Tool Description - Mobeon SMSC Simulator Version: R10A

Content

History

Version	Date	Adjustments
PA1	2004-09-17	
PA2	2004-12-10	Added parameters responsedelay and dropevery
PA3	2006-01-04	Added result code control and web interface description.
PA4	2007-04-25	Added cancel functions and description of cyclic error simulation. Update log file description.

1 Introduction

The Mobeon SMSC Simulator simulates an SMSC. It understands the SMPP protocol, but can not send SMS anywhere. On the other hand it has some extra features for testing. The simulator has an HTTP interface so you can control and monitor it with a web browser. It is designed so that a single instance of the simulator can serve many test labs, but you can run your personal simulator if you prefer.

Each test lab is an ESME (SMPP language for some system that can send SMS to an SMSC). Each ESME can have many accounts, e.g. one for NTF, one for MEC and one for MWS. Each ESME is independent of the others; it has its own accounts, its own log files and its own web pages.

2 Functions

2.1 SMPP Functions

The simulator allows clients to log in and send SMPP submit requests (i.e. send SMS) and SMPP cancel requests.

It sends receipts to SMS type 0 requests after a configurable delay, simulating that the phone is on.

2.2 Management Functions

A web interface allows you to configure and control the behaviour of your ESME, view statistics about messages sent and to view individual messages. The simulator understands the meaning of many parameters and can say that an SMS is e.g. An MWI on request.

2.3 Error Simulating Functions

There are many functions available to simulate error conditions in the SMSC or network.

In the web interface, you can make the simulator respond to one or all SMPP requests of a certain type to your ESME, with a selected error code.

You can also make the simulator respond to all messages to the simulator with message queue full, delay the response, or drop the response altogether, with settings in a configuration file. This affects all ESMEs in the simulator.

You can also simulate an SMSC that repeats a cycle where it crashes and comes up again, crashes and comes up again… The crash is simulated by dropping all connections and refusing new ones. This feature is controlled by the connerr* parameters in the configuration file. This affects all ESMEs in the simulator.

Finally you can simulate and SMSC that repeats a cycle where it stops responding ad the starts responding again. This feature is controlled by the responseerr* parameters in the configuration file. This affects all ESMEs in the simulator.

3 Installation

To install the simulator, unpack *smsc_r10a.xxx.tar* anywhere you like:

```
> tar xf smsc_r10a.xxx.tar
```

4 Files

The file structure when you run the simulator is in the directory *smsc* where the tar-file was unpacked.

There are subdirectories *bin, cfg* and *logs*.

The *bin* directory contains jar files with java classes, and the startscript *rc.smsc*.

The *cfg* directory contains the configuration file *smsc.cfg*.

The *logs* directory is initially empty, and later when the simulator is running, contains the files:

HttpServer.0.log	Logs all connections and HTTP requests on theHTTP interface
SMSCProcess.log	Not so useful. Logs header information for all SMPP requests
smpp.0.log	Logs start, connections and some general debug messages. Logs the bytes of all SMPP messages sent and received. Switches to smpp.1.log etc. when the file grows.
smsc.pid	Contains the pid of the running smsc process.

Each ESME or lab also has its own log subdirectory that contains the files:

sms.log	The most useful log. Logs the SMPP messages in readable format, the same data that is shown in the web interface.
bind.0.log	Logs binds and debug messages per ESME.

Thus, if your lab is hurr11 and you have created an ESME called hurr11 using the SMSC simulator's web interface, the log file <smsc_home>/logs/hurr11/sms.log will contain all SMS messages that have ever been sent from the NTF, MEC and MWS in hurr11, so that you can do end-to-end verification of the SMS notifications for each message.

5 Configuring

The following parameters can be set in *smsc/cfg/smsc.cfg*

SMSCServerPort=5016	The port used for SMPP
HttpServerPort=8080	Port number for the administration interface

LogLevel=ALL	How much log messages to print
LogCount=3	How many generations of log files to keep
LogSize=1000000	The size of each log file
sms0delay=30	How long to wait before returning a receipt if that was requested when an SMS was sent.
responsedelay=0	How long after the submit request, a submit response is sent (milliseconds).
dropevery=0	How often the SMSC fails to respond to a submit request. 0 means do no drop any responses. 10 means drop every tenth response.
percenterror=0	How many percent of submit responses shall have the status code 0x14 (message queue full).
connerrstart	How long after start of the SMSC that cyclic connection error simulation shall start.
connerrok	How long the connection shall be OK in each cycle.
connerbad	How long the connection shall be bad in each cycle.
responseerrstart	How long after start of the SMSC that cyclic response error simulation shall start
responseerrok	How long the responses shall be OK in each cycle.
responseerrbad	How long the responses shall be bad in each cycle.

There is also a configuration file called *smsc/cfg/.labs.xml*. This file contains the configuration of the different labs (ESMEs). It is edited with the web interface, and is thus hidden from normal view (name starts with a dot.)

6 Start and Stop

To start the simulator, run:

```
> smsc/bin/rc.smsc start
```

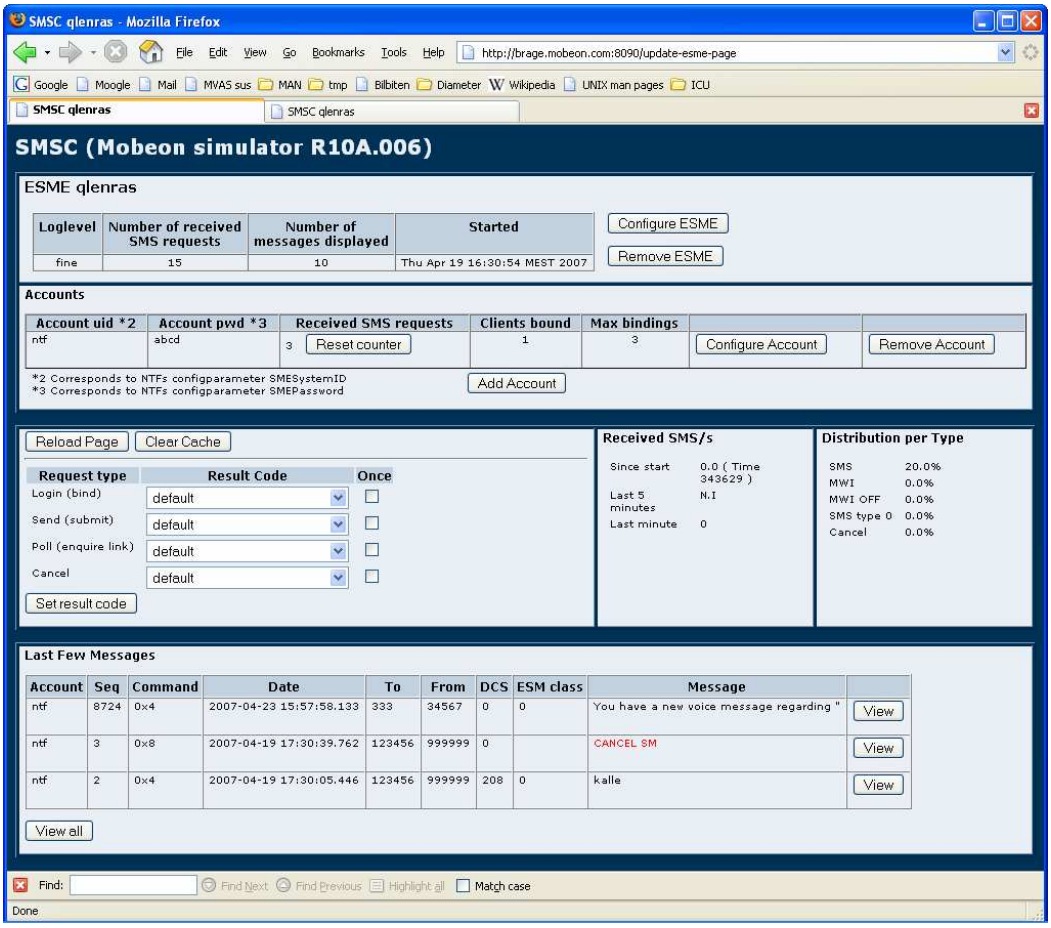
Then go to port 8080 on the simulator host with a web browser. This will give you a web site where you can create more ESMEs, enter an ESME to configure it and watch the messages it receives.

To stop the simulator, run:

```
> smsc/bin/rc.smsc stop
```

7 Web Interface

This is an example of the main window in the web interface:



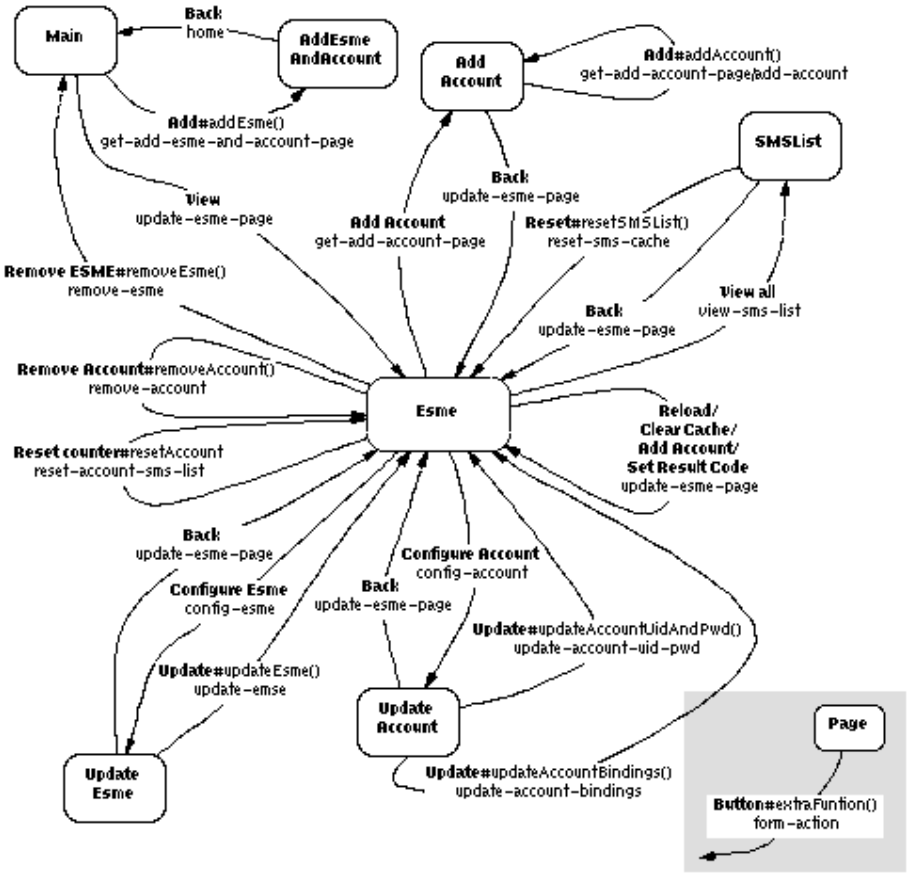
At the top, there is information about the selected ESME and below that, there is information about the accounts for that ESME. Here you find settings and the total number of messages received for the ESME and each account.

In the middle there is one box with the controls you normally use. There is a button to reload the page to see if there have been new messages and one button to empty the list of SMS messages in the bottom of the screen. There you also can set the result codes for different types of requests. A result code can be valid from now on, or just once (for the next request). To the right of the control box, there are two boxes with information about the traffic rate and distribution among different types of messages.

At the bottom, there is a list of the latest messages (the size is controlled by the ESME settings). To view more detailed information about a message, click the *View* button to the right. To view details for all messages, click the *View all* button.

The picture below describes how you can move between different pages. The bold text on the arrows is the button click that takes you from one page to another.

The picture is made for development of the simulator and may or may not be useful to understand the user interface.



8 Terminology

Term	Explanation