

An Overview of Multi-Platform Mobile Application with Mobile Push Message Technology

R. Jayaraj

Assistant Professor Department of Information Systems and Network Engineering, St. Joseph University in Tanzania, Tanzania

Abstract: Many Mobile applications are widely spread with push and pull technology. Mobile applications operate through multimedia data from the server with the mobile network. Any updates of data done for the application in the server uses push mechanism or technology. Central server initiates the transaction, so that the server push can work. New data is fetched on a server with the help of simple mechanism, Push frameworks. This notifies mobile applications to contact the server. Push gateway is implemented and interconnected with the push framework so that the developers can use those frameworks. This paper explains multiple platform mobile applications operating with push technology to integrate diverse existing push messaging gateways and how CPNS send push messages to iPhone, Android and Windows Phone devices. Gateways involved as APNS, C2DM, MPNS, CPNS and 3rd party gateway.

Keywords: mobile applications push framework, framework design, APNS, C2DM, MPNS, CPNS.

1. Introduction

When smart phones were introduced to the world, many mobile applications were been developed [6] and been used in various areas like education[1]and Business[2].But they need more extended push functions than SMS and MMS. They cannot send messages to a specific mobile application. The main aim for mobile push was to deliver the content to users in efficient and flexible manner [7].Mobile applications required data from the mobile network. Polling and Pushing are the two ways of approaches to update data in the applications. The drawback of polling mechanism, it itself polls a server to check new data periodically [11]. Even though there is no new data available on the server, this approach should have a check of server, which cause using unnecessary mobile network bandwidth and consuming the battery of the mobile devices. Also, if there is any extend in polling period, will reduce these drawbacks, a mobile phone user should wait for the next polling period to get new data. Another mechanism is pushing, which the server transfers data to the mobile applications if new data is available on the server [6]. In this approach, the mobile applications do not consume unnecessary network bandwidth and a mobile phone user can get new data without waiting for the next polling period. Therefore, if the data does not updated periodically, pushing is the preferred solution. To push new data to the mobile applications, a push messaging framework was introduced including Apple introduced Apple Push Notification Service (APNS) [1],

Google introduced Cloud to Device Messaging (C2DM) [5], Microsoft introduced MPNS [12]and WNS. It is a service which helps mobile application developers send new data from a server to their mobile applications on iOS [8], Android [11] and Windows devices. This service provides a simple mechanism that a server can notify mobile applications to contact the server to fetch new data on a server.

This paper proposes a new framework to show how those gateways APNS, MPNS, C2DM are integrated along with CPNS and 3rd party gateway, the server side developers do not have to implement many kinds of gateway functions.

Once they have implemented an interface with the new framework, they can send push notification to diverse mobile applications by using APNS, C2DM, MPNS, CPNS or 3rd party push notification framework.

2. Related Works

SMS and MMS were used to push multimedia information to mobile devices [6]. Subscribers are offered with services such that they can exchange messages, images, videos, and photos with the help of mobile network operators. The service providers use 'push', so that the updated information can be pushed in the form of push notifications. With these push notifications; the device does not have to wake up periodically to check whether new data is available. Developers who target iPhone, Android and Windows Phone need a separate data provider for each type of device since they use different protocols for push notifications. Developers have to use the Apple Push Notification Service (APNS) for ios devices, Microsoft Push Notification Service (MPNS) for Windows Phone Devices and Cloud to Device Messaging Framework (C2DM) for Android devices. Mobile app developers need to manage different types of devices and they need to interface with each notification service in order to push messages.

3. APNS

APNS is a push notification service framework made by Apple for iPhone mobile applications [10].Because iOS has not supported any of the background processes; mobile applications could not get messages by the mechanism of polling. So the apps that want to receive messages from the remote servers can use APNS framework. It is implemented in iOS 3.0 or above. Figure 1 shows the flow of the APNS messages. Service provider generates a notification request and sends it to the APNS gateway. The APNS gateway receives the request and forwards it to the APNS server. While the APNS server has active sessions with APNS clients, it can send the request to the ios devices. Then iOS can wake up the specified application and the application can be active and process the request. Since there are air interface between the APNS server and the APNS client, the

message can be dropped easily if the air condition is not good enough or if the device is turned off.

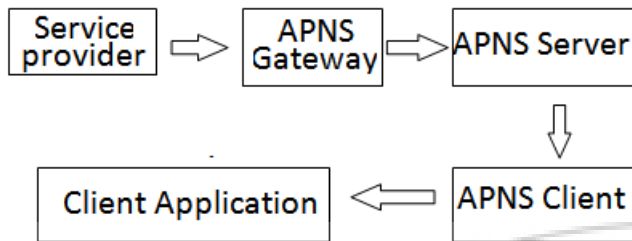


Figure 1:

Apple Push Notification service (APNs) is the centerpiece of the push notifications feature. It provides a highly efficient service for propagating information to ios devices. Each device establishes a qualified and encrypted IP connection with the service and receives notifications over this constant connection. If a notification for an application arrives and when that application is not in running state, the device alerts the user that the application contains the information waiting for it. Software developers originate the notifications in their server software. The provider connects APNS through a persistent and secure channel while monitoring incoming data intended for their client applications. When fresh data for an application arrives, the provider prepares and sends a notice through the channel to APNs, which pushes the notification to the specified device.

In addition to being a simple but efficient and high-capacity transport service, APNs includes a default qos component that provides store-and-forward capabilities.

4. Push Notification and Its Path

Apple Push Notification service transports and routes a notification from a given provider to a given device. A notification is a message in short consisting of two major pieces of data: the device token and the payload. The device token is comparable to a phone number; it contains information that enables APNs to locate the device on which the client application is installed. APN also uses it to validate the routing of a notification. The payload is a property list that specifies how the user of an application on a device is to be alerted.

5. C2DM

C2DM is a push notification service framework made by Google for Android mobile applications. It is developed in Android 2.2 (Froyo) or above. It can post messages to the mobile application which is registered at the Google Play [9]. For using the service, the users should login to their Android phones with their Google accounts. The Flow of the C2DM message is very similar with the flow of the APNS message. Service provider generates a notification request and sends it to the C2DM gateway. The C2DM gateway receives the request and forwards it to the C2DM server. Now the C2DM server has active sessions with the C2DM clients and it can send its request to the Android devices. Then Android can wake up the specified application and the application can be active and process the request. There are some more relevant push technologies for the Android platform, similar to

C2DM.They are XMPP, Xtify and Urban Airship[10].Cloud to Device Messaging (C2DM) was made available from Android 2.2, where the aim was to make it easier for mobile applications to sync data with servers [3]. C2DM is a free service, and the maxi-mum number of messages that can be sent is approximately 200,000 per day [4];

The detail operation mechanism of the C2DM is in Figure 2.

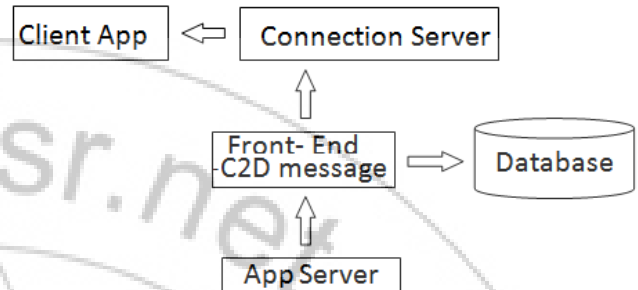


Figure 2: Flow of C2DM

6. MPNS

MPNS is a service for windows phone. Which provides a resilient, dedicated, and persistent channel to send data to a windows phone application [14]&[13].

7. Flow of Service

- 1) An app requests a push notification URI from the MPNS.
- 2) The app sends the notification URI to a 3rd Party server.
- 3) The 3rd party server uses the notification URI to send a push notification to MPNS.
- 4) MPNS routes the push notification to the app.

The MNPS in Windows Phone is an asynchronous; here the best-effort service offers third-party developers a channel to send data to a Windows Phone app from a cloud service in a power-efficient manner. Single connection between the device and notification service. Dedicated persistent channel to send data to a device from a service.

It has three types of push notifications that can be sent to a windows phone app:

- 1) **Toast Notification:** Displays a toast notification, if the app is not running. Tapping on that notification launches the app.
- 2) **Tile Notification:** Visually updates the applications tile.
- 3) **Raw Notification:** Delivers raw data to the app and is Available only when the app in already running.

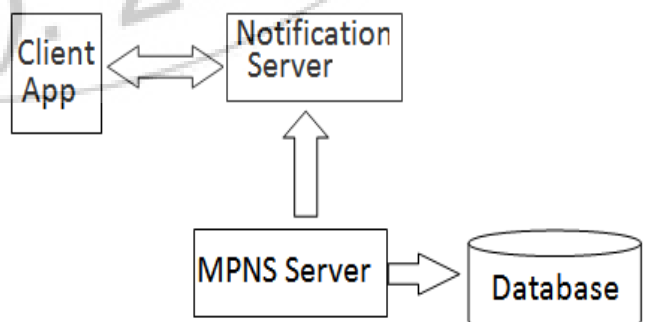


Figure 3: Data Flow of MPNS

8. Common push notification Services(CPNS)

The Microsoft Interoperability Strategy Group is a Microsoft team focused on interoperability. It has developed CPNS Service sample and published it as an open source project that makes it simple to send push messages to iPhone, Android and Windows Phone devices with the appropriate protocol. The CPNS has the following design goals for developers

1. Freedom from the burden of managing different types of devices
2. Freedom for the developers from implementing and operating separate data providers each using different push notification mechanisms.

Ease of extending the notification service to support additional device types in future. Common Push Notification Service sample is implemented as an Azure service. It consists of a web task that exposes REST end points for device, payment and notification management and a worker role.

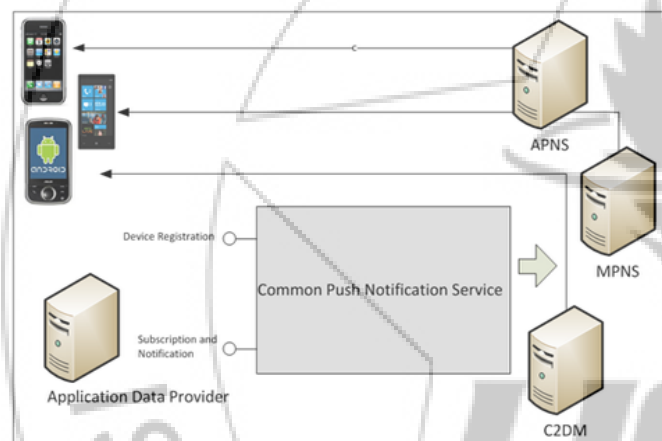


Figure 4: Design of a Sample Common Push Notification Service

A CPNS sample has three primary objects, namely, subscriptions or payment, devices and notifications.

- Subscriptions:** Subscriptions encapsulate a logical group of notifications on a particular topic. Person responsible for the mobile application can also create the subscription in a CPNS.
- Devices:** Users sign up, for the subscriptions using the mobile application, explicitly or implicitly. Users receive notifications through applications on their mobile device.
- Notifications:** Application managers use CPNS to send notifications for subscription. CPNS provides notification messages to devices that are signed up for those subscriptions via the device specific notification service.

8.1 Service End Points

Common Push Notification Service has two ending points:

- Endpoint used by devices to sign up for notification subscriptions.

- Endpoint used by the data provider web service to manage subscriptions and send notifications to devices that are signed up for subscriptions.

8.2 Common Notification Service Engine:

The engine manages the list of subscriptions and the devices that have signed up for the subscriptions. It keeps the device information in order that is needed to send notification messages via each proprietary notification service. It also manages the devices so that the messages to can be sent to them.

a) Message Pump:

Message pumps forward notification messages to Android, iPhone, and Windows Phone devices via C2DM, APNS or MPNS respectively. These notification services forward the messages to the devices. This section guides us through the steps to run the sample service and explains the basics of the service. These sample includes three different steps:

b) Azure Push Messages:

It is a Common Push Notification Service sample.

c) Message Receiver:

A Windows Phone 7 sample application where the user is allowed to sign up for subscriptions and to receive notification messages.

d) Push Message Data Provider:

This is a Windows application that is a sample of a notification data provider. A real data provider is likely to be a web service. Where the application is been allowed sp that the application manager to create subscriptions and send notifications to subscribing devices.

9. Proposed Framework

Since there are many kinds of push notification frameworks including APNS, C2DM, MPNS, CPNS and 3rd party push notification frameworks, all the service provider should have implemented each notification gateway respectively to send multimedia messages to their mobile applications as shown in Figure 5. Most of the mobile devices can lose its connection with a mobile network when they enter the area which are shadowed, the notification messages can also be dropped or stopped. Even though some notification frameworks provide retransmission function, the messages were dropped, because of their retransmission policies might not meet up service provider's requirements.

In this paper, an integrated multimedia push framework is introduced. Where the proposed framework supports APNS, C2DM, MPNS, CPNS and 3rd party push notification framework as well as always-on-based mobile app. A function named store and forward which stores the notification messages when connection is almost lost and messages are send once the connection is provided. If a service provider uses this framework, it does not have to implement all the push notification gateways, since the framework offers interfaces for sending the messages.

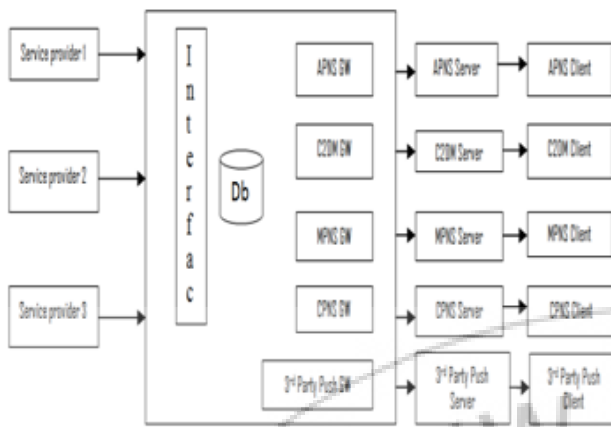


Figure 5: Simple Architecture of APNS, C2DM, MPNS, CPNS and 3rd Party Push Gateway

10. Conclusion

In this paper CPNS Gateways sends messages to APNS, C2DM, MPNS and CPNS itself acts as an gateway for itself. Finally a framework is implemented to show all the gateways with the 3rd party gateway. This proposed framework gives a suitable performance compared with the existing gateways.

References

- [1] Apple Inc., "Apple Push Notification Service", <http://developer.apple.com/library/mac/#documentation/Networkinginternet/Conceptual/RemoteNotificationsPG/ApplePushService/ApplePushService.html>, (2009).
- [2] Erastus Thoronjo "the impact of push and pull mobile technology on business development in africa" *itu news june (2009)*
- [3] Google, "Android Cloud to Device Messaging Framework," 2011. <http://code.google.com/android/c2dm>
- [4] Google, "Android Cloud to Device Messaging: Quotas," 2012. <http://code.google.com/android/c2dm/quotas.html>
- [5] J. Hansen, T. M. Gronli and G. Ghinea, "Cloud to Device Push Messaging on Android: A Case Study", Proceedings of the 26th International Conference on Advanced Information Networking and Applications Workshops, (2012).
- [6] H. Y. Ho and L. Y. Syu, "Uses and gratifications of mobile application users", Proceedings of the International Conference on Electronics and Information Engineering, (2010).
- [7] Ivana Podnar, Manfred Hauswirth and Mehdi Jazayeri "Mobile Push: Delivering Content to Mobile Users"(2009)
- [8] Jarle Hansen¹, Tor-Morten Grønli^{1,2}, Gheorghita Ghinea, "Towards Cloud to Device Push Messaging on Android: Technologies, Possibilities and Challenges", Published Online December 2012 (<http://www.SciRP.org/journal/ijcns>)
- [9] Q. H. Mahmoud and P. Popowicz, "Toward a Framework for the Discovery and Acquisition of Mobile Applications", Proceedings of the Ninth International

Conference on Mobile Business and Ninth Global Mobility Roundtable, (2010).

- [10] I. Podnar, M. Hauswirth and M. Jazayeri, "Mobile push: delivering content to mobile users", Proceedings of the 22nd International Conference on Distributed Computing Systems Workshops, (2002).
- [11] Patricia Thornton and Chris Houser "Using Mobile Phones in Education" Proceedings of the The 2nd IEEE International Workshop on Wireless and Mobile Technologies in Education (WMTE'04), (2004) European Commission under contract IST-1999-10288 (OPELIX).
- [12] B. Sanz, I. Santos, C. Laorden, X. Ugarte-Pedrero and P.G. Bringas, "On the automatic categorisation of android applications", Proceedings of the Consumer Communications and Networking Conference, (2012).
- [13] P. Sutton, R. Arkins and B. Segall. "Supporting Disconnectedness-transparent information delivery for mobile and invisible computing". *proc ieee symposium on cluster computing and the grid* ieee press los Alamitos calif. pp87-90, (2001).
- [14] Vasudevan.R, Chandramohan.K, Manimegalai.s, "Autonomic notification for a context aware system using push server technology", Proceedings of international journal of scientific research and management (IJSRM), (2014).

Author Profile



R. Jayaraj received his B.E. (CSE) in KVCET, 2008 and M.E (Multimedia Technology) from Anna University, Chennai, and 2011. He is currently working in St. Joseph University in Tanzania as Assistant Professor under the Department of Information Science and Network Engineering. His area of interest is networking, multimedia, database and data mining.