

Robo Tester Evolution (RTE) Automation Tool for GSM Tariff Tests

Orhan Orhan and Esmâ Akbey

Nokia R&D Department, Organize San. Bolgesi 2. Cadde No: 17/1
Umraniye Istanbul 34776, Turkey

Abstract

In this work an automation tool for testing and verification of Nokia's Surepay GSM tariff and promotion management system introduced. The advantages of this RTE tool can be counted as elimination of manpower required to make tests, reduced error rate and decreased time to finish test scenarios. Our solution uses diameter protocol to connect related network elements and LDAP database used to collect test scenarios to be performed. Our solution makes it possible to use Android GW to make real calls by using Android handsets that are connected to the test environment via USB connectors. The important advantages of RTE can be counted as low cost, fast results and low error rate in performing tests.

Keywords: Robo Tester Evolution (RTE), Surepay, GSM, Tariff, Automation.

Introduction

Because of increased competition between GSM operators, they are continuously developing new promotions and new tariffs, and they try to serve with these tariffs to the customers in a fastest possible way. Nokia's (former Alcatel-Lucent) surepay tariff and promotion management system integrated with the OSS/BSS system and can manage subscribers and services on various different platforms. In order to test the Surepay tariff management system we developed an automated test application which is called Robo Tester Evolution (RTE) [1]. RTE product is going to minimize tariff errors in the identification process and will minimize the cost of tariff testing. Existing products on the market can respond to certain of these needs; for example there is a traffic generator system in GSM network but there is no solution to check charging verification for this generated traffic. In order to fully cover all requirements RTE is developed. With RTE on GSM network traffic can be generated as a single call, when need be it is capable of creating traffic of thousand simultaneous calls and for these generated calls charging verification can be done through its own automatic reporting system. In comparison to the similar products [2,3] RTE is best fit for the Surepay product and it has some advantages in comparison to other systems such as speed, flexibility, low error rate and capacity to handle many scenarios in a single configuration. The RTE provides for both manual and automated off-line testing of charging configurations. The only difference between manual and automated testing is simply that manual allows a tester to run tests immediately whereas automated testing means scheduling the test run. Manual mode allow the user to try out different tests whereas once a set of tests are established they are run in automatic mode.

With the RTE from release 4.14 onwards it is possible to use a real handset based on Android technology that it helps to run real call test scenarios. For more information on Android technology please refer to [4]. RTE enables the creation of extensive test and automation scenarios that can be executed at convenience possibly from remote, taking advantage without the need for extensive human interaction. The key abilities to automate the process of call simulation and to specify charging response criteria and side effects whilst continuing with further user actions enable the automation of the entire SurePay tariff testing. Using RTE, the simulation of the call scenarios can be replaced by real call scenarios by using Android handsets that are

connected to the test environment via USB connectors. In order to drive real call scenarios ANDROID GW module is used. This module processes HPI formatted Android test cases and drive handset by using Android API instructions. Android network element should be created in order to define test cases.

Automation of Tariff Tests with Robo Tester Evolution

Currently, the operator tariffs tests are usually performed manually by creating a test call or creating a simple test steps. The disadvantages of these manually performed test methods are as follows. The cost is high: Calls are needed to perform by a GSM device. For testing scenarios large numbers of staff are required to use the GSM device. The system is slow: It takes a long time to realize the GSM call. The error rate is high: the testing staffs are likely to make mistakes in each scenario will take place. Staffs need to be on call with a GSM device for each 3 minutes long and considering the realization of a scenario, the scenario of 30,000 units tested by a team of 50 people will take 300 hours ($30000 \times 3/50 = 18.000 \text{ min.} = 300 \text{ hours}$). Conducting a test, as can be seen from this example has a substantial labor/time requirements cost.

Our RTE solution is developed for Nokia's Surepay charging server talking through messages with diameter and calls from the network on Surepay product simulates and performs the test if there have been the charge correctly. The advantages compared to other solutions on the market, Nokia's developed RTE product can be specified as follows. The cost is low: There is no need for calls to GSM devices. Employing a small team is sufficient to use the product. (This staff will work to prepare the overall test scenarios. When after the preparation of test scenarios can be stored and desirable condition with a single command.)The system is fast: The product is communicated over thousands of calls to interface with a diameter Surepay can simulate. Firstly 30.000 scenarios designed to test products within 8 hours, 30,000 scenarios can be run in as little as 3 hours. The error rate is low: Once a scenario is created once and can be replicated scenarios can be tested after the condition as a whole. Thus minimizing the possibility of human error. In RTE solution, both in the system before the tests cenario was created as a subscriber's account can bring to the scenario of the first case, you can change and view. In addition, the call scenario sound, SMS,MMS, and data can be used. As a result of this type of subscriber accounts are checked again call the desired spending (expected to be acquired or promotion)that appear in the subscriber's account can be controlled properly.

In addition, after performing scenarios, CDRs will be produced as a result in Sure pay system can be compared with RTE logs to see if there is a problem in charging. Generally, they provide a single test judging scenario is as follows. One or a group of the subscriber's account can be reset (can be determined that the first condition). Subscriber account can be changed as desired. Voice, SMS, MMS and data calls can be performed as required or can be in any order. Subscriber account can be checked for accuracy of pricingtrans actions made. Log produced by Surepay and call log files can be checked. A particular situation scenario is going to subscribe after runninga waited SMS messages or customer relationship management system (CRM)submission expected messages (subscriber's credits is over, the account is frozen etc.) can be controlled. Also connected to the system can manage Android OS based Smartphone and is able to perform test scenarios on the GSM network through these phones. Thus, it can be used not only for Surepay tariff tests, but also to test for other network elements in the GSM network.

RTE can remotely control phones and with remote connection facility there is no importance of testing personnel's location. For example, a test team in Turkey; there will be one staff sent to the German with the system, telephones (to make connecting to the system and maintenance of the device) can perform remotely through thousands of scenarios. The RTE is a tool that is to be used to ensure that the tariffs uploaded to the Surepay Online Charging System (OCS) are correctly applied for any given call scenario thereby verifying that the Nokia's Surepay Open Rating Engine is functioning properly. By using RTE following functions are possible. Define test sets that can be reusable.Run big amount of test cases without human interaction.Schedule test sets to run in a certain time.Run test sets both serial and pallellel mode. RTE allow multiple users to create/run tests at thesame time.Figure 1 shows RTE interaction with Surepay.

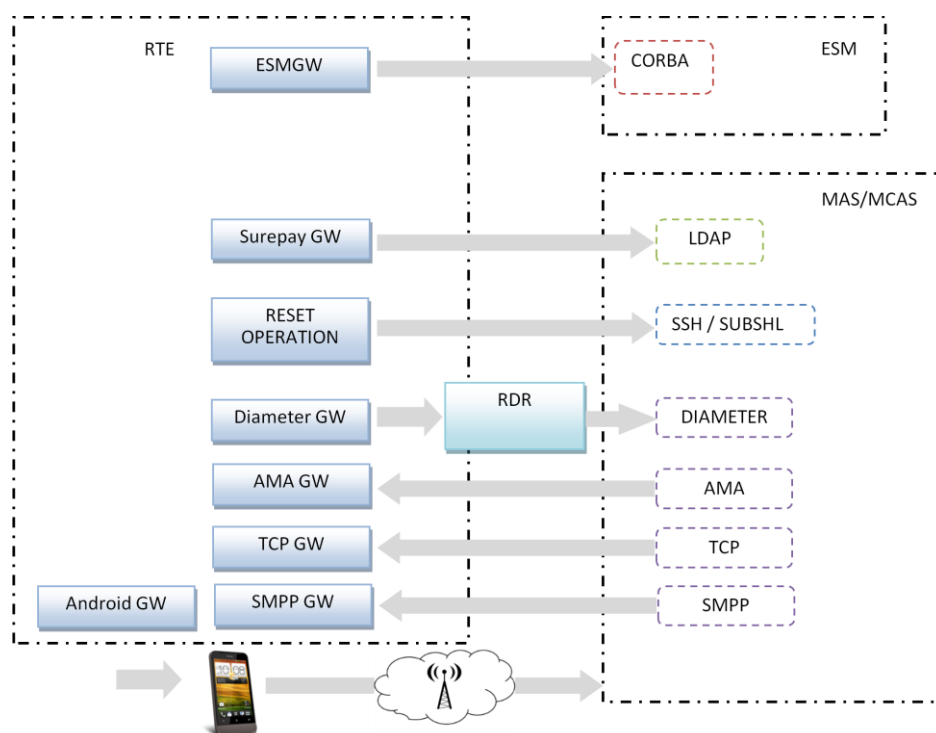


Fig. 1. RTE interaction with Surepay

RTE is a solution that consists of different UMF (Unified Mediation Framework) modules. RTE consists of modules listed below. UMF modules that are used in RTE solution are UMF-Middleware, Surepay Gateway, ESM Gateway, Diameter Gateway, SMPP Gateway, TCP Gateway, CDR-POP / AMA Gateway. RTEGW which is built as UMF process based on UMF-Middleware. RTE GUI. Openldap database used as storage. The RTE tool is designed to allow users to build tests through a web interface and execute remotely from the test laboratory. The RTE is built upon a services middleware platform known as the UMF. It has a building block structure which allows the instruction of commands for various different protocols from a single API commands structure through modules known as Gateways. The RTE today has many modules and gateways built to make up the feature set described below. Figure 2 shows the architecture and modules used to build RTE. Modules are summarized as: **RTE FE APP** – This application is running within the Apache server on board the UMF, executing PHP scripts to generate and validate the forms within the Web Interface. The PHP backend within Apache receives the test forms from the user interface and validates and writes them into the open LDAP DB. **OpenLDAP DB** – This is the storage mechanism to which all the test data is stored within. It is configured to run in memory for performance. The PHP FE Application has an LDAP API to write the tests directly into the LDAP DB. **RTE BE APP** – This application is the main logic for reading the tests from the database and executing and validating the results. The RTE BE application is also tasked with resetting the subscriber profile using ssh to instruct the OCS to load the subscriber profile directly into the backend RTDB. Additionally a sub process thread monitors the SurePay debugLog port to capture any debug traces that come from the system and compares pattern to match by the user. **Diameter GW** – This is a generic UMF module which converts HPI which tests are driven with into Diameter requests which are constructed by means of an AVP diameter template definition. This allows applications to introduce new diameter flavors more easily. **LDAP GW** – This is a generic UMF module which converts HPI which tests are driven with into SurePay LDAP Requests, or SurePay LDAP IMOM requests or SurePay LDAP IMOM requests.

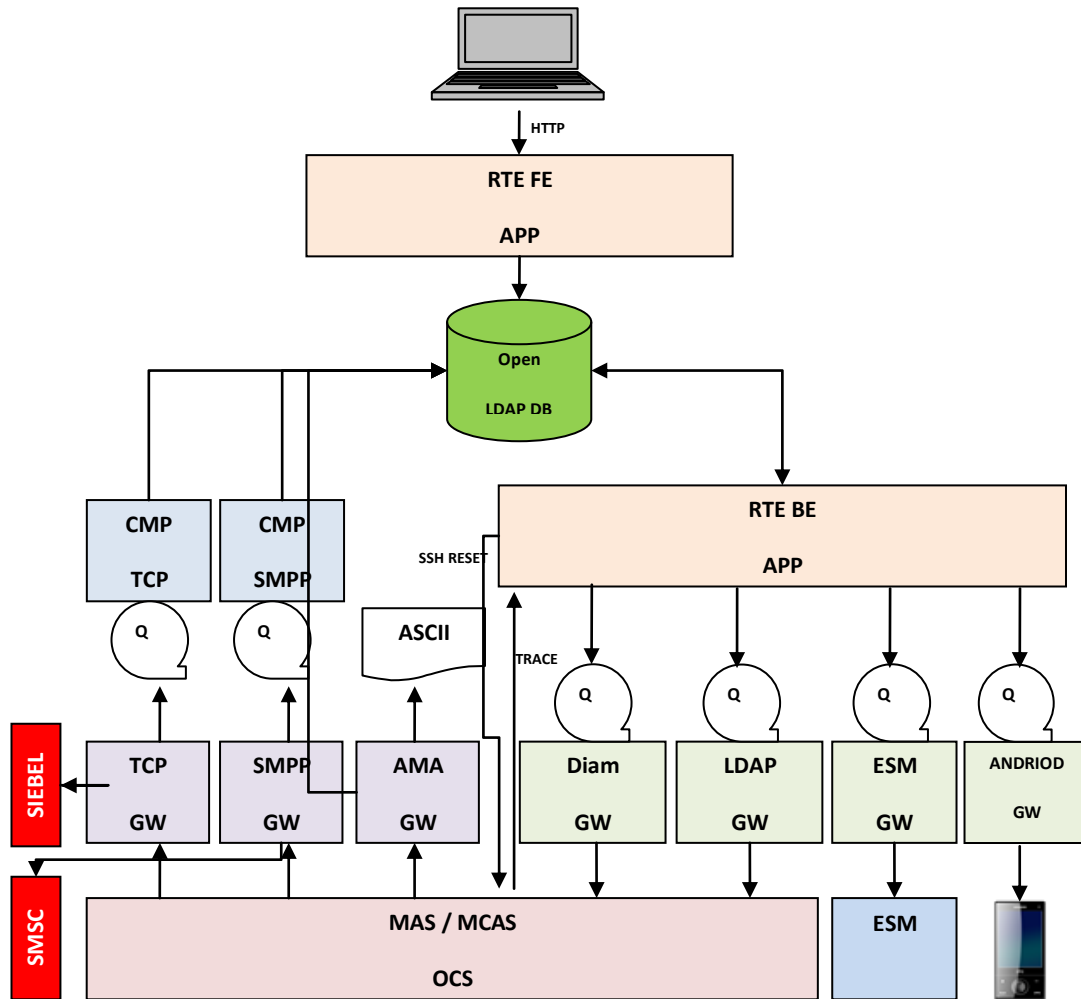


Fig. 2. RTE Modules

ESM GW - This is a generic UMF module which converts HPI which tests are driven with into ESM CORBA requests used for provisioning subscribers. **AMA GW** – This is generic UMF module which downloads binary AMA from the OCS and converts to ASCII HPI format. It has the ability to load custom rules modules to manipulate the data as it is transformed which is where RTE introduces a custom rule to validate the AMA data against matching expected AMA based on the users test. **SMPP GW** – This is a generic UMF module which converts incoming SMPP Binary data into internal HPI format. It has the ability to load custom modules to manipulate the data as it is transformed which is where RTE incorporates two custom modules to **a) Proxy** – This sends the data onward to an SMSC in binary form which allows the user to continue validating SMS's at their handset. **b) Alert** – This sends the data onward in HPI format to the comparison application. **TCP GW** – This is a RTE Specific UMF module which converts incoming TCP ASCII data into internal HPI format. It has the ability to load custom modules to manipulate the data as it is transformed which is where RTE incorporates two custom modules to: **a) Proxy** – This sends the data onward to upstream system such as Siebel in order to process real time notifications. **b) Alert** – This sends the data onward in HPI format to the comparison application. **CMP TCP APP & CMP SMPP APP** – Both are the same software configured differently in order to process data separately according to matching either SMPP data or TCP data in the openLDAP DB. This module will act as a recorder of incoming data and match with the user expected tests during the lifecycle of the test regression. It has basic instructions to Start Recording, Stop Recording, Compare Data, and Validate Data. **ANDROID GW** – This RTE specific module converts HPI which tests are driven with into Android API instructions on the handset to drive real; Voice Calls, SMS Messages, Data Sessions, and USSD Callback.

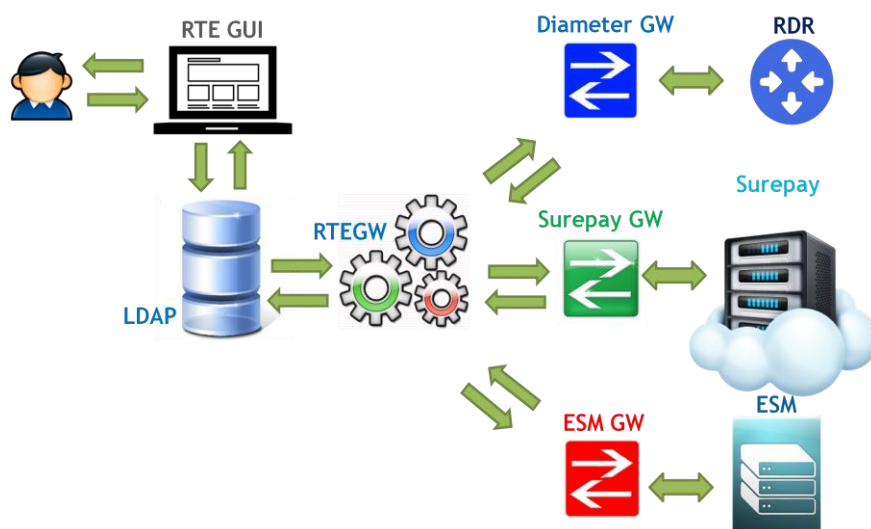


Fig. 3. RTE functionality and interaction with other systems

Figure 3 shows RTE functionality and its interaction with other systems. The steps of end to end working scenario can be summarised as in following. On the first step user creates tests on RTE GUI. Tests are saved under create tree in LDAP database. User can view/modify test cases and update database. Once test cases are ready user can initiate the test run on RTE GUI. When a run operation is initiated the test to be run are copied to run tree in LDAP database. RTE GW checks run tree on database continuously when it detects a test in run tree, it starts processing immediately. RTE GW sends requests to UMF GW modules based on the test component type. UMF modules are connected to related network elements. Retrieved results from UMF modules are saved in database under test tree. Once results are saved in database user can view them via GUI.

Conclusions

In order to use RTE in cloud environment necessary improvements are planned. Secondly RTE Evolution product will be standard part of surepay to be offered to all customers. To reduce integration time and costs an automatic configuration tool will be deployed in the product.

References

- [1] Nokia (former Alcatel-Lucent), Robo-Tester Evolution (RTE) Low Level design, document number: 8BH 04155 0081 UEZZA, 2014
- [2] Nexus Telecom, Nexus Verifier Test Call Generator (TCG) for Billing and Revenue Assurance, 2013
- [3] Allround, StabilityTesting, Testing Validation and Auditing, 2006
- [4] Amit Kumar Saha, A Developer's First Look At Android, <http://tailieuandroid.googlecode.com/svn-history/r8/trunk/Andoid--tech.pdf>, 2011