

WebRTC 기반 컨퍼런싱 시스템의 기술 분석

김성환*, 하윤기*, 최규범*, 윤찬현*

*한국과학기술원 전기 및 전자 공학과

e-mail : {s.h_kim, milmgas, mosfet1kg, chyoun}@kaist.ac.kr

Technical Analysis of WebRTC based Conferencing System

Seong-Hwan Kim*, Yun-Gi Ha*, Gyu-beom Choi*, Chan-Hyun Youn*

*Dept. of Electrical Engineering, Korea Advanced Institute of Science and Technology

요 약

WebRTC(Web Real-Time Communication)는 플러그인의 도움 없이 간단한 웹 표준 API(Application Programming Interface)를 이용하여 웹 브라우저 환경에서 Peer-to-Peer 실시간 통신을 가능케 하고자 하는 기술이다. 실시간 통신으로 오디오, 비디오 스트림은 물론이고 데이터 스트림을 포함한다. 해당 기술은 브라우저간에 직접적으로 Peer-to-Peer 세션을 형성하여 스트리밍을 수행하므로 중계 서버를 이용하는 통신에 비하여 향상된 네트워크 성능을 보이지만 연결을 구성하기 위한 보조 기술들이 요구된다. WebRTC 는 웹 기반 기술이기 때문에 기존의 웹 어플리케이션이 가지는 장단점을 공유한다. 본 논문에서는 WebRTC 기술과 해당 기술을 구성하고 있는 세부 기술들의 구조 및 기능을 분석한다. 또한 해당 기술을 이용하여 간단한 컨퍼런스 시스템을 설계하고 구현한 예제를 보인다.

1. 서론

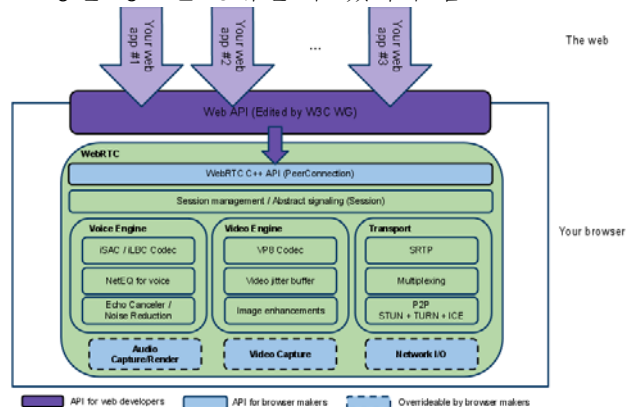
WebRTC[1]는 웹 브라우저간에 직접적인 통신 세션을 구성하여 오디오, 비디오 및 데이터 스트림을 송수신하기 위해 설계된 웹 표준을 지향하는 통신 프로토콜의 조합이다. 기존의 컨퍼런싱 시스템은 네이티브 어플리케이션이나 웹 브라우저 상의 플러그인을 (i.e. Adobe Flash 기술) 이용한 RIA 의 형태를 가진다. 반면 WebRTC 기반의 컨퍼런싱 시스템은 브라우저에서 지원하는 간단한 웹 표준 JavaScript API 를 이용한 웹 어플리케이션의 형태이기 때문에 설치, 업데이트 및 배포의 용이함, 멀티플랫폼을 지원하는 높은 접근성 등의 장점과 네이티브에 앱에 비해 낮은 성능, 하드웨어 접근성이 떨어지는 단점 등 기존의 웹 어플리케이션들이 보이는 장단점을 공유한다. 브라우저간 직접적으로 세션을 형성하여 데이터를 송수신하므로 좋은 네트워크 성능 (latency, delay, loss rate, ...)을 보일 수 있다. 다만 브라우저 간의 직접적인 통신 세션을 형성하기 위해서 Abstract Signaling 기술을 이용하며 NAT 와 방화벽과 같은 제약사항을 극복하기 위해 NAT Traversal 기술과 같은 보조 기술을 이용한다. 최근 기업에서는 이러한 특징들을 수용하여 WebRTC 를 이용한 사내 컨퍼런싱 시스템을 도입하려는 요구 사항이 발생하는 추세이다[2]. 본 논문에서는 WebRTC 를 이용한 응용 서비스의 요구가 높아지는 추세에 맞추어 WebRTC 기술과 해당 기술을 구성하고 있는 세부 기술들의 구조 및 기능을 분석한다. 또한 해당 기술을 이용하여 간단한 컨퍼런스 시스템을 설계하고 구현한 예제를 보인다.

2. WebRTC 의 구조

WebRTC 는 그림 1 과 같이 웹 브라우저 내에 스크립트가 아닌 네이티브 C++ 코드를 기반으로 작성된 소프트웨어 스택으로 Session Management, Abstract Signaling, Voice/Video Engine, Transport Protocol 및 이러한 기능을 웹 상에서 제공하기 위한 JavaScript API Wrapper 로 구성되어 있다. 핵심적인 모듈의 네이티브 코드 작성을 통해 웹 기술이 가지는 낮은 성능을 극복하려고 노력하였으며 W3C 를 통해 표준화 한 JavaScript API [3]를 통해 웹 표준으로 제공 가능한 멀티플랫폼 기술이다.

WebRTC 를 사용하기 위해서는 다음과 같은 요구사항을 만족하여야 한다.

- 웹 브라우저에서 API 를 통해 디바이스의 음성, 화상, 데이터 스트리밍을 받아들 수 있어야 함
- NAT 이나 Firewall 이 존재하더라도 peer 의 네트워크 정보, Resolution, Codec 과 같은 미디어 정보를 상호간 공유할 수 있어야 함



(그림 1) WebRTC 의 구조 [1]

- 통신을 생성/유지/종료 할 수 있도록 Signaling 과정을 통하여 세션 컨트롤을 할 수 있어야 함
- 세션을 통해 스트림 송수신이 가능해야 함



(그림 2) WebRTC 요구 기술

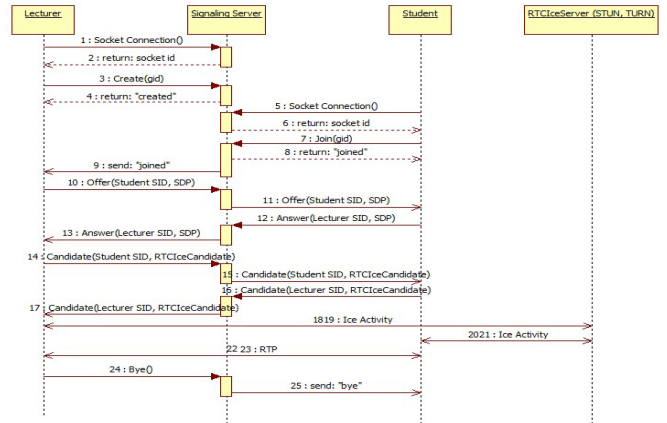
해당 요구사항을 만족하기 위하여 WebRTC 제공하는 API 및 기술들을 아래에 기술하였으며 필요 커넥션 및 프로토콜을 그림 2에 나타내었다.

- **MediaStream API:** 사용자의 카메라, 마이크와 같은 디바이스의 음성, 화상 스트림을 획득 가능
- **RTCPeerConnection API:** 암호화, 대역폭 제어 기능과 함께 음성, 화상 통신을 제공. UDP를 기반으로 하는 RTP(Real-time Transport Protocol) 프로토콜을 이용하여 송수신 패킷의 구조를 정의하고 RTCP(RTP Control Protocol)을 통하여 송수신 흐름을 제어
- **RTCDataChannel API:** Peer 간 데이터 통신을 제공
- **Signaling 기술:** 세션의 생성/유지/종료를 위해 상호간 메시지 전송 프로토콜로 특정 규격을 정하지 않은 Abstract Signaling을 이용하며 세션 생성을 위해 상호간 네트워크, 미디어 정보교환을 위해 SDP (Session Description Protocol)를 이용. 기존 SIP(Session Initiation Protocol)를 이용한 Signaling 시스템의 구축도 가능
- **NAT Traversal 기술:** 브라우저간의 직접적인 세션 연결을 위해 브라우저 간의 네트워크에 존재하는 연결 방해요소인 NAT 나 Firewall을 극복하기 위한 기술. NAT 과 Firewall의 종류를 검출하고 NAT의 Hole-punching을 통해 연결이 가능하도록 하며 최악의 경우 중계 서버를 이용하여 통신을 가능케 함. 지원 기술로 ICE가 있고 다수의 STUN과 TURN 서버를 이용하여 최적의 네트워크 연결을 수행

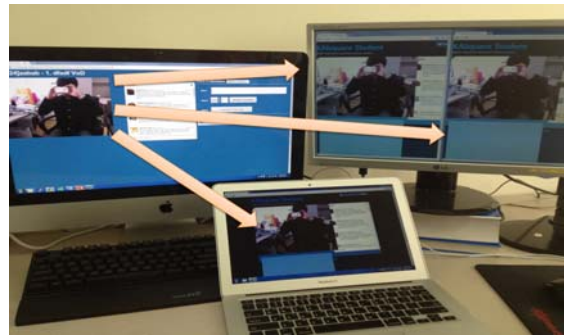
3. WebRTC 기반 컨퍼런스 시스템

WebRTC 시스템을 구축하기 위해서는 2장에서 기술한 바와 같이 웹 페이지 상에서 음성 및 화상 스트림을 획득하고 Peer Connection을 생성하기 위한 API 제어가 필요하다. 또한 세션 생성을 위한 Signaling Server와 NAT Traversal을 위한 STUN, TURN 서버가 필요하다. 이외의 실제적인 미디어 전송 작업은 웹 브라우저 내의 WebRTC 스택에서 수행하게 된다. 웹 서버와 Signaling 겸용의 서버의 구축하기 위하여 Multi Thread 기반이 아닌 Event 기반으로 동작하는 JavaScript 기반 서버 기술 node.js를 기반으로 작성하

였으며 WebSocket, AJAX long polling 등의 기술을 이용하여 브라우저 환경과 버전에 독립적으로 실시간 메시지 전달을 수행할 수 있는 Socket.IO 모듈을 이용하여 Signaling Event 제어를 수행하였다. STUN과 TURN 서버는 구글에서 제공하는 공용 도메인의 서버를 이용하였으며 해당 기술들의 조합으로 그림 2와 같은 환경을 구축하고 그림 3과 같은 세션 구성 프로시저를 통해 그림 4과 같은 웹 기반 실시간 강의 시스템을 구축하였다.



(그림 3) 세션 구성 Sequence Diagram



(그림 4) WebRTC 기반 실시간 강의 시스템

4. 결론

본 논문에서는 웹을 통해 화상, 음성, 데이터를 공유할 수 있는 실시간 통신 기술인 WebRTC의 구조, 기능 및 세부 기술들을 분석하고 이를 응용하여 실시간 강의 시스템을 설계하고 구현한 예제를 보였다. 웹의 높은 접근성이 가지는 장점과 점진적인 표준화를 통해 WebRTC는 앞으로 더욱 발전할 것으로 예상된다.

Acknowledgement

본 논문은 2013년도 지식경제부의 재원으로 '유전체 분석용 슈퍼컴퓨팅 시스템 개발' 사업의 지원을 받아 수행된 연구임(No. 10038768)

참고문헌

[1] <http://www.webrtc.org/>
 [2] Alan Johnston 외 2명, "Taking on WebRTC in an Enterprise"
 [3] <http://dev.w3.org/2011/webrtc/editor/webrtc.html>