

A Disembarking Notification System in Public City Buses using Smart Device and High Frequency

Myoungbeom Chung*

*Assistant Professor, Dept. of Faidea, Sungkyul University, Anyang, Korea

[Abstract]

Recently, people have come to enjoy using movies and music applications, such as YouTube, Netflix, and Watcha Play, on their smart devices while riding public transportation. However, they sometimes miss their destination stations because of these activities. Therefore, in this paper, we propose a disembarking notification system for public city buses using smart devices and high frequencies generated via the buses' speakers. The high frequencies are generated by the public buses' station information broadcasters. Smart devices then analyze the high-frequency signals with their inner microphones, and the proposed application displays a disembarking notification to the user when the user's destination or stop-over station's signal is the same. To evaluate this system's performance, we tested 1,000 real-time disembarking notifications, and the test results showed 98.9% accuracy. Moreover, we compared these results to those using only OpenAPI, and our proposed system featured far better outcomes. Thus, this proposed notification system can prove a useful technology for many people who often use public city buses, as it can notify specific users of their destination stations. Furthermore, this system will become innovative technology for global public transportation by informing users of their desired stations using speakers.

▶ **Key words:** Public city bus, Disembarking notification, Audible frequency, High frequency

[요 약]

최근 대중교통을 이용하면서 스마트 기기를 이용하여 유튜브, 넷플릭스, 왓챠 등과 같은 동영상이나 음악을 즐기는 사람들이 많아졌다. 그러나 이러한 현상으로 사람들은 자신이 하차해야 할 곳을 지나쳐 내리는 경우가 빈번히 발생한다. 따라서 본 논문에서 우리는 버스 내부의 스피커에서 발생하는 고주파 신호를 이용한 스마트 기기 기반 시내 공공 버스 하차 알림 서비스 시스템을 구축하였다. 고주파 신호는 버스의 정류장 안내 방송과 함께 발생하며, 스마트 기기는 내장 마이크를 통해 입력되는 신호를 분석하여 사용자가 설정한 하차 역에 도달할 경우 스마트 기기에서 하차 알림을 발생한다. 제안 애플리케이션과 서비스 시스템의 성능 검증을 위해 총 1,000 번의 실시간 알림을 측정하였으며, 그 결과 제안 시스템은 98.9%의 정확도를 나타냈으며, 기존 Open API만을 이용한 알림 서비스에 비해 우수한 성능을 보였다. 따라서 제안 시스템은 시내 공공 버스를 이용하는 사람들에게 목적지 도착 시 하차 지점을 정확히 알려줄 수 있는 유용한 기술이 될 것이다. 또한, 이 기술은 스피커로 도착지를 안내 방송하는 세계의 다양한 대중교통 서비스에 활용할 수 있는 우수한 기술이 될 것이다.

▶ **주제어:** 공공 시내 버스, 하차 알림, 가청 주파수, 고주파

-
- First Author: Myoungbeom Chung, Corresponding Author: Myoungbeom Chung
 - Myoungbeom Chung (nzine@sungkyul.ac.kr), Dept. of Faidea, Sungkyul University
 - Received: 2020. 07. 30, Revised: 2020. 08. 20, Accepted: 2020. 08. 20.

I. Introduction

최근 스마트 기기의 보급이 급격히 증가하고, 4G LTE(Long Term Evolution) 통신 이후 5G 통신으로 네트워크 속도가 매우 향상됨에 따라 다양한 장소에서 스마트 기기를 이용한 영화, 음악 콘텐츠를 즐기는 사람들이 늘어났다. 카페, 식당, 패스트푸드점에서도 많은 사람들이 인터넷 강의 시청, 유튜브, 넷플릭스 등에서의 영화, 드라마 콘텐츠 감상, 스트리밍 음악 청취 등을 스마트 기기 하나로 손쉽게 할 수 있으며, 이러한 모습들은 회사 출·퇴근, 학교 등·하교를 하는 사람들이 대중교통을 이용할 때에도 많이 볼 수 있다. 그러나 지하철, 버스 등과 같은 대중교통을 이용하는 사람들은 즐기고 있는 콘텐츠에 집중을 하여 하차할 정류장을 지나치는 일이 종종 발생한다. 따라서 지하철, 버스 등에서 자신이 하차할 정류장을 알려주는 서비스 요구가 점차 생기기 시작하였으며, 이를 위한 스마트 기기 기반 다양한 기술 및 서비스 시스템들이 연구되었다.

대중교통 하차 알림 서비스로는 스마트 기기의 GPS(Global Positioning System)를 이용한 기술 [1, 2], Wi-Fi AP(Access Points)를 이용한 Wi-Fi 핑거프린트(Fingerprint) 기술 [3], 개인 정보 보호를 위해 스마트 기기의 GPS 대신 공공데이터인 '버스 정류장 Open API'를 활용한 기술[4] 등이 있다. GPS를 이용한 기술은 버스에서 사용이 효과적이거나, 버스가 터널을 지나가는 경우 혹은 애플리케이션이 백그라운드 상태로 가능 경우에는 GPS 센서의 정확도가 낮아지는 문제점이 있다. Wi-Fi 핑거프린트 기술은 각 장소에 위치한 AP 신호 값을 기반으로 사전 수집 데이터를 서버에 저장하고, 그 정보를 이용하여 하차 알림을 제공한다. 그러나 이 기술은 주변에 수많은 Wi-Fi AP가 존재하고, 그 신호의 영향을 받는 버스 정류장에서는 사용이 어려움이 있어, 지하철에서 보다 효과적인 정보 제공이 가능하다. 공공데이터를 활용하는 기술은 버스 회사에서 운영하는 각 버스들의 GPS 정보를 서버에서 통합하고 그 데이터를 스마트 기기에 전송하여 사용자의 현재 위치를 알려주고 하차 알림을 주는 방식이다. 그러나 이 기술은 사용자가 타고 있는 버스 위치와 하차 알림을 위해 스마트 기기와 서버 간의 통신을 지속적으로 해야 하는 문제점이 있다.

따라서 본 논문에서는 스마트 기기와 시내버스에 배치되어 있는 스피커에서 출력 가능한 고주파를 이용한 새로운 방식의 버스 하차 알림 시스템을 제안한다. 일반적으로 버스에 설치된 스피커는 사람들이 들을 수 있는 소리인 20Hz ~ 20kHz 주파수 범위를 출력할 수 있으며, 스마트 기기에 내장된 마이크는 가청주파수 범위를 인식할 수 있다. 따라서 우리는 버스의 스피커가 각 정류장에 대한 안

내 방송을 할 때 그 스피커를 통해 가청주파수의 높은 주파수에 해당하는 18kHz ~ 20kHz 범위의 특정 고주파 신호를 발생하게 한다. 스마트 기기의 애플리케이션은 수신된 소리를 분석하고 최근 스마트 기기 기반 데이터 통신 연구, 안내 애플리케이션 등에 사용되었던 두 개 고주파를 검출하여 전달된 특정 고주파 신호를 확인하고 해당 버스 노선의 정차할 위치를 비교하며 사용자가 설정한 정류장에 도달할 시 하차 알림을 제공할 수 있다. 이때 각 시도별 공공 버스 노선은 상당히 많은 수가 있기 때문에 수신된 고주파 신호와 버스 노선의 정류장을 매칭하기 위한 데이터는 사용자가 애플리케이션에 해당 버스 번호를 입력하면 서버를 통해 그 버스의 노선 정보를 단 한번 수신 받게 하고, 한 쌍의 고주파 값들과 버스 노선의 정류장을 각각 매칭하여 사용한다. 제안 시스템의 성능 검사를 위해 스마트 기기 기반 시내 공공 버스 하차 알림 애플리케이션과 서비스 서버를 구축하였으며, 비교 실험을 위해 특정 버스의 Open API 정보를 이용한 하차 알림 애플리케이션을 개발하여 총 10대의 스마트 기기를 이용한 하차 알림 실험 1,000회를 진행하였다. 실험 결과 제안 애플리케이션과 시스템은 98.9%의 하차 알림 정확도를 나타냈으며, 비교 실험 결과 제안 방법이 Open API를 이용한 방법과 동일한 성능을 보이면서 배터리 사용과 데이터 사용면에 더 우수함을 확인 하였다. 즉, 제안 애플리케이션과 시스템은 사람들에게 들리지 않는 고주파 소리를 이용하여 스마트 기기와 버스에 내장된 스피커를 활용한 시내 공공 버스 이용객에게 정확히 하차 알림을 하는 기술로써, 스피커로 도착지를 안내하는 전 세계 다양한 대중교통 서비스에 이용할 수 있는 뛰어난 기술이 될 것이다.

본 논문의 구성은 다음과 같다. 제 2 장에서 기존 대중교통 하차 알림 서비스에 관하여 설명하며, 본 제안 방법에서 사용한 가청주파수 영역의 고주파 신호 처리 연구를 설명한다. 제 3 장에서는 제안 애플리케이션과 서버를 이용한 시내 공공 버스 하차 알림 서비스에 사용되는 고주파 신호와 서비스 시스템의 흐름을 설명한다. 제 4 장에서는 제안 애플리케이션을 구현하여 시내 공공 버스 하차 알림의 성능 확인을 위한 실험과 기존 애플리케이션과의 비교 실험 및 그 결과를 설명하며, 제 5 장에서는 결론과 향후 연구 방향을 제시한다.

II. Related Work

본 장에서는 대중교통 하차 알림에 사용된 기존 연구 및 서비스에 관하여 설명하며, 제안 방법에서 사용한 가청주파수 영역의 고주파 신호 처리 연구를 설명한다. 대중교통

관련 서비스는 크게 지하철과 시내버스 두 개로 구분하며, 초기 서비스로는 운영 시간 및 교통 정보 등에 따라 정해진 장소에 버스가 도착 하거나, 정해진 역에 지하철이 도착하는 예측 정보를 제공하는데 그쳤었다. 이러한 서비스로는 오스트리아의 NEXTSTOP[5], SCOTTY[6], 미국의 TransitTimes+[7] 등이 있으며, 스마트 기기 보급이 활발해짐에 따라 이 서비스들은 사용자가 설정해 놓은 정차역에 다가올 때 사용자에게 알려주는 능동적인 서비스로 바뀌기 시작하였다. 이와 비슷하게 말레이시아에서는 사용자 스마트 기기의 GPS와 Geo-Fence를 이용한 버스 알림[8], RFID를 활용한 실시간 버스 위치 추적 서비스[9] 등이 제안 되었으며, 국내에서도 TMap, 카카오버스, Smart Subway 등이 Wi-Fi 핑거프린트 정보, GPS 정보를 기반으로 스마트 기기에 대중교통 하차 알림 서비스를 제공하고 있다. Wi-Fi 핑거프린트를 이용한 알림 서비스는 Wi-Fi AP 정보 변화가 적은 지하철 알림으로 주로 사용되고 있으며, AP 정보 변화에 따라 핑거프린트 값이 변화되는 경우 정확도 문제가 발생 할 수 있기 때문에 이 서비스를 제공하는 회사에서는 지속적인 업데이트와 유지 보수가 필요한 단점이 있다. GPS 정보를 기반으로 한 대중교통 하차 알림 서비스는 초기에 스마트 기기에 내장된 GPS 위치 값만을 이용한 방식을 초기 사용하였다. 그러나 점차 차량 내에서의 GPS 값이 부정확한 경우를 보완하고자 공공 데이터인 버스 GPS 정보 통합 서버를 활용한 현재 위치 알림과 하차 알림 서비스를 사용하고 있다. 그리고 최근 제안된 BLE 비콘과 컴퓨터비전을 이용한 스마트 버스 시스템 또한 Open API 정보를 수신하여, 하차 알림, 하차벨 자동 작동 등을 구현하였다[10].

본 연구에서 사용하는 고주파를 이용한 기존 연구들은 가청주파수 범위(20Hz ~ 20kHz)의 고주파 영역인 18kHz

~ 22kHz를 사용하여 스마트 기기의 내장 스피커와 마이크를 사용해 지정된 신호를 전달하거나 제어한다. Bihler는 20kHz와 22kHz 고주파를 사용하여 FSK(Frequency Shift Keying) 기술로 신호를 전달하고, 전달 신호의 오류를 보정하기 위해 해밍 코드를 적용하여 스마트 기기에 신호를 보내어 정보를 제공하는 스마트 가이드 시스템을 제안하였다[11]. Chung은 신호를 보다 세계 전달하기 위해 2개의 기반 신호와 1개의 가변 신호를 사용하여 고주파 신호와 Wi-Fi를 이용한 스마트 기기간의 데이터 공유 방법을 제안하였으며, TV 시청자에게 고주파 신호를 전달하여 특정 상품의 광고를 제공할 수 있는 애플리케이션을 제안하였다[12]. 또한 고주파 신호만을 이용하여 PC와 스마트 기기 간의 데이터 통신을 위한 고주파 근거리 통신 방법을 제안하고, 그 정확도를 높이기 위해 CRC(Cyclic Redundancy Check)를 적용하였다[13]. 즉, 기존 가청주파수 내의 고주파를 이용한 방법들은 스마트 기기가 위치한 주변 소리 정보를 내장 마이크를 통해 수집하고, 수집한 소리들로부터 특정 고주파 신호를 검출하는 순간 그 신호를 분석하며 전달된 신호와 매칭되는 값으로부터 사용자에게 알림, 정보 전달 등의 서비스를 수행하게 된다.

III. A Disembarking Notification System at Public City Bus

본 장에서는 스마트 기기 애플리케이션과 서버를 이용한 시내 공공 버스 하차 알림 서비스에 사용되는 고주파 신호와 서비스 시스템의 흐름을 설명한다. 제안 시스템의 전체 동작 흐름은 Fig. 1과 같다. Fig. 1에서 사용자가 시내버스

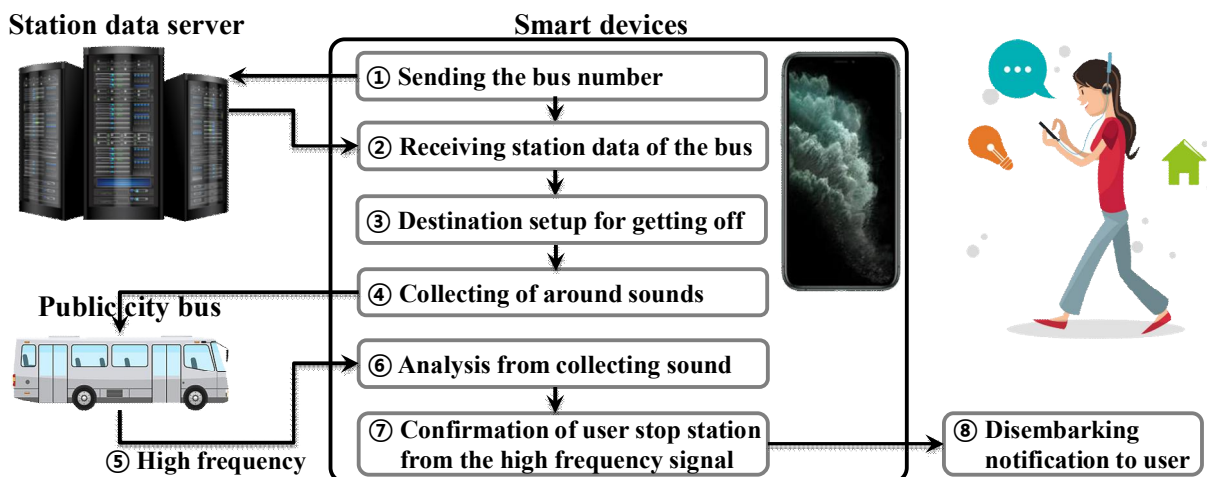


Fig. 1. The flow of the disembarking notification system using smart device and high frequency

를 타면서 그 버스의 번호를 입력하면, 스마트 기기는 서버로 번호를 전송하고(①) 해당 버스의 정류장 데이터를 수신하게 된다(②). 그 후 스마트 기기는 노선의 현재 위치와 버스 경로 정보를 표시하며, 사용자는 자신이 하차할 목적지를 설정한다(③). 제안 애플리케이션은 스마트 기기의 내장 마이크로 주변 소리, 안내 방송, 고주파 소리 등을 수집하며(④), 버스는 각 정류장에 도착 전 안내 방송과 함께 해당 정류장 값을 고주파 신호로 발생한다(⑤). 제안 애플리케이션에서는 수집한 모든 소리들 중 두 쌍 이상의 고주파가 발생하는 지를 분석하며(⑥), 두 쌍의 고주파가 검출되는 경우 사용자가 지정한 하차 목적지와 정차 예정인 정류장의 값을 비교한다(⑦). 즉, 두 값이 일치하는 경우 애플리케이션은 사용자에게 최종적으로 하차 알림을 실행한다(⑧).

사용자의 버스 번호 입력 및 전송은 Wi-Fi, LTE를 이용하여 서버와 통신하며, 이때 사용자의 현재 위치를 파악하기 위해 GPS 데이터를 함께 전송한다. 서버에서는 해당 번호 노선 이미지, 각 버스 정류장 이름, 정류장별 신호 값, 사용자 GPS 정보에 가장 가까운 버스 정류장 이름을 스마트 기기에 전송한다. 예를 들어 사용자가 인천공항버스 “6017”번을 입력하여 버스 번호를 전송하면, Fig. 2와 같이 해당 버스의 노선도를 스마트 기기로 전송한다.



Fig. 2. The 6017 bus route received from server system

Fig. 2에서의 전체 정류장 개수는 16개이며, 각각의 정류장의 안내 방송이 나올 때 해당 정류장에 관한 고주파 신호 한 쌍은 18kHz ~ 20kHz 범위에서 100kHz 단위로 2개를 선택하여 지정한다. 단, 한 쌍의 고주파가 600Hz 이내인 경우 주파수 간섭에 의한 오류가 생길 수 있기 때문에 주파수 간격은 600Hz 이상 차이를 둔다[14]. 즉, 정류장과 매칭하는 한 쌍의 고주파는 “18.0kHz & 18.7kHz”, “18.0kHz & 18.8kHz”,, “19.3kHz & 20.0kHz”까지 총 105 가지

쌍을 구성할 수 있으며, Fig. 2의 각 정류장은 아래 Table 1과 같이 각각의 고주파 신호 한 쌍으로 매칭시킬 수 있다.

Table 1. The matching with bus stations and a pair of high frequencies

Bus station	High frequencies
Seoul National Univ.	18.0 kHz & 18.7 kHz
Nakseongdae Stn.	18.0 kHz & 18.8 kHz
Bongcheon Sageori	18.0 kHz & 18.9 kHz
Bongcheon Stn.	18.0 kHz & 19.0 kHz
...
Airforce Club	18.0 kHz & 20.0 kHz
Incheon Airport Terminal 1	18.1 kHz & 18.8 kHz
Incheon Airport Terminal 2	18.1 kHz & 18.9 kHz

Table 1에서 “18.0kHz & 18.7kHz”로 구성된 고주파 한 쌍은 “Seoul National Univ.” 종점으로 설정한 것이며, “18.1kHz & 18.8kHz”는 인천공항 터미널 1, “18.1kHz & 18.9kHz”는 인천공항 터미널 2로 설정한 것을 확인할 수 있다. 그리고 이렇게 구성된 고주파 신호는 버스에서 스피커를 통해 정류장 안내 방송이 나올 때 함께 발생할 수 있도록 방송 파일을 제작한다. 이때 고주파 신호는 안내 방송이 나오는 내내 발생하지 않으며, 1초 간격으로 1초간 2회, 전체 3초 동안 발생한다. 이는 Chung의 실험에서 2.8초 이상 시간 동안 고주파 한 쌍을 발생할 때 스마트 기기에서의 고주파 검출 정확도가 높기 때문이다. 아래 Fig. 3은 버스 정류장 안내 방송과 함께 송출되는 고주파 신호의 시간에 따른 고주파 송출을 표현한 것이다.

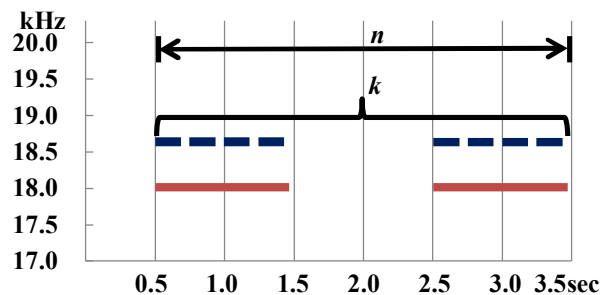


Fig. 3. A pair of high frequencies matched with bus station

Fig. 3에서 고주파 신호 한 쌍은 18.0kHz & 18.7kHz이며 이 값은 3초 동안(n), 2회(k) 발생한다. 즉, 0.5초부터 3.5초까지 18.0kHz(아래 빨간색 실선)와 18.7kHz(위 파란색 점선) 신호가 발생하는 것이며, 이 신호는 3초 시간 동

안 1초 발생 후, 잠시 1초 발생을 멈추고, 이어 다시 1초 간 발생 한다. 처음 0.5초에서 1.5초까지 1회, 2.5초에서 3.5초까지 다시 1회, 총 2회를 동일한 신호를 발생한다. 이 고주파 신호가 버스 정류장 안내 방송과 함께 스피커를 통해 발생하면, 사용자의 스마트 기기 애플리케이션은 마이크를 통해 18.0kHz 이상의 모든 고주파를 지속적으로 분석하여, 고주파가 검출되는 경우 다시 반복 검증을 실행하여 동일 고주파 쌍임을 확인한다. Fig. 4는 제안 애플리케이션이 고주파 신호 한 쌍을 검출하기 위한 수행 동작을 슈도 코드(Pseudo code)로 작성한 것이다.

```

While(receive  $S_t$ ) Do
   $i \leftarrow 0$ ;
   $j \leftarrow 0$ ;
   $F_t \leftarrow \text{FFT}(S_t)$ ;
  sortByBin( $F_t$ );
  If ( 1st HF value of  $F_t$  is over 18.0kHz
    and 2nd HF value of  $F_t$  is over 18.0kHz) Then
     $i \leftarrow$  1st HF value of  $F_t$ ;
     $j \leftarrow$  2nd HF value of  $F_t$ ;
    sleep( $m$ )
    receive  $S_{t+1}$ 
     $F_{t+1} \leftarrow \text{FFT}(S_{t+1})$ ;
    sortByBin( $F_{t+1}$ );
    If ( 1st HF value of  $F_{t+1}$  is over 18.0kHz and
      2nd HF value of  $F_{t+1}$  is over 18.0kHz) Then
      If (  $i =$  1st HF value of  $F_{t+1}$  and
         $j =$  2nd HF value of  $F_{t+1}$ ) Then
        theBusStationValue;
      END If
    END If
  END If
END WHILE

```

Fig. 4. The pseudo code for detection a pair of high frequencies matched with bus station

Fig. 4에서 S_t 는 t 시간에 내장 마이크로부터 수집한 오디오 데이터를 의미하며, F_t 는 S_t 를 FFT(Fast Fourier Transform)하여 각 주파수 bin 값을 얻는 배열 데이터 값이다. 따라서 이 배열 데이터 값을 많은 bin 수를 갖는 순서대로 정렬(SortByBin)하고, 그 값들 중 가장 많은 bin 수를 갖는 첫 번째 고주파(1st HF)와 두 번째로 많은 bin 수를 갖고 있는 두 번째 고주파(2nd HF) 모두가 18.0kHz를 넘을 때 각각 i 와 j 변수에 입력한다. 그리고 m 시간인 1초를 대기하고, 다시 S_{t+1} 를 수집, FFT 변환을 실행해 F_{t+1} 값을 계산하며 정렬시킨 후 앞서 검출한 방식과 동일하게 고주파 한 쌍을 검출한다. 이때 검출한 고주파 한 쌍과 기존 저장하고 있는 고주파 한 쌍(i, j)을 비교하여 모두 일치하면 제안 애플리케이션은 고주파 한 쌍의 신호 값을 이용하여 정류장 정보를 선택하고 사용자에게 하차 알림을 수행한다.

IV. Experiments and Evaluation

이번 장에서는 스마트 기기와 고주파를 이용한 시내버스 하차 알림 애플리케이션 화면과 서버 시스템을 소개하며, 제안 시스템을 사용하여 사용자에게 하차 알림 실험을 수행한 것과 그 결과를 분석한다. 사용자에게 하차 알림을 위한 애플리케이션의 화면 구성은 Fig. 5와 같다. Fig. 5는 사용자가 버스를 승차 후 해당 버스 번호를 입력하여 서버에 그 값을 전송한 후 나타나는 제안 애플리케이션 메인 화면이다. Fig. 5의 왼쪽에 위치한 그래프는 Fig. 4에서 언급한 버스 정차역 안내방송 소리를 수집하고 실시간

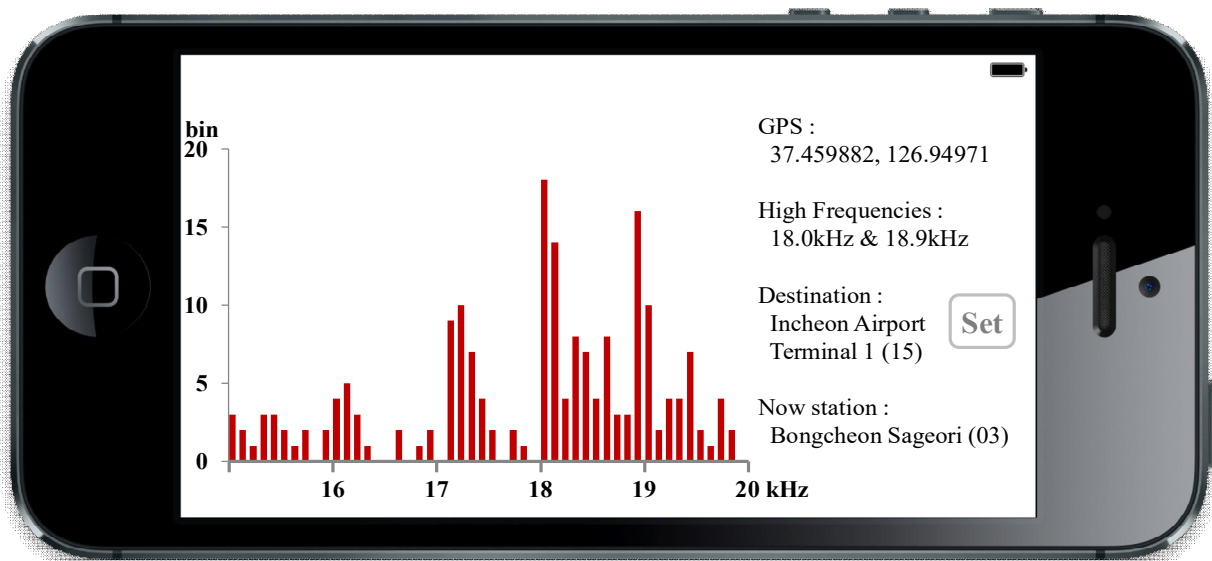


Fig. 5. Screen composition of the proposed application for disembarking at public city bus

FFT 처리한 값들 중 15kHz ~ 20kHz 범위를 그래프로 표현한다. 그래프에서는 18.0kHz와 18.9kHz가 월등하게 나타나는 것을 볼 수 있으며, 이는 Fig. 4의 슈도 코드 과정을 거쳐 오른쪽 High Frequencies에 고주파 신호 값을 표시하게 된다. Fig. 5의 오른쪽에 나타나는 값들은 위에서 서부터 차례로 다음 값을 표현하고 있다.

- GPS : 사용자 GPS 위치 값
- High Frequencies : 검출된 고주파 신호 값
- Destination : 사용자가 지정한 목적지
- Now station : 안내 방송 고주파 신호의 정차역

이때 목적지(Destination) 옆에 위치한 Set 버튼은 사용자가 버스를 승차한 후 목적지를 설정할 때 사용하는 것이며, 버튼 터치 시 서버로부터 전송 받은 해당 버스의 정차역 리스트를 보여주고, 사용자가 선택하게 한다. 그리고 사용자가 리스트에서 하차 지역을 선택하면, Fig. 5의 목적지(Destination) 아래에 해당 목적지가 나타나고, 안내 방송 고주파 신호 값이 목적지 값과 일치할 때 제안 애플리케이션은 사용자에게 하차 알림을 표시한다. 제안 애플리케이션은 사용자가 다른 콘텐츠를 사용하고 있더라도, 배경 모드(Background mode)에서 지속적으로 고주파 수집을 진행하며, 지정한 목적지 값과 현대 도착할 정류장의 값이 동일할 때 목적지 도착 알림을 로컬 알림(Local notification)으로 보여준다. 사용자는 다른 콘텐츠를 사용하던 중이더라도 해당 알림을 터치하여 다시 제안 애플리케이션을 실행하며, 실행 된 제안 애플리케이션은 Fig. 6과 같이 목적지에 도착 예정임을 알려준다.



Fig. 6. The screen of the proposed application when bus is approaching the destination station

Fig. 6에서는 “Incheon Airport Terminal 1 (15)”가 이번 버스 정차역인 것을 크게 보여주게 되며, 사용자는 이 알림을 보고 목적지를 지나치지 않고 내릴 수 있다. 그

리고 사용자는 환승을 할 경우 또 다시 환승하는 버스를 타면서 Fig. 6의 확인(Confirm) 버튼을 터치한 후, 동일한 방법으로 버스 번호 전송, 목적지를 선택하여 다음 하차 알림을 받을 수 있게 된다.

제안 애플리케이션과 서버 시스템의 정확성 및 유효성을 검증하기 위해 본 논문에서는 국내 공항버스 중 “6017” 노선버스의 정류장들을 1번(서울대학교) ~ 16번(인천공항 터미널2)으로 설정하여 애플리케이션 내에 각 값과 정류장 이름을 매칭하였다. 실험은 참가자 10명을 모집하였으며, 참가자들은 자신의 스마트 기기에 제안 애플리케이션을 설치하였다. 실험에 사용된 스마트 기기는 iPhone 8, iPhone 8+, iPhone X, Galaxy S10, Galaxy S20으로 각각 2대씩 총 10대를 사용하였으며, 버스 방송이 안내될 때 고주파 신호를 송출할 스피커와 애플리케이션은 노트북과 별도의 스피커를 활용하여 자바 스크립트(JavaScript) 기반 웹 애플리케이션을 개발하여 사용하였다. 실험 참가자는 “6017” 버스의 종점인 서울대학교 정차역에서 모두 승차를 하고, 버스 내의 편안한 위치에 자리를 잡고 앉았으며, 고주파 신호를 발생하는 스피커와 노트북은 버스의 중앙 위치에 배치하였다. 각각의 참가자는 목적지 설정을 출발 정류장에서 3 정거장 이상을 설정하도록 하였으며, 참가자는 지정한 목적지에 도착 전에 안내 방송이 나올 때 하차 알림이 잘 동작하는지를 확인하였다. 예를 들어 Fig. 2에서 참가자가 서울대학교 역에서 승차한 후, 가장 가까운 하차 정류장 지정은 봉천역 이후를 설정할 수 있으며, 버스 안내에서 봉천역을 안내할 때 하차 알림이 실제 일어나는지를 확인하도록 한다. 그리고 이 참가자는 다음 하차 정류장 지정으로 3 정거장 이후인 구로 디지털 단지역을 지정하고 이를 확인할 수 있다. 즉, 한 참가자는 1회 버스를 탔을 경우 출발지부터 끝까지 최대 5회 하차 알림 실험을 진행할 수 있다. 따라서 본 실험은 일일 왕복 2회 이상, 20회 하차 알림을 실험하고, 5일간 진행하여 참가자별 100회의 하차 알림을 수행하였다. 버스 번호에 따른 정보 제공을 위한 서버 환경은 Apache 2.2.14, MySQL 5.1.39, PHP 5.2.12를 사용하였으며, 하드웨어는 Intel(R) Core(TM) i5 CPU 750, 8G Ram을 사용하였다. Fig. 7은 실험 참가자들의 스마트 기기로부터 버스 안내 방송과 함께 발생하는 고주파 신호를 인식하여 사용자에게 하차 알림을 정확히 수행한 결과 그래프이다.

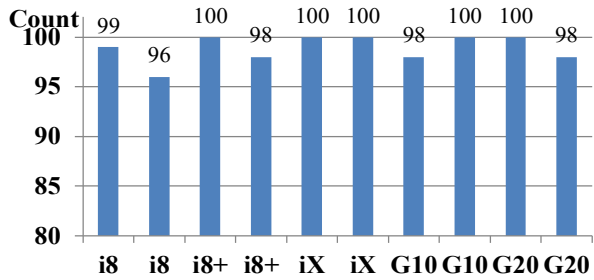


Fig. 7. The experiment result of disembarking notification using smart device and high frequencies

Fig. 7에서 i8은 iPhone 8, i8+는 iPhone 8+, iX는 iPhone X, G10은 Galaxy S10, G20은 Galaxy S20을 의미하며, 각 기기에 표시된 수치는 참가자가 지정한 목적지에 하차 알림이 정확히 수행된 횟수를 나타낸 것이다. Fig. 7에서 5대가 100회 (100%) 정확히 하차 알림 한 것을 볼 수 있으며, 1대의 스마트 기기가 99회, 3대가 98회, 두 번째 위치한 iPhone 8만이 96회 정확성을 나타낸 것을 볼 수 있다. 이 iPhone 8이 다른 기기에 비해 정확성이 낮은 이유는 이 실험 참가자가 다른 참가자들에 비해 고주파를 발생하는 스피커로부터 가장 먼 위치에 앉았었으며, 이에 따라 해당 고주파를 인식하지 못했기 때문이라 예상된다. 따라서 본 실험에서 제안 애플리케이션과 서버 시스템은 98.9%의 하차 알림 정확성을 나타내었다.

다음 실험으로는 “버스 정류장 Open API”를 사용하는 애플리케이션과 제안 방법과의 비교 실험을 진행하였다. 기존 애플리케이션의 Open API 활용과 구현을 하고자 공공데이터포털(<http://data.go.kr>)의 “버스위치정보 조회” API를 사용하여 애플리케이션을 개발하였으며, 실험 참가자는 앞의 실험과 동일한 인원, 동일한 기기를 이용하였다. 기기가 중복되지 않게 5명씩 두 팀으로 나누어 10명이 함께 동일 버스를 타고, 한 팀은 기존 애플리케이션을 사용하며, 다른 한 팀은 제안 애플리케이션을 사용하여 실험을 진행하였다. 실험 횟수는 100회 씩 5일간 진행하였으며, 하차 알림 실험 결과는 Fig. 8과 같다.

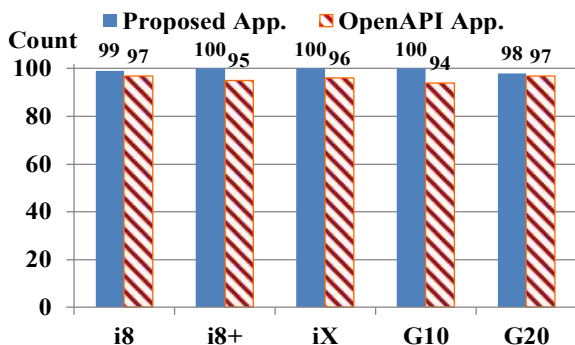


Fig. 8. The comparison experiment result with the proposed application and open API application

Fig. 8에서 제안 방법은 각각 99, 100, 100, 100, 98회 정확한 알림을 하여, 평균 99.4% 정확도를 나타내었다. iPhone 8의 99회와 Galaxy S20의 98회는 실험 참여자가 스피커로부터 가장 먼 거리에 있었거나, 스마트 기기의 마이크를 잠시 가려 고주파 신호를 인식하지 못하였기 때문에 하차 알림을 동작하지 않은 것으로 예상된다. 반면에 Open API를 이용한 애플리케이션의 경우 97, 95, 96, 94, 97회 정확한 알림을 하여, 평균 95.8% 정확도를 나타내었다. 대부분의 경우에서 100% 정확도가 나타나지 않는 이유는 버스의 이동하는 GPS 값을 서버로 지속적으로 전송하고, 그 데이터를 가공하여 Open API로 제공할 때 데이터가 가끔 누락 되거나 하여 현재 버스의 위치 값을 정류장이 지난 후 전달하는 경우가 있기 때문이라 생각된다.

또 다른 실험으로는 위의 비교 실험을 시작할 때 스마트 기기 각각의 배터리 소모를 함께 측정하였다. 실험 시작 전 각 스마트 기기의 배터리는 100% 충전을 한 상태에서 시작하였으며, 배터리 측정은 버스의 각 방향에 종점 도착 총 4회를 측정하였다. 배터리 소모량 측정 실험에서는 참여자가 다른 여러 가지 애플리케이션을 실행하거나 할 경우 Wi-Fi, GPS, 화면 밝기 조절, 그 밖에 센서 등의 사용에 따라 큰 차이를 가져오기 때문에, 실험하는 동안은 해당 애플리케이션 외에 타 애플리케이션의 사용을 자제하였다. Fig. 9는 제안 방법과 Open API를 이용한 방법의 배터리 소모 결과이다.

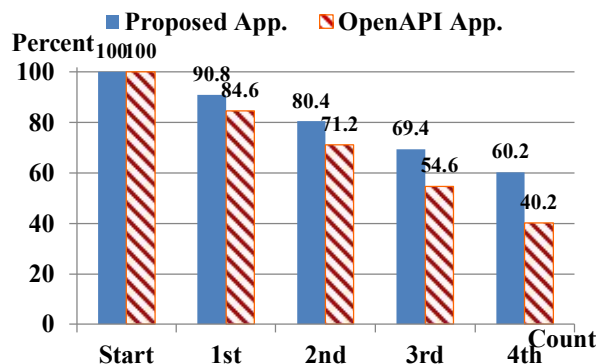


Fig. 9. The comparison experiment result about battery consumption

Fig. 9에서와 같이 제안 애플리케이션은 시작에서부터 각 회 측정 시 9.2%, 10.4%, 11%, 9.2% 순으로 줄어 총 4회 동안 39.8% 배터리가 소모 되었다. 반면에 Open API 애플리케이션은 각 회 측정 시 15.4%, 13.4%, 16.6%, 14.4% 순으로 줄어 총 4회 동안 59.8% 배터리가 소모 되었다. 동일 동작을 수행하는 동안 Open API 애플리케이션이 제안

애플리케이션에 비해 20% 이상 배터리 소모가 많이 되는 이유는 공공데이터 포털에서 제공하는 서버로부터 Open API 데이터를 지속적으로 가져오기 위한 LTE를 사용하고, 가져온 데이터를 가공하여 이 부분을 처리하면서 배터리의 소모가 더 많은 것으로 볼 수 있다. 따라서 우리가 제안한 고주파 신호와 버스 하차 알림을 수행하는 애플리케이션은 시내 공공버스를 이용하는 승객들에게 스마트 기기의 콘텐츠를 즐기고 있더라도 본인의 목적지를 지나치지 않고 하차할 수 있게 알려줄 수 있는 유용한 기술이 될 것이다.

V. Conclusions

본 연구에서는 스마트 기기의 내장 마이크와 버스에 설치된 스피커의 안내 방송에 고주파 신호를 혼합하여 추가적인 통신 없이 새로운 방법으로 사용자에게 하차 알림을 수행하는 애플리케이션과 서버 시스템을 제안하였다. 그리고 실험을 통해 제안 애플리케이션이 시내 공공버스 이용 정차역 안내와 함께 고주파 신호로 다음 도착할 정차역의 값을 판단하며, 사용자가 설정한 역에 도달할 경우 하차 알림을 통해 정확히 사용자에게 알림을 수행하는 것을 확인하였다. 즉, 제안 애플리케이션과 서버 시스템은 정차역 안내를 하는 지하철, 버스 등의 대중교통을 이용하는 사람들에게 다양한 콘텐츠를 즐기고 있더라도 사용자가 하차할 역만 지정해 놓으면, 하차 알림을 제공하는 유용한 애플리케이션 서비스가 될 수 있을 것이다. 또한 스피커로 다음 도착지, 정차역을 안내하는 세계의 여러 종류의 대중교통 서비스에 활용할 수 있는 뛰어난 기술이 될 것으로 예측할 수 있다.

추후 연구로는 제안 방법을 이용하여 시내 공공버스 뿐 아니라 지하철에서 제안 애플리케이션을 사용한 하차 알림 실험을 진행하여 그 성능의 우수성을 확인할 것이다. 또한, 고주파 신호 분석과 매칭의 고도화를 위해 기존 사용되던 Wi-Fi 핑거프린트 기술, 공공 데이터인 버스 GPS 정보 통합 서버를 활용한 기술들을 제안 방법과 융합하여 새로운 알고리즘과 서비스 시스템을 개발함으로써 버스 하차 알림 정확도를 100% 달성할 수 있도록 노력할 것이다. 그리고 본 논문에서는 가청주파수 범위의 고주파를 이용하기 때문에, 주변 다양한 환경 및 잡음에 의한 신호 전달의 정확성이 영향을 받을 수 있다. 따라서 이에 관한 추가 실험 및 전달 정확성 향상에 관한 연구를 진행할 것이다.

ACKNOWLEDGEMENT

This research project was supported in part by the Ministry of Education under Basic Science Research Program(NRF-2019R1H1A1079104, NRF-2020R1F1A1048133), respectively.

REFERENCES

- [1] H. Kasahara, M. Mori, M. Mukunoki, and M. Minoh, "Transportation mode annotation of tourist GPS trajectories under environmental constraints," In *Information and Communication Technologies in Tourism 2015*, pp. 523-535, December 2015. DOI: 10.1007/978-3-319-14343-9_38
- [2] J. H. Park, S. H. Kang, and Y. G. Seo, "Advanced bus information system using smart phone GPS," *Journal of the Korea Society of Computer and Information*, Vol. 19, No. 12, pp. 247-255, December 2014. DOI: 10.9708/jksci.2014.19.12.247
- [3] Smarter Subway, loplax, <https://loplat.com/cases>
- [4] J. Y. Lee, Y. S. Ryu, and J. H. Hwang, "Development of guidance app for public transportation," *Journal of Digital Contents Society*, Vol. 18, No. 1, pp. 115-121, February 2017.
- [5] Fahrplanauskunft and Routenplaner - nextstop.at, MobileCity, <http://www.nextstop.at>
- [6] OEBB, Scotty, http://www.oebb.at/de/Reiseplanung/Fahrplanauskunft/Mobile_Dienste/SCOTTY_mobil/index.jsp
- [7] Zervaas Enterprises, TransitTimes+ Trip Planner, <http://transitiemesapp.com/>
- [8] M. A. Ayob, M. N. F. Saaid, K. M. Dimyati, and K. A. Maarof, "Destination alarm notification for public transportation passenger using Geo-Fence in mobile App," In *IOP Conference Series: Earth and Environmental Science*, Vol. 169, No. 1, pp. 012087, June 2018. DOI: 10.1088/1755-1315/169/1/012087
- [9] B. Y. Low, S. H. Dahlan, and M. H. A. Wahab, "Real-time bus location and arrival information system," *Proceeding of the 2016 IEEE Conference on Wireless Sensors (ICWiSE)*, pp. 50-53, October 2016. DOI: 10.1109/ICWiSE.2016.8187761
- [10] M. You, and E. Rhee, "Smart bus system using BLE beacon and computer vision," *Journal of Institute of Korean Electrical and Electronics Engineers*, Vol. 22, No. 2, pp. 250-257, June 2018. DOI: 10.7471/ikeee.2018.22.2.250
- [11] P. Bihler, P. Imhoff, and A. B. Cremers, "SmartGuide-A smartphone museum guide with ultrasound control," *Procedia Computer Science*, Vol. 5, pp. 586-592, 2011. DOI: 10.1016/j.procs.2011.07.076
- [12] M. B. Chung, "Effective near advertisement transmission method

for smart-devices using inaudible high-frequencies," *Multimedia Tools and Applications*, Vol. 75, No. 10, pp. 5871-5886, March 2015. DOI: 10.1007/s11042-015-2553-1

[13] M. B. Chung, "Short distance data transmission method using inaudible high-frequencies between smart devices," *Telecommunication Systems*, Vol. 70, No. 4, pp. 583-594, August 2018. DOI: 10.1007/s11235-018-0497-2

[14] J. B. Kim, J. E. Song, and M. K. Lee, "Authentication of a Smart Phone User using Audio Frequency Analysis," *Journal of Korea Institute of Information Security and Cryptology*, Vol. 22, No. 2, pp. 327-336, April 2012.

Authors



Myoungbeom Chung received the B.S., M.S. and Ph.D. degrees in Dept. of Digital Media from Soongsil University, Korea, in 2004, 2006 and 2010, respectively. Dr. Chung worked on BK21 project as a post-doctoral

follow at the Soongsil University at Seoul, in 2010 and 2011. From 2012 to 2014, he was with the School of Information and Communication Engineering, Sungkyunkwan University (Korea). Since 2015, he is now an assistant professor of Division of Computer Engineering, Sungkyul University (Korea). His research interests include high frequency communication, copyright protection technique, mobile computing, mobile software development, audio signal processing, and recommendation system.