

## Location Based System Using Google Cloud Messaging

<sup>1</sup>Harminder Singh, <sup>2</sup>Dr Sudesh Kumar, <sup>3</sup>Harpreet Kaur

<sup>1,3</sup>SGHS Khalsa College, Panjokhra Sahib, Ambala

<sup>2</sup>BRCMCET, Bahal

<sup>1</sup>harminder.singh2125@gmail.com, <sup>2</sup>sudeshjakhar@gmail.com, <sup>3</sup>harpreetsial@gmail.com

**Abstract-** Android is the smart mobile phones platform that is very flexible and contains numerous number of interesting features. This platform is launched by google. Android mobile phone devices contains a location based service that can locate a mobile device and show the position of the device on the map. There are various applications that efficiently implement this service. There are multiple location providers that helps in finding the exact location of a given device. Each location provider has its own pros and cons. So far, programming a location based service and map application was considered as complex and time consuming, but android platform provides various libraries, API's and user interface controls that make this location based services very easy to implement. Android platform provides an another very useful service that is generally used for sending the push notification to the user devices. It is named as Google Cloud Messaging (GCM). This service uses the third party google cloud servers to send notifications to the clients. Till now the use of this service is limited only in providing update notifications to various applications installed on the mobile device and some chatting apps also use this service. But there are much more fields in which this service can be used. This paper introduces the basic architecture and component models of Android. After explaining android architecture we will focus on the research that how GCM service and location service can be combined to develop a new kind of service and what are the powers and the limitations of this newly created service. **Keywords:** Android platform, location service, GPS provider, network provider, Google cloud messaging.

### I. ANDROID PLATFORM ARCHITECTURE

Android operating system consists of a stack of software components which can be divided into five sections and four layers as shown in the fig 1.1 given below. The bottom most layer in android architecture is Linux Kernel.

*a) Linux Kernel-* This layer provides all the basic system functionality that are typically provided by any operating system like Memory management, Process management and device management like Bluetooth, camera, Wi-Fi, keypad, LED/LCD display etc. This layer also take care of several functions like networking, handling device drivers and interfacing to peripheral devices.

*b) Libraries-* Second layer on the top of Linux kernel layer is Libraries layer. In this layer there is a set of libraries that also includes open-source Web browser engine WebKit, a very well known library libc, SQLite database for android which is a used for storing application data and sharing of application data. It also includes libraries to record and play audio and video, SSL libraries responsible for Internet security etc.

*c) Android Runtime* – This layer is a part of the Libraries layer. This section provides a very important component of android architecture namely DVM (Dalvik Virtual Machine) which is like a JVM (Java Virtual Machine) specially designed and optimized for Android platform. The

DVM makes use of Linux core features like process management, memory management and multi-threading. The DVM enables every Android application to run in its own process, with its own instance of the DVM. The Android runtime layer also provides a set of core libraries which enable Android application developers to write Android applications using standard Java programming language.

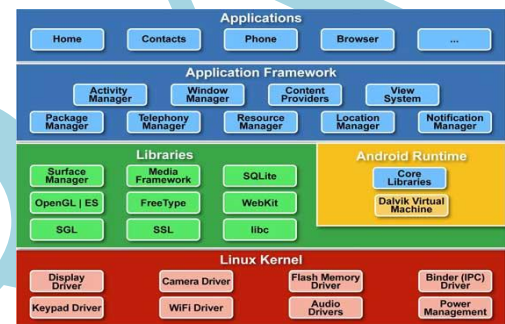


Fig 1 Basic Android Architecture

*d) Application Framework* - The Application Framework layer that is present above libraries layer provides many high-level services to android applications in form of Java classes. Application developers can use these services in their applications.

*e) Application* - All the Android application are at the top of this layer. The application that can be installed on android platform is programmed in this layer only. For example: Google maps, Gallery, Contacts Books, Browser, Games etc.

### II. LOCATION BASED SERVICES

A Location Based Service (LBS) is an informatory service that can be accessed via a mobile device. This service make use of the mobile network or the GPS (Geographical positioning System) to locate users mobile. The output of this service is the latitude and the longitude value of the device. These latitude and the longitude values can be then viewed on a map to get the location in graphical form. The location based system can be used in a variety of fields such as transport, work, home, and someone's personal life. For example, It can provide the driving directions to some destination place from the current place, We can design an app that switch off the employees mobile phones when they enter conference hall, parents can track the activities of their child etc. Location based services can be classified as triggered LBS (also called push service) and user-requested LBS (also called pull services). Triggered LBS are the one which automatically updates the location of the device on the map when the device moves a minimum specified distance in any direction from the current position or we can specify the

time interval after which the location of the device is automatically updated. On the other hand user-requested LBS are the services that provide the location coordinates of a device only when it is requested by the user.

### III. COMPONENTS OF LOCATION BASED SERVICES

Location based service is the collection of various small components that are used collectively to achieve the common goal of finding the location of the user device and show these device on a google map. Main components of the location based service are as follow

- a) MOBILE DEVICE – A mobile device that is connected to some wireless technology like GSM, CDMA etc. or a GPS enabled mobile device that contains all the requirements needed for a GPS module to perform its function.
- b) LOCATION MANAGER- This class is used to access the location services. This class contains mechanism that allow applications to obtain periodic updates of device's geographical location. This class is not instantiated directly. The object of location manager class is instantiated as follows

LocationManager locMan =  
Context.getSystemService(Context.LOCATION\_SERVICE).

This class contains some constants that helps to select the location provider and set the frequency of updates that the app should deliver to the user.

- c) LOCATION PROVIDERS- The android platform contains two type of location providers namely

- i) Network provider
- ii) GPS Provider

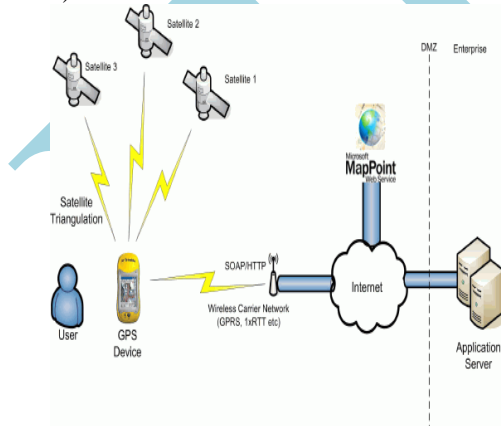


Fig 2- Various Network Providers

Both of these providers have their own pros and cons. The network location provider is fast in operation but it is not accurate. It does not locate the device to its exact coordinate position. GPS provider on the other hand is more accurate than the network provider but in some case GPS provider may be slow in operation or moreover be unavailable. GPS provider consumes a lot of device battery power, this is also a drawback of the GPS provider.

- d) LOCATION LISTENER- Location listener is an interface provided by android development platform that provides the

location coordinates of a device when the device moves by a specified distance in any distance. It provides various methods that can be used by the programmers to find and locate a device. Some methods that must be overridden when we implement location listener are as follows.

i) *Public abstract void onLocationChanged(Location location)* - This method is automatically called when the location of the device is changed. This method takes one parameter that is the object of the Location class. Using this object we can get the latitude and longitude values of the newly changed position as follows.

```
double latitude = Double.toString(location.getLatitude());
double longitude = Double.toString(location.getLongitude());
```

ii) *public abstract void onProviderDisabled(String provider)* – This method is called when user disables the location provider. It can be used to show any alert message to the user or do some task that are no longer valid without enabling the provider. The string parameter provider contains the name of the location provider that is associated with this disabled update.

iii) *public abstract void onProviderEnabled(String provider)* – This method is called when user enables the location provider. It can be used to show any welcome message to the user or do some initialization tasks. The string parameter provider contains the name of the location provider that is associated with this enabled update.

iv) *public abstract void onStatusChanged(String provider, int status, Bundle extras)*- This method is called when the status of the provider changes. When a location provider is not able to fetch a location, or if a location provider becomes available after some unavailability programmer can handle these situations by using this method. This method contains three parameters location is the name of the location provider, status represents the current status of the location provider it may be OUT\_OF\_SERVICE, TEMPORARILY\_UNAVAILABLE or AVAILABLE. Extras is an optional Bundle which will contain provider specific status variables.

### IV. GOOGLE CLOUD MESSAGING SERVICE

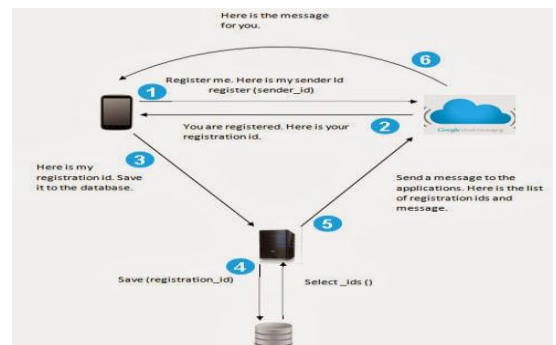


Fig 3- Sending Message Through GCM

Google Cloud Messaging (GCM) is a service available with android platform that allows us to send data from organization's web server to users having Android-powered devices. This service can also receive messages from user devices using the same connection

The GCM service handles all aspects related with queuing of sent messages and delivery of these messages to the target Android device. This service is provided by google and it is completely free.

For using this service each device needs to be registered with GCM server. Only those devices can get push messages that are registered with the servers . Several different id's are generated during the registration process. Each of this id has some specific functionality. The life cycle of sending and receiving message via GCM is demonstrated in figure 3 shown below.

a)PROJECTS API KEY- Once the new project is setup and we get the Project number, we will now create a server API key. This key is used to provide authentication later during sending push notifications . To create an API Key we need to click create new key present under link APIs & auth in the sidebar of the google developer's console.

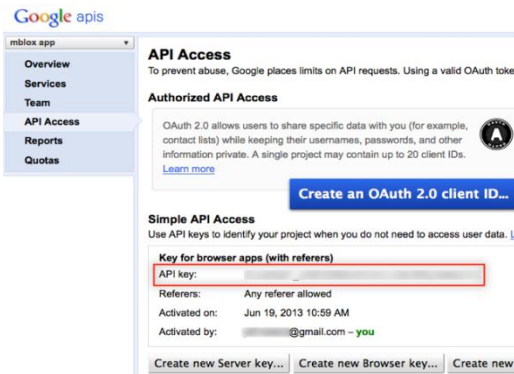


Fig 4 Creating API Key

b)CREATING GCM CLIENT- APP Android platform provides *GoogleCloudMessaging* Api for implementing GCM client app. GCM requires the android devices installed with Android 2.2 or higher and these devices must have installed google play store application. If we want to test our app on emulator then the emulator must be installed with Android 2.2 or higher and have google API installed. A full GCM implementation requires both a client implementation and a server implementation. In order to implement GCM service our app needs the following permissions and components

i)*com.google.android.c2dm.permission.RECEIVE* – This permission is needed so that the android device can register and receive messages.

ii)*android.permission.INTERNET* – This permission is required so that the app can send registration id to server.

iii)*android.permission.WAKE\_LOCK* – This permission is required so that the application prevent the processor from sleeping when some message is received.

iv)A receiver, with the category of receiver set to applicationPackage. The receiver requires the *com.google.android.c2dm.permission.SEND* permission, so that only the GCM Framework can send a message to it. If our app uses an [IntentService](#) this receiver should be an instance of [WakefulBroadcastReceiver](#).

v)We will use a [Service](#) (typically an [IntentService](#)) to which the broadcast receiver hands off the work of handling the GCM message at background without interfering with the interface.

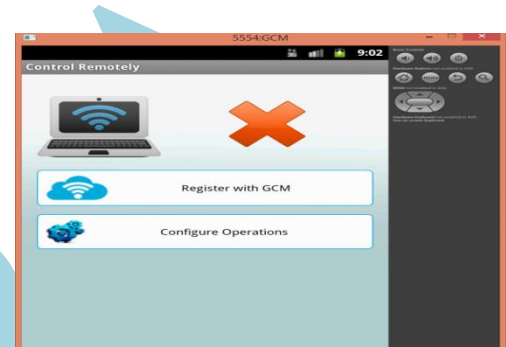


Fig 5- GCM client app

b)CREATING SERVER FOR GCM DEVICES- After creating the GCM client app we need to create a GCM server this server will store the registration ids of all the registered devices with GCM. If we want to send push notifications to selected GCM registered devices, server sends the selected registration id to GCM server with the message to send to these devices. If the targeted devices are switched off or not connected to the network, then the messages would be queues and all the messages will be delivered to these devices when these devices become available. The message delivery using the GCM technique is unpredictable. The send messages are not acknowledged and the messages may reach their destination with different latency. One of the major benefit of GCM is that the client app need not to be running on the device in order receive the push message.

The GCM server can be created using any of the server side technology like java, php, asp.net , python etc. The server just need API key for the use of GCM server to provide authentication.



Fig 6- Server for Sending

### c) COMBINING LOCATION BASED SERVICES AND GOOGLE CLOUD MESSAGING SERVICE

These two techniques discussed above can be combined together to create some powerful applications. These applications can be very useful in various fields. We can create a new client app for android platform let call it "Control Remotely" that can be used to control the android device from a remote server. This app can register with the GCM service and after registering with the GCM service it will store the registration id allocated to that device to our server's database. Now if we want to control the android device remotely we can just send some specific message to the GCM server the GCM server will transfer that message to the Android device. Now the client app can be controlled from the server by sending some predefined messages. Various type of operations that can be performed remotely on the client devices can be starting gps provider, stoping gps provider, get location of device through network provider, get location of device through gps provider. This app can be very useful in various fields. For example in transport sector, the manager or owner of a vehicle can track their vehicle at any instance of time and the exact location can be seen in google map as shown in figure 5.1. Client android device can share their location with server, sharing can be to a single person or location can be broadcasted. This app can be further enhanced by scheduling the device for broadcasting its position various times a day and at specified times. This will allow the owner to track the whole day activities of particular vehicle. Despite various advantages there are several limitations of this app. As the message delivery using GCM is unpredictable, the devices may respond to user commands with different delay. If the user device is switched off then all the commands sent from the server will be queued and delivered to the target device when the device become available. This is the main drawback of this system.

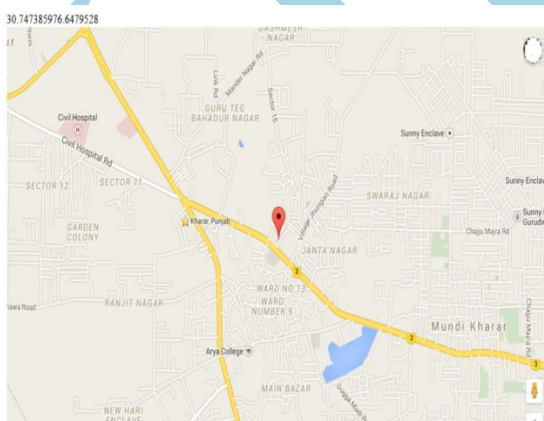


Fig 7- Google Map Showing device location

### V. RELATED WORK

In this paper the major focus was on combining the feature Google Cloud Messaging with the Location Based System. By combining these two features we get very powerful apps that can track android devices. Some more services can be combined with GCM services that will create some other

type of powerful apps. For example Camera service, Voice Recording Service, can be combined with GCM to remotely control various devices. Clients devices can be read for Contacts, Images, Videos and other data remotely. Further embedded devices with these apps can be created and installed at required places.

### VI. CONCLUSION

By combining the features of GCM and LBS we can develop powerful tracking device. The apps currently in the market can track the device remotely. But they uses lot of system resources. This combination can save various system resources and thus works efficiently. There are some limitations of this system, as the GCM message delivery is unpredictable. The output from the system can't be expected with certainty. If the target device is not available (not connected to internet) the messages will be queued and will be delivered as soon as the device becomes available. This combination will work effectively if the device has stable internet connection.

### REFERENCES

- [1] "Google Cloud Messaging for Android — Android Developers." [Online]. Available: <http://developer.android.com/google/gcm/index.html>
- [2] Whipple, J. Arensman, W. Boler, M. S., "A public safety application of GPS-enabled smartphone and the android operating system," IEEE International conference on Systems, Man, Cybernetics, pp 2059-2061, 2009.
- [3] Gupta, A. Kumar, S. Qadeer, M. A., "Location based services using android (LBSOID)," IEEE International conference on Multimedia services architecture and applications, pp 1-5, 2009.
- [4] [http://androidexample.com/Android\\_Push\\_Notifications\\_using\\_Google\\_Cloud\\_Messaging\\_GCM/index.php?view=article\\_discription&aid=119&aaid=139](http://androidexample.com/Android_Push_Notifications_using_Google_Cloud_Messaging_GCM/index.php?view=article_discription&aid=119&aaid=139)
- [5] [http://www.tutorialspoint.com/android/android\\_location\\_based\\_services.htm](http://www.tutorialspoint.com/android/android_location_based_services.htm)
- [6] <http://www.eazytutz.com/android/android-architecture/>