VOL. 10, NO. 2, FEBRUARY 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

# P2P FILE SHARING APP FOR ANDROID DEVICES SPORTING IEEE 802.15.4 ZIGBEE RADIOS

Akmal Nurhananie Abd Rahman[1], Mohamed Hadi Habaebi[1] and Mahamod Ismail[2]
[1]Department of Electrical and Computer Engineering, International Islamic University Malaysia, Kuala Lumpur, Malaysia
[2]Department of Electrical, Electronics and System Engineering, National University of Malaysia, UKM, Bangi, Selangor, Malaysia
E-Mail: akmalnurhananie@gmail.com

## ABSTRACT

The peer-to-peer (P2P) file sharing protocol has evolved from depending on Internet connection provided by the cellular network, to free Internet connectivity through Wi-Fi technology. On the other hand, IEEE 802.15.4 is considered as today's one of the top growing wireless protocol. This paper introduces ZigBee technology, which falls under this standard, into the mobile P2P environment, where it involves devices with Android 4.0 (Ice Cream Sandwich), which is the lowest version of the Android platform that can be found among mobile devices. This paper integrates ZigBee into Android P2P environment and preliminary results showed that it is possible to send and receive files over ZigBee radio, yet the time taken for complete file transfer depends on the file size.

**Keywords:** peer-to-peer, zigbee, android, message broadcasting, file sharing.

## INTRODUCTION

Peer-to-peer (P2P) is a very popular type of network communication for file sharing, where the data transfer is faster as it does not need to pass through the services of a server to share files and data. For the time being, the successful P2P applications have been running on either the wired or wireless Internet connection, as well as over PAN network such as Bluetooth and Near-Field Communication (NFC). At present, peer-to-peer (P2P) communication is considered as a problem by both the wired and wireless Internet providers as it generates a large amount of traffic. This problem is substantial for mobile networks, as they not only have to serve users of mobile devices, but also the mobile subscribers that prefer mobility over traffic.

Android operating system (OS), built and designed by Google, is an open-source platform that supports Java programming. It is currently available on a wide variety of mobile devices. Its open-nature allows manufacturers and wireless providers to customize the software, creating diversity among the devices. Besides that, almost anyone can create an application to run on Android, and all of these applications can be easily obtained from the Android Market and other download markets. Android has no restrictive policies about development tools that are used to create the applications. Most of the Android devices that can be found in the market nowadays are preinstalled with either Android 4.0.3, also known as Ice Cream Sandwich (ICS), Android 4.1.x and Android 4.2.x, which are known as Jellybean, or the latest platform version is Android 4.4 (KitKat).

IEEE 802.15.4 radio technology was introduced in 2003, and is considered as today's one of the top growing wireless protocol. The technology is designed focusing more on energy-saving. The main reason for its development is to provide simple devices with a reliable, robust wireless technology that could run for a very long time, even years, on standard primary batteries. ZigBee® standard is built on top of the IEEE 802.15.4's radio layer to specify the network, security and application layers. It can be seen that ZigBee is widely used in Home Automation and Wireless Sensor Network (WSN) and the technology is widening its usage into the health and communication industry, as shown in Figure-1.

On February $14^{th}$, 2011, Texas Instruments Incorporated (TI Inc.) had introduced an Android 2.2 software development platform which is designed to simplify the mobile device integration of the ZigBee and ZigBee RF4CE protocol stacks. Being introduced for the first time in the industry, users are enabled to manage commands on smart TV or other utilities through mobile devices [1]. This achievement leads to maximization of the mobile devices' functionality and mobility for the users.


**Figure-1.** ZigBee applications in various industry.

Hence, with the enlightenment of the above situation, this research is conducted in a similar direction, yet it will be converging more towards development of the application for message broadcasting and file sharing through peer-to-peer (P2P). The project is focused on using ZigBee-based technology. The development of mobile platforms is very rapid and significant lately, especially for Android, where it is even faster than the development of hardware devices that enables the new data applications. Therefore, the project involved mobile devices of Android ICS and Jellybean platforms and ZigBee motes in the form of USB dongles. The dongles are connected directly to the devices via USB OTG cable, enabling data flow from the device, and data transmission over ZigBee radio.

This paper presents the application development conducted for Android devices and the current results achieved. Brief introduction on Android devices and ZigBee Technology is presented in Section II. The research mainly focuses on Android 4.0 based devices and beyond for its vast availability in the market, and on ZigBee for its rapid growth of usage in various industry. The system design and prototype development of the designed Android application (app), named ZigBeeComm is presented in Section III in details, while the preliminary results of the evaluation conducted on the designed app is exhibited in Section IV, where two different activities are devised in the app, namely File Sharing/File Receive and Broadcast Message activities. Following that is the conclusion of this paper in Section V.

## P2P IN MOBILE ENVIRONMENT

Users nowadays have the desire for their mobile devices to work as similar as their desktop. In addition to that, the numbers of applications that can run on mobile devices of various mobile platforms keeps on increasing, instigating that the application of P2P to be highly significant to users in terms of communication and collaboration, distributed computing, and file/content sharing. Hence, a number of mobile P2P protocol had been introduced in solving this problem, such as Napster, Gnutella, eDonkey, BT, etc.[2]

In the beginning, the only way for a mobile phone to connect to the Internet is via subscribed cellular radio services. Consequently, the data traffic of both subscribers of standard cellular services and subscribers of Internet users' overloads, forcing the service providers to charge different rate for Internet connection. Therefore, the subscribers incurred higher data charges. Until recently, commercial smart phones are incorporated with Wi-Fi system in order to enable them to connect to the Internet without extra charges.

On top of that, there are various P2P communication technologies that are introduced into the mobile communication. Some are developed for specific mobile platforms, such as the SymTorrent [3] and Symella[4] mobile P2P applications, while some others are cross platform. Besides that, several researches had also been conducted in either combining Wi-Fi technology with IEEE 802.15 technologies, such as Bluetooth, ZigBee, RFID, etc. or creating a new P2P network that does not have to rely on Internet connectivity, i.e. P2P communication between mobile devices relying on IEEE 802.15 only. It utilizes the wireless technology that enables the mobile devices users to share file without going through the Internet backbone, for instance on using the Bluetooth technology that is integrated in almost all mobile devices.

For instance, P2PBluetooth[5] is essentially a mobile-to-mobile application that enables file sharing among mobile phones that exploits the Bluetooth technology installed in mobile devices. Every mobile device publishes its own list of available files and directly requests the desired data or files by the user to mobile phones nearby by using Bluetooth connectivity. Downloads are requested by specifying the keywords. Once running, P2PBluetooth automatically initiates a search by querying other Bluetooth enabled devices in proximity for their list of available contents. When an answer is received, the data of interest is automatically transferred. All the process of file transferring is completed without user mediation. The user only need to specify the data identifiers needed for the download.

Another P2P application that utilizes Bluetooth technology is known as BlueTorrent [6]. It is a cooperative P2P file sharing application that enables its users to share audio/video contents as they are in mobile. This application achieves a greater performance improvement in download time compared to the traditional AP only mode. Moreover, it allows random peers to connect to each other. In spite of the short link duration, BlueTorrent application uses BitTorrent-like file swarming in order to effectively share the contents. Content is divided into a number of small pieces, and mobile users can exchange whatever pieces are available. BlueTorrent uses a cooperative carry and forward strategy, where pieces are forwarded whenever a connection is available in minimizing download latency. Whenever a connection is available, peers first exchange their bitmaps to find out missing pieces through simple bit operations. The prerequisite of content distribution is to know where the content is. The availability of the contents that are desired can be searchable through indices which include a unique ID, title, producer, media type, etc. Users who are interested in specific information can proactively query other peers.

Wi-Zi cloud [7] is a system that utilizes a dual WiFi-ZigBee radio on mobile phones and Access Points (AP), which adopts the designed WiZi-cloud protocols. The goal of this system is to achieve ubiquitous connectivity, high energy efficiency, and real-time intra-device/inter-AP handover, while being transparent to the application, where it switches seamlessly between WiFi and ZigBee links depending on the traffic load. This research has proven to be enhancing the energy consumption of a mobile phone, as both of the WiFi technology and ZigBee technology complement each other in terms of low-power requirement and large bandwidth requirement respectively.

## SYSTEM DESIGN AND PROTOTYPE

This project will involve only the application layer of the architecture, as shown in Figure-2. The API involve only mobile devices with Android 4.0 software and beyond, while the security and network layer will be as per provided by the ZigBee dongles.

VOL. 10, NO. 2, FEBRUARY 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

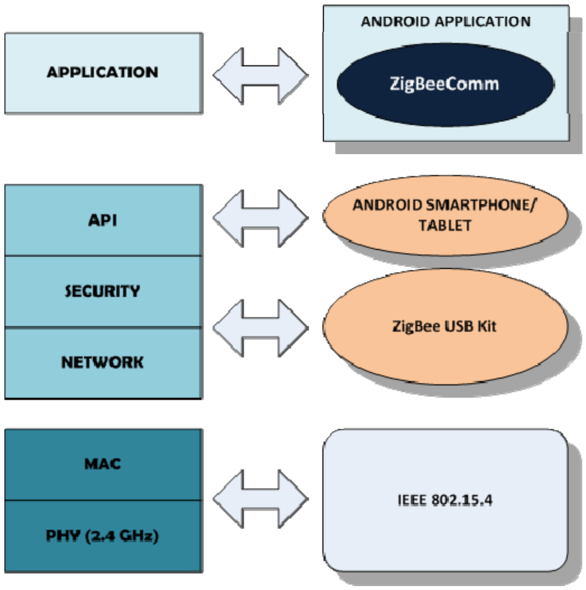


**Figure-2.** Protocol layer overview.

**Hardware**

The ZigBee dongles used are manufactured by SZHOMA, which is the ZigBee USB Gateway HCV105. It is a low cost wireless transceiver product that work in the 2.4 GHz ISM frequency band, based on the IEEE 802.15.4 protocol development. The dongle is mainly used in monitoring and control, as well as data acquisition and transmission. This converter provides a USB serial interface, enabling it to be used with computer equipment. For it to be able to be used with Android devices, the device itself needs to be preinstalled with PL2303 driver, where it is included in the application package file (.apk). The performance parameters of the dongle are as shown in Table-1. The ZigBee dongle is then connected to the Android device via USB connection, where a USB OTG cable is needed.

**Table-1.** Performance parameters.

| **SZHOMA ZigBee USB gateway HCV105** | |
|---|---|
| **Parameters** | **Details** |
| **Input voltage** | **USB 5V input** |
| Baudrate | 9600 BPS (default), can be installed 19200 BPS, 38400 BPS, 57600 BPS, 115200 BPS |
| Radio frequency | 2.4 GHz |
| Wireless protocol | ZigBee 2007 / PRO |
| Transmission distance | LOS of 100 meters |
| Emission current | 280 mA (maximum) |
| Receiver sensitivity | -97 dBm |
| Temperature range | 40ºC to 85ºC |

**Software**

ZigBeeComm is an Android application developed through Eclipse SDK software, which operates with "pure" Java code and has no direct dependence on the underlying operating system; hence the chief dependence is on the Java Platform itself. This apps is designed to detect the USB dongles and enables the Android devices to communicate with each other through ZigBee. As mentioned previously, the Android devices require a specific serial driver, namely Prolific PL2303 driver, in order to communicate with the ZigBee transceiver that is being used. The javascript for the driver is available from the Google Code [8], and can be downloaded and implemented directly into the ZigBeeComm application package file.

The flow of the application is shown in Figure-3. The Android device first detects the USB dongles that is plugged onto it and prompt the user to start the apps. Upon launching the apps, the user is given the choice to start the File Sharing activity, File Receive activity or Message Broadcast activity. In each of the activity, the data are sent and received via the connected ZigBee dongle, which then will communicate with another party over the ZigBee radio.

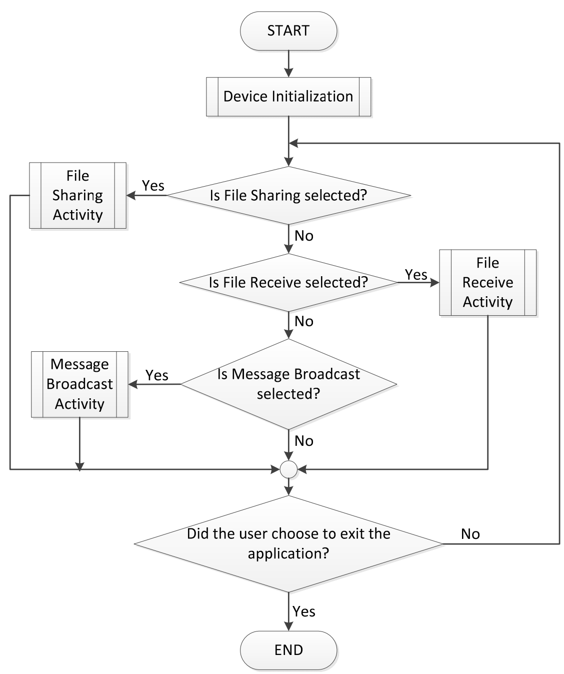


**Figure-3.** Flowchart of ZigBeeComm app on Android.

**ZIGBEECOMM**

Some of the preliminary results are presented in this section, mainly the screen-captures of the application window. As stated in the previous section, the user will be prompted to start the ZigBeeComm application as the Android device detects the connected USB dongle. Upon launching the application, the main window will be displayed as in Figure-4 and the user can decide which

VOL. 10, NO. 2, FEBRUARY 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences



activity to start to communicate with other Android devices, given that the other parties are running the same application. Figure-5 shows the initialization process of the USB dongle on the Android device, where connections are established between the dongle and the device via the USB endpoints.


**Figure-4.** ZigBeeComm application window.

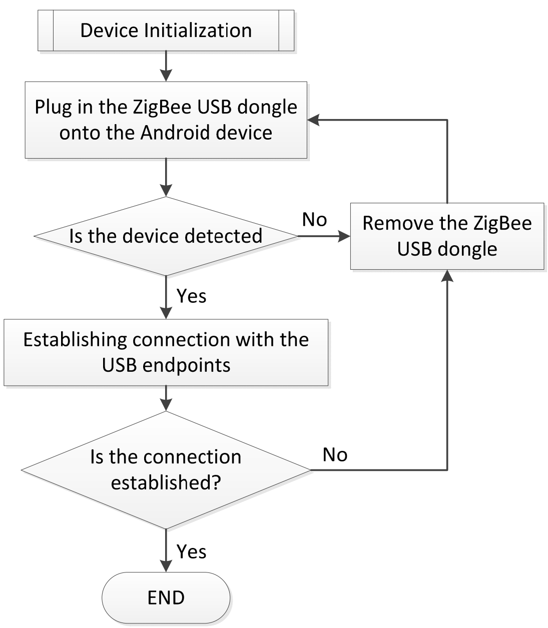


**Figure-5.** Flowchart of device initialization in ZigBeeComm apps on Android.

**File Sharing and File Receive**

The File Sharing part of the application is where the users are enabled to browse the files and folders that are stored within the Android device's internal and external memory. The selected file is then converted into byte array, and segmented into 64-byte length of byte streams, before sending them to the serial port of the USB dongle. This is because the endpoint buffer size of the dongle for writing output (write-out) and for reading input (read-in) is of 64-byte respectively. The same procedure is applied to the receiving part of the application, where the received file will be in the form of byte array and then converted into file format as specified by the user.

For file sharing activity, the user that wants to share the file can click the File Sharing button, and chooses the file to be shared, whereas the user that wants to receive the file must run the File Receive activity and be prepare to receive the incoming file transfer. The flowchart in Figure-6 and Figure-7, summarizes the File Sharing and File Receive activity respectively.

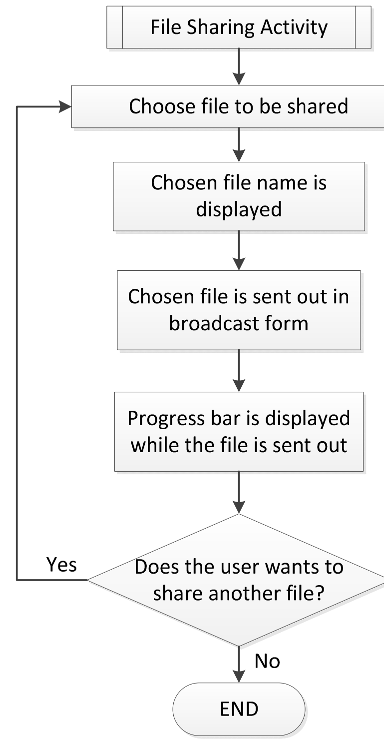


**Figure-6.** Flowchart of file sharing activity.

A total of four tests had been conducted on this activity. All these tests involve a total of seven different text file of size 100 bytes, 500 bytes, 1000 bytes, 5000 bytes, 10 kilobytes, 50 kilobytes and 100 kilobytes. The tests were done in four different situations for three and four communicating Android devices, where one of the devices will act as the transmitting device, running the File Sharing activity, while the rest is running the File Receive activity simultaneously. First is where three devices are within close proximity, whilst the second test was where the devices are placed 10 meters apart from each other. These two tests are then repeated for a number of four devices. The recorded time of the file transfer for each file size with respect to each test condition was plotted, where additional 100, 200 and 300 seconds are added to the second, third and fourth series for presentation purposes, as depicted in Figure-9.

VOL. 10, NO. 2, FEBRUARY 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

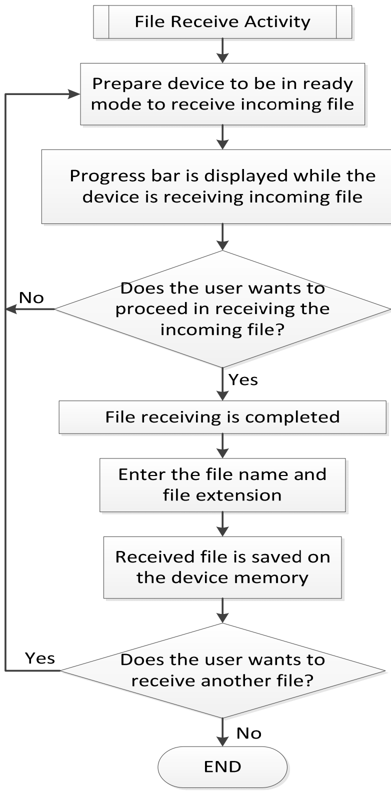


**Figure-7.** Flowchart of file receive activity.

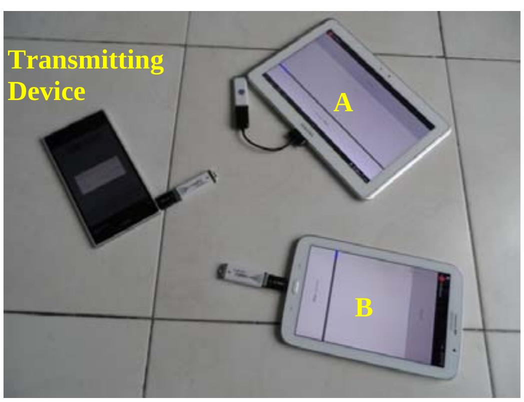


**Figure-8.** The transmitting device and devices A and B on close proximity on file sharing activity

For each condition, the time taken for the respective file size was fully transferred, was recorded and plotted into four different graphs. All four plotted graphs shows the same result and pattern, which means that the file transfer activity over ZigBee radios can be achieved, yet it took longer time for large file sizes to be fully transferred.

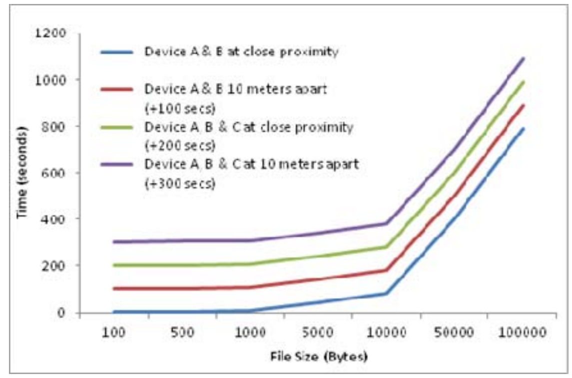


**Figure-9.** Graphs of time taken for file transfer to be completed with respect to different file sizes.

Another test had also been conducted, where the device is tested to transfer an image file (.png). For this test, only two devices are involved. The following Figure-10 shows the transmitted image from device A and the received image on device B. It can be seen that the received image file is corrupted. This is because some of the packets transferred are lost during the transmission process.

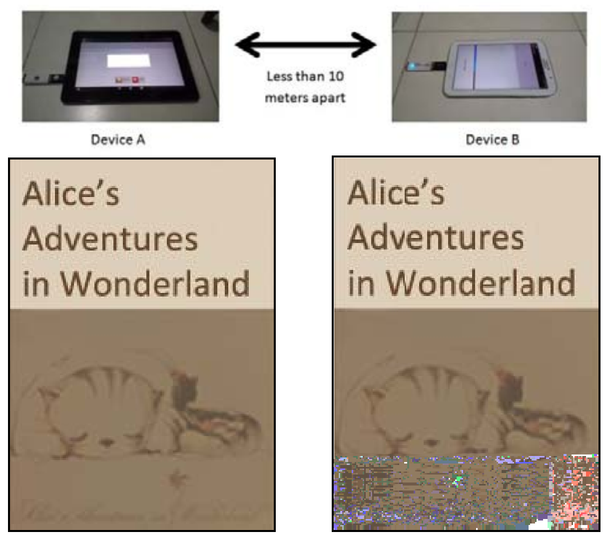


**Figure-10.** The sent image from device A and received image on device B.

**Broadcast message**

When the user selects the Broadcast Message activity, a new window will open where the user has to turn on the connect button in order to begin broadcasting message. The following Figure-11 shows the screen-capture of the Broadcast Message activity. First and foremost, identical as File Sharing and File Receive activity, connection is established between the device and the USB dongle endpoints, then only will the user be enabled to send messages to all Android devices that have ZigBeeComm application actively running on it. Hence

ARPN Journal of Engineering and Applied Sciences



the communication is done over ZigBee radios. The flowchart in Figure-12 summarizes this activity.

**Figure-11.** ZigBeeComm broadcast message window: (a) Connection is established with the USB endpoints. (b) Message is broadcasted and (c) received by other devices.

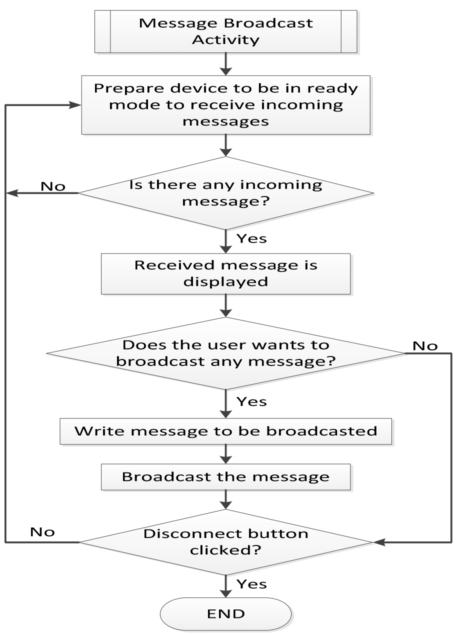


**Figure-12.** Flowchart of the message broadcast activity.

Similar as the tests condition in File Sharing and File Receive activity, the Broadcast Message is placed under test with three to four devices communicating with each other. As a result, all devices can communicate without any problems, where each device successfully receive and send messages to other devices.

## CONCLUSIONS

Based on this research, the aim of integrating ZigBee with Android device is achieved. It has also proven that the ZigBee dongles can be used with an Android device, given that the Android device is preinstalled with the required driver. There are different types of driver available for USB devices, such as Prolific PL2303 driver and FTDI FT232R, just to name a few. Besides that, it is proven that the file transferred between two Android devices is successful. Yet, due to the low-data rate character of ZigBee, larger file size is taking longer time to be fully transferred to the receiving device. More test will be conducted, to see the time response in file sharing activity with at a distance of more than 10 meters apart, as well as testing with different type of files, such as video and audio files.

## ACKNOWLEDGEMENTS

This work is supported by the Ministry of Higher Education (MoHE) of Malaysia, under the Ensemble Mobile Platform for Innovation Digital Economy Environment - Enhancing Protocol for Peer-to-Peer and Peer-to-Multipoint Communications Project (LRGS/TD/2011/UKM/ICT/02).

## REFERENCES

[1] TI demonstrates world's first Android development platform bringing ZigBee and ZigBee RF4CE to Smartphones and Tablets. [Online] Available: http://newscenter.ti.com/Blogs/newsroom/archive/2011/02/14/ti-demonstrates-world-s-first-android-development-platform-bringing-zigbee-174-and-zigbee-rf4ce-to-smartphones-and-tablets-620349.aspx.

[2] Y. Sun, X. Wen, Z.Zhao and Y. Li. 2010. Mobile P2P content distribution in wireless networks environment. International Conference on E-Business and E-Government.

[3] Kelenyi I., B. Forstner and B. Molnar. 2006. Symella. [Online] Available: http://amorg.aut.bme.hu/projects/symella.

[4] Kelenyi I., P. Ekeler and Z. Pszota. 2006. SymTorrent. [Online] Available: http://amorg.aut.bme.hu/projects/symtorrent.

[5] C.E. Palazzi, Armir Bujari and E. Cervi. 2009. P2P file sharing on mobile phones: Design and implementation of a prototype. Proc. of 2nd IEEE International Conference on Computer Science and Information Technology, ICCSIT. pp. 136-140.

[6] S. Jung, U. Lee, A. Chang, D. Cho and M. Gerla. 2007. Blue Torrent: cooperative content sharing for Bluetooth users. Proc. of the 5th Annual IEEE International Conference on Pervasive Computing and Communications (PerComm '07). pp. 47-56.

[7] Tao Jin, G. Noubir and Bo Sheng. 2011. WiZi-Cloud: Application-transparent dual ZigBee-WiFi radios for low power Internet access. Proc. of IEEE INFOCOM. pp. 1593-1601, April.

[8] PL2303driver.java. Retreived from https://code.google.com/p/indiserver/downloads/list.