Proceeding of IC-ITS 2015 e-ISBN:978-967-0850-07-8
International Conference on Information Technology & Society
8-9 June 2015, Kuala Lumpur, MALAYSIA

Multicast File and Screen Sharing using Wi-Fi Connectivity

Marylene Saldon-Eder

Minndanao University of Science and Technology
Lapasan Cagayan de Oro City, Philippines

mseder@must.edu.ph

Abstract. In this study, File Sharing is the act of distributing files such as music, movies, documents, photos, or games to other device over the internet. Screen Sharing enables user to share desktop content from our mobile to other people's device. File and screen sharing mobile application is designed and developed in order to address the issues regarding slow process, vulnerable to data corruption, and inability to broadcast file. It can be accessed to any devices such as Android, IOS, and Windows. By using the app, time and effort can be saved in transferring file using Wi-Fi since Wi-Fi can transfer over long distances between two devices. Based on thorough analysis and results, it is concluded that the application improved for a better and simple way to share files and screen to any device. Using the application, user can now share files with a faster rate, protect data from corruptions, and capable to broadcast file.

Keywords: Bandwidth, File Sharing, Screen Sharing, Ad hoc Network

INTRODUCTION

File Transfer is the transfer of files from PC to PC over a network using any of several different software packages and methods. Bypassing your computer network and moving files using USB keys is also an option. File transfer is a popular method for swapping large files on the Internet, particularly music and videos. Unlike FTP, most P2P file sharing systems do not use any central servers but instead allow all computers on the network to function both as a client and a server [7]. A peer-to-peer (p2p) network allows computer hardware and software to function without the need for special server devices. On the other hand, file sharing is the practice of distributing or providing access to digitally stored information, such as computer programs, multi-media (audio, video) resources, documents, or electronic books. A peer-to-peer (p2p) network provides a scalable and fault-tolerant mechanism to locate nodes anywhere on a network without maintaining large amount of routing state. This allows for a variety of applications beyond simple file sharing, including in multicast systems, anonymous communications systems, and web caches. This work surveyed security issues that occur in the underlying p2p systems, as well as legal and trust issues that occur in file sharing and p2p networking. It also, analyses different p2p software and their features. This work has established that file sharing and peer-to-peer networking generates a lot of traffic on the

network and is associated with malicious software [1]. Peer-to-peer file sharing is relatively new in the growing list of mobile-based technologies, making it the focus of interest in the world of information technology in the last couple of years. In a P2P networking model, each of the computers on the network could act as both servers and clients thus distributing the network load and overhead across all the devices on the network, forming a network of individual peers with ability to search the network for shared resources. The characteristics of P2P architecture, such as reliability, easy one-to-one communication, and extensible distribution of resources, make it one of the most suitable networking technologies for mobile system [1]. Peer to peer networking is common on small local area networks (LANs) particularly home networks. Both wired and wireless home networks can be configured as peer to peer environments.

A study conducted by Students of College of Engineering in Bangladesh University of Engineering and Technology in 2007 stated that with more and more devices being made Wi-Fi enabled including cell phones, having ad-hoc networks available is becoming easier day by day. When communication between groups is required to be secured like covert teams avoidance of public network is necessary. An ad hoc network consists of one or more computers that are interconnected, but doesn.t need a centralized access point, such as a router, to transfer data back and forth. Ad hoc networks are ideal for sharing documents or other files with employees in the workplace, because the network is temporary and requires only a wireless adapter. Such networks are also ideal if your router stops broadcasting, because you can still connect to other computers and access files despite the missing Wi-Fi signal. In sharing information stored on your computer with other people nearby and everyone's computer has a wireless network adapter, a simple method of sharing is to set up an ad hoc wireless network. In spite of the fact that members must be within 30 feet of each other, this type of network presents a lot of possibilities. For example, you might consider establishing an ad hoc network at a meeting of mobile computer users so that you can share information with other attendees on their own screens rather than an overhead projector.

File and Screen Sharing is a technology that enhances classroom management strategies by easily assessing and addressing multiple learning styles, creating guided pathways to learning, and providing channels to individualize access to content and communication. On-going assessments, targeted learning experiences, and regular communication are core components of differentiated instruction. To understand what a child knows at a given point in time to inform daily instruction, the teacher requires technology-based tools that can assess and analyse student skills and competencies on-demand. Likewise, teachers require the ability to identify and then channel targeted resources to individual students to support multiple levels of learning and presents challenges that are appropriate for each student [6].

LanSchool provides precisely the right collection of 21st century learning management tools to support today.s teacher with organizing and structuring the wealth of digital resources within the learning environment. These tools enable the teacher to keep students on task, promote student-to-teacher conversations, and encourage cooperation and collaboration with their peers. The easy to use and configure LanSchool console, along with the web control and limiting features, provide the teacher with a quick snapshot of what each student is doing within their pre-established learning arena.

The screen sharing controls allow the teacher to share his/her screen or an individual student screen with the rest of the class for discussion, critique, brainstorming, and problem solving. Additional features of LanSchool technologies are it can share files and distributed to the students screen. Students can also download the shared files. Features that address security, power monitoring, and wireless support all help to ensure that the technology rich classroom will function to its maximum potential with minimal technical problems. Teachers grow in their confidence that the resources are used appropriately by students and function properly on a daily basis [6].

Another technology called NetSupport Technology is the state of the art software solution for optimizing training in networked classrooms. It gives teachers the ability to interact with their students on a one to one basis, to a pre-defined group or the overall class. Combining advanced multiple PC monitoring with an innovative customized test designer and the ability to create automated lesson plans, this latest version meets the challenge and requirements of today.s modern classroom [8]. Transferring large files over the internet has never been a simple task. Anyone that has ever tried to transfer a file larger than 100MB can vouch for the slow transfer speeds, multiple disconnects, data corruption, complexity of the task and security issues surrounding File Transfer Protocol (FTP) [3]. The same problem exists for web developers trying to implement web-based file transfer functionality. Ideally, the end user should be able to upload or download large files without leaving the web browser to open an FTP client. Most web developers also require some kind of mechanism that will notify their front end (HTML/JavaScript) or backend (PHP, JSP, ASP) code when a transfer event (complete, cancelled, paused, error) happens, as well as which files were transferred.

This study aims to address issues that are surrounding the difficulties in transferring files between devices due to its traditional way of transferring file which is the send and receive process, the user have to confirm the transferred file in able to begin the process. Also, users who are unable to found wireless connection in an area, has an alternative way to create an ad hoc network to provide a peer-to-peer connection. These users can organize themselves in an ad hoc connection as they communicate, collaborate and share with each other to complete the tasks at hand (e.g. file sharing). So, we came up with a proposal which aims to create a Multicast File and Screen Sharing using Wi-Fi Connectivity for using mobile devices, particularly to those who have difficulties in sending and receiving functions between devices and for professors who have difficulties in conducting a class without any resources such as projector and PCs.

METHODOLOGY

Designing a Mobile Application for File and Screen Sharing

Analysis

Through thorough analysis of the existing studies presented by many authors about the file and screen sharing, the researchers came up with the design not using the traditional send and receive process; instead the researchers used the file sharing system

where the user broadcast the shared file to facilitate the process and data corruptions in the middle of sending. The Peer on the other hand was dependent in downloading the shared file. Graphical models were used in the study to simplify how the process was being realized.

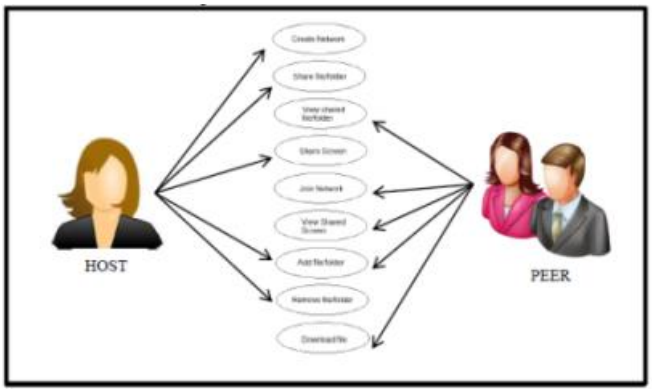

FIGURE 1. Use-Case Diagram of the Application

The use-case diagram that composed of the following actors: Host and Peer. The Host will be the one creating a network, Share a File/Folder, Share Screen, Add File/Folder, and Remove the added files. The Peer has only limited access on the application where he can only view the shared files/folder and screen. He can also download the shared file/folder.

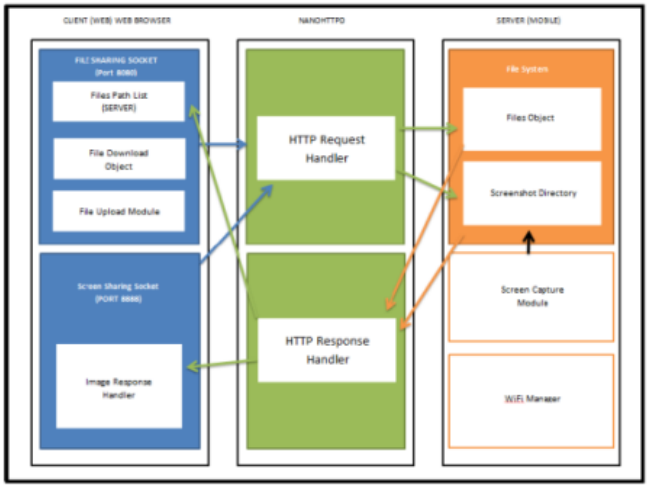

FIGURE 2. System Architecture of the application

The system architecture shows how the application shares file and screen. First, the server will create a hotspot connection if it is not connected to any network and the client must be on the same network. In file sharing, the server picks a file to share and open a port from its IP address. Then, the client requests for the shared file from the given url of the server and the server response. When downloading the file, the client requests the file to the server and the server sends the file to the client with HTTP POST.

In Screen Sharing, the server will open a different port. Then, the client will request from the server, upon the request of the client it takes a screen capture and save it to the phone storage. After saving the image, it sends the image back to the client where the request came from and the client will request for its screen every second as it sets.

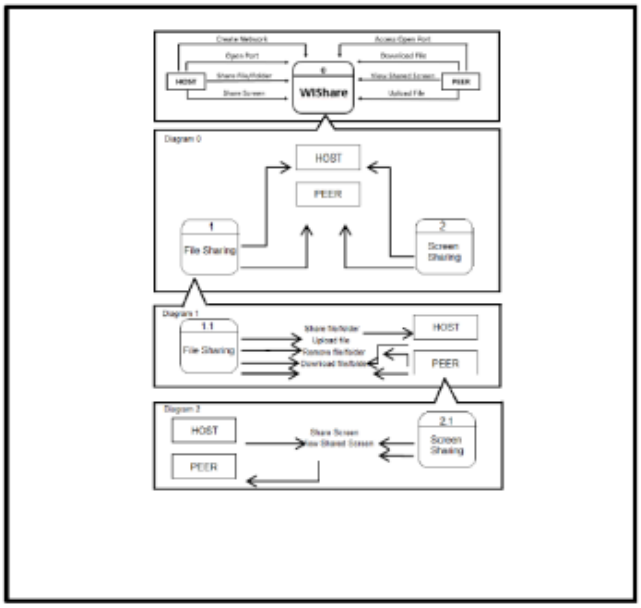

FIGURE 3. Layers of Dataflow Diagram

The figure shown above is the user interaction with the application. The diagram comprises two actors which represent the different user rights; Host and Peer. Host role applies to administrators who can only create network and Peer role may refer as the receiver. The Host can share files and remove shared files, while Peer can download the shared files and upload files. In terms of sharing screen, Host be the only one allowed to share screen and Peer be the one view the screen.

Algorithms in File Sharing

Step 1: Choose File to Shared
The shared file path will become <a href="shared/file/path">filename<a> in index.html file.
Step 2: Share File
the server open a port (8080) from its IP address.
eg. 192.168.43.1:8080
Step 3: Server Response
URI = filepath;
eg. 192.168.43.1:8080/, URI is equal "/"
Client Request URI **Server Response**/ = index.html
URI is a file = gets the file from its filesysem and sends file with HTTP POST
URI is a folder = it will list all files path and each file path will become <a href="shared/file/path">filename<a> in index.html file. Then, serve the index.html file.

Algorithms in Screen Sharing

```
Step 1: Share Screen
        the server open a port (8888) from its IP address.
Step 2: Server Response if there is a client connected.
        Loop{
        Screen Capture
        Save File
        Get File
        Send File to the client
        delay 1 second
        }
```

Development of the Application

Software Specification

a. Android Studio - Android Studio is the official IDE for Android application development which allows creating mobile application with Javascript. In the proposed study, Android Studio was used in order to make android application in easier way. Thus, Android studio is based on IntelliJ which is surely be able to run pure java code with sets of API to be implemented.
b. Android SDK - Android SDK is a set of software development tools that allows the creation of mobile applications for a certain software package, software framework, and etc. It may be as simple as something of one or more APIs in the form of some libraries to interface to a particular programming language.
c. Javascript - Javascript is a dynamic programming language commonly used as part of Web browsers, whose implementation allows client-side scripts to interact with the user, control the browser, and alter the document content that is displayed. Javascript was used in the program to be able to communicate to the server and purely browser-based file sharing which the file sharing still go to the server over WebSockets.
d. Bootstrap - Bootstrap is the most popular software in designing by the use of HTML, CSS, and JS Framework for developing responsive applications in both Web and Mobile. It is made for folks of all skill levels, devices of all shapes, and projects in all sizes. Bootstrap is very helpful in designing part of the program. It has a responsive design all over the platform used such as Android, Windows, and IOS.

Hardware Specification

Android users must have versions such as Android 2.3 Gingerbread, Android 3.0 Honeycomb, Android 4.0 Ice cream sandwich, Android 4.1-4.3 Jelly Bean, and Android 4.4.1- 4.4.2 Kitkat. Device must have atleast 350 MB of non-volatile storage available for application private data. Android can boot and run in configurations with less memory such as 256 MB RAM but it is less recommended to users if you are using third-party applications. As of November 2013, Android 4.4 recommends at least 512 MB of RAM while lowest RAM devices must have 340 MB. Moreover, the average of the apps requires minimum of 800MHz processor or better but it is highly recommended to have minimum of a dual core 1GHz processor. Devices must have screen sizes of at least 426

dp x 320 dp minimum and maximum of 960 dp x 720 dp. Android 4.0 device implementation should include support for one or more forms of 802.11. But if a device implementation does not include support for 802.11, it must implement the corresponding Android API. Screen sharing must have 1GB of RAM, 1.4GHz quad-core of processor, 16GB of internal memory, and should be rooted. These features were based during the testing part of the researchers. Android 2.3 Gingerbread to Android
4.4.2 Kitkat is compatible with the application. It is better to have high specifications like having 1GB of RAM and 1GHz of processor in order to have high standards and fast performance in sharing a file and screen.

```
public void createHotspot(String type){
    wifiApManager.setWifiApEnabled(wifiConfig(), true);
    loadingdialog = ProgressDialog.show(getActivity(), "Creating Hotspot...", "Please Wait", true);
    hotspotCreation = new Thread(new HotspotListener(type));
    hotspotCreation.start();
}
```

FIGURE 4. Source Code in Creating Hotspot

Figure 4 shows the screenshot of source code in creating a hotspot. If it is not connected to any network, creating hotspot loading process will execute.

```
public WifiConfiguration wifiConfig(){
    String ind = "ssid";
    try{
        BufferedReader br = new BufferedReader(new FileReader(hotspotfile.getAbsolutePath()));
        for(String line; (line = br.readLine()) != null; ) {
            if(ind.equals("ssid")){
                SSID = line.trim();
                ind = "pass";
            }else if(ind.equals("pass")){
                pass = line.trim();
            }
        }
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
    wifiConfig = new WifiConfiguration();
    wifiConfig.SSID = SSID;
    wifiConfig.preSharedKey = pass;
    wifiConfig.status = WifiConfiguration.Status.ENABLED;
    wifiConfig.allowedGroupCiphers.set(WifiConfiguration.GroupCipher.TKIP);
    wifiConfig.allowedGroupCiphers.set(WifiConfiguration.GroupCipher.CCMP);
    wifiConfig.allowedKeyManagement.set(WifiConfiguration.KeyMgmt.WPA_PSK);
    wifiConfig.allowedPairwiseCiphers.set(WifiConfiguration.PairwiseCipher.TKIP);
    wifiConfig.allowedPairwiseCiphers.set(WifiConfiguration.PairwiseCipher.CCMP);
    wifiConfig.allowedProtocols.set(WifiConfiguration.Protocol.RSN);
    return wifiConfig;
}
```

FIGURE 5. Source Code for Wi-Fi Hotspot Configuration

Figure 5 shows the source code of Wi-Fi hotspot configuration. In configuring a Wi-Fi hotspot, SSID and password is being required by the host. Generating SSID and password is also optional.

```
try {
    fis = new FileInputStream(rootDir);
    response = new Response(Response.Status.OK, getMimeTypeForFile(rootDir.getName()), fis);
    response.addHeader("content-disposition", "attachment; filename=" + URLEncoder.encode(rootDir.getName(), "UTF-8"));
} catch (FileNotFoundException e) {
    e.printStackTrace();
} catch (UnsupportedEncodingException e) {
    e.printStackTrace();
}
return response;
```

FIGURE 6. Source Code in Downloading a File

Figure 6 shows the snippet of code in downloading a file. This code is the servers HTTP POST response when client do requests to download a file. After getting the file path from the clients' request, this snippet is being executed.

```
try {
    Process sh = Runtime.getRuntime().exec("su");
    OutputStream os = sh.getOutputStream();
    os.write(("/system/bin/screencap -p " + "/sdcard/out.jpg").getBytes("ASCII"));
    os.flush();
    os.close();
    sh.waitFor();
} catch (IOException e) {
    e.printStackTrace();
} catch (InterruptedException e) {
    e.printStackTrace();
}
```

FIGURE 7. Source Code in Screen Capture

Figure 7 shows the snippet of code in screen capture. Every 1 second, the screen capture is implemented and stored into specific path which is /sdcard/out.jpg. During the implementation of the code, execution of super user is required. Super User is in priviledge mode where devices should be rooted.

RESULTS AND DISCUSSION

The researchers conducted a two-way test suitable for the problems stated namely: slow process, vulnerable to data corruption, and inability to broadcast file. First test is the demonstration of the application to the respondents regarding functionalities. Another test is by using appthwack tool which can test the applications performance and

functional interactive to ensure the quality and agile speed of the application.

TABLE (1). Functionality Test in the Application

FUNCTIONALITTY		
	YES	NO
1. Does the application can share files?	✓	
2. Does the application can share folder?	✓	
3. Does the application can share files to multiple devices?	✓	
4. Does the application can download file?	✓	
5. Does the application can download folder?	✓	
6. Does the application can create hotspot?	✓	
7. Does the application can share screen?	✓	
8. Does the application can share screen to multiple devices?	✓	

This is the list of functionalities that the researchers achieved. First, the host can create hotspot which requires the SSID and password where host and peer able to connect to each other and share file(s). The application achieved the basic functionalities such as it can share files/folder to multiple devices and the peer can download the shared file. The Host can also share screen to multiple devices with a minimum of 2 device(s) up to 5 devices depending on the device specifications.

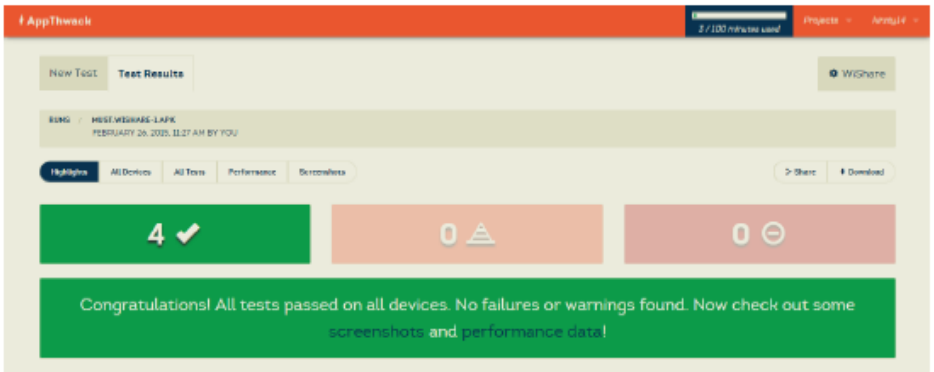

FIGURE 8. Performance test using AppThwack Tool

The screenshot of the test results of the applications functionality. In this Figure, all tests are passed on all devices. No failures or warnings found in the applications functionality.

Application Benchmark

Screenshot to the problem in slow process

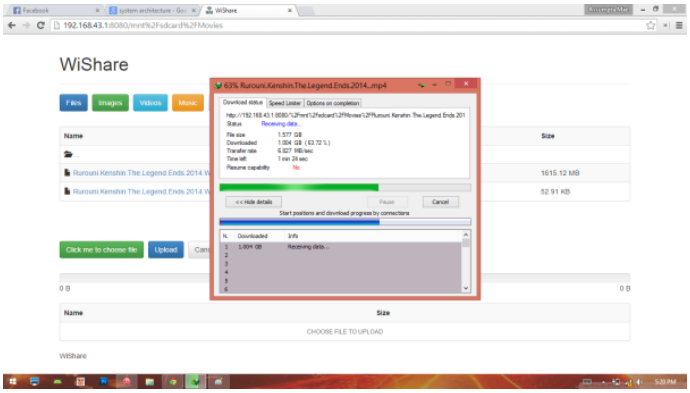

FIGURE 9. Screenshot of Downloading a File

Figure 9 shows the screenshot in downloading a movie with size of 1GB. The researchers test the application regarding the transfer rate and is concluded that the application (WiShare) is capable of transferring large file at a faster speed. WiShare has approximately 6.8MB per second where the download lasts approximately 4 minutes only.

1a. Bluetooth VS. WiShare Transfer using Ad hoc Network

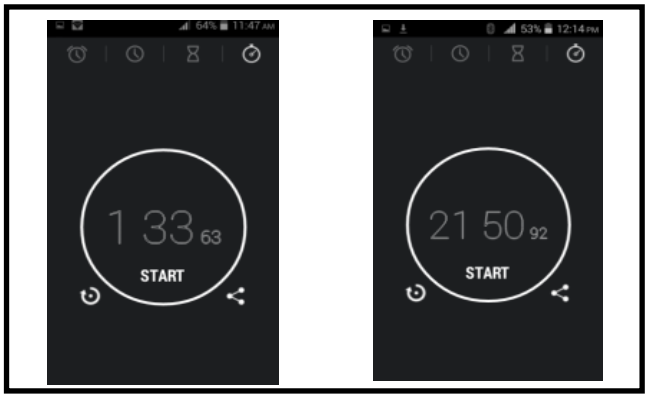

FIGURE 10. Screenshot of Bluetooth VS. WiShare Transfer Rate

Figure 10 shows the screenshot of the time scale of transferring files using ad hoc network. The researchers tested a file with a size of 255MB. Sending the 255MB file using Bluetooth takes 21 minutes and 50 seconds to finish while using WiShare it takes 1 minute and 33 seconds only.

Router VS. Ad hoc Network

FIGURE 11. Screenshot of Router VS. Ad hoc Network Transfer Rate

The screenshot in figure 11 shows the screenshot of a transfer rate time scale using router and using ad hoc network. The researchers tested both network with a file size of 255MB. Router takes 1 minute and 41 seconds while ad hoc network takes 1 minute and 33 seconds after the transfer of file is done. Both networks have its own disadvantages; Router's connection is dependent in the area or location and Ad hoc network is limited only in connected users.

Justification to the problem vulnerable to data corruption

The application uses HTTP POST which is reliable in transferring data through HTTP because HTTP can handle multiple files and request in one socket. HTTP uses TCP protocol which guarantees delivery of data and also guarantees the packets that can be delivered in the same order in which they were sent and it is suited for an application that requires high reliability, congestion control and transmission time is relatively less critical.

Response to the problem in inability to broadcast file

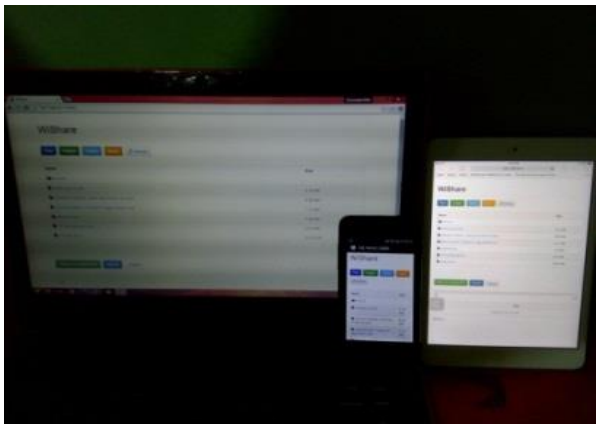

FIGURE 12. Broadcast File with Multi-platform Devices

The image in figure 12 shows the ability of the application to broadcast shared file to multiple devices. The application is also capable of sharing files to multi-platform devices such as Windows, IOS, Android, and PC/Laptops. Multiple users can download the file simultaneously.

System Usability Scale Evaluation

TABLE (2). Evaluation of the application's usability survey

QUESTIONS	AVERAGE SCORE
1. I think that I would like to use this application frequently.	3.55
2. I found this application unnecessarily complex.	2.45
3. I thought this application was easy to use.	2.80
4. I think that I would need assistance to be able to use this application.	2.75
5. I found the various functions in this application were well integrated.	3.25
6. I thought there was too much inconsistency in this application.	2.85
7. I would imagine that most people would learn to use this application very quickly.	3.15
8. I found this website very cumbersome/ awkward to use.	3.30
9. I felt very confident using this application.	3.20
10. I needed to learn a lot of things before I could get going with this application.	2.55

This image shows the average score for each questions answered by the respondents during the survey done by the researchers. For each question, an over-all score was calculated by summing up all the scores in each question and then dividing the sum by the number of respondents in order to get the average. For example, the sum of question 1 (sum = 71) was divided by 20 respondents which results to an average score of 3.55. The average score of 3.55 corresponded to higher rate since the various questions use different rates. The odd question numbers indicates a positive question and the even

question numbers indicates a negative question. The odd numbers rating should be 4 & 5 to gain positive feedback; even numbers should be 1 & 2 to gain positive feedback. Both odd & even numbers has resulted a positive feedback based on the gathered average scores from the respondents.

System Usability Scale of the Application Chart

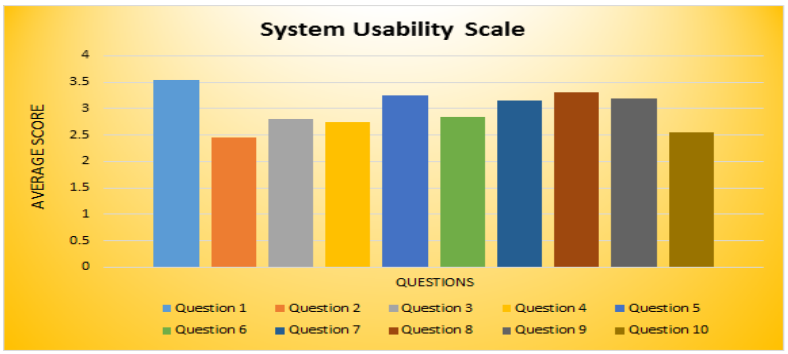

FIGURE 13. System Usability Scale Chart

This image illustrates the graphical presentation of the tabulated average scores as shown in figure 13. It is concluded that the application being evaluated indicates that the application has high demand from the respondents for its functionality and usability.

CONCLUSION AND RECOMMENDATIONS

The study was developed in order to address the problems in file and screen sharing namely slow process, vulnerable to data corruption, and inability to broadcast file. Based on the results, it is then concluded that the development of the application improved for a better and simple way to share files and screen to any device. Thus, during the demonstration and survey, the application gained positive feedback from the respondents. Using the application, the application can share file with fast transfer rate, protect data from corruptions, and capable to broadcast file and screen to multiple devices.

During the development of the application, some loopholes were found. In order to further enhance the proposed study, the following issues are recommended to address for future works.

1. Screen Sharing should be able to enhance by creating new libraries in Android Studio. Streaming the screen to other devices is highly recommended to future researchers by using Android Native Development Kit.
2. Screen Sharing should be available in rotate mode of the screen and should have an auto-adjust width and height.
3. Do not limit the connected user to the hot spot network.

ACKNOWLEDGMENTS

The author would like to thanks my colleagues, students, to the faculty of Information Technology Department and to the reviewers to make this research paper a success.

REFERENCES

1. Abiona et al (2009)., Architectural Model for Wireless Peer-to-Peer (WP2P) File Sharing for Ubiquitous Mobile Devices
2. Anastasi, G. et al.(2004).*IEEE 802.11 Ad Hoc Networks: Performance Measurements.*
3. Chamlac, Imrich et al (2003). *Mobile ad hoc networking: imperatives and challenges.*
4. John (2012). Comparison of web-based file transfer methodologies.*from http://www.filecatalyst.com/comparison-of-web-based-file-transfer-methodologies*
5. LanSchool (2010). Classroom Management Tools for Effective Teaching & Enhanced Learning.
6. LST, (2010) Classroom Management Tools for Effective Teaching &EnhancingLearning.*fromhttps://www.google.com.ph/url?sa=t&rct=j&q=&esrc=s&sourctee=web&cd=1&ved=0CC8QFjAA&url=http%3A%2F%2Fwww.datainnovations.fi%2Ffiles%2Fdatainnovations%2FLanschool_Whitepaper.pdf&ei=qq0JVNXvFtG68gXTh4Iw&usg=AFQjCNFKoaIoPT_NgSVerrJmFeDdcDr6tw&sig2=A2bX8QQGBtwpa6bffNtH1w&cad=rja*
7. Mitchell (2014).File Sharing 101 Introduction to Network File Sharing. *fromhttp://compnetworking.about.com/od/basicnetworkingconcepts/a/file_sharing.htm*
8. NST, (2014). NetSupport Ltd. *NetSupport School Getting Started Guide.*
9. Opara&Aririguzo, “Characteristics of file sharing and peer to peer networking. 2011”,*fromhttp://www.ajol.info/index.php/ijonas/article/view/78017*