

## Investigating Network Latency Problems

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



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

A latency problem in the general case looks like this:

2.7k



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.

The first thing to do here is to simplify the environment as much as possible. Specifically:

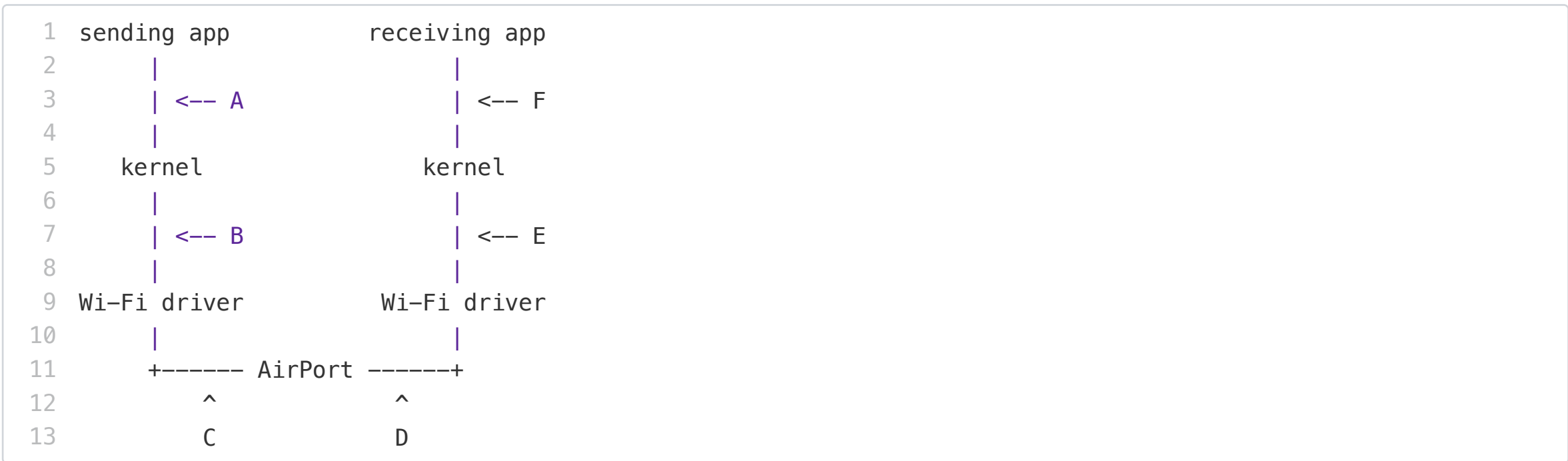
- Test with Bluetooth off — On iOS devices specifically, Bluetooth and Wi-Fi present unique coexistence challenges. If you're primarily interested in Wi-Fi, test with Bluetooth off to see if that affects things.
- Eliminate user space libraries — If you're using a user space library (something like Multipeer 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 using the lowest-level API possible, BSD Sockets.
- Switch to UDP — TCP is a complex protocol and it can have a non-obvious impact on latency. If you're using TCP, you should switch to UDP.

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

**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, to investigate the problem you should test using a BSD Sockets UDP program.

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 on Ethernet, it's likely that you'll need to use at least a Wi-Fi access point (AP). In that case I strongly 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.

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:

- You can add logging to your app to probe points A and F.
- You can use a standard packet trace to probe points B and E.
- You can use the **SO\_TIMESTAMP** option to probe point E.
- You can use a Wi-Fi level packet trace to probe points C and D.
- On the Mac, you can use DTrace to probe between A and B and between E and F.

**IMPORTANT** If you're working on the Mac, you can take a standard packet trace using **tcpdump**. If you're working on iOS, you can take a standard packet trace using RVI. 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). You can investigate the latter in more depth using the Wi-Fi level packet trace.
- If there's a delay between C and D, it could be an AP problem, an issue with Wi-Fi QoS, or that the receiving Wi-Fi driver has entered a 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, the System Trace instrument can help you understand thread scheduling delays.

Once you understand the cause of your latency, you can then think about how to reduce it. This might be very simple (for example, fixing a thread scheduling bug in your app) or quite hard (reporting a bug in the Wi-Fi driver so that Apple can fix it).

Share and Enjoy

Quinn "The Eskimo!" @ Developer Technical Support @ Apple

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

Changes:

- 3 Apr 2015 — Added a section about turning off Bluetooth.
- 9 Oct 2015 — Minor editorial changes.
- 19 Jan 2016 — More editorial changes.
- 11 Sep 2018 — Update to describe how to really turn off Bluetooth, along with some minor editorial changes.
- 1 Mar 2019 — Fixed some links now that QA1176 has been retired.
- 27 Feb 2021 — Fixed the formatting. Minor editorial changes.

Network

Asked 4 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](#).