ứng cao và chi phí bỏ ra thấp.

* 1. Phạm vi đề tài

Vì sự hạn chế về thời gian, kiến thức và trang thiết bị. Nên đề tài chỉ nghiên cứu về giải pháp ứng dụng công nghệ WebRTC và các công nghệ liên quan phục vụ cho hệ thống giảng dạy trực tuyến. Bao gồm:

* Nguyên lý hoạt động của WebRTC, cách ứng dụng.
* Nguyên lý hoạt động của codec video và audio, cách ứng dụng.
* Nguyên lý kết nối, quản lý kết nối client – server trong mô hình giảng dạy trực tuyến và cách hiện thực.

Nhóm sẽ không nghiên cứu sâu về kiến thức liên quan đến hạ tầng mạng của hệ thống, kiến thức liên quan đến bảo mật đường truyền hay kiến thức chuyên sâu về mã hóa, giải mã video / audio.

* 1. Các vấn đề cần giải quyết:

Vì đề tài hướng đến một công nghệ của tương lai đang được hiện thực, do đó nhóm đã định hình được những bước thử thách cần vượt qua như sau:

* Bước 1: Tìm hiểu công nghệ, cập nhật các tin, chức năng của công nghệ một cách liên tục và sâu sát nhằm có cái nhìn tổng quát về hệ thống và rõ ràng về công nghệ. Giúp đưa ra giải pháp đúng nhất để hiện thực đề tài.
* Bước 2: Tìm hiểu, hiện thực, thử nghiệm các thành phần cơ bản của công nghệ, viết tài liệu để chuẩn bị cho việc xây dựng hệ thống
* Bước 3: Tiếp tục cập nhật công nghệ, xây dựng kiến trúc hệ thống, định nghĩa các chức năng – giải pháp công nghệ tương ứng.
* Bước 4: Hiện thực hệ thống, áp dụng các kiến thức đã tìm hiểu được. Cập nhật tài liệu để tiếp tục phát triển hoàn thiện và tối ưu hệ thống sau này.
* Bước 5: Phát triển hoàn thiện và tối ưu hệ thống, ứng dụng vào thực tế.

1. Đồ án và hướng phát triển luận văn:
   1. Mục đích đồ án:

Tìm hiểu kiến thức nền tảng, có cái nhìn tổng quan về xu hướng công nghệ và ứng dụng của nó. Chuẩn bị tiền đề cho phát triển, hiện thực đề tài.

* 1. Phạm vi đồ án:

Dựa vào các vấn đề nhận định được, nhóm nhận định đề tài sẽ phải hoàn thành thông qua 3 giai đoạn chính:

* Giai đoạn 1: hoàn thành Bước 1 và Bước 2 của mục 1.4
* Giai đoạn 2: hoàn thành Bước 3 và Bước 4 của mục 1.4
* Giai đoạn 3: hoàn thành Bước 5 của mục 1.4

Trong phạm vi đồ án, nhóm đặt mục tiêu hoàn thành giai đoạn 1 của đề tài.

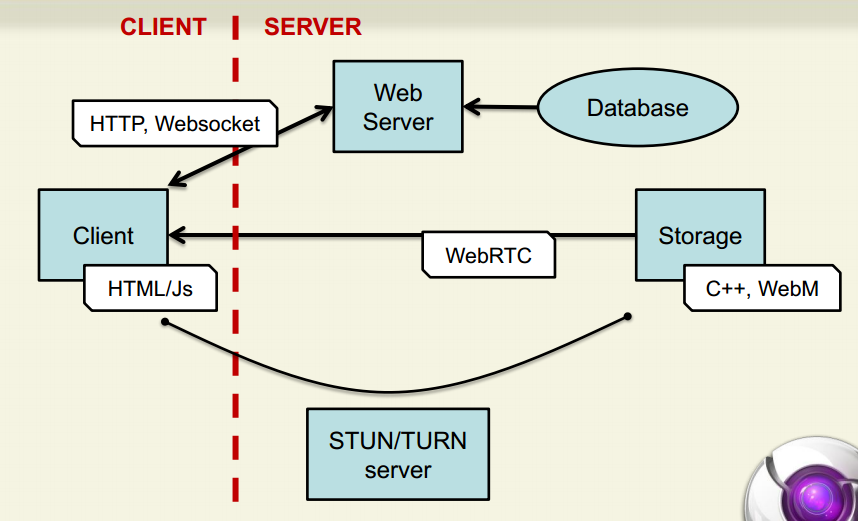
* 1. Phát triển luận văn:

Theo dự tính, nhóm sẽ phát triển đề tài hoàn tất giai đoạn 2 khi nhóm phát triển lên luận văn.

Giai đoạn 3 sẽ được hoàn thành trong giai đoạn sau luận văn.

PHẦN II:  
CƠ SỞ LÝ THUYẾT

1. Khái quát mô hình lý thuyết



1. WebRTC
   1. WebRTC là gì ?

WebRTC là một công nghệ giao tiếp video trên trình duyệt.

WebRTC được chuẩn hóa trong 2 phần: giao thức và cách thức vận hành được điều khiển bởi IETF (RTCWeb); API cho nhà phát triển web được điều khiển trong hệ thống chuẩn của W3C (WebRTC Web API). WebRTC là một chuẩn của HTML5.

Với WebRTC trong một trình duyệt, ứng dụng web bây giờ có thể hướng dẫn trình duyệt tạo ra kết nối real-time voice/video với một trình duyệt khác, một ứng dụng WebRTC hoặc một thiết bị WebRTC sử dụng RTP (Real-time Transport Protocol). WebRTC cho phép nhà phát triển web phát triển các ứng dụng giao tiếp real-time media hoạt động trực tiếp trên trình duyệt mà không cần cài đặt thêm các plugin. Mục đích của nó là tạo ra một nền tảng RTC (Real-time Communication) mạnh giữa các trình duyệt và được thống nhất toàn cầu.

Để nắm bắt công nghệ WebRTC, chúng ta đi qua các khái niệm sau :

* + 1. RTC

RTC viết tắt của Real-time communication, là một nền tảng giao tiếp tích hợp dựa trên chuẩn WebRTC sử dụng các thành phần khác như HTML5/JavaScript, iLBC audio coedc và VP8 video codec.

RTC được hỗ trợ bởi các trình duyệt phổ biến bao gồm Internet Explorer, Firefox, Google Chrome, Opera. RTC cho phép thực hiện cuộc gọi video/voice, trao đổi dữ liệu real-time media cũng như khả năng kết nối với các mạng điện thoại sử dụng (VoIP) SIP protocol.

Bảng 1: Các thành phần của RTC

|  |  |
| --- | --- |
| WebRTC | Thành phần trong trình duyệt |
| HTML5 | Thành phần trong trình duyệt |
| JavaScript | Thành phần trong trình duyệt |
| ROAP | RTCWeb Offer/Answer Protocol |
| JSEP | JavaScript Session Establishment Protocol |
| Internet Expolorer, Firefox, Google Chrome, Opera | Trình duyệt hỗ trợ |
| iLBC, iSAC, G.711, G.722 | Voice codec |
| VP8 | Video codec |

* + 1. WebRTC Web API

Là API lập trình ứng dụng web, nằm trong hệ thống chuẩn của HTML5 được điều khiển bởi W3C. API giúp nhà phát triển web tạo ra ứng dụng truyền tải nội dung real-time media bằng HTML5/JavaScript hoạt động trên trình duyệt.

* + 1. RTCWeb

Là tên gọi các giao thức và cách thức vận hành được điều khiển bởi IETF. RTCWeb định nghĩa cách thức trình duyệt hoặc các thiết bị WebRTC kết nối, thực hiện cuộc gọi video/voice và truyền dữ liệu media. Không chỉ giới hạn trong các trình duyệt, bất kỳ thiết bị hoặc ứng dụng nào hiện thực đúng theo các chuẩn RTCWeb đều có thể thực hiện giao tiếp bằng WebRTC.

* + 1. HTML5

Là phiên bản thứ 5 của ngôn ngữ HTML dùng tạo ra cấu trúc các tài liệu web và ứng dụng web. HTML5 cũng là một tập hợp API mới tích hợp vào trình duyệt mở rộng khả năng của ứng dụng web với nhiều tính năng mạnh mẽ. Một trong số đó là khả năng dựng hình 3D, xem video, quay camera hay đàm thoại trực tuyến ngay trong trình duyệt, điều mà trước đây chỉ có thể thực hiện bằng cách cài đặt thêm plugin. HTML5 tạo ra một nền tảng chung thống nhất giữa các trình duyệt, giúp ứng dụng web có thể hoạt động trên các trình duyệt khác nhau, hệ điều hành khác nhau và các thiết bị khác nhau. HTML5 được điều khiển bởi W3C.

* + 1. JavaScript

Là ngôn ngữ lập trình hoạt động bên trong trình duyệt, kết hợp với HTML và CSS tạo ra các ứng dụng web.

* + 1. ROAP - RTCWeb Offer/Answer Protocol

Là giao thức được thiết kế bởi IETF nhằm cung cấp khả năng khởi tạo cuộc gọi và trả lời cuộc gọi cho WebRTC.

* + 1. JSEP – JavaScript Establishment Protocol

Là giao thức được thiết kế bởi IETF nhằm thiết lập kết nối Peer to Peer trong WebRTC bằng JavaScript. API tương ứng trong WebRTC API được gọi là PeerConnection API.

* + 1. Trình duyệt

Là một phần mềm ứng dụng được thiết kế vởi chức năng nhận và trình bày các tài nguyên web. Các trình duyệt phổ biến bao gồm Internet Explorer, Firefox, Google Chrome, Apple Safari, Opera và các trình duyệt trên thiết bị di động. Đối với ứng dụng web, trình duyệt tạo ra một môi trường để thực thi ứng dụng, kiểm soát cách ứng dụng vận hành và giao tiếp với môi trường bên ngoài (hệ điều hành, mạng internet và thiết bị).

* + 1. Audio codec và Video codec

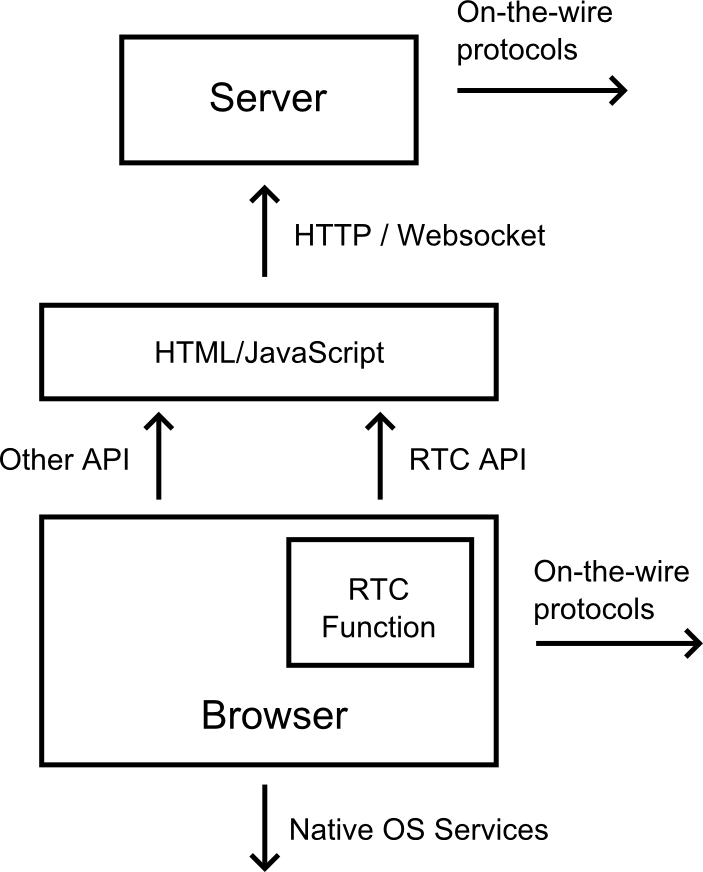
Chuẩn mã hóa và giải mã audio/video được sử dụng trong WebRTC.

* 1. Tình trạng phát triển hiện tại

WebRTC là một nỗ lực khởi động bởi Google, nhằm xây dựng một chuẩn Media Engine dựa trên real-time trên các trình duyệt hiện tại. Sử dụng công nghệ được mua lại từ Global IP Solutions, Google đã tạo ra một phiên bản mã nguồn mở của WebRTC Media Engine và hiện thực nó trong Chrome. Chuẩn WebRTC được phát triển bởi hai tổ chức: IETF RTCWEB Workgroup thành lập vào tháng 04/2011 và W3C WEBRTC Workgroup thành lập vào tháng 05/2011.

Phiên bản thử nghiệm đầu tiên hoạt động trên Google Chrome dev vào tháng 1/2012. Tiếp theo, Mozilla hiện thực công nghệ này trên bản Firefox nightly build vào tháng 4/2012. Microsoft và Opera tuyên bố sẽ hỗ trợ công nghệ này trên trình duyệt của họ. Apple chưa đưa ra bình luận về việc hỗ trợ công nghệ này. Các chuẩn vẫn đang được phát triển và hy vọng đến năm 2013 sẽ có các bản hiện thực hoàn chỉnh.

* 1. Kiến trúc tổng quan WebRTC





Web Server

Browser

Browser

Signaling

path

Web Server

Media path

Application defined over  
HTTP / Websockets



Application defined over  
HTTP / Websockets

* 1. WebRTC có thể làm được gì ?
     1. Giao tiếp trong tương lai

Có các trình duyệt với khả năng real-time sẽ mở ra một loạt các ứng dụng real-time mới. Không thể hoàn toàn dự đoán tất cả các ứng dụng mới, chúng ta có thể thấy trước một vài ví dụ. Rất quan trọng để suy nghĩ rằng đây không chỉ là một công nghệ PC đơn giản. Ngày càng nhiều thiết bị di động và máy tính bảng hỗ trợ trình duyệt WebRTC và thế hệ wireless network thứ 4 sẽ cho phép sử dụng liên tục, công thể này có thể trở thành trung tâm của các giao tiếp qua thiết bị.

Điều này có nghĩa là gì với việc giao tiếp trong 5 năm tới, có thể chỉ đơn thuần là một tính năng thú vị hay là một sự biến đổi hoàn toàn. Rõ ràng đối với các trung tâm liên lạc, sự tác động này rất đáng kể. Với phần lớn trung tâm liên lạc và việc tương tác theo sau việc ghé thăm một trang web, sự thay đổi này sẽ diễn ra ngay lập tức. Với mỗi trang web có một đối tượng read-time communications, luồng từ cấu trúc thông tin trang web vào trung tâm liên lạc có thể được tích hợp chặt chẽ. Điều này sẽ nhanh chóng dẫn đến sự hội tụ của trung tâm liên lạc với đội ngũ phát triển web trong nhiều tổ chức.

Với các nhà phân phối có sẵn, khả năng dễ dàng hỗ trợ ứng dụng khách (soft clients) trên một phạm vi rộng các thiết bị sẽ tăng cường khả năng của họ để thích ứng với sự bùng nổ của xu hướng BYOD (Bring Your Own Device). Đơn cử như, trong môi trường Android, có hơn 50 sản phẩm riêng biệt cần quan tâm của các nhà sản xuất khác nhau. Với WebRTC, nhà phân phối có thể xây dựng một số ít phiên bản HTML cho từng kích thước màn hình thiết bị sử dụng chuẩn chung WebRTC cho media. Nhiệm vụ hỗ trợ một lượng lớn thiết bị được giảm đáng kể.

Có khả năng, WebRTC và HTML5 có thể tạo ra sự chuyển đổi đối với giao tiếp real-time tương tự như cách các trình duyệt đã làm đối với kỷ nguyên thông tin. Năm 1990, thách thức là có các server để chuyển đổi thông tin giữa các cá nhân. Email (và Tin nhắn nhanh - Instant Message) là sản phẩm còn lại của kỷ nguyên này. Sau sự xuất hiện của trình duyệt, quá trình tương tác thay đổi thành việc người dùng cuối truy cập vào trình duyệt nơi lưu trữ thông tin hoặc ứng dụng. Tương tự, WebRTC có thể cho phép tôi truy cập từ trình duyệt của tôi vào server của bạn và giao tiếp với bạn mà không cần đụng đến hệ thống. Kết quả của thay đổi này, kết hợp với việc tìm kiếm địa điểm, đã tạo ra sự biến chuyển lớn của Internet thay đổi các ngành công nghiệp, xã hội và chính trị. WebRTC và trình duyệt hỗ trợ voice/video có thể là khởi đầu của một sự biến chuyển mới. Với sự chứng nhận của LinkIn rằng chúng ta đã kết nối, tôi có thể bắt đầu nói chuyện với bạn trên server của bạn với trình duyệt của tôi.

* + 1. Ứng dụng hội thoại trực tuyến

Ứng dụng đầu tiên và dễ hình dung nhất là cho phép hai trình duyệt mở một kênh giao tiếp.



Web Server

Hình : Mô hình giao tiếp RTC đơn giản

Ở đây một website làm cho WebRTC client trên cả desktop và di động kết nối với nhau bằng một kết nối trực tiếp RTP/IP. Trong trường hợp này, việc điều khiển đến từ webserver, còn dữ liệu media được truyền tải trực tiếp giữa hai thiết bị. Điều này cho phép website kiểm soát việc giao tiếp trên website mà không cần phải cung cấp hay quản lý bandwidth giữa hai thiết bị. Mức ứng dụng này thích hợp để thêm tính năng giao tiếp vào bất kỳ website nào cần tạo liên lạc giữa hai bên.

* + 1. Ứng dụng hội thoại trực tuyến giữa trình duyệt và điện thoại

Một nhà cung cấp dịch vụ điện thoại cho phép khách hàng sử dụng trình duyệt web để truy xuất dịch vụ của họ. Sau một thao tác đăng nhập đơn giản, người dùng có thể gọi và nhận cuộc gọi giống như một điện thoại thông thường; người dùng có thể quay số để gọi hoặc nhận cuộc gọi từ một số điện thoại. Khi một cuộc gọi được tạo hoặc nhận, thông tin từ người đối diện được truy xuất từ danh bạ và hiển thị lên màn hình.

* + 1. Ứng dụng hội thoại trực tuyến khía cạnh doanh nghiệp.

Tương tự với 2.4.2, bổ sung khía cạnh sử dụng dịch vụ cho đối tượng doanh nghiệp.

Một doanh nghiệp sử dụng một ứng dụng WebRTC để giám sát tất cả các phiên WebRTC khác được sử dụng trong công ty tới bất kỳ đơn vị bên ngoài. Để có thể làm điều này, họ triển khai một TURN server nằm giữa ranh giới của mạng nội bộ và mạng bên ngoài.

Firewall này sẽ chặn tất cả các ý định sử dụng STUN với đích bên ngoài trừ khi họ đi qua TURN server để giám sát. Trong trường hợp nhân viên sử dụng ứng dụng WebRTC được cung cấp bởi nhà cung cấp dịch vụ bên ngoài, họ sẽ vẫn muốn dữ liệu di chuyển trong mạng nội bộ và không tải TURN server trên, nên họ triển khai một STUN server cho phép ứng dụng WebRTC xác định địa chỉ của server phải nằm trên mạng nội bộ. Điều này cho phép trường hợp khi hai bên kết nối nằm trong mạng nội bộ thì toàn bộ dữ liệu di chuyển hoàn toàn bên trong mạng nội bộ.

Do đó phải có khả năng cấu hình trình duyệt sử dụng bên trong doanh nghệp với các TURN và STUN server được chỉ định. Điều này có thể được thực hiện bằng cấu hình tự động. Chức năng WebRTC sẽ cần sử dụng cả hai mạng được chỉ định bởi STUN và TURN. STUN và TURN server được cung cấ bởi ứng dụng web.

* + 1. Ứng dụng hội thoại trực tuyến nhiều bên

Tương tự như 2.4.2, được mở rộng bằng cách cho phép nhiều bên tham gia phiên giao tiếp. Không thông qua server trung tâm, trình duyệt của mỗi bên gửi và nhận thông tin đến và từ tất cả các bên tham gia. Ứng dụng web trong trình duyệt của mỗi người dùng khởi tạo kênh dữ liệu cho tất cả các bên.



Web Server



* + 1. Ứng dụng truyền tải media

Bằng việc thêm một Media Server với khả năng hội thoại trực tuyến và các dịch vụ khác, WebRTC bây giờ có thể được dùng để cung cấp hội thoại trực tuyến, như trong hình.



**Web Server**

**Media Server**

Media server này có thể nằm cùng địa điểm với web server hoặc nằm ở các địa điểm riêng biệt. Điều này mở ra một tập ứng dụng mới. Một tổ chức làm việc theo nhóm có thể sử dụng điều này để dễ dàng họp trong một môi trường thích hợp. Với các tổ chức kinh doanh, chính phủ hoặc phi lợi nhuận, điều này có thể tạo ra các nhóm giao tiếp dễ dàng hơn. Trong thực tế, tạo ra một trang web hội thoại trực tuyến đơn giản với HTML5 và WebRTC rất dễ. Và, không cần phải download client hoặc đăng ký vào một nhóm, chỉ cần đơn giản gửi một địa chỉ URL đến tất cả những người tham dự để bắt đầu một cuộc họp. Thêm vào một Media Server mở ra cách cổng cho các dịch vụ cộng thêm như thu hình hay lưu trữ cuộc họp. Một lần nữa, điểm mấu chốt là, việc thêm tính năng này vào website cực kỳ đơn giản, mặc dù cần đầu tư công sức và tiền bạc để phát triển media server.



**Web Server**

**Media Server**



**Media Server**



Có thể thêm vào các media server một cách hỗn hợp và phân tán để có thể phân tải băng thông trên các khu vực khác nhau. Việc này đảm bảo mỗi người dùng có thể kết nối vào media server gần họ nhất và nhận được tốc độ và chất lượng video tốt nhất.

* + 1. Ứng dụng thu hình trực tiếp hỗ trợ ghi hình và phát lại

Các nội dung được thu hình trực tiếp - live stream video - sẽ được sao chép từ máy nguồn vào các media server và được chuyển tải trực tiếp về thiết bị của người dùng.



**Web Server**

**Media Server**



**Media Server**



**Live Video**

Media server có thể cung cấp dịch vụ lưu trữ và phát lại video cho những người dùng muốn xem lại. Nhà cung cấp dễ dàng tùy biến các trải nghiệm người dùng như chèn thêm logo, các đoạn tin tức, quảng cáo và kiểm soát thời gian, cách thức người dùng sử dụng dịch vụ của họ.

* + 1. Mô hình liên lạc giữa các nhà cung cấp dịch vụ khác nhau



**Providers 1**



**Providers 2**



**SIP**

Hai người dùng đăng nhập vào hai ứng dụng web khác nhau, được cung cấp bởi các nhà cung cấp dịch vụ khác nhau. Nhà cung cấp dịch vụ có thể kết nối bằng nhiều cách, nhưng không trao đổi thông tin nào khác ngoài thông tin về người dùng thông qua một giao thức nào đó.

Kết nối giữa các nhà cung cấp dịch vụ nằm ngoài phạm vi của đặc tả WebRTC, được các nhà cung cấp dịch vụ thống nhất sử dụng, dựa trên các giao thức sẵn có (như SIP).

* + 1. On-cloud speech / video recognization

Các media server có thể hoạt động như là một nền tảng hoặc một dịch vụ. Các cách hiện thực có thể khác nhau và cung cấp nhiều khả năng mới như dịch vụ nhận diện giọng nói, video từ thiết bị người dùng. Dịch vụ Google Goggles cho phép người dùng tải ảnh từ thiết bị và tìm kiếm dựa trên nội dung hình ảnh đó. Tưởng tượng một dịch vụ nâng cấp của Google Goggles mà người dùng có thể thu hình real-time từ camera của họ và máy chủ sẽ nhận diện âm thanh, tiếng nói hay phiên dịch và trả về thiết bị.

* + 1. On-cloud real-time game play, real-time rendering.

Một tiềm năng ứng dụng quan trọng của WebRTC nằm ở khả năng chuyển các xử lý phức tạp lên server và nhận về kết quả dưới dạng media trên thiết bị. Thiết bị sẽ đóng vai trò như một remote điều khiển và các game đòi hỏi cấu hình cực cao sẽ được chạy trên máy chủ và phát về thiết bị như nó đang chạy trực tiếp trên thiết bị người dùng. Lúc này thay vì phải đầu tư một dàn máy tính cao cấp, người dùng chỉ cần thiết bị với cấu hình cơ bản và đóng phí dịch vụ hàng tháng. Họ có thể chơi hàng trăm game mà không phải trả tiền mua từng game, không phải mất công sức cài đặt và có thể chơi mọi lúc mọi nơi. Còn nhiều rào cản kỹ thuật cho vấn đề này, nhưng các dịch vụ đang được các nhà cung cấp dịch vụ trên thế giới nghiên cứu và giới thiệu (nguồn).

Một trường hợp khác khi một sinh viên cần truy cập vào ứng dụng dựng hình 3D trên di động. Họ không cần chạy ứng dụng trên máy tính của mình, chỉ cần kết nối vào dịch vụ và điều khiển trực tiếp trên thiết bị di động, nội dung dưới dạng media sẽ được truyền tải trực tiếp về thiết bị. Điều này có thể tạo ra các đoạn phim 3D tương tác chạy trên thiết bị: người dùng ra lệnh và đoạn phim vận hành theo đó (nguồn).

* 1. Chuẩn RTCWeb

Các nổ lực RTCWEB / WEBRTC bao gồm hai phần

* Đặc tả protocol, bởi IETF.
* Đặc tả JavaScript API, bởi W3C.
  + 1. Kiến trúc và nhóm chức năng

Mô hình của việc hỗ trợ real-time cho các ứng dụng web không giả định rằng trình duyệt sẽ bao gồm tất cả tính năng cần thiết để thực hiện các chức năng như gọi hay video-conference. Tầm nhìn là trình duyệt sẽ có các chức năng cần thiết cho một ứng dụng Web, làm việc trong sự kết hợp với server để hiện thực những chức năng này.

Điều này có nghĩa là có hai inteface cần được đặc tả:

* Giao thức các trình duyệt giao tiếp với nhau, mà không cần đến server trung gian.
* API cho các ứng dụng JavaScript để tận dung các chức năng của trình duyệt.

Lưu ý rằng HTTP và Websocket cũng được cung cấp cho ứng dụng JavaScript thông qua API của trình duyệt.

Giống như với tất cả các giao thức và đặc tả API, không có ràng buộc rằng giao thức này chỉ được sử dụng cho việc giao tiếp với trình duyệt khác. Bởi vì nó được đặc tả đầy đủ, bất kỳ thiết bị nào hiện thức giao tiếp này đề có thể tương tác với ứng dụng chạy trong trình duyệt.

Mô hình chung của việc hiện thực như bên dưới



Web Server

Browser

Browser

Signaling

path

Web Server

Media path

Application defined over  
HTTP / Websockets



Application defined over  
HTTP / Websockets

Trên hình này, phần qua trọng cần lưu ý là kết nối media trực tiếp giữa các trình duyệt và kết nối tín hiệu (signaling path) giữa các server có thể chỉnh sửa, dịch hoặc điều chỉnh tính hiệu khi cần thiết.

Nếu hai Web server được vận hành bởi hai tổ chức khác nhau, giao tiếp server cần được thống nhất, bằng việc chuẩn hóa hoặc một thỏa thuận giữa hai bên. Các giao thức sẵn có (như SIP hay XMPP) có thể được sử dụng để giao tiếp giữa server, trong lúc một giao thức (độc quyền) cần được sử dụng giữa browser và web server.

Lấy ví dụ, nếu hai server hiện thực SIP, SIP có thể được sử dụng để giao tiếp giữa các server, có thể cùng với một phương thức tín hiệu chuẩn (như SIP over Websockets) hoặc một phương pháp độc quyền giữa browser và web server. Tương tự, nếu cả hai server hiện thực XMPP (XMPP over Websockets hoặc XMPP over BOSH).

Sự lựa chọn giao thức và định nghĩa việc trao đổi giữa chúng (các giao thức) nằm ngoài phạm vi của chuẩn RTCWEB mô tả trong tài liệu này.

Nhóm chức năng cần thiết trong trình duyệt có thể được mô tả là:

* Data transport (Trao đổi dữ liệu): TCP, UDP và phương pháp bảo mật kết nối giữa các bên, cũng như chức năng quyết định khi nào gửi dữ liệu: quản lý tắc nghẽn, dự trù băng thông, ...
* Data framing: RTP và các kiểu dữ liệu khác được phục vụ như các container với chức năng bảo mật và nhúng dữ liệu.
* Data format: Đặc tả codec, đặc tả định dạng và chức năng của dữ liệu trao đổi giữa các hệ thống. Audio và video codec, cũng như các định dạng dữ liệu và chia sẻ tài liệu. Để sử dụng định dạng dữ liệu, cần một cách để mô tả chúng.
* Quản lý kết nối: Khởi tạo kết nối, thống nhất định dạng dữ liệu, thay đổi định dạng dữ liệu trong quá trình gọi; bao gồm cả SIP và Jingle/XMPP.
* Trình bày và điều kiển: Điều gì sẽ xảy ra để xử lý các sự kiện. Có thể bao gòm điều khiển nền, bố cục màn hình, tắt mở âm thanh và các chức năng khác - nơi một phần của hệ thống cần sự giao tiếp giữa các bên.
* Local system support functions: Có những điều cần được chỉ định thống nhất, vì mỗi đơn vị có thể lựa chọn cách làm riêng, mà không ảnh hưởng đển các bit truyền. Ví dụ như việc định danh, xác thực phương pháp, truy cập hệ điều hành hoặc khả năng thu âm cuộc hội thoại.

Trong mỗi nhóm chức năng, rất cần duy trì cả khả năng tự do đổi mới và đặc tính cho giao tiếp toàn cầu. Tự do đổi mới được hỗ trợ bằng việc chỉ đặc tả các interface, không đặc tả cách hiện thực; bất kỳ hiện thực nào đáp ứng interface đều hợp lệ và có thể giao tiếp với nhau. Khả năng giao tiếp toàn cầu được hỗ trợ bằng việc có các đặc tả cốt lõi và bằng việc có cách định dạng và giao thức được mô tả đầy đủ cho phép các đơn vị độc lập hiện thực.

Một điều có thể nghĩ về ba nhóm đầu tiên tạo nên một hạ tầng truyền tải media, và ba nhóm cuối cùng tạo thành một dịch vụ media. Trong nhiều hoàn cảnh, có thể sử dụng một đặc tả chuẩn cho hạ tầng truyền tải media, có thể được nhúng trong trình duyệt và truy cập bằng các API chuẩn.

Bao gồm cách audio và video codec cần cho client WebRTC. Để đảm bảo một mức tối thiểu giữa các ứng dụng client, một tập tối thiểu các codec được chỉ định ở dưới.

* + 1. Audio codec

PCMA/PCMU: audio một kênh được mã hóa 8 bits per sample, tần số 8 kHz. Thích hợp cho việc truyền giọng nói.

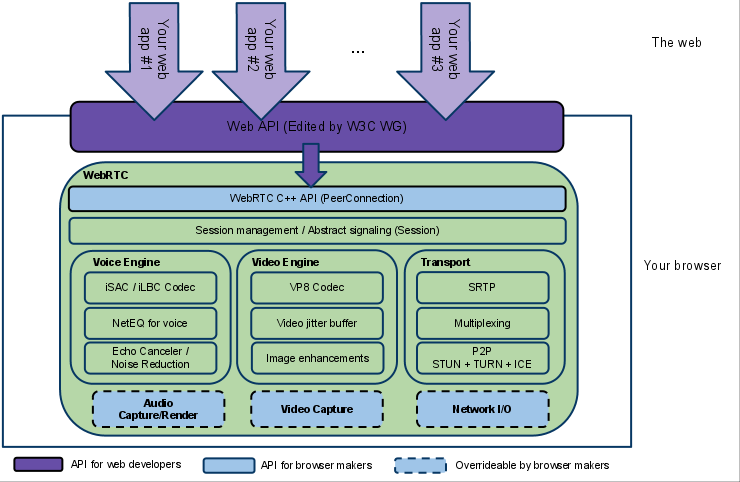
Telephone Event: tín hiệu điện tín.

Opus: codec audio.

* + 1. Video codec

Video codec được hỗ trợ phải đáp ứng các yêu cầu

* Phải hỗ trợ tối thiểu 10 fps và nên hỗ trợ 30 fps.
* Nên hỗ trợ codec VP8.
* Hỗ trợ độ phân giải màn hình tối thiểu 320x240. Các độ phân giải màn hình khuyên dùng là 1280x720, 1024x768, 800x600, 640x480, 640x360, 320x240.
  1. Kiến trúc WebRTC API



Hình 2: Kiến trúc API của WebRTC

Có 2 lớp tách biệt

1. WebRTC C++ API dành cho nhà phát triển trình duyệt

2. Web API dành cho nhà phát triển web

\* Ứng dụng web

Nhà phát triển ứng dụng web với khả năng giao tiếp video và audio sử dụng WebRTC Web API.

\* WebRTC Web API

API dành cho nhà phát triển ứng dụng web được điều khiển bởi W3C.

\* WebRTC Native C++ API

Lớp API cho phép nhà phát triển trình duyệt dễ dàng hiện thực WebRTC Web API.

\* Transport / Session

Sử dụng lại các thành phần từ libjingle, mà không sử dụng/cần đến giao thức xmpp/jingle

RTP Stack: Network stack for Real Time Protocol.

STUN/ICE: thành phần cho phép cuộc gọi sử dụng phương pháp STUN và ICE để thiết lập kết nối qua nhiều kiểu network khác nhau.

Session Management: Lớp trừu tượng, cho phép khởi tạo cuộc gọi và quản lý layer. Nó cho phép nhà phát triển quyết định việc hiện thực protocol.

\* VoiceEngine

Framework để xử lý audio.

- iSAC: Audio codec băng tần rộng và siêu rộng dành cho VoIP và streaming audio. iSAC sử dụng tần số 16 kHz hoặc 32 kHz với bitrate từ 12 đến 52 kbps.

- iLBC: Speech codec băng tần hẹp dành cho VoIP và streaming audio. Sử dụng tần số 8 kHz với bitrate 15.2 kbps cho các khung hình 20ms và 13.33 kps cho các khung hình 30ms. Định nghĩa bởi IETF RFCs 3951 và 3952.

- NetEQ for Voice: Bộ đệm chống nhiễu và giải thuật che giấu lỗi sử dụng để che giấu các tác động xấu từ chất lượng mạng và việc mất gói tin. Giữ độ trễ thấp nhất có thể trong lúc vẫn duy trì chất lượng giọng tốt nhất.

- AEC (Acoustic Echo Canceler): là một thành phần loại bỏ hiện tượng tiếng vang phát sinh do âm thanh được phát từ loa đi vào micro.

- NR (Noise Redution): thành phần loại bỏ một số loại tiếng ồn không liên quan đến cuộc trò chuyện (như tiếng hút gió, quạt quay ...)

\* VideoEngine

- VP8: Video codec từ dự án mã nguồn mở WebM của Google. Thích hợp cho RTC vì nó được thiết kế cho độ trễ thấp.

- Video Jitter Buffer: Bộ đệm chống nhiễu cho video. Giúp giảm tác động của chất lượng mạng và việc mất gói tin.

- Image enhancement: Tăng cường chất lượng hình ảnh, như loại bỏ nhiều video từ hình ảnh thu bởi camera.

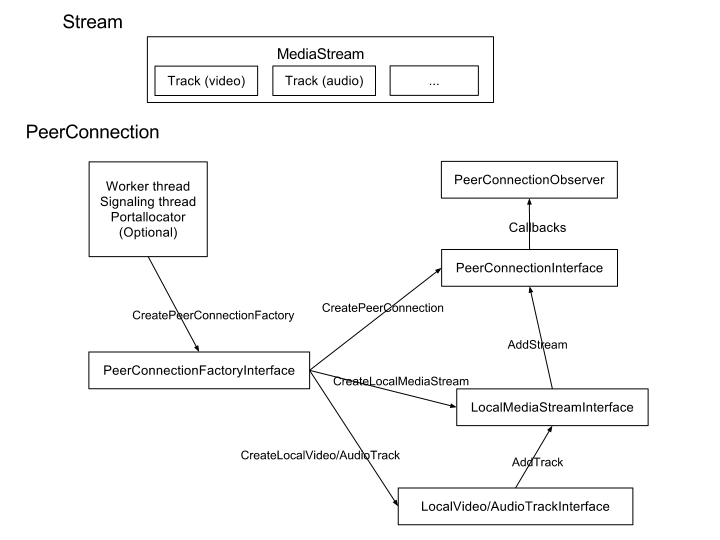
* 1. Mô hình hoạt động của WebRTC Native API

WebRTC Native API sử dụng hai tiến trình global: signaling thread và worker thread. Phụ thuộc vào cách PeerConnection được tạo ra, ứng dụng có thể cung cấp cả hai tiến trình hoặc để chúng được tạo tự động.

Lời gọi đến Stream API và Peer Connection API thông qua một trung gian (proxy) đến signaling thread, nghĩa là ứng dụng có thể gọi nhưng API này từ bất kỳ tiến trình nào.

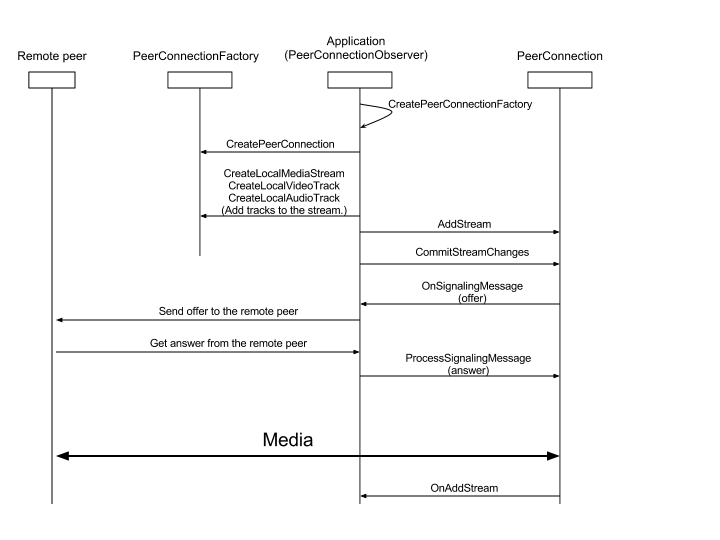
Tất cả các phản hồi (callback) sẽ được tạo ra trên signaling thread. Ứng dụng cần trả lời callback nhanh để trách block signaling thread. Các thao tác cần nhiều tài nguyên nên được thực hiện trên một tiến trình riêng.

* + - 1. Block Diagram



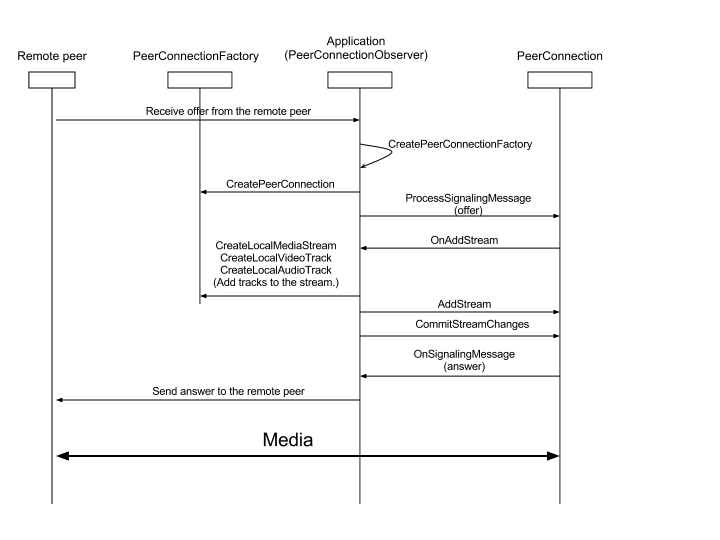
Hình : Block Diagram

* + - 1. Bắt đầu cuộc gọi video



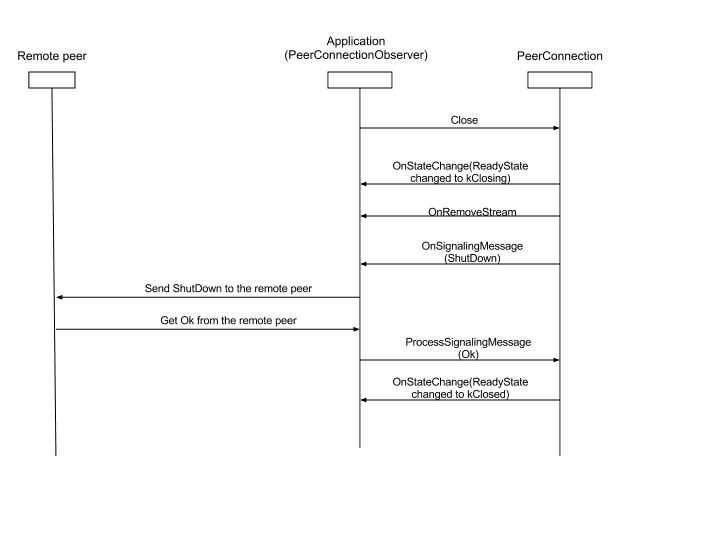
Hình : Bắt đầu cuộc gọi

* + - 1. Nhận cuộc gọi video



Hình : Nhận cuộc gọi

* + - 1. Kết thúc cuộc gọi video



Hình : Kết thúc cuộc gọi

* 1. Các công nghệ liên quan
     1. HTML

HTML là ngôn ngữ mô tả cấu trúc của trang web. HTML cung cấp cho tác giả phương tiện để:

- Xuất bản tài liệu trực tuyến với tiêu đề, chữ, bảng biểu, danh sách, hình ảnh, ...

- Lấy thông tin trực tuyến từ hypertext link, hoặc click vào một nút.

- Thiết kế form nhằm trao đổi với dịch vụ từ xa, như để tìm kiếm thông tin, đăng ký, mua hàng, ...

- Chèn các bảng tính, video clip và các ứng dụng khác trực tiếp trong tài liệu.

Với HTML, tác giả mô tả cấu trúc trang sử dụng markup. Thành phần của ngôn ngữ tương ứng với các phần nội dung như đoạn văn (paragraph), danh sách (list), bảng (table), ...

* + 1. HTML5
    2. CSS

CSS là ngôn ngữ mô tả cách thể hiện của trang web, bao gồm màu sắc, bố cục và kiểu chữ. Nó cho phép một nội dung được trình bày thích hợp với nhiều kiểu thiết bị khác nhau như màn hình lớn, màn hình nhỏ, hoặc máy in. CSS độc lập với HTML và có thể được sử dụng với các ngôn ngữ đánh dấu dựa trên XML. Tách HTML khỏi CSS làm việc duy trì website dễ dàng hơn, chia sẻ các định dạng chung trên nhiều trang, và điều chỉnh trang web thích hợp với nhiều môi trường khác nhau. Điều này được gọi là tách biệt cấu trúc và nội dung khỏi trình bày (the separation of structure / content from presentation)

* + 1. JavaScript

Một script là mã lập trình không cần phải biên dịch trước khi chạy. Trong môi trường trình duyệt, script thường chỉ các mã lập trình được viết bằng JavaScript, thực thi bởi trình duyệt khi một trang web được tải về, hoặc phản hồi lại một sự kiện gây ra bởi người dùng.

Sử dụng script làm nội dung trang web động hơn. Lấy ví dụ, không cần phải tải phiên bản mới của một trang web, script có thể chỉnh sửa nội dung của trang web đó, hoặc cho phép nội dung được thêm vào hoặc gửi đi từ trang web. Đầu tiên nó được gọi là DHTML (Dynamic HTML) và sau đó là AJAX (Asynchronous JavaScript and HTML).

Bên cạnh đó, script dần cho phép nhà phát triển tạo ra một cầu nối giữa trình duyệt và nền tảng trình duyệt đang chạy trên, mở ra khả năng, ví dụ như, tạo một trang web kết hợp các thông tin từ người dùng như vị trí địa lý, thông tin liên lạc, ...

API căn bản nhất được phát triển bởi W3C là DOM (Document Object Model), cho phép chương trình và script có thể truy cập và thay đổi nội dung, cấu trúc và cách trình bày của tài liệu. Đặc tả DOM là phần cốt lõi của DHTML.

Việc chỉnh sửa nội dung sử dụng DOM bởi người dùng và các sự kiện giúp nhà phát triển có thể xây dựng các giao diện người dùng phong phú (rich user interface).

* + 1. - XMLHttpRequest:

Cho phép tải các nội dung từ website mà không cần tạo tài liệu mới, thành phần cốt lõi của AJAX.

* + 1. - Media Capture API:

Cho phép truy cập hình ảnh và âm thanh từ thiết bị của người dùng.

* + 1. Push Notification

Cho phép server chủ động gửi thông điệp đến ứng dụng web bằng sự kiện Server Send Event.

* + 1. WebSocket

Cho phép tạo kết nối stream hai chiều giữa server và ứng dụng web, bổ sung cho kết nối một chiều HTTP truyền thống.

* + 1. Tag <video> và <audio>

Cho phép thể hiện nội dung video và audio trong trình duyệt.

* + 1. Canvas

Cho phép thực hiện các thao tác vẽ hình, xử lý video trong trình duyệt.

* + 1. Offline Application

Cho phép ứng dụng web được thực thi ngay cả khi không truy cập được Internet.

1. Lưu trữ và truyền tải video:

Để hiểu về các định dạng lưu trữ video, ta cần tìm hiểu 02 khái niệm: Tập tin chứa dữ liệu dữ liệu đa phương tiện (Multimedia container formats) và Thuật toán mã hóa / giải mã (Codec).

* 1. Giới thiệu:
     1. Audio & Video là gì ?

Thuật ngữ audio và video thông thường chỉ định dạng lưu trữ nội dung đa phương tiện theo thời gian (time-based media) dành cho âm thanh/nhạc và hình ảnh chuyển động. Việc ghi âm và ghi hình kỹ thuật số - còn gọi là audio và video codec - có thể là dạng thô (uncompressed), nén không mất dữ liệu (lossless compressed) hoặc nén mất dữ liệu (lossy compressed) phụ thuộc vào chất lượng mong muốn và các trường hợp cụ thể.

Audio codec thông thường có thể chứa một kênh audio (mono), hai kênh audio (stereo) hoặc nhiều hơn (như 5.1). Lấy ví dụ, tiếng nói của con người được ghi âm sử dụng một kênh audio trong khi nhạc thường sử dụng hai hoặc nhiều kênh hơn. Chất lượng sẽ khác nhau phụ thuộc vào bitrate, là số bit sử dụng trên mỗi đơn vị thời gian.

Video codec chứa một chuỗi các khung hình (frame), như ảnh và chuyển động giữa các ảnh (đối với định dạng nén). Chất lượng sẽ khác nhau phụ thuộc vào số khung hình mỗi giây (fps - frames per second), không gian màu, độ phân giải, ...

Định dạng lưu trữ video sẽ chứa audio codec stream, video codec stream, phụ đề (caption) và các thông tin khác. Chúng được kết hợp để cung cấp audio hoặc video với các nội dung thay thế hoặc tăng cường. Nhìn chung một video sẽ có một video codec stream, một hoặc nhiều audio codec stream để thay thế lẫn nhau, và có thể có nhiều phụ đề.

* + 1. Streaming Video
       1. Streaming Video là gì

Streaming Video là một phương thức để máy chủ đặc biệt cung cấp video thông qua Internet. Streaming Video thường được sử dụng trong lĩnh vực giải trí hoặc dạy học, dùng để lưu trữ các tuyển tập các tập tin video hoặc các bài học, cung cấp cho người dùng các tiện ích như tìm kiếm, liệt kê, và khả năng hiển thị hoặc hiển thị lại các dữ liệu video theo yêu cầu.

Với các định dạng tập tin video truyền thống, dữ liệu chỉ có thể hiển thị khi đã được tải về (download) toàn bộ, vì vậy đối với các tập tin video chất lượng cao có dung lượng lớn sẽ tiêu tốn rất nhiều thời gian để chờ thực hiện công việc này.

* Streaming Video tiết kiệm thời gian cho người dùng bằng cách sử dụng các công nghệ giải nén kết hợp với player hiển thị dữ liệu đồng thời trong lúc vẫn tiếp tục download. Quá trình này được gọi là buffering:
* Thay vì được gửi một lần duy nhất, dữ liệu streaming video sẽ được truyền đi thành các gói nhỏ.
* Ban đầu player sẽ lấy về một phần chia nhỏ đó của dữ liệu video trước khi hiển thị, đồng thời trong lúc hiển thị các gói dữ liệu còn lại sẽ lần lượt được lấy về để kịp cho việc hiển thị tiếp theo.
* Streaming Video được thể hiện dưới hai dạng:
* Video theo yêu cầu (on demand): là các dữ liệu video được lưu trữ trên multimedia server và được truyền đến người dùng khi có yêu cầu, người dùng có toàn quyền để hiển thị cũng như thực hiện các thao tác (tua, dừng, qua …) với các đoạn dữ liệu này.

Video thời gian thực (live event): là các dữ liệu video được phát trực tiếp từ các nguồn cung cấp dữ liệu theo thời gian thực (máy camera, microphone, thiết bị phát dữ liệu video…).

* + - 1. Quá trình Video Streaming

Video Streaming là thuật ngữ dùng để nói về quá trình máy chủ cung cấp video thông qua Internet. Với Streaming, máy tính của người sử dụng không phải tải toàn bộ video về cùng một lúc. Để thực hiện việc tăng tốc độ lần tải về, tại bất kỳ thời điểm nào máy tính chỉ cần tải về một phần thông tin của video.Với kỹ thuật này, chúng ta cần một máy chủ đặc biệt kiểm soát việc cung cấp các nội dung video. Để bất kỳ phần nào của đoạn video được lưu trữ trên server có thể được truy cập bất kỳ lúc nào, hơn là cần nó chờ để tải về trước khi truy cập nó.

* + - 1. Video Streaming:
* True Streaming: tín hiệu video đến theo thời gian thực và hiển thị ngay lập tức cho người xem.
* Download and Play: Tải toàn bộ video về, sau đó phát video.
* Progressive Download and Play: Đây là công nghệ lai giữa hai công nghệ trên. Trong công nghệ này, video sẽ được chia nhỏ ra thành nhiều phần nhỏ; chương trình phát video (player) sẽ hiện thị ngay những đoạn (segment) vừa được tải về và lưu trữ dữ liệu đó ở ổ đĩa cục bộ của máy. Khi tua lại những đoạn video mà đã tải về, Player sẽ lấy dữ liệu từ ổ đĩa cục bộ để tải lên.

Video Streaming sử dụng các giao thức RTP, RTMP, MMS, HTTP,.. để truyền dữ liệu theo dạng streaming qua mạng Internet, đồng thời sử dụng các chuẩn nén để giảm dung lượng dữ liệu, cung cấp khả năng nén dữ liệu tại nhiều mức nén, nhiều kích thước hiển thị để có thể phù hợp với độ rộng băng thông của nhiều mạng truyền dẫn để tối ưu hoá việc truyền dữ liệu qua mạng. Cũng chính vì vậy việc truyền các Streaming video qua mạng sẽ phụ thuộc rất nhiều vào các sản phẩm phần mềm video Streaming Server.

* + - 1. Kiến trúc hệ thống Streaming

Khi hiện thực một hệ thống Streaming trên một mạng IP (hoặc một vài kỹ thuật khác), chúng ta cần đầu tư cho hệ thống một Streaming Server (có nhiệm vụ phân phối Stream cho mỗi thiết bị người dùng).

Streaming Server hoạt động sẽ lấy nội dung (những tập tin video hoặc những tập tin khác trong hệ thống) và tạo Stream cho mỗi yêu cầu người dùng gửi đến. Những Stream này có thể hiện thực theo hai phương thức (mỗi phương thức được điều khiển bởi cơ chế khác nhau) là unicast và multicast. Phổ biến nhất hiện nay là cơ chế unicast.

Chức năng thường được tích hợp trong Streaming Server là lưu trữ và phục hồi nội dung. Công việc xử lý quan trọng của Streaming Server là tạo một gói tin cho mỗi Stream theo thời gian thực, giải quyết các yêu cầu về mã hóa gói dữ, nén dữ liệu để gửi đến các kênh Stream khác nhau. Một vấn đề quan trọng khác trong hệ thống là phân tích dữ liệu và đưa ra giải thuật giải quyết vấn đề quản lý bộ nhớ (Cache), đảm bảo hiệu năng hệ thống một cách hiệu quả trong Streaming Server.

* 1. Các phương pháp lưu trữ và truyền tải video
     1. Tập tin chứa dữ liệu dữ liệu đa phương tiện (Multimedia container formats):

Ta tạm gọi các tập tin chứa nội dung đa phương tiện là các container. Các container chứa thông tin nhận diện về codec và dữ liệu xen kẽ giữa nhiều định dạng media khác nhau (tùy vào đặc tả của mỗi container). Các loại container đơn giản có thể chứa nhiều định dạng âm thanh khác nhau, trong khi các định dạng container nâng cao có thể chứa nhiều luồng (stream) âm thanh, video, phụ đề, thông tin các chương, và meta-data (tags) - kèm theo đó là thông tin để đồng bộ các luồng dữ liệu nhằm mục đích hiển thị, chơi cùng lúc các luồng đó. Các định dạng container được thiết kế riêng cho các mục đích khác nhau, các định dạng dành cho truyền dữ liệu qua internet sẽ khác với các định dạng chất lượng cao v.v…

Trong mỗi container, ta có các gói (packet) dữ liệu để chứa dữ liệu. Các định dạng khác nhau gọi tên các gói đó khác nhau, trong RIFF và PNG là "chunks, trong QuickTime/MP4 là "atoms", trong MPEG-TS là "packets" và là "segments" trong JPEG. Nội dung bên trong các gói dữ liệu đó được gọi là "data" hay "payload". Hầu hết các định dạng container lưu trữ các gói trong sequence, mỗi gói có header riêng, ngoại trừ TIFF lưu theo offsets. Cách lưu trữ này giúp chúng ta dễ dàng khôi phục trong các trường hợp lỗi file, mất hình (drop frame),…

* Một vài container phổ biến đã được đăng ký riêng cho định dạng âm thanh:
* AIFF (định dạng IFF, sử dụng trên nền tảng Mac OS)
* WAV (định dạng RIFF, sử dụng trên nền tảng Windows)
* XMF (Định dạng âm thanh mở rộng)
* Một và container khác thì được đăng ký riêng cho hình ảnh:
* FITS (Flexible Image Transport System) hình ảnh tĩnh, dữ liệu thô và metadata.
* TIFF (Tagged Image File Format) hình ảnh tĩnh và metadata.
* Các container có thể chứa nhiều loại âm thanh, video và các loại dữ liệu đa phương tiện khác phổ biến hiện nay (Multimedia file container) như:
* 3GP (Sử dụng trên hầu hết các thiết bị di động; xây dựng dựa vào chuẩn ISO)
* ASF (được dùng để chứa 2 định dạng là WMA và WMV của Microsoft, tuy nhiên, ngày nay rất ít được dùng vì WMV đã có thể “tự chứa” được WMA)
* AVI (định dạng container chuẩn của Microsoft, xây dựng trên RIFF)
* DVR-MS ("Microsoft Digital Video Recording", được phát triển bới Microsoft, dựa trên ASF)
* Flash Video (FLV, F4V) (chứa video và audio của hãng Adobe Systems)
* IFF (định dạng container độc lập với nền tảng đầu tiên)
* Matroska (MKV) (Định dạng này là một chuẩn mở (open standard), mã nguồn mở, có thể chứa không giới hạn các định dạng media khác).
* MJ2 – Định dạng Motion JPEG 2000 xây dựng dựa vào chuẩn ISO về file media được đặc tả qua MPEG-4 Part 12 và JPEG 2000 Part 12
* QuickTime File Format (chuẩn video QuickTime của Apple Inc.)
* MPEG program stream (chuẩn container cho MPEG-1 và MPEG-2; thường sử dụng trên các DVD-Video hay VCD)
* MPEG-2 transport stream (còn gọi là MPEG-TS) (chuẩn container cho các định dạng chất lượng cao, có thể chứa nhiều luồng dữ liệu, chương trình điều khiển,… được sử dụng trên các đĩa Blue-ray hiện nay)
* MP4 (chuẩn container cho các định dạng đa phương tiện MPEG-4, đặc tả trong MPEG-4 Part 12 và JPEG 2000 Part 12).
* Ogg (chuẩn container âm thanh Vorbis của Xiph.org và định dạng video Theora)
* RM (RealMedia; chuẩn container dành cho RealVideo và RealAudio)
* WebM (Chuẩn container mở, xây dựng trên MKV, được hiện thực cho nền tảng HTML5)

Ngoài ra, còn rất nhiều định dạng container phổ biến khác như NUT, MXF, GXF, ratDVD, SVI, VOB và DivX,… mà trong giới hạn của đồ án, chúng tôi không thể giới thiệu hết.

* + 1. Thuật toán mã hóa (Codec = Compressor-Decompressor):
       1. Codec là gì?

Tổng quát, Codec là một thiết bị phần cứng hoặc là một chương trình phần mềm cho phép mã hóa và giải mã các luồng dữ liệu số hoặc là tín hiệu. Codec là từ được kết hợptừ Compressor-Decompressor, hoặc là Coder-Decoder. Về mặt phần cứng, có thể lấy ví dụ chip xử lý âm thanh là một minh họa về codec. Codec phần cứng được thực hiện bằng các mạch điện tử được thiết kế chuyên biệt nên được thực hiện rất nhanh, tuy nhiên lại kém linh hoạt. Ta không thể sử dụng card âm thanh để xử lý tín hiệu…video được (!). Còn codec phần mềm được thực hiện bằng “công cụ vạn năng” là CPU nên hết sức mềm dẻo và linh hoạt. Tuy nhiên, xử lý bằng phần mềm không thể nhanh và tối ưu bằng phần cứng, nhưng với tốc độ của các CPU hiện nay, hầu như không có loại codec nào có thể “làm khó” máy tính của bạn.

* + - 1. Tại sao phải sử dụng Codec?

Codec thực tế có nhiều ứng dụng, tuy nhiên, ở đây chúng ta chỉ xét trong lĩnh vực đa phương tiện – multimedia. Hẳn chúng ta ai cũng đã biết, các file multimedia nguyên gốc rất lớn. Lấy ví dụ về âm thanh. Một file wav (âm thanh chưa nén) chuẩn (lấy mẫu 16-bit ở tần số 44.1Khz) với độ dài khoảng 5 phút sẽ có dung lượng vào khoảng 50MB. Với video, con số này sẽ “khủng khiếp” hơn rất nhiều vì mỗi giây phải lưu giữ ít nhất 24 khung hình, mà mỗi khung hình, nếu theo chuẩn BITMAP chưa nén sẽ là ~2,6MB cho kích thước HD-720p. Điều này gây ra nhiều khó khăn trong việc lưu trữ cũng như phát hành các file đa phương tiện qua mạng. Codec chính là giải pháp tuyệt vời cho trường hợp này. Với Codec, bạn có thể thu nhỏ các file multimedia xuống nhiều lần mà chất lượng vẫn giữ nguyên hoặc chỉ suy giảm chút ít. Chẳng hạn, với file âm thanh ở trên, nếu được mã hóa dưới dạng FLAC (một chuẩn âm thanh) thì dung lượng chỉ còn 20-25MB. Nếu chấp nhận bỏ bớt 1 chút chất lượng âm thanh, có thể dùng định dạng MP3 để thu nhỏ xuống còn 10 MB (với bitrate 320kbps) hay thậm chí hơn nữa…

Để chơi được các file multimedia, player phải thực hiện 2 công việc: đọc được header của file để nhận diện xem đó file đó thuộc loại nào (video/audio), sử dụng chuẩn mã hóa nào (DivX/Xvid, x264 hay Quicktimes,…). Và sau đó là giải mã nội dung thật sự của file bằng trình giải mã tương ứng để đưa ra trình diễn.

* + - 1. Các loại codec video:

Có khá nhiều loại codec video khác nhau. Trong quy mô đồ án, chúng tôi sẽ chỉ giới thiệu các codec phổ biến nhất, nổi bật nhất hiện nay. Cụ thể:

* Chuẩn MPEG-4

MPEG-4 là một chuẩn nén video được phát triển bởi một nhóm gọi là “Moving Picture Experts Group” (hay gọi là MPEG). MPEG-4 được biết đến như là một “chuẩn nén của hình ảnh & âm thanh bit-rate thấp”. Có rất nhiều chuẩn được phát triển bởi nhóm này và đã được chấp nhận bởi tố chức tiêu chuẩn quốc tế (ISO (International Standards Organization).

MPEG-4 là một chuẩn nén video với rất nhiều các phần mở rộng mà đặc biệt được thiết kế để đạt tới chất lượng cao nhất của chuẩn nội dung video. Chuẩn nội dung ở đây là chuẩn các video trong thế giới thật, nó không được thiết kế để nén video dựng bởi các trình 3D, phim hoạt hình trên TV thông thường hay là anime (đó là lý do tại sao các video thuộc thể loại trên nén không tốt lắm với MPEG-4).

Các codec thuộc họ MPEG-4 là DivX, 3ivx, Quicktime MPEG4 và XviD. Mặc dù chúng là họ hàng với nhau, tuy nhiên một phim được nén bởi một trong các codec trên sẽ không thể được giải mã bằng codec anh em của nó. Vì giữa chúng vẫn có những điểm khác nhau cơ bản không thể thay thế. Lấy ví dụ bằng việc xử lý tập tin video có nhiều hơn 1 tỷ frame – đó là việc mà DivX không thể thực hiện được, trong khi XviD lại có thể. Ví dụ khác sẽ là XviD có đến 3 điểm warppoint GMC (Global Motion Compensation) trong khi DivX chỉ có 1 điểm – điều này dẫn đến sự khác biệt trong cấu trúc phần cứng để có thể giải mã được đoạn video được nén bằng codec DivX hay XviD (phần cứng để giải mã DivX sẽ nhẹ nhàng hơn).

Cho đến nay, hầu hết các đầu đọc DVD phù hợp chuẩn MPEG-4 có thể đọc được video DivX. Để mở rộng khả năng đọc các chuẩn MPEG-4 khác sẽ không khó khăn mấy, việc này chỉ cần được thực hiện bằng một số thay đổi nhỏ trong EEPROM liên kết với bộ xử lý video của đầu đọc – hay nói cách khác là flash lại bộ EEPROM của đầu đọc.

Để có thể nén video với dung lượng thấp nhưng vẫn gữ được chất lượng cao đòi hỏi cả một hệ thống giải thuật phức tạp. MPEG-4 làm việc này bằng cách bỏ bớt những thông tin mà người xem không cảm nhận được bằng cách biến đổi dữ liệu về các điểm ảnh trên video thành công thức toán học gần giống với đoạn dữ liệu đó. Việc biến đổi này gần giống nhau đến mức con người bình thường sẽ rất khó có thể nhận ra sự khác biệt giữa video nguồn và video kết quả (tất nhiên, đó chỉ là lý thuyết).

Đầu tiên, mảng màu của khung hình ban đầu sẽ được biến đổi thành một mảng màu đặc biệt được gọi là YV12. Mắt người thông thường kém nhạy cảm với các màu sắc hơn là với các mức độ sáng tối (liếc nhìn thật nhanh qua bầu trời, bạn sẽ cảm nhận rõ nhất trời sáng hay trời tối hơn là bầu trời có màu gì). Chính vì vậy, MPEG-4 đánh giá độ sáng tối của hình ảnh quan trọng hơn là màu sắc. Điều này đem đến một kết quả là độ sáng tối được ghi lại trong mỗi pixel còn thông tin màu sắc sẽ chỉ được ghi lại sau mỗi 4 pixel.

Các codec thuộc họ MPEG-4:

* DivX

Đây là codec ra đời sớm nhất, nổi tiếng và rất rất phổ biến trong họ MPEG-4, như đã nói ở trên, có thể coi DivX đã lên dạng lão làng khi có hầu hết đầu đọc DVD đọc luôn cả DivX. So sánh giữa chuẩn MPEG-2 và DivX thì chất lượng có thể coi là như nhau có điều DivX trội hơn ở bit-rate thấp, điều này đồng nghĩa với việc chúng ta có thể có một đoạn phim tương tự DVD video nhưng nhẹ hơn chỉ cỡ bằng 1/2 của DVD video.

Tuy phổ biến như vậy nhưng so với các codec chị em của nó gần đây thì DivX bắt đầu tỏ ra yếu thế, đặc biệt là trước XviD và H264 và một số codec đang phát triển khác.

Trang chủ DivX: http://www.divx.com

* XviD

XviD là một codec đi theo chuẩn MPEG-4 part 2 nên nó còn được biết đến như là MPEG-4 chuẩn ISO #14496-2. Nó có thể nén một file video thành những dữ liệu theo chuẩn của MPEG-4 và có thể được lưu trữ dưới dạng container .AVI .OGM .MP4 hay khác. Bản thân codec này không thể tự nén video mà nó phải nhờ một công cụ khác giúp nó, đó có thể là trình VirtualDub.

Xvid codec là một dự án mã nguồn mở GNU-GPL. Nghĩa là mã nguồn của nó có thể được download bởi bất kỳ ai và chỉnh sửa theo bất kỳ ý thích nào của họ. XviD là chuỗi kí tự viết ngược của DivX (phát triển bởi một công ty thương mại) – có lẽ nó giống như một sự thách thức với codec DivX. Sự thật cho thấy XviD vượt trội về chất lượng nén hơn hẳn DivX.

Trang chủ XviD codec: http://www.koepi.info/xvid.html

* H.264

H264 là codec thể hiện rõ nhất sức mạnh của họ MPEG-4. Sức mạnh đó có thể thể hiện qua một đoạn anime dài 24 phút, phân giải 704×400 pixel với âm thanh MP3 128kbps mà chỉ có 76.2MB (video bit-rate là 301kbps) – chất lượng của nó gần bằng đoạn anime tương tự như thế nén với DivX nhưng nặng vào cỡ 233MB!

Trong một vài trường hợp, H264 có thể mang lại một đoạn video đẹp hơn cả video gốc ban đầu từ MPEG-1/ MPEG-2 (?!). Nguyên nhân của sự “lạ lùng” ấy là do H.264 sử dụng một thuật toán làm mờ đi các chi tiết mà nó thấy không cần thiết trong các khung hình. Các khung hình được de-cube - xoá nhoà các vết ô vuông của codec MPEG-1 tạo ra - mang lại cho người xem cảm giác hình ảnh mịn hơn, màu sắc đẹp hơn.

Mặc dù đây là phiên bản tính phí (chi phí sử dụng từ MPEG-LA khá cao), nhưng hiện có một project giúp H264 đến người dùng cá nhân miễn phí đó là compressor / decompressor x264.

Chi tiết tại: http://www.videolan.org/developers/x264.html

* RM/RMVB

Codec rất quen thuộc và phổ biến nhất ở Trung Quốc có dạng \*.rmvb. Đây là codec riêng từ RealOne. Loại này cho ra ảnh phim rõ nét, sáng đẹp và dung lượng của file khá nhỏ, đặc biệt có chút nhỉnh hơn DivX và XviD nếu như rip từ DVD. Tuy nhiên, codec này có mã nguồn đóng, và tài liệu về nó không được RealOne cập nhật rõ ràng lắm.

Trang chủ: http://real.com

* VP8

Kể từ khi Google quan tâm tới VP8 bằng việc mua lại On2, VP8 bắt đầu trở thành đề tài nóng. Tháng 5-2010, một chuyện được xem là “động trời” với HTML5, khi Google mở mã VP8, miễn phí sử dụng, cùng với định dạng container mới WebM và dự án www.webmproject.org.

Đây cũng chính là codec video mà đồ án quan tâm, là codec video ***duy nhất*** được đặc tả trong công nghệ webRTC.

Với VP8, người dùng cuối không cần quan tâm tới các Plugin hay cài các codec khác mà được bản thân các trình duyệt thế hệ mới như Mozilla Firefox, Opera, Google Chrome và IE9 sẵn sàng hỗ trợ, trong khi Safari cũng hỗ trợ bất kì codec nào cài vào QuickTime và tương lai cũng sẽ cho phép WebM.

Epiphany đã bắt đầu hỗ trợ WebM thông qua nền tảng Gstreamer. Các trình phát đa phương tiện như VLC, Miro, Moovida, Winamp, Mplayer cũng đã sẵn sàng hỗ trợ. Nhóm phát triển Ffmpeg cũng nhanh chân hỗ trợ VP8. Theo thống kê, hiện có đến gần 50 đối tác, nền tảng đã sẵn sàng với WebM và VP8, trong đó có Android (công bố vào quý 4 – 2010), AMD, ARM, Broadcom, Intel, Qualcomm, NVIDIA, Sorenson Media, Logitech,…

Đây được xem là codec đáp ứng đủ 2 yếu tố tối ưu và miễn phí, được hỗ trợ trực tiếp với nền tảng HTML5 và hứa hẹn sẽ là codec thay thế các codec “già yếu” trước đây để trở thành đối thủ nặng ký của H264 và flash.

Trang chủ: http://www.webmproject.org

* + - 1. Các codec audio:
* MP3:

Nhắc đến codec audio, không thể không nhắc đến MP3 – codec phổ biến nhất thế giới hiện nay.

MP3 là một dạng file đã được nén bằng cách nén lossy (có mất mát dữ liệu). Nó là một dạng âm thanh PCM (pulse-code modulation-encoded) nhỏ hơn rất nhiều so với dữ liệu ban đầu do bỏ đi những phần âm thanh được cho là không quan trọng trong khoảng nghe được của con người, tương tự như cách nén JPEG dành cho hình ảnh.

Tên của dạng này bắt nguồn từ "MPEG-1, lớp âm 3", còn được gọi chính thức hơn là ISO/IEC 11172-3 lớp 3. Những tập tin theo dạng này được lưu với phần mở rộng tên .mp3. Đôi khi những tập tin theo tiêu chuẩn MPEG-2, lớp âm 3 cũng sử dụng phần mở rộng này.

Có rất nhiều kỹ thuật đã được dùng trong chuẩn nén MP3 để xác định phần nào nên bỏ đi, trong đó có tâm thần âm học (psychoacoustic). Dữ liệu MP3 có thể được tạo ra với nhiều bitrate khác nhau để có thể dễ dàng chọn lựa giữa chất lượng cao hay cỡ tập tin đầu ra nhỏ.

Cốt lõi của kỹ thuật nén MP3 là một dạng biến đổi phức để chuyến tín hiệu sóng ngang theo thời gian thành tín hiệu dựa theo tần số.

MP3 vòm là một định dạng MP3 hỗ trợ 5.1 kênh cho âm thanh vòm, được giới thiệu vào tháng 12 năm 2004. MP3 vòm có tính tương thích ngược với chuẩn MP3 trước đây, và kích cỡ file sau nén cũng tương tự.

Tuy nhiên đây là codec âm thanh mã nguồn đóng.

* FLAC (Free Lossless Audio Codec)

FLAC là viết tắt của Codec Free Audio Lossless, đây là định dạng âm thanh nén tương tự như MP3, nhưng không giảm chất lượng, có nghĩa là âm thanh được nén FLAC không có bất kỳ tổn thất nào về chất lượng (lossless). Điều này là tương tự như với cách nén Zip một file như thế nào, tuy nhiên với FLAC dữ liệu được nén tốt hơn nhiều bởi vì nó được thiết kế đặc biệt cho âm thanh, và ta có thể chơi lại tập tin nén FLAC trong máy nghe nhạc hoặc xe hơi hoặc dàn âm thanh nổi tại nhà giống như một tập tin MP3.

Là codec mã nguồn mở hỗ trợ nhạc Lossless (định dạng âm thanh chất lượng cao) nhanh và tốt nhất hiện nay với đầy đủ tài liệu, đặc tả API,…

Trang chủ: http://flac.sourceforge.net/

* AAC (Advanced Audio Coding):

Advanced Audio Coding (AAC) - (ISO 14496-3) là một định dạng âm thanh đa năng nén kiểu lossy được định nghĩa theo tiêu chuẩn MPEG-2 và được phát triển bởi liên minh Fraunhofer, Dolby, Sony và AT&T. AAC được phát triển nhằm thay thế cho định dạng âm thanh đã quá nổi tiếng MP3 để tích hợp trong container MP4-một container của MPEG-4 tiêu chuẩn hỗ trợ đầy đủ các tính năng phụ.

Dạng định này được phát triển để xóa đi những chỗ yếu của MP3 và nâng cao phương pháp mã hóa đã có. Do vậy những tín hiệu thu của âm thanh hay tiếng động sẽ được nhận biết và mã hóa 1 cách hiệu quả hơn hoặc những vấn đề của Pre-Echo sẽ giảm thấp xuống nhiều.

AAC có thể tích hợp tới 48 kênh âm thanh (có sample rate tới 96KHz) cộng thêm 15 kênh âm thanh tần số thấp (Low Frequency Enhancement-LFE) giới hạn sample rate ở 120 Hz.

AAC là dạng định nén lossy, được sử dụng rộng rãi qua các kênh mua bán nhạc trực tuyến như iTunes Store, Real Music Store, LiquidAudio được gắn kèm với hệ thống chống sao chép DRM (ví dụ như FairPlay của Apple).

* Vorbis:

Vorbis là một dự án miễn phí mã nguồn mở của Xiph.Org. Đây là một định dạng nén âm thanh theo kiểu lossy có chất lượng và mức độ nén gần tương đương với AAC, thường được chứa trong container OGG.

Thư viện của codec này đang được cập nhật liên tục với tài liệu cũng như API khá đầy đủ. Bộ thư viện mã nguồn mở theo chuẩn GNU.

Đây là codec được sử dụng trong container WebM để tạo ra file multimedia dành riêng cho HTML5. Đã được hỗ trợ bởi hầu hết các trình chơi âm thanh và trình duyệt web.

Vorbis có mức nén gần tương đương với MP3, và hiện tại đang được Google xúc tiến tích hợp vào WebM trên nền HTML5. Có thể trong tương lai, codec này sẽ phổ biến tương đương với MP3.

* iSAC / iLBC:

Như đã trình bày ở phần 1.1, đây là 2 codec băng thấp được hỗ trợ trong hệ thống RTC để chuyển tải nội dung thoại (voice và speech). Trong đó:

* iSAC (internet Speech Audio Codec) là codec mã hóa âm thanh wideband, phát triển bởi Global IP Solutions (GIPS) được mua lại bởi Google Inc vào năm 2011. Nó phù hợp cho các ứng dụng VoIP và âm thanh trực tuyến. [http://en.wikipedia.org/wiki/Internet\_Speech\_Audio\_Codec]
* iLBC (Internet Low Bitrate Codec) là một codec mã hóa âm thanh narrowband phát triển bởi Global IP Solutions (GIPS) được mua lại bởi Google Inc năm 2011 và kể từ đó, nó trở thành một thành phần của dự án WebRTC. iLBC phù hợp cho các ứng dụng VoIP, truyền tải âm thanh, lưu trữ và tin nhắn. Thuật toán là một phiên bản của mã hóa dự đoán tuyến tính khối độc lập (block-independent linear predictive coding), với sự lựa chọn của độ dài khung dữ liệu của 20 và 30 phần nghìn giây. Các khối mã hóa phải được đóng gói trong một giao thức thích hợp cho vận chuyển, thường là Real-time Transport Protocol (RTP).

[http://en.wikipedia.org/wiki/Internet\_Low\_Bit\_Rate\_Codec]

* Opus:

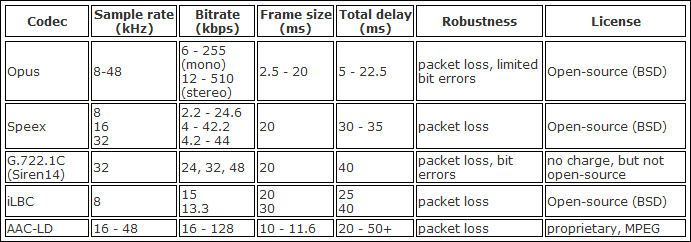
Đây là codec âm thanh được đặc tả trong RTCWeb bởi IETF và là codec âm thanh chất lượng cao có độ trễ thấp được quan tâm trong phạm vi đồ án.

Codec Opus được thiết kế cho đàm thoại trực tuyến và truyền tải âm thanh qua Internet. Nó được thiết kế bởi IETF Codec Working Group và kết hợp công nghệ từ codec SILK của Skype và Celt của Xiph.Org.

Codec này đã vượt mặt các lão làng trong hệ thống codec audio như MP3, AAC,… về dải tần, chất lượng và độ trễ, thể hiện khá rõ vào tháng 3-4/2011 thông qua bài kiểm tra của HydrogenAudio.

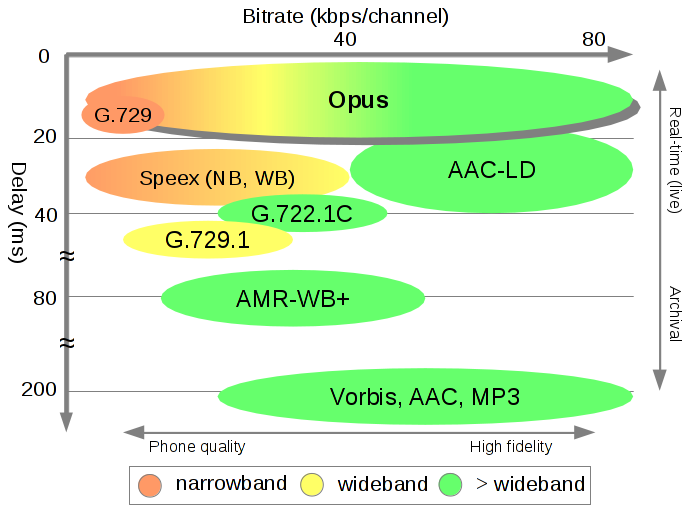
Bảng so sánh về đặc điểm với các loại codec khác:

[http://www.opus-codec.org/comparison/]

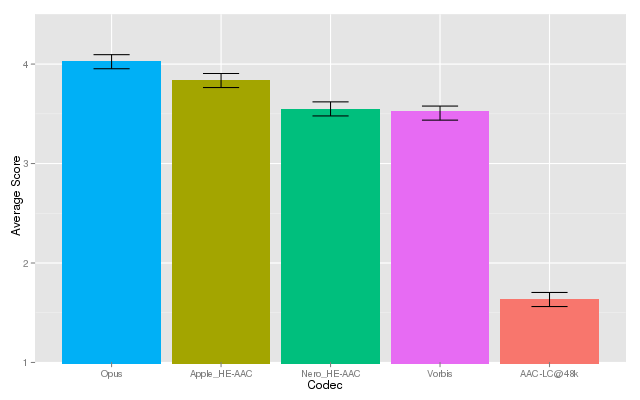


Bảng so sánh về bitrate và độ trễ (Latency):

[http://www.opus-codec.org/comparison/]



Bảng so sánh về chất lượng (kết quả bài kiểm tra của HA):



Tuy nhiên, codec này vẫn còn trong giai đoạn phát triển, chưa có phiên bản chính thức nào được công bố. Vì vậy các công cụ để mã hóa, giải mã, chuyển đổi chưa có để thử nghiệm. Thư viện của Opus đang được cập nhật các phiên bản thử nghiệm tại trang chủ. Do tài liệu vẫn chưa được cập nhật đầy đủ nên hiện tại nhóm vẫn đang nghiên cứu và sẽ đưa vào thử nghiệm trong thời gian sớm nhất có thể.

* 1. Truyền tải video/audio thông qua WebRTC:

WebRTC được đặc tả với codec duy nhất cho video là VP8, các codec hỗ trợ âm thanh là iSAC, iLBC và Opus. Tuy nhiên, 2 codec iSAC / iLBC là codec dành cho truyền tải âm thanh băng thấp, chủ yếu để truyền đạt nội dung thoại nên trong phạm vi đồ án, chúng ta sẽ nghiên cứu về codec Opus, cho phép truyền tải dữ liệu âm thanh cả chất lượng thấp và cao.

* + 1. VP8:

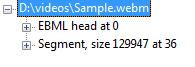
Được Google mở mã và phát triển cùng WebM, nên để đọc và truyền tải được VP8, ta sẽ tìm hiểu về cấu trúc lưu trữ của nó trong WebM.

WebM dựa trên định dạng contaner MKV (Matroska), trong đó sử dụng cấu trúc EBML để lưu trữ thông tin, gắn thẻ và tổ chức dữ liệu trong cấu trúc cây phân lớp (hierachial tree-like structure). EBML là một phiên bản cơ bản nhị phân của XML, trong khi XML là dựa trên văn bản.

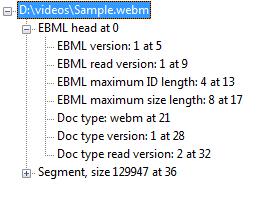
Tất cả các tập tin WebM hợp lệ phải có một tập hợp thẻ (tags) đúng chuẩn được đề nghị (ví dụ, một tập tin WebM mà không có các thông tin header là không hợp lệ). Ví dụ, tất cả các tập tin WebM phải ít nhất có hai yếu tố Cấp 0: một EBML và một Segment.

Container WebMchir chứa một track audio, một track videovà không đặc tả track lưu giữ phụ đề. Có một số phần mềm phân tích các tập tin EBML, trong phạm vi đồ án, chúng tôi khuyến khích sử dụng MKVInfo.

Đây là một ví dụ về file WebM được phân tích bằng MKVInfo:



Chúng ta có thể thấy hai yếu tố cấp 0 được thể hiện là EBML và Segment. Yếu tố EBML chứa thông tin để xác nhận nó là một tập tin EBML hợp lệ, phiên bản của nó (các trình phân tích sẽ không cố gắng đọc nếu phiên bản của nó không hỗ trợ).



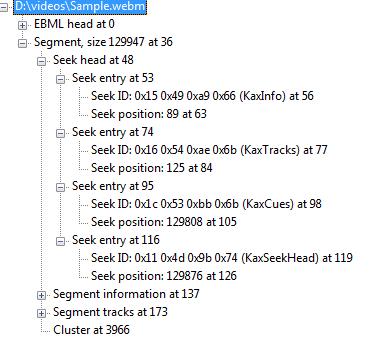
Thông tin về Segment:



Như chúng ta thấy, các phần tử của Segment có thể hơi phức tạp. Tuy nhiên, có 4 loại phần tử chính ta cần quan tâm, đó là:

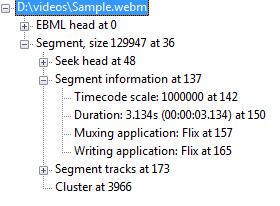
* Phần tử cấp 1: Meta Seek Information (Seek Head element)

Chứa thông tin về vị trí của các phần tử cấp một khác như KatInfo (thông tin file), KatTracks (thông tin track), KatCues (thông tin các điểm hỗ trợ seeking/scrubbing) và KatSeekHead (trỏ đến các SeekHead khác nếu có),…



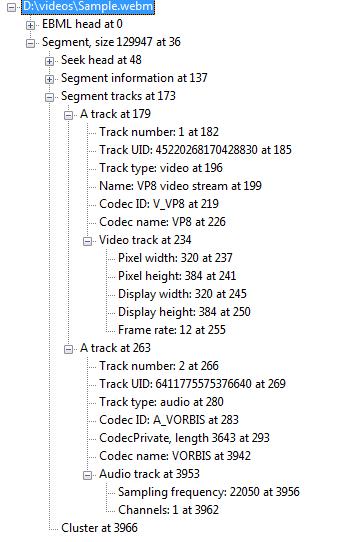
* Segment Information

Chứa thông tin cơ bản về Segment tương ứng, bao gồm độ dài, phần mềm được dùng để tích hợp (Muxing) và phần mềm đã ghi ra file,…



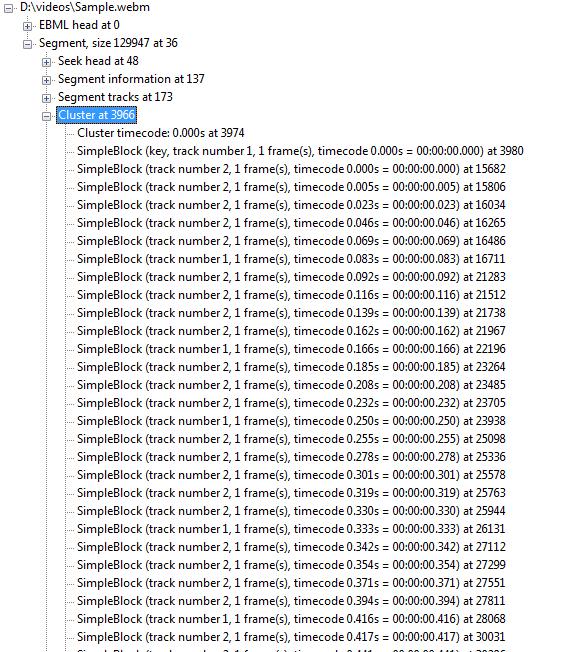
* Tracks Information

Chứa thông tin về các track của file, có khá nhiều thông tin nhưng hầu hết chúng được thể hiện khá rõ nghĩa qua MKVInfo. Lưu ý: các track trong cùng file không có cùng UID, trong WebM, video phải có Codec V\_VP8 và audio phải có codec A\_VORBIS



* Clusters

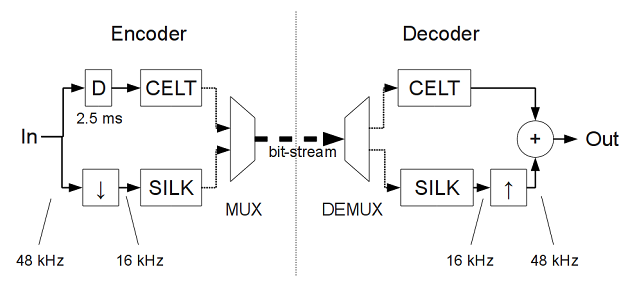
Chính là nơi lưu giữ dữ liệu của các track, được chia nhỏ thành các block với các thông tin tương ứng



Các block này chính là dữ liệu sẽ được dùng để truyền tải qua internet.

* Từ đây ta có thể nhận xét:
* Clients chính là người nhận các tín hiệu video dưới dạng block, sắp xếp lại và từ đó tiến hành giải mã, hiển thi cho người dùng.
* Server chỉ cần đảm nhận vai trò lưu trữ file, đọc các block và có giải thuật phân phối các block này về cho client.
  + 1. Opus

Opus là codec audio được mô tả như hình :



Trong đó SILK là codec phụ trách phần âm thanh thoại băng rộng (speech & voice); CELT là codec xử lý âm nhạc.

Về công nghệ :

* SILK là codec dành cho âm thanh thoại, xây dựng trên thuật toán dự đoán tuyến tính (LPC), có chất lượng xử lý tốt trên băng tần thoại rộng và hẹp (wideband và narrowband speech) nhưng không tốt đối với nhạc, cần băng tần rộng hơn nhiều (fullband). Codec này đã được thay đổi rất nhiều để tích hợp vào Opus, và không còn khả năng tương thích với codec SILK gốc.
* CELT là codec dành cho đàm thoại lẫn âm nhạc có độ trễ cực thấp, sử dụng giải thuật MDCT (modified discrete cosine transform), có hiệu quả cao với âm thanh fullband. Tuy nhiên, không được tốt với âm đàm thoại có bitrate thấp.

Về phát triển, hiện tại codec được phát triển trên các công cụ:

* Ogg Encoder / Decoder
* Matroska encoder/decoder
* Được hỗ trợ bởi Firefox

Về các chức năng và chất lượng, Opus khắc phục được nhược điểm của các codec khác:

* Điều chỉnh quyết định mã hóa tốt hơn
* Cải thiện chất lượng trong môi trường VBR (Variable BitRate)

Codec này đang trong quy trình chuẩn hóa với IETF, và trong tương lai gần sẽ là codec hỗ trợ cho RTCWeb, streaming HTML5, Skype.

[http://www.opus-codec.org]

1. Các công công nghệ network VoIP:
2. 1. SIP:

SIP, từ viết tắt của Session Initiation Protocol (Giao thức Khởi tạo Phiên) là một giao thức tín hiệu điện thoại IP dùng để thiết lập, sửa đổi và kết thúc các cuộc gọi điện thoại VOIP. SIP được phát triển bởi IETF và ban hành trong tài liệu RFC 3261.

SIP đã chiếm lĩnh thế giới VOIP nhanh như vũ bão. Giao thức này giống như giao thức HTTP, là giao thức dạng văn bản, rất công khai và linh hoạt. Do vậy, nó đã thay thế rộng rãi cho chuẩn H323.

Session Initiation Protocol là một giao thức được sử dụng trong việc truyền thông đa phương tiện thông qua mạng IP.Cung cấp một định dạng để thiết lập voice,video.Truyền theo kiểu point-to-point,confederation,và text message.

Nó là một giao thức điều khiển lớp ứng dụng để tạo mới,chỉnh sửa và kết thúc các session. Các request được tạo ra bởi các máy client và gởi tới server. Server xử lý các request và gởi một response lại cho client. Một request và response là một yêu cầu để tạo ra 1 phiên giao dịch. SIP có các thông điệp INVITE và ACK. SIP giả định cơ bản về giao thức truyền tải (transport protocol) của hệ thống mạng của client. Protocol này chính nó cung cấp độ tin cậy và nó không phụ thuộc vào độ tin cậy của TCP. SIP phụ thuộc vào Session Description Protocol (SDP) trong việc mang ra ngoài các thỏa thuận để xác minh codec. SIP hỗ trợ việc mô tả các session mà cho phép các bên tham gia có thể đồng ý thiết lập các kiểu phương tiện tương thích. SIP cung cấp các dịch vụ như:

* User Location: Xác định hệ thống cuối cùng để sử dụng cho việc truyền thông.
* Call Setup: ringing và thiết lập các thông số cho call tại hai bên called và calling.
* User Availability: Xác định sự hài lòng của bên gọi để cam kết trong truyền thông.
* User Capabilities: Xác định phương tiện và các thông số của phương tiện được sử dụng.
* Call handling: Chuyển giao và kết thúc cuộc gọi.
  1. NAT / TURN / STUN / ICE:
* NAT (Network Address Translation):

NAT hay còn gọi là Network Address Translation là một kỹ thuật được phát minh lúc khởi đầu dùng để giải quyết vấn đề thiếu IP (IPv4), nhưng dần dần nó chứng tỏ nhiều ưu điểm mà lúc phát minh ra nó người ta không nghĩ tới, một trong những lợi điểm của NAT ngày nay được ứng dụng nhiều nhất là NAT cho phép:

* Chia sẻ kết nối internet với nhiều máy bên trong LAN với một địa chỉ IP của WAN
* Một lợi điểm nữa của NAT là nó có thể làm việc như một
* Firewall, nó giúp dấu tất cả IP bên trong LAN với thế giới bên ngoài, tránh sự dòm ngó của hackers.
* Tính linh hoạt và sự dễ dàng trong việc quản lý
* NAT giúp cho các home user và các doanh nghiệp nhỏ có thể tạo kết nối với internet một cách dễ dàng và hiệu quả cũng như giúp tiết kiệm vốn đầu tư.

Và nói nôm na cho dễ hiểu thì NAT dùng để phiên dịch từ địa chỉ IP private sang địa chỉ IP Public (IP hiên nay được chia làm 2 loại: 1: IP Public, 2: IP Private).Và NAT không chỉ phiên dịch được 2 loại IP này với nhau mà nó còn cho phép NAT cả 2 lớp IP Private với nhau.

* TURN (Traversal Using Relays around NAT):

TURN là một giao thức cho phép một yếu tố đằng sau NAT hoặc tường lửa nhận dữ liệu đến trên các kết nối TCP hoặc UDP. Nó hữu ích nhất cho các đối tượng đằng sau NAT hoặc tường lửa muốn được nhận một kết nối ngang hàng (P2P).

TURN không cho phép cho người dùng chạy máy chủ trên các cổng đằng sau NAT, nó hỗ trợ các kết nối của một người sử dụng là peer duy nhất đằng sau một NAT chỉ có duy nhất một peer.

TURN cung cấp chức năng bảo mật cho người dùng cuối bên trong mạng nội bộ với ưu tiên là kết nối nhận đầu cuối hơn là kết nối gởi.

* STUN (Session Traversal Utilities for NAT):

STUN là một bộ tiêu chuẩn hóa các phương pháp, bao gồm một giao thức mạng , sử dụng trong NAT cho các ứng dụng thời gian thực bằng giọng nói, video, nhắn tin, và truyền thông tương tác IP khác.

STUN là một giao thức Client-server đơn giản chỉ có hai phương thức: truy vấn và hồi đáp. Được hiện thực phía client để người dùng sử dụng nó để giao tiếp với nhau như VoIP, cuộc gọi thoại hay tin nhắn.

* ICE (Interactive Connectivity Establishment):

ICE là một kỹ thuật được sử dụng trong mạng máy tính liên quan đến phiên dịch địa chỉ mạng (NAT) trong các ứng dụng Internet của VoIP, truyền thông peer-to -peer, video, tin nhắn tức thời và các thông tin tương tác khác. Nó giải quyết các vấn đề về NAT, firewall để cho phép xây dựng kết nối VoIP.

PHẦN III:   
ỨNG DỤNG WEBRTC TRONG   
GIẢNG DẠY TRỰC TUYẾN

1. Đào tạo từ xa
   1. Giới thiệu

Đào tạo từ xa (Distance education hay distance learning) là một lĩnh vực giáo dục tập trung vào phương pháp giảng dạy và công nghệ nhằm đưa nội dung giảng dạy đến các sinh viên không thể đến lớp (vì thời gian, khoảng cách địa lý hay nhiều lý do khác) như mô hình đào tạo truyền thống.

* + 1. Lịch sử đào tạo từ xa.

Đào tạo từ xa ra đời sớm nhất vào năm 1728 khi có một quảng cáo đăng tìm kiếm sinh viên cho một khóa học và các bài học sẽ được gửi đến hằng tuần.

Đào tạo từ xa phụ thuộc vào việc xuất hiện dựa vào việc xuất hiện các công nghệ mới. Đào tạo từ xa đầu tiên dựa trên sự phát triển của dịch vụ thư tín, khởi đầu tại Anh vào những năm 1840. Thế hệ đào tạo từ xa đầu tiên dựa vào công nghệ in ấn. Thế hệ đào tạo từ xa thứ hai dựa trên sự kết hợp nhiều loại hình công nghệ bao gồm in ấn, đài phát thanh và truyền hình. Ở giai đoạn này, sự liên hệ giữa sinh viên và cơ sở đào tạo rất hạn chế. Sinh viên nhận được rất ít sự hỗ trợ.

Thế hệ đào tạo từ xa thứ ba gắn liền với sự phát triển của máy tính và internet. Điều này khiến cho việc phân phối tài liệu nhanh hơn và rẻ hơn. Sinh viên có thể dễ dàng ghi danh và tham gia khóa học phù hợp. Năm 1996 ở Mỹ xuất hiện trường ĐH đầu tiên hoàn toàn đào tạo trực tuyến.

Hiện nay, có rất nhiều tổ chức trên toàn thế giới cung cấp hình thức đào tạo trực tuyến trên toàn thế giới. Ở Việt Nam, các trường ĐH đã bắt đầu mở các khóa học đào tạo trực tuyến bên cạnh chương trình đào tạo chính quy. Năm 2009, chúng ta có 17 trường ĐH có chương trình đào tạo từ xa. Học viên theo học ở nhiều ngành như sư phạm, kinh tế, tài chính, tin học...

* + 1. Xu hướng công nghệ

Ngày nay, với sự phát triển của một thế hệ công nghệ mới nhanh hơn, mạnh hơn, phổ cập hơn và luôn sẵn sàng hầu như mọi lúc mọi nơi với người dùng

- Mạng băng thông rộng thế hệ thứ 4 được thiết kế phù hợp với việc truyền tải video chất lượng cao và cung cấp khả năng kết nối không dây liên tục giữa các thiết bị.

- Sự phổ cập của smart phone và thiết bị di động cá nhân.

- Xu hướng điện toán đám mây: các dữ liệu và ứng dụng được lưu trữ trên máy chủ có thể được truy cập mọi lúc mọi nơi.

- Công nghệ tăng cường sự tương tác với người dùng như Google Glass, Microsoft Kinect.

- Công nghệ lập trình ứng dụng Web và ứng dụng di động biến tài liệu học từ những cuốn sách tĩnh thành những nội dung sống động và tương tác.

- Và một công nghệ đã được trình bày trong tài liệu này là công nghệ đàm thoại trực tuyến trên trình duyệt, sẵn sàng với người dùng và có khả năng cung cấp nội dung real-time media chất lượng cao.

Tập hợp các công nghệ thế hệ mới tăng cường sức mạnh và hứa hẹn sẽ tạo nên một bước tiến mới của hình thức đào tạo từ xa. Các chương trình đào tạo từ xa có thể trở nên phong phú hơn, đa dạng hình thức tổ chức; tăng cường sự tương tác giữa giảng viên – học viên và học viên – học viên; cơ hội cho học viên truy cập vào kho tài liệu trực tuyến phong phú; tài liệu học tập được thiết kế sinh động và thể hiện dưới nhiều hình thức mới: website tương tác, ebook tương tác, video tương tác, phòng thí nghiệm ảo, ứng dụng di động…

* 1. Đặc điểm của đào tạo từ xa
     1. Thế mạnh của đào tạo từ xa

Không giới hạn số lượng sinh viên tham gia. Các khóa học có thể lên đến hàng trăm nghìn hay hàng triệu sinh viên ghi danh.

Nội dung học có thể được truy cập mọi lúc mọi nơi. Học viên có thể chủ động sắp xếp thời gian và kế hoạch học tập phù hợp với bản thân. Thích hợp với các học viên gặp khó khăn về khoảng cách địa lý, những người đi làm hoặc vì nhiều lý do khác không thể tham gia một khóa học chính quy ở ĐH.

Đào tạo từ xa trở thành một hình thức đào tạo thay thế, mở ra cánh cửa tiếp cận chương trình đào tạo chất lượng trên thế giới dành cho sinh viên các nước, những người không có điều kiện để du học hoặc ghi danh vào học chính quy.

Hình thức tương tác và chấm điểm học viên thực hiện trên mạng, dễ dàng theo dõi và đánh giá chi tiết đến từng cá nhân học viên: số lượng bài tập hoàn thành, số lượng câu hỏi tham gia thảo luận trên diễn đàn, số lượng tài liệu học đã theo dõi. Sinh viên cũng dễ dàng truy cập vào thư viện số và kho tài liệu số đa dạng với chi phí thấp.

Các tài liệu trong các một số khóa đào tạo từ xa, ví dụ như trường ĐH MIT hoặc UKOU, sẽ được đưa lên mạng miễn phí. Sinh viên thế giới và trường ĐH khác có thể tiếp cận chương trình đào tạo chất lượng cao. Các trường ĐH khác có thể thiết kế khóa học giữa trên các tài liệu này, giảm thiểu công sức soạn thảo tài liệu, dành nhiều thời gian cho giảng dạy và nghiên cứu.

* + 1. Đảm bảo chất lượng đào tạo từ xa

Cần biên soạn giáo trình riêng và thiết kế chương trình phù hợp với hình thức đào tạo từ xa. Khóa học cần được nhóm chuyên gia bao gồm nhiều lĩnh vực tham gia thiết kế: giảng viên, chuyên gia CNTT, biên tập viên, người xử lý video.

Sinh viên cần được sự hỗ trợ của các trợ giảng trong quá trình học tập và thực hành. Việc trao đổi giữa sinh viên và giảng viên, giữa sinh viên và sinh viên cần được thiết kế hiệu quả: trao đổi qua diễn đàn, hệ thống email, trao đổi qua các buổi giảng trực tuyến hay các buổi thảo luận trực tuyến.

Đào tạo từ xa phụ thuộc vào hạ tầng công nghệ. Cần xây dựng cơ sở hạ tầng ổn định, công nghệ và hỗ trợ kỹ thuật sẵn sàng 24/7.

Một số chương trình đào tạo từ xa như chương trình cao học, sinh viên sẽ hạn chế trao đổi với nhau nếu khác biệt trình độ quá lớn, nên tổ chức thi tuyển đầu vào để phân loại học viên.

Bài tập, bài thực hành và bài thi cần được thiết kế phù hợp với đặc thù của chương trình đào tạo từ xa. Cần đưa ra phương pháp đánh giá chất lượng tham gia và hoàn thành nội dung học của học viên. Cần có phương pháp xác thực sinh viên khi thi.

* 1. Ví dụ mô hình đào tạo từ xa
     1. Đào tạo từ xa ở trường ĐH Bách Khoa TP. HCM

Hình thức đào tạo ĐH qua mạng ở trường ĐH Bách Khoa TP. HCM bắt đầu tuyển sinh vào năm 2011 chuyên ngành Công nghệ thông tin với thời gian đào tạo 05 năm, áp dụng cho các đối tượng đã tốt nghiệp Trung học (cấp 3).

* Cách thức tổ chức giảng dạy

- Dạy trực tiếp (phòng học truyền thống): thời gian giảng viên gặp sinh viên và giảng dạy tại lớp chiếm 15% - 25% tổng quỹ thời gian môn học.

- Dạy trực tiếp - trực tuyến (online): theo phương thức từ xa sử dụng “video conference” , sinh viên có thể theo dõi lịch học để tham gia tại các phòng học ở các cơ sở liên kết đào tạo hoặc tại trường. Sinh viên sẽ có thể trao đổi ý kiến với giảng viên đang ở xa; Sinh viên cũng có thể học ở nhà nếu có các thiết bị đa phương tiện đã đăng ký. Thời lượng giảng theo hình thức này chiếm 75% -85%.

- Dạy không trực tuyến (offline): toàn bộ khối lượng bài giảng trực tuyến sẽ được đưa lên mạng cho sinh viên tham khảo trong thời gian nhất định do GV thông báo.

- Dạy bằng bài giảng điện tử, băng hình và âm: sinh viên sẽ được cung cấp học liệu điện tử trong quá trình học.

- Sinh viên thực tập trong các phòng máy tại các cơ sở đào tạo và được giáo viên hướng dẫn trực tiếp và/hoặc thực tập trên mạng trong phòng thí nghiệm ảo (trong một số môn - khi có điều kiện).

- Trợ giảng sẽ chữa bài tập trực tuyến. Trong giờ bài tập trực tuyến sinh viên và giảng viên có thể trao đổi các kiến thức chuyên môn.

* Về chương trình

Chương trình đào tạo từ xa của trường chỉ mới triển khai từ năm 2011 và chỉ đào tạo một ngành là Công nghệ thông tin. Chương trình phù hợp với các sinh viên không thể sắp xếp thời gian đến lớp, có thể chủ động thời gian tham gia hoặc sinh viên vì điều kiện địa lý không thể đến trường. Trong tương lai chương trình có thể tiếp tục được mở rộng về số lượng ngành có đào tạo từ xa và đối tượng sinh viên tham gia cũng như cải tiến hình thức tổ chức.

* + 1. Udacity (www.udacity.com)
* Udacity trường đại học trực tuyến miễn phí được thành lập bởi Sebastian Thrun (ĐH Stanford) và Peter Norvig (Giám Đốc Nghiên Cứu Google). Bắt đầu từ một lớp học Introduction to AI diễn ra vào T10-T12 năm 2011 (www.ai-class.com). Sau một vài tuần, khóa học thu hút trên 160.000 sinh viên từ trên 190 nước đăng ký. Từ thành công đó, các giáo sư quyết định thành lập đại học Udacity với tiêu chí trở thành trường đại học chất lượng và chi phí thấp, với những giáo viên tốt nhất và mở cửa cho mọi người trên toàn thế giới.
* Mô hình và phương pháp giảng dạy
* Các khóa học dành cho tất cả mọi người trên toàn thế giới và hoàn toàn miễn phí. Học viên có thể dễ dàng ghi danh trên website, không có giới hạn đầu vào. Quá trình học và thi hoàn toàn diễn ra trực tuyến trên website.
* Mỗi khóa học kéo dài trong 5 tuần. Số lượng sinh viên mỗi khóa học lên đến hàng trăm nghìn người.
* Nội dung học được cung cấp mỗi tuần bao gồm: các bài giảng video ngắn, các câu đố (quiz) trắc nghiệm kiến thức, bài tập luyện tập và tính điểm, bài tập lớn (assignment) và một bài thi cuối khóa. Cấu trúc điểm 50% (bài tập) + 50% (thi cuối khóa). Mỗi tuần cũng bao gồm một buổi trả lời các câu hỏi hay từ học viên.
* Sinh viên thảo luận trên diễn đàn và theo dõi tiến độ học tập trong profile cá nhân.

1. Đào tạo từ xa từ góc nhìn kỹ thuật

Đào tạo từ xa phụ thuộc vào công nghệ và hạ tầng kỹ thuật để thực hiện chuyển tải nội dung giảng dạy đến học viên. Cùng với sự phát triển của công nghệ, đào tạo từ xa có cơ hội áp dụng các công nghệ tiên tiến vào nhằm nâng cao chất lượng đào tạo và đáp ứng tốt nhất nhu cầu của học viên.

Một số tình huống kỹ thuật có thể trong đào tạo từ xa

* 1. Xác thực học viên

Tình huống

* Học viên ghi danh đăng ký tham gia chương trình đào tạo từ xa. Mỗi người được cấp một hồ sơ để lưu giữ thông tin trong suốt quá trình đào tạo.
* Cần có phương pháp xác thực học viên. Chỉ có học viên đã ghi danh mới có thể truy cập nội dung học, tham gia vào bài giảng trực tuyến hay làm bài tập.
* Hạn chế và phát hiện việc học viên gian lận trong khóa học như sao chép đáp án. Khi tổ chức kiểm tra thực tuyến, phải xác thực rằng người đi thi đúng là người đã đăng ký.
  1. Trình chiếu video và slide bài giảng

Tình huống

* Học viên đăng nhập vào khóa học trên website và truy cập vào nội dung bài giảng. Nội dung bài giảng bằng video được phát song song với slide bài giảng. Khi video đang được phát, slide sẽ thay đổi tương ứng với nội dung đang được nhắc đến. Xen kẽ trong quá trình phát video, xuất hiện các câu đố trắc nghiệm, bài tập thực hành ngay tại chỗ dành cho học viên.
* Học viên không thể download và lưu trữ video vào máy tính của họ. Học viên bắt buộc phải xem video trực tiếp trên website. Các video đã xem được đánh dấu để học viên có thể theo dõi tiến độ học tập.
* Học viên có thể truy cập bằng các phương tiện di động như smart phone, máy tính bảng vào bất kỳ thời điểm nào thích hợp.
  1. Giảng dạy và thảo luận trực tuyến

Tình huống

* Bên cạnh các bài giảng được soạn trước và đưa lên website dưới dạng video, chương trình đào tạo từ xa còn tổ chức các buổi giảng dạy trực tuyến với sự trình bày của các giảng viên. Học viên đăng nhập vào website và tham gia vào buổi giảng dạy bằng thiết bị của họ. Trên màn hình học viên có màn hình video của giảng viên và slide bài giảng. Khi giảng viên trình bày, slide thay đổi phù hợp với nội dung được trình bày. Học viên có thể tua về trước để xem lại phần đã giảng cũng như ngay lập tức quay trở lại thời gian thực. Bài giảng được ghi lại để học viên có thể xem lại sau này.
* Các buổi thảo luận trực tuyến được diễn ra song song. Học viên đăng nhập vào website, chuẩn bị camera, microphone và tham gia vào buổi thảo luận. Có nhiều nhóm thảo luận và học viên có thể lựa chọn nhóm để tham gia vào. Học viên có thể đổi nhóm thường xuyên. Trên màn hình của học viên có hình ảnh của người đang trình bày, slide trình bày và một giao diện tương tác giữa các học viên (như một khung chat, một tấm bảng vẽ) Khi muốn phát biểu ý kiến, học viên chọn một nút trên màn hình (tương tự việc giơ tay). Lần lượt từng học viên sẽ được chuyển thành người nói chính theo thứ tự giơ tay. Màn hình video chính sẽ chuyển thành hình ảnh của học viên.
  1. Làm bài tập, thi và theo dõi quá trình học tập

Tình huống

* Sau mỗi bài học, học viên được yêu cầu làm các bài tập và phải gửi bài làm trước bài học tới. Bài tập được học viên làm trực tiếp trên website hoặc gửi bài làm qua website. Ví dụ khi học lập trình Java, học viên có thể viết mã lập trình Java trên giao diện website rồi chọn nút Run để thực thi và đọc kết quả. Việc này giúp học viên có thể ngay lập tức làm bài tập bằng thiết bị và vào thời điểm thích hợp. Học viên không cần phải download, cài đặt và cấu hình Java, họ cũng không phải luôn sử dụng máy tính để thực hành bài tập mà có thể làm ngay trên thiết bị di động.
* Có diễn đàn để học viên thảo luận về nội dung bài học, bài tập và giao lưu.
* Mỗi học viên có một hồ sơ lưu trữ quá trình học tập tại trường. Hồ sơ theo dõi và lập biểu đồ đánh giá học viên. Qua quá trình làm bài tập, hồ sơ có thể xác định học viên đang hổng kiến thức phần nào và đưa ra lời khuyên cần học kỹ hơn.
* Tùy vào đặc thù của loại hình đào tạo từ xa, kỳ thi có thể được tổ chức ngoại tuyến hay trực tuyến, tập trung hay phân tán. Kỳ thi ngoại tuyến tập trung yêu cầu học viên phải có mặt tại địa điểm vào thời điểm quy định và làm bài thi. Kỳ thi trực tuyến tập trung yêu cầu học viên phải đăng nhập vào website tại thời điểm diễn ra thi và làm bài thi. Thi trực tuyến phân tán cho phép học viên đăng nhập và làm bài thi vào bất kỳ lúc nào trong thời gian diễn ra kỳ thi. Thi tập trung cần đảm bảo hạ tầng kỹ thuật hoạt động ổn định trong suốt thời gian thi. Thi trực tuyến cần có phương pháp xác thực người dự thi. Thi phân tán dẫn đến nguy cơ lộ đề thi nên chỉ thích hợp cho một số loại hình đào tạo từ xa chuyên biệt.
  1. Phòng thí nghiệm ảo

Tình huống

* Trong quá trình đào tạo, học viên cần làm các bài thực hành trên các phần mềm chuyên dụng. Học viên có thể đăng nhập vào máy chủ và điều khiển từ xa phần mềm bằng thiết bị của họ. Màn hình thiết bị hiển thị hình ảnh diễn ra trên phần mềm dưới dạng real-time media và cung cấp các cơ chế điều khiển từ xa. Mỗi học viên được giới hạn thời gian truy cập vào phòng thí nghiệm ảo.
* Học viên truy xuất một mô hình 3D trong chương trình đào tạo từ xa bằng thiết bị di động. Hình ảnh mô hình được xuất ra, gửi về và hiển thị trên màn hình di động dưới dạng real-time media. Học viên có thể điều khiển và tương tác với mô hình bằng giao diện điều khiển được cung cấp trên thiết bị.
  1. Chịu được lượng lớn người dùng truy cập đồng thời

Tình huống

* Chương trình đào tạo từ xa thu hút hàng trăm nghìn học viên ghi danh học. Khi một bài học mới được tải lên dưới dạng video, hệ thống chịu hàng nghìn yêu cầu xem video đồng thời. Cần có cơ chế phân tán và cache để đảm bảo hệ thống hoạt động ổn định, không xảy ra hiện tượng trễ, lag khi xem video.

1. Sử dụng WebRTC/HTML5 trong hệ thống đào tạo từ xa

Một số tình huống kỹ thuật có thể được cải tiến bằng việc sử dụng WebRTC/HTML5. Phần này sẽ trình bày lần lượt từng tình huống kỹ thuật đã nêu ở trên, các giải pháp hiện tại đang sử dụng và khả năng cải tiến bằng WebRTC. Không phải tất cả các tính năng đều cần đến WebRTC/HTML5, chỉ một số tính năng được tình bày.

* 1. Trình chiếu video và slide bài giảng

Giải pháp hiện tại

* Giải pháp 1: Sử dụng một kênh chia sẻ video như Youtube để trình chiếu video.

Giải pháp này nhanh chóng và đơn giản, không tiêu tốn tài nguyên hệ thống. Nhược điểm là không đồng bộ slide bài giảng, nội dung bài giảng được công khai trên mạng.

* Giải pháp 2: Sử dụng một trình chơi flash để trình chiếu video. Slide bài giảng cũng được trình bày bằng flash.

Giải pháp này đồng bộ được slide bài giảng. Nhược điểm là học viên phải cài đặt plugin flash để sử dụng, do đó không thể xem trên các thiết bị di động.

* Giải pháp 3: Sử dụng thẻ <video> phát trực tiếp file video trên trình duyệt, sử dụng JavaScript để tương tác với trang web, slide bài giảng.

Giải pháp này đồng bộ được slide bài giảng, không yêu cầu học viên phải cài đặt plugin để sử dụng, có thể áp dụng trên các thiết bị di động. Nhược điểm là video có thể được tải về và chia sẻ trên mạng.

Giải pháp WebRTC/HTML5

* Truyền file video bài giảng về dưới dạng real-time stream. Trong quá trình chiếu video, ứng dụng web biết được người dùng đang xem đến đoạn nào và đồng bộ với nội dung hiển thị của slide. Server kiểm soát nội dung được truyền về người dùng, có thể chủ động thay đổi nội dung phù hợp, tạo nên các video tương tác.

Đánh giá

* Mở rộng khả năng của video, không chỉ trình chiếu file video đơn thuần. Có thể tích hợp thêm các tính năng khác vào video: người xem phải đăng nhập mới có thể xem nội dung video; người xem chủ động tương tác với video.
* Người xem không thể download video về máy, chỉ có thể xem trực tiếp trên website. Người xem không cần cài đặt thêm plugin hay ứng dụng khác, có thể xem trên các thiết bị di động.
* Nhược điểm: Truyền dưới dạng live-stream có thể gây ra trễ, lag nếu sử dụng mạng internet không ổn định.
  1. Giảng dạy và thảo luận trực tuyến

Giải pháp hiện tại

* Giải pháp 1: Sử dụng một ứng dụng như Team Viewer hoặc Skype để đàm thoại trực tuyến.
* Giải pháp 2: Xây dựng một giải pháp đàm thoại trực tuyến sử dụng ứng dụng chuyên biệt.

Cả hai giải pháp yêu cầu học viên phải sử dụng ứng dụng được cung cấp, do đó hạn chế khả năng tiếp cận của học viên (cần phải cài đặt, chỉ có thể chạy trên hệ điều hành và thiết bị hỗ trợ ứng dụng). Các ứng dụng có sẵn không đáp ứng đúng nhu cầu và đặc thù đào tạo từ xa của đơn vị. Các ứng dụng chuyên dụng cần công sức và chi phí phát triển cao.

Giải pháp WebRTC/HTML5

* Xây dựng một hệ thống hội thoại trực tuyến trên trình duyệt đáp ứng nhu cầu và đặc thù đào tạo từ xa bằng WebRTC/HTML5.

Đánh giá

* WebRTC/HTML5 được thiết kế nhằm phục vụ hội thoại trực tuyến hoạt động trên trình duyệt. Công nghệ nhằm giảm công sức của nhà phát triển, tăng tính ổn định. Kết hợp với các công nghệ khác trong chuẩn HTML5, có thể dễ dàng phát triển và tùy biến, cung cấp nhiều tính năng phong phú đa dạng khác, có thể tích hợp sâu vào website. Hoạt động trên nền tảng JavaScript phổ biến với hàng triệu nhà phát triển trên thế giới, chi phí và công sức phát triển thấp hơn so với các công nghệ hiện tại như SIP.
* Học viên có thể ngay lập tức sử dụng dich vụ trên thiết bị bằng cách truy cập vào website mà không cần download và sử dụng các phần mềm chuyên biệt.
* Có thể phát triển các tích năng phong phú khác như slide trình chiếu, bảng vẽ tương tác... bằng JavaScript/HTML5
  1. Phòng thí nghiệm ảo

Giải pháp WebRTC/HTML5

* Hình ảnh từ ứng dụng trên máy chủ được truyền về thiết bị của học viên dưới dạng live-stream bằng công nghệ WebRTC. Học viên dùng thiết bị của mình điều khiển hoạt động ứng dụng.

Đánh giá

* Việc thí nghiệm có thể được học viên thực hiện trên thiết bị di động bất kỳ lúc nào. Giảm thiểu công sức học viên phải cài đặt, cấu hình ứng dụng phức tạp.
* Mở ra một khả năng ứng dụng phong phú: Ứng dụng trên máy chủ có thể trực tiếp tích hợp công nghệ WebRTC vào và kiểm soát quá trình truyền video và thao tác điều khiển. Giao diện trở nên thân thiện hơn với trình duyệt và thiết bị di động.

PHẦN IV:   
KẾT QUẢ

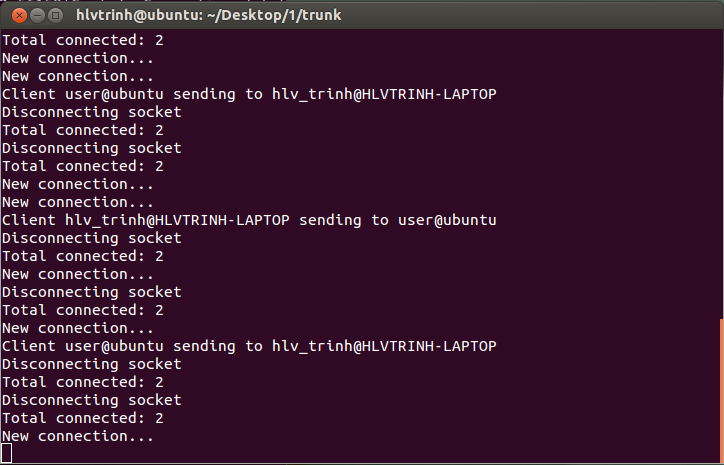
1. Những phần việc đã làm được và chưa làm được:

* Những việc đã làm được:
* Tìm hiểu được công nghệ chính: WebRTC – nguyên lý, mô hình hoạt động.
* Tìm hiểu về các codec mã hóa và giải mã video / audio cần thiết.
* Tìm hiểu được mô hình kết nối, quản lý kết nối của công nghệ WebRTC.
* Tìm hiểu được mô hình giảng dạy trực tuyến của các trường / tổ chức trong và ngoài nước.
* Hiện thực được các demo của WebRTC trên HTML5, trên C++ ở môi trường Linux và Windows
* Những việc chưa làm được:

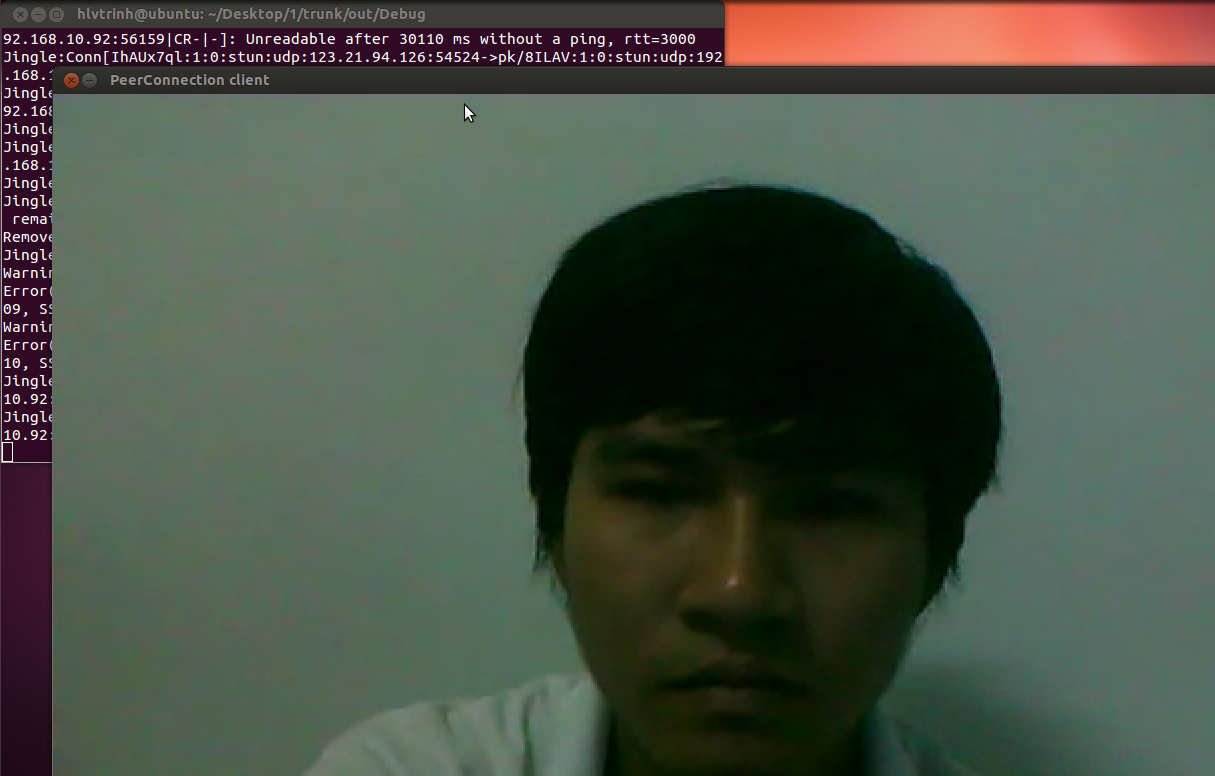
1. Các khó khăn, thuận lợi:

* Khó khăn:
* Toàn bộ các công nghệ đang tìm hiểu đều rất mới, hầu hết đều chỉ mới được phát triển từ năm 2010 đến nay, thậm chí có công nghệ vẫn chưa được công bố phiên bản chính thức (Codec Opus). Vì vậy đặc tả, tài liệu và demo của các công nghệ này không nhiều. Gây khó khăn không nhỏ cho nhóm khi tiến hành nghiên cứu.
* Hiện tại phần cứng lẫn phần mềm đều chưa hỗ trợ đầy đủ cho công nghệ mới, vì vậy nhóm gặp không ít khó khăn khi thử nghiệm, hiện thực các thành phần của công nghệ.
* Trong thời gian nhóm thực hiện đồ án, thầy Bùi Hoài Thắng – GVHD phải đi nước ngoài, không thể theo dõi và chỉ bảo thật sự sát sao.
* Thuận lợi:
* Nhóm nhận được sự quan tâm, động viên, định hướng rõ ràng từ thầy Bùi Hoài Thắng.
* Nhóm được sự giúp đỡ chu đáo, nhiệt tình của anh Đặng Xuân Hiếu và Nguyễn Hoàng Nguyên trong quá trình tìm hiểu, khám phá công nghệ.

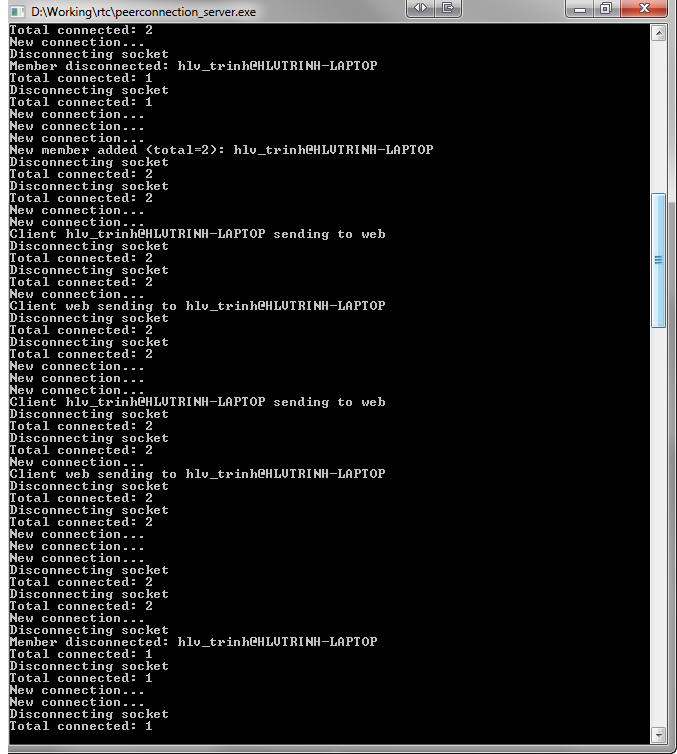
1. Demo:
   1. Demo native WebRTC trên Linux (Ubuntu):
      1. Server:



* + 1. Client:



* 1. Demo native WebRTC trên Windows:
     1. Server:

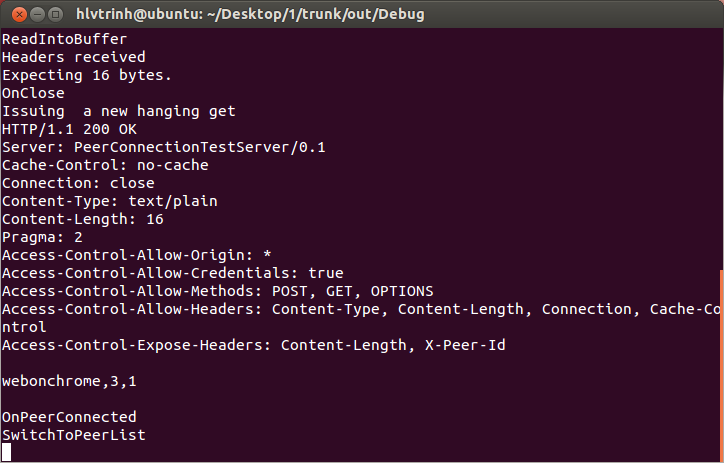


* + 1. Client:

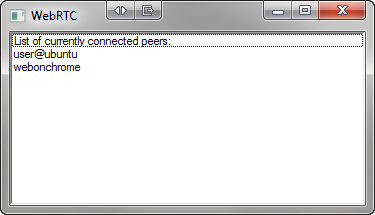


* 1. Demo giao tiếp WebRTC giữa Brower, Linux và Windows:

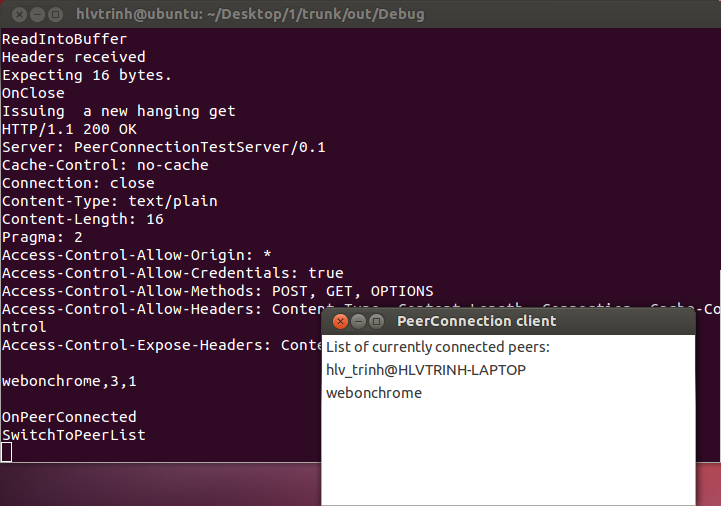
Với server chạy trên nền tảng bất kỳ, các client có thể giao tiếp và nhìn thấy nhau dễ dàng. Ở đây, chúng tôi chạy server trên môi trường Linux



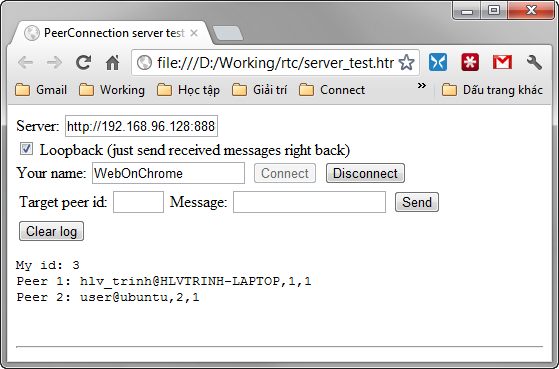
* + 1. Client trên Windows:



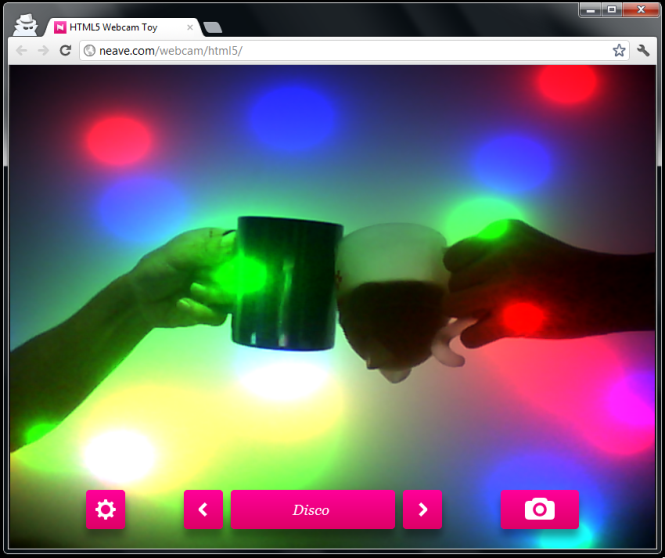
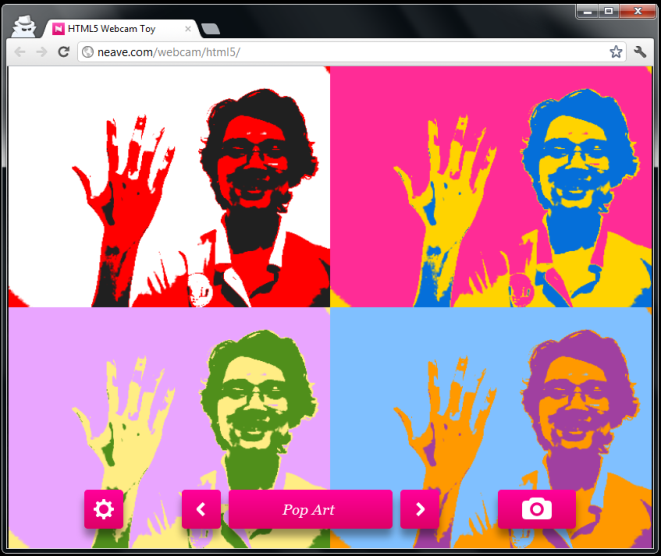
* + 1. Client trên Ubuntu:



* + 1. Web loopback client:



* 1. Demo nhận video từ camera và xử lý bằng HTML5/JavaScript
* Công nghệ: HTML5
* API: Media Capture API, Canvas 2D và thẻ Video.
* Demo nhận stream video từ camera, xử lý thêm hiệu ứng và xuất ra màn hình.

PHẦN V:  
PHỤ LỤC

TÀI LIỆU THAM KHẢO

* <http://wikipedia.org>
* <http://webrtc.org>
* <http://www.webmproject.org/>
* <http://toncar.cz/>
* <http://www.xiph.org/>
* <http://opus-codec.org/>