

Developed Java Software For Switching Secured Short Message Service (SMS)

G. I. Ighalo¹, D. Mbama²
Ambrose Alli University, Ekpoma - Nigeria

ABSTRACT: *This work implements a software based SMS switching system using GLOBACOM and MTN as network carriers. JAVA programming language with focus on socket programming will be used for the software development. The application client, which is the portion that runs on the mobile phone, uses Java macro Edition, while the server section involved in the switching uses Java Servlet. When tested on the windows 7 operating system using oracle and the SUN mobile phone simulators, there was quick response and transmission of the SMS text from one end to the other. The application was similarly simulated on the UNIX based operating system such as Linux, and Solaris. There was high level of success on the simulation showing that the results are reliable. However the results could only handle up to 100 characters per SMS and a 64 bit encryption technique was used.*

Keywords: gateway; IP; Java; module; SMS

I. INTRODUCTION

Text messaging refers to the exchange of brief written messages between fixed-line phone or mobile phone and fixed or portable devices over a network. They are used to interact with automated systems such as ordering products and services for mobile phones or participating in contests. Advertisers and service providers use texts to notify mobile phone users about promotions, payment due dates and other notifications that can usually be sent by post, e-mail or voicemail.

The technical development of SMS was a multinational collaboration supporting the framework of standards bodies, and through these organizations the technology was made freely available to the whole world [1].

Short message Mobile Terminated (SMS-MT)/ Point-to-Point: the ability of a network to transmit a Short Message to a mobile phone. The message can be sent by phone or by a software application. There is also the Short message Mobile Originated (SMS-MO)/ Point-to-Point: these have the ability to transmit a Short Message sent by a mobile phone. The message can be sent to a phone or to a software application [2]

There is also the SMS gateway which is a way of sending a text message with or without using a mobile (cell) phone. Specifically, it is a device or service offering SMS transit by either transforming messages to mobile network traffic from other media or by allowing transmission or receipt of SMS messages with or without the use of a mobile phone. Typical use of a gateway would be to forward simple email to a mobile phone recipient. This is useful in developing web applications which interact with via SMS.

Some SMS gateway providers can be classified as aggregators or SS7 providers. The aggregator model is based on multiple agreements with mobile carriers to exchange 2-way SMS traffic into and out of the operator's Short Message Service Centre (SMS-C), also known as "local termination model". Aggregators lack direct access into the SS7 protocol, which is the protocol where the SMS messages are exchanged. These providers have no visibility and control over the message delivery, being unable to offer delivery guarantees. SMS messages are delivered in the operator's SMS-C, but not the subscriber's handset [3]

II. MATERIALS AND METHODS

The proposed SMS switching system is designed using adaptive routing technique. This routing describes the capability of a system, through which routes are characterized by their destination, to alter the path that the route takes through the system in response to a change in conditions. The adaptation is intended to allow as many routes as possible to remain valid (that is, have destinations that can be reached) in response to the change. This enables SMS message to be switching from the source network to the destination network if both are not on the same network. In addition to the adaptive routing technique, the design of the system is done using the software development life cycle (SDLC) which is IEEE standard for software development [4].

The SMS client hardware requirement include: -

- At least a 640K base memory
- Colour or chrome display
- Keypad/touch pad for text entering
- Wireless carrier SIM card,

Most client software requirement for the mobile phones to connect to the SMS switch is Java Virtual Machine (JVM) using Java Micro Edition. In addition, the mobile phone can run any mobile phone's operating system such as Android, Windows mobile, Symbian, etc [5]

The SMS switching would require a server with the following standard: -

- Pentium IV system or other compatible system
- 1.20GHz Processor capability
- 512 MB memory capacity
- 20 GB Hard disk space for the purpose of the system installation

In addition to the software required, the SMS switching would require the following for the application to be deployed:

- Windows Vista or higher or UNIX compatible system
- Java Virtual Machine (JVM)

Deploying the application

For this developed application to be used, it has to be deployed. Deployment of the client portion of the system; centred on how the application will be installed on the mobile phones. The procedure below gives the detailed description:

- The application is converted to deployable unit using the NETBEAN IDE 7.0
- The deployable unit can be transferred into the intended mobile phones using either USB cable or through the Bluetooth wireless.
- When the deployable unit has been successfully transferred, the installation can now be done. For instant on NOKIA mobile phones, browse to the location of the deployable unit on the mobile phone and select the unit and select install, follow the onscreen display to complete the installation. Successful installation means successful deployment. At this stage, the application is ready to be used on the mobile phone.

SMS switching deployment

The deployment of the actual SMS Switching system is done on the computer system that is intended to be the server. This is possible after the application has been made into deployable units. The conversion of the application into deployable units was done using NETBEAN IDE 7.0. Likewise, the application deployment was done using the same NETBEAN IDE 7.0. After successful deployment of the SMS Switching application on the server, the system is now ready to do switching of SMS messages received from the mobile phones through the client.

Operating the applications

The two applications are tied together using the java interconnectivity. In this regard, the applications mode of operation is outlined below:

- The application on the mobile phone is used to compose SMS message; which is sent to the intended receiver.
- The message is first sent to the SMS Switching application on the server. The SMS Switching verifies the intended destination network through the message header information. The message is then forwarded to the destination network SMC; which then routes the message to the intended mobile phone.

The protocol employed in the client and SMS Switch is the Internet Protocol (IP). This implies that the SMS Switching application requires hosting; for the purpose of this work, wireless interface between the mobile phone and the system will be used. Figure 2.1 shows the interconnectivity between mobile phone and a wireless interface.



Figure 2.1 Interconnection between mobile phone and SMS switching using Wireless Interface [3]

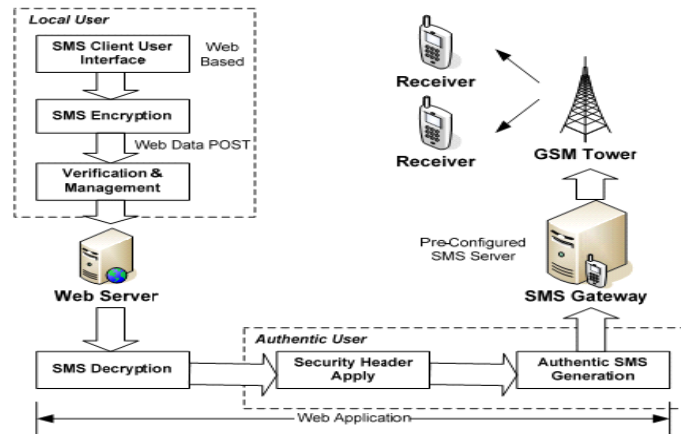


Figure 2.2 Software Level Execution Flow

Figure 2.2 shows the software execution process in which the client application shows that there are four modules for a successful secured SMS switching software process -

Interface module: The interface module is developed using J2ME; this is the module that displays the user interface on the mobile phone. The SMS composer is generated by this module.

Encryption module: This module adds security level to the application.

Verification and management module: This module verifies that all the needed information is included in the message. After successful verification, the information is handled over to the send module.

Send module: This is another J2ME code that collects the SMS message and other necessary information from the mobile phones and send them to the SMS Switching application. On its own the SMS switching mode has three stages – the server verification module, the search module and the send module respectively[6]

III. DEVELOPED SOFTWARE

The programming methodology used for the development of the SMS Switching system is Object oriented programming. Object-oriented programming (OOP) is a programming paradigm using "objects" – data structures consisting of data fields and methods together with their interactions – to design applications and computer programs.

Programme listing

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

import java.io.*;

import java.util.Enumeration;

import java.util.Hashtable;

import javax.microedition.io.*;

/**
 * @authorIghalo & Declan.
 */

public class SwitchingClient extends MIDlet implements CommandListener {

    private Display display;

    private Form mainForm;

    private Form startForm;
```

```
private TextBox smsUI;
private TextField sendToTextField;
private Command sendNow;
private Command sendBack;
private Command sendto;
private Command back;
private Command start;
private Command exit;
private String url="http://localhost/smsServer/smsServer.php?";
HttpConnection http;
InputStream in;
OutputStream out;
int rc;
public SwitchingClient(){
    super();
}
public void startApp() {
    display=Display.getDisplay(this);
    mainForm=new Form("SMS Switching Client");
    String intro="Welcome to SMS Switching Client application. Program written by Ighalo & Declan\n";
    intro=intro+"Department of Electrical/Electronic Engineering\n";
    intro=intro+"Ambrose Alli University, Ekpoma";
    start=new Command("Start",Command.OK,1);
    exit=new Command("Exit",Command.EXIT,2);
    mainForm.append(intro);
    mainForm.addCommand(start);
    mainForm.addCommand(exit);
    mainForm.setCommandListener(this);
    display.setCurrent(mainForm);
}
public void pauseApp() {
}
```

```
public void destroyApp(boolean unconditional) {
}

public void commandAction(Command c, Displayable d){
    if(c==start){
        startClient();
    }else if(c==sendto){
        sendTo();
    }else if(c==sendNow){
        try{
            String msg=smsUI.getString();
            String add=sendToTextField.getString();
            String data="message="+msg+"&address="+add;
            processGet(data);
        }
        catch(Exception o){
            o.printStackTrace();
        }
    }else if(c==back){
        display.setCurrent(mainForm);
    }else if(c==sendBack){
        display.setCurrent(smsUI);
    }
}

public void startClient(){
    //startForm=new Form("Send SMS");
    smsUI=new TextBox("SMS","",256,0);
    sendto=new Command("Send to", Command.OK,1);
    back=new Command("Back", Command.BACK,2);
    smsUI.addCommand(sendto);
    smsUI.addCommand(back);
    smsUI.setCommandListener(this);
    display.setCurrent(smsUI);
}
```

```
}  
  
public void sendTo(){  
    startForm=new Form("Add recipients separate with ;");  
    sendToTextField=new TextField("Recipients: ", "", 256, 0);  
    sendNow=new Command("Send SMS", Command.OK, 2);  
    sendBack=new Command("Back", Command.BACK, 1);  
    startForm.append(sendToTextField);  
    startForm.addCommand(sendNow);  
    startForm.addCommand(sendBack);  
    startForm.setCommandListener(this);  
    display.setCurrent(startForm);  
}  
  
public void processGet(String data) throws Exception{  
    //Thread t = new Thread();  
    //url=url+data;  
    //url=encoder(url);  
    http = (HttpConnection) Connector.open(url+data);  
    http.setRequestMethod(HttpConnection.GET);  
    http.setRequestProperty("IF-Modified-Since", "10 Nov 2006 17:29:12 GMT");  
    http.setRequestProperty("User-Agent", "Profile/MIDP-2.0 Configuration/CLDC-1.1");  
    http.setRequestProperty("Content-Language", "en-US");  
    in = http.openDataInputStream();  
    out = http.openDataOutputStream();  
    rc = http.getResponseCode();  
    if (rc != HttpURLConnection.HTTP_OK) {  
        throw new IOException("HTTP response code: " + rc);  
    }  
    int ch;  
    StringBuffer buff = new StringBuffer();  
    while ( (ch = in.read()) != -1){  
        buff.append( (char) ch);  
    }  
}
```

```
startForm.append(new StringItem("Response: ", buff.toString()));

if (in != null)
    in.close();

if (out != null)
    out.close();

if (http != null)
    http.close();
}

public String encoder(String url){
// url = "http://www.so.com?name1=value1&name2=value2&name3=value3";

Hashtable values = new Hashtable();

int s = url.indexOf("?");

int e = 0;

while (s != -1) {
    e = url.indexOf("=", s);

    String name = url.substring(s + 1, e);

    s = e + 1;

    e = url.indexOf("&", s);

    if (e < 0) {
        values.put(name, url.substring(s, url.length()));
    } else {
        values.put(name, url.substring(s, e));
    }

    s = e;
}

String key=null;

for (Enumeration num = values.keys(); num.hasMoreElements(); ) {
    key = (String)num.nextElement();

    System.out.println(key + " " + values.get(key));
}

return key;
}
```

IV. CONCLUSION

The short message service has been found to unify the Internet and mobile network. The software development is based on short messaging service (SMS) system for delivering messages through SMS gateway. The developed application was tested on Windows 7 operating system using Oracle and SUN mobile phone simulator. Likewise, the application can be simulated on UNIX based operating system such as Linux, Solaris, etc. This software was designed based on typical practical applications. From the various tests carried out, the designed software was found to be reliable and secure with 64 bit encryption.

REFERENCES

- [1] Hillebrand, Trosby, Holley, Harris (2010) SMS the creation of Personal Global Text Messaging, Wiley
- [2] Tomasi, Wayne (2004) Introduction to Data Communications and Networking, 1st ed. Upper Saddle River, NJ: Prentice Hall.
- [3] Gil Held (2001) "Data over Wireless Networks". Wiley, NJ
- [4] Gredle Rose (2005) GSM standards and services publication of IEEE, 2005
- [5] Rheingold Victor (2002). Mobile Computing Handbook. Artech House Mobile Communications Library. Artech House..
- [6] Sun Microsystems (2009), Java 2 Platform, Micro Edition (J2METM Platform). Available as <http://java.sun.com/j2me/>
- [7] Shannon Hills (2007) Introduction to Mobile Computation, 3rd ed., NJ: Prentice Hill