

WebRTC 1.0 표준기술 개요 및 의미

글. 김평수 한국산업기술대학교 교수
pskim@kpu.ac.kr



최근, 웹 분야의 대표적인 국제 표준화 기구 W3C의 WEBRTC WG에서 개발된 문서 ‘WebRTC 1.0: Real-Time Communication Between Browsers’가 WebRTC API 권장 표준이 되었고 이를 통해 웹 어디에서나 오디오 및 비디오 통신을 제공할 수 있다고 공식 발표하였다. 한편, WebRTC 1.0 표준은 인터넷 분야의 대표적인 국제 표준화 기구인 IETF에서 개발한 프로토콜 표준과 함께 공동 개발되어 왔다.

본 고에서는 웹 브라우저 간에 플러그인이나 중간자 도움 없이 오디오 및 비디오 미디어를 P2P 연결 기반으로 서로 통신할 수 있도록 설계된 WebRTC에 대한 역사, 기술 개요, W3C 및 IETF 세부 표준기술 등을 소개하며, COVID-19 팬데믹 시대에서의 표준화의 의미를 제시한다.

서론

COVID-19 팬데믹으로 인해 재택근무와 원격수업에 필요한 시스템의 니즈가 확대되었고, 언택트(비대면) 산업 분야를 창출하고 활성화해야 한다는 목소리가 커지게 되었다. 근무와 수업 외에도 엔터테인먼트 분야에서의 유튜브, 넷플릭스, 모바일 게임 이용이 늘어나는 등 다양한 산업 분야에서 급변하게 되었다. 재택근무에 사용되는 다양한 ICT 서비스 중 가장 많이 사용한 것은 화상 회의라 할 수 있는데, 화상 회의의 비디오 및 오디오 스트리밍 기술은 미디어를 생성하고, 효율적인 전송을 위해 압축하고, 이를 P2P 혹은 중계 서버를 통해 실시간으로 전달하는 다양한 요소 기술들이 포함되어 있다. 아울러, 유튜브와 같은 1인 방송이 더욱 대중화되면서 시청자와 유튜버와 같은 스트리머가 다양한 방식으로 소통할 수 있게 되었는데, 이를 가능하게 할 수 있었던 이유 중 하나가 라이브 스트리밍을 처리하는 미디어 전송 프로토콜이라 할 수 있다.

본 고에서는 비디오와 오디오 데이터 전달에 사용되는 대표적인 기술인 WebRTC(Web Real-Time Communications) 프로토콜을 다룬다. 지난 2021년 1월 26일, W3C(World Wide Web Consortium)는 최근 발표를 통해 시스코, 구글, 모질라 주도로 WEBRTC WG에서 개발된 문서 ‘WebRTC 1.0: Real-Time Communication Between Browsers[1]’가 WebRTC API(Application Programming Interface) 권장 표준(W3C Recommendation)이 되었고 이를 통해 웹 어디에서나 오디오 및 비디오 통신을 제공한다고 밝혔다. 이 표준은 IETF(Internet Engineering Task Force) RTCWEB(Real-Time Communication in WEB-browsers) WG에서 개발한 프로토콜 표준과 함께 개발되어 왔다. 즉, 웹브라우저 간에 플러그인이나 별도 소프트웨어의 도움 없이 상호 통신이 가능하도록 하는 W3C의 WebRTC API는 전송 프로토콜인 TCP/UDP, 실시간 소통을 지원하는 RTP, 연결 과정을 지원하는 ICE 프로토콜 등의 여러 IETF 표준기술로 구성된 WebRTC 프로토콜 스택을 이용하여 동작한다.

WebRTC 기술의 개요 및 역사

WebRTC는 Web Real-Time Communications의 약자로 웹페이지에서 실시간 통신을 지원하는 프로토콜이며 웹 애플리케이션, 네이티브 앱 애플리케이션 및 웹사이트들이 별도의 소프트웨어 없이 비디오, 오디오와 같은 미디어뿐만 아니라 텍스트, 파일 같은 데이터를 웹브라우저끼리 실시간으로 주고받을 수 있게 만든 오픈 소스 기술이며 HTML5의 일부 기술이다[2]-[4]. 대표적인 웹브라우저인 구글의 크롬, 모질라의 파이어폭스, 오페라 등 최신 버전들과 iOS와 안드로이드 같은 운영체제 기반의 네이티브 앱 라이브러리를 지원한다. WebRTC 기술을 바탕으로 만들어진 잘 알려진 솔루션들로는 페이스북 메신저, WhatsApp과 같은 인스턴트 메신저, 언택트 화상 회의 열풍을 일으킨 Zoom을 포함 구글 Meet, 마이크로소프트 Teams 등이 있다. WebRTC는 전통적인 스트리밍 서비스처럼 서버-클라이언트 구조가 아니라, 사용자가 P2P 연결을 맺고 미디어 통신을 수행하는 구조이다.

구글은 2010년 오직 웹만으로 비디오와 오디오 실시간 통신이 가능한 솔루션을 개발하기 위해서 WebM 프로젝트를 시작하였고, 개방형 고화질 비디오 압축 포맷을 오픈소스로 개발하였으며, 2011년 5월 WebRTC를 오픈 소스화하여 API 형태로 공개하였다. 구글의 오픈 프로젝트에서 시작된 WebRTC는 이러한 문제를 말끔히 해결할 수 있었다. WebRTC는 HTML5와 Javascript API를 사용해 웹상에서 별도의 플러그인이나 소프트웨어를 설치할 필요 없이 실시간으로 오디오 통화, 비디오 통신, P2P 파일 공유 등을 할 수 있도록 해주었다. 표준기술 개발을 위해 2011년 웹 분야의 대표적인 국제표준화기구인 W3C 내에는 WebRTC의 표준과 API 정의를 정하기 위한 워킹그룹이 만들어졌고, 인터넷 분야의 대표적 국제표준화기구인 IETF에서도 프로토콜의 표준화 작업을 시작했다. 이후 WebRTC는 웹브라우저에서의 실시간 통신을 가능하게 한다는 혁신성을 강조하면서, 모질라, 오페라 등과 같은 웹브라우저 시장 주요 글로벌 기업들의 지원을 받으며 빠르게 성장했다. 2013년에는 모질라의 파이어폭스가 WebRTC를 적극 지원하였다. 2015년은 WebRTC의 성능 향상이 컸던 시기이며 WebRTC 관련 벤더들 간 교류도 활발하게 진행되었다. 2016년에는 크롬과 마이크로소프트 엣지가 오디오 연결에 성공했으며, 이는 이중의 웹브라우저 간에 WebRTC 동작이 가능함을 확인하게 되었고 구글과 마이크로소프트가 협력하고 있음을 알리는 계기가 되었다. 2016년 W3C 회의에서는 웹 기반 증강현실(AR)/가상현실(VR) 지원 등과 같은 WebRTC의 다음 버전에 대한 고려사항이 거론되었다. 2017년에는 애플의 사파리가 WebRTC를 지원하기 시작했고, 동시에 WebRTC의 표준이 완성 단계에 접어들게 됨에 따라 전 세계 대다수의 사람들이 사용하는 주요 브라우저들에서 WebRTC를 모두 지원하게 되었다. 이러한 일련의 상황을 통해, 2017년부터 표준이 곧 완료될 것이라고 기대를 했었지만 결국 이후로 3년 넘게 흘러 최근 2021년 1월 말에서야 표준이 완료되었다. 2017년 이후 지속적인 기

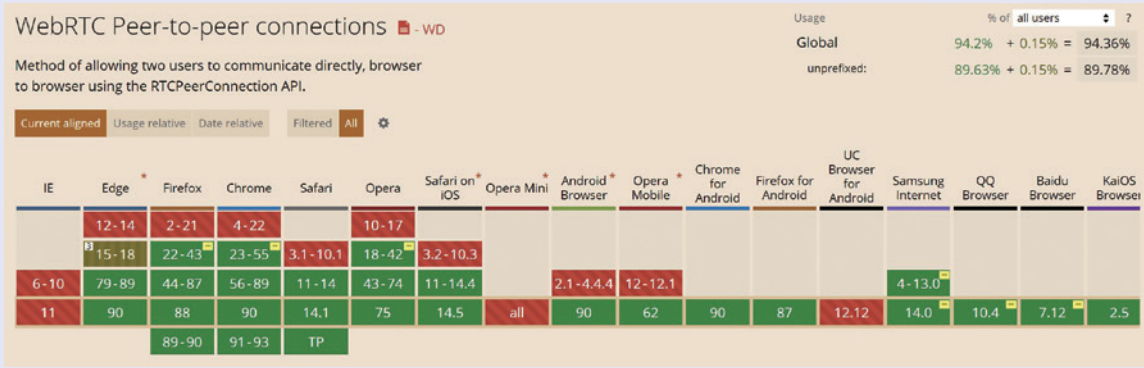


그림 1. WebRTC 지원 웹브라우저 현황 / 출처 : Can I use?(www.caniuse.com)

능 추가 및 개선이 진행되었으며, [그림 1]과 같이 ‘Can I use?’(www.caniuse.com)에 따르면 2021년 5월 현재 WebRTC를 지원하는 브라우저는 마이크로소프트 엣지, 파이어폭스, 크롬, 사파리, 오페라, iOS 사파리, 오페라 모바일, 안드로이드 크롬, 삼성 인터넷 등이 있음을 확인할 수 있다[4].

W3C 및 IETF의 WebRTC 관련 표준 기술

W3C WebRTC API 표준기술 개요

WebRTC API는 W3C의 웹 표준에서 웹이 문서저장, 출판의 역할을 넘어 프로그래밍 가능한 플랫폼으로 진화하는데 실시간 미디어 통신에 대한 역할을 담당하고 있는 중요한 표준이다. WebRTC API 표준화의 시작은 2010년으로 거슬러 올라가는데, 구글이 주최하고 W3C, 모질라, 마이크로소프트, 인텔, 애플, 아이비엠, 에릭슨, 시스코, 로지텍, 오페라, 야후 등 여러 업체가 참석한 RTC Web 워크숍에서 웹에서 실시간 미디어 통신을 하기 위해 필요한 통신 방법, 코덱, 암호화, 보안, API 수준 등의 내용을 처음으로 논의했다. 이후, W3C WebRTC 워킹그룹에서 로컬 미디어 장치 연동을 위한 API 규격 작업을 진행하였다[3].

WebRTC API 기술은 P2P 통신에 최적화가 되어 있으며 다음과 같이 3가지 세트로 구성되어 있다고 할 수 있다.

- 오픈소스 코덱(비디오, 오디오)을 포함한 미디어 엔진 세트
- 통신 프로토콜 세트
- 자바스크립트 API 세트

아울러, WebRTC의 세부적인 API는 [표 1]과 같다.

주요 API	역할
getUserMedia	사용자의 디바이스에서 오디오와 비디오 데이터 추출할 장치를 준비. 미디어 장치에 접근하여 캡처하기 위한 API
getDisplayMedia	로컬 장치의 디스플레이에 접근하기 위한 API
MediaRecorder	사용자의 디바이스에서 오디오와 비디오를 녹화하기 위한 API
RTCPeerConnection	사용자 간(P2P) 오디오와 비디오 스트림 전달을 위한 연결 생성을 위한 API
RTCDataChannel	사용자 간 실시간 데이터(텍스트, 파일, 기타 데이터)를 송수신하기 위한 API
getStats	PeerConnection의 상태 정보에 접근하기 위한 API

표 1. WebRTC의 세부적인 API

IETF WebRTC 프로토콜 표준기술 개요

이미 언급한 대로, 웹브라우저 간에 플러그인이나 별도 소프트웨어의 도움 없이 상호 통신이 가능하도록 하는 W3C의 표준기술인 WebRTC API는 전송 프로토콜인 TCP/UDP, 실시간 소통을 지원해주는 RTP, 연결 과정을 지원하는 ICE 프로토콜 등의 여러 IETF 표준기술로 구성된 WebRTC 프로토콜 스택을 이용하여 동작된다. 다양한 IETF 표준기술로 구성된 WebRTC 프로토콜 스택은 [그림 2]와 같이 나타낼 수 있으며, 스택을 구성하는 세부 프로토콜은 [표 2]와 같다 [4][5].

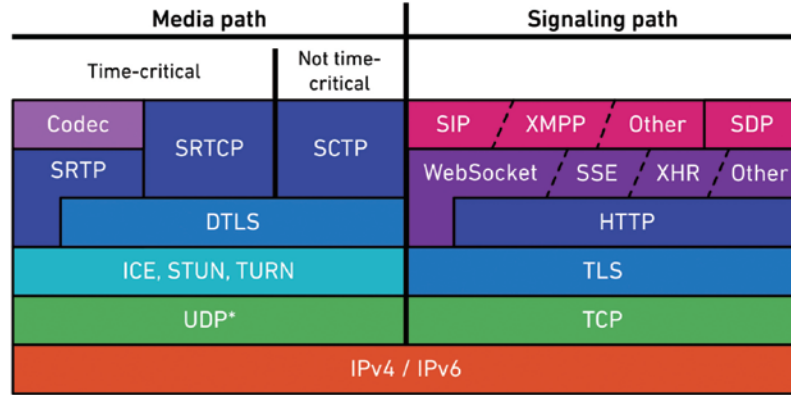


그림 2. WebRTC 프로토콜 스택 / 출처 : M Meszaros, F. Trojahn, Bachelor Thesis, Hochschule für Telekommunikation Leipzig, December 2017[5]

주요 프로토콜	역할
NAT(Network Address Translation)	연결된 인터넷 디바이스에 사설(Private) IP 부여
ICE(Interactive Connectivity Establishment)	P2P 간 직접 통신을 위한 기술로 통신을 위한 최적의 경로 탐색
STUN(Session Traversal Utilities for NAT)	기기의 공인 IP를 찾고 P2P 연결을 방해하는 요소를 탐지
TURN(Traversal Using Relays around NAT)	STUN 서버를 이용한 연결이 실패한 경우 오디오, 비디오 등을 릴레이
SDP(Session Description Protocol)	P2P로 서로 간 주고받는 데이터에 대해 이해하는데 사용
ICE Candidate	데이터 교환을 위한 경로들로 SDP 결정 이후 최적의 ICE를 선택하여 통신
Signaling Server	기기 사이에서 커넥션을 생성하는 역할

표 2. WebRTC 주요 프로토콜

전체적인 WebRTC의 동작 개념도는 [그림 3]과 같이 표현할 수 있으며 각 세부과정을 설명하면 다음과 같다. 만약, 두 디바이스(Peer) A와 B 사이에 기존의 전통적인 서버가 중간자 역할을 한다면 사용자 간 통신의 모든 부하를 서버가 떠 맡게 되고 확장성에 한계가 생기며 유지하는데 비용이 증가한다. A와 B가 직접 P2P 통신을 하게 된다면 사생활 보호 측면에서 유리하며 서버 유지비용도 없다. 물론, 두 디바이스 A와 B에 유일한(Unique) 공개(Public) IP 주소가 할당되었다면 두 디바이스는 서로 어렵지 않게 통신할 수 있다. 보통 디바이스들은 각각의 라우터에 연결되어 있고, 통신을 한다는 것은 라우터가 상대방의 위치까지 찾아갈 수 있다는 것을 의미한다. 그런데, 공유기와 같은 라우터는 공개 IP 주소를 갖고 있고 대부분의 디바이스들이 라우터에 연결되어 있으며 해당 라우터로부터 사설(Private) IP 주소를 할당받는 NAT(Network Address Translation) 구조를 갖는다. 결국, 보통의 인터넷 환경에서 두 디바이스 A와 B가 직접 P2P 통신을 하는 것은 어려울 수 있다. 따라서, P2P를 지향하는 WebRTC에서는 이러한 문제를 해결하기 위한 요구 사항을 고려하였다.

WebRTC 동작은 시그널링(Signaling), 연결(Connection), 보안(Security), 통신(Communication)의 4단계로 이루어

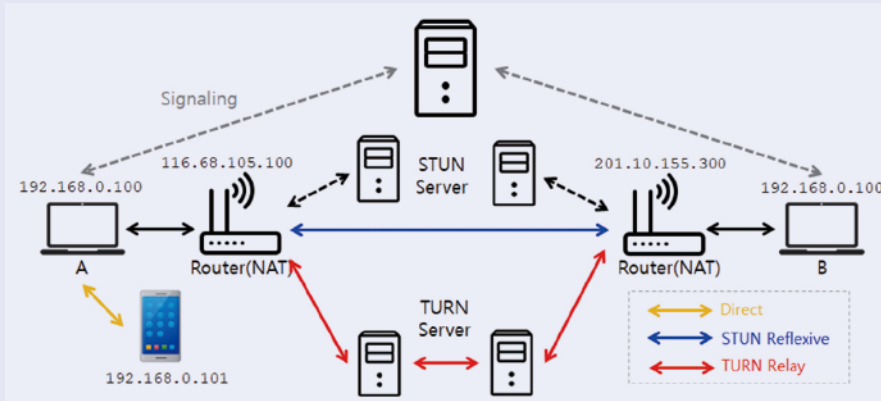


그림 3. WebRTC 동작 구성도 / 출처: 김평수, 'IETF-W3C WebRTC 1.0 표준기술 개요와 팬데믹 시대에서의 의미', April 2021[4]

어지며 각 단계에서 다양한 IETF 프로토콜이 적용된다. 이 중에서 WebRTC 연결 과정에서 적용되는 IETF 표준 기술로는 ICE(Interactive Connectivity Establishment), STUN(Session Traversal Utilities for NAT), TURN (Traversal Using Relays around NAT)이 있다. ICE는 RFC8445로 표준화된 기술로 웹브라우저의 P2P 연결이 가능하도록 하게 해 주는 프레임워크이다.

WebRTC 기반의 P2P 통신을 하기 위해서는 3가지 경우를 고려한다. 첫 번째의 경우는 라우터에 접속되어 있는 두 디바이스 간의 P2P 통신이다. 이미 언급했듯이, 공개 IP 주소를 갖는 라우터에는 여러 사설 IP 주소가 내부적으로 라우팅되어 있다. 라우터에 연결된 노트북과 핸드폰은 사설 IP를 사용하고 있지만 두 디바이스가 P2P 통신을 하는데 전혀 장애물이 없다.

두 번째의 경우는, 라우터에 연결된 디바이스가 다른 라우터에 연결된 디바이스와 통신하는 것이다. 노트북과 핸드폰처럼 사설 IP를 할당받은 디바이스들은 자신의 공개 IP 주소를 모른다. 따라서, Peer A가 다른 라우터에 연결된 Peer B와 통신한다면 A는 자체적으로 자신의 공개 IP 주소를 알고 있어야 한다. 이것을 가능하게 하는 것이 STUN 서버이다. STUN은 IETF RFC8489로 표준화된 기술이며 Peer가 자신의 공인 IP 주소를 발견하는 프로토콜이다. 즉, 서로 다른 라우터에 연결된 두 Peer가 통신하기 위해서 자신의 공개 IP 주소를 알아내기 위해 STUN 서버를 사용한다. Peer A는 STUN 서버를 통해 자신의 공인 IP 주소를 알게 된다. 이렇게 알아낸 공인 IP 주소를 토대로 상대방과 통신한다. 마찬가지로 Peer B도 자신의 공인 IP 주소를 알아낸 뒤 이를 기반으로 통신한다.

세 번째 경우는 위의 두 가지 경우와 다르게 어떠한 이유로 P2P 통신이 불가능한 상황이다. 실제로 NAT 환경에서의 인터넷 통신은 매우 복잡하다. NAT는 일반적으로 요청하지 않은 패킷은 차단하는 특성 때문에 NAT 뒤에 있는 클라이언트에게 전달되는 패킷을 차단할 수 있으며, 이 때문에 P2P 간의 직접적인 연결을 불가능하게 만든다. Peer A에서 Peer B까지 단순히 연결하는 것으로는 작동하지 않는 것에 대한 이유는 많이 있다. 만약, 최종적으로 P2P 통신이 불가능하면 연결을 중재해줄 서버가 필요하며 이를 TURN 서버라고 부른다. TURN은 IETF RFC8656으로 표준화된 기술이며, 만약 P2P 연결이 방화벽이나 NAT 구성으로 인하여 결국 실패하면 TURN 중계 서버를 통해 통신한다. TURN 서버는 일종의 백업 서버라고 할 수 있다.

위에서 설명한 세 가지 경우의 모든 과정이 ICE라는 프레임워크 위에서 이루어지는 것이며 두 개의 디바이스가 P2P 연결을 가능하게 하도록 최적의 경로를 찾아주는 프레임워크가 되는 것이다. ICE 프레임워크가 STUN, 또는 TURN 서버를 이용해 상대방과 연결 가능한 후보들을 갖고 있다는 것이다. 그러니까 두 브라우저가 P2P 통신

을 위해 통신할 수 있는 주소만큼은 확실하게 알아내는 것이다. 결국, ICE는 연결 작업을 수행하기 위해 STUN과 TURN 서버 모두를 사용하거나 혹은 하나의 서버를 사용한다.

WebRTC 1.0 표준화 완료의 의미와 향후 계획

W3C-IETF 공동 협력을 통한 WebRTC 1.0 표준화 완료의 의미

10여 년 전, 리치 웹 애플리케이션이 초기 단계에 접어들었을 때 웹 및 실시간 통신 업계의 엔지니어들이 함께 모여 “IP 기술을 통해 최신 오디오 및 비디오 기술을 웹 기반 유비쿼터스 플랫폼으로 가져올 수 있을까?”라는 도전적인 문제를 결국 해결하게 된 것이다. 과거 IETF ART(Applications and Real-Time) Area 디렉터였으며, 현재는 IETF 전체 의장(Chair)을 맡고 있는 시스코의 Alissa Cooper는 “VoIP 범위를 극적으로 확대한 WebRTC 기술은 전 세계 커뮤니케이션 방법에 대변혁을 가져올 것이며 COVID-19 팬데믹 시대에서 WebRTC에 의해 수십억 인구가 장차나 지역 관계없이 서로 이어져 교류를 유지할 수 있을 것”이라고 밝혔다. W3C API와 관련된 프로토콜의 전체적인 시스템에 개요와 보안 요구 사항은 W3C 표준화 완료와 같은 시기에 완료된 IETF 표준문서인 RFC8825(Overview: Real-Time Protocols for Browser-Based Applications)와 RFC8826(Security Considerations for WebRTC)에서 확인할 수 있다.

WebRTC 1.0 표준으로 확정되기 전인 지난 몇 년간 WebRTC 기술은 웹 브라우저를 포함하지 않는 많은 서비스를 포함하여 오디오 또는 비디오를 사용하는 최신 서비스의 일부로 배포 및 사용되었다. WebRTC는 온라인 커뮤니케이션 및 협업 서비스의 초석 역할을 하고 있고 웹 플랫폼에 통합함으로써 그 범위가 크게 확장되었으며, IP 상에서의 오디오 및 비디오를 통해 사람들이 전 세계에서 소통하는 방식에 혁명을 일으켰다. 이번 WebRTC 1.0 공식 표준으로 인해 WebRTC 코드, API 및 표준의 가용성으로 인해 모든 애플리케이션에 실시간 통신 기능을 간단하게 추가할 수 있게 되었다. 웹브라우저, 기타 디바이스 및 플랫폼에서 로열티가 없는 기능으로 표준화되고 배포된 기반을 통해 WebRTC로 보안 오디오/비디오 통신 시스템을 설정하는 기능이 내장되어 플러그인을 설치하거나 별도의 애플리케이션을 다운로드할 필요가 없다.

WebRTC 1.0 표준 발표의 의미는 웹브라우저만으로 실시간 통신이 가능하게 되었다는 사실에서 그치지 않는다. WebRTC는 오랜 세월 동안 접근이 쉽지 않았던 실시간 미디어 통신 기술 산업을 개방형 기술로 변화시키는 중대한 역할을 했다. 코덱의 무료화, 미디어 엔진의 오픈소스화, 클라이언트의 대중화 등 WebRTC의 개발은 실시간 통신 기술 및 관련 산업에 획기적인 확장성을 가져다줄 수 있었던 것이다. WebRTC는 이러한 높은 확장성을 바탕으로 다양한 분야에서 활용되고 있다.

아울러, 이번 WebRTC 1.0 표준화 완료라는 획기적인 성과는 매우 시기적절하다고 할 수 있다. COVID-19 팬데믹에 직면하여 세계는 점점 더 가상으로 변했다. 정보 공유, 실시간 통신 및 엔터테인먼트 분야에서 웹이 더욱 중요해지고 있다. COVID-19 팬데믹 시대에 비하면 산업 분야의 성장이 빨라지면서 WebRTC의 활용 및 성장 가능성 또한 빠르게 증가할 것으로 예상된다. 화상 회의, 웨비나, 협업, 교육, 전문가 상담, 원격진료, 고객지원, 라이브 비디오 채팅, 방송, 엔터테인먼트 등의 광범위한 분야에서는 물론이고, 의료 및 국방과 같은 영역에서는 증강현실과 가상현실, 사물인터넷(IoT) 등과의 결합을 통한 새로운 분야로의 개척 또한 활발하게 이뤄질 것으로 기대하고 있다. 또한, 가상 학습 플랫폼으로 전환한 학교와 대학 영역, 라이브 스트리밍 및 대화형 방송을 사용하고 있는 클라우드 게임 및 소셜 네트워크 영역, 경기장 내 경험을 재현하려는 스포츠 영역 등의 다양한 분야에서 WebRTC는 중요한 역할을 할 것이다.

WebRTC 표준화 및 기술 개발에 기여한 주요 기업

앞서 언급했듯이 구글은 WebRTC를 오랜 기간 개발해왔고 표준화를 주도한 주요 기업 중 하나였으며, WebRTC가 실시간 통신을 위한 보편적이고 개방적인 표준으로 인정받는 데 크게 기여한 기업이다. 구글은 상용화 서비스 초기 WebRTC를 위해 오픈소스 구현 및 VP8 동영상 코덱과 같은 무료 사용을 지원하였고 이는 서비스가 발전하고 성장하는 생태계의 기반을 형성했다. 또한, Google Duo, Google Meet 및 Stadia를 포함한 수많은 구글 제품 및 서비스에서 WebRTC를 기본 프로토콜로 채택하였다. 에릭슨은 업계에서 WebRTC의 첫 번째 논의가 시작된 이래 W3C 및 IETF의 WebRTC 표준 작업 초창기 주요 기여자였다. 특히, 에릭슨은 여러 상용 솔루션에서 WebRTC를 성공적으로 사용했으며 3GPP 표준에 WebRTC의 핵심 요소를 포함하여 향후 대화형 통화로 더 풍부한 5G 음성 서비스를 가능하게 하는데 기여했다. 최근 React.js, React Native 및 Node.js, Docker 기반으로 엔터프라이즈 WebRTC 제품 개발을 진행 중이다. 시스코는 WebRTC 표준을 통해 Webex를 포함한 모든 웹 회의 응용 프로그램에 대한 손쉬운 배포를 활성화함으로써 전 세계적으로 연결하여 더 풍부하고 포괄적인 경험을 제공할 수 있다고 밝혔다. 모질라의 파이어폭스 웹브라우저는 초창기인 2013년부터 음성 영상 통화 표준인 WebRTC를 지원하였다. 또한, 그 당시에 이미 새로운 자바스크립트 모듈을 탑재해 웹 애플리케이션 실행 속도를 대폭 향상시켰으며, 구글 크롬 브라우저와의 호환성도 지원하기 시작했다. 그 당시 모질라는 자사 모바일 폰인 파이어폭스 폰을 개발했는데 WebRTC를 파이어폭스 운영체제를 중심으로 한 자사 모바일 폰 전략의 기반으로 삼기도 했다.

WebRTC 확장을 위한 향후 표준화 계획

WebRTC의 사용이 초기 핵심 설계를 넘어 웹브라우저 및 기타 에코 시스템(예; 네이티브 앱)에서 화상 회의 및 협업 시스템을 강화하기 위해 확장됨에 따라 이제 더 많은 기능과 더 많은 최적화가 필요하다.

IETF에서는 WebRTC를 확장하기 위한 작업이 이미 진행 중이다. WEBTRANS (WebTransport) WG는 다양한 전송 속성에 대한 추가 웹 지원을 구축하는 것을 목표로 한다. WebTransport 기술 역시 WebRTC 기술과 마찬가지로 웹 기반 실시간 양방향 통신을 위한 프로토콜이다. WebTransport는 TCP 기반의 HTTP/1.1 상에서 동작하던 IETF 표준기술 WebSocket(RFC 6455)을 UDP 기반의 HTTP/3.0과 QUIC 상에서 동작하도록 하는 최근 IETF에서 표준화가 시작된 신기술이라고 할 수 있다. WebTransport나 WebSocket을 사용하면 웹브라우저와 웹 서버 간의 전이중 통신, 즉 양방향 통신이 가능하다. WebRTC의 PeerConnection API를 통해 연결하면 두 웹브라우저 간에 전이중 통신이 가능하다는 측면에서, WebTransport와 WebRTC는 어느 정도 비슷한 개념의 프로토콜이라 할 수 있다. WISH(WebRTC Ingest Signaling over HTTPS) WG는 방송 도구와 실시간 미디어 방송 네트워크 간의 단방향 WebRTC 기반 시청각 세션을 지원하는 프로토콜 개발에 초점을 맞추고 있다. 핵심 WebRTC 표준을 완성하기 위해 수년 동안 수십 명의 IETF 및 W3C 참여자들의 엄청난 노력이 필요했다. 그 결과 매일 전 세계적으로 사람들을 연결하는 인터넷의 핵심 약속을 이행하는 매우 인기 있는 기술 제품군이 탄생했다. IETF 커뮤니티가 이러한 성공을 계속해서 구축함에 따라 미래가 어떻게 될지 보는 것은 흥미로운 것이다.

WebRTC의 사용 사례(Use Cases)는 [그림 4]와 같이 이미 온라인 교육, 사물인터넷, 헬스케어, 엔터테인먼트, 자율주행차 등의 분야에서 고려되어 왔으며[6], 보다 더 다양한 분야로의 확장을 위한 연구가 W3C에서 진행 중이다[7]. W3C 내의 WebRTC WG은 WebRTC의 미래를 계획하기 위해 WebRTC Next Version Use Cases 작업을 시작했다. W3C WebRTC WG은 모든 요구 사항과 우선순위를 이해하는 데 중점을 두고 기존 및 새로운 활용 사례를 연구하고 있다.

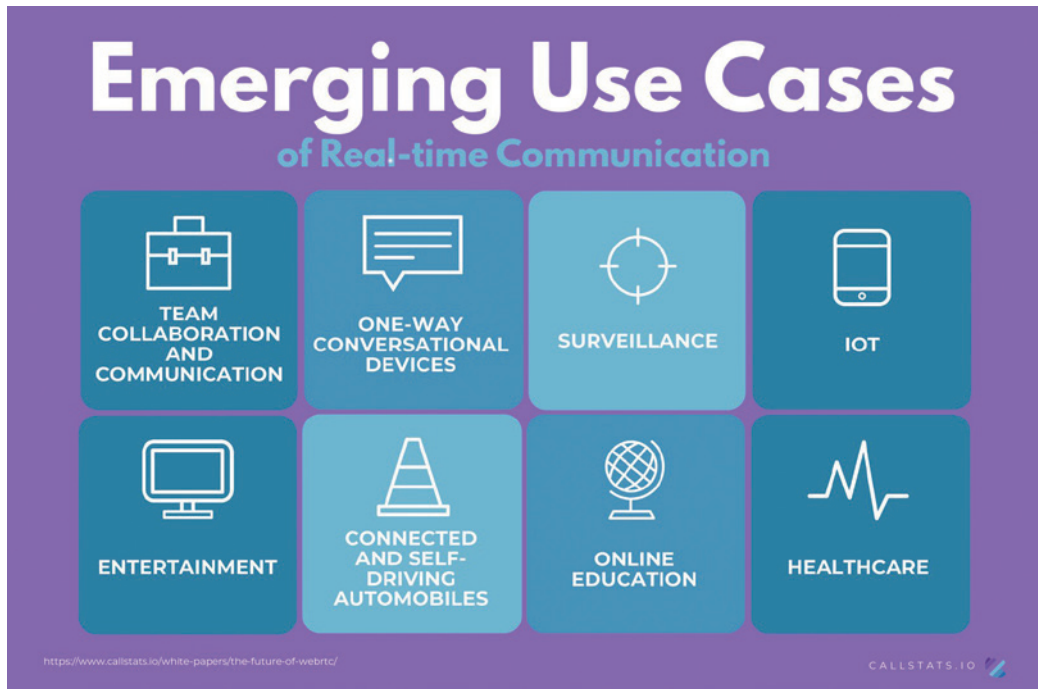


그림 4. WebRTC 사용 사례(Use Cases)

/ 출처 : The Future of WebRTC: Innovative Use Cases of Real-Time Audio and Video Communications, callstats.io, May 2018[6]

결론 및 시사점

WebRTC 1.0 표준의 발표는 전 세계 수십억 명의 사람들이 사용하는 회의 서비스의 다양한 P2P 서비스 개발에 이정표가 될 것으로 기대한다. 아울러, 웹브라우저에서 버튼 클릭만으로 화상 통화 및 화상 회의를 할 수 있다는 개념은 협업, 연결 및 생산성에 대한 거의 무한한 가능성을 제시할 것으로 기대하고 있다. 특히, COVID-19 팬데믹으로 인해 이 표준화 활용 가능성은 광범위한 분야에서 크게 증가할 것이다. 최근 들어, 라이브 미디어 커머스 서비스 기업들은 WebRTC 기반 플랫폼을 사용하여 라이브 스트리밍 인프라를 구축했으며 시청자와 스트리머 간의 저지연성을 특징점으로 내세우며 서비스하고 있다.

지난해부터 본격 시작된 5G 네트워크는 향후 몇 년간 적극적으로 상용화될 것으로 예상되며, 더 많은 숫자의 모바일 인터넷 단말이 5G 네트워크 서비스를 사용할 것으로 예상된다. 5G 네트워크의 저지연성, 빠른 속도, 충분한 대역폭으로 더욱 향상된 WebRTC 기반 모바일 비디오 서비스 및 실시간 스트리밍 서비스를 제공하는 환경이 갖춰질 것으로 기대한다. 📡

참고문헌

- [1] C. Jennings, H. Bostrom, J.-I. Bruaroey, WebRTC 1.0: Real-Time Communication Between Browsers, W3C Recommendation, 26 January 2021.
- [2] Mozilla, WebRTC 프로토콜 소개, MDN Web Docs, Mozilla, March 2021.
- [3] 웹에서의 실시간 비디오, 오디오, 데이터 통신 프로그래밍을 위한 W3C WebRTC 표준해설서, 사실표준화대응 역량강화시리즈, 한국정보통신기술협회, 2020.
- [4] 김평수, "IETF-W3C WebRTC 1.0 표준기술 개요와 팬데믹 시대에서의 의미", 주간기술동향 1993호, 정보통신기획평가원, April 2021.
- [5] M Meszaros, F. Trojahn, Definition and Analysis of WebRTC Performance Parameters as well as Conception and Realization of an End-to-End Audio Quality Monitoring Solution for WebRTC-Based "immmr" Call Scenarios, Hochschule für Telekommunikation Leipzig, December 2017.
- [6] The Future of WebRTC: Innovative Use Cases of Real-Time Audio and Video Communications, callstats.io, May 2018.
- [7] Bernard Aboba, WebRTC Next Version Use Cases, W3C Working Draft 16, March 2021.