

실시간 미디어 통신을 위한 하이브리드 웹앱

Hybrid Web App for Real-Time of Media Communications

최성자(순천향대학교)

차 례

1. 서론
2. 실시간 미디어 통신 플랫폼
 - 2.1. 플랫폼 및 통신프로토콜
 - 2.2. 실시간 미디어 통신 API
3. 결론

■ keyword : | webRTC | Media Communiccation | Server | Real Time |

요약

H/W, S/W 기술과 통신 기술이 발전함에 따라 기술융합이 혁신적으로 이루어지고 있으며, 다양한 미디어 통신을 위한 기술 또한, 쉽고, 빠르게 진화하고 있다. 미디어 통신은 데이터 유형이 다양하고, 미디어 스트림의 경우 전송량이 방대하며, 전송기술 또한 다양하게 적용된다. 최근에는 웹을 활용한 전송기술을 제공하고 있으며, 이에 대한 표준기술로 webRTC 기술을 제공하고 있다. 무비용의 빠르고 쉽게 미디어 전송을 가능하게 하며, 다양한 스마트 및 미디어기기에 동일한 플랫폼이 제공 가능함으로써 개발비용을 단축시킬 수 있다. WebRTC 기술발전 과정 및 제공되는 플랫폼과 프로토콜 스택을 살펴보고, 멀티플랫폼이 제공 가능한 하이브리드 스마트 앱을 제공한다.

1. 서론

인터넷에서 사운드나 비디오 파일을 실시간 전송, 구현하는 기술을 스트리밍이라 하며, 여러 개의 파일 형태로 나누어 흐르듯이(Streaming) 연이어 전송한다. 스트리밍 서비스는 서버와 클라이언트로 구성되며, 인터넷 스트리밍 프로토콜(RTP Streaming)을 사용하여 RTP/UDP/IP 패킷에 실릴 수 있도록 한 스트리밍 프로토콜을 활용하여 오디오/비디오등의 정보를 담은 일련의 패킷화된 정보앞에 RTP 헤더 스트림을 추가하여 전송한다.[1],[2],[3]

1989년 CERN에서 인터넷의 역사가 시작되었으며, HTTP는 메시지 처리 방식으로써, 클라이언트와 서버사

이에 이루어지는 요청/응답 프로토콜이다. 웹상에서 정보를 주고 받을 수 있는 프로토콜이다.[4][5]

2004년 이후 대화형 웹 어플리케이션 제작을 위해 AJAX(Asynchronous JavaScript an XML)솔루션이 제공되었으며, AJAX는 페이지 이동 없이 고속으로 화면을 전환할 수 있고, 비동기 요청이 가능하며, 수신 데이터 양을 줄일 수 있다. 서버 측에서는 SOAP나 XML기반의 웹 서비스 프로토콜을 사용하며 클라이언트 측에서는 자바 스크립트를 사용한다.[6],[7]

2008년 소켓을 이용하여 자유롭게 데이터를 주고 받을 수 있도록 하며, 기존의 요청/응답 관계 방식보다 더 쉽게 데이터를 교환할 수 있으며, WebSockets 프로토콜을 적용한다.[8],[9]

2012년 WebRTC를 통해 오디오나 비디오 스트림을 P2P(Peer-to-Peer)로 송수신하는 것 뿐 아니라, 데이터 전달을 위한 메커니즘을 포함하며, 클라이언트간의 빠른 데이터 교환기능을 제공한다. 또한, 웹브라우저 간에 플러그인의 도움 없이 서로 통신할 수 있도록 설계된 API이며, 음성통화, 영상통화, P2P파일 공유 등으로 활용가능하다.[10],[11],[12]

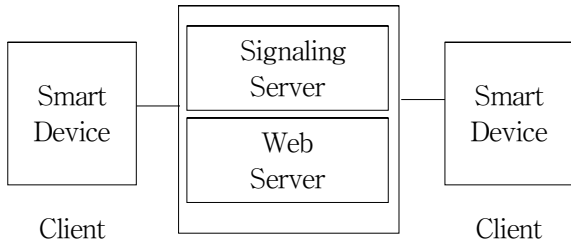
2. 실시간 미디어 플랫폼

WebRTC란 실시간 플러그인 필요없는 영상과 음성, data 통신을 지원하기위한 웹브라우저자체에서 제공되는 무료 오픈소스 API의 표준이다. WebRTC API 유형은 사용자의 카메라와 마이크등의 데이터 스트림 접근 및 암

호와 및 대역폭 관리 기능제공, 오디오, 비디오 연결하고 P2P 통신의 데이터 채널을 제공한다.

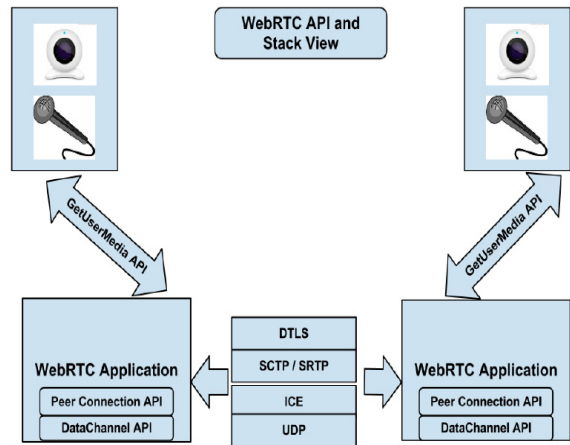
2.1 플랫폼

실시간 P2P 통신을 위해 웹 서버와 시그널링 서버를 구축하여 webRTC 기능을 제공하였다.



▶▶ 그림 1. 실시간 미디어 통신 플랫폼

리눅스 웹서버(CentOS 7)을 기반으로 시그널링 서버는 Asterisk 10을 설치하여 오픈소스 시그널링 서버를 구축하여 실시간 미디어 스트림 서비스를 제공한다. 제시된 솔루션의 특성은 리스닝 포트를 오픈시킬 필요가 없이 실시간 통신이 이루어지므로, 서버에 부하를 줄일 수 있다. 또한, 웹 서버상의 보안계층을 설정함으로써 시그널링 서버에서는 따로 보안계층을 설정하지 않아도 된다. 클라이언트 측에서는 동일한 도메인과 포트를 사용한다. WebRTC 클라이언트 측 주요 API는 상대측 연결 API(Peer Connection API)와 데이터 채널 API(Data Channel API)로 구성되어 있으며, 미디어 통신을 위한 시그널링 채널 구축을 위한 서버 측 주요 프로토콜은 웹 브라우저 간에 피어 투피어 접속을 할 수 있게 해주는 ICE(Interactive Connectivity Establishment), IP 네트워크로 PSTN 시그널링 메시지를 전송하기 위한 프로토콜로 SIGTRAN(Signaling Transport)를 구성하는 요소의 하나인 스트림 제어 전송 프로토콜(SCTP: Stream Control Transmission Protocol)과 실시간으로 전송되는 멀티미디어 데이터를 암호화하여 송수신하는 실시간 전송 보안 프로토콜(SRTP: Secure Real-time Transport Protocol)로 계층화 되어있으며, DTLS (Datagram Transport Layer Security) 프로토콜은 데이터그램 프로토콜을 위한 통신 프라이버시를 제공한다.[14]



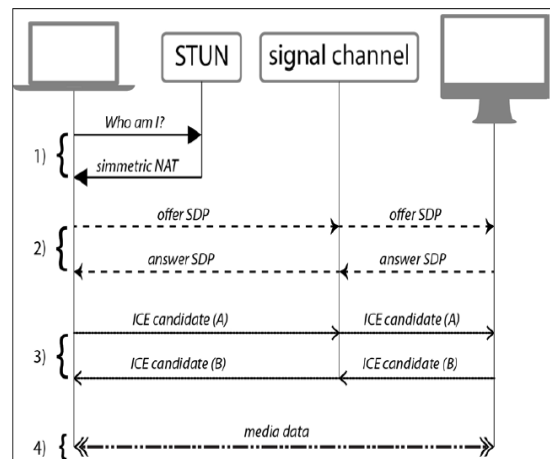
▶▶ 그림 2. WebRTC API 와 스택 뷰

2.2 미디어 통신 기능

WebRTC는 p2p통신이 가능하며, 시그널링을 통해 클라이언트의 통신을 조정하고, 메타데이터 서버와 네트워크 주소변환(NAT) 및 방화벽을 위한 서버가 필요하다. 시그널링 서비스를 구축하기위해 STUN과 TURN 서버를 통해 실제 연결을 위한 타협이 이루어진다.[13]

- STUN Server: 공용주소외에 webRTC 피어(Peer)와 통신하기 어려우므로, 공용 인터넷과 동작하여 요청된 IP:port, 주소를 확인하고 그 주소를 응답으로 되돌려 주는 기능을 수행한다.
- TURN Server: 피어(Peer)들 사이의 오디오/비디오/데이터 스트리밍 릴레이 기능을 제공한다.

그림 3에서는 STUN 서버에 연결된 시그널 통신예를 보여주고 있다.[14]



▶▶ 그림 3. Signal Channel Flow

미디어 통신을 위한 구현 절차는 다음과 같다.

- ① 스트리밍 오디오, 비디오 또는 데이터 가져오기
- ② IP주소, 포트등의 네트워크 정보 가져오고, peers로 알려진 다른 WebRTC클라이언트들과 연결을 위해 정보교환, (NAT/방화벽을 통해 정보교환가능)
- ③ 에러 보고, 세션 초기화, 종료를 위한 신호 통신관리
- ④ 해상도와 코덱등의 미디어와 클라이언트의 capability에 대한 정보 교환
- ⑤ 스트리밍 오디오, 비디오 또는 데이터를 주고 받음

주요 WebRTC에서 제공 가능한 서비스는 표 1에서 보여주고 있다.

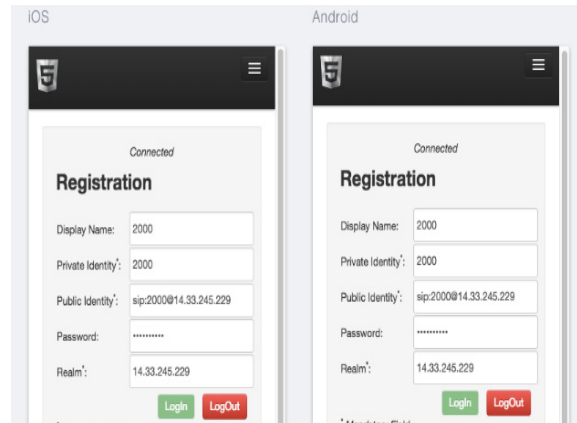
표 1. WebRTC 서비스 기능

Photo Snap Camera Transmit	Password Protection via Cipher
WebRTC Dialing	Password Protection via Cipher
WebRTC Call Receiving	Event History and Call Records
JSON App Messaging	Photo History and Recording Static Snapshots of Calls (only stills)
Multi-party Calling	Connectivity Detection and Auto-Recovery
Audio only Calls Optional	Pre-configured Video Element for Streaming Video/Audio
Broadcasting Mode	Pre-configured Local Camera Video Element for Streaming Video/Audio
Instant Connect Dialing Optional	Network Connectivity Hooks (online/offline)
Security SSL, AES256, ACL Access Control Management	SDK Level Debug Output

제공 가능한 웹브라우저는 Chrome, Firefox, Opera 등으로써, 멀티 플랫폼이 지원되며 다양한 미디어 통신을 실시간으로 제공가능하다.

3. 결론

스마트 환경에서, 무료 네트워크 통신을 통해서 다양한 멀티 미디어 서비스제공을 실시간을 제공하는 플랫폼을 제시하며, 이를 기반으로 하이브리드 웹앱을 구현하였다. 그림 4에서는 피어 투 피어 연결을 위한 하이브리드 웹 앱화면을 보여주고 있다.



▶▶ 그림 4. P2P 연결 설정 화면

제공된 하이브리드 웹 앱을 통해 미디어 스트림을 무료로 송수신 가능하며, 더불어 WebRTC의 적용으로 브라우저간 통신을 통한 미디어 스트림 서비스를 제공 받을 수 있다. 본 콘텐츠를 통해 빠르고 쉽게 미디어 콘텐츠를 제어함으로써 스마트 기기의 활용도를 높일 수 있다.

참고문헌

- [1] FitzGerald, Cary, and Cullen Jennings. "Method and apparatus for media stream monitoring." U.S. Patent No. 7,310,334. 18 Dec. 2007.
- [2] Rand, Ricky. "Apparatus and method for selective insertion and pricing of media data into a media data stream." U.S. Patent Application No. 10/148,292.
- [3] Foote, Jonathan, and Matthew Cooper. "Method and system for harvesting a media stream." U.S. Patent Application No. 10/841,082.
- [4] Gollmann, Dieter. "Computer security." Wiley Interdisciplinary Reviews: Computational Statistics 2.5 (2010): 544-554.
- [5] Franks, John, et al. "HTTP authentication: Basic and digest access authentication." (1999).
- [6] Paulson, Linda Dailey. "Building rich web applications with Ajax." Computer 38.10 (2005): 14-17.
- [7] Galhardas, Helena, et al. "AJAX: an extensible data cleaning tool." ACM Sigmod Record. Vol. 29. No. 2. ACM, 2000.
- [8] Pimentel, Victoria, and Bradford G. Nickerson. "Communicating and displaying real-time data with WebSocket." Internet Computing, IEEE 16.4 (2012): 45-53.
- [9] Wang, Vanessa, Frank Salim, and Peter Moskovits. The definitive guide to HTML5 WebSocket. Vol. 1. Berkeley, Calif, USA: Apress, 2013.

- [10] Johnston, Alan B., and Daniel C. Burnett. WebRTC: APIs and RTCWEB protocols of the HTML5 real-time web. Digital Codex LLC, 2012.
- [11] Rescorla, Eric. "WebRTC Security Architecture." (2015).
- [12] Nurminen, Jukka K., et al. "P2P media streaming with HTML5 and WebRTC." Computer Communications Workshops (INFOCOM WKSHPs), 2013 IEEE Conference on. IEEE, 2013.
- [13] <https://wiki.asterisk.org/wiki/display/AST/Home>
- [14] Sergiienko, Andrii. WebRTC Cookbook. Packt Publishing Ltd, 2015.

저자소개

● 최 성 자(Choi, Sung Ja)



- 1991년 2월 : 한남대학교 컴퓨터공학과(공학사)
 - 1997년 2월 : 한남대학교 컴퓨터공학과(공학석사)
 - 2005년 8월 : 한남대학교 컴퓨터공학과(공학박사)
- <관심분야> : Bio 센서, 뇌공학
- E-Mail : irecomm@naver.com