Journal of Mobile Multimedia, Vol. 8, No.1 (2012) 001-024

STUDY AND DEVELOPMENT OF THE TRANSMISSION METHOD FOR LARGE MULTIMEDIA FILE SIZE USING MULTIMEDIA MESSAGING SERVICE TECHNOLOGY

ANDIK SETYONO
Multimedia University, Malaysia
Dian Nuswantoro University, Indonesia
setyonoandik@gmail.com

MD. JAHANGIR ALAM
Multimedia University, Malaysia
md.jahangir.alam@mmu.edu.my

C. ESWARAN
Multimedia University, Malaysia
eswaran@mmu.edu.my

Received July 21, 2011
Revised November 20, 2011

Mobile technology has grown rapidly in recent years due to the availability of sophisticated mobile phones in the market. Many features offered in the mobile phones encourage people to develop mobile-phone-based systems. Internet applications also have been developed to transmit data by mobile phone in a mobile environment. In this paper, we study and develop transmission methods for large multimedia files using MMS technology. We implement file compression, splitting, masking and cropping techniques to improve the ability of MMS technology in multimedia message delivery so that it can transmit large size data files which are not possible with the existing technologies. We hope that this research can provide a valuable contribution to the development of MMS technology. The experimental results demonstrate that the proposed techniques perform better than the conventional MMS technologies in preserving the quality of data. The modified MMS technology can be used to develop practical client-server applications.

Key words: MMS technology, Multimedia file, Compression, Splitting, Transmission method
Communicated by: J. Ma & T. Shih

1 Introduction

Multimedia messaging service (MMS) is the enhanced version of SMS [1]. MMS is designed to overcome limitations and shortcomings of both SMS and email, making it suitable and effective to work over contemporary and evolving wireless infrastructure, and also to fulfil the market demands of wireless messaging [2]. Mobile messaging continues to progress, including MMS. The open mobile alliance (OMA) has developed the MMS version 1.3 that can be used to send multimedia messages up to 600 KB [3]. Currently MMS version 1.2 [4] which is still used by mobile phone vendors can only transmit data for each sending up to 300 KB. MMS can send multimedia messages to mobile phone recipients and external servers such as email servers and fax servers [5]. Interoperability between email and MMS is simplified because a typical MMS message body follows the multipurpose Internet mail extension (MIME) specifications that consist of a header and a body [6]. The header contains control information, while the body represents the message content. The body is encoded using the MIME multipart encoding scheme and uses a multipart/related structure [7].

Using the MMS technology, one can send not only text files but also image, audio and video files in mobile environments [8, 9]. Currently, the capability of MMS in sending large data files are limited, and hence it needs to be improved. For example in telemedicine applications, large data files are required to be transmitted without any degradation in the quality and this is not possible with the existing MMS technologies. In this paper, we propose techniques to enhance the capability of MMS in multimedia message delivery. We use compression, splitting, masking and cropping techniques to implement the MMS application that is installed on mobile phone devices. It is shown that with the implementation of these techniques, the ability of the MMS technology is enhanced significantly compared to the existing technology.

The compression technique is used to reduce the file size [10] which is followed by a splitting process. The system splits the compressed data file if it is still larger than the permitted MMS capacity. Masking technique is used to send audio or video files. Cropping is a technique used to get selected area of an image file. The aim of including all these techniques is to make the system adaptive to the size and type of files and also to make it more cost-effective. In this paper, we study and discuss these techniques in detail. To know the performance of the techniques, we develop a client-server application. The client side application is MMS application that is installed on the mobile phone and the server application is an Internet application that processes the data sent from the client side.

The main objectives of this research are: (i) to study and develop transmission method using the MMS technology (ii) to explore and enhance the ability of the MMS technology (iii) to develop a client-server system application prototype (iv) to evaluate the performance of the MMS technology in implementing the application and (v) to design an adaptive and efficient system using MMS technology. This research is expected to make the following contributions: (i) providing knowledge about the use of MMS technology to develop a mobile system application (ii) providing a description of the techniques for transmitting large size multimedia files preserving the quality of data, and (iii) evaluating the advantages and shortcomings of the MMS technology.

The remaining part of the paper is organized as follows: section 2 discusses the study of techniques. Development application and discussion is presented in section 3, performance evaluation is presented in section 4 and conclusion and future work are presented in section 5.

2 Study the Techniques

In this section, we perform a study and discuss about the techniques that can be used in developing a client-server application such as compression, splitting, masking and cropping techniques. We implement and improve them to develop MMS application on the mobile phone. We use MMS technology as mobile client application because MMS has advantages such as (i) multimedia capabilities, (ii) mobile phone and email addressing modes, (iii) efficient transport mechanism, where

MMS relies on an efficient message retrieval mechanism, (iv) charging framework leaves freedom to operators for the development of billing models and (v) future-proof open standards and worldwide acceptance [7].

We develop data transmission technique using compression, splitting, masking and cropping for not only enhancing the ability of the MMS technology but also for other reasons such as security, privacy, integrity and implementation cost. If we use other technologies such as SMS, the data is limited in size and it is text only. If we use email, we must create an email server in order to able to control process and manipulate the data. Securing email server from spam and cracker is essential for the success of the system which will be an additional work. If we use Internet connection to access server applications using 3G network on mobile phone, we will have difficulties because mobile phones in general have small memory and display capabilities and also low processing power which will make the system access inconvenient. Further, in all these methods, the system will be less secured and ineffective. Besides, the coverage of the 3G network is limited; it commonly covers big cities or densely populated areas only. The 3G mobile network does not cover remote and rural areas, whereas 2.5G or GPRS is already established in remote areas in most of the countries in the world.

We choose to develop client-server application using MMS technology because MMS has advantages such as MMS needs lower bandwidth and cost compared to smart phones for accessing server applications. MMS also needs ordinary mobile phones and low-price device. MMS data is more secured and also protected from unauthorized user compared to sending data through smart phone to the Internet application directly. Further almost all of the existing mobile phones have MMS features like the smart phones. In the following sections, we explain in detail the techniques that are used to transmit large multimedia file using MMS technology.

2.1 Study of the Compression Technique

In this section, we discuss about compression techniques. The aim of data compression is to reduce the redundancy (useless information). Compression is needed to save the storage space and bandwidth requirements and also to reduce the time for data transmission [11]. There are two types of compression techniques: lossy and lossless compression. Lossy compression causes the file size smaller but eliminates some information, while lossless compression technique does not eliminate any information [12]. There are several techniques for lossy compression, for example in image file compression we use techniques such as colour reduction, chrome sub-sampling and transform coding (DCT, DWT) [13]. There are several types of lossless compression algorithms such as entropy encoding (Huffman, Arithmetic), Shannon-Fano, Lempel-Ziv-Welch (LZW), Run Length Encoding and Dynamic Markov Compression (DMC) [10]. In JPEG baseline compression process, there are several steps such as discrete cosine transform (DCT), quantization up to the Huffman coding [14]. The reduction of file size will occur at the DCT and quantization stage, which would remove valuable information and redundancies [15]. In Huffman encoding stage, information that appears frequently will have smaller number of bits and the information that rarely appears will have larger number of bits [16]. The JPEG format uses sequential DCT-based mode where the image is compressed in a single scan using

DCT-based lossy compression technique. As a result, the decoded image is not an exact replica, but an approximation of the original image. Figure 1 shows the process of JPEG compression.

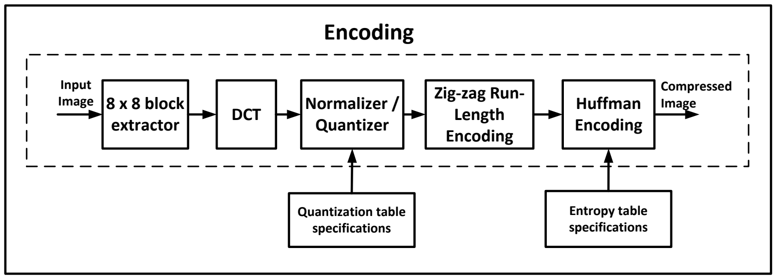

Figure 1 Compression process of JPEG compression.

The lossy and lossless compression techniques have advantages and limitations. This depends on the purpose for doing the compression process. For example, when we are not particular about the quality, meaning that the image can be correctly perceived by human eyes, then we can use lossy compression. On the other hand when we need information with excellent quality to ensure the accuracy of the data then we should use lossless compression. One of the advantages of lossy compression is the compressed data will have much reduction in file sizes and simplifies the process of transferring data in communications network. While in lossless compression, the data would not significantly decrease in size and so we still commonly need other techniques. A technique that is usually used is the splitting data technique. The advantage of lossless compression is that the quality of data will remain intact where it can be obtained as the original file, while for the lossy compression technique the data quality will experience a significant degradation. According to 3GPP, with MMS technology, we can send message contents using certain formats only. This is influenced by the ability of the mobile phone, communications network and MMS Center that are provided by the operator. The details of message content that can be sent using MMS technology is shown in Table 1.

Table 1 Message content classes for 3GPP2 devices.

Content domanis	Core Message Content Domain						Content core contain domain	
Content classes	Class Text	Class Image Basic	Class Image Rich	Class Video Basic	Class Video Rich	Class Megapixel	Class Content Basic	Class Content Rich
Text	US-ASCII UTF-8, UTF-16	US-ASCII UTF-8, UTF-16	US-ASCII UTF-8, UTF-16	US-ASCII UTF-8, UTF-16	US-ASCII UTF-8, UTF-16	US-ASCII UTF-8, UTF-16	US-ASCII UTF-8, UTF-16	US-ASCII UTF-8, UTF-16
Still Image	None	Baseline JPEG	Baseline JPEG	Baseline JPEG	Baseline JPEG	Baseline JPEG	Baseline JPEG	Baseline JPEG
Bitmap Image	None	GIF87a, GIF89a, WBMP	GIF87a, GIF89a, WBMP	GIF87a, GIF89a, WBMP	GIF87a, GIF89a, WBMP	GIF87a, GIF89a, WBMP	GIF87a, GIF89a, WBMP	GIF87a, GIF89a, WBMP
Vector Graphics	None	None	None	None	None	None	None	SVG Tiny
Speech	None	AMR Narrowband	AMR Narrowband	AMR Narrowband	AMR Narrowband	AMR Narrowband	AMR Narrowband	AMR Narrowband
Audio (Music)	None	None	None	None	None	None	None	MPEG4 AAC
Synthetic Audio	None	None	SP-MIDI	SP-MIDI	SP-MIDI	SP-MIDI	SP-MIDI	SP-MIDI
Video	None	None	None	H.263 with AMR-NB (3GP)	H.263 with AMR-NB (3GP)	None	H.263 with AMR-NB (3GP)	H.263 with AMR-NB (3GP)
Max Image Resolution	Not Applicable	160 x 120	640 x 480	640 x 480	640 x 480	1600 x 1200	640 x 480	1600 x 1200
Message Size	≤ 30 KB	≤ 30 KB	≤ 100 KB	≤ 100 KB	≤ 300 KB	≤ 300 KB	≤ 100 KB	≤ 300 KB

We conduct an observational measurement for the compression results by using the existing software. We use some samples of data such as text, image, audio and video to find out the compression ratio so that we can find the best compressed file. It is also highly influenced by the capability of the communications network, especially mobile communications network. The selection of an appropriate compression algorithm will produce file sizes much smaller making it easier to process the data transmission in a network. We use JPEG image format to simulate the compression process. We can also develop lossless compression using GIF, PNG or BMP image format. We choose JPEG format as a sample because the image format that is the most commonly used by mobile phone normally for digital image storage standard. Therefore, we can take picture using camera on mobile phone directly without doing adjustment on the image format. We can specify the quality factor on the quantization process to reduce the file size. The file will be small file but it can be understood by human eye well. The reason is we can afford to lose a lot of more information in the chrominance components than in the luminance component. The human eye is not sensitive to high-frequency chrome information. Table 2 shows some results of the different compression methods.

Table 2 the Compression result of multimedia files.

Data	Size	Type	Compression	Result	Ratio
Text (*.doc)	4.305 KB	**Lossless**	Zip	2.744 KB	0.64
			Rar	2.383 KB	0.55
			Docx	2.931 KB	0.68
			PDF	426 KB	0.10
			Plain Text	32 KB	0.01
			Web page	106 KB	0.02
Image (*.bmp)	769 KB	**Lossless**	TIFF (LZW)	572 KB	0.74
			GIF (LZW)	229 KB	0.29
		Lossy	JPEG (75%)	57 KB	0.07
			JPEG2000 (75%)	335 KB	0.43
Audio (*.wav)	2.017 KB	**Lossless**	ZIP	834 KB	0.41
			Rar	590 KB	0.29
			FLAC	310 KB	0.15
		Lossy	MP3 Audio	129 KB	0.06
			AAC Audio	62 KB	0.03
			MPEG-4 Audio	63 KB	0.03
			WMA	224 KB	0.11
			Monkey's Audio	232 KB	0.12
			Ogg Vorbis	64 KB	0.03
Video (*.avi)	436.738 KB	**Lossless**	ZIP	79.138 KB	0.18
			Rar	81.215 KB	0.19
		Lossy	Mobile Phone MP4	204 KB	0.0004
			Flash Video Movie	571 KB	0.001
			MPEG I	1.248 KB	0.003
			MPEG II	3.512 KB	0.008
			MP4	2.163 KB	0.005
			WMV	1.701 KB	0.004
			Matroska	778 KB	0.002

2.2 Study of the Splitting Technique

In this section, we conduct the study about splitting file techniques. Splitting techniques are very suitable to be applied if the compression result still produces a fairly large file size that cannot be delivered using the existing technology through mobile communication network. For example, suppose that we want to send an image file size around 1 MB using MMS technology, where this technology uses the 2.5G network. We know that MMS can only deliver a maximum size 300 KB of the multimedia message depending on the capabilities of the mobile phones, communications networks and service providers used. To solve this problem, image file is compressed using an algorithm (lossy or lossless) so that the file size will be smaller. But if the file size is still large, the splitting technique can be implemented. Each split-file is sent one by one and then merged in the receiver side with the developed merging application. If we want to recover the original image, we can use the decompression technique to reconstruct the file on the server.

In the study of splitting techniques, we take an example case on the image file. We can perform the splitting file technique based on the file size using rectangular-split method (pixel) and bytes-split method (binary). The splitting file that uses the pixel, the method is employed if the original file size (*OFS*) is larger than the maximum capacity (*MC*) permitted. If *M = OFS/MC*, the number of split-files *N* is given by

$$N = \begin{cases} M & if\ M\ is\ an\ integer \\ Integer(M+1) & if\ M\ is\ a\ fraction \end{cases} \quad (1)$$

Let the original image is of size (*w x h*) where *w* and *h* denote the width and height of the image, respectively. Then the size of each of the *N* split images will be equal to {*w x (h / N)*}. The algorithm of the splitting technique is shown in figure 2.

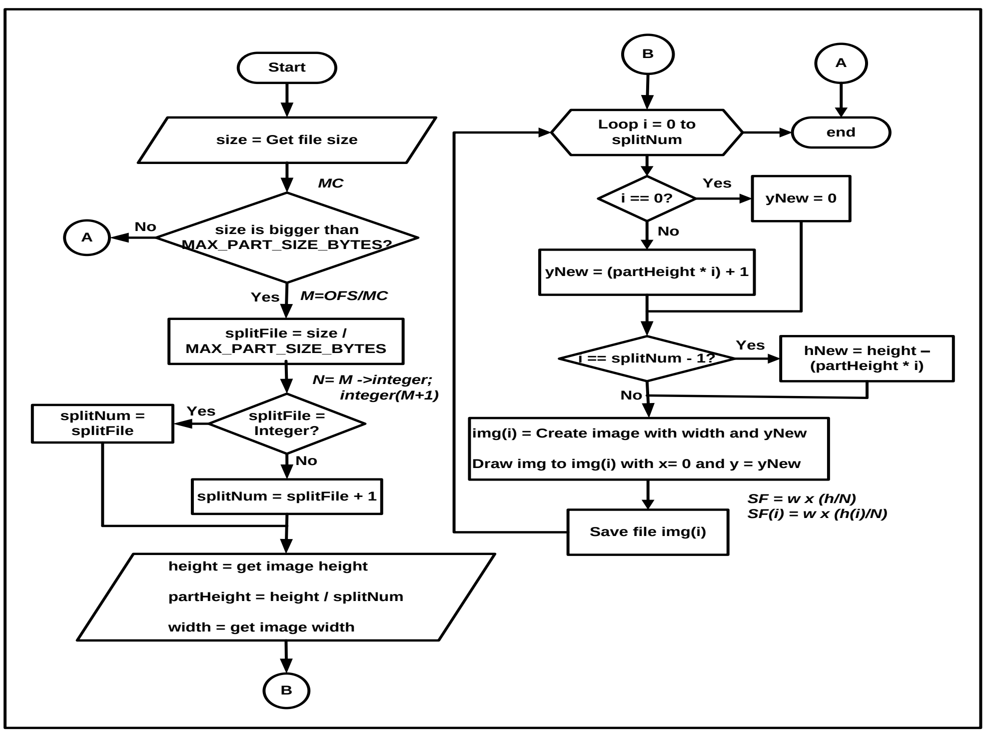

Figure 2 Algorithm of splitting message technique based on pixel.

The splitting file based on file size using bytes (binary), for an example of an image file of 500 KB size, it splits file using a maximum size, for an example 200 KB, the file will be split into 200 KB, 200 KB and 100 KB. We assume that the file size is made like a straight line. The split application will scan the file based on bytes (binary) that exist on the image. If the file size that is already scanned reaches 200 KB or equal with the maximum size, therefore the file is split. The split process will be repeated until the file size is reached at 200 KB.

The split technique which is shown in figure 1 is implemented to send an image file (JPEG). We also simulate the technique using Java technology. This technology cannot be used to develop MMS application for sending audio and video files yet, this is because there is no existing technique to solve this problem. We use the masking technique to solve the problem. We use bytes-split technique to split the compressed file, and then we use image header that is used to mask audio and video files. The image header is added in each split-file formed. This aim of this technique is to split-files that can be known by MMS center and forward to target recipient. The header of image file is 128 bytes size. Figure 3 shows the algorithm for splitting technique based on file size using bytes (binary) method for audio and video files, where we should add header on each split-file.

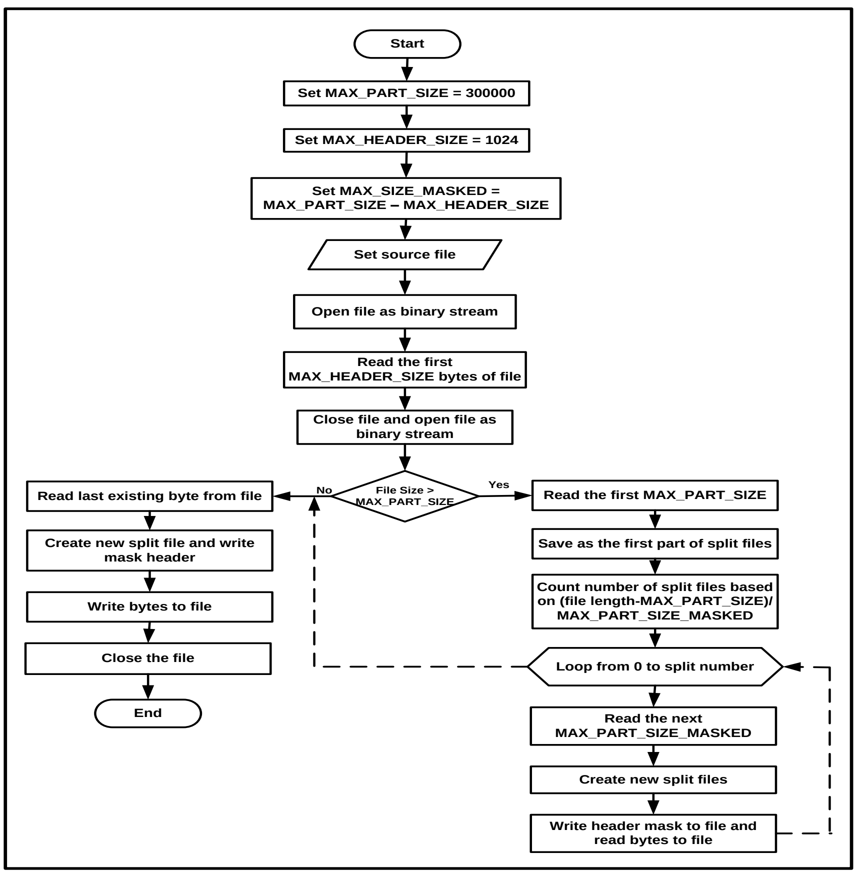

Figure 3 Algorithm of splitting and masking techniques based bytes-split method

The splitting techniques based on the pixel and bytes which are implemented on the multimedia file can have advantages and shortcomings. The split technique commonly uses bytes method but in several cases, we can use other method like splitting file by pixel and also hybrid method like in sending audio and video files. We use splitting and masking techniques based on bytes to add image header for each split-file of audio and video files. This is determined and adjusted with requirements to get a benefit to make the system adaptive and efficient. The explanation of the algorithm which in figure 3 is as follow:

```
MAX_PART_SIZE = 30000;
MASK_HEADER_SIZE = 128*1;
MAX_PART_SIZE_MASKED = MAX_PART_SIZE – MASK_HEADER_SIZE;          (2)
If fileSize > MAX_PART_SIZE_MASKED
    numParts = fileSize / MAX_PART_SIZE_MASKED;                   (3)
    endSize =  fileSize % MAX_PART_SIZE_MASKED;                   (4)
End.
```

In implementing of splitting technique in MMS application on the mobile phone is seemed to be an easy work, but the delivery of split-files are complex. The split-files are not known by MMS center if they do not have header content suitable for MMS format. How the split-files can be known by MMS center is the main problem. To solve this problem, we give an header for each file, and then package them into byte and finally sent.

2.3 Study of the Cropping Technique

The other technique that can be used to manipulate data for sending multimedia file using MMS technology is cropping technique. The existing MMS technology commonly uses resize technique to solve problem sending larger file size than maximum size allowed. This technique will reduce the image, quality and file size. An image that has large image size will produce large file size too, whereas we need information only in small part of the image. To solve the problem, we use cropping technique to determine selected area of the image which will be sent by using MMS technology. This technique will produce small image size and file size so that it enables the image file is sent without compression and split techniques. It will produce an adaptive and efficient system by reducing cost for sending the file but the transmitted information has a good quality. It is possible, if the cropping file result is still larger than the maximum size, it will be split in order to be transmitted. Figure 4 shows the developed MMS framework using cropping technique.

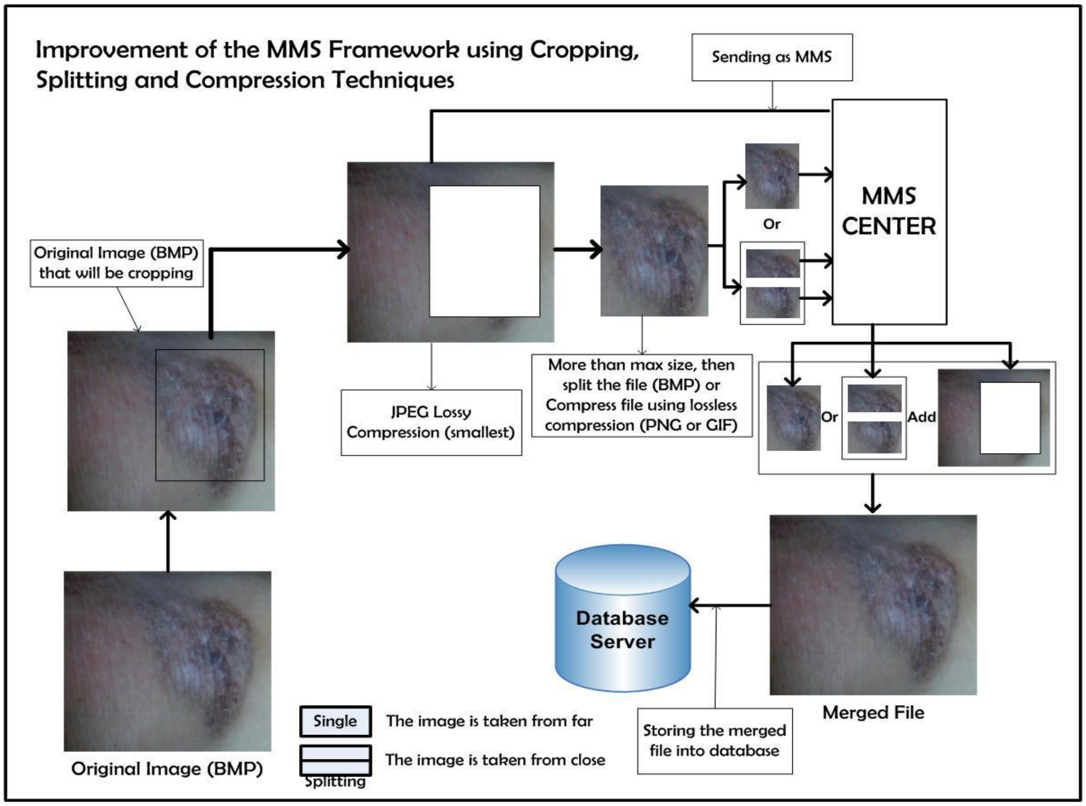

Figure 4 Development of the MMS framework using cropping technique.

3 Development Application and Discussion

In this section, we develop a client-server application. We use mobile phone device as client and computer with Internet connection as server. On the client side, we develop MMS application that is adjusted to the information needed by implementing compression, splitting, masking and cropping techniques before sending the message. It aims to improve the capability of the MMS technology which cannot be done by the existing technologies. On the server side, we develop Internet application to process the data from the client, and then is stored into database server. Many Internet applications can be developed on the server side which can be accessed in mobile environment using 2.5G or 3G networks. In this research, we develop a telemedicine system on the server side to prove that the technique works properly. The developed system can be used to send and receive multimedia data in mobile environment using MMS technology. Figure 5 shows the architecture of the client-server application that is developed in this research.

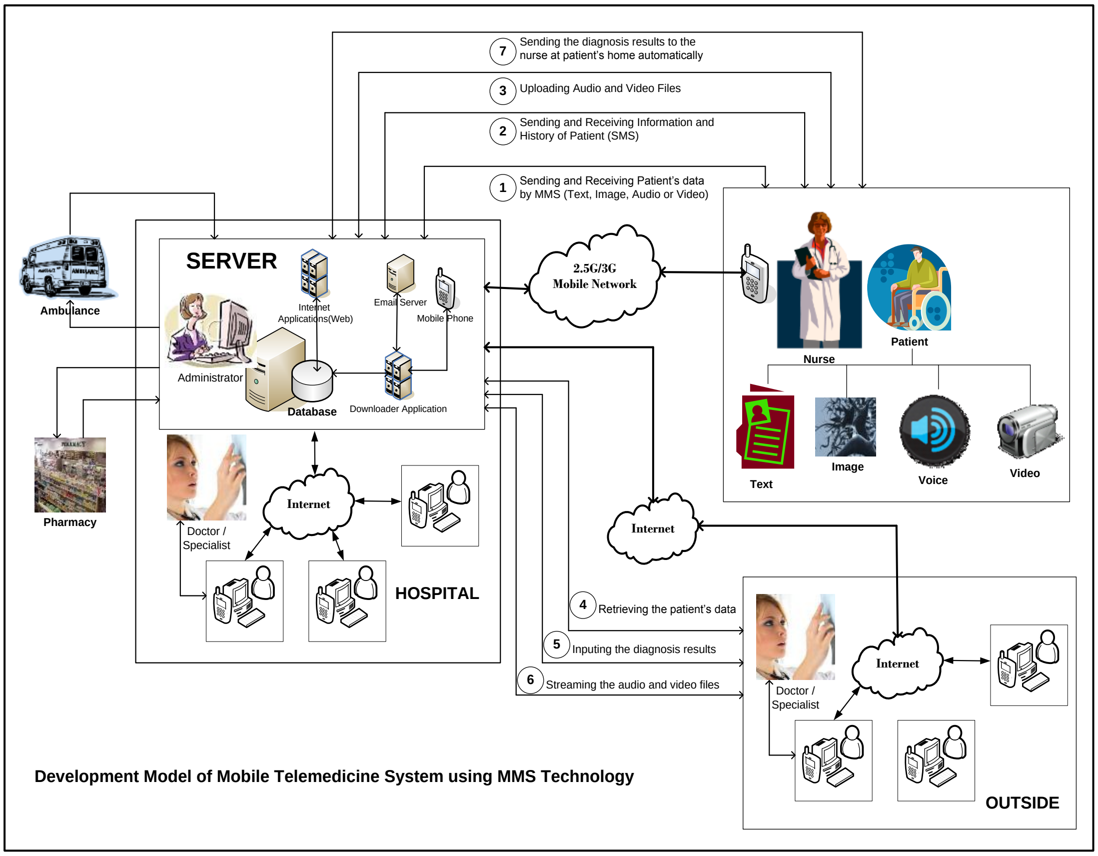

Figure 5 MMS framework for mobile telemedicine system application.

The MMS technology is used to develop mobile telemedicine system. The multimedia message with a file size larger than the allowed size can be sent to mobile phone recipient (MMS gateway). The nurse at the patient's home sends multimedia message using MMS technology to the hospital, then the doctor can retrieve the data from the server using the telemedicine application on the server. After performing diagnosis, the doctor can input the diagnosis result into the system. Many client-server applications use MMS technology [17, 18]. This is caused the MMS technology supports multimedia message delivery, likewise with email. Email feature on the mobile phone is already set by mobile phone vendor with limiting the size of email. This is influenced by the communication network that is provided by the service provider. MMS is also can be used to develop streaming application [19, 20]. By using mobile phone as client, this application can be accessed in mobile environment. We can also request and receive information such as the history of a patient in the form of text message using SMS technology.

3.1 Simulation for Sending Image

This MMS application is developed using Java technology, which allows only 30 KB each sending so that in this case is the maximum size is 30 KB. In this simulation, we assume that the 30 KB is the maximum size of the technology that is already provided by mobile phone vendors which is built by using the different software. For example, we send the JPEG image that has size 100 KB using MMS technology. Due to message size larger than 30 KB, the message will be applied the compression technique and if the compression result is still greater than 30 KB, the message will be split to become smaller in size. Figure 6 illustrates an MMS application that implements the compression and splitting techniques.

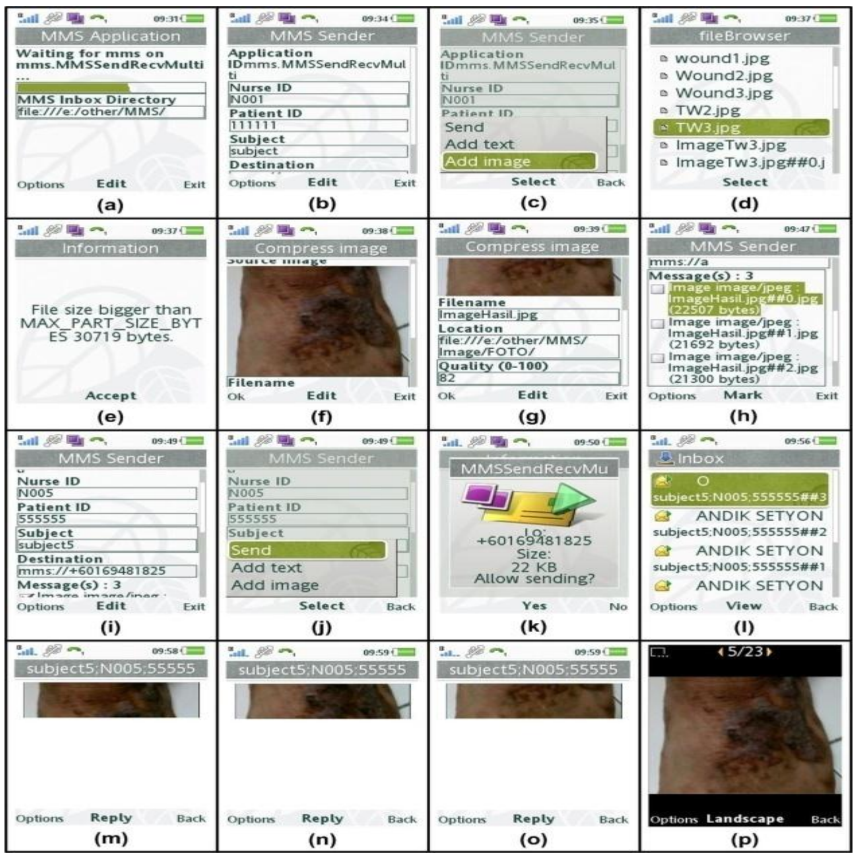

Figure 6 Simulation for sending large JPEG image file size.

The clarification of the simulation results are: (a) display for setting an inbox, (b) user interface for sending MMS, (c) file (image) that will be sent, (d) selecting a file that will be sent, (e) the information in the form of notification that the file is more than 30 KB, (f) display the image before compressing and splitting, (g) compressing files with a selected value of the quality factor, (h) splitting files, (i) determining the destination number (SIM card) transmission, (j) delivery process, where in one time multiple messages are sent sequentially, (k) notification that the messages is ready to be sent and needs the approval, (l) MMS inbox (sending the multimedia message successfully), (m) the first part of the split image, (n) the second part of the split image, (o) the third part of the split image and (p) the split-files are merged to be a single file before decompressing.

3.2 Simulation for Sending Audio and Video Files

In this paper, we will also simulate for sending audio and video files that have larger file size than the maximum size allowed on the Java platform. It has been mentioned previously that the MMS application using Java platform has a maximum size only 30 MB for a single delivery and Java technology does not fully support yet for developing MMS application for sending audio and video files, so we need a new method to send them. The addition of the header for each split-file also becomes a problem in the simulation. We focus on the study to develop architectures and algorithms of MMS framework for Internet applications. We assume that the maximum size in Java is equal with the maximum size in the existing MMS application that is provided by the mobile phone vendors. We can also assume that the existing and developed system are using Java platform, and then we can compare them to get a better system. Figure 7 shows the simulation results for sending audio and video files.

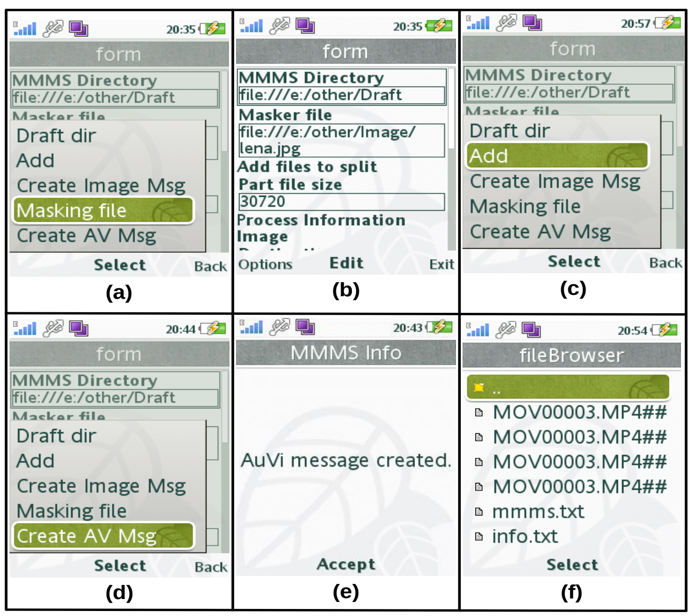

Figure 7 Simulation sending audio and video files.

In figure 7, we present the simulation results for sending audio and video files using mobile phone. The clarification of the simulation results are: (a) load an image that will be used to mask the split-files, (b) display the MMS directory and masked file, (c) browse audio or video file that will be sent, (d) create audio and video message, (e) audio or video message created successfully and (f) browse the created message result. The audio and video files that will be sent is already in the form of compressed file so that it is not needed to compress file first. This is different with JPEG image compression, where we can determine the quality of the image for JPEG files accordance with our needs. Before sending the message, the splitting process is done with masking technique. This method uses JPEG image to get header of the image (JPEG) so that the message will be known by MMS center. In this simulation, we can see and monitor the delivery result of MMS on the network monitor that is provided by an emulator. Figure 8 shows the simulation results which show the delivery message on the network monitor in details.

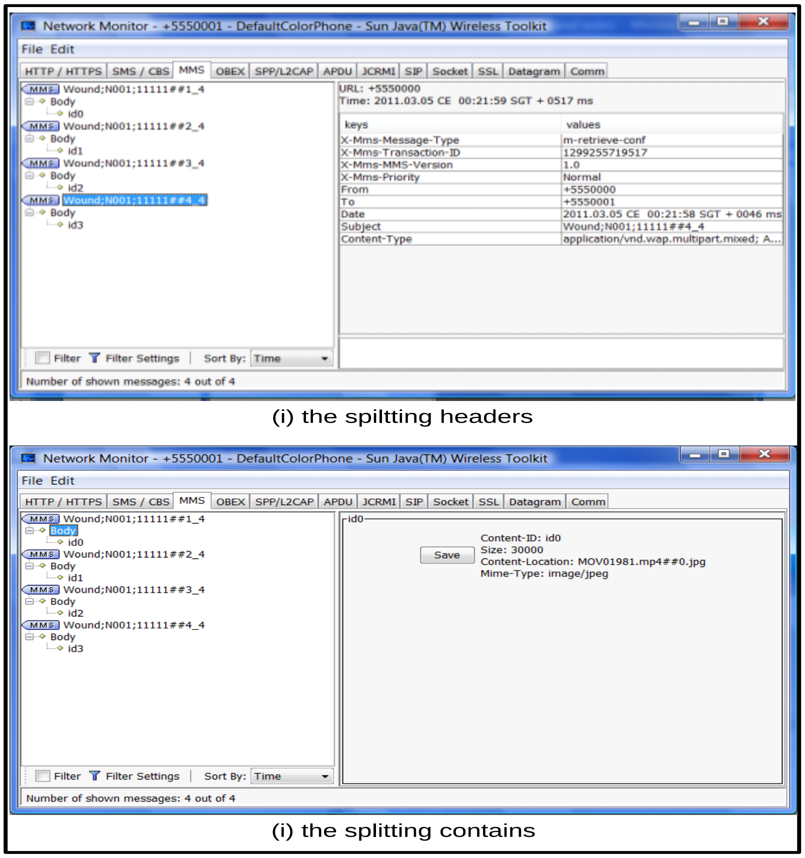

Figure 8 Network monitor for sending audio and video files.

In figure 8, we can explain about the output of the delivery audio and video files using MMS technology. (i) Displays the MMS parts, where each packet of MMS consists of header and body parts and we can also see the detail of the header for each split-file, (ii) displays the splitting contains, where each split-file has 30000 bytes (29872 bytes for body part and 128 bytes for header part). The MMS application for sending image, audio and video files can be made in one application but to explain the process of sending, we separate simulation the process one by one.

3.3 Simulation for Cropping Image Files

We also make cropping image application to simulate the selected image area. This technique is used before creating message on MMS application. The sent image can be taken from mobile phone camera, and then the image is loaded and opened to determine selected image area which will be sent. After getting the selected image area, the next step is to store the cropped file into the memory of mobile phone. Figure 9 shows the simulation results for cropping an image file.

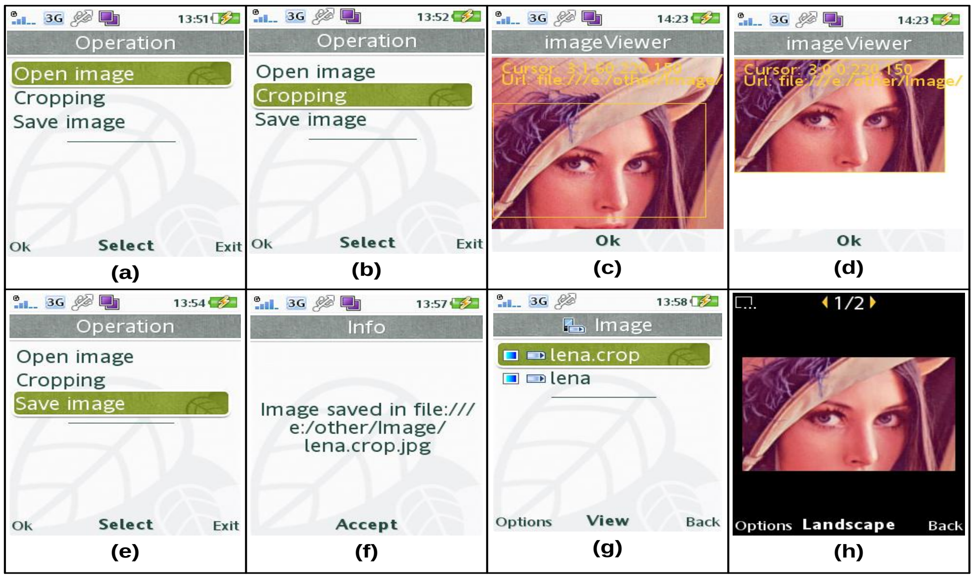

Figure 9 Simulation for cropping image file.

3.4 The Method for Making the Message Header

If the compressed file is larger than the maximum file size, the file is split to become few smaller files. Those files are sent one by one with adding the message header suitable that are allowed by MMS center. Before sending, the message is packaged into bytes and sent by pressing a button on the user interface of the MMS application. The header is given to each split-file, so that the MMS center will be known the received message and forward to recipient target. Then, the next step is to download the data which is successfully sent to mobile phone recipient into database server. Figure 10 shows the source code for adding the header on each split-file.

```
    Generated fields

    /**
     * The MMSSendRecvMulti constructor.
     */
    public MMSSendRecvMulti() {
        // get the header of image.jpg sample file
        try {
            InputStream in = getClass().getResourceAsStream("/image.jpg");
            DataInputStream reader = new DataInputStream(in);
            reader.read(jpegHeader, 0, jpegHeader.length);
            MAX_PART_SIZE_BYTES2 = MAX_PART_SIZE_BYTES - jpegHeader.length;
        } catch (Exception e) {
            System.out.println("error load jpg: " + e);
        }
    }

  int iend = istart + MAX_PART_SIZE_BYTES2;
  int len = iend > contents.length ? contents.length - istart : MAX_PART_SIZE_BYTE
  iend = istart + len;
  System.out.println("istart: " + istart + " ; iend: " + iend + " ; len: " + len);

  // adding JPEG Header to mask the file as JPEG File
  byte[] msgcontent = new byte[len + jpegHeader.length];
  System.arraycopy(jpegHeader, 0, msgcontent, 0, jpegHeader.length);
  System.arraycopy(contents, 0, msgcontent, jpegHeader.length, len);

  // create image message part for mms
  mp = new MessagePart(msgcontent, 0, msgcontent.length, mimeType,
          "id" + counter, contentLocation, null);
```

Figure 10 Source code for making header of MMS.

3.5 The Method for Merging and Decompression

Multimedia message sent by MMS will be received by mobile phone or email server as MMS gateway between mobile client and Internet server. The message is downloaded and then performed merging and decompressing processes before storing into database server. Sometimes, the sending process of split-files fails. This can happen due to disruption in network, disruption in MMS center or peak traffic network of the operator used. To solve this problem, after server successfully downloading the whole message on the mobile phone gateway, the server examines one by one the message successfully sent using information on the info.txt. If the file successfully downloaded is not same as the number of split-files which is informed on the info.txt, the server will give a message notification using SMS technology to the sender and request for sending back.

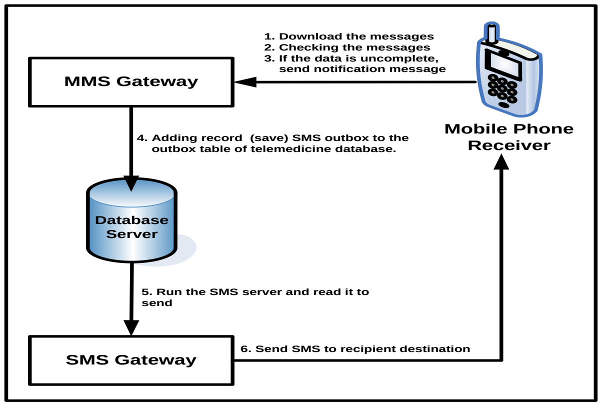

Figure 11 Checking and sending message notification.

To send the failure part of the file, the sender can open the creation message result on MMS application and take the parts of the file which are not sent can be sent back. It will create a system to be cost-effective, because we do not need to send back all the parts of the file to avoid spending more cost. It would cost more if we must send back all of the split-files. The next discussion is about merging process using merge technique. This technique is used to make the split-files into single file again. Figure 12 shows the merging file process that is split using bytes-split method.

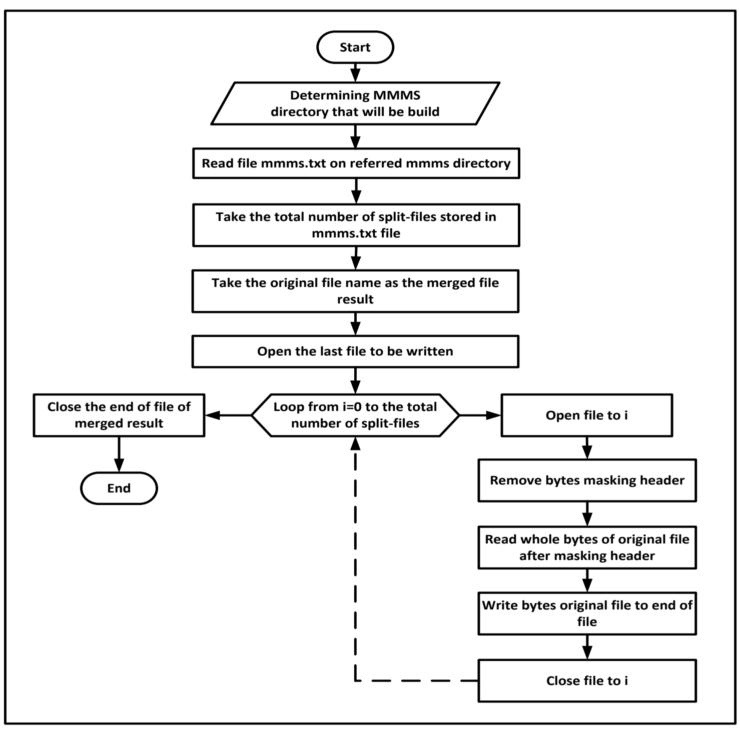

Figure 12 Sending and Merging file process using bytes-split method.

In figure 12, we can explain that each header that is added in each message (audio or video files) is removed for merging file process to be a single file. The next step is decompression the merged files to be reconstructed. The decompression is suitable for image file because we can decompress JPEG image to be original file format as bitmap image, convert into other format or increase the compression ratio. The video files do not need decompression because if we reconstruct the quality of data, then the result will be worse than the previous one. This is because the size and quality of the video files is already in the smallest one, so that if we make enlargement, the file will be broken and unclear.

4 Performance Evaluation

In this section, we conduct several experiments and measurements to determine the performance of the MMS application in sending multimedia messages such as evaluate the compression and split file result using the existing software and evaluate the developed client-server application. The observation results of the existing MMS application that is provided by the operators in Indonesia and Malaysia, there is no significant difference between sending multimedia messages to mobile phone and email server [21]. Both can be used to develop client-server application as the gateway. The difference of time needed is also influenced by the capability of the MMS center that has been provided by the service provider and also network traffic condition at the time of experiments. The MMS technology can send multimedia messages up to 300 KB data size depending on the service provider's facilities. The 2.5G (GPRS) and 3G (UMTS) network provide offering access to high bandwidth connections. Every operator or service provider of MMS uses different bearer technology, network and system specification, so that this matter affects the capability of transferring data. The transfer rate is between 115 until 384 Kbps [21]. The handsets with higher camera and screen quality make users more easily send large MMS. Larger message is good for customer to maintain the good quality of data. Operator will increase the message size and ensure that its MMSC has high capacity and can handle the growth of messages efficiently.

MMS technology is expected to grow with maximum size much larger than now along by the development of mobile communication network, smart phone devices and operator's technologies, so that the data transmission method will be more effective and adaptive. This will be possible in future because the mobile communication network will move from 3G to 4G that has wider bandwidth. To create a client-server system or mobile system that can be more adaptive and efficient within the transmission method, we implement some techniques such as compression, splitting, masking and cropping on the client side and merging and decompression on the server side. As we know that mobile phone has limited memory, small display, low power and slow data processing speed. These limitations make difficult to implement some techniques and to install on it.

In the simulation, we use JPEG image because almost all of mobile phones have default JPEG format for image file. The techniques can also be implemented for bitmap image such as GIF, WBMP and PNG. We perceive, if lossy compression can be implemented successfully on mobile phone, therefore lossless compression too, but may be more difficult because the compression result is still big, then it will be difficult to simulate using Java technology that allow maximum size 30 KB only. Although lossless compression has always been the first choice in special case like telemedicine, lossy compression (JPEG) is gaining momentum itself with a number of studies have been carry out to study the lossily compressed medical images.

We choose MMS technology for developing this mobile system for some reasons such as security, privacy, robustness and integrity. We implement splitting technique where the MMS message is already-encrypted is broken down into small-files so that if there are tapping the information obtained is incomplete. For privacy reason, the message which is sent from client to server side does not connect with database server directly but through mobile phone recipient as gateway so that the user such as patient cannot open the server application directly. It can be opened by doctor and administrator only using Internet network coverage with virtual private network (VPN) for server security. For robustness reason, we develop SMS technology to send message notification to the user as response from server to solve the problem, for examples if there is any part of file failure sent, therefore server will send SMS to the sender requesting to send back the part of the file which is unsuccessfully sent. When the doctor finishes inputting the diagnosis result into the server, the server will send message notification to the sender automatically for further action. For integrity reason, we can say that the mobile telemedicine system is a unity system, if there is trouble at one part of the system, then they will not be able to run well.

The simulation and experimental results show that all of the implemented techniques can run-well and we can install them on mobile phone devices. The applications on mobile phone device can be run successfully, so that the developed system in accessibility and acceptability already have fulfilled. After that, we perform some experiments to send the various types of data from client to server side. From the experimental results, we can get information about data quality in an image sample. We measure the Peak Signal to Noise Ratio (PSNR), Root Mean Squared Error (RMSE), Universal Image Quality Index (UIQI), Mean Squared Error (MSE), Mean Signal To Noise Ratio (SNR) and Absolute Error (MAE) for original image before compression and splitting process on the client side and reconstructed image after merging and decompression or conversion or increase the quality factor of the merged image on the server side. Table 3 shows the measurement results for compressing JPEG image with bit rate 1.5 bits per pixel (bpp) for each quality factor.

In Table 3, we can explain that the source image without compression (BMP image format) is compressed by using JPEG compression with setting bit rate 1.5 bits per pixel, then we can get the value of encoded bitrate, MSE, SNR and PSNR for each quality factor. Then, we perform the measurement to know the image quality between original image file, for an example Lena.BMP which has size 512 x 512 pixels (uncompressed) so that it generates file size 769 KB compared with compression and decompression result in the desktop computer and mobile phone. Table 4 shows the comparison results about time need and file result on desktop computer and mobile phone device.

Table 3 Measurement results for compressing JPEG image with bit rate 1.5.

Set quality	**Encoded bitrate**	**MSE**	**SNR**	**PSNR**
85	1	4.3	27	41.8
80	0.8	5.3	26.1	40.9
75	0.7	6.4	25.3	40.1
70	0.6	7.3	24.7	39.5
65	0.6	8.2	24.2	39
60	0.5	9.1	23.8	38.6
55	0.5	9.9	23.4	38.2
50	0.5	10.7	23	37.8
45	0.4	11.6	22.7	37.5
40	0.4	12.8	22.3	37.1
35	0.4	14.1	21.8	36.6
30	0.3	16	21.3	36.1
25	0.3	18.6	20.6	35.4
20	0.3	22.5	19.8	34.6
15	0.2	28.7	18.8	33.6
10	0.2	42	17.1	31.9
5	0.1	91.1	13.7	28.5

Table 4 Comparison results of the compression and decompression processes

No.	Quality Factor	Time needed in mili seconds				File result in KB			
		Desktop Computer		Mobile phone		Desktop Computer		Mobile phone	
		Comp.	Decomp.	Comp.	Decomp.	Comp.	Comp. Ratio	Comp.	Comp. Ratio
1	85	93	93	15950	3896	59	0.077	56	0.073
2	80	93	78	16796	3716	49	0.064	44	0.057
3	75	63	78	15666	3697	42	0.055	36	0.047
4	70	63	78	15648	3748	37	0.048	32	0.042
5	65	78	78	15622	3810	33	0.043	28	0.036
6	60	62	62	15843	3777	30	0.039	26	0.034
7	55	63	78	16126	3745	27	0.035	24	0.031
8	50	62	62	15378	3776	25	0.033	22	0.029
9	45	78	78	15277	3760	23	0.030	21	0.027
10	40	62	62	15218	3843	21	0.027	19	0.025

Table 4 shows that there are different times needed to execute compression and decompression process on the desktop computer and mobile phone significantly. The file results of compression also experience decreasing but it is not too significant. The files of decompression result have same size with the original files, but the information of the decompression files have been changed so that the quality of the file decompression decreased by eliminating redundancy. We perform measurement of image quality that running on desktop computer or personal computer and mobile phones device, where the implemented algorithms is same. The measurement results show that there are alterations which tend to decline when run on a mobile phone. This is probably due to the calculation in the mobile phone is not as precise as on the desktop computer or mobile programming language skills are not same as the programming language that runs on desktop computer. We use Java 2 Standard Edition (J2SE) for desktop application and Java 2 Micro Edition (J2ME) to mobile phone (MMS) application. The compression results are different for different image depend on the variety the colour. For example Blackbuck.BMP after compression process has higher PSNR and UIQI than Lena.BMP and Baboon.BMP. This is because Blackbuck.BMP has black background. Table 5 and 6 show the measurement results of PSNR and UIQI among original files with compression and decompression files on desktop computer and mobile phone.

Table 5 Measurement PSNR among original files with compression and decompression files.

Quality Factor	Measurement PSNR between Original vs Compression and Decompression Files in Computer						Measurement PSNR between Original vs Compression and Decompression Files in Mobile Phone					
	Compression File			Decompression File			Compression File			Decompression File		
	Lena	Baboon	Blackbuck	Lena	Baboon	Blackbuck	Lena	Baboon	Blackbuck	Lena	Baboon	Blackbuck
85	38.952	34.398	47.287	38.952	34.398	47.287	36.384	31.203	43.343	34.620	28.364	41.972
80	37.615	32.630	46.216	37.615	32.630	46.216	35.495	29.844	42.604	34.155	28.039	41.508
75	36.702	31.376	45.369	36.702	31.376	45.369	34.845	28.853	42.653	33.838	27.698	41.114
70	36.117	30.509	44.661	36.117	30.509	44.661	34.401	28.174	41.479	33.509	27.384	40.775
65	35.601	29.777	44.099	35.601	29.777	44.099	33.984	27.585	41.467	33.261	27.060	40.390
60	35.178	29.185	43.525	35.178	29.185	43.525	33.636	27.111	40.431	33.003	26.770	39.995
55	34.825	28.683	43.016	34.825	28.683	43.016	33.319	26.741	40.347	32.784	26.509	39.598
50	34.526	28.265	42.576	34.526	28.265	42.576	33.054	26.397	39.899	32.543	26.251	39.268
45	34.226	27.861	42.103	34.226	27.861	42.103	32.798	26.081	38.243	32.373	25.991	38.969
40	33.879	27.416	40.995	33.879	27.416	40.995	32.422	25.719	38.595	32.082	25.682	38.488

Table 6 Measurement UIQI among original files with compression and decompression files.

Quality Factor	Measurement UIQI between Original vs Compression and Decompression Files in Computer						Measurement UIQI between Original vs Compression and Decompression Files in Mobile Phone					
	Compression File			Decompression File			Compression File			Decompression File		
	Lena	Baboon	Blackbuck	Lena	Baboon	Blackbuck	Lena	Baboon	Blackbuck	Lena	Baboon	Blackbuck
85	0.893	0.951	0.909	0.893	0.951	0.909	0.778	0.909	0.960	0.758	0.883	0.943
80	0.859	0.934	0.902	0.859	0.934	0.902	0.728	0.886	0.956	0.711	0.866	0.940
75	0.825	0.921	0.904	0.825	0.921	0.904	0.688	0.864	0.907	0.674	0.850	0.934
70	0.799	0.909	0.903	0.799	0.909	0.903	0.657	0.846	0.948	0.647	0.835	0.931
65	0.774	0.898	0.901	0.774	0.898	0.901	0.632	0.827	0.903	0.623	0.820	0.926
60	0.750	0.887	0.898	0.750	0.887	0.898	0.610	0.809	0.938	0.603	0.805	0.922
55	0.728	0.877	0.901	0.728	0.877	0.901	0.592	0.794	0.899	0.586	0.791	0.917
50	0.708	0.868	0.897	0.708	0.868	0.897	0.579	0.778	0.885	0.571	0.777	0.914
45	0.687	0.858	0.895	0.687	0.858	0.895	0.566	0.763	0.826	0.561	0.762	0.911
40	0.662	0.846	0.847	0.662	0.846	0.347	0.549	0.743	0.822	0.544	0.743	0.905

We also simulate to transmit the JPEG file as original file on mobile phone sender to computer server successfully. The JPEG image file as the original file can be reduced the size by using setup the quality factor. Then we also perform measurement to compare PNSR and UIQI between original file and increasing the quality factor of merged file converting to other format and decompression. The measurement results show that the quality of the original file decreases slightly after merging all the files. This proves that the PSNR for split-files before and after sending is identical but after merge the file PSNR between the compressed image on the client and merged image on the server is not identical. To reconstruct the merged with increased quality factor, convert the files to other format and decompression to BMP (original image without compression) the retain the quality of the original one. Decompressing the merged files, the size will be larger because the image is changed into the early form of the file without compression. The amount can be known by calculating pixel:

$$height\ x\ pixel\ width\ x\ number\ of\ bits\ per\ pixel\ (RGB\ equal\ 24\ bits\ or\ 3\ bytes) \tag{5}$$

If we want to decompress Lena JPEG which has file size of 512 x 512, it will generate file size of 512 x 512 x 3 = 786432 bytes and added the header of the image size that is 128 bytes to be 786560 bytes. Table 7 shows the measurement results of the PSNR and UIQI for JPEG image file as original image which is sent to computer server.

Table 7 the Measurement results of the PSNR and UIQI for JPEG image file.

Quality Factor (%)	Lena.JPEG (512x512) size= 69214 Bytes			
	PSNR		UIQI	
	On the Client	On the Server	On the Client	On the Server
	After Compression	After Merging & Decompression	After Compression	After Merging & Decompression
85	+36.38447	+34.62010	0.77797	0.75838
80	+35.49460	+34.15548	0.72814	0.71070
75	+34.84501	+33.83808	0.68763	0.67376
70	+34.40077	+33.50887	0.65749	0.64722
65	+33.98426	+33.26122	0.63167	0.62330
60	+33.63648	+33.00327	0.61033	0.60344
55	+33.31938	+32.78432	0.59155	0.58608
50	+33.05356	+32.54285	0.57922	0.57149
45	+32.79801	+32.37335	0.56579	0.56123
40	+32.42172	+32.08210	0.54850	0.54380

To measure the data quality after the reconstruction process, we use some indicators such as peak-signal to noise ratio (PSNR) and universal image quality index (UIQI). The PSNR can be defined as follow [22]:

$$\mathrm{PSNR} = 10 \log \frac{(2^{\mathrm{n}} - 1)^2}{\mathrm{MSE}} \tag{6}$$

Where

$$\mathrm{MSE} = \frac{1}{\mathrm{MN}} \sum_{\mathrm{j}=1}^{\mathrm{M}} \sum_{\mathrm{k}=1}^{\mathrm{N}} \left(\mathrm{X}_{\mathrm{j,k}} - \mathrm{X'}_{\mathrm{j,k}}\right)^2 \tag{7}$$

To calculate universal image quality index (UIQI) can be defined as follow [23]:

Let x= $\{\chi_i \mid i = 1, 2, 3, \ldots N\}$ as Original image (8)

Let y= $\{y_i \mid i = 1, 2, 3, \ldots N\}$ as reconstructed or corrupted image (9)

The proposed quality index is defined as:

$$\mathrm{UIQI} = \frac{4\,\sigma_{xy}\,\bar{\chi}\,\bar{\gamma}}{\left(\sigma_\chi^2 + \sigma_y^2\right)\left[(\bar{\chi})^2 + (\bar{\gamma})^2\right]} \quad (10)$$

Where

$$\bar{\chi} = \frac{1}{N}\sum_{i=1}^{N}\chi_{i,} \qquad \bar{\gamma} = \frac{1}{N}\sum_{i=1}^{N}y_{i,} \quad (11)$$

$$\sigma\frac{2}{\chi} = \frac{1}{N-1}\sum_{i=1}^{N}(\chi_i - \bar{\chi})^2, \quad \sigma\frac{2}{y} = \frac{1}{N-1}\sum_{i=1}^{N}(y_i - \bar{\gamma})^2 \quad (12)$$

$$\sigma_{xy} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{\chi})\,(y_i - \bar{\gamma}) \quad (13)$$

We hope this research can provide the knowledge about the MMS technology for developing client-server applications. There are some researchers who have developed the technology to process MMS telemedicine [24, 25], but no one has done the improvement of the MMS technology using the combination of compression and splitting techniques. We have done simulations to verify the built application which can be run-well. The application is still a prototype system that opens a great opportunity which will be developed in further.

5 Conclusion and Future Work

In this study, we develop a client-server application using MMS technology. The client application implements the compression, splitting, masking and cropping techniques that are installed on the mobile phone. For server side, we develop a telemedicine application that receives data from the user in mobile environment to be processed on the computer server. We implement the merge and decompression techniques to reconstruct the data which is almost similar to the original file. The reconstructed data is then stored into the database server.

We have presented techniques that can be used for transmitting large multimedia files using MMS technology. Using simulation experiments, we have proved that the proposed techniques can be easily implemented in a mobile environment and can be used to develop client-server application. We hope that this research can provide contributions to the development of a new MMS framework for Internet applications. Our future research will be focused on the development of adaptive and efficient MMS frameworks for Internet applications based on the techniques proposed in this paper.

References

1. Daniel R. P. G., MMS Technology, Usage and Business Models, Jhon Wiley & Sons Ltd, the Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England, 2004.
2. Miraj E. M., MMS-The Modern Wireless Solution for Multimedia Messaging, Nokia Corporation, Visiokatu 3, 33720 Tampere, Finland, pp. 2466 – 2472, Feb 2002.
3. Open Mobile Alliance, Multimedia Messaging Service Requirement Candidate Version, OMA-MMS-ENC-V1_3-20080128-C, Approved Version 1.3, 2008
4. Open Mobile Alliance, Multimedia Messaging Service Encapsulation Protocol, OMA-MMS-ENC-V1_2-20050301-A, Approved Version 1.2, 2005.
5. Stephane C. and Guido G., Multimedia Adaptation for the Multimedia Messaging Service, IEEE Communication Magazine, July 2004.
6. Majid G. and Srinivasan K., Multimedia Messaging Service: System Description and Performance Analysis, School of Computer Science University of Waterloo, Waterloo, ON N2L 3G1, Canada, 2004.
7. Gwenae¨l L. B., Mobile Messaging Technologies and Services SMS, EMS, MMS, Second Edition, Jhon Wiley & Sons, Chichester, England, 2005.
8. Gwenae¨l L. B., Multimedia Messaging Service: An Engineering Approach to MMS, Jhon Wiley & Sons, Chichester, England, 2003.
9. Miraj E. M., Improved Implementation Solution and General Mobile Network Architecture for Interworking between MMS and Streaming, International Journal of Communication Systems, 2005.
10. Tinku A. and Ping-Sing T., JPEG2000 Standard for Image Compression: Concepts, Algorithms and VLSI Architectures, A Jhon Wiley & Sons, Canada, 2005.
11. Linawati and Henry P. P., Perbandingan Kinerja Algoritma Kompresi Huffman, LZW dan DMC pada Berbagai Tipe File [Comparative Performance of compression algorithm of Huffman, LZW and DMC on Different Types of Files], Vol. 9 No. 1, pp. 7-16 (2004).
12. Jhon C. R., Image Processing Handbook Fifth Edition, CRC Taylor and Francis, USA, 2007.
13. IM2023 Multimedia, Kompresi Citra [Image Compression], Fakultas Teknik Informatika, Universitas Kristen Duta Wacana, Yogyakarta.
14. Syed A. K., the Discrete Cosine Transform (DCT): Theory and Application, Department of Electrical and Computer Engineering, Michigan State University, 2003.
15. David S., Data Compression, the Complete Reference, Third Edition, Springer, USA, 2004.
16. Nelson M. and Gailly J. L., The Data Compression Book, Second Edition, M&T Books (1996).
17. Asadullah S., Muhammad M., Muniba S. M., A System Design for a Telemedicine Health Care System. International Multi Topic Conference, 2008.
18. Tahat A. A., Mobile Personal Electrocardiogram Monitoring System and Transmission Using MMS, Proceedings of the 7th International Caribbean Conference on Devices, Circuits and Systems, April 28-30, 2008.

19. Alireza G., Tomoya E., and Makoto T., A Multi-Source Streaming Model for Mobile Peer-to-Peer (P2P) Overlay Networks, Spriger-Verlag, pp. 122-131, 2008.
20. Govind S. T., Ganish S., and Vishal G., Multimedia Streaming Technology 4G Mobile Communication Systems, International Journal on Computer Science and Engineering Vol. 02, No. 03. pp. 695-699, 2010.
21. Andik S., Jahangir M. A., Raed A. A., Evaluation and Analysis of Multimedia Messaging Service (MMS) in Message Delivery, CITA'09 University Malaysia Serawak, pp. 241-245, July 2009.
22. Athi N. S., Picture Quality Measures Calculation. http://www.ksrce.ac.in/
23. Zhou W., A Universal Image Quality Index, IEEE Signal Processing Letters, Vol 9, No. 3, 2002.
24. Cheng W., Ming-Feng Y., Kuang-Chiung C., and Ren-Guey L., Real-Time ECG Telemonitoring System design with Mobile Phone Platform, Elsevier, pp. 463-470, 2007.
25. Ching S. W., Hunang J. L., and Tong Y. C., Mobile Telemedicine Application and Technologies on GSM, IEEE, 2007.