HyperCast 4 Android Implementation

ZIAN HU

January 9, 2019

1 Overview

HyperCast 4 Android is an Android app that enables the establishment of HyperCast networks on Android devices. With currently two different choices of underlay networks, users can send messages to other devices.

2 Libraries and Tools

The HyperCast 4 Android app is developed using Java under Android development framework. Android APIs are used for managing sockets and building user interface. The `hypercast.jar` and `xom-1.2.10.jar` are included in the project library to implement HyperCast 4 functionalities.

3 Device Support

The HyperCast 4 Android app supports a minimum SDK (software development kit) version of 15, which corresponds to Android 4.0 Ice Cream Sandwich and is supported by approximately 100% devices distributed according to Android Studio (Appendix A). The target SDK version is 26 (Android 8.0 Oreo). See the `build.gradle` file in the project.

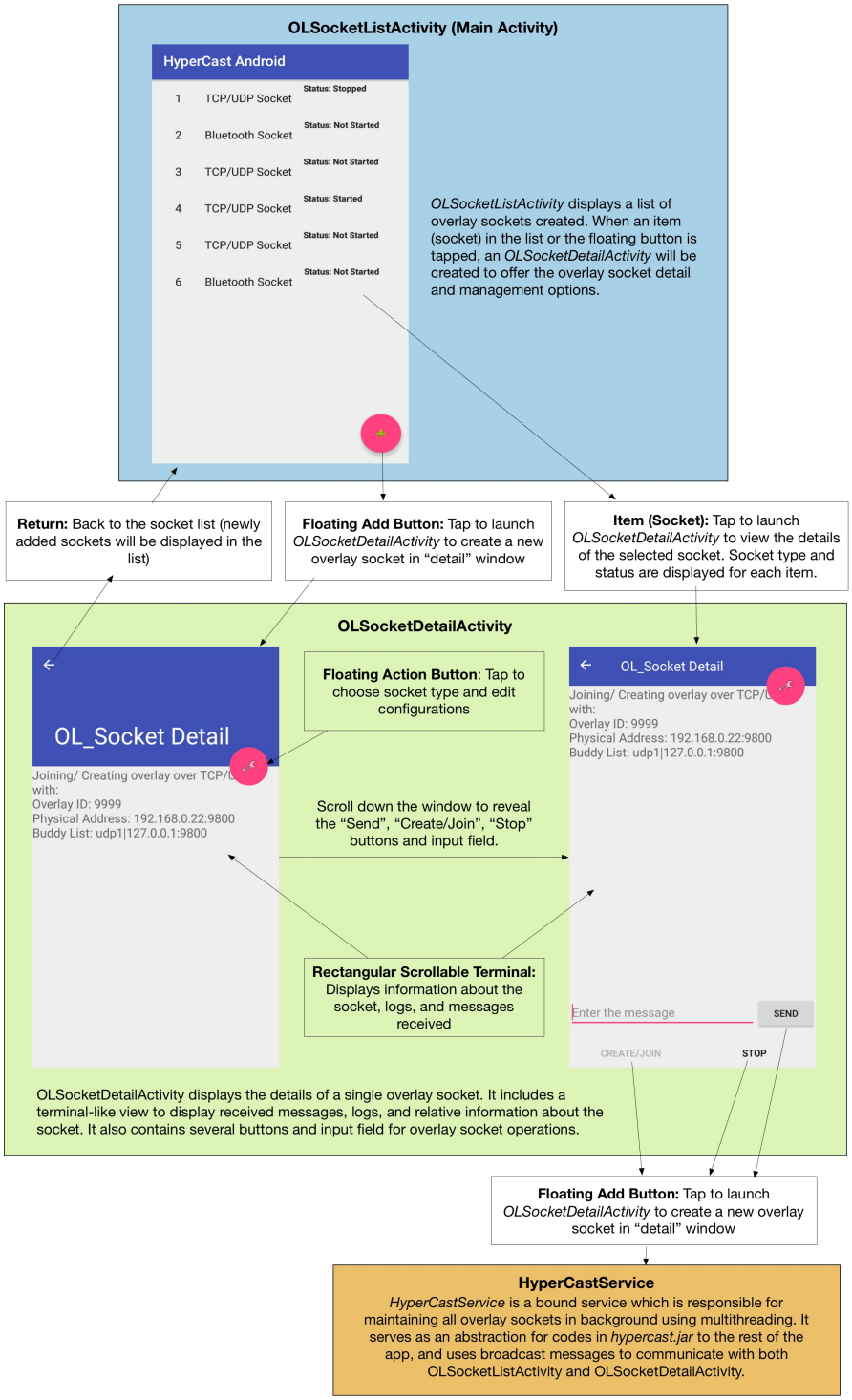


Figure 1: HyperCast 4 Android Design

4 Major Project Classes and Files

The HyperCast 4 Android app consists of the following major classes and files.

- OLSocketListActivity
- OLSocketDetailActivity, OLSocketDetailFragment
- ListAvailableConfigs
- OLSocketConfig, OLSocketConfigBT
- HyperCastService
- BTReceiver
- Bluetooth_Interface_SPT.xml, hypercast.xml
- AndroidManifest.xml

4.1 OLSocketListActivity

UI Layout Files: activity_olsocket_list.xml, olsocket_list_content.xml

OLSocketListActivity is responsible for creating and managing the launching UI window of the HyperCast 4 Android app. It consists of following two inner classes:

- `SimpleItemRecyclerViewAdapter`: Manages the UI recycler view which displays all the sockets available currently with their types and status.
- `OLSocketItem`: Serves as a database to store all relevant information of one initialized socket.

4.2 OLSocketDetailActivity, OLSocketDetailFragment

UI Layout File: olsocket_detail.xml

OLSocketDetailActivity is used to manage the UI window for each individual overlay socket. It contains a terminal-like view to display information about

current socket and messages received, a message input field, CREATE/ JOIN and STOP buttons for overlay socket operations, and a floating action button for configuring the current socket. `OLSocketDetailFragment` is used to pass item ID, and is also responsible for the layout of two-pane mode on tablets.

4.3 ListAvailableConfigs

UI Layout Files: activity_list_available_configs.xml, activity_list_available_configs_detail.xml

ListAvailableConfigs is a simple ListView that displays all the available socket configurations to choose from after the floating action button in OLSocketDetailActivity is clicked. Note that names of available configurations are declared in `app/res/values/strings.xml`.

4.4 OLSocketConfig, OLSocketConfigBT

UI Layout Files: activity_olsocket_config.xml, activity_olsocket_config_bt.xml

OLSocketConfig and OLSocketConfigBT are responsible for managing the creation and modification of configurations files for each socket. They parse user inputs and compose a HyperCast 4 configuration file for TCP/UDP or Bluetooth socket from the template "Bluetooth_Interface_SPT.xml" or "hypercast.xml", which are placed under the "asset" directory and are discussed more in Section 4.7. They then save modified files to the internal storage of the app. Templates always remain unchanged after the app is installed.

Please note that since all configuration files are created from templates, when configuration file entries are updated for later versions of HyperCast, codes in function `InitializeHyperCastConfigFile`, and templates in the "asset" directory should also be modified.

4.5 HyperCastService

HyperCastService is a bound service. It runs in background and uses multithreading to manage all TCP, UDP or Bluetooth connections. It utilizes broadcast messages to communicate with OLSocketListActivity and OLSocketDetailActivity.

It also contains an inner class `HyperCastAdapter` which serves as the interface between `HyperCastService` and codes in `hypercast.jar`. Each newly created socket will bind to `HyperCastService` first, and then starts to run on a separated new thread. All threads are implemented as finite states machines with four distinct states.

4.6 BTReceiver

BTReceiver is a manifest-declared broadcast receiver. It catches broadcast messages registered with following actions declared in `AndroidManifest.xml`, and perform operations accordingly:

- START_MODULE: Shows whether the BT_Module underlay has been started. If set to `false`, all bluetooth sockets will be stopped automatically.
- BLUETOOTH: Indicates whether a certain socket within the list is configured as a TCP/UDP or a Bluetooth socket.
- STARTED: Indicates whether a socket has started and in active state.
- MSG_RECEIVED: Shows if a new message has been received by a certain socket. Used to update message received in database.

4.7 Configuration Files: Bluetooth_Interface_SPT.xml, hypercast.xml

These socket configuration files are placed in "asset" folder and contains some default configurations of Bluetooth and TCP/UDP sockets. Specifically, only the following entries can be modified by users during runtime through UI, other configurations have to be changed in Bluetooth_Interface_SPT.xml, hypercast.xml, and the app has to be reloaded using Android Studio after changes.The modified configuration file will be saved as a new file into the internal storage of the app.

Bluetooth Sockets:

- OverlayID: Tag of the overlay identifier
- INETV4AndOnePort: Overlay socket's physical address

- InterfaceAddress: Address of buddy
  Default: If no InterfaceAddress is provided, the overlay address of the host itself is set for this field

TCP/UDP Sockets:

- OverlayID: Tag of the overlay identifier
- BTModuleAddress: Address of BT_Module for TCP connection

4.8 AndroidManifest.xml

The manifest file describes essential information about the app to the Android build tools, the Android operating system, and Google Play.This includes information about services and permissions. For HyperCast 4 Android app, the permission for INTERNET must be declared in the manifest file. It is also responsible for declaring the broadcast receiver `BTReceiver` as described in Section 4.6.

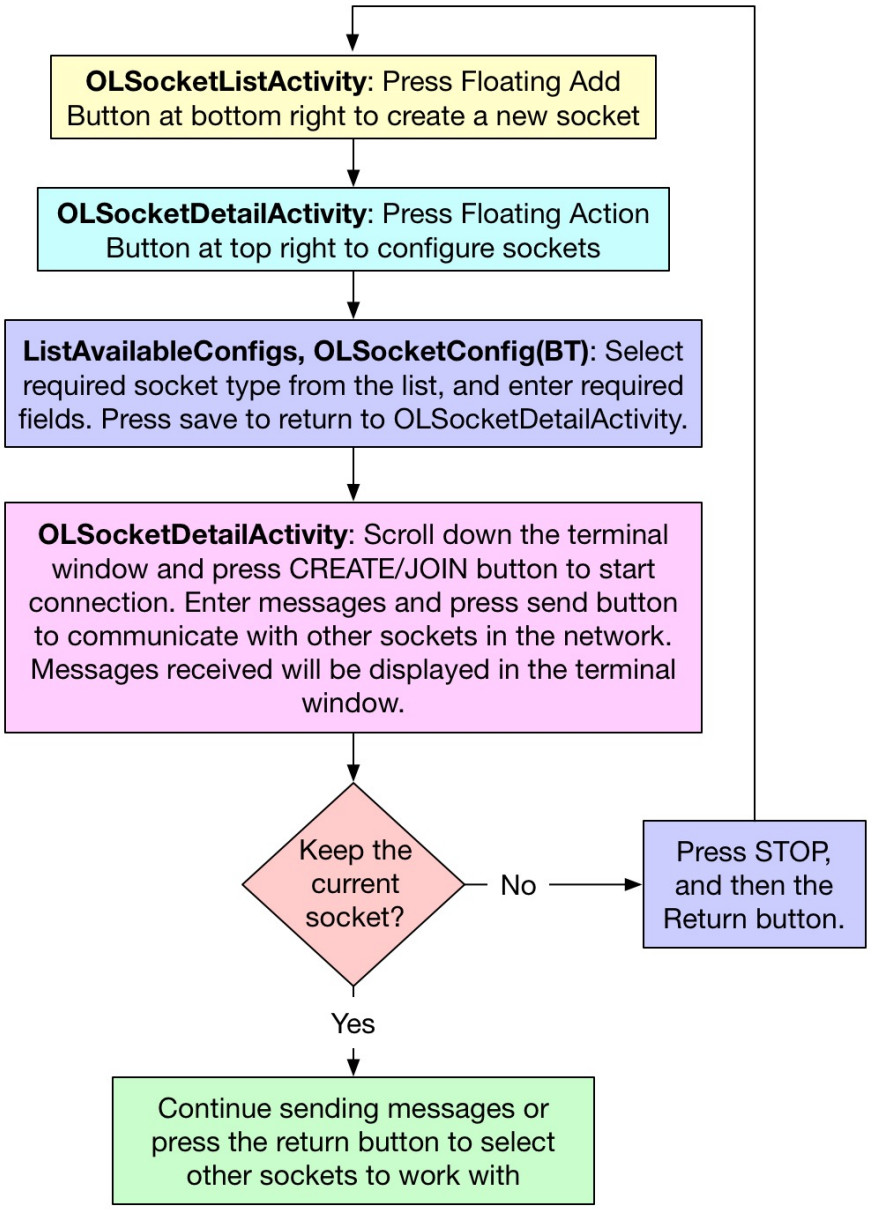


Figure 2: Normal Workflow of HyperCast 4 Android

5 App LifeCycle

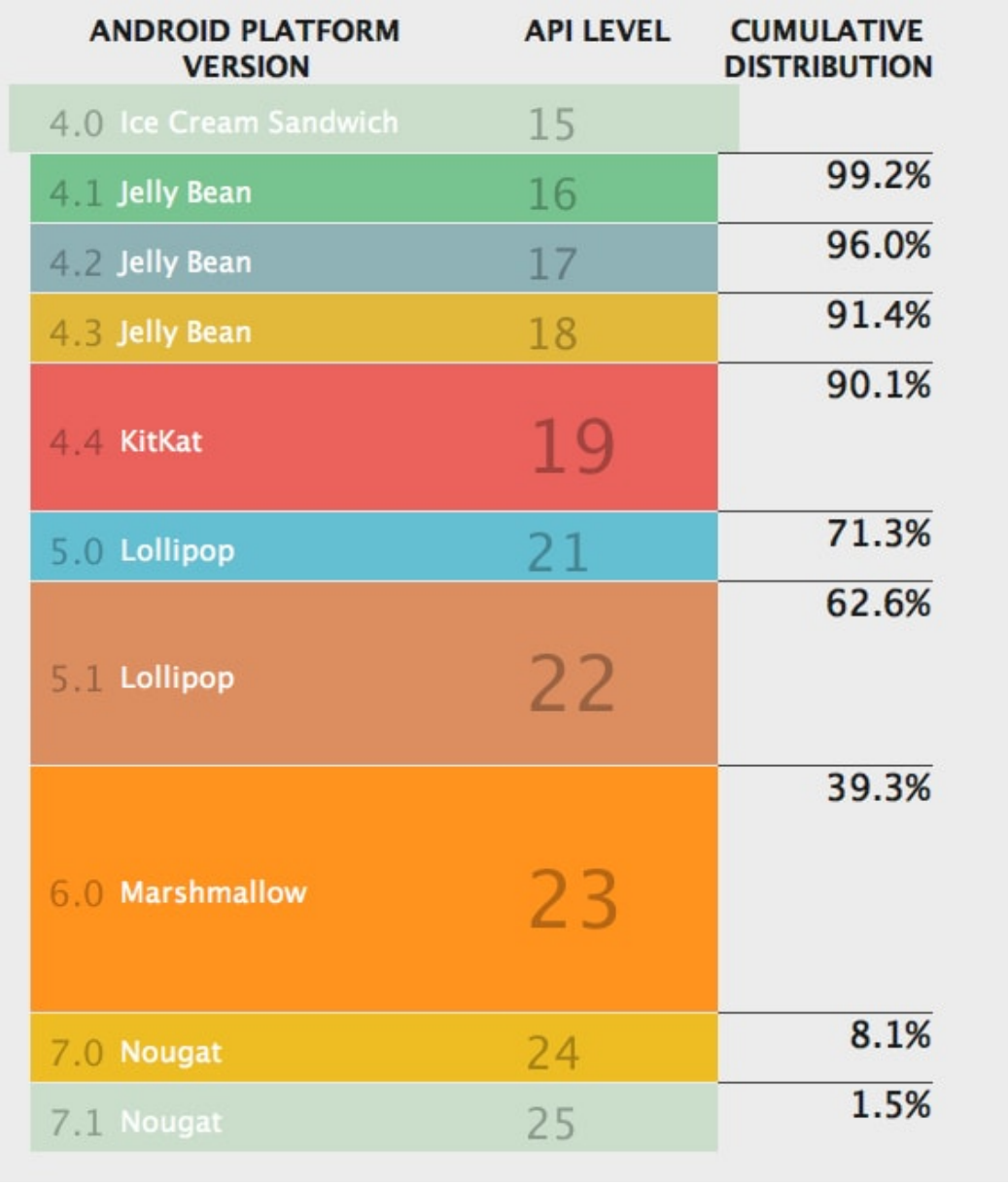
HyperCastService is a bound service that manages and maintains all connections. Each instance of `OLSocketDetailActivity` will bind to HyperCastService in its `onCreate()` method, and unbind when its `onDestroy()` method is executed. However, the socket associated with this activity is not closed and will continue receiving messages and updating database after the acitivity is destoryed (tap the back button in `OLSocketDetailActivity`). A socket can only be closed by the STOP button or by exiting the app. Therefore, HyperCastService is able to maintain all sockets and connections until the user exits the app explicitly (kill the process in task manager). Turning the screen off, entering sleep mode, and tapping Home button will not close HyperCastService.

6 Useful Websites

HyperCast Documentations:
https://www.comm.utoronto.ca/hypercast/design.html

General Android API:
https://developer.android.com/guide/components/fundamentals.html

A Android Studio Device Support



ANDROID PLATFORM VERSION	API LEVEL	CUMULATIVE DISTRIBUTION
4.0 Ice Cream Sandwich	15	
4.1 Jelly Bean	16	99.2%
4.2 Jelly Bean	17	96.0%
4.3 Jelly Bean	18	91.4%
4.4 KitKat	19	90.1%
5.0 Lollipop	21	71.3%
5.1 Lollipop	22	62.6%
6.0 Marshmallow	23	39.3%
7.0 Nougat	24	8.1%
7.1 Nougat	25	1.5%

Figure 3: Android Studio Device Support Chart