# WEBRTC – CÔNG NGHỆ WEB TRUYỀN THÔNG THỜI GIAN THỰC

*** ThS. Hoàng Thị Kim Chi**

Ngày nay, người sử dụng đã quen thuộc với giao diện Web, sử dụng các ứng dụng Web trong công việc và đời sống. Những tính năng được người dùng ưa thích như Blog (nhật ký web), các mạng xã hội (Facebook, Twitter và MySpace) hay Wiki (trang web cập nhật nội dung trực tuyến) của công nghệ Web 2.0 đã tạo "các cộng đồng trực tuyến" nhờ khả năng kết nối năng động và dễ dàng chia sẻ thông tin trên Web. Công nghệ WebRTC (Web Real-Time Communications) được phát triển hiện nay cho phép liên lạc thời gian thực bằng video, âm thanh cũng như các loại dữ liệu khác. WebRTC cho phép gọi điện video ngay trong trình duyệt mà không cần đăng ký tài khoản, cũng không cần cài thêm plugin nào. Ngoài ra, nhiều loại ứng dụng cũng được phát triển đựa trên WebRTC.

## 1. GIỚI THIỆU CHUNG VỀ CÔNG NGHỆ WEBRTC

Ý tưởng phát triển WebRTC được nhóm kỹ sư chịu trách nhiệm cho GoogleHangouts đưa ra từ năm 2009. Vào thời gian đó, để truyền tải video, hình ảnh trên web thì người ta thường phải dùng Flash. Nhóm kỹ sư Hangouts lại không muốn sử dụng công nghệ này, nên họ bắt đầu tự làm một chuẩn riêng cho mình.

Đến năm 2010, Google mua lại hai công ty On2 và Global IP Solutions (GIPS) để tiếp cận công nghệ truyền dữ liệu thời gian thực làm nền tảng cho WebRTC như các bộ codec và các kỹ thuật loại bỏ tiếng vang.

Vào tháng 5/2011, Google ra mắt một dự án nguồn mở dành cho việc giao tiếp thời gian thực giữa trình duyệt với nhau, và từ lúc này dự án mang tên WebRTC. Song song đó, Hiệp hội W3C (World Wide Web Consortium) và Tổ chức IETF (Internet Engineering Task Force) cũng đang phát triển một số giao thức để dùng cho việc việc kết nối thời gian thực, vì vậy họ bắt tay nhau tiếp tục hoàn thiện và sau đó kết hợp chung vào WebRTC.

Ngày 27/10/2011, Hiệp hội W3C ra mắt bản "nháp" đầu tiên của WebRTC.

Tháng 11/2011, Chrome 23 ra mắt, trở thành trình duyệt đầu tiên có tích hợp WebRTC ngay từ bên trong. Công nghệ WebRTC vẫn đang tiếp tục được phát triển và liên tục cập nhật.

* Học viện Hành chính Quốc gia

Đến nay, vào ngày 22/3/2016, Google đã phát hành phiên bản Chrome 50 (trong bản Beta Channel). Theo đó, trình duyệt Chrome 50 có sự hỗ trợ chuẩn video H.264 (codec video) của WebRTC.

Tiếp đó, ngày 29/8/2016, Mozilla đã phát hành phiên bản trình duyệt Firefox 49. Firefox 49 đã được tăng cường hỗ trợ tiêu chuẩn WebRTC trong kiểm soát băng thông. Các cải thiện quản lý băng thông bao gồm: cho phép cho phương tiện truyền thông một chiều và xử lý băng thông tốt hơn. Đồng thời, trình duyệt Firefox 49 cho phép kiểm tra NAT traversal trong các topo NAT (Network Address Translation) khác nhau.

Hiện nay, WebRTC vẫn còn đang tiếp tục được phát triển chứ chưa hoàn thiện. Tuy nhiên, các nhà nghiên cứu và phát triển ứng dụng bắt đầu giới thiệu về công nghệ WebRTC dành cho các thiết bị di động chạy hệ điều hành IOS và Android như:

César Guirao (tại Tokbox) với "Android Development with WebRTC", (RealTime Weekly ngày 22/11/2016). César Guirao đã trình bày ba phương pháp chính có thể xây dựng một ứng dụng Android với WebRTC.

Và Arik Halperin (tại Greenfield Tech) với "IOS Development with WebRTC", (RealTime Weekly ngày 22/11/2016). Arik Halperin sử dụng AppRTC để xây dựng ứng dụng WebRTC. Ứng dụng được dùng pristine.io của AppRTC, mã nguồn mở của Google trên GitHub.

## 2. WEBRTC LÀ GÌ?

Theo **Arin Sime** - CEO của công ty AgilityFeat.com có trụ sở tại Charlottesville, bang Virginia, Hoa Kỳ, người đã đưa ra định nghĩa rất ngắn gọn về WebRTC như sau:

**"WebRTC là một tiêu chuẩn HTML5 cho mã hóa peer-to-peer (P2P) trao đổi video, âm thanh và dữ liệu, trực tiếp giữa hai máy tính"**.

Các tính năng nổi bật nhất của WebRTC đó là được tích hợp trực tiếp vào trình duyệt, và chúng ta dùng HTML5 và Javascript để sử dụng nó.

WebRTC không chỉ là một sản phẩm hay một hàm API duy nhất. Nó là một tập hợp rất nhiều các hàm, có thể được lập trình viên sử dụng cho nhiều mục đích khác nhau. Có hàm chỉ để làm những việc đơn giản như đòi quyền truy cập vào webcam và microphone của máy tính, có hàm phức tạp hơn thì để thiết lập kết nối giữa hai người dùng với nhau, có hàm còn dùng để chia sẻ màn hình với người dùng khác. Và có hàm để kết nối (gọi) video, đây là chức năng quan trọng nhất của WebRTC.

Tuy nhiên, tất cả các hàm lập trình nằm trong bộ API có đặc điểm chung là thực thi hầu hết các tác vụ theo thời gian thực. Và nó không chỉ được dùng cho việc gọi video giữa hai trình duyệt mà còn có thể thực hiện nhiều tác vụ

khác, để hai hoặc nhiều người dùng liên lạc với nhau. WebRTC được hỗ trợ chính thức bởi Google, Mozilla, Opera cùng nhiều hãng khác.

### 3. ĐẶC ĐIỂM CỦA WEBRTC

Công nghệ WebRTC có một số đặc điểm sau đây:

#### *a. Các hàm API của WebRTC*

WebRTC có ba loại hàm API thường được dùng:

**getUserMedia**: truy cập vào camera và microphone của người dùng

**RTCpeerConnection**: gửi và nhận dữ liệu hình ảnh, giọng nói

**RTCdataChannel**: gửi và nhận dữ liệu không phải là hình ảnh, giọng nói giữa ứng dụng/trình duyệt

Các trình duyệt web hỗ trợ 3 hàm AIP WebRTC trên đây (getUserMedia, RTCpeerConnection và RTCdataChannel) được liệt kê ở bảng dưới đây:

| TÊN HÀM | TRÌNH DUYỆT WEB HỖ TRỢ HÀM API |
|---|---|
| **getUserMedia** | • Chrome desktop 18.0.1008+; Chrome for Android 29+<br>• Opera 18+; Opera for Android 20+<br>• Opera 12, Opera Mobile 12 (dựa trên: Presto engine)<br>• Firefox 17+<br>• Microsoft Edge |
| **RTCPeerConnection** | • Chrome desktop 20+; Chrome for Android 29+<br>• Opera 18+;<br>Opera for Android 20+<br>• Firefox 22+ |
| **RTCDataChannel** | • Chrome 25 (phiên bản thử nghiệm); Chrome 26+ (phiên bản ổn định và có khả năng tương tác với Firefox); Chrome for Android 29+<br>• Opera 18+; Opera for Android 20+<br>• Firefox 22+ |

Để sử dụng các hàm lập trình WebRTC, các lập trình viên có thể sử dụng rất nhiều loại ngôn ngữ lập trình quen thuộc: Nếu viết trang web thì họ có thể dùng JavaScript, nếu viết ứng dụng (App) cho Android thì dùng Java, viết cho iOS thì dùng Objective-C, còn viết App cho Windows thì dùng $C^{++}$. Nếu gọi điện cho nhau bằng trình duyệt Chrome trên Android thì không cần cài thêm gì.

Tuy nhiên, có sự khác biệt về số lượng hàm API WebRTC được hỗ trợ trong các trình duyệt Web. Điều này làm giảm đi khả năng hoạt động của các ứng dụng WebRTC. Do đó, lập trình viên sẽ phải dành nhiều công sức hơn để tinh chỉnh lại trang Web hoặc App của mình cho phù hợp với từng trình duyệt, phần nào giảm đi lợi ích cốt lõi của WebRTC, chưa kể đến rủi ro phát sinh lỗi cao hơn.

Ngày 3/8/2016, Hiệp hội W3C đã đưa ra bản Dự thảo mới nhất của WebRTC1.0, đề cập đến những thay đổi liên quan đến số lượng lớn các điều khiển cấp thấp truy cập qua RTCRtpSender, RTCRtpReceiver, và RTCRtpTransceiver. Bao gồm: Làm rõ hoạt động khi **setDirection**() và làm rõ ảnh hưởng của **RTCRtpReceiver.track.stop**(). Cụ thể:

- RTCRtpReceiver.track.stop(): dừng cuộc gọi đối tượng liên quan của RTCRtpReceiver.

- Khi **setDirection()** có hiệu lực: RTCRtpTransceiver.setDirection() là đề nghị cách mới để thay đổi hướng cuộc gọi.

- RTCRtpTransceiver.stop (): để dừng lại người gửi và người nhận.

Như vậy, các hàm WebRTC vẫn đang được tiếp tục cập nhật chuẩn hóa.

**b. Chuẩn video**

WebRTC chưa có chuẩn video thống nhất. Đặc điểm nổi bật nhất của WebRTC là khả năng kết nối video từ trình duyệt web của người sử dụng. Nhưng hiện nay, các hãng trình duyệt cũng chưa thống nhất với nhau là chuẩn video nào sẽ được dùng cho WebRTC.

Google và Mozilla thì muốn dùng VP8 hoặc VP9, một **codec video** do chính Google phát triển theo mô hình mã nguồn mở. Thực tế cho thấy, đối với trình duyệt web Chrome 50 trở lên, codec VP9 tốt hơn. Nó hỗ trợ cho chất lượng tốt hơn trong truyền tải phương tiện truyền thông nhờ khả năng dự phòng (RED), cơ chế FEC tương tác với ước lượng băng thông và kiểm soát các thông số (NACK, POI, RTX, remb, ...) trong trình duyệt, khả năng tương tác với các trình duyệt khác tốt hơn VP8. Do đó, VP9 cho độ phân giải cao hơn, giảm thiểu các lỗi *(nhờ hỗ trợ cho Error Correction Forward).*

Trong khi đó, Microsoft và một số công ty khác thì muốn đề xuất sử dụng H.264 hoặc H.265 cho WebRTC, vốn đang là codec được dùng phổ biến nhất hiện nay trên Internet. Và tất nhiên, H.264 lại thuộc quyền sở hữu của Hiệp hội MPEG LA và phải trả phí bản quyền để sử dụng.

Hiện nay, nhóm làm việc của Google gồm Niklas Blum, Justin Uberti và Per Ahgren đã thực hiện được những cải tiến về hiệu suất, đó là trình duyệt Chrome đã có thể bớt thời gian kết nối trung bình từ 2 giây xuống còn 650ms cho một kết nối 1Mbps. Họ cũng đang làm việc để mang lại VP9 – codec video

đến các thiết bị di động, nhờ đó sẽ cho phép đạt chất lượng video cao với yêu cầu băng thông ít hơn.

**c. Tính an toàn, bảo mật của công nghệ WebRTC**

• Công nghệ WebRTC sử dụng giao thức bảo mật DTLS và SRTP.

- DTLS (Datagram Transport Layer Security) cho phép các ứng dụng được thiết kế để ngăn chặn việc nghe trộm, giả mạo, hoặc tin nhắn giả mạo. Các gói giao thức DTLS đảm bảo các ứng dụng không bị chậm trễ kết hợp với việc xử lý khi mất gói tin và dữ liệu.
- Giao thức SRTP (Secure Real-time Transport Protocol) cung cấp tính bảo mật, tính toàn vẹn và xác thực thông báo trong môi trường truyền thông (tiếng nói và hình ảnh). Bằng việc sử dụng thuật toán dẫn xuất khoá nguyên thuỷ, SRTP có khả năng tối thiểu hoá tính toán và sử dụng tài nguyên để sinh các khoá mật mã qua cơ chế quản lý khoá ngoài.

• Mã hóa là bắt buộc đối với tất cả các thành phần WebRTC, bao gồm các cơ chế báo hiệu (signaling).

• WebRTC được không phải là một plugin: thành phần của nó chạy trong sandbox của trình duyệt, các thành phần không yêu cầu cài đặt riêng, và được cập nhật bất cứ khi nào trình duyệt được cập nhật.

Sandbox là một kỹ thuật quan trọng trong lĩnh vực bảo mật có tác dụng cô lập các ứng dụng, ngăn chặn các phần mềm độc hại để chúng không thể làm hỏng hệ thống máy tính, hay cài cắm các mã độc nhằm ăn cắp thông tin cá nhân của người sử dụng.

**Tóm tại, công nghệ WebRTC với những ưu điểm nổi bật sẽ mang lại các ứng dụng phong phú, chất lượng cao và chạy theo thời gian thực. Công nghệ này có thể được phát triển cho các trình duyệt, nền tảng di động, thiết bị Internet of Things, và cho phép tất cả chúng liên lạc với nhau thông qua một bộ giao thức chung.**

---

***Tài liệu tham khảo:**


1. Sam Dutton, "Getting Started With WebRTC", tại website: https://www.html5rocks.com/en/tutorials/webrtc/basics/ , 21/02/2014.
2. "A Study of WebRTC Security" tại website: http://webrtc-security.github.io/
3. Trang Web: https://webrtc.org/