Understanding of 802.11 connection process

A basic configuration of a BSS (Basic Service Set) operating in infrastructure mode consists of a station (STA) and an access point (AP) attached to a backbone network.

# Wi-Fi network discovering

There are two ways that a STA detects the presence of an AP:

* **Passive scan** - using beacon frame: the APs broadcast periodically the frames that identify themselves as well as provide information about the networks behind. We call this passive scan, because the Wireless NIC (WNIC) receives these frames without sending any data. Default beacon broadcasting interval is set to 100TU (1TU = 1024 μs).

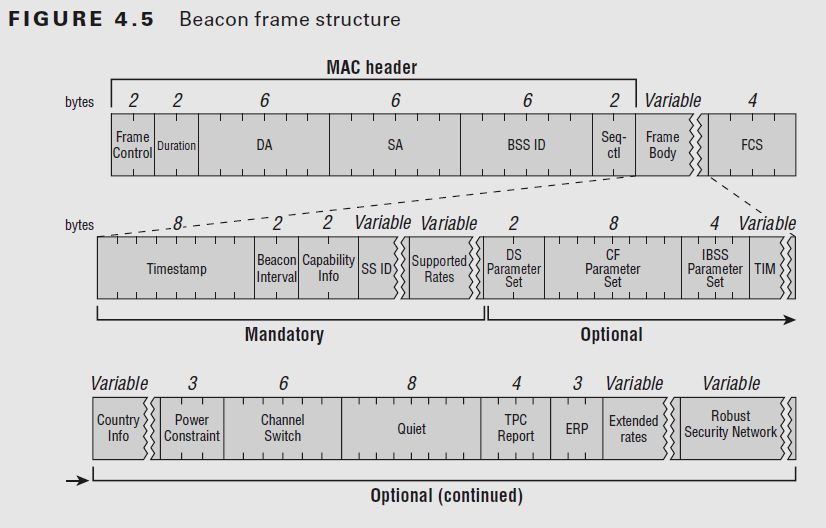


Figure 1.1 – Structure of a beacon frame

* **Active scan** - using probe request/response: the STA actively scans for APs by sending probe request on each channel. Probe request is a broadcast Ethernet packet containing STA information and capabilities (supported data rates, channels). If an AP realizes that it has the **same capabilities**, an unicast packet is responded to the STA. MinCT is the maximum waiting time of a probe response, and MaxCT is the additional waiting time for other responses if there were any probe response received during MinCT.

A full network scan is an active scan that the probe is sent on every channel, trying to discover as many of access points as possible. This is obviously useful in a desktop environment, but unnecessarily being used in a real-time context.

Instead, the STA can perform a scan only on non-overlapping channels (ex. 1-6-11), on which it can be sure that there must be some access points operating, which are predefined by system architects.

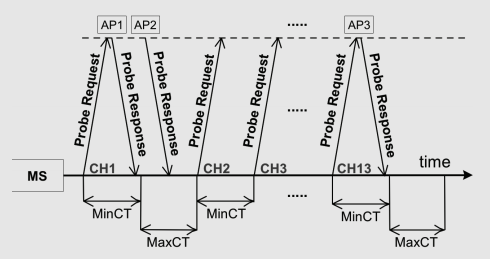


Figure 1.2 – Active scan process

In addition, MinCT and MaxCT can also be reconfigured, to be adapt with the current scan result, which reduces scanning time even better [2].

# Wi-Fi network authenticating and associating

## Two basic phase of a connection

After an active scanning process, the STA has a list of “connectable” access points.

Suppose that our STA wants to connect to an access point AP. This process of two phases then has to be happened:

* **Authenticating phase:** this phase is used to authenticate a STA to an AP. In the past, there were two methods of authenticating:
  + Open System: this method performs *no client verification*. At the birth of 802.11, this method allows any STA to connect with the AP without providing any password, creating an open network.\*
  + Pre-shared key: in this method, the STA and the AP agree on a passphrase that will be used to challenge the STA each time it wants to connect with the AP. This method was then deprecated because of its security vulnerabilities.

In this phase, the STA sends an *authentication request*, and the AP responds with an *authentication response* indicating the success of authentication process.

After authenticating itself to the AP, the STA goes into “*authenticated, unassociated state”.* Both the STA and the AP save this information.

*\*Nowadays, modern Wi-Fi standards use Open System as their default authentication method, along with an Extended Authentication Protocol (EAP) that happens after the association phase.*

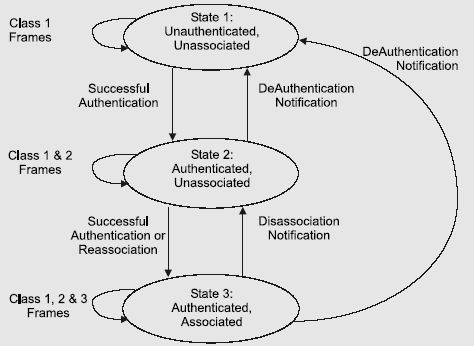
****

Figure 2.1 – 802.11 connection state machine

* **Association phase:** this phase happens when a STA has already been authenticated (aka being in “authenticated, unassociated” state) and decides to connect to the desired AP. This is when the AP and the STA agree on the common parameters that will be used for the communication later on.

In the same way, the STA sends an *association request* to the AP it wants to connect, and the AP responds with an association response. Depend on the configuration, the AP can either accept or refuse the association request (ex. overloading, weak security policy...).

After successfully associated, the STA is in “authenticated, associated” state. From now on, it is connected to the Wi-Fi network.

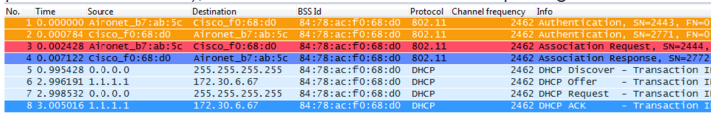
****

Figure 2.2 – Authentication and association frame exchanging

## High-level authentication protocols

This is what happens after a STA has successfully been connected to a Wi-Fi network (authenticated, associated). If the security configuration is set, before that any data frame could be forwarded between the STA and the network, another “high-level” authentication phase will take place.

Again, it is pre-shared key with *4-way hand shake* or *key exchanging protocol* through *EAPoL* (Extended Authentication Protocol over LAN).

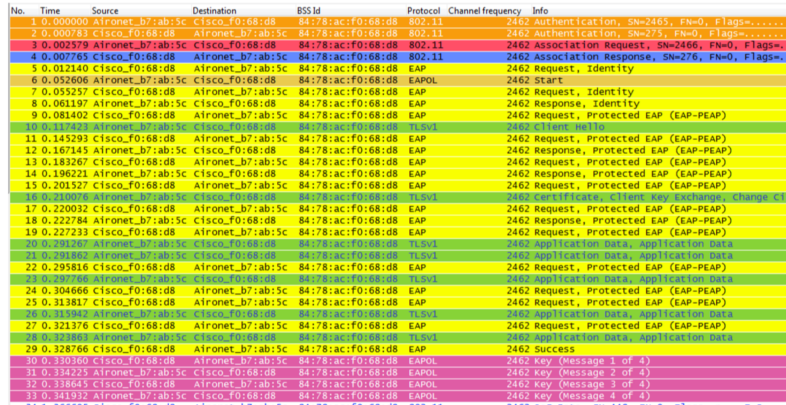


Figure 2.3 – WPA-EAP key exchanging after successful association

# Dis-authentication, Roaming and Fast Secure Re-association

## Dis-authentication

* **Active leaving**: when a STA wants to leave an AP, or an AP decides to stop serving the STA, a dis-authentication frame will be sent with info about the reason. After being notified, the status of the STA is set back to “unauthenticated, unassociated”.
* **Passive leaving:** in case of brutal corruption of AP, the STA must wait for a certain time before notice that the signal has been lost. How?  
  Remember of beacon frame. The beacon frame is the way the AP confirms its presence to the STA. Missing AP beacon frame, the STA will try to scan all the network to find the AP again. If the AP is not found in the result, the connection is considered to be lost.

## Re-association

When a lost-signal STA tries to reconnect itself to the network, it must completely perform those phases that described above, including the key exchanging protocol to obtain a new encryption key. The only difference is that, if the new AP is in the same network with the old one, a re-association frame is sent instead.

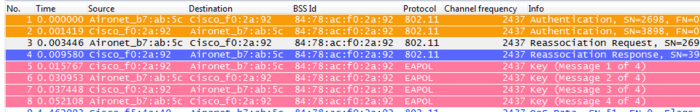


Figure 3.1 – Re-association frame exchanging with PSK

Note that a STA can only associate with one AP at a time. The re-association request notifies the new AP that the STA is roaming within the network from another AP. This allows the network to reallocate the necessary resources for handling the new STA (ex. remove the information of the STA in the previous AP).

## Fast-Secure Roaming

Fast-Secure Roaming attempts to reduce the frames exchanged during re-association phase, with reduces the overall roaming time. This technique is based on caching necessary information at first-time-authentication, then reuse it later. These information are embedded inside the re-association request and response,

There are several methods to achieve Fast-Secure Roaming. Cisco invented its own algorithm named Cisco Centralized Key Management (CCKM) which is only supported on some Cisco devices. Nevertheless, CCKM is one of the-best-out-there for fast secure roaming, thanks to these following advantages:

* CCKM is the fastest fast-secure roaming method mostly deployed on enterprise WLANs. Clients do not need to go over a key management handshake in order to derive new keys when a move between APs takes place, and are never again required to perform a full 802.1X/EAP authentication with new APs during the client lifetime on this WLAN.
* CCKM supports all of the encryption methods available within the 802.11 standard (WEP, TKIP, and AES), in addition to some legacy Cisco proprietary methods still used on legacy clients.

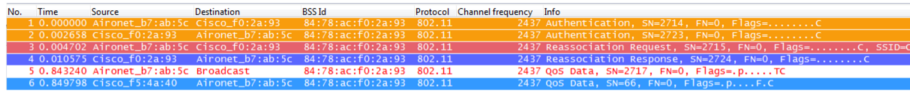


Figure 3.2 – Re-association with CCKM enabled

# Improvement suggestions

Linux’s default behavior tents to perform a full network scan each time it recognizes any change (a disconnection, reconnection, beacon of new APs), which slowdowns the connection process. This can be simply replaced by a fix frequency scan.

In case a full network scan is required, an adaptive timer scanning can be performed.

CCKM should be also taken into consideration for purpose of faster roaming.

**REFERENCES**

[1] Fast secure roaming

<https://supportforums.cisco.com/document/9879826/80211-wlan-roaming-and-fast-secure-roaming-cuwn>

[2] **German Castignani, Andres Emilio Arcia Moret, Nicolas Montavont.** *A study of the discovery process in 802.11 networks.* ACM Sigmobile - Mobile computing and communications review, 2011, 15 (1), pp.25-36. <10.1145/1978622.1978626>.<hal-00609309>