

# Investigating Network Latency Problems

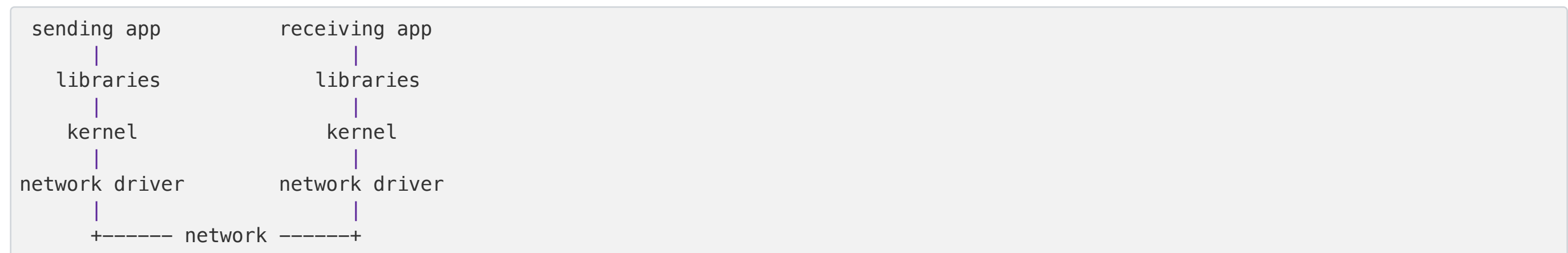
This thread has been locked by a moderator.



5.2k

This post explains how to investigate network latency problems. It does not necessarily show how to fix the problem, but it describes techniques to find the problem and, from there, contemplate potential fixes.

A latency problem in the general case looks like this:



The problem is that that the sending app sends a packet of data which is not received by the receiving app in a timely fashion. This discussion assumes that the sending and receiving apps are both running on Apple platforms (macOS, iOS, tvOS). If you're using another platform, you'll have to ask that platform vendor about equivalent techniques.

**Note** Apple platforms have a user-space networking stack that runs in parallel to the networking stack in the kernel. To simplify this discussion, I'm using the term *kernel* to encompass both stacks.

The first step is to simplify the environment as much as possible. Specifically:

- Test with Bluetooth off — On iOS devices specifically, Bluetooth and Wi-Fi present unique coexistence challenges. If you're primarily interested in Wi-Fi, test with Bluetooth off to see if that affects things.
- Eliminate user space libraries — If you're using a user space library (something like Multipeer Connectivity) it's possible, albeit unlikely, that your latency problem is related to that library. The library itself might have a bug, or you might be using it incorrectly. To eliminate this possibility, switch to a low-level API, either BSD Sockets or `NWConnect ion`.
- Disable peer-to-peer networking — Peer-to-peer networking is a common cause of network latency issues on Wi-Fi, so make sure it's disabled. Peer-to-peer networking is not supported by BSD Sockets, so there's nothing to do there. If you're using Network framework, leave the `includePeerToPeer` [property](#) at its default value of false. Also, check other parts of your code to make sure that you haven't enabled peer-to-peer networking in some other, unrelated place.
- Switch to UDP — TCP is a complex protocol and it can have a non-obvious impact on latency. If you're using TCP, switch to UDP.

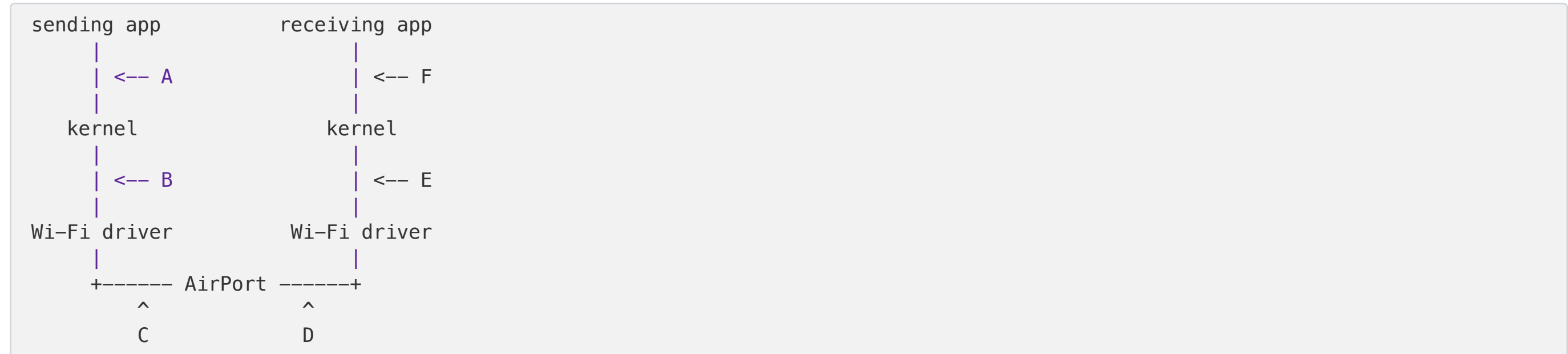
**IMPORTANT** The steps listed above are part of the investigation, not an end in themselves. You may find that, once you understand the real cause of the latency, you go back to using your user space library, or go back to using TCP. However, run your initial investigation with a BSD Sockets or Network framework UDP program.

**IMPORTANT** Turn off Bluetooth using Settings > Bluetooth. On modern systems turning off Bluetooth in the Control Center [does not turn it off completely](#).

The next step is to simplify your network environment. Ideally you'd want to directly connect the two machines, and thus avoid any possibility that network infrastructure is causing the latency. However, if you're working with iOS, which while it supports Ethernet is not commonly used with it, it's likely that you'll need to use at least a Wi-Fi access point (AP). In that case I recommend that you use an Apple AP: an AirPort base station or Time Capsule. These aren't necessarily better than third party APs — although there are a lot of broken APs out there! — but, by using an Apple AP, you guarantee that any problems you see are the responsibility of one vendor.

**Note** While discussing Wi-Fi I tend to slip into the habit of using low-level Wi-Fi terms, like AP. For an explanation of those, see [Wi-Fi Fundamentals](#).

After following the steps above your setup will look something like this.



From there, you can insert probes into the network path to see where the latency is coming from. Specifically:

- Add logging to your app to probe points A and F.
- Use a standard packet trace to probe points B and E.
- Use receive timestamps to probe point E. In BSD Sockets, set the `SO_TIMESTAMP` option and access the timestamps by looking at the `SCM_TIMESTAMP` value returned from `recvmsg`. In Network framework, set the `shouldCalculateReceiveTime` [property](#) and access the timestamps using the `receiveTime` [property](#).
- Use a Wi-Fi level packet trace to probe points C and D.
- On the Mac, use DTrace to probe between A and B and between E and F.

**IMPORTANT** Use `tcpdump` on your Mac to record a packet trace. If you're working on iOS, set up an RVI interface. Both of these are explained in [Recording a Packet Trace](#), but you will also want to look at the `tcpdump` [man page](#). For instructions on how to record a Wi-Fi level packet trace, see [Recording a Wi-Fi Packet Trace](#).

With all of these probes in place you can understand where the packet got delayed. For example:

- A delay between A and B would be pretty unusual when using UDP, but could be the result of congestion within the kernel's TCP/IP stack.
- If there's a delay between B and C, you know that the sending device is having problems sending, either because of a problem within the Wi-Fi driver or because of a Wi-Fi level problem, for example, link-layer retransmissions. To investigate the latter in more depth, use a Wi-Fi level packet trace.
- If there's a delay between C and D, that could be an AP problem, an issue with Wi-Fi QoS, or the receiving Wi-Fi driver entering low-power mode. Again, the Wi-Fi level packet trace will help you understand this.
- A delay between D and E is most likely caused by the receiving Wi-Fi driver but there could be other causes as well, like link-layer retransmission.
- A delay between E and F could be caused by a bug in the kernel, congestion within the TCP/IP stack, or a thread scheduling problem within your app. In the last case, use the System Trace instrument to investigate thread scheduling delays.

Once you understand the cause of your latency, you can then think about how to reduce it. This might be something you can do yourself. For example, you might uncover a thread scheduling bug in your app. OTOH, the fix might be something that only Apple can do. For example, this might be a bug in the Wi-Fi driver and so all you can do is [report that bug](#).

Share and Enjoy

Quinn “The Eskimo!” @ Developer Technical Support @ Apple  
`let myEmail = "eskimo" + "1" + "@" + "apple.com"`

Revision history:

- **2022-01-25** — Added a discussion of peer-to-peer Wi-Fi. Mentioned Network framework as an alternative to BSD Sockets. Added a note about the user-space networking stack. Made other editorial changes.
- **2021-02-27** Fixed the formatting. Made minor editorial changes.
- **2019-03-01** Fixed some links now that QA1176 has been retired.
- **2018-09-11** Added a description of how to really turn off Bluetooth, along with some minor editorial changes.
- **2016-01-19** Made more editorial changes.
- **2015-10-09** Made minor editorial changes.
- **2015-04-03** Added a section about turning off Bluetooth.

NetworkCore WLAN

Reply

Posted 7 years ago by eskimo

Add a Comment

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](#).

Developer Forums

## Platforms

iOSiPadOSmacOStvOSwatchOS

## Tools

SwiftSwiftUISF SymbolsSwift PlaygroundsTestFlightXcodeXcode Cloud

## Topics & Technologies

AccessibilityAccessoriesApp ExtensionsApp StoreAudio & VideoAugmented RealityBusinessDesignDistributionEducationFontsGamesHealth & FitnessIn-App PurchaseLocalizationMaps & LocationMachine LearningSecuritySafari & Web

## Resources

DocumentationCurriculumDownloadsForumsVideos

## Support

Support ArticlesContact UsBug ReportingSystem Status

## Account

Apple DeveloperApp Store ConnectCertificates, IDs, & ProfilesFeedback Assistant

## Programs

Apple Developer ProgramApple Developer Enterprise ProgramApp Store Small Business ProgramMFi ProgramNews Partner ProgramVideo Partner ProgramSecurity Bounty ProgramSecurity Research Device Program

## Events

App AcceleratorsApp Store AwardsApple Design AwardsApple Developer AcademiesEntrepreneur CampTech TalksWWDC