

Wi-Fi Fundamentals

!

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



Wi-Fi is *way* more complex than you might think. This post attempts to explain some of that complexity, and how that affects the design of your network apps.

Note I'm not a Wi-Fi expert. All of the following is the result of hard-won experience gained while investigating various oddball Wi-Fi problems. As a result it's a *vast* simplification of how things really work. If you actually get to talk to a Wi-Fi expert, they'll be happy to explain to you how Wi-Fi is even more complex than what I've explain below.

Terminology

As this post is going to talk about the fundamentals of Wi-Fi, I'm going to use Wi-Fi technical terms. Specifically:

- STA** (station) — This is a Wi-Fi 'client' device.
- AP** (access point) — This is the hardware running a single Wi-Fi network. The definition of *Wi-Fi network* is more complex than you might think, as I'll explain next.
- SSID** (Service Set Identifier) — This is what most folks think of as the Wi-Fi network. It's the user-visible network identifier string that you see throughout the system.
- BSSID** (Basic Service Set Identifier) — This defines a single Wi-Fi network at the Wi-Fi level. It's identified by the [MAC address](#) of the AP, something that's generally not user visible.

In a typical home Wi-Fi network there's a one-to-one relationship between SSID and BSSID. This is not true in more complex Wi-Fi setups. For example, in my home I have an Ethernet backbone with multiple APs bridged on to that backbone. Each AP has a different BSSID, but they all share the same SSID so that STAs can roam between APs without disrupting their network.

This sort of setup is very common in enterprise environments.

I also use various terms that are less widely accepted but are, nevertheless, important when discussing common scenarios:

- Wi-Fi hotspot** — This is a Wi-Fi network where the user must interact with the network to gain access to the wider Internet (1).
- Wi-Fi accessory** — This is an accessory which communicates over Wi-Fi. I use *accessory* in favour of *device* because, when working in the Apple ecosystem, *device* refers to your iOS device.

Finally, I don't use the term *ad-hoc Wi-Fi*. In my experience this term is so heavily overloaded as to be meaningless. See the next section for more.

(1) Apple APIs are not as consistent about this as they should be. For example, the *hotspot* in [NEHotspotHelper](#) is correct but the *hotspot* in [NEHotspotConfigurationManager](#) is not (the API can be used to configure the device to join any Wi-Fi network, not just a Wi-Fi hotspot).

Ad-Hoc Wi-Fi

I don't use the term *ad-hoc Wi-Fi* because, in my experience, this term means different things to different people:

- Some folks interpret it to mean [IBSS](#).
- Some folks interpret it to mean [Wi-Fi Direct](#).
- Some folks interpret it to mean Apple peer-to-peer Wi-Fi (aka AWDL or its predecessor). This is the mechanism used by Network framework when you set the [includePeerToPeer](#) flag, Multipeer Connectivity, and so on.
- Some folks interpret it to mean an infrastructure Wi-Fi network that doesn't lead to the wider Internet, for example, one published by a Wi-Fi accessory.

Given this confusion it's best to avoid this term in favour something more specific.

Unicasts

Wi-Fi implements a link-level positive acknowledgement mechanism for unicast traffic. This is really important because the physical packet loss on a Wi-Fi network is pretty bad.

In Wi-Fi, all unicast traffic is from STA to AP or vice versa. This makes sense when you think about it. You can't send from STA to STA because:

- The STAs might be located such that each STA can see the AP but the STAs can't see each other (for example, this might be a home network with the AP located in the middle of the home and the STAs located on the extremities)
- The STAs might be talking to different APs (that is, they're on different BSSIDs)

Wi-Fi unicast traffic is fast because the AP can set the speed of the link to be appropriate for the STA in question.

Some APs refuse to forward STA-to-STA traffic. This is most often seen with Wi-Fi hotspots, where the hotspot isolates each STA as a security measure (this is, IMO, [security theatre](#) but there you go).

Broadcasts

Note In this context, *broadcasts* also includes *multicasts*.

Wi-Fi broadcasts work very differently from Wi-Fi unicasts. In a broadcast, the STA sends the packet to the AP and the AP then transmits the broadcast and hopes that all the other STAs pick it up.

The AP does two things to help improve the chances that the STAs will pick up the broadcast:

- It sends the broadcast at the lowest supported speed — This makes sense when you think that the AP might have a mix of STAs, some of which support high speed modes and some of which don't.
- It typically ramps up its transmission power.

These measures help, but they don't guarantee that all the STAs will pick up the broadcast.

If the network has multiple APs, the AP will typically forward the broadcast to the other APs and they will also broadcast the packet. However, this does not always happen. Many organisations have large flat networks, and thus put a limit on Wi-Fi broadcasts to prevent the whole network being flooded with broadcasts. In fact, the AP might not even forward broadcasts from its own STAs (for example, a hotspot that implements STA isolation as I discussed earlier).

IMPORTANT When you're designing a network protocol that will commonly run over Wi-Fi, you must take into account the peculiarities of Wi-Fi's broadcast support. For example, if you're only transmitting to a few peers (less than 10 say), it may be better to send a unicast to each peer rather than send a broadcast; the unicasts may be faster (because Wi-Fi will send each one at the highest speed supported by that peer) and will certainly be more reliable.

Power Managerment

A STA will often turn off its radio in order to save power. When this happens the STA sends the AP a packet telling it how long it's going to have its radio off, and the AP buffers packets for that STA for the duration. Cool beans!

This feature is also used to support radio and antenna multiplexing. On iOS there are two scenarios where that's necessary:

- iOS devices commonly have a single antenna for Bluetooth and Wi-Fi, so the device must periodically turn off Wi-Fi so it can use the antenna for Bluetooth.
- If the device has a single Wi-Fi radio (which is common), it may need to change the channel on that radio in order to deal with peer-to-peer Wi-Fi.

It should go without saying that, if the AP sends a broadcast while the STA isn't listening, the STA won't see that broadcast.

Examining Wi-Fi Mechanics

If you're interested in seeing how Wi-Fi *really* works, you can take a Wi-Fi level packet trace using the instructions in [Recording a Wi-Fi Packet Trace](#). This will show you STA-to-AP traffic, AP-to-STA traffic, link-level positive acknowledgement and retransmission, Wi-Fi power management, and so on.

Share and Enjoy

—

Quinn "The Eskimo!"
Apple Developer Relations, Developer Technical Support, Core OS/Hardware
`let myEmail = "eskimo" + "1" + "@apple.com"`

Change history:

- 18 Apr 2016 — First posted.
- 1 Mar 2019 — Fix a link to QA1176, which is no more. Minor editorial changes.
- 11 May 2021 — Added the *Ad-Hoc Wi-Fi* section. Expanded the *Terminology* section. Minor editorial changes.

Network

Core WLAN

Asked 5 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](#).