RCS-e stack

Installation guide

Edition 1.4
Date: 18/02/2014

Author:
Jean-Marc AUFFRET, jeanmarc.auffret@orange.com, Orange Labs

Summary

1. Introduction 3
2. Installation 3
3. Configuration 3
   3.1. RCS settings application 3
   3.2. Provisioning application 5
   3.3. Provisioning application 6
4. Startup 8
   4.1. Start the stack 8
   4.2. RCS account 9
5. Uninstallation 10
   5.1. Uninstall the stack 10
   5.2. Uninstall the provisioning application 10
6. Debug 11

Figures

Figure 1 – RCS settings application 4
Figure 2 – RCS settings application 4
Figure 3 – Provisioning application 5
Figure 4 – Start the stack via its settings 8
Figure 5 – RCS status notification 9
Figure 6 – RCS account 10
Figure 7 – Debug traces 11
Figure 8 – SIP Trace viewer tool 12

1. Introduction

The RCS-e stack contains two Android packages:

1. The **RCS-e stack** itself. This package contains also the **RCS Settings** application.

2. The **Provisioning application** to locally modify the default parameters of the RCS-e stack. This application is not mandatory and is only used by developers or integrators (for debug only).

The RCS-e stack is implemented via an Android service which runs is background and which offers RCS services (see RCS API) to any other application or UI.

2. Installation

Note: `adb` is the Android Debug Bridge tool available from the Android SDK (http://developer.android.com).

Use the following command to install the **RCS-e stack**:

```
adb -d install RCS_core.apk
```

3. Configuration

Any parameter of the RCS-e stack may be changed via:

- OTA interface (not yet implemented).
- The **Provisioning application**.

There are three levels of configuration:

- Configuration concerning the end-to-end service (e.g. ringtone) which may be modified by the end user: see the **RCS settings application**.
- Configuration concerning the end-to-end service (e.g. presence service) but not editable by the end user: see tab “Service” of the **Provisioning application**.
- Configuration concerning the stack: see tabs “Profile”, “Stack” and “Logger” of the **Provisioning application**.

3.1. RCS settings application

The **RCS Settings application** may be loaded from its application shortcut or from the top notification bar or from the RCS account:

Figure 1 – RCS settings application

The current stack release may be displayed from the menu “About”:

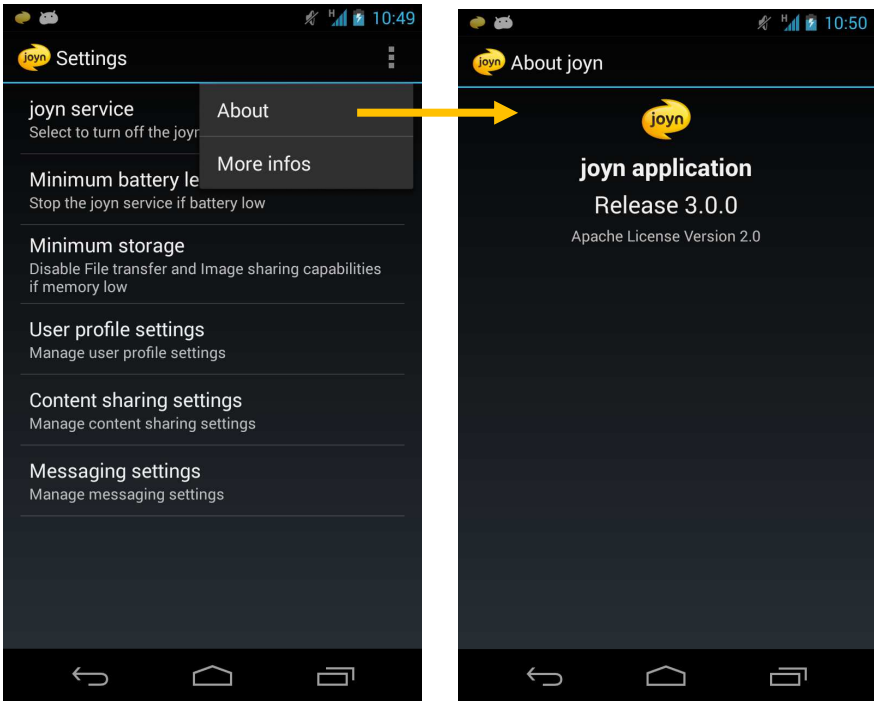


Figure 2 – RCS settings application

The end user can manage the following parameters:

- Stop/start the RCS service.
- Authorize or not the RCS service in roaming.

- Content sharing settings: invitation parameters, video codecs.
- Presence settings only if this service is activated: invitation parameters, predefined freetext.
- Messaging settings: invitation parameters, auto accept mode.

3.2. Provisioning application

The **Provisioning application** is only used to change the default stack parameter.

Parameters are sorted into four tabs:

- Tab “Profile” contains parameters related the IMS user profile (e.g. default P-CSCF address).
- Tab “Stack” contains internal stack parameters (e.g. default SIP port).
- Tab “Service” contains service parameters (e.g. presence service activation).
- Tab “Logger” contains trace management parameters.

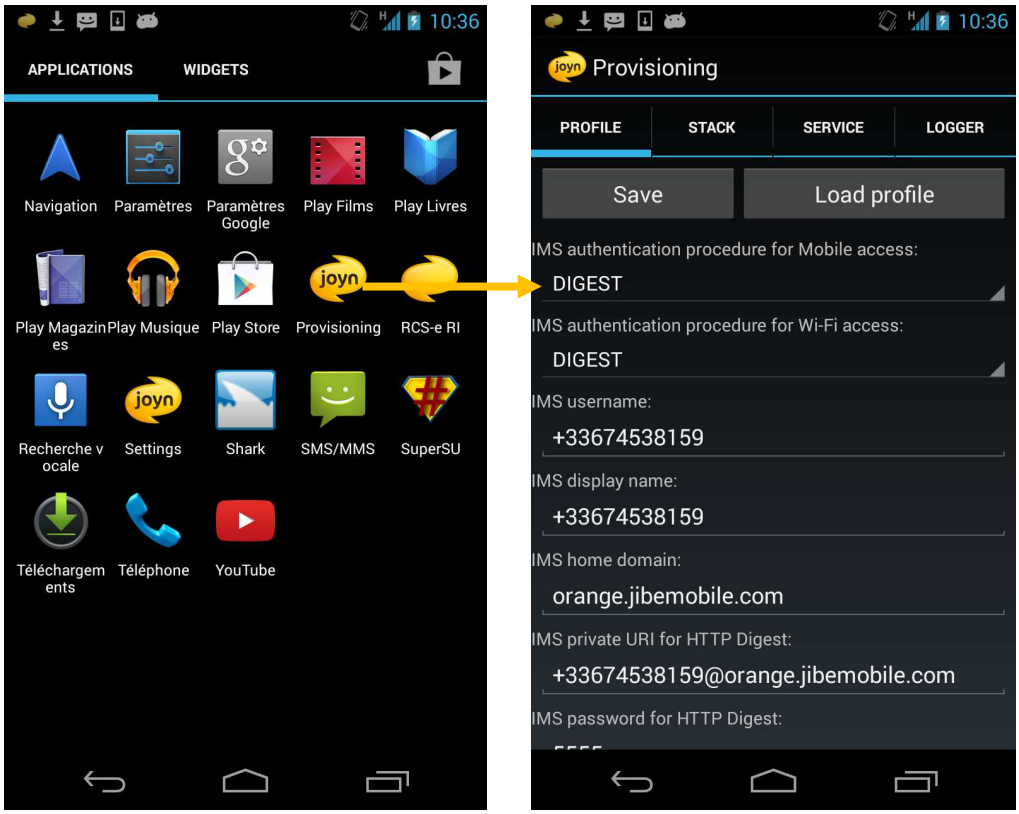


Figure 3 – Provisioning application

Note: when a parameter is changed, don’t forget to save it by using the menu “Save”.

Note: when a parameter is changed, don’t forget to restart the RCS service to take into account the modification.

3.3. Provisioning application

By default, GIBA (or early-IMS) authentication mode is activated and a default P-CSCF address and XDMS server address are provisioned. Then the other parameters of the IMS profile are discovered during the GIBA procedure (e.g. IMPU). The default parameter may be updated via the **Provisioning application** or OTA interface.

If the HTTP Digest procedure is activated, each parameter of the IMS profile may be updated via the **Provisioning application** or OTA interface. The following IMS parameters should be entered manually:

- Username: this is the username part of the IMPU.
- Display name: may be any string.
- Private URI: this is the IMPI.
- Password: password used for authentication during registration.
- Domain: home domain.
- Outbound proxy address for mobile access: address (IP or FQDN) of the IMS platform (SBC or P-CSCF).
- Outbound proxy port for mobile access: port of the IMS platform (SBC or P-CSCF).
- Outbound proxy address for Wi-Fi access: address (IP or FQDN) of the IMS platform (SBC or P-CSCF).
- Outbound proxy port for Wi-Fi access: port of the IMS platform (SBC or P-CSCF).
- Country code of the home domain: used to format into international format.
- APN: used to connect to the IMS platform. If null any APN may be used to connect to the service platform.
- XDM server: address of the XDMS.
- XDM login: login to connect to the XDMS.
- XDM password: password to connect to the XDMS.
- Conference factory URI for group chat.
- End user confirmation request URI for terms and conditions.

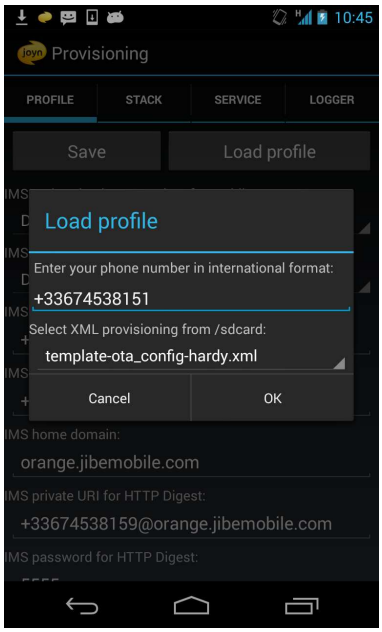
The IMPU is then built by using the username and the domain.

Sample:

- Username = “abcd”.
- Display name = “ABCD”.
- Private URI = “abcd_private@domain.com”.
- Password = “xxxx”.
- Domain = “domain.com”.
- Outbound proxy address= “127.0.0.1”.
- Outbound proxy port = “5060”.
- XDM server = ”http://127.0.0.1:8080/services”.
- XDM login = “yyyy”.
- XDM password = “xxxx”.
- Conference factory UI = “sip:conference-factory@domain.com”.

Note: for any update of the user profile, the RCS service should be restarted.

The provisioning can be loaded manually by clicking the “Load profile” button of the “Provisioning” tab.



Manual provisioning of the RCS stack:

- Enter your phone number in international format
- Select an provisioning file from the list of XML files available on the SD card

A template for the XML provisioning file (called “template-ota_config-generic.xml”) is provided with the RCS stack release. After customization of this generic file, download it onto the SDcard.
Use the following command to install the XML provisioning file onto the SD card:

```
adb push template-ota_config-generic.xml
```

You may install several provisioning files to test various configurations.

4. Startup

4.1. Start the stack

The RCS-e stack is started:

1. At device boot automatically.
2. By using the “RCS Settings” application:

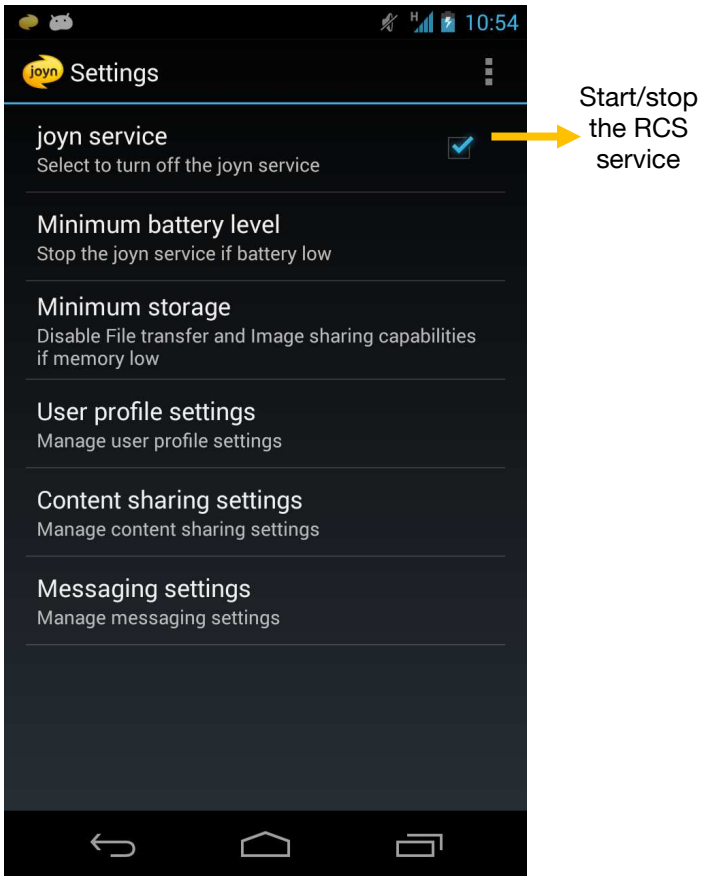


Figure 4 – Start the stack via its settings

When the stack is started a dedicated notification is added in the top notification bar of the device:

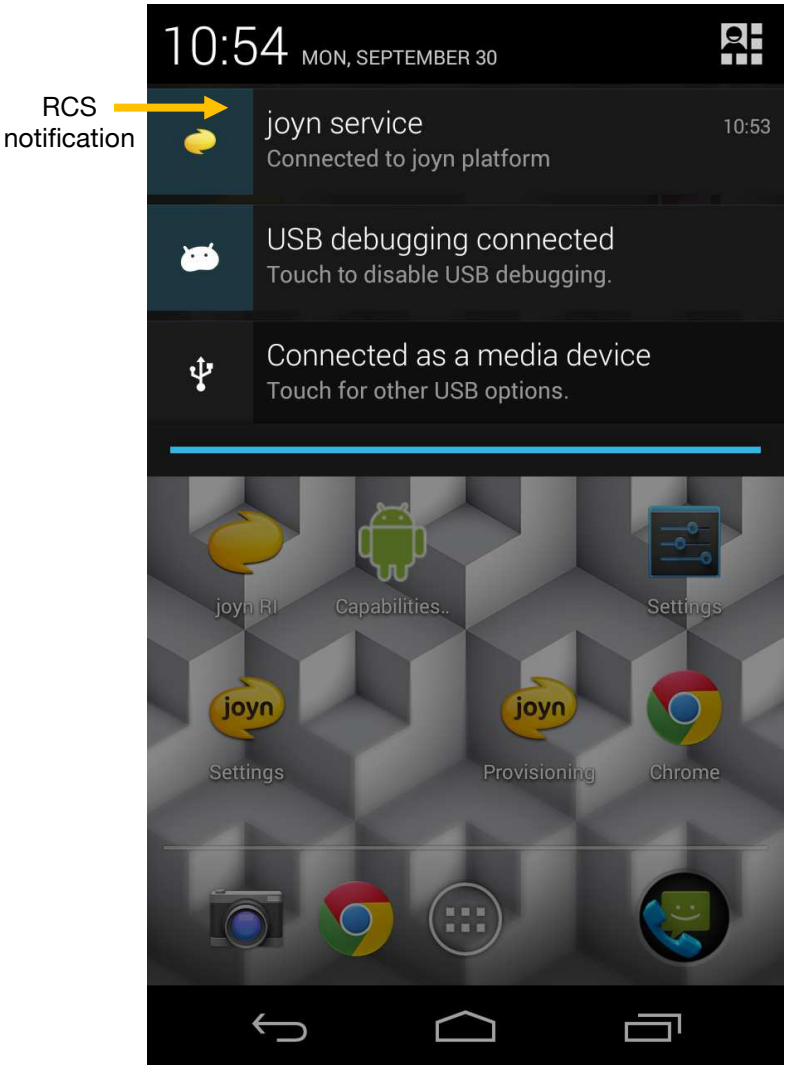


Figure 5 – RCS status notification

When the icon is grayed  , this means that the stack is not connected to the IMS/RCS platform (no coverage, IMS down, user not known, …etc).

When the icon is  it means you are registered to IMS/RCS platform with success (e.g. registration is successful).

The notification permits also to have a direct access to the RCS settings.

4.2. RCS account

At the first startup of the RCS-e stack a RCS account is automatically created. At any time the end user may remove or reactivate its RCS account.

The RCS account is managed from the device settings:

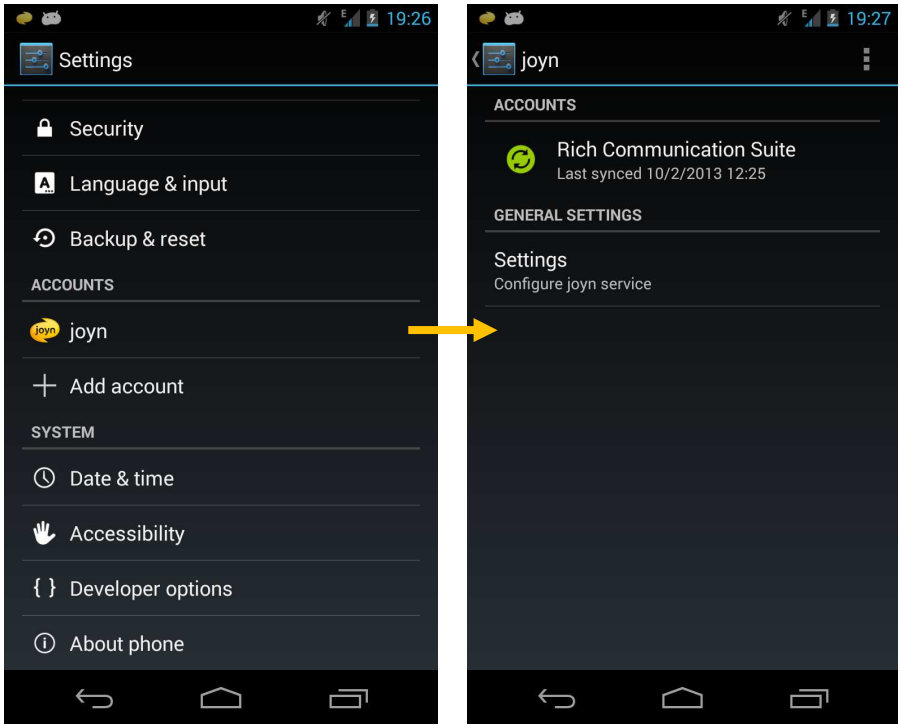


Figure 6 – RCS account

From the RCS account, the end user can:

- Access to its RCS settings.
- Remove the account: RCS data associated to each contact will be removed and lost.
- Synchronize the account: a new discovery operation if initiated to update the capabilities of each contact in the address book.

5. Uninstallation

5.1. Uninstall the stack

Use the following command to uninstall the **RCS-e stack**:

```
adb shell pm uninstall -k com.orangelabs.rcs
```

Note: `-k` permits to keep the application data.

5.2. Uninstall the provisioning application

Use the following command to uninstall the **Provisioning application**:

```
adb shell pm uninstall -k com.orangelabs.rcs.provisioning
```

Note: `-k` permits to keep the application data.

6. Debug

The level of trace may be changed via the **Provisioning application**, see folder “Logger”.

All traces starts with the tag “[RCS]” in the Android logcat ouput.

Concerning SIP traces they are available at two levels:

- From STDOUT:

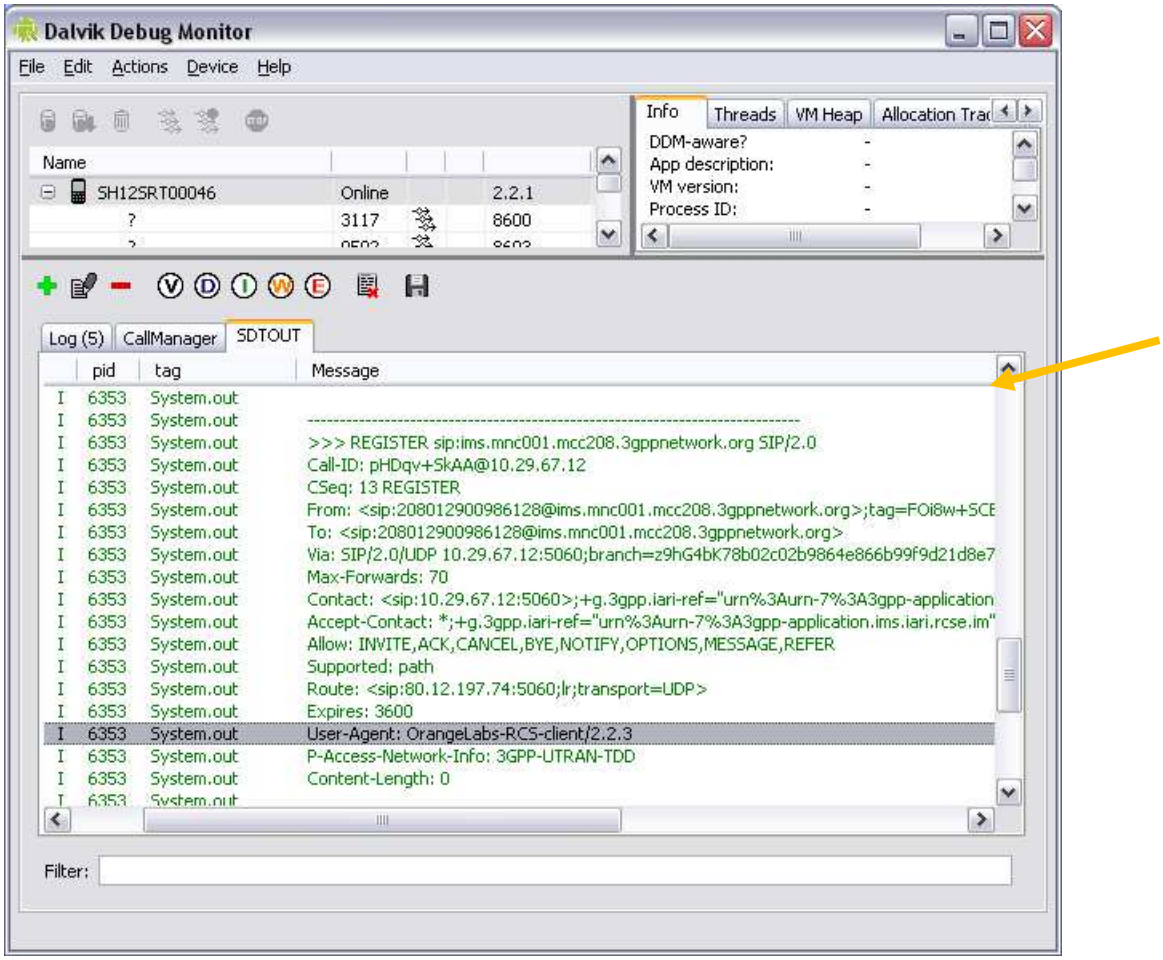


Figure 7 – Debug traces

- From `/sdcard/sip.txt` if the flag “SIP trace activation” is checked. Here the traces may be analyzed via the **Trace Viewer** application provided by NIST (public domain):

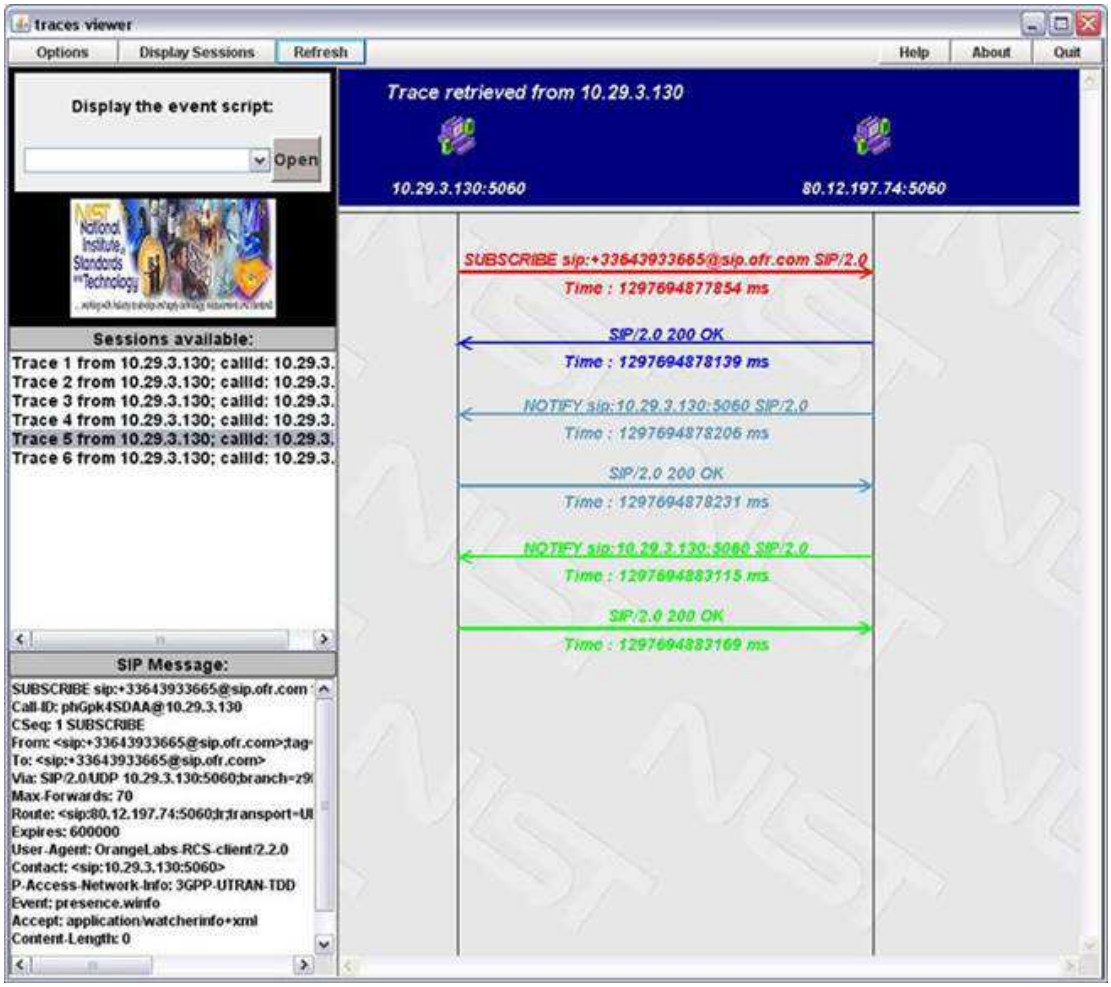


Figure 8 – SIP Trace viewer tool