One-to-many data transmission for smart devices at close range

Myoungbeom Chung
Division of Computer Engineering
610 ho Jaelimkwan, Sungkyul University
Anyang City, South Korea
nzin@sungkyul.ac.kr

***Abstract*—In this paper, we propose a new method for one-to-many data transmission in smart devices at close range. Existing methods require the use of an extra application service where the operating system differs between smart devices. In contrast, the proposed method makes use of the smart device's built-in speaker and microphone to confirm the transmission signal. Data is then transmitted via Wi-Fi or long term evolution. The proposed method therefore overcomes operating system issues with existing methods. Besides, BUMP technology works in a similar way to the method we propose, it only supports one-to-one transmission. To evaluate the efficacy of the new method, we tested one-to-many data transmission in an experiment: the results showed a 96% success rate. As a result, we believe that the proposed method is an effective tool for one-to-many data transmission for smart devices at close range.**

Keywords—Near data share; High frequencies; Data transmission; Signal processing; Smart device

I. Introduction

Today, as different modules are added and their performance shows significant improvement, smart devices have become personal communication instruments as well as important business tools. Not long ago, when we wanted to share data or electronic files in a conference, each party had to turn on their computer to send or receive the data via email, or they could share files by connecting to the same access point (AP). Today, however, we can send files and share data much more easily, using smart devices.

Typical methods for sending files or sharing data include communication and data sharing technologies that use Bluetooth, Near Field Communication (NFC), other file sharing methods dependent on Wi-Fi, and application services such as Wi-Fi Direct or AirDrop technology, which are supported by the operating system (OS) of the smart device. Developed by Ericsson Inc. in 1994, Bluetooth technology is the industry standard for personal area networks (PANs) [1]. Bluetooth 4.0, announced in June 2010, enhanced Bluetooth Low Energy (BLE) technology on the Bluetooth 3.0 platform, which can send large data such as image and video files at a 24 Mbps transmission speed [2]. However, Bluetooth 4.0 requires pairing for data transmission between smart devices, and it cannot work where the OS of the smart devices differs.

This research project was supported in part by the Ministry of Education under Basic Science Research Program (NRF-2013R1A1A2061478), respectively.

NFC data sharing technology uses the NFC Reader, a built-in smart device, and NFC Tag, which can read and write to smart devices. NFC can easily transmit data and files between smart devices. With technology based on Radio-frequency identification (RFID), it was first used on the Nokia 6131 in 2007 [3]. NFC works well within a 10 cm range, with a maximum transmission speed of 424 bps [4]. Smart devices fitted with NFC technology can make use of the following three modes: NFC Card Emulation, NFC Read and Write, and NFC Peer-to-Peer (P2P) mode. Android devices now support Android Beam or S Beam for data sharing, again using NFC technology [5]. However, NFC technology does not work on iPhone, iPad, or on currently-released Android devices, which do not yet include an NFC module.

We next look at existing file-sharing methods that use Wi-Fi, including the following application services: Whatsapp, Kakaotalk, Viber, and AllJoyn, among others [6-8]. These application services use a built-in Wi-Fi module on the smart device and an application service server. The application services focus on communication between application users and support data sharing between service users as an additional function. As a result, users can only share data with friends who are members of the same application service.

To resolve issues with existing data sharing methods, BUMP Inc. launched the BUMP application in 2010, supplying a BUMP application programming interface (API) that enables the BUMP data sharing function to work with any application [9, 10]. BUMP technology helps overcome some of the drawbacks of existing methods, such as pairing, the need for an NFC module, the need to sign up to a specific application service, and different OS issues. BUMP technology is very popular because it supports easy one-to-one (1:1) data transmission between smart devices. BUMP uses the shaking motion of the smart device as a trigger signal to start data transmission. When a user shakes his or her smart device, the device sends global positioning system (GPS) information to the BUMP server, which determines whether to start data transmission depending on the distance between the smart devices. When the BUMP server permits data transmission, the sending smart device starts to send data to the receiving device. BUMP Inc. then released data sharing technology for use between smart devices and personal computers (PCs). The limitations to BUMP technology are that it only supports 1:1 data transmission, whether between smart devices or a between a smart device and a PC.

978-1-4673-9948-7/16 $31.00 © 2016 IEEE
DOI 10.1109/IoTDI.2015.15

As a result, we propose a new technology to overcome drawbacks with existing methods and to support one-to-many (1:N) data transmission between multiple smart devices. The proposed method uses the high frequency signals that are emitted from a smart device’s built-in speaker as the trigger signal, in preference to the shaking action of the BUMP API. The work flow of 1:1 data transmission in the proposed method is similar to the work flow of the BUMP API. The high frequency signals are not transmitted via the audio data hiding method, based on phase information, but on a signal processing method, based on the high frequency of inaudible sound [11]. The high frequencies used in the proposed method are divided into a range for the transmission of real data (19.0 kHz ~ 21.0 kHz), and a check range (21.0 kHz ~ 22 kHz) for error detection. Real data from the sending device includes a specific key value, which is sent to the receiving smart device via high frequency audio signals. The sending device transmits a sharing device number and GPS information about itself to a sharing server, via Wi-Fi or LTE. After detecting the trigger signal (high frequency signals from the built-in microphone), the receiving devices send a receiving key value and GPS information to the sharing server, asking the server to transmit the data. We developed a 1:N data transmission application to evaluate the performance of the proposed method. We carried out both 1:1 and 1:N data transmission experiments using the application. For 1:1 data transmission, the success rate was 97%, which is 8% higher than the BUMP API success rate. For 1:N data transmission, the success rate was 96%. We therefore believe that the proposed method could be an effective tool for 1:N data transmission between multiple smart devices at close range.

The present paper is organized as follows. In Section 2, we describe existing data transmission technology that uses high frequency signals and we look at the work flow of the BUMP API, which has enabled data sharing between devices at close range. In Section 3, we explain the processing technique used by the new method that we have developed and we look at the work flow for 1:N data transmission. In Section 4, we show how our data transmission application applied the proposed method, and we discuss the results of the experiment in terms of the accuracy and usability of 1:1 and 1:N data transmission using the proposed application. We present the conclusions in Section 5.

II. Previous work

This section examines existing data transmission technology using high frequency signals and looks at the work flow of the BUMP API, which is a popular technology for data sharing between smart devices at close range. Much research has been carried out in the transmission technology field regarding inaudible information in audio signals, including audio watermarking, steganography, covert communication, and so on [12-14]. Some studies on the use of high frequency audio signals have identified tools that track the indoor positioning of smart devices.

Research has gradually focused on information transmission technology based on high frequency signals using the speakers and microphones that are built into smart devices. Viacheslav proposed an indoor position tracing algorithm using a built-in speaker and four microphones [15]. Bihler demonstrated transmission technology using the high frequencies of 20 kHz and 22 kHz to send 8 bits of data at 208 ms [16]. Because the volume of data transmission per second was so small, Bihler used the high frequencies as the trigger signal. The real data was received by a data information server according to the value of the trigger signal. Lee suggested a data transmission method for user authentication between a smart device and a PC with information transmission technology that used high frequency signals [17]. This method used a combination of two high frequencies and was able to send 2 bytes of data in 8 seconds. Chung proposed an effective advertisement transmission method that could actively transmit advertisements to smart devices by applying Bihler’s trigger signal with Lee’s authentication method [18].

Next, we look at the work flow of the BUMP API (see Fig. 1). The workflow is made up of two smart devices and a BUMP server.

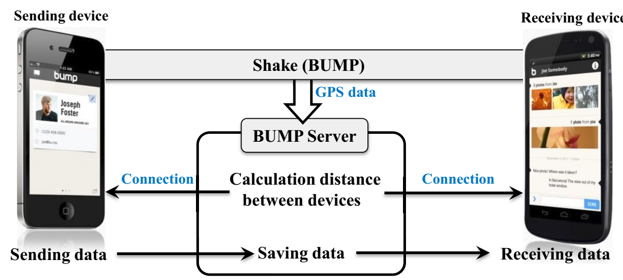

Fig. 1. The work flow of the BUMP API and its composition

In Fig. 1, when the users shake their smart device, each device sends GPS information to the BUMP server. The BUMP server decides whether or not to establish a connection between the devices depending on the distance between them. If a connection is established, the sending device sends the real sharing data to the BUMP server, which transmits the data to the receiving smart device. The release of the BUMP API has contributed greatly to data sharing between smart devices. However, this method has one drawback: namely, that it only works for 1:1 data transmission. In other words, if three users shake their smart devices at the same time, the BUMP server cannot tell to which smart device it should send the shared data. As a result, if three users want to share data with each other using the BUMP API, they must shake in order: from user A to user B, and from user B user to user C.

III. One-to-many data sharing among smart devices

This section explains the work flow of the proposed new method, which supports 1:N data transmission between smart devices at close range. The section also explores the high frequency processing technique that the new method uses. This high frequency processing technique uses the speaker and microphone built into the smart device, as described in Section 2. We use 19.0 kHz ~ 21.0 kHz as the trigger signal for sharing data, because most people can barely hear sounds in the 20 kHz ~ 22 kHz range. We use 19.0 kHz ~ 21.0 kHz as the real range

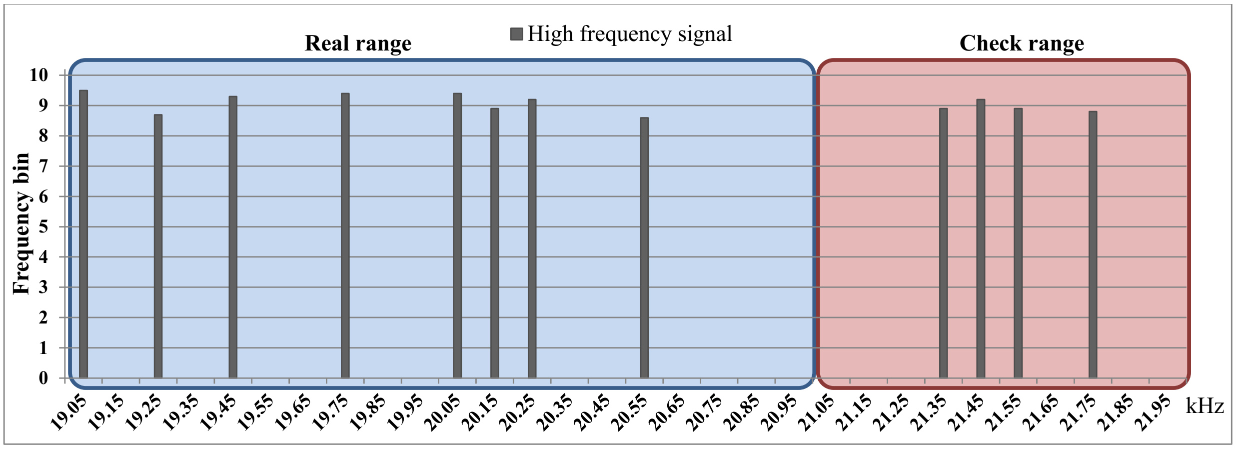

Fig. 2. Example of signal data expression using high frequencies

to send key values, and 21.0 kHz ~ 22.0 kHz as the check range to detect errors during data transmission. Data values using high frequencies are divided by 0.1 kHz. A data value expresses 1 when a relevant high frequency exists, and 0 when a relevant high frequency does not exist. Fig. 2 gives an example of signal data expression using high frequencies via the proposed method.

In Fig. 2, the real range shows 16 bits of data using the 19.0 kHz ~ 20.6 kHz range. The value of the real data is 10101001 00111001. The check range shows 8 bits of data using the 21.0 kHz ~ 21.9 kHz range. The value of the check sum is 00011101. Neither the real range nor the check range used in this example are fixed. If a user wants to send more data, the proposed method can support up to 40 bits using high frequencies within the 18 kHz ~ 22 kHz range.

High frequencies from the built-in speaker of the sending smart device are collected by the built-in microphone of the receiving smart device. We show the values of the collected high frequencies through a fast Fourier transform (FFT) in Fig. 3.

Fig. 3 shows the FFT result of the smart device receiving the high frequency signals shown in Fig. 2. If we check the frequency values above the threshold, we see that the value of the receiving data is 10101001 00111001 00011101. In this way, we can check that this is the same as the value of the sending data. The threshold in Fig. 3 is used to confirm the presence or absence of the relevant high frequency. We define the threshold at 50% of the maximum of the total high frequency signals received. Finally, when the receiving smart device analyzes the collected high frequencies, it confirms whether or not the data received is error-free through the 16 bits of real data and the 8 bits check sum.

The work flow of the data transmission between smart devices via the proposed method is shown in Fig. 4. In Fig. 4, we see that when the sending smart device emits high frequency signals, it sends GPS information and the key value of the high frequencies to the sharing server at the same time. When the receiving smart devices, which are located at close range, detect high frequencies from around sounds, each smart device sends GPS information and the key value of the detected high frequencies to the sharing server.

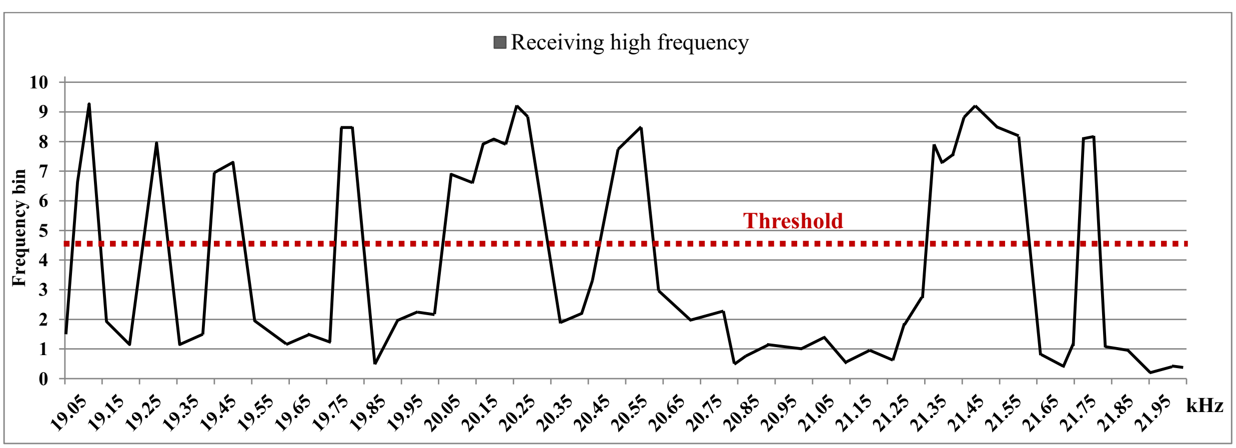

Fig. 3. Collecting high frequencies value through FFT

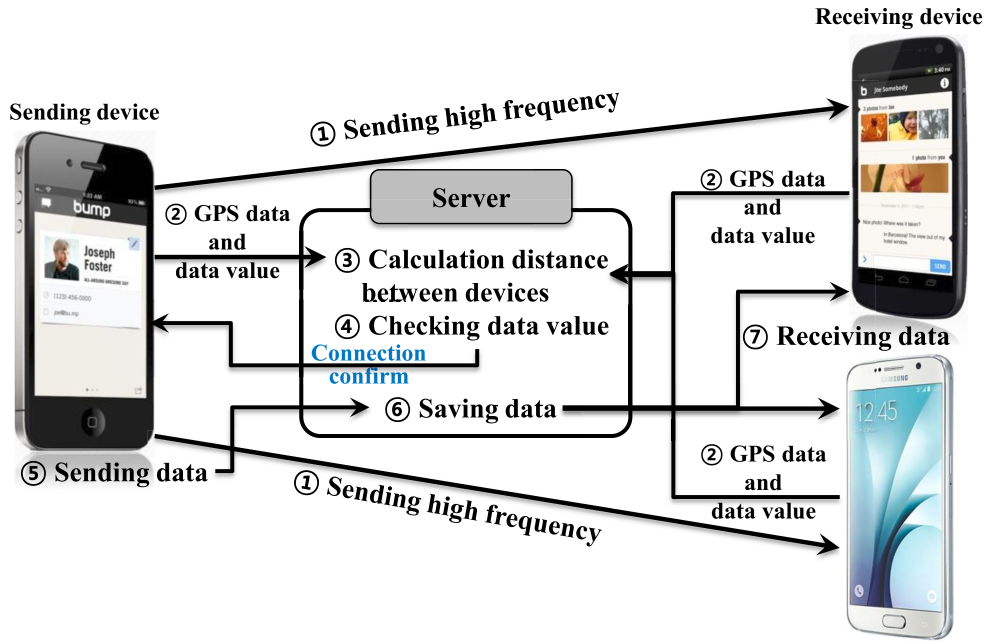

Fig. 4. One-to-many (1:N) data transmission work flow between smart devices

The server calculates the distance of all the smart devices using the GPS information and checks whether the receiving key value is the same. If the distance between the smart devices is close and the key value is equal, the server sends a message to each smart device, confirming the connection. The sending device then sends data to the server and the server shares the data with the receiving devices. The length of the high frequencies is shown in ***k*** seconds when the smart devices should be able to detect the high frequencies. The sampling rate to analyze the FFT of the receiving devices is 48,000 [19]. The sampling rate refers to the number of sound samples per second when smart devices record the sound. Most methods to date have used 48,000. We used 8192 (8K) as the sample size [20, 21]. The sharing server calculates distance using Euclidean distance [22], as in

$$d = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \quad (1)$$

Equation 1, ***d*** refers to the distance between the sending device and receiving devices; $\boldsymbol{p_i}$ and $\boldsymbol{q_i}$ represent the GPS information of each smart device. If the distance is under ***m*** (the threshold), the sharing server confirms a connection between the smart devices. This allows us to see that 1:N data transmission between smart devices is supported by the proposed method, using high frequencies and the work flow shown in Fig. 4.

IV. Experiment with and Analysis of 1:N Data Transmission

In this section, we describe the development of the application that enabled 1:N data transmission using the proposed method. We evaluate the performance of the application through the 1:N data transmission experiment. The main screen in the sending and receiving application is shown in Fig. 5 and Fig. 6.

Fig. 5. Main screen of iOS application

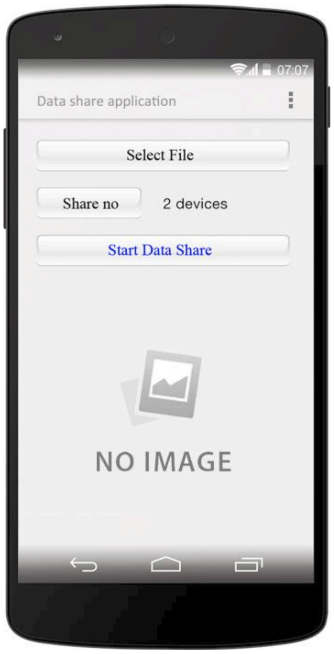

Fig. 6. Main screen of Android application

Fig. 7. Screen of the smart device receiving data

We developed two applications to work on iOS and Android. In Fig. 5 and Fig. 6, the user of the sending device chooses data sharing via the "Select File" button and sets the number of receiving devices via the "Share no" button. When the user touches the "Start Data Share" button, the sending device starts to share data. If the receiving devices detect high frequencies from the sending device, their main screens look the same as in Fig. 7. A progress bar allows users to check the status of the data transmission. Finally, when the data transmission ends, if the receiving data is an image or photograph, the screen shows the image received, as in Fig. 8. We then carried out one-to-two (1:2) and one-to-three (1:3) data transmission experiments with the developed application. We used iPhone 6 Plus as the sending device with iPhone 6 and Galaxy S5 as receiving devices in the 1:2 experiment. We used iPhone 6, Galaxy S5, and Nexus 6 in the 1:3 experiment. The sharing server was Intel Core i5-4690 with 8G RAM. The server's OS was based on Linux, Apache 1.3.41, PHP 5.2.6, and MySQL 5.0.51. The $\boldsymbol{k}$ length of high frequencies was 0.5 seconds and the threshold $\boldsymbol{m}$ was 10 m. We used a 1.7 Mb photo image, captured by iPhone 6 Plus, as the sharing image. We carried out 100 transmission attempts in each experiment. As we received errors with the first high frequencies, the sending device sent out the high frequencies twice. Table 1 shows the results of the 1:2 and 1:3 data transmission experiments.

TABLE I. RESULT OF 1:N DATA TRANSMISSION EXPERIMENTS

Device	1st receipt	2nd receipt	Device	1st receipt	2nd receipt
iPhone 6	96	4	iPhone 6	95	5
Galaxy S5	94	6	Galaxy S5	96	4
			Nexus 6	97	3

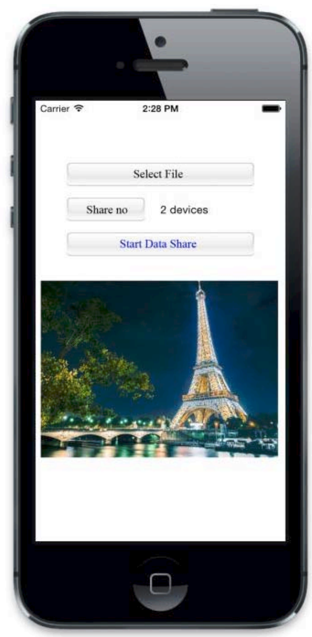

Fig. 8. Screen of the device after saving image data

In Table 1, the results show 96 and 94 counts for the first attempt in the 1:2 experiment and 100 counts for the second attempt. Thus, the success rates of the proposed method were 95% at the first attempt, and 100% at the second attempt. In the 1:3 experiment, the results showed 95, 96, and 97 counts for the first attempt and 100 counts for the second attempt. Thus, the success rates were 96% at the first attempt and 100% at the second attempt in the 1:3 data transmission test. This shows that some errors may occur in data transmission using the proposed method. We think this was because the ***k*** length of high frequencies was too short. Therefore, if we lengthen ***k*** or make the sending device send out the high frequencies twice, success rates can be increased.

In the next experiment, we compared the existing BUMP method with the proposed method. However, because the BUMP API has not been in service since 2014, we developed a BUMP server on the same computer which we used as the sharing server and we then developed an application which could work like the BUMP API. Because the BUMP API only supports 1:1 data transmission, we carried out 100 transmission attempts with the BUMP API and 100 attempts using the proposed method for the 1:1 data transmission test. Again, as we received some errors from the first high frequencies, we made the sending device send out the high frequencies twice, as in the previous experiment. Table 2 shows the results of the 1:1 experiment with the BUMP API and the proposed method.

TABLE II. THE RESULT OF THE COMPARATIVE EXPERIMENT WITH BUMP API

Transmission method	1st receipt	2nd receipt
BUMP API	88	10
Proposed method	97	3

In Table 2, we see that the BUMP API sent and received 88 counts at the first attempt and 10 counts at the second. Thus, the total success rate of the BUMP API was 98%, up to the second attempt. Sending and receiving data via the proposed new method produced 97 counts at the first attempt and 3 counts at the second. Thus, the total success rate of data transmission using the proposed method was 100%, up to the second attempt.

V. CONCLUSION

In this paper, the new method that we propose provides effective technology to support 1:N data transmission between smart devices, without any advance preparation. The proposed method resolves issues with existing data transmission methods, such as joining a specific application, Bluetooth pairing, or the need for the existence of an NFC module. Moreover, the new method, which uses the speakers and microphones built into smart devices, supports 1:N data transmission and can also support sending small amounts of data and user authentication between smart devices. It should be part of the Internet of things (IoT) and should have many applications in the communications field.

As future research, we intend to examine the ***k*** length of high frequencies and transmission errors in order to improve data transmission success rates. We will also explore signal processing technology for large amounts of data and information transmission methods that use only high frequencies.

REFERENCES

[1] H. Monson, H, "Bluetooth technology and implications," SysOpt, December 1999.

[2] C. Gomez, J. Oller, and J. Paradells, "Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology," Sensors, vol. 12, no. 9, pp. 11734-11753, 2012.

[3] Z. Antoniou and S. Varadan, "Intuitive mobile user interaction in smart spaces via NFC-enhanced devices," In Proceedings of the 3rd International Conference on Wireless and Mobile Communications, IEEE, pp. 86-86, Guadeloupe, March 2007.

[4] S.C. Alliance, "Proximity mobile payments: Leveraging NFC and the contactless financial payments infrastructure," Smart Card Alliance, September 2007.

[5] M. Roland, "Software card emulation in NFC-enabled mobile phones: great advantage or security nightmare," In Proceedings of the 4th International Workshop on Security and Privacy in Spontaneous Interaction and Mobile Phone Use, Newcastle, UK, June 2012.

[6] C. Eunjeong, "Kakaotalk, a mobile social platform pioneer," SERI Quarterly, vol. 6, no. 1, pp. 63-69, 2013.

[7] K. Church and R. Oliveira, "What's up with whatsapp?: comparing mobile instant messaging behaviors with traditional SMS," In Proceedings of the 15th International Conference on Human-computer Interaction with Mobile Devices and Services, pp. 352-361, Munich, Germany, August 2013.

[8] J.D. Rachid, "Global communication apps," Network Journal, vol. 21, no. 1, pp. 61, 2014.

[9] C. Christina, "The team behind Bump aims to crack photo sharing on smart phones," Fast Company, http://www.fastcodesign.com/1670380/the-team-behind-bump-aims-to-crack-photo-sharing-on-smart-phones

[10] Bump-api-ios, https://github.com/bumptech/bump-api-ios

[11] H. Matsuoka, Y. Nakashima, and T. Yoshimura, "Acoustic OFDM system and performance analysis," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol. 91, no. 7, pp. 1652-1658, 2008.

[12] N. Cvejic, "Digital audio watermarking techniques and technologies: applications and benchmarks: applications and benchmarks," IGI Global.

[13] J.S. Noh and K.H. Rhee, "High quality audio watermarking using spread spectrum and psychoacsoutic model," The Institute of Electronics of Engineers of Korea Signal Processing, vol. 43, no. 5, pp. 48-56, 2006.

[14] Y. Suzuki, R. Nishimura, and H. Tao, "Audio watermark enhanced by LDPC coding for air transmission," In Proceedings of International Conference on Intelligent Information Hiding and Multimedia Signal Processing, pp. 23-26, Pasadena, USA, December 2006.

[15] V. Filonenko, C. Cullen, and J.D. Carswell, "Indoor positioning for smartphones using asynchronous ultrasound trilateration," ISPRS International Journal of Geo-Information, vol. 2, no. 3, pp. 598-620, 2013.

[16] 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.

[17] J.B. Kim, J.E. Song, and M.K. Lee, "Authentication of a smart phone user using audio frequency analysis," Journal of the Korea Institute of Information Security and Cryptology, vol. 22, no. 2, pp. 327-336, 2012.

[18] M.B. Chung and H.S. Choo, "Near wireless-control technology between smart devices using inaudible high-frequencies," Multimedia Tools and Applications, vol. 74, no. 15, pp. 5955-5971, 2015.

[19] R.J.I. Marks, "Advanced topics in Shannon sampling and interpolation theory," Springer Science & Business Media, 2012.

[20] E. Bidet, D. Castelain, C. Joanblanq, and P. Senn, "A fast single-chip implementation of 8192 complex point FFT," IEEE Journal of Solid-State Circuits, vol. 30, no. 3, pp. 300-305, 1995.

[21] R.M. Jiang, "An area-efficient FFT architecture for OFDM digital video broadcasting," IEEE Transactions on Consumer Electronics, vol. 53, no. 4, pp. 1322-1326, 2007.

[22] H. Yoon, Y. Zheng, X. Xie, and W. Woo, "Smart itinerary recommendation based on user-generated GPS trajectories," In Ubiquitous Intelligence and Computing, pp. 19-34, Springer Berlin Heidelberg, 2010.