Ways to guess if C2DM is connected

Asked 13 years, 2 months ago Modified 12 years, 3 months ago

Viewed 8k times Part of Mobile Development Collective

16

I'm trying to place a best guess estimate as to whether C2DM messages can be received.

I've created an application that relies on pushing information to a phone while it is physically inaccessible. I understand that C2DM isn't guaranteed delivery, but I'd at least like to know when the delivery of a message is even possible; when it isn't we fall back to our own push service (and can actually tell when we're connected).

I've noticed C2DM on android will still issue auth tokens even when there is not a logged in google account; messages still seem to be delivered in this instance even though it's stated that they shouldn't be. If GTalk isn't connected (firewall or other reasons), no response at all is returned when requesting an auth token. Auth tokens are returned to the application when the phone is in airplane mode. This means it's not as simple as checking if an internet is available. I can't find a reliable way of checking if GTalk is logged in.

Again, I don't need to *guarantee* the delivery of messages, but I'd at least like to know if delivery is even *possible*. Does anyone have interesting solutions?

MD android google-talk android-c2dm

Share

Improve this question

Follow

asked Oct 26, 2011 at 6:29

Ian Elliott

7,666 ● 5 ● 36 ● 42

Just need clarification, you want to know server side if you can contact the phone? – Warpzit Oct 26, 2011 at 7:46

Either or would be ok... But main goal was to be able to tell on the phone if it was connected, so that I could open my own TCP connection if necessary. – Ian Elliott Oct 26, 2011 at 7:50

Afaik c2dm relies on market so if you can access market and you have been validated ofcourse, you should be clear. – Warpzit Oct 26, 2011 at 8:11

Being validated and being able to *actively* receive messages are not the same thing. – Ian Elliott Oct 26, 2011 at 8:41

8 Answers

Sorted by: Highest score (default)

11

Go watch this video, it's a Google I/O talk about C2DM, how to use it and how it works. AFAIK, you can't know if it's connected or not. Probably most of the time they don't even know (until they have to deliver a message and fail).

However, it is highly recommended (in the video as well) that you do not send important data through C2DM (as messages can get lost). The service should only be used

+400

as a "network tickle" (with a footprint as small as possible). Your application should be woken up by this tickle and it should start fetching the information it needs **itself**.

Now, if you implement it this way, it should be easy to implement a polling mechanism. Since you already separated the "tickle" from the actual information retrieval, you can just trigger the retrieval every once in a while if there's no tickle.

Something you can do to check whether C2DM is connected is something like a ping:

1. Send message to phone via C2DM
2. The app receives (or doesn't receive) the message and sends a "pong" back to your server
3. The server waits for the "pong" for a predetermined amount of time (1-2 minutes, I'd say) before marking the device as "offline".

Edit: relying on GTalk is not feasible. GTalk relies on C2DM just like your app, it doesn't have anything "extra". Also, GTalk is not present on all devices. I'm not sure how the GTalk app determines whether it's offline or not (it's not open source, unfortunately), but I'd guess it just tries to ping a server and fails.

Share Improve this answer Follow

answered Oct 31, 2011 at 16:46

Felix
89.5k ● 44 ● 151 ● 168

I've already done exactly what you describe, but I'd like to do *more*. Looking at logcat dumps, GTalk knows *exactly* when it's connected, and sends keep-alives every 300 seconds. There's just no way for me to interface with GTalk to ask whether or not it's connected. GTalk does not rely on C2DM, it relies on Jabber, and C2DM messages are bundled into the XMPP specification and delivered over GTalk. Also, if GTalk isn't present, then C2DM will not work (this I can actually detect and account for). – Ian Elliott Nov 1, 2011 at 0:27

How do you know C2DM doesn't work if GTalk is not present? Have you encountered devices that lack GTalk and C2DM? I'm not sure it's guaranteed that if GTalk is not present then C2DM is not available. I can't watch that talk again (it's 1 hour), but from what I remember they said all their apps (Gmail, GTalk, etc.) use the same C2DM connection for push messages. The same connection that your application ultimately uses. GTalk might connect to Jabber when you launch the app / when the screen is on, but that's separate from C2DM. – Felix Nov 2, 2011 at 10:20

Also, I don't understand what you mean by "C2DM messages are bundled into the XMPP specification and delivered over GTalk". AFAIK (and I might be wrong), C2DM is completely separated / different than XMPP or Jabber. And GTalk is just the app. – Felix Nov 2, 2011 at 10:22

1 Further inspection reveals its not really reliant on GTalk, but rather GSF... but again I'm not looking for a guarantee of delivery, just some heuristics to use to determine if delivery is possible. I think the reason google doesn't offer guarenteed delivery is because of the added overhead required to ensure the phone received it. Ex. No network connection -> 0% chance of delivery. Other metrics can improve my *guess*. – Ian Elliott Nov 2, 2011 at 12:39

It depends on *where* you need this information as well. If you need it on the server side, I think the only way of knowing is via the ping method described in my answer. If you need it on the device, yes, you can do some other stuff such as check for a network connection and try a simple request as well (such as an HTTP request to google.com, or your own domain) and see if you get anything. – Felix Nov 2, 2011 at 14:04

7

No that is impossible.As you device authenticated once and generate registration ID and send to third party server(As you already know).Now your work is over once the device has been registered.So Wait for message either you got or not(No guarantee of delivering message as C2DM used UDP Protocol).

Alternative Solution

Although its impossible to check from Google side directly as i mentions above,But if you have any urgency to check connection from your phone then you can take approach like this

Step 1): Make one Web service to check connection

Step 2): Call this web service from application that will command to server to send push notification for checking purpose.

Step 3): Now from server side,server will immediately send push notification for particular device(FROM which it get command)

Step 4): Now if you got push notification that means you are still connected to C2DM.

this will not take much time.But follow it only when checking connection is urgent and it is on user

Share Improve this answer Follow

edited Jun 1, 2012 at 5:18

answered Nov 1, 2011 at 12:43

Tofeeq Ahmad
12k ● 4 ● 62 ● 89

I know it's not possible to check a connection, but it is possible to see if a connection is *likely*. Mostly looking for any data that can hint towards such. – Ian Elliott Nov 2, 2011 at 2:21

I think no Because in this protocol is used that is not responsible for delivery assurance of any packet and it establish a connection when it need to send a message.So i think its not possible to check connection as there is no connection all the time – Tofeeq Ahmad Nov 2, 2011 at 4:04

1

This may be a bit naive as I am not an active C2DM user, but wouldn't it be possible to read

> /proc/net/netstat

and see if there are any active TCP connections. If there aren't any, then C2DM can't possibly be working. You could also make this technique more versatile by forming a C2DM whitelist that you would expect to find (or maybe its possible to filter on a special C2DM port?)

Share Improve this answer Follow

answered Nov 3, 2011 at 20:05

Justin Breitfeller
13.8k ● 4 ● 40 ● 47

Can't try this yet but will soon, sounds promising!
– Ian Elliott Nov 3, 2011 at 20:31

▲

0

▼

If the device is inaccessible, even your fallback push messaging system wouldn't work. The C2DM doesn't guarantee that it will deliver your message, but the event of non-delivery would be very rare. So would be the case with any other service. The best workaround that you could have is to poll your server to check if you have any new messages that hasn't been delivered yet. I am assuming that your application is such that it's very important not to miss even a single message in 500 or may be 1000. In that case, you could implement a hybrid of push and pull.

Share Improve this answer Follow

answered Oct 29, 2011 at 16:45

Kumar Bibek
9,107 ● 2 ● 43 ● 71

Something I feared, but that ruins the push methodology. Still hoping there's an abstract solution that doesn't involve a poll.
– Ian Elliott Oct 29, 2011 at 19:16

2 Well, remember... C2DM's purpose isn't to deliver messages when there's no network connectivity available. C2DM's purpose is to allow the phone to freely go to sleep instead of busy-polling the server for activity, confident that C2DM will allow the server to wake up the phone and trigger intentional communication between the phone and server when it happens. Metaphorically, C2DM is a doorbell (or the network equivalent of an IRQ), NOT a message delivery mechanism in its own right. It wakes up the phone and lets your application know that something interesting has happened on the server. – Bitbang3r Nov 6, 2011 at 17:18

0

I've worked a little with C2Dm, I've created my own push 3rd party server . I've implemented a little logic based oh C2DM http response code to know if a push message was sent or not . Here is some of the code I used :

```
int responseCode = conn.getResponseCode();

    if (responseCode == HttpServletResponse.SC_UNAUTHO
HttpServletResponse.SC_FORBIDDEN) {

        LOGGER.warn("Unauthorized - need token");

        return false;
    }
```

here I'm almost sure that the push message was sent from the c2dm servers because I've got an id ont the response:

```
if (responseParts[0].equals("id")) {
        LOGGER.info("Successfully sent data message to
responseLine);
```

```
        return true;
    }
```

I've used other methods to get other result codes from Google if you want I can post them.I hope that I've helped you a bit .

Share Improve this answer Follow

answered Oct 31, 2011 at 16:34

مجيب moujib
752 ● 2 ● 7 ● 26

1 I'm also doing the same on the back end, but the ID from the response only verifies that the google server received the message, not whether the phone will. C2DM offers **no** guarantee of delivery. – Ian Elliott Nov 1, 2011 at 0:23

0

I don't think there's any way to determine in ADVANCE whether there's any chance a push attempt will work, but I can think of a fairly straightforward way to verify receipt (but not queueing for future delivery via C2DM) -- just complete the message loop.

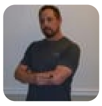
Remember, C2DM's main benefit is that it allows notifications when then phone is asleep and nominally offline. Once your application gets the notification, there's little to stop you from waking up the phone at that point, bringing up the network, and sending a confirmation. I don't think you'd even have to request "keep phone awake" permissions, because I believe the mere act of having registered for C2DM notifications and receiving one is sufficient to wake up the phone and allow the app to continue running normally (at least, long enough to bring up the network and send the confirmation).

While you're at it, you should keep track of confirmations that happen LONG after you expected them to be a lost cause. If you see more than a few, you might have to alter the resend strategy.

The only real-world edge case where this might fail is if you had users who bent over backwards to disable data while leaving voice/SMS enabled (I'm pretty sure C2DM uses 4 bytes of the response datagram sent when a phone polls for incoming calls & text messages that were originally set aside for RIM, then later repurposed for Apple and Google).

Share Improve this answer Follow

answered Oct 31, 2011 at 17:12



Bitbang3r
6,904 ● 5 ● 29 ● 42

This was an idea I had as well, the problem is this method only informs me of the last time a connection was made to the server. It's practically useless to me because it's impossible to tell if the connection still exists without sending another message. – Ian Elliott Nov 1, 2011 at 0:31

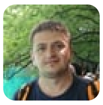
▲
0

try to shut down all network connections and reconnect again. if you will get an registration id, then you can receive messages.

Share Improve this answer

Follow

answered Nov 3, 2011 at 14:43



Buda Gavril

21.6k ● 41 ● 132 ● 201

Registration ID is sent even when there's no network connection... not entirely sure how it's generated and this behaviour seems weird. Registrations are meant to be persistent and are you're not supposed to re-register. – Ian Elliott Nov 3, 2011 at 20:29

0

Something related to the long running C2DM-connection that is used to deliver the triggers:

- On WLAN it sends a heartbeat every 15 minutes.
- On mobile networks the timeout is 28 mintes.

28 Minutes might be to long, depening on the hardware your mobile carrier uses, 2g/3g repeaters in garages, etc.

You can get lots of information about the connection by opening the Google Talk Service Monitor Application: http://www.honeytechblog.com/monitor-google-talk-service-android/

Dial: ##8255##

Theres also a button that sends a heartbeat right now and resets the timeout.

If you want to ensure (on the client side) that c2dm-messages can be received at a given time, your best bet is to re-send the heartbeat. This can be done programmatically - only on rooted devices though. I might release an apk to the market sometime that does exactly that.

Share Improve this answer Follow

answered Sep 12, 2012 at 13:24

icyerasor

5,222 ● 1 ● 47 ● 57