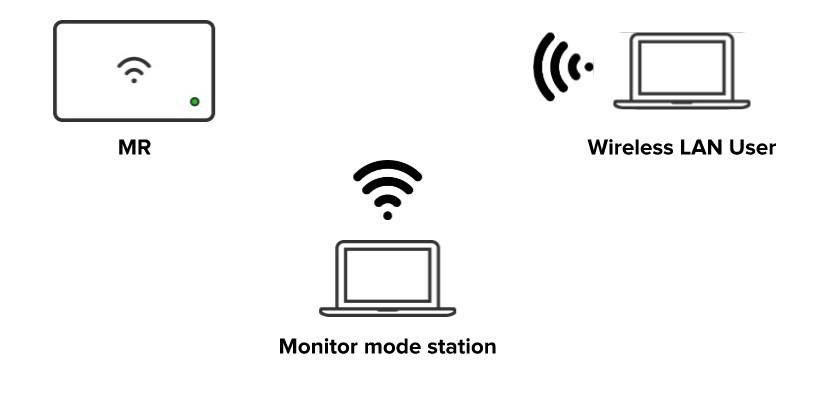
The main purpose of the document is to give an understanding of the 802.11 packet structure and how to analyze wireless packet captures. Wireless packet captures are an important part of troubleshooting complex wireless connectivity issues. This document can be a good tool to reference if you have acquired wireless packet captures and need to analyze them.

There are multiple considerations in wireless communication which make it different as compared to wired packet captures. There are several important considerations, like the ever-evolving 802.11 wireless protocol, wireless signal interference, and the continuous sniffing for wireless packets in your radio environment. This document will discuss the analysis of wireless packets and challenges in analyzing them, with the help of packet capture examples.

## Monitor Mode for Wireless Packet Captures



There are different wireless card modes like *managed*, *ad-hoc*, *master,* and *monitor* to obtain a packet capture. **Monitor mode** for packet captures is the most important mode for our purpose as it can be used to capture all traffic between a wireless client and AP. A client running Wireshark in monitor mode would listen to all packets it can hear in the air and the device will stop transmitting and receiving any of its own data.

## 802.11 Packet Structure

The main difference between wired and wireless packets is the addition of an 802.11 header. There are three major 802.11 frame types.

* **Management frames**: These packets are used to discover APs and to join a BSS. Some important subtypes are ***Beacon***, ***Probe Request*** *&* ***Response***, ***Authentication*** *&* ***Deauthentication***, ***Association***, and ***Disassociation***
* **Control frames**: These packets are used to acknowledge successful transmission and reserve the wireless medium. Control packets allow for the delivery of management and data packets. Common subtypes are *ACK*, *request-to-send*, and *clear-to-send*
* **Data frames**: These packets contain actual data and are the only packets that will be forwarded from the wireless to the wired network. Types of data frames include *Data*, and *null function*

## Physical Layer

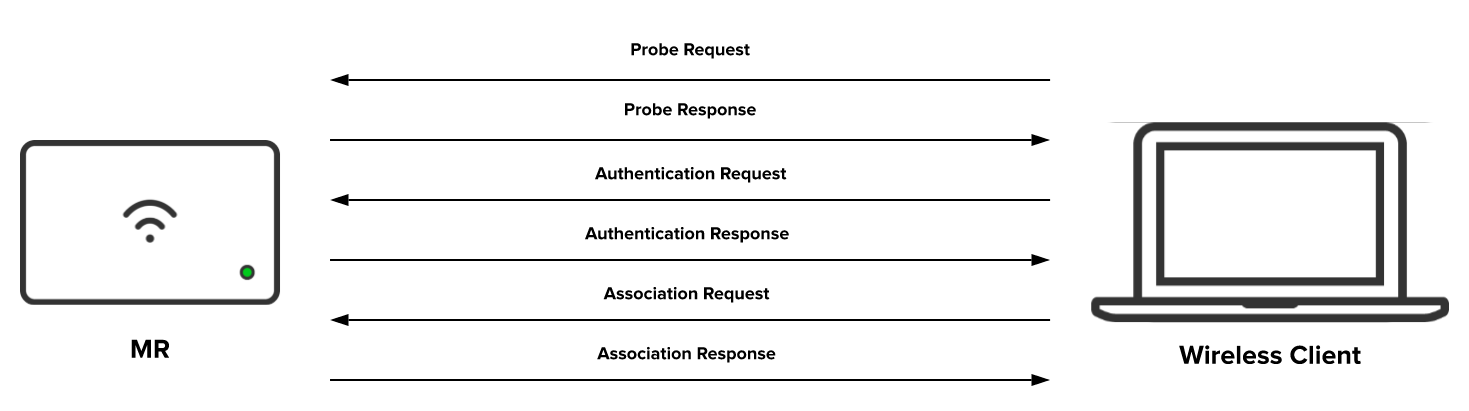
Unlike wired packet analysis, the wireless physical layer is more complex. It is important to get an understanding of the physical layer of the capture before diving into the capture to analyze the upper layers. Some important physical layer values you need to be aware of are *channel*, *data rate*, and *signal strength*.



* **Channel (frequency):** As a wireless LAN may support anywhere from 3 to 25 different channels, it’s crucial to know exactly which channel(s) your capture was taken from. If your intention is to get a capture focused on a specific AP, then lock your capture to that AP’s channel, and validate that the capture was on that channel
* **Data rate:** To understand why data transmissions don’t always make it from a transmitter to receiver, you must know what data rates are being used
* **Signal strength**: Signal strength indicates the power level in dBm at which the sniffing adapter received the packet. This information is useful for determining the expected quality of the signal

| **Signal strength** | **Expected Quality** |
| --- | --- |
| -90dBm | Chances of connecting are very low at this level |
| -80dBm | Unreliable signal strength |
| -67dBm | Reliable signal strength– the edge of what Cisco considers to be adequate to support Voice over WLAN |
| -55dBm | Anything down to this level can be considered excellent signal strength. |
| -30dBm | Maximum signal strength, you are probably standing right next to the access point. |

## 802.11 Client Authentication Process

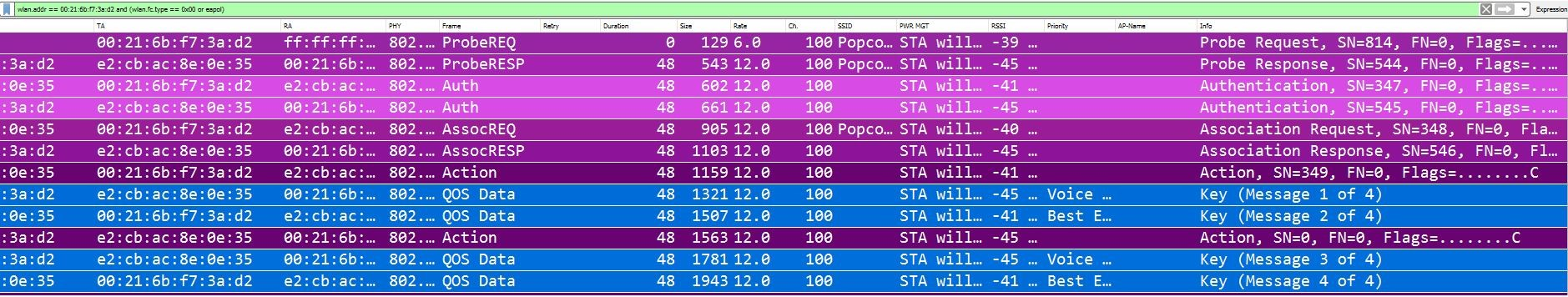


Based on 802.11 specifications, the client authentication process consists of the following:

1. Access points (APs) continuously send out **beacon frames** which are picked up by nearby WLAN clients, advertising their SSIDs and data rates
2. The client broadcasts **probe request** frames on every channel, to all APs. Probe requests advertise the mobile stations supported data rates and 802.11 capabilities such as 802.11n
3. Access points within range respond with a **probe response** frame, advertising the SSID (wireless network name), supported **data rates**, encryption types if required, and other **802.11 capabilities** of the AP
4. The client decides which AP is the best for access (based on compatibility with received probe responses) and sends an **authentication request** to the AP it deems best to connect to
5. The access point sends an **authentication reply,** inviting the client to authenticate to the SSID
6. Upon successful authentication, the client sends an **association request** frame to the access point
7. The access point will reply with an **association response** with a success message, granting network access to the client
8. The client is now able to pass traffic to the access point

Here is an example of a complete client authentication process from the above packet capture.

The packet capture is shown here in Wireshark. The display filter used was **"wlan.addr == 00:21:6b:f7:3a:d2 and (wlan.fc.type == 0x00 or eapol)"**

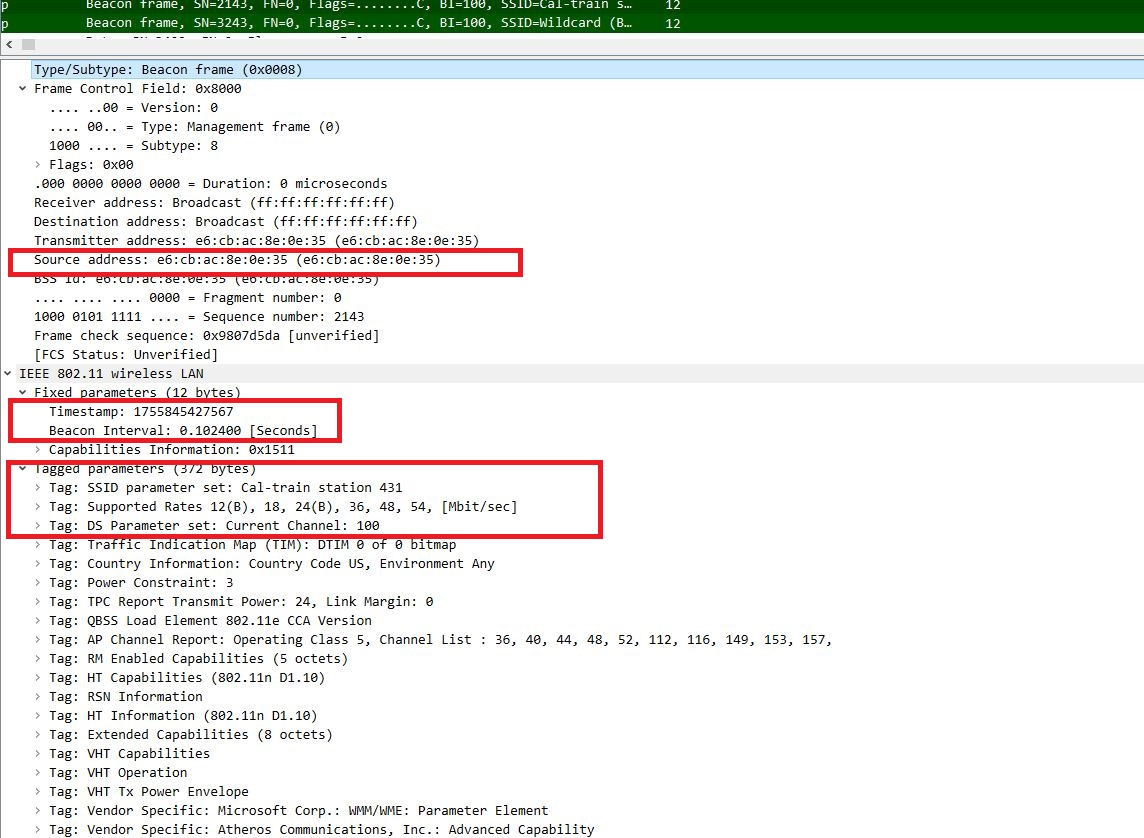
****

**As mentioned above in the 802.11 packet structure section, there are 3 types of frames used in 802.11 MAC layer communications happening over the air which manage and control the wireless link. The *management*, *control*, and *data frames*. The sections below investigate those frames with the help of packet capture given above.**

## Beacon Frame

**The beacon frame is one of the most information-dense wireless packets. The access point sends a beacon frame as a broadcast to announce its presence to any wireless clients. It relays information about the parameters that must be set on the client side in order to connect to it. Radio NICs continually scan all 802.11 radio channels and listen to beacons as the basis for choosing which access point has the best signal and availability to associate with.**

**The Wireshark display filter for Beacon packets is “wlan.fc.type\_subtype == 0x08”**

****

**SSID parameter set:** The SSID (network name) broadcasted by the access point

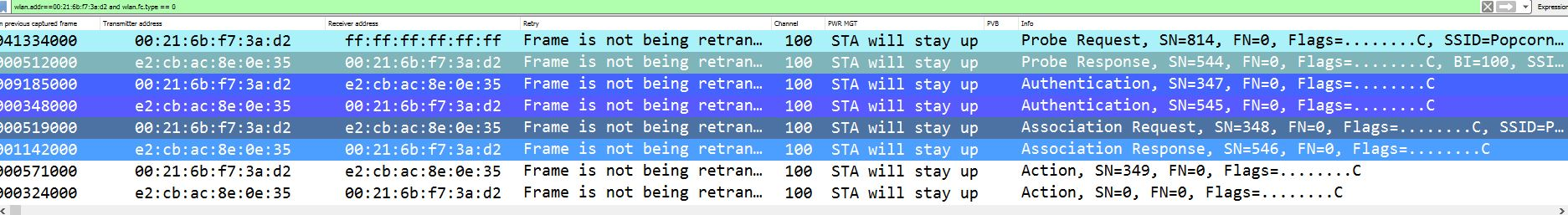
**Supported rates:** The data transfer rates supported by the access point

**DS parameter set:** The channel on which access point is broadcasting

### Management Frames

802.11 management frames enable stations to establish and maintain communications. Management packets are used for **authentication**, **association**, and **synchronization**.

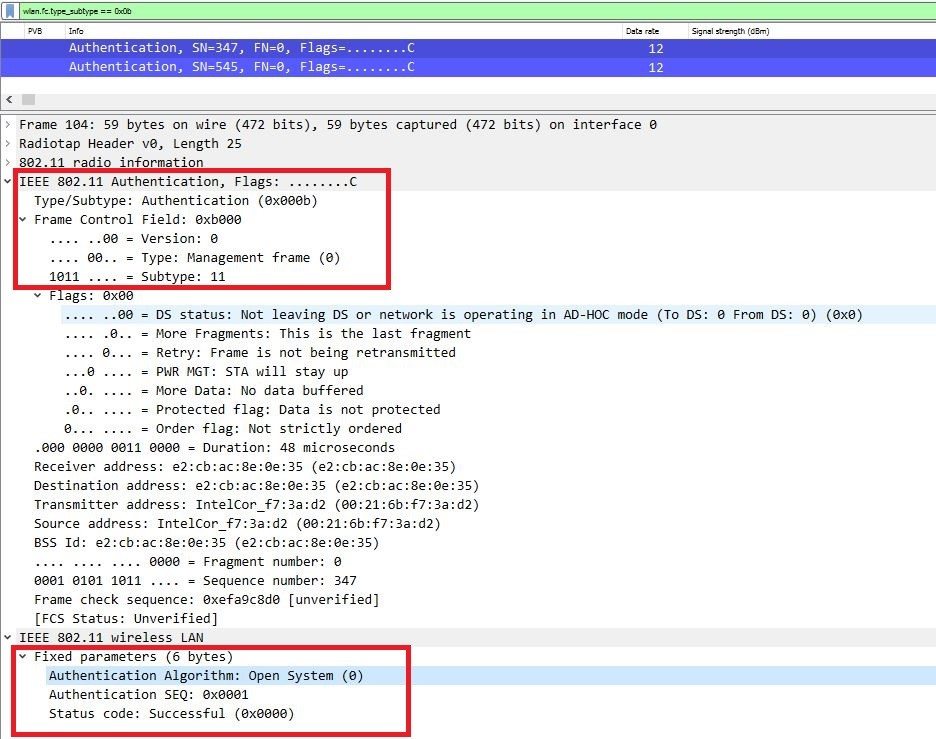
The Wireshark display filter for **Management packets** is **“wlan.fc.type == 0”**



##### Authentication

The wireless client begins the process by sending an authentication frame containing its identity to the access point. With open system authentication (the default), the radio NIC sends only one authentication frame, and the access point responds with an authentication frame as a response indicating acceptance (or rejection). Authentication is handled by a request/response exchange of management packets. The number of packets exchanged depends on the authentication method employed.

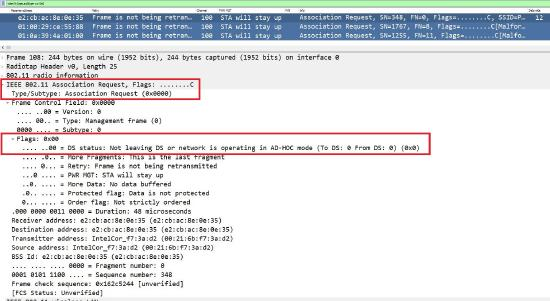
The Wireshark display filter for **Authentication packets** is **“wlan.fc.type\_subtype == 0x0b”**



##### Association

Association frames are also **request/response** type frames. The association process enables the access point to allocate resources for and synchronize with the wireless client. These frames carry information about the NIC of wireless clients (e.g., supported data rates) and the SSID of the network it wishes to associate with. After receiving the association request, the access point considers associating with the client. If accepted, it reserves memory space, establishes an association ID, and sends an association response back to the client.

The Wireshark display filter for **association request** **frames** is **“wlan.fc.type\_subtype == 0x00”** and **Response frames** is **"wlan.fc.type\_subtype == 0x01"**

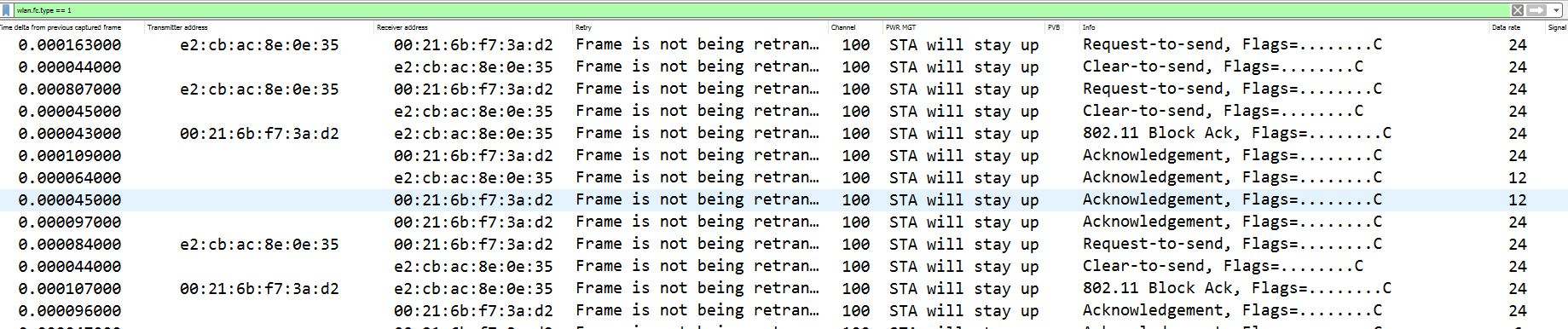


The **reassociation request** frame is similar to an association request but has a different purpose. It is mainly useful in client roaming situations. If a **client roams away** from the currently associated access point and finds another access point with a stronger beacon signal, the client will send a reassociation frame to the new access point. The new access point then coordinates the forwarding of data frames that may still be in the buffer of the previous access point, waiting for transmission to the radio NIC. The sender must already be authenticated in order to gain a successful association.

### Control Frames

The **RTS/CTS function** is optional and reduces frame collisions present when hidden stations have associations with the same access point. **ACK** is sent from one station to another after receiving a data frame and no errors are found in the data frame. If ACK is not received by the sending station then it will retransmit the frame.

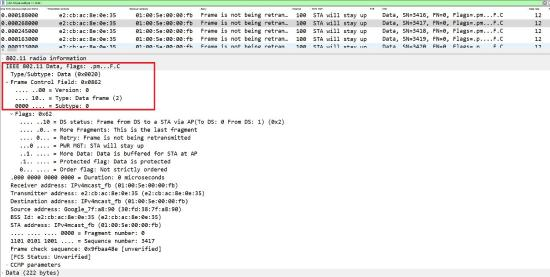
The Wireshark display filter for **control frames** is **wlan.fc.type == 1**



### Data Frames

Data frames come later in the communication process, when the WLAN communication has already been established between client and AP. In the packet capture, observe the contents of the frame body within 802.11 data frames for interesting/relevant traffic.

The Wireshark display filter for **data frames** is **"wlan.fc.type\_subtype == 0x20"**



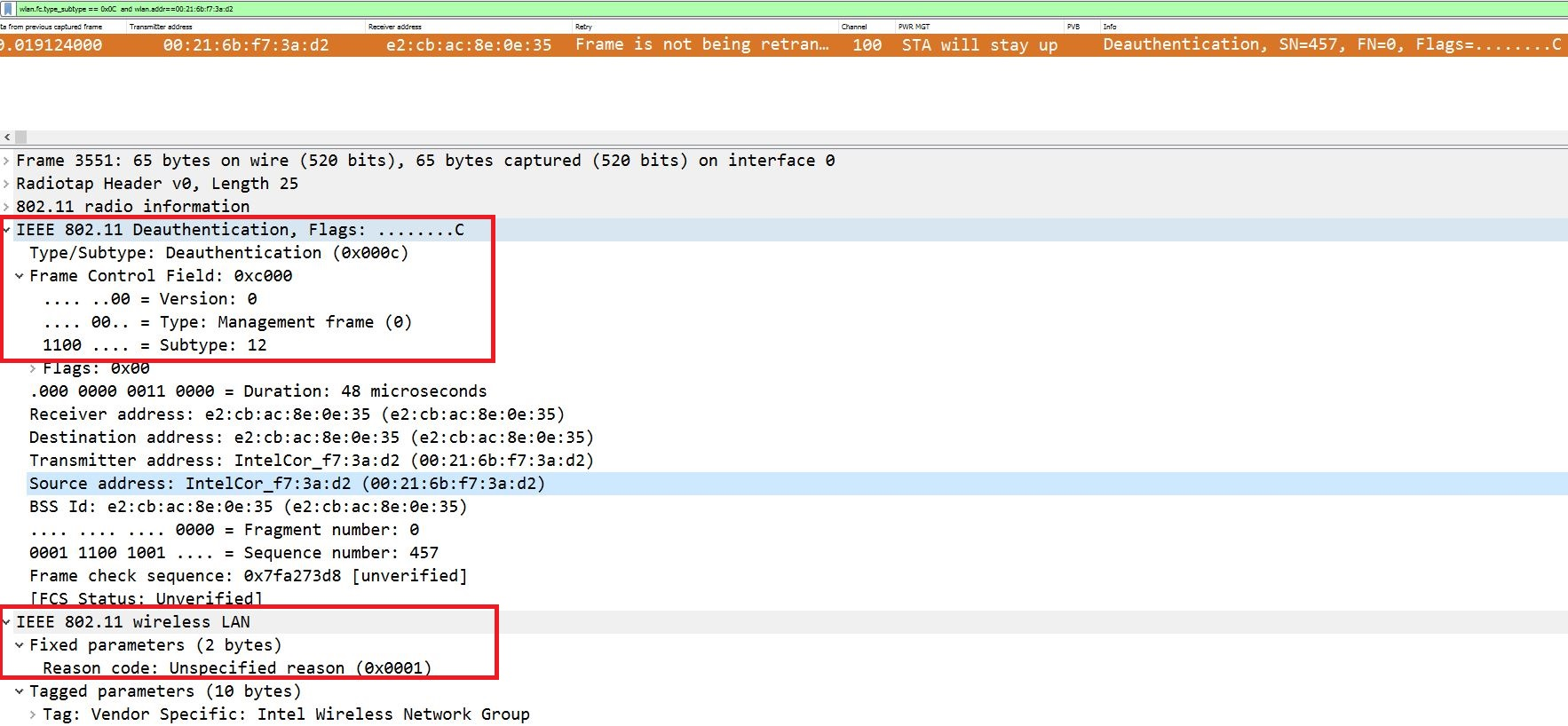
### Common Issues

Use this [link](https://documentation.meraki.com/@api/deki/files/10703/monmode_deauth.pcapng?revision=1) to download an example packet capture file that can be referenced for subtopics like deauthentication, disassociation, and failed WPA Authentication.

##### Deauthentication

This is an announcement by a station that sends a deauthentication frame to another station if it wishes to terminate secure communications. Deauthentication frames can be sent for multiple reasons in order to end a connection. If an AP receives any frame other than an authentication or probe request from a mobile station that is not authenticated it will respond with a deauthentication frame placing the mobile into an unauthenticated an unassociated state. Deauthentication frames are also used by Meraki Access Points as part of the process for containing rogue access points when using the [Air Marshal feature](https://documentation.meraki.com/MR/Monitoring_and_Reporting/Air_Marshal).

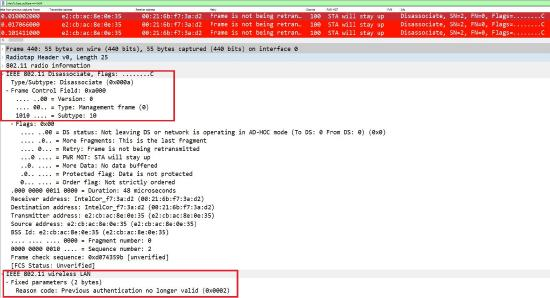
The Wireshark display filter for **deauthentication frames** is **“wlan.fc.type\_subtype == 0x0C”**



##### Disassociation

Dissociation frames are sent to terminate the connection. For example, a wireless client that is shut down normally can send a disassociation frame to alert the access point that the NIC is powering off. The access point can then relinquish memory allocations and remove the radio NIC from the association table. Disassociation is a simple declaration from either an access point or a device.

The Wireshark display filter for **Disassociation packets** is **"wlan.fc.type\_subtype == 0x0a"**



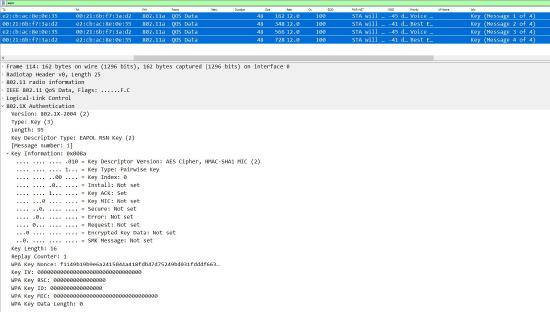
### EAPOL Key Exchange

EAPOL stands for **Extensible Authentication Protocol (EAP)** over LAN. It is described as a **4-way handshake**. The 4-way handshake is used in PSK (WPA-Personal) or 802.1x (WPA2-Enterprise) configured SSIDs. It is a process of exchanging 4 packets between an access point and a wireless client. This process is responsible for generating encryption keys which can be used to encrypt data over the wireless medium. The key exchange process happens after a client is authenticated and associated. After the completion of key exchange, the control frames will take over.

The Wireshark display filter for **4-way handshakes** is **"eapol"**

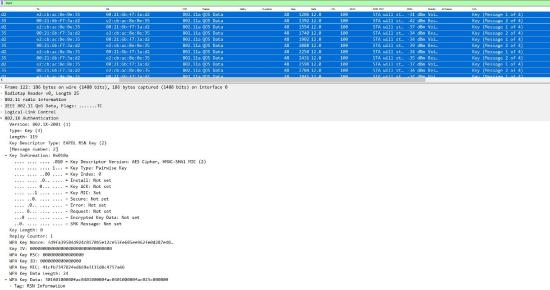
##### Successful WPA Authentication

The WPA handshake consists of the WPA challenge and response as shown in the screenshot below. There are two challenges and responses and each can be matched with the other based on Replay counter field under the 802.1x authentication header. After the WPA handshake is completed and authentication is successful, data begins transferring between the wireless client and the access point.



##### **Failed WPA Authentication**

In this case, the challenge text the client sends back to the AP is incorrect. As a result, the sequence is repeated. Once the handshake process has been attempted and failed four times, the communication is aborted.



### Open Authentication for Troubleshooting

If permitted, it is recommended to troubleshoot WLAN problems in the RF environment using open authentication, in order to reduce potential connection issues. A test SSID for troubleshooting purposes can be created as this approach would surface RF connectivity issues, and those can be corrected before moving to stronger encryption and higher layers of the OSI layer.

### Wireshark Filters

The filtering of wireless packets is different as compared to wired filters on wireshark. Below are some examples of wlan filters.

| **Frame type/subtype** | **Filter syntax** |  |  |
| --- | --- | --- | --- |
| **Probe request** | wlan.fc.type\_subtype == 0x04 | **MAC address** | wlan.addr == MAC\_address |
| **Probe response** | wlan.fc.type\_subtype == 0x05 | **BSSID** | wlan.bssid == AP\_radio\_MAC\_address |
| **Power save poll** | wlan.fc.type\_subtype == 0x1A | **SSID** | wlan\_mgt.ssid == (Name of SSID) |
| **Request to send** | wlan.fc.type\_subtype == 0x1B | **Transmitter Address (TA)** | wlan.ta == MAC\_address |
| **Clear to send** | wlan.fc.type\_subtype == 0x1C | **Receiver Address (RA)** | wlan.ra == MAC\_address |
| **Acknowledgement (ACK) frame** | wlan.fc.type\_subtype == 0x1D | **Source Address (SA)**  **Destination Address (DA)** | wlan.sa == MAC\_address  wlan.da == MAC\_address |
| **Reassociation request** | wlan.fc.type\_subtype == 0x02 | **Specific Channel** | radiotap.channel.freq == frequency |
| **Reassociation response** | wlan.fc.type\_subtype == 0x03 | **Specific Data Rate** | radiotap.datarate == rate\_in\_Mbps |
| **QoS data** | wlan.fc.type\_subtype == 0x28 | **Retry** | wlan.fc.retry==1 |
| **Null QoS data** | wlan.fc.type\_subtype == 0x2C | **NULL Data** | wlan.fc.type\_subtype eq 36 |
| **Contention free period end** | wlan.fc.type\_subtype == 0x1E | **Action frames** | wlan.fc.type\_subtype == 13 |

It is recommended to use or create a wireshark profile specifically for analyzing wireless packet captures. The profile can be added with multiple colorizations of different types of packets as seen in the above screenshots. To save a custom profile, first configure wireless columns and filters to your liking. Then right-click the active profile listing at the bottom right of the screen and click **New**. Name the profile **Wireless** and click **OK**.