Sipdroid User Guide

About Sipdroid

Sipdroid is a basic Sip Client application built on the MjSip port released by HSC earlier. In the current scope Sipdroid is quite basic and just the showcase example for helping developers to create next generation applications on Android using SIP. Sipdroid and MjSip have been both release under GPL and we welcome contributions from the entire developer community. We will be launching a more complete version of Sipdroid and an IMS compliant version of MjSip stack in time to come on both the Android as well as J2SE platforms in some time to come.

Building Sipdroid for Android

HSC has delivered you both the source and the package file for Sipdroid. Please refer to http://code.google.com/android/documentation.html for details on how to setup a development environment, compiling or installing applications on the Android Emulator.

Launching the Emulator

To start the emulator, change to the **tools/ folder** of the SDK and enter emulator or. /emulator. This initializes the Android system and you will see the emulator window appear on your screen.

To stop the emulator, close the emulator window.

Configuring the Emulator

Since, Sipdroid for now is only uses UDP (however not restricted to) as a transport; you would be required to perform some emulator specific configuration before the emulator can be send and receive end-to-end messages. The following text explains what this configuration is and how it is achieved.

Since, Android executes in a sandbox, we are required to map the ports on the host machine (i.e. the machine on which Android is running) to the ports on the emulator. This will allow the host machine to appropriately route the incoming datagrams into the emulator instead of the host discarding them.

As per Android documentation located at http://code.google.com/android/reference/emulator.html

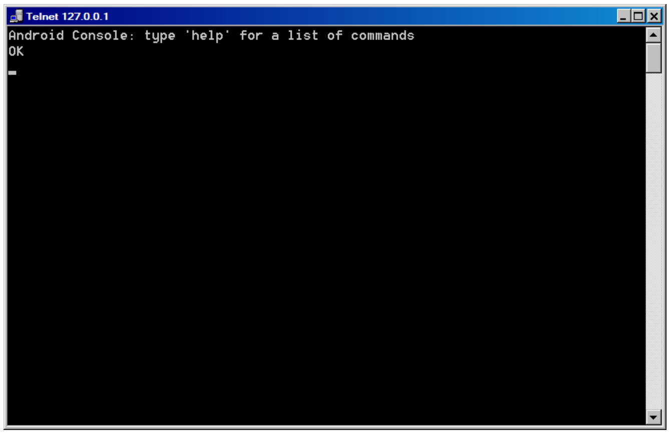
Each running emulator instance includes a console facility that lets you dynamically query and control the simulated device environment. For example, you can use the console to dynamically manage port redirections and network characteristics and simulate telephony events. To access the console and enter commands, you use telnet to connect to the console's port number.

To connect to the console of any running emulator instance at any time, use this command:

```
telnet localhost <console-port>
```

An emulator instance occupies a pair of adjacent ports: a console port and a debug port. The port numbers differ by 1, with the debug port having the higher port number. The console of the first emulator instance running on a given machine uses port 5554 and the debug port uses port 5555.

Open the console for the emulator instance you are running, which should look like this.



```
Android Console: type 'help' for a list of commands
OK
```

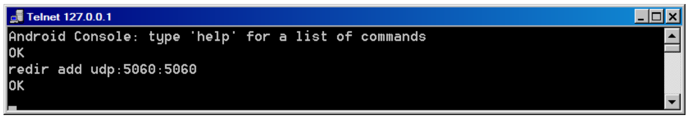
On the console type in the following command

```
redir add udp:5060:5060
```

The syntax for the redir command is as follows:

```
redir add <protocol>:<host-port>:<guest-port>
```

where <protocol> must be either "tcp" or "udp", <host-port> is the port number to open on the host, <guest-port> is the port number to route data to on the emulator/device.



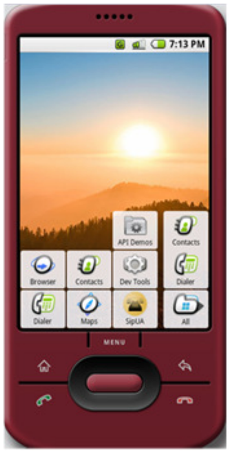
```
Android Console: type 'help' for a list of commands
OK
redir add udp:5060:5060
OK
```

As you would have understood by now, this will create an indirection for port 5060 on the host machine to that on the emulator and hence allow the emulator to send and receive data from the port it has been configured to do so by the application (Sipdroid in our case).

Note: Please note that you will be required to execute these commands each time you launch the emulator.

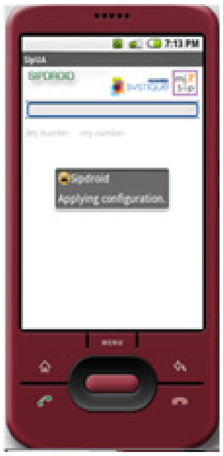
Launching Sipdroid

To launch Sipdroid click on the 'All' icon being displayed on the Android home screen, this will cause the emulator to display the top level menu of the applications present in Android. Clicking on the SipUA icon will launch the Sipdroid. Alternatively, you can also launch Sipdroid from within Eclipse. Please refer to the Google's documentation located here http://code.google.com/android/intro/tools.html for more details.



Sipdroid splash screen

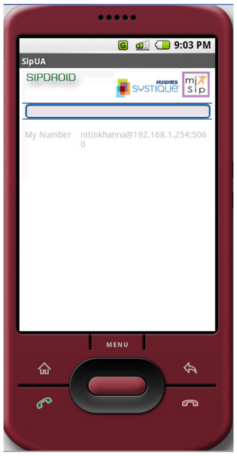
As Sipdroid is launched it loads the configuration and validates same. During this time it is in-active and displays status messages to user as displayed in the figure below.



Sipdroid

Applying configuration.

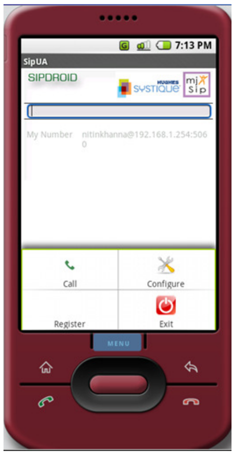
Once Sipdroid has been initialized you will see a screen shot like the one captured in the figure below.



My Number nitinkhanna@192.168.1.254:5060

Sipdroid Context Menu

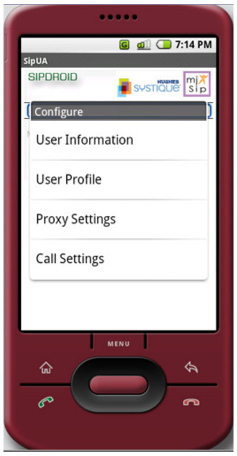
Clicking on the menu button on the emulator will display the context menu to the user. This is also captured in the figure below. There are four options in the main context menu. These are Call, Register, Configure and Exit. Before you can use Sipdroid you would be required to configure it since, because of some limitations in the present Android SDK certain parameters cannot be automatically determined but have to be provided by the user. While some of the configuration parameters as you would see are simply the place holder for features that you might see in the future releases, you have to provide configuration parameters as part of the User Profile configuration form to be able to use Sipdroid. Please refer the following sections on more details on the configuration parameters.



My Number nitinkhanna@192.168.1.254:5060

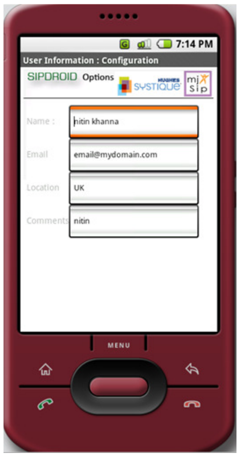
Configure Context Menu

The following menu is displayed to the user when the user clicks on the Configure menu item in the main context menu. The following menu has four menu options. The ‘User Information’ menu item allows the users to configure general user info details. Please refer to the User Information Screen section for more details on this. The ‘User Profile’ menu opens the User Profile screen which can be used to configure the Sip User Profile. The Proxy Settings menu opens the Proxy Configuration Screen which can be used to configure the domain, registrar and outbound proxy to be used for SIP Messages. The Call settings screen allows the user to configure different services and their user behavior. Sipdroid does not support user services for now as part of GUI, although these are implemented as part of the underlying UA.



User Information Screen

Using this screen user can configure name of the user, email address or comments. This configuration can be skipped.



User Information : Configuration

Name : nitin khanna
Email email@mydomain.com
Location UK
Comments nitin

User Profile Screen

The user can configure the SIP UA profile using this screen.

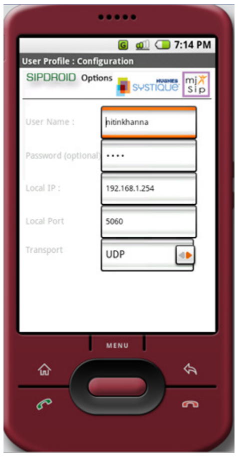
User Name: provides the local SIP user name which will be used to create the SIP URI.

Password: The password that should be used for authentication in case the registrar challenges the registering UA.

Local IP: Local IP of the machine on which emulator is running. Due to the present limitation of the Android emulator, the local IP address cannot be resolved. Take caution to provide this address properly, or else 127.0.0.1 will be assumed as a place holder. In case 127.0.0.1 is specified the UAS will not be able to route messages back to the client(Sipdroid) in the case of outgoing transactions since, 127.0.0.1 in the via would cause the message to looped back into the UAS.

Local Port: The port which has to be used for SIP. This has to be the same port for which port redirection was setup using the ***redir*** command, as explained in the start of this document.

Transport: This has two values, TCP and UDP. The default supported protocol is UDP (although not a limitation of MjSip). Any change in this value will be ignored by the application.



User Profile : Configuration

User Name : nitinkhanna
Password (optional) ••••
Local IP : 192.168.1.254
Local Port 5060
Transport UDP

Proxy Configuration Screen

Using this screen user can configure the domain, the outbound proxy, registrar, default registration duration to be used by UA.

Domain: Name of the domain to which the user specified in the User Profile settings belongs.

Use Outbound Proxy: if marked will configure the UA to route all outgoing messages via the specified outbound proxy [This is for future implementation and is not supported at present].

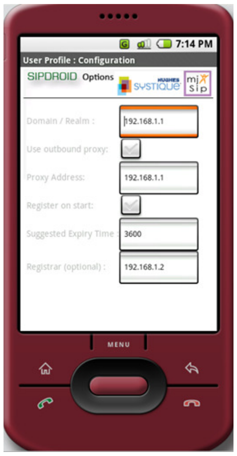
Outbound Proxy: The IP address of the outbound proxy to be used for messages which are sent out by the UA [This is for future implementation and is not supported at present].

Register On Start: If set, will initiate automatic registration of the Sipdroid with credentials supplied in the User Profile screen with the domain server specified in the current screen.

Suggested Registration Duration: This value is filled in the Expires header field, when a new registration request is sent out of Sipdorid, the value is specified in milliseconds. Once registered, Sipdroid refreshes registrations as per the procedures defined in RFC 3261.

Registrar: Specify this setting if the registrar is a different host then what has been specified in the domain [This is for future implementation and is not supported at present].

.



Domain / Realm : 192.168.1.1
Use outbound proxy: ☐
Proxy Address: 192.168.1.1
Register on start: ☐
Suggested Expiry Time : 3600
Registrar (optional) : 192.168.1.2

Call Options Configuration

Some of these settings will be used in the future release of Sipdroid and are basically used for enabling/disabling the services and the user behavior for these services. The following section briefly defines each of these service settings.

Automatically Accept Call: If this is set then all the incoming calls are automatically accepted on behalf of the user and a visual/vibratory alert is generated for them.

Ignore Calls After: If checked then an incoming call is ignored or rejected if it is not answered within the time period specified by the user [This service is not supported in the present release of Sipdroid].

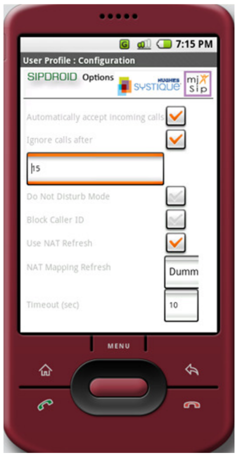
Do No Disturb Mode: If this mode is set then all the incoming calls are blindly rejected by Sipdroid.

Block Caller ID: Send out calls using the anonymous SIP uri as per the provisions of RFC 3261.

Use NAT Refresh: Not supported in the present release. Although, this feature will allow Sipdorid to refresh NAT findings, when Sipdroid is running behind the firewall.

NAT Mapping Refresh: The method that is to be used for NAT binding refresh. There are two mechanisms which are commonly used. While the first method is to send an OPTIONS message to peer, while the second is based on sending a dummy packet to peer [This service is not supported in the present release of Sipdroid].

Timeout: The time period after which NAT bindings should be refreshed [This service is not supported in the present release of Sipdroid].



User Profile : Configuration

Automatically accept incoming calls ☑

Ignore calls after ☑

15

Do Not Disturb Mode ☐

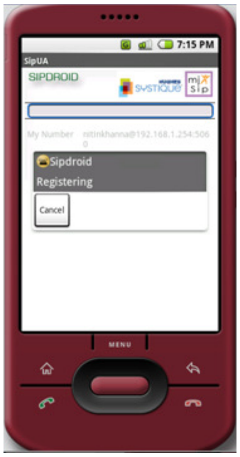
Block Caller ID ☐

Use NAT Refresh ☑

NAT Mapping Refresh Dumm

Timeout (sec) 10

Registration In Progress



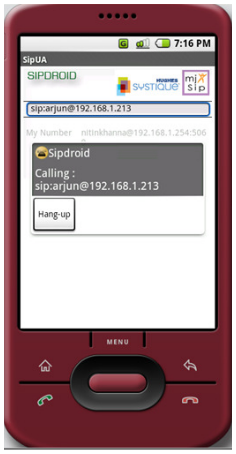
My Number nitinkhanna@192.168.1.254:5060

Sipdroid

Registering

When the user clicks on the registration menu or an automatic registration is configured, Sipdroid initiates registration of the UA with the specified registrar. The user will see the following screen.

Making an outgoing call



sip:arjun@192.168.1.213

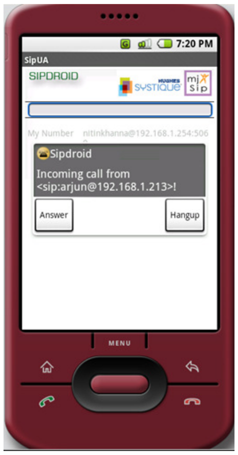
My Number nitinkhanna@192.168.1.254:5060

Sipdroid

Calling :
sip:arjun@192.168.1.213

To make an outgoing call, simply type in the SIP URI of the called party in the supplied box, and select the call menu option from the main context menu. This will initiate an outgoing call towards the specified peer.

Receiving an incoming call



My Number nitinkhanna@192.168.1.254:506

Sipdroid

Incoming call from <sip:arjun@192.168.1.213>!

When an incoming call is received from the peer a message box is displayed to user, as shown in the figure. Clicking on answer connects the call, while clicking on the Hang-up button rejects the incoming call and terminates the session.