

# Implementation of Mobile Internet E-mail & Chatting Program using WIPI Platform

**Byung-Kon Hwang\***

\*School of Computer & IT, Dae-gu University

E-mail : [bkhwang@daegu.ac.kr](mailto:bkhwang@daegu.ac.kr)

***Abstract** - In this paper, we implemented an e-mail and chat program (client) for communication between mobile devices and PCs based on WIPI, a South Korea's open wireless platform form developing device and content independent software to break from the way of developing contents that is dependent on different wireless platforms to each mobile communications company.*

*While traditional e-mail and chat programs are written for PC-to-PC communication, we develop them for mobile-to-PC and mobile-to-mobile communications, and we expect to see various mobile contents in the future.*

## 1. Introduction

Recent days e-mail and chatting become parts of our lives. Due to the free of restricted space, the defect of the traditional internet platform , the rate of using wireless internet is growing day by day. That is, now we can send e-mails or chat with friends at anytime and anywhere on wireless internet platform.

Each mobile communication company has adopted different wireless application protocol(WAP, ME) and the development environments for CP of each company is also varies, however, it is not easy for the wireless internet to be revitalized [3-5].

With increase of amount used of wireless internet the mobile communication companies start adopting the base infra (CDMA, 20001x, W-CDMA); therefore, the positive propulsion of the standardization at the wireless internet market is necessary. To be specific, the development of the two seconds and third infra that service high speed packet transmission from the second generation mobile communication that serviced low level transmission quality makes multimedia services possible.

The confrontation to standardize that is earnestly demanded, therefore, WIPI(Wireless Internet Platform

for Interoperability), which is South Korea's standard wireless internet platform accepting various languages and application programs and satisfying functional requirements that are performance, scalability, stability and etc, was proposed by mobile communication companies and contents manufactures under government control.

This paper is to implement sending and receiving e-mails and chatting services among PCs and wireless internet platform with WIPI.

## 2. The theory of Mail / Chat services

### 2.1 Technique of send-recvieve between Mail server and E-mail

#### 2.1.1 Mail Server

A mail server is a server within network that works as a virtual post office. E-mail is generally delivered by two kinds of agent programs and protocol. The two kinds of agent programs are MUA(Mail User Agent) and MTA(Mail Transfer Agent); the former is a program, used for users to write and send mails, and includes Netscape messenger, MicroSoft Outlook, etc. the latter is a program, transferring e-mail messages

from one computer to another, and includes Sendmail, Qmail, etc. [5].

In SMTP, POP3, IMAP, etc. SMTP(Simple Mail Transfer Protocol) is a protocol, which sends mails to another server or receive them from other servers. POP3(Post Office Protocol) and IMAP(Internet Mail Access Protocol) are used for clients to receive mails that are transmitted by SMTP.

### 2.1.2 Technique of send-receive

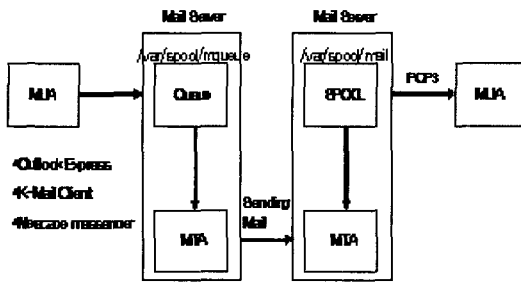


Figure 1. Process of mail send-receive.

When a user write and send a mail to mail server on MUA, the mail is queued in mqueue(Mail Queue) of mail server and transmitted to external mail server by MTA. Then the mail server is able to receive the mail by MTA and the mail is stored to a recipient's account. Figure 1 shows the process of mail send-receive in which the mails are sent and received among external servers by using SMTP protocol. Besides, when MUA is used for receiving them, POP3 or IMAP is used.

Table 1 shows the difference between POP3 and IMAP. In order to transmit e-mails, first the messages are encoded by Base64 with the form of MIME(Multi-purpose Internet Mail Extensions), then they are transmitted by SMTP. The MIME is the standard among groups of internet protocols for supporting multimedia of e-mail. That is, it's protocol to solve the problem of internet mail standard protocol, which only processes 7 bit ASCII characters, therefore, it can process 8 bit characters and various binary files by mails. The Base64, an encoding method to transform from binary data to ASCII texts or vice versa, makes 24 bit characters with three of 8 bit characters and

divides them by 6 bit in order to place each of them in the range between 0x00 and 0x3F. Then four units of 7 bit character are to be transformed. As explained, the SMTP is a standard protocol between servers and clients to send mails[7].

Table 1. Difference between POP3 and IMAP.

	POP3	IMAP
Merits	Needless to contain connection in order to work with messages.	Needless to download messages and store message to server.
Demerits	If download messages, impossible to work the downloaded data with another computer.	Need to connect to the server during working with mails

So as to receive an e-mail, the applicable instructions are transmitted to a server after connecting with an ID and password then the received mail is to be brought.

The received mail is decoded by the Base64. The procedure of bringing the received mail from the POP3 server with the execution instructions for the POP3 services is as follows. First a user connects to the POP3 server with an ID and a password, then the POP3 server sends a message, "greeting(connection confirm)", after the connection is completed. Besides, the user can send the instruction that asks how many currently unread mails are received in the server, then the server returns the information to the client. In addition the user is able to know the number and size of currently received mails with the LIST instruction(requirement of mail lists) The "RETER message # (requirement of a mail with its message #)" is sent to the POP3 server and makes a client receive the proper mail. The "QUIT(termination)" is sent to terminate the connection with the server.

Table 2. Execution instructions for the POP3 services

Instruction	Arguments	Functions	Responses from Server
USER	User ID or mail address	Response Outcome after searching	+OK, -ERROR
PASS	Password	Deliver user's password to the mail server	+OK, -ERROR
STAT	Nothing	Request mail information	+OK5 1000, -ERROR
LIST	Nothing or message #	Request information of each mail	+OK31000(1K byte) -ERROR
RETR	Message #	Request message transmission	+OK100 octets

※ Directions

( When communicating with the POP3 server, the packets must contain 'enter' string("\r\n") at the end of each packet.

The POP3 server also puts the 'enter' string at the end and send with them. )

### 2.2 The procedure of chat services

The socket of server differs from the one of clients; the former waits external connection passively on the other hands, the latter tries to connect actively. Moreover, being connected to only to only one server, the client uses only one temporary port; however, the server cannot support other clients as one client already occupies a socket. Therefore, the server socket needs to return a new socket when connecting at the outside and lets existing port be used continuously. It is necessary that a multi-thread method be used to support many clients. In order to make pairs of sockets, the server needs to wait for connecting with clients at the specific ports. This kind of sockets used at the server is the role that the ServerSocket class does in Java.

The connection flow between a server and a client using a socket as shown in the figure 2 is illustrated as

follows. The server first generates a socket, waits to connect and sends data from the client after linking the client to the server socket. Then data sent from the server is displayed and finally the socket connection is to be released.

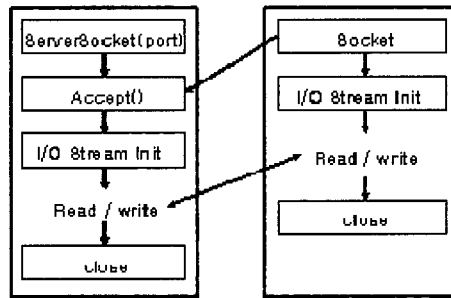


Figure 2. Connection between a server and a client using a socket.

### 3. Jlet based WIPI

The WIPI, an API for application programmers, supports C language as well as Java. The mobile mails are developed by inheriting the Jlet. The Jlet is an application program running on the WIPI environment, and it has a life cycle as shown in the figure 3. Besides, it should be inherited org.kwis.msp.lcdui.Jlet and methods are invoked for particular conditions.

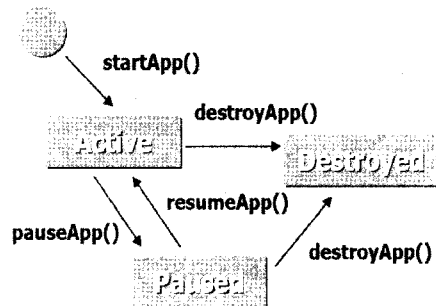


Figure 3. Life cycle of Jlet.

Generally the methods to be implemented when we develop the Jlet program are the ones in the figure 4. A startApp() is called as the Jlet is started, an args[0] is the name of Jlet and an args[1] and above become the arguments that users input. A pause() is invoked as the program is temporarily interrupted and a resumeApp()

is called as the interrupted program is resumed. The last method, a destroyApp(), is called as the program is terminated. If the argument bool becomes true, the Jlet has to be terminated right away, but if not, the program can deny the termination by generating JletStateChangeException.

```

import org.kwis.msp.Icdul.*;
Public class MobileMail extends Jlet {
    protected void startApp(String args[]) {}
    protected void pause App() {}
    protected void resumeApp() {}
    protected void destroyApp(boolean bool) {}
}

```

Figure 4. Inheritance of Jlet for Mobile Mail.

## 4. Implementation of e-mail / chat service among mobiles and PCs

### 4.1 Implementation of e-mail among mobiles and PC

As we can see an entire diagram in the figure 5, e-mails are received and sent with a mail server through a mobile and a mobile server.

Similarly webs save the contents of mails at the mail server and the messages travel via the mobile mail server then mobiles can receive them.

The mobile mail server transforms messages got from mobiles into data that the mail server is able to receive.

The figure 6 shows a diagram of mobile mail server which packetizes the messages sent by the mail server into the type of message that the mobile can receive then send them to the mobile.

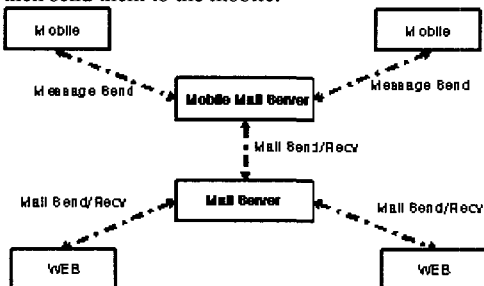


Figure 5. Entire Diagram of Mobile Mail among PC and Mobiles.

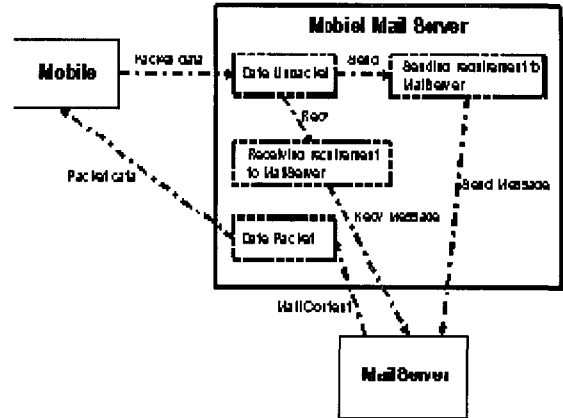


Figure 6. Diagram of Mobile Mail Server.

The program below shows the code indicating send-receive of mobile mail server.

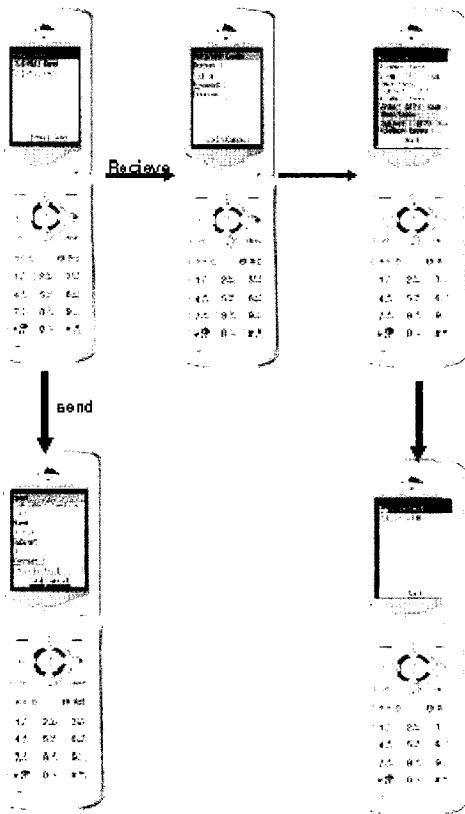
```

// Tokenize the data sent by socket communication
StringTokenizer st = new
StringTokenizer(msg, TOKEN);
String str = st.nextToken();
Int protocol = Integer.parseInt(str.trim());
switch(protocol) {
// Executes sendMail
    case SEND_MSG: // SEND_MSG
try{
// Connects and sends tokenized data with SMTP
MobileMailSend dMsg =
new MobileMailSend(st);
}catch(Exception e){}
break;

// Executes receiveMail
case MAIL_LIST: // Requests MailList
// Acquires the account information, which is to send
to pop3 server, from the tokenized data
account = st.nextToken();
password = st.nextToken();
try{
// connects and receives tokenized data with POP3
mailList(st);
}catch(Exception e){}
break;
} //end of switch

```

The figure 7 shows mobile screens implementing an E-mail send-receive client by using WIPI and we can see the contents of mails, brought from the mail server, on the screen.



The figure 7. Screens of mobile client.

#### 4.2 Implementation of chat services among mobiles and PC

The chat services diagram in the figure 8 shows the message send-receive between a mobile and a PC. The server packetizes messages of PC into the type of message that a mobile can receive, on the other hands, it also packetizes messages of type that a PC can receive in order to send them from the mobile to the PC.

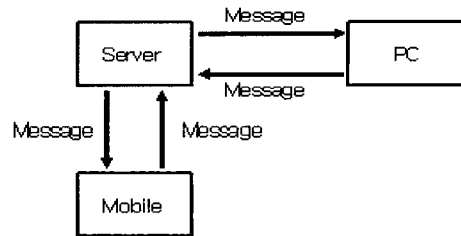


Figure 8. Diagram of Chat Service.

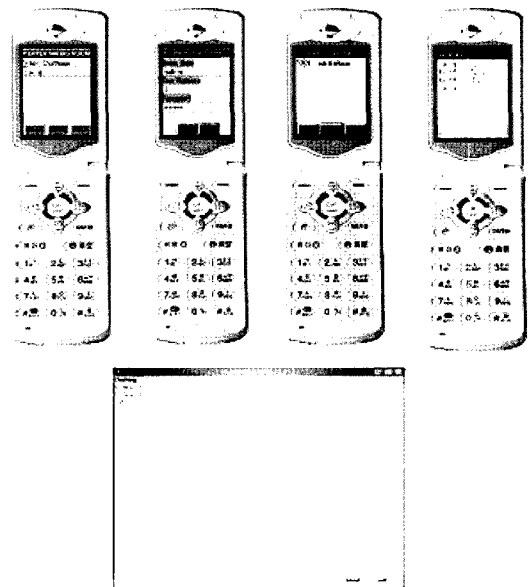


Figure 9. Screens of chatting between a PC and a mobile.

We can see a flow of chatting between a PC and a mobile client implemented by WIPI in the figure 9. At first a user logs in to the program in a mobile, chooses a chat room from the list and after connecting each other he can chat with the PC client.

### 5. Conclusion

Mobile contents used on the wireless internet that has been being developing rapidly are mainly characters for fun, melody downloads, games, e-mail send-recv or chat services. Therefore, this paper implemented e-mail send-recv between web and mobile, frameworks and operation of mail system and techniques of chat services between mobiles as a part of wireless mobile contents.

## 6. Reference

- [1] SukGu Kim, “WIPIMobile Programming”, Young Jin Presss, 2004
- [2] Choi-Sung , Suk Chun Park,”Mobile Contents Service and Diection of Progress”, Journal of Information Processing , 2002
- [3] Hyun uoo Lee, *Java Bible for JDK1*. Young Jin Dot Com, 2004.
- [4] redhat, *Linux Network & Web Server*, Gil But Press. 2003.
- [5] <http://www.kwisforum.org> .