

웹 실시간 통신을 활용한 반려동물 관리용 영상통화 안드로이드 앱 개발에 관한 연구*

권혁** · 문병철*** · 이진홍****

요약: 최근 반려동물의 시장 확대와 1인 가구가 증가되고 있는 상황에서 본 연구의 목적은 반려동물의 분리 불안증을 해소하기 위해 웹 실시간 통신(Web RTC)을 활용한 반려동물 관리용 영상통화 안드로이드 앱에 대한 시스템 설계 방안을 분석하는 것이다. 연구를 위해 소프트웨어 개발 내용, 안드로이드 앱 구성, AWS 서버, 키 페어 생성, 안드로이드 앱 소스코드에 대한 전체적인 시스템 설계 방안을 분석하였으며, 그 결과, Linux 기반의 임베디드 시스템과 안드로이드 스마트폰 간의 P2P 영상통화 서비스가 가능한 반려동물 관리용 영상통화 안드로이드 앱을 개발하였다. 본 연구를 통해 반려동물을 양육하는 가정에 보다 안전하고 편리하게 반려동물을 관리할 수 있는 수단을 제공하였으며, 앞으로 반려동물 서비스가 더욱 확산될 수 있도록 기술과 관련된 후속 연구를 진행하고자 한다.

핵심어: 웹 실시간 통신, 안드로이드 앱, 반려동물, 영상통화, 스마트폰

The Study on the Development of the Video Call Android App for Monitoring Companion Animals Using Web Real-time Communication

Hyuk Kwon, Byoungchul Moon, & Jinhong Lee

Abstract: The purpose of this study is to analyze system design methods for Android app, which is a video call for pet management using Web RTC (Web RTC), in order to solve separation anxiety of pets in the market for pets and single-person households. For research, we analyzed the overall system design method for software development, Android App configuration, AWS server, key pair generation, and Android App source code. As a result, we developed a video call Android App for pet management that enables P2P video call service between Linux-based embedded systems and Android smartphones. In this study, we provided a safer and more convenient means of managing pets for families raising pets, and we would like to conduct further technology-related research to ensure that services for pets will become more.

Key Words: Web RTC, Android App, Companion Animal, Video Call, Smart Phone

□ 접수일: 2019년 11월 26일, 수정일: 2019년 12월 19일, 게재확정일: 2019년 12월 20일

* 본 과제(결과물)는 교육부와 한국연구재단의 재원으로 지원을 받아 수행된 사회맞춤형 산학협력 선도대학(LINC+) 육성사업의 연구결과임.

** 주저자, 건국대학교 문헌정보학과 조교수

(First Author, Assistant Professor, KonKuk Univ., Email: k21720222@kku.ac.kr)

*** 교신저자, 엔티아이 대표

(Corresponding Author, CEO, New Think & Innovation, Email: mbc9596@gmail.com)

**** 공동저자, 건국대학교 경찰학과 조교수

(Co-Author, Assistant Professor, KonKuk Univ., Email: ljhtough@kku.ac.kr)

I. 서론

국내 반려동물의 시장 확대와 1인가구의 증가(2015년 27.2%, 2025년 31.9%)에 따른 반려동물 케어 기기 및 서비스 시장이 확대되고 있어 반려동물에 대한 관심도가 증가하고 있다. 특히 60대 이상에서 반려동물을 가족으로 생각하는 경향이 높은 것으로 나타났다. 특히, 1인가구가 증가하는 현 상황에서 외출 시 집안에 반려동물이 혼자 있는 비중이 높은 것으로 나타났고, 평균 5시간 30분 정도 집에 혼자 있는 것으로 분석되었다(황원경·정귀수·김도연, 2018).

많은 사람들이 여행, 직장 생활로 인해 반려동물 양육을 어려워하고 있으며, 이를 해소하고자 권대완·박동원(2016)은 반려동물의 체중 관리를 위해 사용자가 직접 데이터를 입력 및 공유할 수 있는 안드로이드 앱을 개발하였고, 이에 일부 연구(임지수 외, 2018)에서는 반려동물의 건강정보를 직접 입력하여 언제든지 정보를 공유할 수 있도록 안드로이드 전용 앱인 챗봇을 개발하였다. 그러나 실제 반려동물을 혼자 두고 외출하게 되면 반려동물의 분리 불안증 및 기타 부작용이 발생 함에도 현재 적절한 기기 및 서비스가 부족한 실정이며, 반려동물의 분리 불안증을 적절히 해소할 수 있는 제품은 거의 전무한 상황이다.

한편, 웹 비표준기술인 액티브 엑스(Active-X)가 보안상의 문제로 퇴출되어, 등장한 웹 실시간 통신(Web RTC)이 Web Assembly 1.0, Web RTC 1.0 등 표준제정임박의 제정(안)이 정보화 제정 개정법안에 상정되면서 웹 실시간 통신(Web RTC)을 이용한 활용 사례에 관한 중요도가 높아지고 있는 실정이다(류지웅, 2019. 11. 1. 검색).

상기에서 제기한 반려동물의 분리 불안증에 대한 문제를 적절히 해소하고자 집안에 설치된 스마트 TV를 활용하여 외부 스마트기기 및 기타 기기와의 양방향 영상통화 장치 및 서비스를 개발함으로써 반려동물의 분리 불안증을 해소하고, 분리 불안증으로 인한 부작용과 의료비 지출 등을 저감하도록 하여, 반려동물을 양육하는 가정에 보다 안전하고 편리하게 반려동물을 보호 및 관리하고자 안드로이드상의 P2P 영상 통화 Caller App을 개발하는 것이 본 연구의 목적이다. 기존의 영상 통화 수단은 스마트폰-스마트폰, PC-PC, PC-스마트폰 등으로 구성되어 운영되고 있음으로 상대방이 통신기기를 사용할 수 없는 상황에서의 영상통화 및 의사소통의 수단이 존재하지 않는 점과 현재 IP-Camera 등과 같은 CCTV의 경우 단방향 영상 수신과 양방향 음성 통화는 가능하지만, 양방향 영상통화로 구성되어 있는 것은 없는 점, 특히 기존의 TV 및 Monitor를 활용하는 기술에 대한 부분이 확인되지 않고 있는 실정에서 웹 실시간 통신(Web RTC)을 활용한 영상통화 안드로이드 App 개발의 기술

활용성에 대한 타당성 연구를 위해 소프트웨어 개발 내용, 안드로이드 앱 구성, AWS 서버, 키 페어 생성, 안드로이드 앱 소스 코드에 대한 전체적인 시스템 설계 방안을 알아보하고자 한다. 이번 연구를 통해 라즈베리 파이로 대변되는 Linux 기반의 임베디드 시스템과 안드로이드 스마트폰 간의 P2P 영상통화 서비스를 가능한 개발 결과물 사례를 제공하여 보다 가격적으로 장점이 있는 양산화 된 제품과 영상통화 기반의 많은 반려동물 서비스가 제작 및 확대되길 기대한다.

II. 연구의 배경

1. 이론적 배경

1) 웹 실시간 통신(WebRTC) 및 안드로이드 App 동향

웹 실시간 통신(WebRTC)는 저장되어 있는 영상을 피동적으로 받지 않고, 이용자가 자발적으로 웹상에서 API를 활용하여 파일을 공유하거나 실시간 음성 통화를 할 수 있는 기능을 하고자 공개된 통신시스템으로 월드 와이드 웹 컨소시엄(W3C)에서 제정하였다 (Venkataraman & Chatterjee, 2011).

최근 웹 실시간 통신(WebRTC)의 활용 사례 동향을 살펴보면 PC와 스마트폰을 연동하여 영상과 음성을 전송하는 WebRTC 기반 MCU(multipoint control unit) 서버 기술이 활용되고 있으며, 실시간 쌍방향 수업이 가능한 플랫폼을 개발하는 (주)구루미가 관련 기술을 발표한 바 있다(김현아, 2019. 11. 1. 검색).

관련한 적정 기술로 RTMP, HLS, RTSP 등의 범용 프로토콜이 사용되고 있는데, RTSP는 통신 시스템 상에서 스트리밍 미디어 서버를 제어하고자 설계된 네트워크 제어 프로토콜이다. 미디어 서버의 클라이언트는 서버에서 클라이언트(video on demand) 또는 클라이언트에서 서버(voice recording)로 단방향 스트리밍 하며, 재생, 녹화, 정지 등을 실행할 수 있다. 그리고 HLS(HTTP live streaming)은 Apple Inc에서 QuickTime, Safari, OS X 및 iOS 소프트웨어로 구현 한 통신 시스템이다. MPEG-DASHin과 유사하여 전체 스트림을 일련의 작은 HTTP 기반 파일 다운로드로 분할하여 작동할 수 있는 특징이 있다. 또한 RTMP는 처음에는 Macromedia가 인터넷을 통해 오디오 플레이어, 비디오 및 데이터

를 플래시 플레이어와 서버 간에 스트리밍을 하고자 독자적으로 개발 한 프로토콜이다. Macromedia는 현재 Adobe에서 소유하고 있으며, 이 프로토콜은 공용 프로토콜 버전을 출시하였다(Wikipedia, 2019. 11. 1. 검색).

관련 기술을 바탕으로 웹 기반 강의를 많은 대학 및 무료공개강좌를 통해 활용되고 있다 (Panagiotaropoulou et al., 2014). 그러나, 현재 활용되고 있는 기술들은 사용자가 능동적으로 활용하는 형태가 아니며, 주로 단방향의 영상 및 음성의 스트리밍을 위한 것으로 양방향 교육의 한계점이 존재하는 것도 사실이다.

이처럼 웹 App에서의 소통을 위해 각 기능들을 연결할 수 있도록 Device API 기술이 개발되어 원격 교육, 여러 화상회의 시스템에 활용되고 있으며, 관련 안드로이드 App도 활발하게 개발되어 사용되고 있다(Nurminen et al., 2013).

앞선 여러 문제들을 해결하기 위해 문병철(2019)은 TV 스마트폰 통신망 통해 연결하여 영상 통화를 구현할 수 있도록 TV 시청 제어 장치를 연구하였고, 본 연구를 통해 HDMI 채널을 2채널로 확대하여 다양한 멀티미디어 서비스를 구현할 뿐만 아니라 별도의 통신수단의 사용이 불편한 노약자, 유아, 아동들의 위급 상황시 간단한 버튼 동작으로 TV-스마트 폰 간 영상 통화를 지원할 수 있도록 기술개발을 하였다.

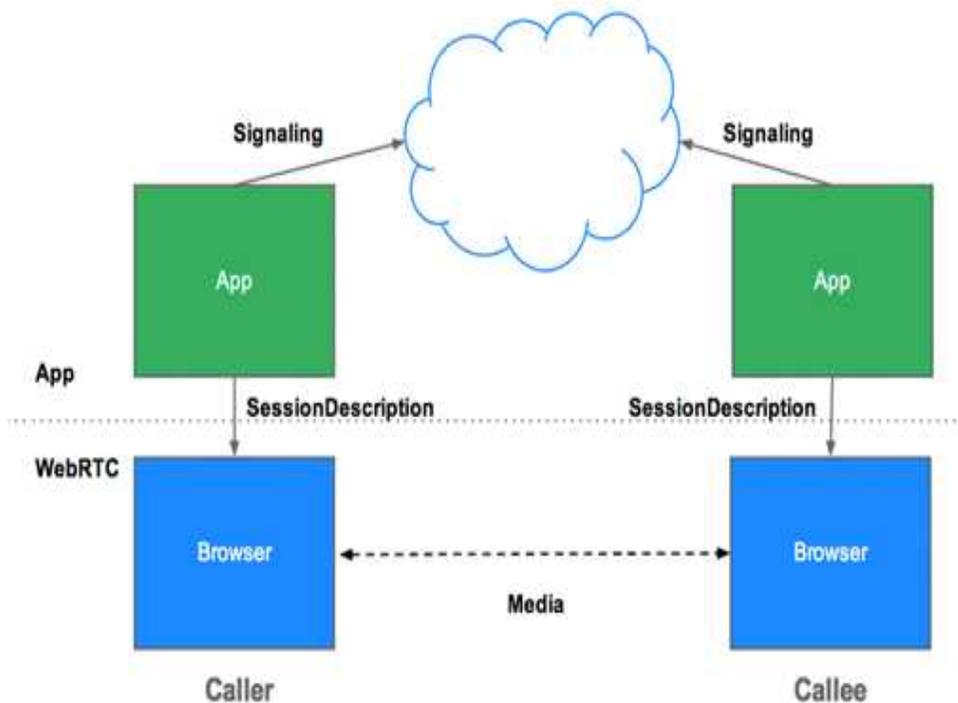
한편 기존의 영상 통화 수단은 스마트폰-스마트폰, PC-PC, PC-스마트폰 등으로 구성되어 운영되고 있으므로 상대방이 통신기기를 사용할 수 없는 상황에서의 영상통화 및 의사소통의 수단이 존재하지 않는 점과 현재 IP-Camera 등과 같은 CCTV의 경우 단방향 영상 수신과 양방향 음성 통화는 가능하지만, 양방향 영상통화로 구성되어 있는 것은 없는 점, 특히 기존의 TV 및 monitor를 즉시 활용할 수 있는 방안이 확인되지 않고 있는 실정에서 웹 실시간 통신(Web RTC)을 활용한 영상통화 안드로이드 App 개발의 기술 활용성에 대한 연구를 통해 관련 기술 및 시스템 설계 방법을 제안하고자 한다.

Ⅲ. 기술 개요 및 설계 목표

1. 시스템 기술 개요

1) WebRTC

Embedded 시스템은 시스템상의 리소스가 제한되어 있으므로 embedded 상의 웹브라우저에서 WebRTC API를 호출하여 영상 및 음성 통화를 실시간으로 제공하기에는 많은 어려움이 있다. 아래 <그림 1>은 WebRTC 통신시의 신호 흐름도로서 caller와 callee 간에 signaling을 통하여 서로를 찾고 연결(connection)이 맺어지고 난 이후에는 P2P 통신을 통하여 서로 영상과 음성의 media가 상호간에 전달이 된다. 이러한 앱(APP)은 모두 Web 기반에서 동작하고 있으나 저가의 임베디드 기기에서 웹브라우저 기반으로 WebRTC를 실시간으로 운영하기는 거의 불가능하다.



<그림 1> Webrtc 통화시 신호 연결도

따라서, 별도의 WebRTC용에 맞도록 임베디드 기기에서 운영이 가능한 제한적인 기능의 전용서버를 개발하게 되는데 여기서는 Linux용으로 개발된 UV4L 서버를 구성하여 웹브라우저 없이 WebRTC의 수신단(CALLEE)을 구성하고 스마트폰의 안드로이드 앱이 송신단이 된다(caller). 양방향 통화가 가능하기 위해 embedded 기기에는 전체 시스템의 제어를 위

한 별도의 SW가 있으며, 이 SW가 안드로이드 앱단과 mqtt 기반의 양방향 보조 통신을 함으로써 embedded쪽에서도 안드로이드 앱에게 전화통화가 가능하고, 언제든지 상호간에 양방향 통화를 요청할 수 있게 되는 것이다.

2) 안드로이드 App 및 임베디드 SW

안드로이드 앱은 임베디드 SW와의 보조통신을 위한 mqtt 통신 부분과 안드로이드 내부의 웹뷰상에서 WebRTC 기반의 HTTPS 통신을 행하게 된다. 또한, 임베디드 시스템에서는 WebRtc 통신을 위한 UV4L WebRTC서버와 mqtt 보조통신을 위한 소프트웨어가 개발된다. 이 소프트웨어를 통하여 임베디드 쪽에서 안드로이드 폰으로 통화 요청이 가능하게 된다.

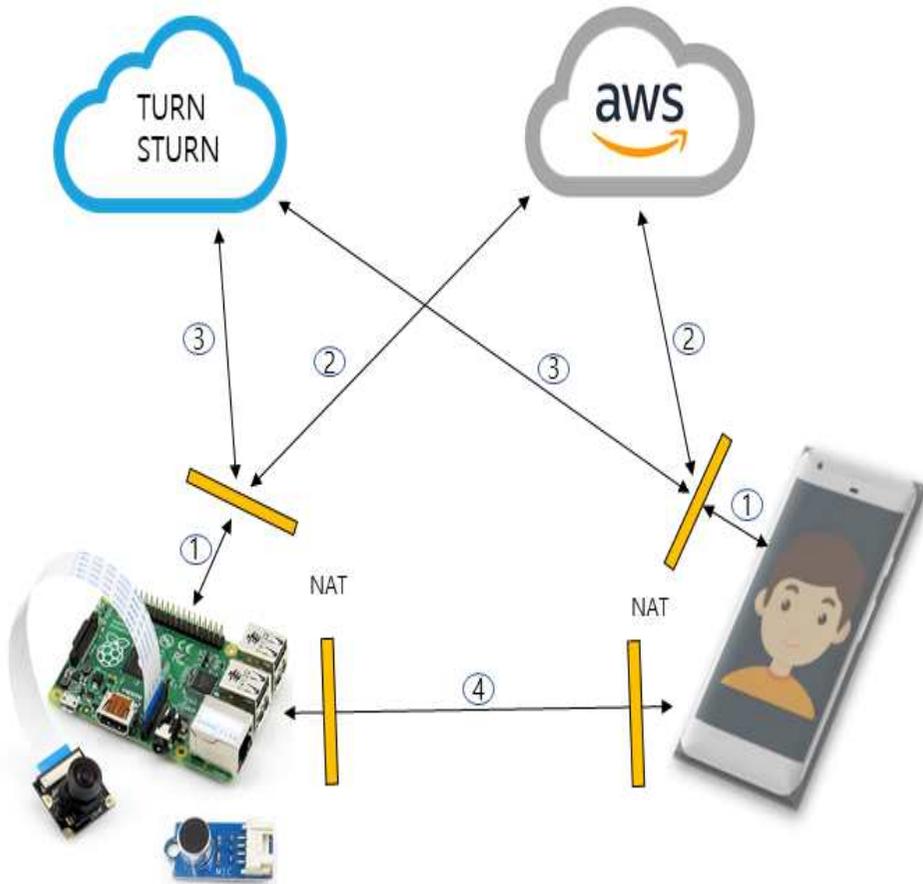
2. 설계목표

설계 목표는 첫째, 저가의 임베디드 시스템에서 WebRTC 통신이 가능하도록 비디오 카메라, 마이크, 스피커 또는 TV로의 HDMI 출력을 가지는 UV4L WebRTC 서버를 개발하고, 또한 양방향 통신이 가능하도록 임베디드 시스템에서의 간단한 통화요청 버튼으로 WebRTC 통화가 가능하도록 보조적인 통신을 할 수 있는 mqtt 소프트웨어를 개발하는 것이며, 둘째, 안드로이드 앱 내부에 mqtt 보조통신 부분과 이와는 별도로 웹브라우저 상의 웹뷰(WebView) 상에서 https 보안통신 기반의 WebRTC가 가능한 앱을 개발하고자 한다.

IV. 시스템 구현 결과

1. 시스템 구성도

개발한 시스템의 전체 구성도는 다음 <그림 2>와 같다.

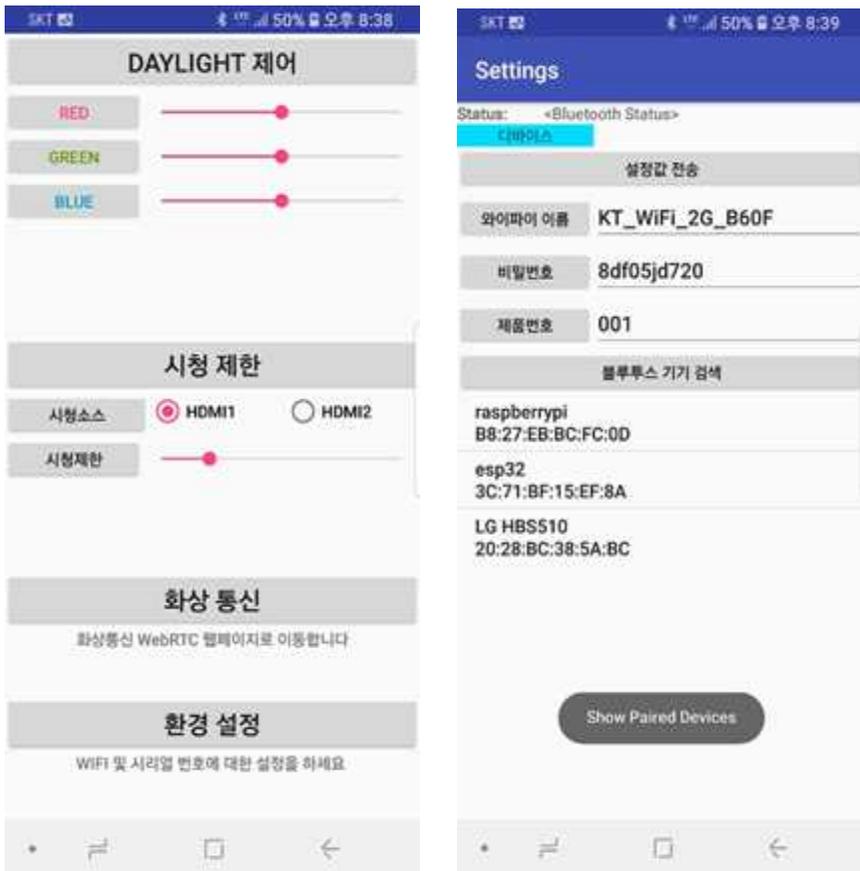


<그림 2> 시스템 전체 구성도

먼저 임베디드 시스템에는 카메라와 마이크가 있으며 HDMI 출력으로 TV에 연결되어 영상을 보고 스피커로 음성을 들을 수 있다. 안드로이드 폰이나 임베디드 시스템에서 ①의 신호는 가정의 무선이나 유선 AP의 NAT 방화벽을 통하여 ②의 신호로 바뀌어 AWS 상의 랑데부 서버에 해당이 될 수 있는 mqtt 서버를 통하여 서로의 상태와 통화 요청에 대한 정보를 주고받는 것이 특징이다. 또한 어느 한 쪽이 영상통화를 요청할 경우, ③의 신호로써 TURN 및 STUN 서버를 통하여 signaling을 주고받아 임베디드 기기와 안드로이드 폰이 ④의 직접 연결(connection)을 맺게 된다. 이후에는 P2P 형태로 영상과 음성 데이터를 직접 주고받는다.

2. 안드로이드 앱 구성

현재 기본 구성이 완료된 안드로이드 App는 다음 <그림 3>와 같다.



<그림 3> 안드로이드 App 구성도

3. 소프트웨어의 구성

임베디드 기기의 소프트웨어는 크게 2가지로 구성된다. 먼저 UV4L WebRTC 서버는 마이크의 음성과 비디오 카메라의 영상을 입력받아서 안드로이드 폰으로 전송하고, 안드로이드 폰으로부터 수신된 영상을 HDMI로 출력하고, 수신된 음성은 HDMI의 오디오부를 통하

여 출력한다. 또한, Node JS 기반의 보조통신 소프트웨어는 임베디드 기기의 상태 및 제어 신호를 안드로이드 폰과 주고 받음으로써 PC기반의 WebRTC 통신이나 스마트폰 간의 WebRTC 통신과 차이가 없도록 기능을 구현하였다. 또한 안드로이드 폰의 앱은 크기는 웹 앱 기반의 Chrome browser를 사용하여 WebRTC를 구현하였으며, 안드로이드 native 부분의 background상에서 mqtt 통신을 구현하였다. 또한, AWS에 실시간 통신이 가능한 mqtt 서버를 구성하여 전체적인 flow 제어를 담당하도록 하였다.

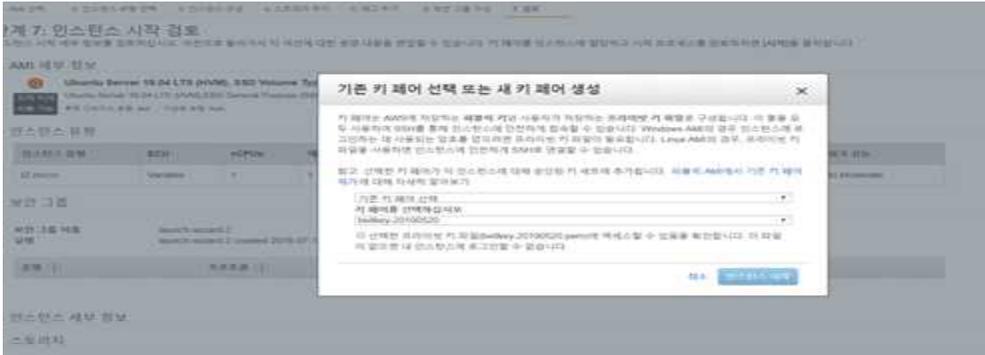
1) AWS서버

먼저 AWS 계정으로 들어가서 새로운 instance를 생성하였고, 다음 <그림 4>과 같다.



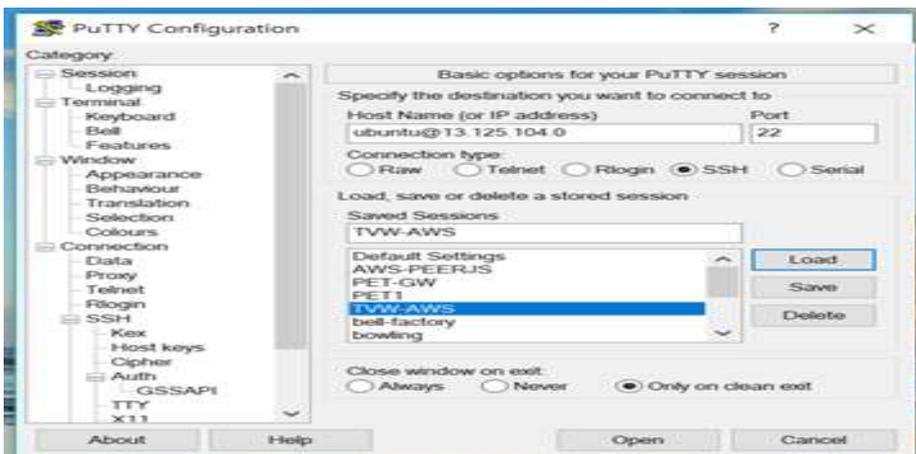
<그림 4> Instance를 생성도

다음으로 우분투 서버 18.04나 16.04를 선택하고, 적절한 keypair를 신규 생성하였고, 기존 keypair를 선택하였다. 다음, 키페어를 다운로드하여 PC에 저장한 뒤, 생성된 서버 인스턴스(instance)에서 맨 우측의 보안그룹을 눌러서 사용할 포트를 세팅해야 하며, 키페어 생성도는 <그림 5>와 같다.



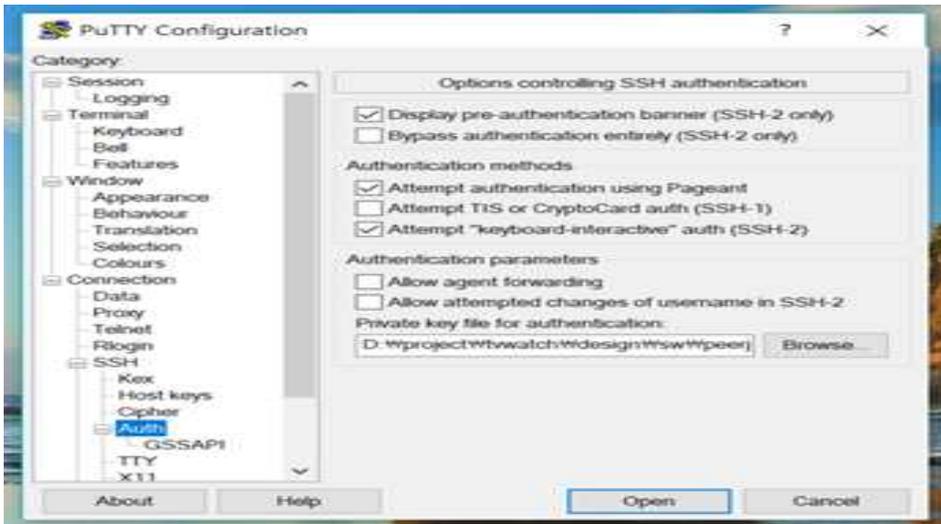
<그림 5> 키 페어 생성도

다음으로 80, 22, 1883, 1884, 443 포트를 정의한다. 여기서 80포트는 웹서버용이며, 안드로이드 웹뷰에서 사용한다. 22포트는 디버깅용으로 서버에 프로그램을 올리고 테스트 할 때 사용한다. 1883, 1884는 ESP8266 디바이스나 서버의 WTHAP에서 MQTT 서비스에 접속 할 때 사용하며, 3033 포트는 안드로이드 웹뷰에서 MQTT에 접속할 때 사용한다. 443 포트는 추후 https가 가능한 웹서버를 위하여 미리 설정한 부분으로 아직 사용하지는 않는 것이 특징이다. 이상의 과정을 통하여 AWS상에 기본적인 Linux 시스템이 만들어졌다. 다음으로 MQTT를 세팅해야 하며, 다음 <그림 6>과 같다. 이를 위해서는 라즈베리원격 Access와 유사하게 원격 접속이 필요하다. 여기서는 putty를 사용한다. 먼저 putty를 다운로드하여 윈도우 pc에 설치하였으며, 다운로드 가능한 주소는 <<https://www.putty.org/>>이다.



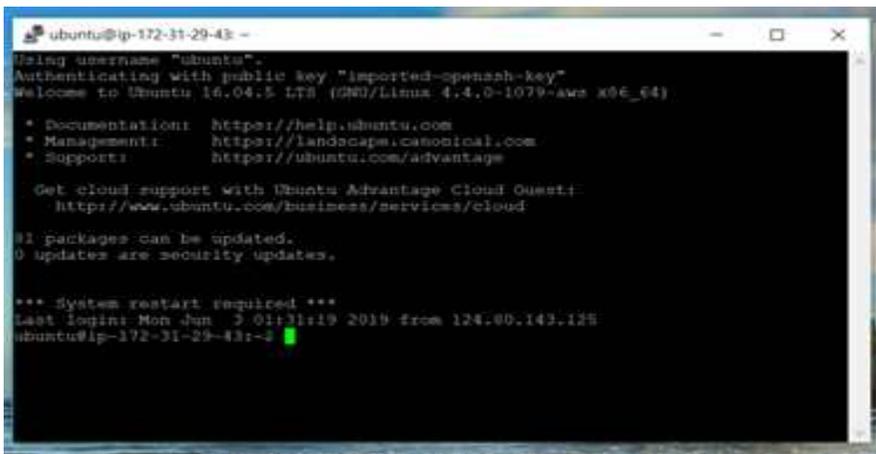
<그림 6> MQTT를 세팅 생성도

다음 <그림 7> 과 같이 맨 아래의 private key file의 browse를 눌러 이전에 저장한 인증서를 선택하고, 다시 위 화면으로 돌아가서 설정의 이름을 정하고 저장한 후에 open을 눌러서 들어간다.



<그림 7> MQTT를 세팅 생성도II

이제 AWS상의 Linux 서버에 접속하였고 directory 위치는 ubuntu이며, 다음 <그림 8> 과 같다.



<그림 8> Linux 서버 생성도

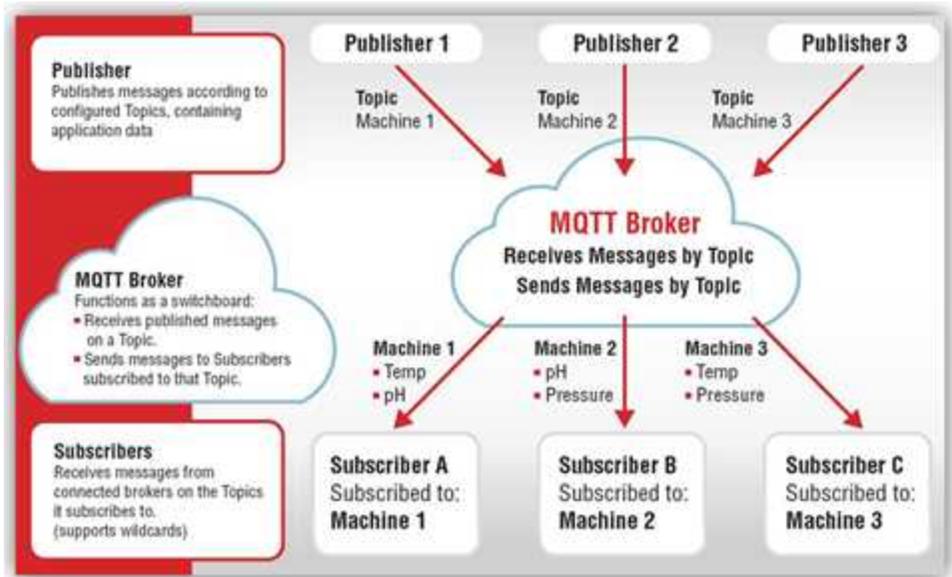
다음으로 아래와 같이 세팅한다.

```
$ sudo apt-get install mosquitto mosquitto-clients  
mosquitto broker and the publish / subscribe clients.  
2가지를 install한다.
```

```
$ sudo nano /etc/mosquitto/conf.d/default.conf  
listener 1883  
listener 1884  
listener 3033  
protocol websockets
```

```
$ sudo systemctl restart mosquito (Mosquitto를 restart)한다.
```

끝으로 MQTT 프로토콜은 HTTP에 비해 수십-수백배 빠른 속도로 양방향으로 응답을 주고 받을 수 있는 것이 장점이며 IOT 디바이스에 특화된 PROTOCOL이다. 이러한 mqtt 통신은 MQTT broker가(여기서는 mosquito) AWS 서버에 설치되며, 안드로이드 앱에는 웹 앱 기반으로 javascript mqtt client가 실행이 되고, 서버에서는 bnaster.js 내부적으로 nodejs 기반의 mqtt clienr가 실행이 된다. 이러한 client는 각자 메시지를 mqtt publishing(발행 또는 보내고)과 mqtt subscring(수신 또는 접수)하게 되는데, 모든 메시지는 MQTT broker에게 접수되어 broker가 메시지를 각각의 client에게 발송한다. 전술한 바와 같이 AWS Ec2를 생성하였고 time zone은 seoul로 하였다 mqtt에는 1883, 3033을 사용하였으며, 최종 구성도는 다음 <그림 9>와 같다.



<그림 9> MQTT Broker 구성도

4. 안드로이드 앱

최종적으로 개발된 안드로이드 앱의 소스 코드는 아래와 같다.

```
<RemoteActivity.java>
```

```
Remote Activity
```

```
package com.example.nti.ntiwebtrc;
```

```
import android.content.ActivityNotFoundException;
```

```
import android.content.Intent;
```

```
import android.content.SharedPreferences;
```

```
import android.net.Uri;
```

```
import android.os.Bundle;
```

```
import android.app.Activity;
```

```
import android.provider.SyncStateContract;
```

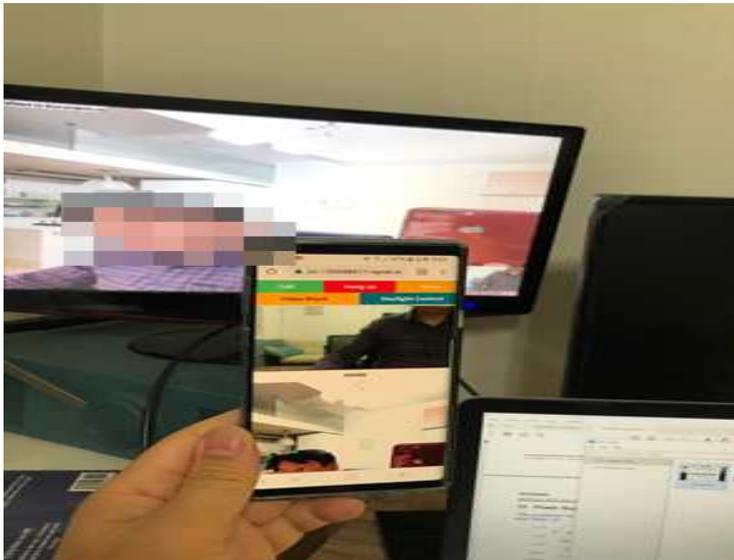
```
import android.util.Log;
```

```
import android.view.View;
import android.widget.Button;
import android.widget.RadioButton;
import android.widget.SeekBar;
import android.widget.Toast;

import org.eclipse.paho.android.service.MqttAndroidClient;
import org.eclipse.paho.client.mqttv3.IMqttDeliveryToken;
import org.eclipse.paho.client.mqttv3.MqttCallbackExtended;
import org.eclipse.paho.client.mqttv3.MqttException;
import org.eclipse.paho.client.mqttv3.MqttMessage;

import helpers.MqttHelper;
```

하기 <그림 10>은 개발된 안드로이드 앱을 이용하여 WebRTC 통신을 하는 상황을 나타낸 것이다. 모니터 화면과 안드로이드 화면에 각자의 모습과 상대방의 모습이 나타나며 영상과 음성 대화가 가능하다.



<그림 10> WebRTC 통화의 실제 사례

IV. 결론

이번 연구는 상기에서 제기한 반려동물의 분리 불안증에 대한 문제를 적절히 해소하고자 집안에 설치된 스마트 TV를 활용하여 외부 스마트기기 및 기타 기기와의 양방향 영상통화 장치 및 서비스를 개발함으로써 반려동물의 분리 불안증을 해소하고, 분리 불안증으로 인한 부작용과 의료비 지출 등을 저감하도록 하여, 반려동물을 양육하는 가정에 보다 안전하고 편리하게 반려동물을 관리할 수 있는 수단을 제공하기 위해 안드로이드상의 P2P 영상 통화 caller App을 개발하고자 하였다. 기존의 영상 통화 수단이 스마트폰-스마트폰, PC-PC, PC-스마트폰 등으로 구성되어 운영되고 있음으로 상대방이 통신기기를 사용할 수 없는 상황에서의 영상통화 및 의사소통의 수단이 존재하지 않는 점과 현재 IP-Camera 등과 같은 CCTV의 경우 단방향 영상 수신과 양방향 음성 통화는 가능하지만, 양방향 영상통화로 구성되어 있는 것은 없는 점, 특히 기존의 TV 및 monitor를 즉시 활용할 수 있는 방안에 대해 확인되지 않고 있는 상황에서 웹 실시간 통신(Web RTC)을 활용한 영상통화 안드로이드 App 개발의 기술 활용성에 대한 타당성을 검증한 것이 중요하다고 하겠다. 이번 연구의 한계점 및 향후 연구 방향으로는 본 연구를 통해 가격 경쟁력이 있는 반려동물 관리용 영상 통화 안드로이드 앱을 만들 수 있는 제반을 만들었으나 예산상의 문제로 양산에 가깝도록 소프트웨어 및 시스템을 고도화 하는 데는 어려움이 있었다. 후속 연구에서는 보다 많은 예산이 확보하고, 기능을 좀 더 보완하여 양산화할 수 있도록 노력하고자 하며, 향후에는 더 많은 사람들이 편리하게 활용할 수 있는 서비스를 개발하고자 한다.

【참고문헌】

- 권대완·박동원(2016), “반려견 관리를 위한 앱의 설계 및 구현”, 『한국융합학회논문지』, 7(2): 7-12.
- 문병철(2019), “양방향 영상 통화가 가능한 TV 시청 제어 장치”, 『등록특허공보』, 세종: 대한민국특허청.
- 임지수·김다영·조수민·유견아(2018), “챗봇과 규칙기반 전문가시스템을 이용한 애견 건강관리 시스템의 개발”, 『한국디지털콘텐츠학회논문지』, 19:(11): 2059-2066.
- 황원경·정귀수·김도연(2018), 『반려동물 연관 산업 현황과 양육실태』, 서울: KB 금융지주 경영연구소.
- Nurminen, J. K., Meyn, A. J. R., Jalonen, E., Raivio, Y., & R. Garcia Marrero (2013), “P2P media streaming with HTML5 and WebRTC”, 『IEEE Conference on Computer Communications Workshops』, 63-64, Turin: Italy.
- Panagiotaropoulou, S., Basat, A., Piccioli, V., Hauser, B., & Son Tung Bui(2014), 『E-Learning market trends & forecast 2014-2016』, Docebo.
- Venkataraman M. & M. Chatterjee(2011), “Effects of internet path selection on video-QoE”, 『in Proc. of ACM Multimedia Systems Conference』, 45-56.
- 김현아(2019), “구루미, ‘W3C HTML5 Conference 2015 in Seoul’에서 WebRTC 기술 발표”, <<https://www.edaily.co.kr/news/read?newsId=01377606609597864&mediaCodeNo=257>>(2019. 11. 1. 검색)>.
- 류지웅(2019), “국내 최대규모 웹기술 행사 ‘HTML5 Conference 2019 & Web Solution 전시회’ 개최”, <<http://www.digitaltoday.co.kr/news/articleView.html?idxno=215763>>(2019. 11. 1. 검색)>.
- Wikipedia(2019), <<https://en.wikipedia.org/wiki/>>(2019. 11. 1. 검색)>.