

Fault Management System

Nikhil Ramchandra Nipane

Marathwada Mitra Mandal's Institute of Technology
Pune, India

Ajinkya Vasant Patre

Marathwada Mitra Mandal's Institute of Technology
Pune, India

Amit Kailas Bhagwat

Marathwada Mitra Mandal's Institute of Technology
Pune, India

Rakesh Dattatray Shahane

Marathwada Mitra Mandal's Institute of Technology
Pune, India

Abstract— As increase in the demand for the telecommunication network is rising, there is also increase in problems to handle the vast network. Expansion of network causes faults which leads to disinterruption in the network availability. As a result, to handle these type of problems in the network, an implementation of a system is required which may analyze the faults and let know the concern people. The system should avoid the unavailability of the network upto some extent. Fault management system is an operation environment which deals with such faults. When the faults are identified they are filled in the FMS as per their occurrence. These faults are then handled according to their priorities. An SMS/Mail is generated to the concern people regarding the fault along with its detailed description. As a result these faults are then sorted and removed from system. Also a detailed report for unsorted faults can be seen for further analysis of network.

Keywords- Client Server architecture, Email Generation, SMS Generation, Report Generation.

I. INTRODUCTION

Fault management system manages the faults occurred in the telecommunication network. Mostly the telecommunication network is in the ring topology form. These rings join different cities in a circular form. In between these rings, a fault can occur which may lead in loss of network. These faults are identified by the NMC (Network Monitoring Center) and are given input to the FMS. The FMS further informs the detailed information of the fault occurrence to the concern people of the fault occurred region. It also generates report on different aspects.

Fault Management System is used to enhance the fault handling capabilities in the network and get a fault free network to increase the network performance. It records the faults occurred in the network along with its detailed description and also maintains the status and severity with respect to the fault.

These faults are interpreted to the respective people through mails and messages. The respective people identify and solve the fault accordingly and report the clearance of the fault. When the fault is sorted it is removed from the FMS. The FMS also generate the reports for analysis of unsorted faults in the network. In this way, a fault is recorded and handled to get sorted accordingly and removed from the FMS.

Faults are fed through FMS, with unique ID for each fault so that the same can be used for further analysis at a later stage and corrective/preventive measure can be taken to reduce fault rates.

NMC Team keeps constant watch and pursues with Indoor / Outdoor units, distant end stations, for early fault rectification. NMC updates the fault data and closes the fault in FMS, when found restored. FMS has the provision of booking all categories of faults like cable fault, system fault, traffic fault and equipment failures. In general, FMS does the tracking of fault restoration with periodical feedback, to all concerned, by updating the docket in FMS and removing when the fault is sorted.

II. MODULES REQUIRED

In FMS, functionality of the system can be divided into modules. There are some basic modules which together are implemented in the system. Each module governs its own significance.

The modules can be classified as:

A. Client – Server Architecture

The system is developed on client-server architecture in order to get accessed from any location and in distributed environment.

B. Mail Generation

The mails are generated by FMS to alert about faults occurred in the network. These mails can be generated with the help of SMTP protocol.

C. Message Generation (SMS)

SMS are generated to concern people describing the fault occurrence. SMS are generated with the help of modem.

D. Report Generation

Report generation is the additional functionality of the system. FMS allows to generate reports for faults occurred. The reports in general describe the fault which are pending and has been repaired.

The reports can be described on daily, weekly and monthly basis. These reports can be helpful for network analysis.

A. CLIENT – SERVER ARCHITECTURE

Client-server architecture of a computer network is in which many clients (remote processors) request and receive service from a centralized server (host computer). Client computers provide an interface to allow a computer user to request services of the server and to display the results the server returns. Servers wait for requests to arrive from clients and then respond to them. Ideally, a server provides a standardized transparent interface to clients so that clients need not be aware of the specifics of the system (i.e., the hardware and software) while providing the service. Today clients are often situated at workstations or on personal computers, while servers are located elsewhere on the network, usually on more powerful machines. This computing model is especially effective when clients and the server each have distinct tasks that they routinely perform. Client-server systems are constructed so that the database can reside on a central computer, known as a server, and be shared among several users

In hospital data processing, for example, a client computer can be running an application program for entering patient information while the server computer is running another program that manages the database in which the information is permanently stored. Many clients can access the server's information simultaneously, and, at the same time, a client computer can perform other tasks, such as sending e-mail. Because both client and server computers are considered intelligent devices, the client-server model is completely different from the old "mainframe" model, which utilized a centralized mainframe computer that performed all the tasks for its associated "dumb" terminals.

Users run an application on their local computer, known as a client that connects over a network to the server running SQL Server. The client application runs both business logic and the code to display output to the user. The client-server characteristic describes the relationship of cooperating programs in an application. The server component provides a function or service to one or many clients, which initiate requests for such services

Functions such as email exchange, web access and database access, are built on the client-server model. Users accessing banking services from their computer use a web browser client to send a request to a web server at a bank. That program may in turn forward the request to its own database client program that sends a request to a database server at another bank computer to retrieve the account information. The balance is returned to the bank database client, which in turn serves it back to the web browser client displaying the results to the user. The client-server model has become one of the central ideas of network computing. Many business applications being written today use the client-server model. So do the Internet's main application protocols, such as HTTP, SMTP, Telnet, and DNS

The Client-Server environment handles many requests at a time. In order to make it possible threads are created. One thread for each request is assigned. When the work is done the thread is closed. Threads are useful to create a distributed architecture. [1]

B. MAIL GENERATION

The javax.mail package defines classes that are common to all mail systems. The javax.mail.internet package defines classes that are specific to mail systems based on internet standards such as MIME, SMTP, POP3, and IMAP. Javax.mail.internet package supports features that are specific to Internet mail systems based on the MIME standard (RFC 2045, RFC 2046, and RFC 2047)[2][5]

The basic protocols used in the mail generation are classified as:

- * SMTP
- * MIME

SMTP

The Simple Mail Transfer Protocol (SMTP) is defined by *RFC 821*. It defines the mechanism for delivery of e-mail. In the context of your Java Mail-based program will communicate with your company or Internet Service Provider's (ISP's) SMTP server. That SMTP server will relay the message on to the SMTP server of the recipient(s) to eventually be acquired by the user(s) through POP or IMAP. This does not require your SMTP server to be an open relay, as authentication is supported, but it is your responsibility to ensure the SMTP server is configured properly.

MIME

MIME stands for Multipurpose Internet Mail Extensions. It is not a mail transfer protocol. Instead, it defines the content of what is transferred: the format of the messages, attachments, and so on. There are many different documents that take effect here: *RFC 822*, *RFC 2045*, *RFC 2046*, and *RFC 2047*. As a user of the Java Mail API, you usually don't need to worry about these formats. However, these formats do exist and are used by your programs.[4]

The code to send a plain text message can be as simple as the following:

```
Properties props = new Properties();
props.put("mail.smtp.host", "my-mail-server");
props.put("mail.from", "me@example.com");
Session session = Session.getInstance(props, null);

try {
    MimeMessage msg = new MimeMessage(session);
    msg.setFrom();
    msg.setRecipients(Message.RecipientType.TO,
        "you@example.com");
    msg.setSubject("JavaMail hello world example");
    msg.setSentDate(new Date());
    msg.setText("Hello, world!\n");
}
```

```
Transport.send(msg);
} catch (MessagingException mex) {
    System.out.println("send failed, exception: " + mex);
}
```

C. MESSAGE GENERATION (SMS)

1. Send SMS using AT commands

Some advanced GSM modems like WaveCom and Multitech, support the SMS text mode. This mode allows us to send SMS messages using AT commands, without the need to encode the binary PDU field of the SMS first. This is done by the GSM modem.

2. Check if GSM phone or modem supports SMS text mode

To check if modem supports this text mode, we can try the following command.

```
AT+CMGF=1 <ENTER>
```

If the modem responds with "OK" this mode is supported. Please note that using this mode it is only possible to send simple text messages. It is not possible to send multipart, unicode, data and other types of messages

3. Setting up the modem

If the modem contains a SIM card with is secured with a PIN code, we have to enter this pin code first:

```
AT+CPIN="0000" <ENTER> (replace 0000 with your PIN code).
```

Please not that in most cases we have only 3 attempts to set the correct PIN code. After setting the PIN code, wait some seconds before issuing the next command to give the modem some time to register with the GSM network.

In order to send a SMS, the modem has to be put in SMS text mode first using the following command:

```
AT+CMGF=1 <ENTER>
```

In text mode there are some additional parameters that can be set. Using the following command we can read the current values:

```
AT+CSMP? <ENTER>
```

The modem will response with a string like this:

```
+CSMP: 1, 169, 0,0OK
```

The first value is a combination of some option bits show in the table:

TABLE I: Option bits table

bit 7	RP	Reply path, not used in text mode
bit 6	UDHI	User Data Header Information
bit 5	SRR	Set this bit to request a delivery report
bit 3,4	VPF	Validity Period, set b4=1 if a VP value is present
bit 2	RD	Reject Duplicates, do not return a message ID when a message with the same destination and ID is still pending
bit 0,1	MTI	Message Type Indicator b1=0 & b0=0 -> SMS-DELIVER b1=0 & b0=1 -> SMS-SUBMIT

Bit 0 of the message is always set when sending messages (SMS-SUBMIT). So the first value should be 1 or higher. The second parameter sets the Validity Period of the message. This value is encoded as follows:

TABLE II: Validity Period of the message

0 – 143	(VP + 1) x 5 minutes
144 – 167	12 Hours + ((VP-143) x 30 minutes)
168 – 196	(VP-166) x 1 day
197 – 255	(VP-192) x 1 week

The third parameter contains the PID (Protocol Identifier). This parameter is only used for advanced messaging. The fourth parameter contains the DCS (Data Coding Scheme). This parameter is used to select the character set/message type. When setting the DCS parameter to '0' standard 7 bit text is send. When setting this parameter to '16' the message is sent as a flash message.

To send a message with a validity period of 1 day, the parameters have to be set like this:

Bit 0 and 4 of the first field has to be set, so the first value will become 1 + 16 = 17.

Send the following command to the modem to set this parameters:

```
AT+CSMP=17, 167,0,16 <ENTER>
```

If the modem responds with "OK" ,the modem is ready to send (flash) text messages with a validity period of 1 day.

4. Sending the message

To send the SMS message, type the following command:

```
AT+CMGS="+31638740161" <ENTER>
```

Replace the above phone number with your own cell phone number. The modem will respond with:

>

You can now type the message text and send the message using the <CTRL>-<Z> key combination:

Hello World! <CTRL-Z>

After some seconds the modem will respond with the message ID of the message, indicating that the message was sent correctly:

+CMGS: 62

The message will arrive on the mobile phone shortly.

5. Sending an Unicode SMS message

Some modems also have the capability to send unicode or UCS2 messages without encoding a PDU. You can send unicode messages by only converting the unicode data to a HEX string and send this string to the modem

To check whether your modem supports this mode, just type the following command:

AT+CSCS=?

This command displays the codepages supported by the modem. The modem will respond like this:

+CSCS: ("GSM","PCCP437","CUSTOM","HEX")

If this string contains "HEX" or "UCS2", unicode seems to be supported. To specify that you will use an HEX string to send the message, set the codepage to "HEX" or "UCS2" depending on the modem response. In our example we will set the modem to "HEX" :

AT+CSCS="HEX" <ENTER>

Next, we have to specify the correct DCS (Data Coding Scheme) for Unicode messages, which is 0x08. We can set this value by changing the fourth parameter of the AT+CSMP command to '8':

AT+CSMP=1, 167, 0, 8 <ENTER>

The modem is now ready to send messages as Unicode. Now is the time to send the actual message:

AT+CMGS="+31638740161" <ENTER>

Replace the above phone number with your own cell phone number. The modem will respond with:

>

The only thing you have to program by yourself is a simple routine which converts the Unicode string to a hexadecimal string like this:

ابحرم

Which is 'Hello' in Arabic will be converted like this:

"06450631062D06280627"

You can send this hexadecimal string to the modem:

06450631062D06280627 <CTRL-Z>

After some seconds the modem will respond with the message ID of the message, indicating that the message was sent correctly[3]

D. REPORT GENERATION

Report generation helps to analyse the network stability. The system consists the records of sorted and unsorted faults. These fault records are arranged in detailed manner in the form of reports. The reports consist of repaired and unrepaired faults in the network which helps to understand the pending faults and repaired till now.

Reports can be generated on the basis of daily, weekly, and monthly generations.

TABLE III :Example of a fault Report

Sr. No	Ring No /Name	Section	Status: APS/Interrupti on	Time	Date
1	W-496	RAIGARH -RANCHI	APS	2:42	8/4/2011
2	W-497	RANCHI-DALTON GANJ	INTERRUPTI ON	11:02	8/5/2011
3	W-551	NP-PDI	APS	11:24	8/5/2011
4	WR 16 (457)	NP-HYD	APS	18:35	8/4/2011

The above table shows a basic report format generated from the system. The table describes the ring no., section, status of the ring, time of fault occurred, and date of occurrence. The status shows the APS (Auto Protection Switch) and interruption mode.

III. CONCLUSION

Fault Management System (FMS), a package has been developed to take care of the activities like near real time fault booking, unique ID (Docket) for each fault, escalation through SMS, details of traffic affected, fault closure, report generation to achieve the goal of optimum utilization and fault free

maintenance of networks. It is also an effective tool for planning and taking management decisions regarding augmentations and reengineering of systems.

REFERENCES

- [1] Andrew S. Tanenbaum, Maarten Van Steen , "Distributed System Principles and Paradigms", 4th Edition,Prentice Hall.
- [2] Javax.mail Package
(<http://docs.oracle.com/javase/5/api/javax/mail/package-summary.html>)
- [3] Send SMS using AT commands
(<http://www.smssolutions.net/tutorials/gsm/sendsmsat/>)
- [4] JavaMail 2: (<http://cloudcomputing.syscon.com/node/1203943>)
- [5] Package javax.mail.internet
(<http://docs.oracle.com/javase/5/api/javax/mail/internet/package-summary.html>)