

Maintaining a TCP Connection in the Background

This thread has been locked. Questions are automatically locked after two months of inactivity, or sooner if deemed necessary by a moderator.



1.7k

A common question in both the [iOS Multitasking](#) and [Networking](#) topic areas is: Is it possible to maintain a TCP connection while my app is in the background? For example, you might want to work with a low-level TCP protocol (XMPP is a common request), or interact with a Wi-Fi based accessory.

The short answer to this question is “No”, but there’s a bunch of nuances and I thought I’d take the time to write them down.

Share and Enjoy — Quinn “The Eskimo!” Apple Developer Relations, Developer Technical Support, Core OS/Hardware

```
let myEmail = "eskimo" + "1" + "@apple.com"
```

First and foremost, you should read Technote 2277 [Networking and Multitasking](#). As explained in that technote, this issue is not about being in the background, but rather about being *suspended* in the background. If you app doesn't get suspended, there's no problem maintaining a TCP connection in the background.

TN2277 also explains a bunch of strategies you can use here, for example:

- You can use a UIApplication background task to delay your app from being suspended when it goes into the background. This can be used to:
 - Improve your user experience, by avoiding a reconnect delay if the user leaves your app and returns soon thereafter
 - Give you the opportunity to cleanly close your connection
- Note** For more information about UIApplication background tasks, see [this post](#).
- You can leave the connection open while your app is suspended in the hope that it doesn’t tear (either because of network activity or socket resource reclaim) before you get resumed.

Folks might suggest that you use the legacy VoIP architecture for this. I recommend against that for a bunch of reasons:

- The legacy VoIP architecture only supports outgoing TCP connections; UDP and listening for incoming TCP connections won’t work.
- The legacy VoIP architecture has been deprecated and you should not, in general, write new code that uses deprecated APIs.
- We are actively making it harder for folks to use the legacy VoIP architecture on modern systems; see QA1938 [iOS 10 and the Legacy VoIP Architecture](#) for details.
- For devices with WWAN, the system saves power by disassociating from Wi-Fi when it's not needed. Legacy VoIP apps are expected to transition to WWAN when this happens. This won't be possible if you’re talking to a service on the local Wi-Fi.

One specific thing to watch out for here is the debugger. Attaching with the debugger will prevent your app from being suspended, so you can write a bunch of code, test it in Xcode, have it all work, and then discover it fails when you run the app from the Home screen. When developing code that runs in the background you should regularly perform real world tests, meaning:

- Run the app on a real device, not the simulator
- Run the app from the Home screen

Network
Asked 3 years ago by eskimo

Reply to this question

This site contains user submitted content, comments and opinions and is for informational purposes only. Apple disclaims any and all liability for the acts, omissions and conduct of any third parties in connection with or related to your use of the site. All postings and use of the content on this site are subject to the [Apple Developer Forums Participation Agreement](#).