

# WebRTC기술을 활용한 웹 기반 실시간 비디오 스트리밍 로봇

Deeply(박미희, 박세정, 이혜인)

세종대학교 지능기전공학부

[https://github.com/LEEHYEIN-098/21-1\\_Capstone\\_deeply](https://github.com/LEEHYEIN-098/21-1_Capstone_deeply)

## 요약

WebRTC 기술을 활용해 라즈베리파이4 기반 로봇을 원격으로 제어하며 실시간 비디오 스트리밍이 가능한 웹을 구축하였다. 또한 라즈베리파이에 연결된 웹캠의 영상을 획득하기 위해 UV4L을 사용하였다. 모든 과정은 WIFI 통신 아래 진행되었다.

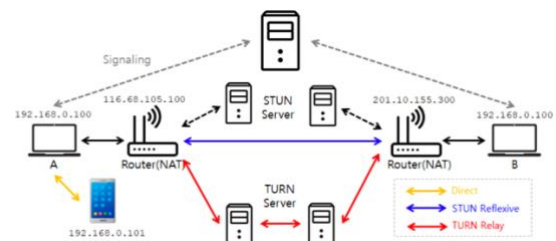
## 1. 서론

COVID-19 팬데믹으로 인해 재택근무와 원격 수업이 보편화되어 전세계적으로 비대면, 원격 관련 온라인 솔루션에 대한 의존도가 높아졌다. 유튜브와 넷플릭스, 모바일 게임 이용 등 산업 분야에서 환경이 급변하여 언택트 시스템의 니즈가 확대되었다.

‘사이언스 로보틱스’에 기고된 논문에 따르면 코로나 19 대응 과정 속 로봇 기술은 전 세계 제조업과 경제에 중요한 역할을 하고 있다. 해당 논문을 작성한 연구진들은 이 상황속에서 원격 제어 시스템에 대한 연구 필요성이 높아지고 있다는 점을 지목했다.<sup>1</sup> 로봇 원격 제어에 있어 가장 중요한 점은 실시간성이다. 본 팀은 최근 실시간 통신에 있어 우수한 기술로 주목받고 있는 WebRTC 기술을 활용하여 웹 기반 실시간 비디오 스트리밍 로봇을 구현하고자 한다.

## 2. 사용 기술

### 2.1 WEBRTC 스트리밍 프로토콜



(자료) Mozilla, WebRTC 프로토콜 소개, MDN Web Docs, Mozilla, Mar. 2021.

그림 1. 전체적인 WebRTC 동작 구성도

WebRTC는 HTML5와 Javascript API를 사용하여 웹상에서 플러그인이나 소프트웨어를 별도로 설치할 필요 없이 오디오, 비디오, 파일 등을 두 피어가 직접적으로 공유할 수 있도록 해준다. 또한 최근 2021년 1월 말 표준이 완료되었으며 “Can I use?”(<https://www.canIuse.com/>)에 따르면 2021년 3월 기준으로 마이크로소프트 엣지, 파이어폭스, 크롬, 사파리, 오페라, iOS 사파리, 오페라 모바일, 안드로이드 크롬, 삼성 인터넷 등 많은 브라우저에서 지원한다.<sup>2</sup>

WebRTC는 P2P 방식의 커뮤니케이션이나, 웹 브라우저는 외부 네트워크에서 접근 가능한 주소를 갖고 있지 않기 때문에 통신 설정 초기 단계에 통신을 원하는 디바이스들의 공인 IP를 알아내는 과정이 필요하다. 이 과정을 Signaling이라 한다. 하지만 NAT, DHCP, 방화벽과 같은 요인으로 공인 IP를 알아내기 어려운데, 그 해결책으로 NAT traversal 작업을 통해 P2P 연결에 필요한 정보를 얻을 수 있다.

<sup>1</sup> Erika Yoo, 해외 로봇 과학자들 "코로나19, 로봇 기술 연구 촉진", <http://www.irobotnews.com/news/articleView.html?idxno=20151>

<sup>2</sup> 김평수, IETF-W3C WebRTC 1.0 표준기술 개요와 팬데믹 시대에서의 의미, 주간기술동향, p.18

보통 NAT traversal 작업은 STUN(Session Traversal Utilities for NAT) server에 의해 이루어지며 라우터의 보안 정책에 따라 TURN(Traversal Using Relay NAT) server를 대안으로 이용할 수 있다. 하지만 TURN 방식은 데이터 전달 시 중개 서버를 거치게 되므로 엄연한 P2P 통신이 아니게 된다는 점을 유의해야 한다.

이 작업은 두 피어간 최적의 경로를 찾아주는 ICE(Interactive Connectivity Establishment) 프레임워크 위에서 이루어지며 결과적으로 획득한 정보를 ICE candidate이라 부른다.

미디어와 관련된 초기 세팅 정보를 기술하는 SDP(Session Description Protocol)로 어떠한 미디어에 대한 정보를 교환할지 제안하고 응답하는 과정을 마지막으로 WebRTC 통신을 위한 P2P 연결이 완료된다.

## 2.2 UV4L

UV4L은 실제 또는 가상 비디오 입출력 장치용 Video4Linux2 호환, 크로스 플랫폼, 유저 스페이스 드라이버 등의 모듈 모음으로 의도하여 개발되었다.<sup>3</sup> UV4L은 커스텀 웹 어플리케이션을 통해 양방향 데이터 채널, 오디오와 비디오 스트리밍과 같은 built-in 서비스를 제공한다. 또한 자체 맞춤형 어플리케이션을 구현하려는 개발자를 위해 RESTful API를 제공하고, HTTP/HTTPS 프로토콜을 모두 지원한다. UV4L과 라즈베리파이에서 실행되고 있는 application 사이에는 Unix Domain Socket이 만들어지며 데이터는 이 socket을 통해 이동한다.

## 3. 결과

### 3.1 시스템 구성도

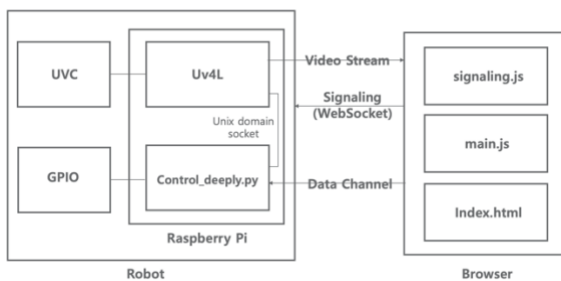


그림 2. 시스템 구성도

본 프로젝트의 구성도는 위와 같다. WebRTC의 Signaling 과정은 WebSocket을 이용하여 구현하였으며 이를 통해 연결된 두 명의 피어는 양방향 통신이 가능하다. WebRTC Video Channel로 비디오 스트림을 전송하고, WebRTC Data Channel로 로봇 명령어를 전달한다. 여기서 Video Channel은 UDP 기반이며 Data Channel은 SCTP 기반이다. 라즈베리파이 내의 UV4L과 파이썬 파일은 Unix Domain Socket이 이 둘을 브릿지처럼 연결하고 있어 데이터를 손쉽게 주고 받는 것이 가능하다. Data Channel로 명령어를 전달받은 로봇은 Python 함수를 통해 라즈베리파이의 GPIO를 제어한다.

### 3.2 로봇 구현

라즈베리파이와 모터 보드가 연결되어 있고, 이 모터 보드는 서보 모터, DC 모터와 연결되어 있다. 명령에 의해 로봇팔과 집게, 카메라 각도의 조절이 가능하다. 로봇의 제어는 Python을 통해 구현하였다.

### 3.3 Client 웹 구현

웹은 HTML, CSS를 사용하여 구축하였고, Javascript로 WebRTC 기술을 구현하였다. 웹 상의 비디오 스트리밍 영상은 얼굴 인식이 가능하며, 로봇 제어가 용이하도록 브라우저를 설계했다.



그림 3. Client의 웹 브라우저

### 3.3 구현 결과



그림 4. 지연 시간 측정 및 구현한 로봇

<sup>3</sup> UV4L 공식 홈페이지, [https://www.linux-](https://www.linux-projects.org/uv4l/tutorials/webrtc-data-channels/)

[projects.org/uv4l/tutorials/webrtc-data-channels/](https://www.linux-projects.org/uv4l/tutorials/webrtc-data-channels/)

위와 같은 과정을 거쳐 웹 기반 실시간 스트리밍 로봇을 구현하였다.

라즈베리파이와 웹으로 지연 실험을 해본 결과, 같거나 다른 네트워크에 속한 경우 모두 대략 0.3~0.4 sec 의 딜레이 시간을 확인할 수 있었으며 이는 여타 스트리밍 프로토콜들과 비교했을 때 실시간성이 매우 뛰어나다.

#### 4. 기대 효과

WebRTC 기술을 로봇에 접목시킨 기술은 앞으로 많은 산업 분야에서 핵심적으로 쓰일 수 있다. VR/ AR 분야에서 소프트웨어를 설치하지 않고 실시간으로 영상을 전송하며 로봇과 접목시켜 현실감있는 체험이 가능하기 때문에 메타버스(Metaverse)에서의 활용이 기대된다. 또한 급박한 상황에서 빠른 대응이 필요한 의료 업계나 위험한 상황이 자주 일어나는 공장, 군사 업계에서도 WebRTC 기술을 접목시켜 실시간 원격 로봇의 실시간성이 중요한 역할을 할 것이다. 더불어 피어 간 연결이 가능하다는 점에서 드론 제어에도 기술을 적용할 수 있을 것이다. 본 팀이 구현한 로봇은 Face Detection 이 가능하기 때문에 찾아가는 대화 서비스와 같이 특정 대상을 로봇이 찾아가 대화를 주고 받는 등 서비스화 시킬 수 있다. 뿐만 아니라 Computer Vision 을 응용하여 Object Detection 으로 로봇의 동작을 자동화하는 기능을 기대할 수 있다.

#### 감사의 글

3년 6개월가량의 학부과정을 거쳐 졸업 프로젝트를 진행하였습니다. 3월부터 약 4개월간 김형석 교수님의 지도하에 많은 배움을 얻었고 유익한 피드백으로 상당한 도움을 받았습니다. 이에 진심으로 감사드리며, 끝까지 포기하지 않고 좋은 결과를 내준 조원들에게도 감사의 인사를 전합니다. 또한 로봇과 관련된 기술에 도움을 주신 Sharif 에게도 감사의 인사를 드립니다.

#### 참고문헌

[1]Erika Yoo, 해외 로봇 과학자들 "코로나 19, 로봇 기술 연구 촉진", <http://www.irobotnews.com/news/articleView.html?idxno=20151>

[2]UV4L 공식 홈페이지, <https://www.linux-projects.org/uv4l/tutorials/webRTC-data-channels/>

[3]WebRTC 프로토콜 소개, [https://developer.mozilla.org/ko/docs/Web/API/WebRTC\\_API/Protocols](https://developer.mozilla.org/ko/docs/Web/API/WebRTC_API/Protocols)

[4]XIAO R GEEK 공식 홈페이지, <http://www.xiao-r.com/Study/catalog/cid/15>

[5]김평수, IETF-W3C WebRTC 1.0 표준기술 개요와 팬데믹 시대에서의 의미, 주간기술동향

[6]정종윤, WebRTC 는 어떻게 실시간으로 데이터를 교환할 수 있을까?, <https://wormwlrn.github.io/2021/01/24/Introducing-WebRTC.html>