

Wireless with Wireshark

Scanning wireless to find devices with Wireshark

Megumi Takeshita, Packet Otaku
Ikeriri network service

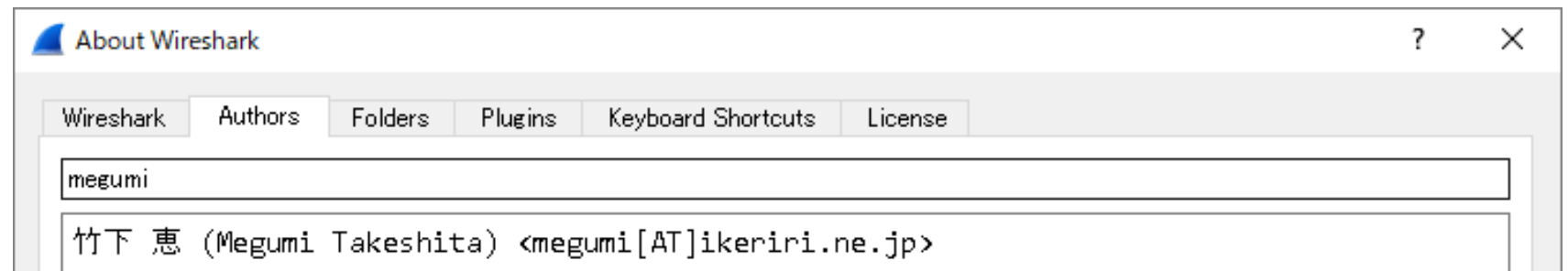
please download the trace files and scripts at
<https://www.ikeriri.ne.jp/sharkfest/ikeriri25eu.zip>



Megumi Takeshita, Packet Otaku, ikeriri network service



- Worked SE/IS at BayNetwork, Nortel
- Reseller of CACE technologies in 2008
- Founder, ikeriri network service co., Ltd
- Reseller of packet capture / wireless-tools
- Wrote 10+ books about Wireshark in Japanese
- Instruct Wireshark to JSDF etc.
- Lecturer of CHUO University
- One of the contributors to Wireshark
- Translate Wireshark into Japanese



Wireshark is a tool for analyzing both wireless and wired networks. In this session, Megumi will show you how to find devices on your wireless network. With various scanning tools, we can explore different standards of WiFi networks and Bluetooth using Wireshark. Additionally, tshark and other Wireshark command-line interface (CLI) tools are helpful in collecting and summarize device information. Use Wireshark to manage your wireless environment and enhance your security!!



Customize Wireshark for Wireless

Set Aliases, Columns, Display Filter buttons, Coloring Rules

- Open trace file wifi7.pcapng
- It is the typical wifi7 connection sequence between Buffalo AP and iphone16 Pro STA
- Let's start Wireless analysis with Wireshark

| No. | Time | Source | Dest | Protocol | Length | Info |
|-----|----------|----------------|--------------|----------|--------|---|
| 1 | 0.000000 | Buffalo_45:1 | Bro_ 002:11 | 802.11 | 508 | Beacon frame, SN=2986, P=0, Flags=....., BI=100, SSID="ikeriri7" |
| 2 | 0.000247 | 22:de:13:07:.. | Bro_ 002:11 | 802.11 | 103 | Probe Request, SN=1704, P=0, Flags=....., SSID="ikeriri7" |
| 3 | 0.000411 | Buffalo_45:1 | Bro_ 002:11 | 802.11 | 687 | Probe Response, SN=1027, P=0, Flags=....., BI=100, SSID="ikeriri7" |
| 4 | 0.136344 | 22:de:13:07:.. | Buf_ 002:11 | 802.11 | 75 | Authentication, SN=1798, P=0, Flags=..... |
| 5 | 0.136368 | | 22:.. 002:11 | 802.11 | 42 | Acknowledgement, Flags=..... |
| 6 | 0.136367 | Buffalo_45:1 | 22:.. 002:11 | 802.11 | 62 | Authentication, SN=19, P=0, Flags=..... |
| 7 | 0.136374 | | Buf_ 002:11 | 802.11 | 42 | Acknowledgement, Flags=..... |
| 8 | 0.138331 | 22:de:13:07:.. | Buf_ 002:11 | 802.11 | 247 | Association Request, SN=1799, P=0, Flags=....., SSID="ikeriri7" |
| 9 | 0.138341 | | 22:.. 002:11 | 802.11 | 42 | Acknowledgement, Flags=..... |
| 10 | 0.163541 | Buffalo_45:1 | 22:.. 002:11 | 802.11 | 320 | Association Response, SN=20, P=0, Flags=..... |
| 11 | 0.163562 | | Buf_ 002:11 | 802.11 | 42 | Acknowledgement, Flags=..... |
| 12 | 0.177528 | Buffalo_45:1 | 22:.. 002:11 | 802.11 | 53 | VHT/HE/EHT/RANBND NDP Announcement, Sounding Dialog Token=11, *lags=..... |
| 13 | 0.177979 | 22:de:13:07:.. | Buf_ 002:11 | 802.11 | 405 | Action No Ack, SN=2, P=0, Flags=..... |
| 14 | 0.177998 | Buffalo_45:1 | 22:.. EAPOL | EAPOL | 187 | Key (Message 1 of 4) |
| 15 | 0.177996 | | Buf_ 002:11 | 802.11 | 42 | Acknowledgement, Flags=..... |
| 16 | 0.179295 | 22:de:13:07:.. | Buf_ EAPOL | EAPOL | 208 | Key (Message 2 of 4) |
| 17 | 0.179502 | | 22:.. 002:11 | 802.11 | 42 | Acknowledgement, Flags=..... |
| 18 | 0.183007 | Buffalo_45:1 | 22:.. EAPOL | EAPOL | 253 | Key (Message 3 of 4) |
| 19 | 0.183014 | | Buf_ 002:11 | 802.11 | 42 | Acknowledgement, Flags=..... |
| 20 | 0.183062 | 22:de:13:07:.. | Buf_ EAPOL | EAPOL | 105 | Key (Message 4 of 4) |
| 21 | 0.183969 | | 22:.. 002:11 | 802.11 | 42 | Acknowledgement, Flags=..... |
| 22 | 0.185016 | 22:de:13:07:.. | Buf_ 002:11 | 802.11 | 85 | Action, SN=1801, P=0, Flags=p..... |
| 23 | 0.193412 | Buffalo_45:1 | Bro_ 002:11 | 802.11 | 108 | Data, SN=4071, P=0, Flags=p....F.. |
| 24 | 0.193421 | Buffalo_45:1 | Bro_ 002:11 | 802.11 | 108 | Data, SN=4072, P=0, Flags=p....F.. |
| 25 | 0.193438 | Buffalo_45:1 | Bro_ 002:11 | 802.11 | 108 | Data, SN=4073, P=0, Flags=p....F.. |
| 26 | 0.204758 | Buffalo_45:1 | Bro_ 002:11 | 802.11 | 509 | Beacon frame, SN=2952, P=0, Flags=....., BI=100, SSID="ikeriri7" |

Frame 23: Packet, 108 bytes on wire (864 bits), 108 bytes captured (864 bits) on interface unknown, Id 0

Radiotap Header v0, Length 12

802.11 radio information

IEEE 802.11 Data, Flags: p....F..

Data (44 bytes)

0000

00 00 30 00 ef 00 04 00 a9 a7 3a 85 7c 24 06 00

0010

00 0c 47 18 00 00 c3 a7 00 00 00 00 47 18 35 00

0020

00 42 24 00 ff ff ff ff ff ff 84 e6 cb 43 1d b6

0030

04 a3 cb 45 1d a0 78 fa 05 02 00 00 00 00 00

0040

4c ee ee 40 b2 b7 65 84 12 b2 d6 2a 35 69 9b df

0050

e7 a5 bb 32 1a ef 62 48 e9 a7 d2 51 be 27 88 cd

0060

d7 d9 f4 b4 55 63 43 c0 af 31 95 85

Add alias name on ethers

In wireless analysis, you need to work with MAC addresses rather than IP addresses.

Adding an alias to the ethers file in your profile is essential.


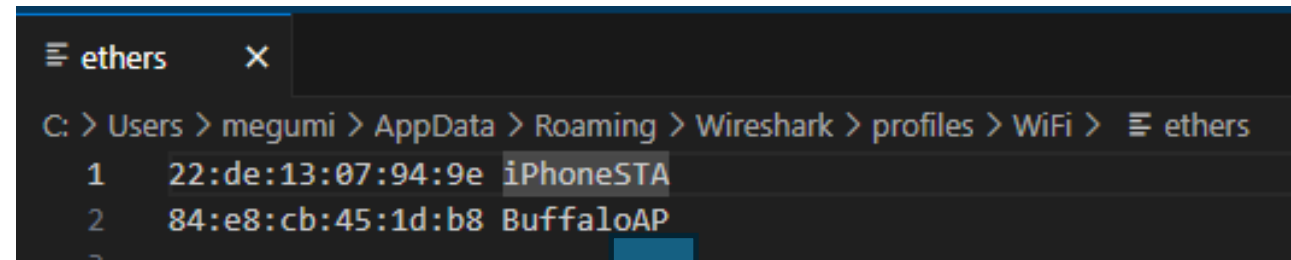
Aliases help you easily identify your known Aps and STAs.

Edit ethers on your Wireshark profile directory

Help>About Wireshark

Select the Folders tab and look for personal configuration

We can use alias names in display filter, Endpoints, Conversations, I/O Graph, Flow diagram and so on



| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|----------|-----------|-------------|----------|--------|------------------------|
| 1 | 0.000000 | BuffaloAP | Broadcast | 802.11 | 509 | Beacon frame, SN=2986, |
| 2 | 0.008247 | iPhoneSTA | Broadcast | 802.11 | 183 | Probe Request, SN=1794 |
| 3 | 0.009414 | BuffaloAP | Broadcast | 802.11 | 487 | Probe Response, SN=298 |
| 4 | 0.136344 | iPhoneSTA | BuffaloAP | 802.11 | 75 | Authentication, SN=179 |
| 5 | 0.136360 | | iPhoneSTA | 802.11 | 42 | Acknowledgement, Flags |
| 6 | 0.136367 | BuffaloAP | iPhoneSTA | 802.11 | 62 | Authentication, SN=19, |
| 7 | 0.136374 | | BuffaloAP | 802.11 | 42 | Acknowledgement, Flags |
| 8 | 0.138331 | iPhoneSTA | BuffaloAP | 802.11 | 247 | Association Request, S |
| 9 | 0.138341 | | iPhoneSTA | 802.11 | 42 | Acknowledgement, Flags |
| 10 | 0.163541 | BuffaloAP | iPhoneSTA | 802.11 | 320 | Association Response, |
| 11 | 0.163562 | | BuffaloAP | 802.11 | 42 | Acknowledgement. Flags |

Add columns on Packet List

| No. | Time | PHY type | CH | Freq. | Signal | MCS | Rate | Retry | Type/Subtype | Source | Destination | SSID | BSSID | STA | Length | Info |
|-----|----------|-----------------|----|----------|----------|-----|------|-------|----------------|----------------|-------------|-----------|-------------------|-----|--------|------------------|
| 1 | 0.000000 | 802.11a (OFD... | 53 | 6215M... | -62 d... | | 6 | | Beacon frame | Buffalo_45:... | Broadcast | "ikeri... | Buffalo_45:1d:... | | 509 | Beacon frame, S |
| 2 | 0.008247 | 802.11a (OFD... | 53 | 6215M... | -58 d... | | 6 | | Probe Reque... | 22:de:13:07... | Broadcast | "ikeri... | Buffalo_45:1d:... | | 183 | Probe Request, S |
| 3 | 0.009414 | 802.11a (OFD... | 53 | 6215M... | -62 d... | | 6 | | Probe Respo... | Buffalo_45:... | Broadcast | "ikeri... | Buffalo_45:1d:... | | 487 | Probe Response, |

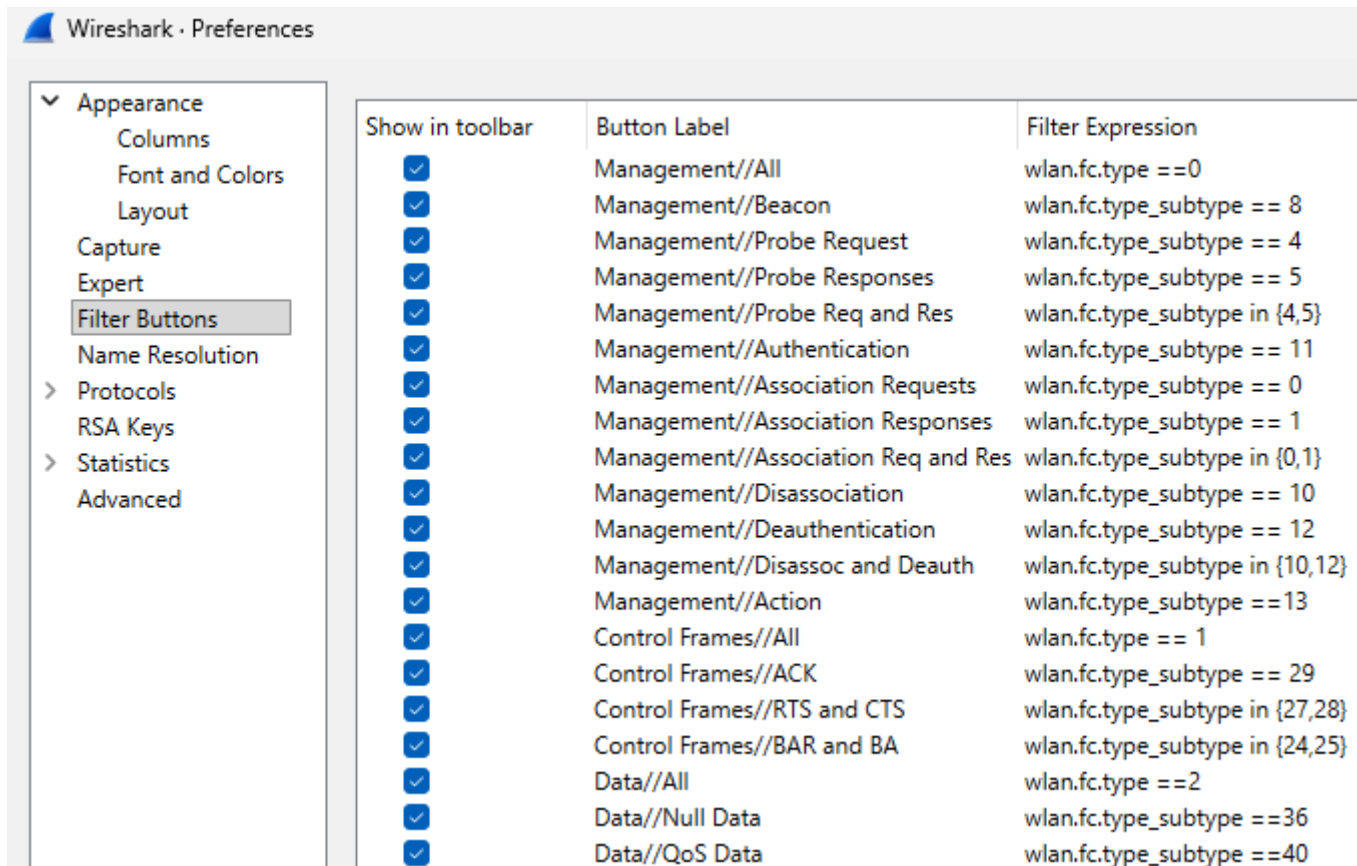
| Displayed | Title | Type | Custom Expression |
|-------------------------------------|--------------|----------------------------|------------------------|
| <input checked="" type="checkbox"/> | No. | Number | |
| <input checked="" type="checkbox"/> | Time | Time (format as specified) | |
| <input checked="" type="checkbox"/> | PHY type | Custom | wlan_radio.phy |
| <input checked="" type="checkbox"/> | CH | Custom | wlan_radio.channel |
| <input checked="" type="checkbox"/> | Freq. | Custom | wlan_radio.frequency |
| <input checked="" type="checkbox"/> | Signal | Custom | radiotap.dbm_antsignal |
| <input checked="" type="checkbox"/> | MCS | Custom | radiotap.mcs.index |
| <input checked="" type="checkbox"/> | Rate | Custom | radiotap.datarate |
| <input checked="" type="checkbox"/> | Retry | Custom | wlan.fc.retry.expert |
| <input checked="" type="checkbox"/> | Type/Subtype | Custom | wlan.fc.type_subtype |
| <input checked="" type="checkbox"/> | Source | Source address | |
| <input checked="" type="checkbox"/> | Destination | Destination address | |
| <input checked="" type="checkbox"/> | Length | Packet length (bytes) | |
| <input checked="" type="checkbox"/> | STA | Custom | wlan.staa |
| <input checked="" type="checkbox"/> | BSSID | Custom | wlan.bssid |
| <input checked="" type="checkbox"/> | SSID | Custom | wlan.ssid |
| <input checked="" type="checkbox"/> | Info | Information | |

- Add columns to your Packet List
- You can drag and drop the field into Column tabs directly, or select Edit > Preferences > Appearance > Columns
- Add columns for WiFi analysis. Recommended fields are PHY type, CH, Freq, Signal, MCS, Rate, Retry, Type/Subtype, STA, BSSID and SSID
- We do not need the Packet length, so check out the Displayed checkbox

IEEE802.11 frame type/subtype

| Type | Type and Subtype | |
|-------------------------------|--|--|
| Management wlan.fc.type==0 | Used by management of WiFi network | Beacon wlan.fc.type_subtype==8 |
| | | Probe Request wlan.fc.type_subtype==4 |
| | | Probe Response wlan.fc.type_subtype==5 |
| | | Authentication wlan.fc.type_subtype==11 |
| | | Deauthentication wlan.fc.type_subtype==12 |
| | | Association Request wlan.fc.type_subtype==0 |
| | | Association Response wlan.fc.type_subtype==1 |
| | | Disassociation wlan.fc.type_subtype==10 |
| | | Action wlan.fc.type_subtype==13 |
| Control wlan.fc.type==1 | Used by control of data communication | RTS (Request To Send) wlan.fc.type_subtype==27 |
| | | CTS (Clear To Send) wlan.fc.type_subtype==28 |
| | | ACK (ACKnowledge) wlan.fc.type_subtype==29 |
| | | BAR (Block ACK Request) wlan.fc.type_subtype==24 |
| | | BA(Block ACK) wlan.fc.type_subtype==25 |
| Data wlan.fc.type==2 | Data(with ACK) wlan.fc.type_subtype==20 | |
| | Null Data(No data) wlan.fc.type_subtype==36 | |
| | QoS Data(with BlockACK) wlan.fc.type_subtype==40 | |

- First, we need to know frame types and subtypes of the IEEE802.11 frame
- Here are the common types and subtypes of WiFi.



- Choose Edit -> Preference, select Appearance -> Filter Buttons.
- Create Parent Item // Children Items and set Filter Expression on each Items
- You can filter multiple WiFi type/subtypes like `wlan.fc.type_subtype in { 4,5 }`

- The filter button helps you to distinguish frames and understand the WiFi sequence easily


Add coloring Rules for WiFi analysis



Wireshark · Coloring Rules Default

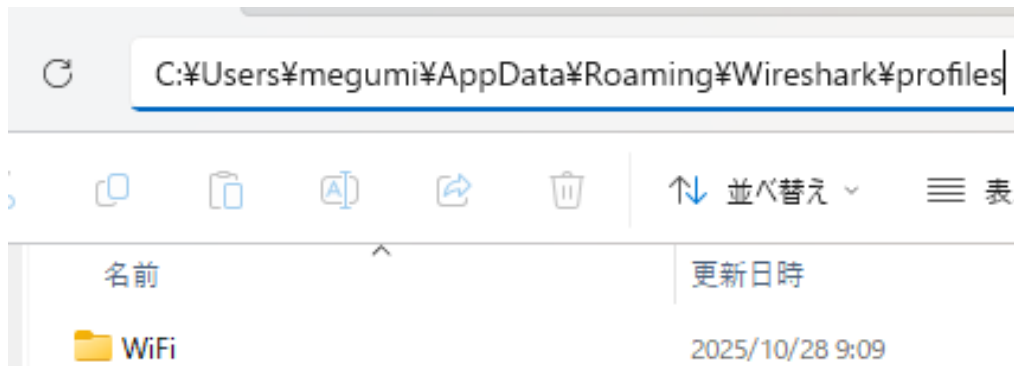
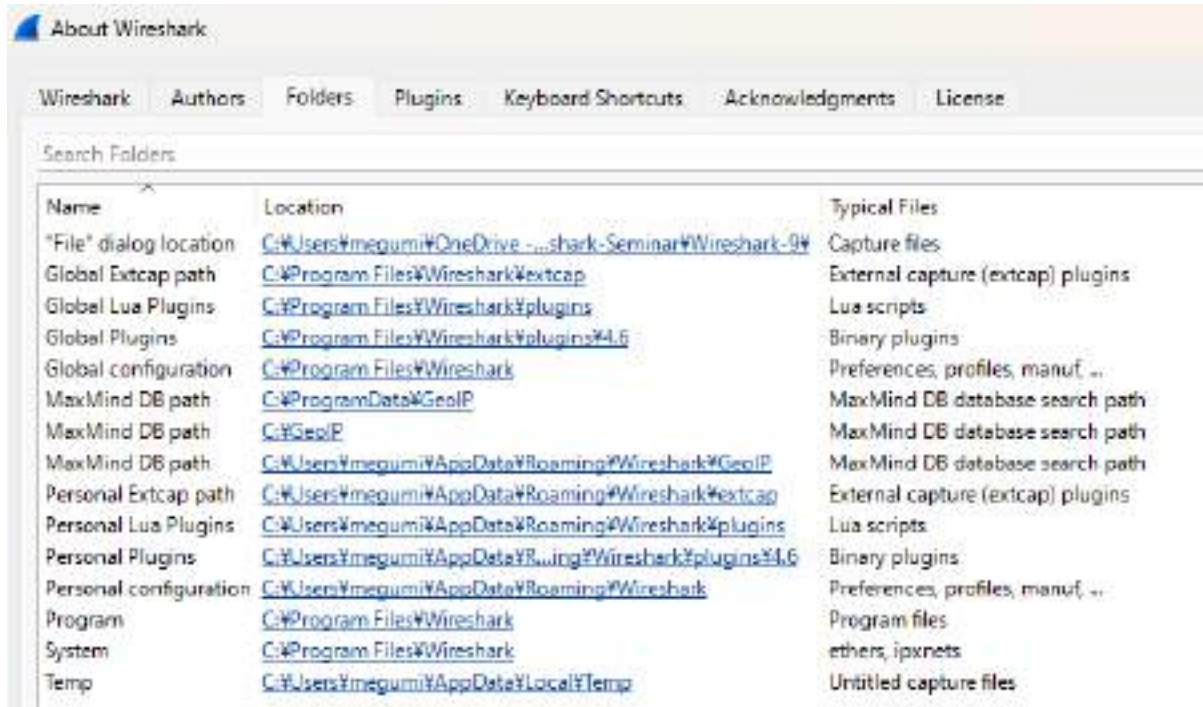
| Name | Filter |
|--|----------------------------|
| <input checked="" type="checkbox"/> Beacon | wlan.fc.type_subtype == 8 |
| <input checked="" type="checkbox"/> Probe Request | wlan.fc.type_subtype == 4 |
| <input checked="" type="checkbox"/> Probe Response | wlan.fc.type_subtype == 5 |
| <input checked="" type="checkbox"/> Authentication | wlan.fc.type_subtype == 11 |
| <input checked="" type="checkbox"/> Deauthentication | wlan.fc.type_subtype == 12 |
| <input checked="" type="checkbox"/> Association Request | wlan.fc.type_subtype == 0 |
| <input checked="" type="checkbox"/> Association Response | wlan.fc.type_subtype == 1 |
| <input checked="" type="checkbox"/> Disassociate | wlan.fc.type_subtype == 10 |
| <input checked="" type="checkbox"/> Action | wlan.fc.type_subtype == 13 |
| <input checked="" type="checkbox"/> ACKnowledgement | wlan.fc.subtype == 29 |
| <input checked="" type="checkbox"/> Block ACK | wlan.fc.subtype == 24 |
| <input checked="" type="checkbox"/> Block ACK Request | wlan.fc.subtype == 24 |
| <input checked="" type="checkbox"/> RTS | wlan.fc.type_subtype == 27 |
| <input checked="" type="checkbox"/> CTS | wlan.fc.type_subtype == 28 |
| <input checked="" type="checkbox"/> Data | wlan.fc.type_subtype == 32 |
| <input checked="" type="checkbox"/> QoS Data | wlan.fc.type_subtype == 40 |
| <input checked="" type="checkbox"/> Null Data | wlan.fc.type_subtype == 36 |
| <input checked="" type="checkbox"/> EAPOL | eapol |

- View>Coloring Rules
- Set different colors by frame types and subtype
Beacon, Probe Request Probe Authentication, Deauthentication, Association Request, Association Response, Disassociate, Action, ACKnowledgement, Block ACK, Block ACK Request, RTS, CTS, EAPOL, Data, QoS Data and Null Data
- It helps to understand the sequence of the WiFi connection easily

[illegible]

- | Time | BuffaloAP | Broadcast | iPhone5TA | Buffalo_45c1db8 | iPhone5TA | Buffalo_45c1da0 | Comment |
|----------|--|-----------|-----------|--|-----------|-----------------|--|
| 0.000000 | Beacon frame, Src=2005, Prio=0, Flags=..., B=130, SSID="iSe" | | | | | | Beacon frame, Src=2005, Prio=0, Flags=..., B=130, SSID="iSe" |
| 0.000247 | Probe Request, Src=1794, Prio=0, Flags=..., SSID="iSe" | | | | | | Probe Request, Src=1794, Prio=0, Flags=..., SSID="iSe" |
| 0.000414 | Probe Response, Src=2005, Prio=0, Flags=..., B=130, SSID="iSe" | | | | | | Probe Response, Src=2005, Prio=0, Flags=..., B=130, SSID="iSe" |
| 0.136344 | Authentication, Src=1794, Prio=0, Flags=... | | | | | | Authentication, Src=1794, Prio=0, Flags=... |
| 0.136367 | Authentication, Src=1794, Prio=0, Flags=... | | | | | | Authentication, Src=1794, Prio=0, Flags=... |
| 0.136331 | Association Request, Src=1794, Prio=0, Flags=..., SSID="iSe" | | | | | | Association Request, Src=1794, Prio=0, Flags=..., SSID="iSe" |
| 0.163541 | Association Response, Src=2005, Prio=0, Flags=... | | | | | | Association Response, Src=2005, Prio=0, Flags=... |
| 0.177520 | | | | WIFIHD,HT,MANAGING NDP Announcement, Sounding... | | | WIFIHD,HT,MANAGING NDP Announcement, Sounding... |
| 0.177578 | Action No Ack, Src=2, Prio=0, Flags=... | | | | | | Action No Ack, Src=2, Prio=0, Flags=... |
| 0.177980 | Key Message 1 of 4 | | | | | | Key Message 1 of 4 |
| 0.178295 | Key Message 2 of 4 | | | | | | Key Message 2 of 4 |
| 0.183007 | Key Message 3 of 4 | | | | | | Key Message 3 of 4 |
| 0.183962 | Key Message 4 of 4 | | | | | | Key Message 4 of 4 |
| 0.185016 | Action, Src=1801, Prio=0, Flags=... | | | | | | Action, Src=1801, Prio=0, Flags=... |
| 0.193412 | | | | Data, Src=4071, Prio=0, Flags=p,p... | | | Data, Src=4071, Prio=0, Flags=p,p... |
| 0.193423 | | | | Data, Src=4072, Prio=0, Flags=p,p... | | | Data, Src=4072, Prio=0, Flags=p,p... |
| 0.193430 | | | | Data, Src=4073, Prio=0, Flags=p,p... | | | Data, Src=4073, Prio=0, Flags=p,p... |
| 0.204750 | Beacon frame, Src=2005, Prio=0, Flags=..., B=130, SSID="iSe" | | | | | | Beacon frame, Src=2005, Prio=0, Flags=..., B=130, SSID="iSe" |

Copy the profile into your environment



- Help -> About Wireshark -> Folders to find your personal configuration location
- Copy WiFi folder (extract from WiFi.zip) into your profile directory (for example, copy into C:\Users\username\AppData\Roaming\Wireshark\profiles if you use Wireshark in Windows)
- Change your Wireshark profile, click the right bottom Profile button and change it from Default to WiFi

Basic Connection sequence

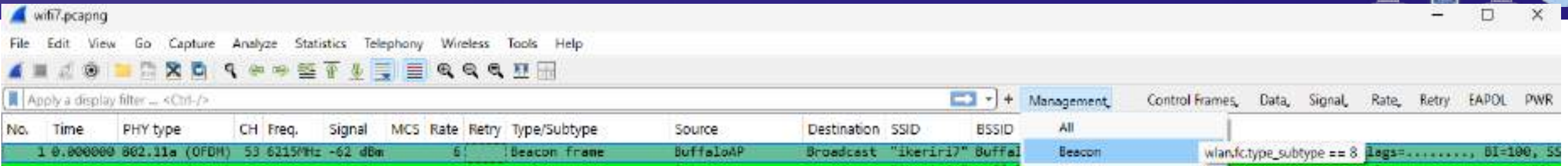
Beacons, Probes, Authentications, Associations and 4-way Handshake

Basic Connection sequence



- APs send Beacons periodically (every 100ms in usual)
- STA sends Probe Requests to determine the best AP, AP sends back Probe Response including AP specifications.
- STA and AP exchange Authentication frames (WPA2: Basic Authentication, WPA3: Simultaneous Authentication of Equals)
- STA sends Association Request and AP sends back Association Response to negotiate exchange Association frames, including the actual connection settings between STA and AP
- STA and AP start EAPOL 4-way handshake to create PTK for unicast, AP sends GTK for multicast, broadcast.

Beacon frame



- Choose Display Filter button, Management -> Beacon and click one of Beacon frames.
- Beacon frame is broadcast by AP periodically, including Wireless specs.
- Open IEEE802.11 Wireless Management -> Tagged parameters.
- You can find SSID in Tag: SSID parameter set if AP is not set in stealth mode.
- There are many Tags, including AP's specs, such as HT(.11n WiFi4), VHT(.11ac WiFi5), HE(.11ax WiFi6) and EHT(.11be WiFi7)

| Generation | Name | band | Wireless LAN Management fields |
|---------------------|------|------------|----------------------------------|
| 4 (IEEE802.11n) | HT | 2.4/5GHz | HT Capabilities, HT Operations |
| 5 (IEEE802.11ac) | VHT | 5GHz | VHT Capabilities, VHT Operations |
| 6/6E (IEEE802.11ax) | HE | 2.4/5GHz | HE Capabilities, HE Operations |
| 7 (IEEE802.11be) | EHT | 2.4/5/6GHz | HT Capabilities, HT Operations |

- We can find supported generations by Wireless LAN Management fields.

▼ Tagged parameters (441 bytes)

▼ Tag: SSID parameter set: "ikeriri7"

Tag Number: SSID parameter set (0)

Tag length: 8

SSID: "ikeriri7"

- > Tag: Supported Rates 6(B), 9, 12(B), 18, 24(B), 36, 48, 54, [Mbit/sec]
- > Tag: Traffic Indication Map (TIM): DTIM 0 of 1 bitmap
- > Tag: Country Information: Country Code JP, Environment Global operating classes
- > Tag: Power Constraint: 0
- > Tag: TPC Report Transmit Power: 22 dBm
- > Tag: Tx Power Envelope
- > Tag: Tx Power Envelope
- > Tag: RM Enabled Capabilities (5 octets)
- > Tag: AP Channel Report: Operating Class 131, Channel List : 1, 5, 9, 13, 17, 21,
- > Tag: RSN Information
- > Tag: QBSS Load Element 802.11e CCA Version
- > Tag: Interworking
- > Tag: Advertisement Protocol

> Tag: Extended Capabilities (11 octets)

> Tag: Vendor Specific: Microsoft Corp.: WMM/WME: Parameter Element

> Ext Tag: HE Capabilities

> Ext Tag: HE Operation

> Ext Tag: Spatial Reuse Parameter Set

> Ext Tag: MU EDCA Parameter Set

> Ext Tag: HE 6 GHz Band Capabilities

> Tag: Vendor Specific: MediaTek Inc

> Tag: Vendor Specific: MediaTek Inc

> Tag: Vendor Specific: Wi-Fi Alliance: Multi Band Operation - Optimized Connectiv

> Tag: Vendor Specific: MediaTek Inc

> Tag: FILS Indication

> Tag: RSN eXtension (1 octet)

> Tag: Vendor Specific: MediaTek Inc

> Ext Tag: Multi-Link (802.11be D3.0)

> Ext Tag: EHT Capabilities (802.11be D3.0)

> Ext Tag: EHT Operation (802.11be D3.0)

> Tag: Vendor Specific: Buffalo.Inc

- Andy-san (Andrew Walding, famous for Wireshark Profiles Repository, founder of CELLSTREAM) submits Wireshark Discord

| Standards | Display Filter |
|-------------------|--|
| Pre-WiFi 4 Legacy | wlan.fc.type_subtype == 0x0008 and wlan.tag.number == 1 and not ((wlan.tag.number == 61) or (wlan.tag.number == 45) or (wlan.tag.number == 191) or (wlan.ext_tag.number == 192) or (wlan.ext_tag.number == 255) or (wlan.ext_tag.number == 108)) |
| WiFi4 11n | wlan.fc.type_subtype == 0x0008 and ((wlan.tag.number == 61) or (wlan.tag.number == 45)) and not ((wlan.tag.number == 191) or (wlan.tag.number == 192) or (wlan.ext_tag.number == 255) or (wlan.ext_tag.number == 108)) |
| WiFi5 11ac | wlan.fc.type_subtype == 0x0008 and ((wlan.tag.number == 191 or (wlan.tag.number == 192)) and not ((wlan.ext_tag.number == 255 or wlan.ext_tag.number == 108))) |
| WiFi6 11ax | wlan.fc.type_subtype == 0x0008 and wlan.tag.number == 255 and not wlan.ext_tag.number == 108 |
| WiFi7 11be | wlan.fc.type_subtype == 0x0008 and wlan.ext_tag.number == 108 |

We can find WiFi generations from tags on Beacon from AP

Probe Request and Probe Response

| wlan.fc.type_subtype in {4,5} | | | | | | | | | | | | | | Management | Control Frame |
|-------------------------------|----------|----------------|----|---------|---------|-----|------|-------|----------------|-----------|-------------|------------|--------|-------------------|---------------|
| No. | Time | PHY type | CH | Freq. | Signal | MCS | Rate | Retry | Type/Subtype | Source | Destination | SSID | BSSID | All | |
| 2 | 0.008247 | 802.11a (OFDM) | 53 | 6215MHz | -58 dBm | | 6 | | Probe Request | iPhoneSTA | Broadcast | "ikeriri7" | Buffa1 | Beacon | |
| 3 | 0.009414 | 802.11a (OFDM) | 53 | 6215MHz | -62 dBm | | 6 | | Probe Response | BuffaloAP | Broadcast | "ikeriri7" | Buffa1 | Probe Request | |
| | | | | | | | | | | | | | | Probe Responses | |
| | | | | | | | | | | | | | | Probe Req and Res | |

IEEE 802.11 Wireless Management

> Fixed parameters (12 bytes)

▼ Tagged parameters (419 bytes)

- > Tag: SSID parameter set: "ikeriri7"
- > Tag: Supported Rates 6(B), 9, 12(B), 18, 24(B), 36, 48, 54, [Mbit/s]
- > Tag: RSN Information
- > Tag: Extended Capabilities (11 octets)
- > Tag: QBSS Load Element 802.11e CCA Version
- > Tag: Interworking
- > Tag: Advertisement Protocol
- > Tag: Vendor Specific: Microsoft Corp.: WMM/WME: Parameter Element
- > Tag: RM Enabled Capabilities (5 octets)
- > Tag: AP Channel Report: Operating Class 131, Channel List : 1, 5,
- > Tag: Power Constraint: 0
- > Tag: TPC Report Transmit Power: 22 dBm
- > Tag: Tx Power Envelope
- > Tag: Country Information: Country Code JP, Environment Global oper
- > Ext Tag: HE Capabilities
- > Ext Tag: HE Operation
- > Ext Tag: Spatial Reuse Parameter Set
- > Ext Tag: MU EDCA Parameter Set
- > Ext Tag: HE 6 GHz Band Capabilities
- > Tag: Vendor Specific: MediaTek Inc
- > Tag: Vendor Specific: MediaTek Inc
- > Tag: Vendor Specific: MediaTek Inc
- > Tag: Vendor Specific: Wi-Fi Alliance: Multi Band Operation - Optim
- > Tag: FILS Indication
- > Tag: RSN eXtension (1 octet)
- > Tag: Vendor Specific: MediaTek Inc
- > Ext Tag: Multi-Link (802.11be D3.0)
- > Ext Tag: EHT Capabilities (802.11be D3.0)
- > Ext Tag: EHT Operation (802.11be D3.0)
- > Tag: Vendor Specific: Buffalo.Inc

- Choose the Display Filter button, Management -> Probe Req and Res
- Ctrl + [Up], Ctrl + [Down] to compare IEEE802.11 Wireless Management Header
- STA sends SSID, HE, EHT and Vendor information by Probe Request, and AP sends back Probe Response with SSID, QBSS, HE, Multi-Link (Aggregating 2.4/5/6 GHz connections) and Vendor information

QBSS Load Element 802.11e CCA

▼ Tag: QBSS Load Element 802.11e CCA Version

Tag Number: QBSS Load Element (11)

Tag length: 5

QBSS Version: 2

Station Count: 0

Channel Utilization: 6 (2%)

Available Admission Capacity: 31250 (1000000 us/s)

| Field name | Contents |
|------------------------------|--|
| Station Count | Number of Stations associated with the BSS |
| Channel Utilization | Percentage of the channel was sensed as busy |
| Available Admission Capacity | Remaining capacity available for new QoS |

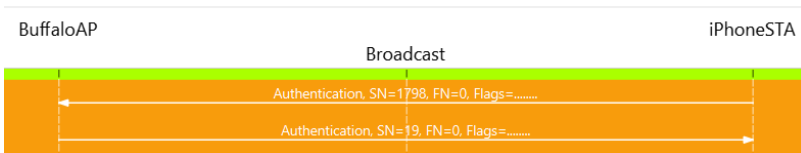
- QBSS(QoS Basic Service Set) Load Element is defined by IEEE802.11e
- CCA (Clear Channel Assessment) is the PHY-layer mechanism used to determine whether WiFi is busy.
- QBSS Load Element 802.11 CCA Version contains network load information: Station Count, Channel Utilization and Available Admission Capacity in Probe Response by AP
- STA determine which AP is the best from the QBSS Load Element
- Some Beacons may contain QBSS Load Element 802.11e CCA

Authentication (Open Systems/SAE)

| wlan.fc.type_subtype == 11 | | | | | | | | | | | | | | Management | | Control Frames | |
|--|----------|----------------|----|---------|---------|-----|------|-------|----------------|-----------|-------------------------|------|--------|-------------------|--|----------------|--|
| No. | Time | PHY type | CH | Freq. | Signal | MCS | Rate | Retry | Type/Subtype | Source | Destination | SSID | BSSID | All | | | |
| 4 | 0.136344 | 802.11a (OFDM) | 53 | 6215MHz | -59 dBm | | | 6 | Authentication | iPhoneSTA | BuffaloAP | | Buffal | Beacon | | on, | |
| 6 | 0.136367 | 802.11a (OFDM) | 53 | 6215MHz | -61 dBm | | | 6 | Authentication | BuffaloAP | iPhoneSTA | | Buffal | Probe Request | | on, | |
| | | | | | | | | | | | | | | Probe Responses | | | |
| | | | | | | | | | | | | | | Probe Req and Res | | | |
| | | | | | | | | | | | | | | Authentication | | | |
| > Frame 4: Packet, 75 bytes on wire (600 bits), 75 bytes captured (600 bits) on interface unknown, | | | | | | | | | | 0000 | 00 00 20 00 6f 00 04 00 | | | | | | |
| > Radiotap Header v0, Length 32 | | | | | | | | | | 0010 | 00 0c 47 18 00 00 c5 aa | | | | | | |
| > 802.11 radio information | | | | | | | | | | 0020 | b0 00 3c 00 84 e8 cb 45 | | | | | | |

IEEE 802.11 Wireless Management

- Fixed parameters (6 bytes)
 - Authentication Algorithm: Open System (0)
 - Authentication SEQ: 0x0001
 - Status code: Successful (0x0000)
- Tagged parameters (13 bytes)
 - Tag: Vendor Specific: Apple, Inc. (Data: 0)
 - Tag Number: Vendor Specific (221)
 - Tag length: 11
 - OUI: 00:17:f2 (Apple, Inc.)
 - Vendor Specific OUI Type: 10
 - Type: 10
 - Data: 00010400000000



- Choose the Display Filter button, Management -> Authentication
- Ctrl + [Up], Ctrl + [Down] to compare IEEE802.11 Wireless Management Header
- This datalink uses Open System, STA and AP exchange Authentication frame, confirms SSID name for authentication.
- If you find Open System, this connection uses WEP/TKIP/WPA2, we may decrypt if you captured the complete 4-way handshake and we knew PSK.
- If you find SAE, it uses WPA3

Association Request and Association Response



| wlan.fc.type_subtype in {0,1} | | | | | | | | | | | | | | Management | Control Frame |
|-------------------------------|----------|----------------|----|---------|---------|-----|------|-------|----------------------|-----------|-------------|------------|--------|---------------|---------------|
| No. | Time | PHY type | CH | Freq. | Signal | MCS | Rate | Retry | Type/Subtype | Source | Destination | SSID | BSSID | All | |
| 8 | 0.138331 | 802.11a (OFDM) | 53 | 6215MHz | -58 dBm | | | 6 | Association Request | iPhoneSTA | BuffaloAP | "ikeriri7" | Buffal | Beacon | |
| 10 | 0.163541 | 802.11a (OFDM) | 53 | 6215MHz | -61 dBm | | | 6 | Association Response | BuffaloAP | iPhoneSTA | | Buffal | Probe Request | |

| | | |
|--|------|-------------------------|
| > Frame 8: Packet, 247 bytes on wire (1976 bits), 247 bytes captured (1976 bits) on interface ui | 0000 | 00 00 20 00 6f 00 04 00 |
| > Radiotap Header v0, Length 32 | 0010 | 00 0c 47 18 00 00 c6 aa |
| > 802.11 radio information | 0020 | 00 00 3c 00 84 e8 cb 45 |
| > IEEE 802.11 Association Request, Flags: | 0030 | 84 e8 cb 45 1d b8 70 70 |
| > IEEE 802.11 Wireless Management | 0040 | 65 72 69 72 69 37 01 08 |
| > Fixed parameters (4 bytes) | 0050 | 21 02 f2 13 24 02 01 18 |
| | 0060 | 01 00 00 0f ac 04 01 00 |
| | 0070 | 00 11 22 33 44 55 66 77 |

| IEEE 802.11 Wireless Management | |
|--|--|
| > Fixed parameters (6 bytes) | |
| > Tagged parameters (258 bytes) | |
| > Tag: Supported Rates 6(B), 9, 12(B), 18, 24(B), 36, 48, 54 | |
| > Tag: Vendor Specific: Wi-Fi Alliance: IEEE1905 Multi-AP | |
| > Tag: RM Enabled Capabilities (5 octets) | |
| > Tag: Vendor Specific: Microsoft Corp.: WMM/WME: Parameter | |
| > Tag: BSS Max Idle Period | |
| > Ext Tag: HE Capabilities | |
| > Ext Tag: HE Operation | |
| > Ext Tag: Spatial Reuse Parameter Set | |
| > Ext Tag: MU EDCA Parameter Set | |
| > Ext Tag: HE 6 GHz Band Capabilities | |
| > Tag: Extended Capabilities (11 octets) | |
| > Tag: Vendor Specific: MediaTek Inc | |
| > Tag: Vendor Specific: MediaTek Inc | |
| > Tag: Vendor Specific: Wi-Fi Alliance: Multi Band Operation | |
| > Tag: FILS Indication | |
| > Tag: RSN eXtension (1 octet) | |
| > Tag: Vendor Specific: MediaTek Inc | |
| > Ext Tag: EHT Capabilities (802.11be D3.0) | |
| > Ext Tag: EHT Operation (802.11be D3.0) | |
| > Tag: Vendor Specific: Buffalo.Inc | |

- Choose the Display Filter button, Management -> Association Req and Res
- Ctrl + [Up], Ctrl + [Down] to compare IEEE802.11 Wireless Management Header
- WiFi datalink is created between STA and AP, exchanging actual connect specs.
- This connection uses WiFi7 and MLO (we can find EHT, IEEE1905 Multi-AP, Multi Band Operation tags).

EAPOL 4 way handshake

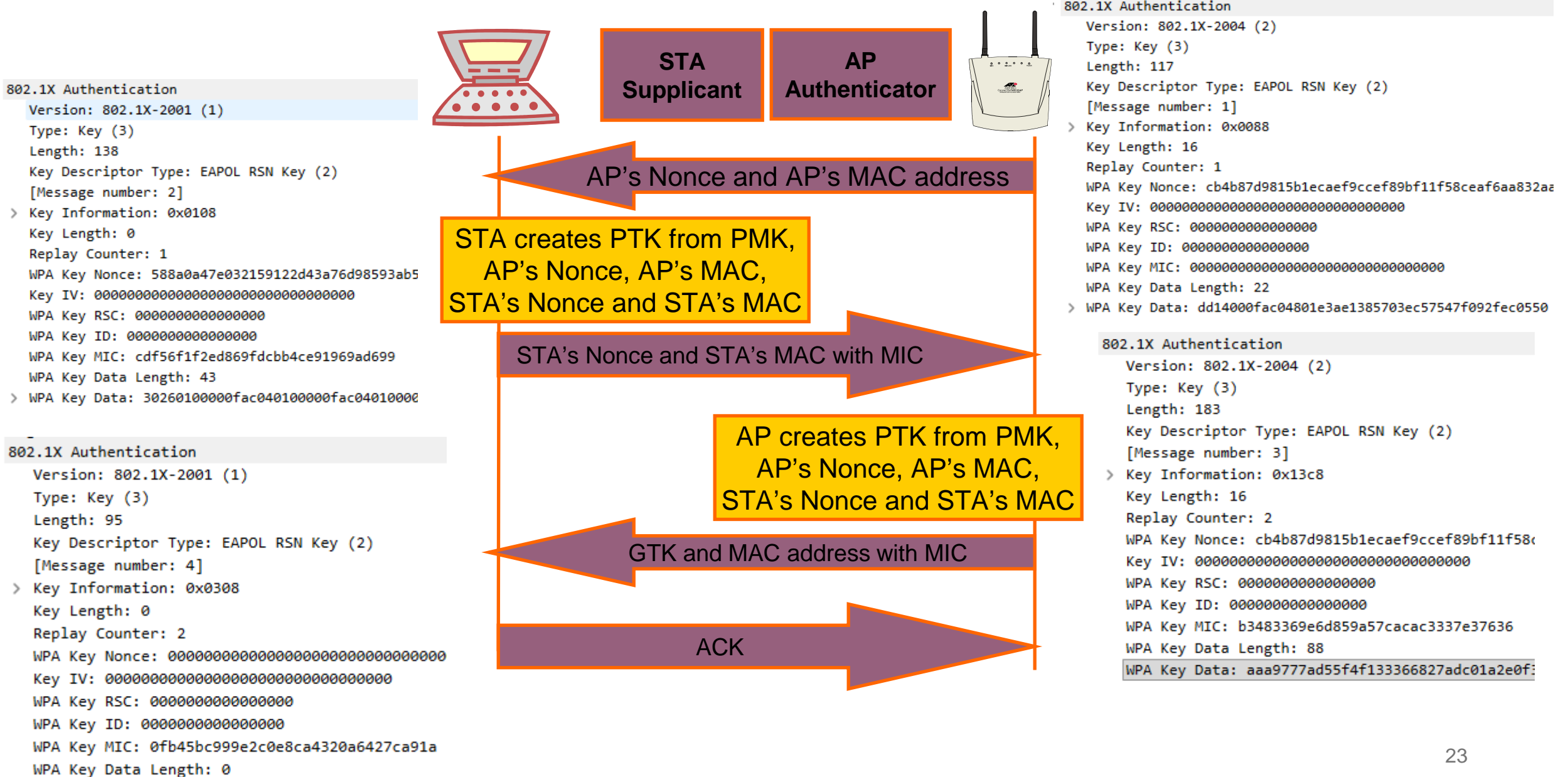
| eapol | | | | | | | | | | | | | | | | | |
|-------|----------|----------------|----|---------|---------|-----|------|-------|--------------|-----------|-------------|------|-----------|-----------|----------------------|--|--|
| No. | Time | PHY type | CH | Freq. | Signal | MCS | Rate | Retry | Type/Subtype | Source | Destination | SSID | BSSID | STA | Info | | |
| 14 | 0.177990 | 802.11a (OFDM) | 53 | 6215MHz | -60 dBm | | | 6 | QoS Data | BuffaloAP | iPhoneSTA | | BuffaloAP | iPhoneSTA | Key (Message 1 of 4) | | |
| 16 | 0.179295 | 802.11a (OFDM) | 53 | 6215MHz | -58 dBm | | | 6 | QoS Data | iPhoneSTA | BuffaloAP | | BuffaloAP | iPhoneSTA | Key (Message 2 of 4) | | |
| 18 | 0.183007 | 802.11a (OFDM) | 53 | 6215MHz | -60 dBm | | | 6 | QoS Data | BuffaloAP | iPhoneSTA | | BuffaloAP | iPhoneSTA | Key (Message 3 of 4) | | |
| 20 | 0.183962 | 802.11a (OFDM) | 53 | 6215MHz | -59 dBm | | | 6 | QoS Data | iPhoneSTA | BuffaloAP | | BuffaloAP | iPhoneSTA | Key (Message 4 of 4) | | |

```
> Radiotap Header v0, Length 32
> 802.11 radio information
> IEEE 802.11 QoS Data, Flags: .....F.
> Logical-Link Control
> 802.1X Authentication
  Version: 802.1X-2004 (2)
  Type: Key (3)
  Length: 117
  Key Descriptor Type: EAPOL RSN Key (2)
  [Message number: 1]
> Key Information: 0x0088
  Key Length: 16
  Replay Counter: 1
  WPA Key Nonce: cb4b87d9815b1ecaef9ccef89bf11f58ceaf6aa832aad66fef9974f3df6f61c8
  Key IV: 00000000000000000000000000000000
  WPA Key RSC: 0000000000000000
  WPA Key ID: 0000000000000000
  WPA Key MIC: 00000000000000000000000000000000
  WPA Key Data Length: 22
> WPA Key Data: dd1400fac04801e3ae1385703ec57547f092fec0550
```

- Choose the Display Filter button, EAPOL, to filter 4-way handshake
- There are 4 packets between AP and STA to create a PTK (Pairwise Transient Key) keyring, including TK(Temporal Key) for the session, AP sends GTK (Group Temporal Key) for broadcast and multicast communication

- Message 1 from AP contains source MAC and AP's Nonce
- Message 2 from STA contains source MAC and STA's Nonce
- Message 3 and 4 are encrypted (Message3 contains GTK)

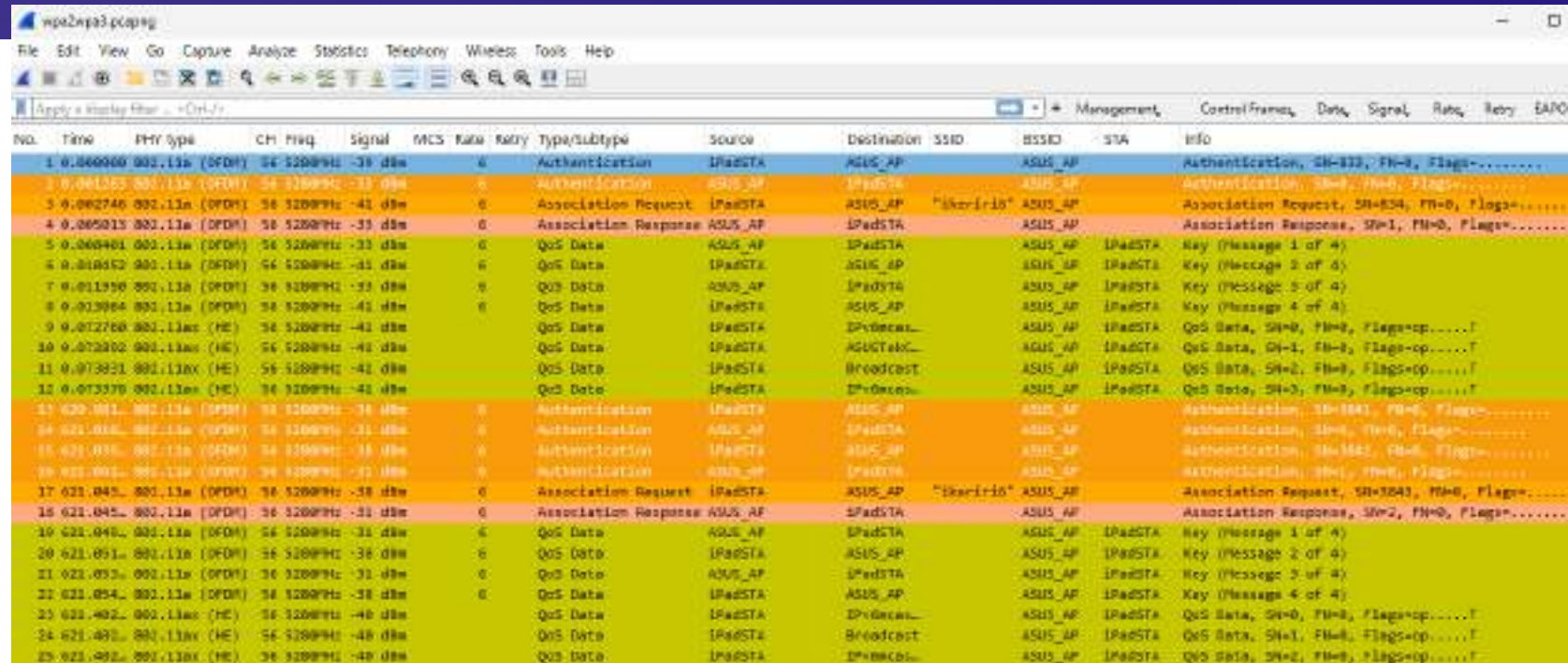
#14 #16 #18 #20 4 way handshake of EAPOL



Decrypting WPA2 PSK

Use Pre Shared Key : SSID to decrypt WPA2-PSK

Wireless with Wireshark



The image shows a Wireshark packet capture of a file named 'wpa2wpa3.pcapng'. The interface displays a list of 27 packets. The first 13 packets (1-13) represent a WPA2 PSK connection, and the next 14 packets (14-27) represent a WPA3 PSK connection. Both connections are from the same STA (e2:da:1e:a8:92:8f) to the same AP (f0:2f:74:c4:f5:c4) using the SSID 'ikeriri6'.

| No. | Time | PHY type | CH | Freq | Signal | MCS | Rate | Retry | Type/subtype | Source | Destination | SSID | BSSID | STA | Info |
|-----|----------|----------------|----|---------|---------|-----|------|-------|----------------------|---------|-------------|------------|---------|---------|--|
| 1 | 0.044000 | 802.11a (OFDM) | 54 | 5200MHz | -38 dBm | 6 | | | Authentication | IPadSTA | ASUS_AP | | ASUS_AP | | Authentication, SN=833, FN=0, Flags=..... |
| 2 | 0.045152 | 802.11a (OFDM) | 54 | 5200MHz | -33 dBm | 6 | | | Authentication | ASUS_AP | IPadSTA | | ASUS_AP | | Authentication, SN=834, FN=0, Flags=..... |
| 3 | 0.062746 | 802.11a (OFDM) | 54 | 5200MHz | -41 dBm | 6 | | | Association Request | IPadSTA | ASUS_AP | "ikeriri6" | ASUS_AP | | Association Request, SN=834, FN=0, Flags=..... |
| 4 | 0.065013 | 802.11a (OFDM) | 54 | 5200MHz | -33 dBm | 6 | | | Association Response | ASUS_AP | IPadSTA | | ASUS_AP | | Association Response, SN=1, FN=0, Flags=..... |
| 5 | 0.066401 | 802.11a (OFDM) | 54 | 5200MHz | -33 dBm | 6 | | | QoS Data | ASUS_AP | IPadSTA | | ASUS_AP | IPadSTA | Key (Message 1 of 4) |
| 6 | 0.068162 | 802.11a (OFDM) | 54 | 5200MHz | -41 dBm | 6 | | | QoS Data | IPadSTA | ASUS_AP | | ASUS_AP | IPadSTA | Key (Message 2 of 4) |
| 7 | 0.069390 | 802.11a (OFDM) | 54 | 5200MHz | -33 dBm | 6 | | | QoS Data | ASUS_AP | IPadSTA | | ASUS_AP | IPadSTA | Key (Message 3 of 4) |
| 8 | 0.070904 | 802.11a (OFDM) | 54 | 5200MHz | -41 dBm | 6 | | | QoS Data | IPadSTA | ASUS_AP | | ASUS_AP | IPadSTA | Key (Message 4 of 4) |
| 9 | 0.072768 | 802.11a (HE) | 54 | 5200MHz | -41 dBm | | | | QoS Data | IPadSTA | IPadSTA | | ASUS_AP | IPadSTA | QoS Data, SN=0, FN=0, Flags=op..... |
| 10 | 0.073202 | 802.11a (HE) | 54 | 5200MHz | -41 dBm | | | | QoS Data | IPadSTA | ASUS_AP | | ASUS_AP | IPadSTA | QoS Data, SN=1, FN=0, Flags=op..... |
| 11 | 0.073931 | 802.11a (HE) | 54 | 5200MHz | -41 dBm | | | | QoS Data | IPadSTA | Broadcast | | ASUS_AP | IPadSTA | QoS Data, SN=2, FN=0, Flags=op..... |
| 12 | 0.073370 | 802.11a (HE) | 54 | 5200MHz | -41 dBm | | | | QoS Data | IPadSTA | Broadcast | | ASUS_AP | IPadSTA | QoS Data, SN=3, FN=0, Flags=op..... |
| 13 | 0.201041 | 802.11a (OFDM) | 54 | 5200MHz | -38 dBm | 6 | | | Authentication | IPadSTA | ASUS_AP | | ASUS_AP | | Authentication, SN=841, FN=0, Flags=..... |
| 14 | 0.211886 | 802.11a (OFDM) | 54 | 5200MHz | -31 dBm | 6 | | | Authentication | ASUS_AP | IPadSTA | | ASUS_AP | | Authentication, SN=842, FN=0, Flags=..... |
| 15 | 0.211893 | 802.11a (OFDM) | 54 | 5200MHz | -31 dBm | 6 | | | Authentication | IPadSTA | ASUS_AP | | ASUS_AP | | Authentication, SN=843, FN=0, Flags=..... |
| 16 | 0.211893 | 802.11a (OFDM) | 54 | 5200MHz | -31 dBm | 6 | | | Authentication | ASUS_AP | IPadSTA | | ASUS_AP | | Authentication, SN=844, FN=0, Flags=..... |
| 17 | 0.211845 | 802.11a (OFDM) | 54 | 5200MHz | -38 dBm | 6 | | | Association Request | IPadSTA | ASUS_AP | "ikeriri6" | ASUS_AP | | Association Request, SN=844, FN=0, Flags=..... |
| 18 | 0.211845 | 802.11a (OFDM) | 54 | 5200MHz | -31 dBm | 6 | | | Association Response | ASUS_AP | IPadSTA | | ASUS_AP | | Association Response, SN=2, FN=0, Flags=..... |
| 19 | 0.211845 | 802.11a (OFDM) | 54 | 5200MHz | -31 dBm | 6 | | | QoS Data | ASUS_AP | IPadSTA | | ASUS_AP | IPadSTA | Key (Message 1 of 4) |
| 20 | 0.211891 | 802.11a (OFDM) | 54 | 5200MHz | -38 dBm | 6 | | | QoS Data | IPadSTA | ASUS_AP | | ASUS_AP | IPadSTA | Key (Message 2 of 4) |
| 21 | 0.211893 | 802.11a (OFDM) | 54 | 5200MHz | -31 dBm | 6 | | | QoS Data | ASUS_AP | IPadSTA | | ASUS_AP | IPadSTA | Key (Message 3 of 4) |
| 22 | 0.211894 | 802.11a (OFDM) | 54 | 5200MHz | -38 dBm | 6 | | | QoS Data | IPadSTA | ASUS_AP | | ASUS_AP | IPadSTA | Key (Message 4 of 4) |
| 23 | 0.211892 | 802.11a (HE) | 54 | 5200MHz | -40 dBm | | | | QoS Data | IPadSTA | IPadSTA | | ASUS_AP | IPadSTA | QoS Data, SN=0, FN=0, Flags=op..... |
| 24 | 0.211893 | 802.11a (HE) | 54 | 5200MHz | -40 dBm | | | | QoS Data | IPadSTA | Broadcast | | ASUS_AP | IPadSTA | QoS Data, SN=1, FN=0, Flags=op..... |
| 25 | 0.211892 | 802.11a (HE) | 54 | 5200MHz | -40 dBm | | | | QoS Data | IPadSTA | Broadcast | | ASUS_AP | IPadSTA | QoS Data, SN=2, FN=0, Flags=op..... |

- Open the trace wpa2wpa3.pcapng
- There are two WiFi connections by same STA (e2:da:1e:a8:92:8f) as iPadSTA and same AP (f0:2f:74:c4:f5:c4) as ASUS_AP
- Both connection use same SSID: ikeriri6 and pass phrase: wireshark
- The first connection from #1 to #13 is WPA2 PSK
- The next connection from #14 to #27 is WPA3 PSK

- Compare Open System of WPA2 and SAE (Simultaneous Authentication of Equals)
- Use Display Filter buttons to select Management -> Authentication, Ctrl + [Up], Ctrl + [Down] to compare 802.11 Wireless Management Header
- Open System confirms SSID, exchange Status Code: Successful
- Each AP and STA sends Scalar and Finite Field Element value, SAE Message Type: Commit
- Then AP and STA exchange Confirm Value, SAE Message Type: Confirm

IEEE 802.11 Wireless Management

▼ Fixed parameters (6 bytes)

Authentication Algorithm: Open System (0)

Authentication SEQ: 0x0001

Status code: Successful (0x0000)

> Tagged parameters (35 bytes)

IEEE 802.11 Wireless Management

▼ Fixed parameters (104 bytes)

Authentication Algorithm: Simultaneous Authentication of Equals (SAE) (3)

Authentication SEQ: 0x0001

Status code: Successful (0x0000)

SAE Message Type: Commit (1)

Group Id: 256-bit random ECP group (19)

Scalar: c67801ac5941d1e0fad412b255567e53c885a0d12a22439a3e021c7d633f37e7

Finite Field Element: f4b7c34e9f0d5444381e1dde353e54dcc838435b372a3933b7cc

IEEE 802.11 Wireless Management

▼ Fixed parameters (40 bytes)

Authentication Algorithm: Simultaneous Authentication of Equals (SAE) (3)

Authentication SEQ: 0x0002

Status code: Successful (0x0000)

SAE Message Type: Confirm (2)

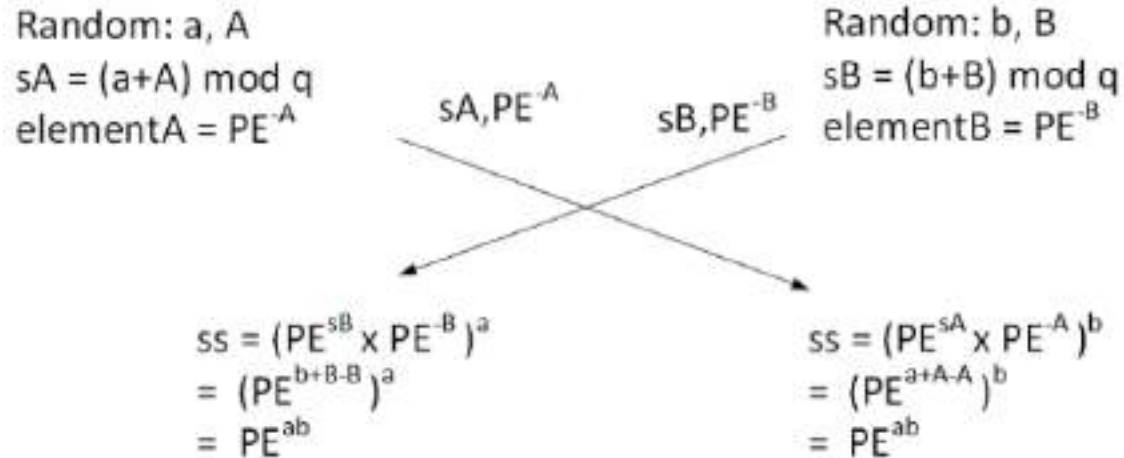
Send-Confirm: 1

Confirm: e05e00747ffce2d04a55d7d7d32296c5b8ffa07e5777d2dfa3f7a8e74fce2343

SAE (Simultaneous Authentication of Equals) Handshake

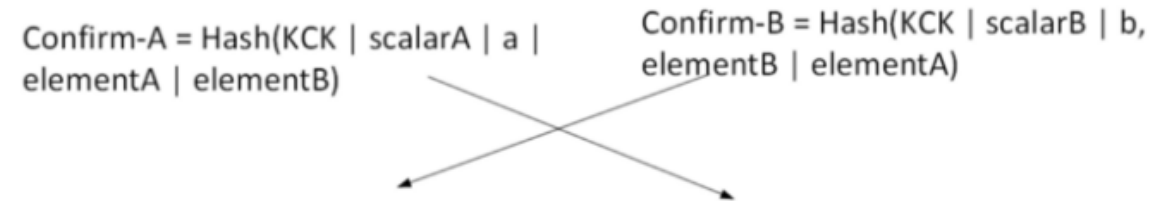
- SAE (Simultaneous Authentication of Equals) is known for Dragonfly key exchange in RFC7664 <https://www.rfc-editor.org/info/rfc7664>

Auth-Commit



- AP and STA calculate their own and the other side Scalar and Finite field element to create and share PE (Password Equivalent) value using Elliptic Curve cryptography

Auth-Confirm

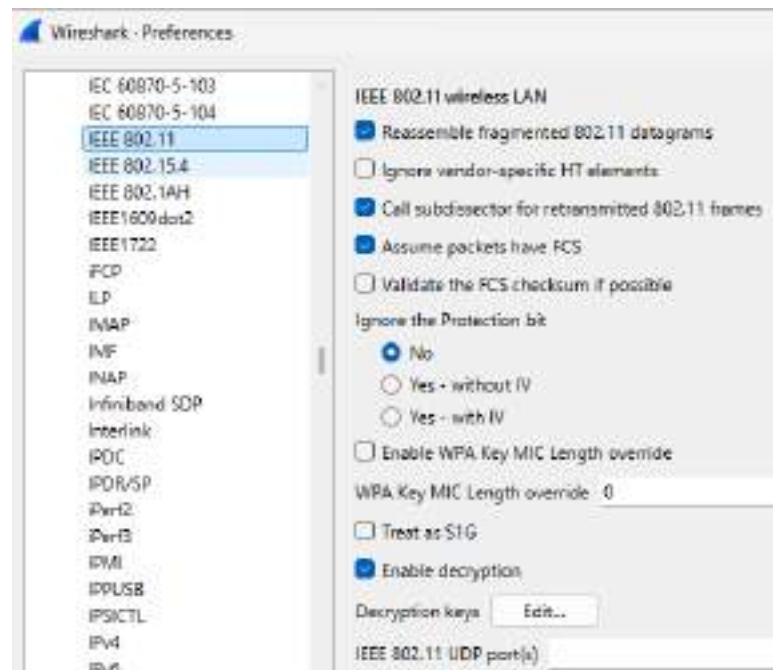


- Each Alice (STA) and Bob (AP) can verify the packet's Confirm value with the calculated Confirm value
- $K = rB \cdot (sA \cdot P + eA)$
 $tr = (sB, eB, sA, eA)$
 $cB = \text{HMAC}(\text{Hash}(K), tr)$
- If the calculated Confirm value is the same as the packet, we can share PE (Password Equivalent) value

Confirm Data frames are encrypted

```
▼ IEEE 802.11 QoS Data, Flags: op.....T
  Type/Subtype: QoS Data (0x0028)
  > Frame Control Field: 0x88c1
    .000 0000 0010 1100 = Duration: 44 microseconds
  > Receiver address: ASUS_AP (f0:2f:74:c4:f5:c4)
  > Transmitter address: iPadSTA (e2:da:1e:a8:92:8f)
  > Destination address: IPv6mcast_ff:01:27:fa (33:33:ff:01:27:fa)
  > Source address: iPadSTA (e2:da:1e:a8:92:8f)
  > BSS Id: ASUS_AP (f0:2f:74:c4:f5:c4)
  > STA address: iPadSTA (e2:da:1e:a8:92:8f)
    .... 0000 = Fragment number: 0
    0000 0000 0000 .... = Sequence number: 0
  [WLAN Flags: op.....T]
  > Qos Control: 0x3016
  > HT Control (+HTC): 0x0000b20f
  ▼ CCMP parameters
    CCMP Ext. Initialization Vector: 0x00000000000002
    Key Index: 0
  ▼ Data (88 bytes)
    Data: df24237032516733724bf86184cfb9531849ca08f1bfff3f180fdb96d
    [Length: 88]
```

- Check data frame #9, #10, #11, #12 encrypted by WPA2 and #23, #24, #25 by WPA3
- Open IEEE802.11 QoS Data -> CCMP parameters header, these frames contain CCMP Ext. Initialization Vector as a packet counter to protect against Replay Attack, AP and STA drops frames with old IVs



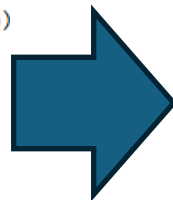
- Wireshark supports WEP, WPA1/2 decryption and doesn't support WPA3
- Wireshark also supports TK(Temporal Key) as a session key and MSK(Master Session Key) style decryption too.
- Right-click the IEEE 802.11 header, select Protocol Preferences -> Open IEEE 802.11 wireless LAN preferences.
- Confirm “Enable decryption” is checked, Open the “Edit...” button of Decryption keys.
- Note: you need to capture complete 4 packets of EAPOL 4-way handshake and you need to know SSID and pre shared key.

WEP and WPA Decryption Keys

WEP and WPA Decryption Keys

| Key type | Key |
|----------|----------------------------------|
| wep | wireshark:ikeriri6 |
| wep | wireshark:ikeriri7 |
| wpa-pwd | wireshark:ikeriri-5g |
| wpa-psk | wireshark:ikeriri-24g2 |
| tk | 99775e9a0854ac7899e11147547dd8f7 |
| msk | |

```
IEEE 802.11 QoS Data, Flags: op.....T
  Type/Subtype: QoS Data (0x0028)
  > Frame Control Field: 0x88c1
    .000 0000 0010 1100 = Duration: 44 microseconds
  > Receiver address: ASUS_AP (f0:2f:74:c4:f5:c4)
  > Transmitter address: iPadSTA (e2:da:1e:a8:92:8f)
  > Destination address: IPv6mcast_ff:01:27:fa (33:33:ff:01:27:fa)
  > Source address: iPadSTA (e2:da:1e:a8:92:8f)
  > BSS Id: ASUS_AP (f0:2f:74:c4:f5:c4)
  > STA address: iPadSTA (e2:da:1e:a8:92:8f)
    .... .... 0000 = Fragment number: 0
    0000 0000 0000 .... = Sequence number: 0
  [WLAN Flags: op.....T]
  > Qos Control: 0x3016
  > HT Control (+HTC): 0x0000b20f
  > CCMP parameters
    CCMP Ext. Initialization Vector: 0x0000000000002
    Key Index: 0
  > Data (88 bytes)
    Data: df24237032516733724bf86184cfb9531849ca08f1bff3f180fdb96d
    [Length: 88]
```



- You can set multiple WEP and WPA decryption Keys in preference.
- Select Key type as wpa-pwd and set Key as [Pre shared key:SSID] style
- Select “wpa-pwd” from the list box and set “wireshark:ikeriri6”

```
IEEE 802.11 QoS Data, Flags: op.....T
  Type/Subtype: QoS Data (0x0028)
  > Frame Control Field: 0x88c1
    .... ..00 = Version: 0
    .... 10.. = Type: Data frame (2)
    1000 .... = Subtype: 8
  > Flags: 0xc1
    .000 0000 0010 1100 = Duration: 44 microseconds
  > Receiver address: ASUS_AP (f0:2f:74:c4:f5:c4)
  > Transmitter address: iPadSTA (e2:da:1e:a8:92:8f)
  > Destination address: IPv6mcast_ff:01:27:fa (33:33:ff:01:27:fa)
  > Source address: iPadSTA (e2:da:1e:a8:92:8f)
  > BSS Id: ASUS_AP (f0:2f:74:c4:f5:c4)
  > STA address: iPadSTA (e2:da:1e:a8:92:8f)
    .... .... 0000 = Fragment number: 0
    0000 0000 0000 .... = Sequence number: 0
  [WLAN Flags: op.....T]
  > Qos Control: 0x3016
  > HT Control (+HTC): 0x0000b20f
  > CCMP parameters
    CCMP Ext. Initialization Vector: 0x0000000000002
    Key Index: 0
    [TK: 4c102fd43613c535404d0777088a6503]
    [PMK: 31bb75a609a424aac01e9929b39458e87ea45b0f30204ff5642bf3067a6fd31f]
  > Logical-Link Control
  > Internet Protocol Version 6, Src: ::, Dst: ff02::1:ff01:27fa
  > Internet Control Message Protocol v6
```

WEP and WPA Decryption Keys

| Info |
|---|
| Authentication, SN=833, FN=0, Flags=..... |
| Authentication, SN=0, FN=0, Flags=..... |
| Association Request, SN=834, FN=0, Flags=....., |
| Association Response, SN=1, FN=0, Flags=..... |
| Key (Message 1 of 4) |
| Key (Message 2 of 4) |
| Key (Message 3 of 4) |
| Key (Message 4 of 4) |
| Neighbor Solicitation for fe80::1c42:c607:6801:27f. |
| Who has 192.168.50.1? Tell 192.168.50.236 |
| DHCP Request - Transaction ID 0xac9e7500 |
| Router Solicitation |
| Authentication, SN=3841, FN=0, Flags=..... |
| Authentication, SN=0, FN=0, Flags=..... |
| Authentication, SN=3842, FN=0, Flags=..... |
| Authentication, SN=1, FN=0, Flags=..... |
| Association Request, SN=3843, FN=0, Flags=..... |
| Association Response, SN=2, FN=0, Flags=..... |
| Key (Message 1 of 4) |
| Key (Message 2 of 4) |
| Key (Message 3 of 4) |
| Key (Message 4 of 4) |
| QoS Data, SN=0, FN=0, Flags=op.....T |
| QoS Data, SN=1, FN=0, Flags=op.....T |
| QoS Data, SN=2, FN=0, Flags=op.....T |

- We can find plaintext data frames such as Neighbour Solicitation of IPv6(#9), ARP Requests(#10), DHCP Request(#11) and Router Solicitation of IPv6(#12) as the result of decryption of WPA2-PSK
- But We still can not decrypt WPA3 if we use the same SSID and Pass phrase, and if we have the complete set of SAE Authentications and 4-way handshake.

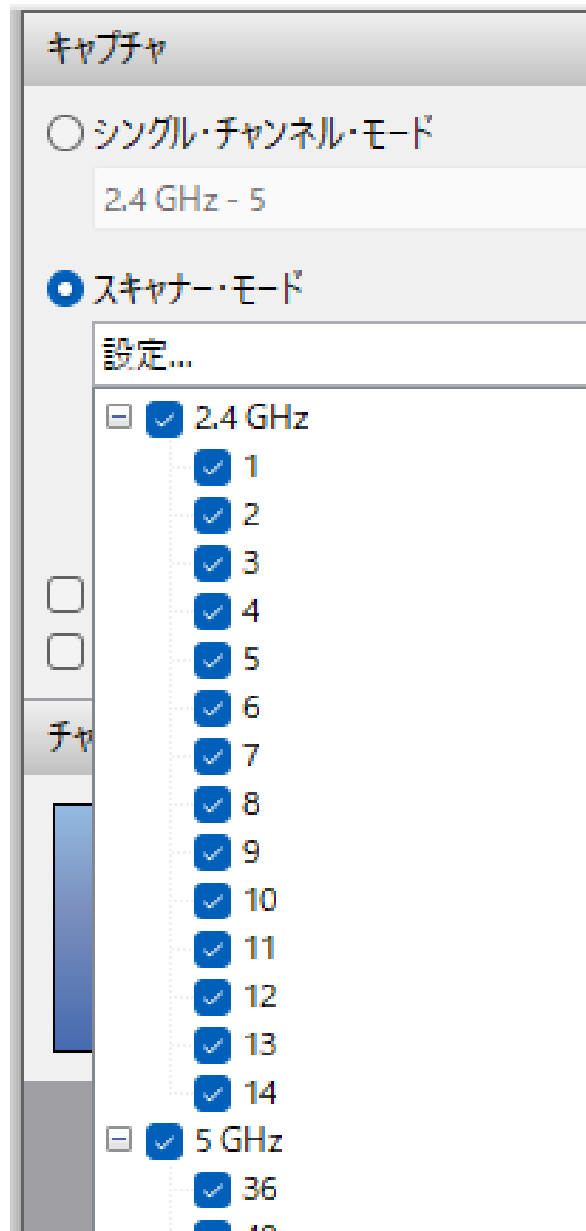
```
> Frame 9: Packet, 170 bytes on wire (1360 bits), 170 bytes captured
> Radiotap Header v0, Length 44
> 802.11 radio information
> IEEE 802.11 QoS Data, Flags: op.....T
> Logical-Link Control
> Internet Protocol Version 6, Src: ::, Dst: ff02::1:ff01:27fa
> Internet Control Message Protocol v6
```

Wireless in Live Action

Actual use case of Wireless packet capturing

- We can capture the complete set of 4-way handshake and connection sequence between AP and STA, if we test in the Anechoic Chamber or in the shield box.
- But wireless packet analysis in live actions is far different, we merely capture the complete sets of 4-way handshakes, nor connection sequences.
- WPA3 and PMF(Protected Management Frame) is common now.
- But we still get the various information about the Wireless environment from the trace file.
- WiFi SIGINT is WiFi Signal Intelligence, get Important Information like finding the needle in a haystack of the WiFi trace file





- It is inevitable to drop many wireless packets, and we need a huge amount of time and money to get all frames, and it may be impossible. We do not have to pursue full capturing.
- We do not have to lock in fixed WiFi channel, but switching channel in short interval, scanning wireless environment.
- 200ms/CH is good parameter of WiFi scanning, because 100ms is usual time of Beacon interval of the AP
- For example, set your WiFi capture equipment to scanner mode, set all 2.4/5/6GHz channels, capture WiFi packets at 200ms/CH



- We have smart devices everywhere and every time, and all devices have WiFi and their MAC address.
- When you visit your favorite cafe, you bring your MAC address and your information.
- We can map the network, identify devices and access points, detect unknown STAs, rogue APs, attackers and so on

WiFi SIGINT (WiFi Signal Intelligence)

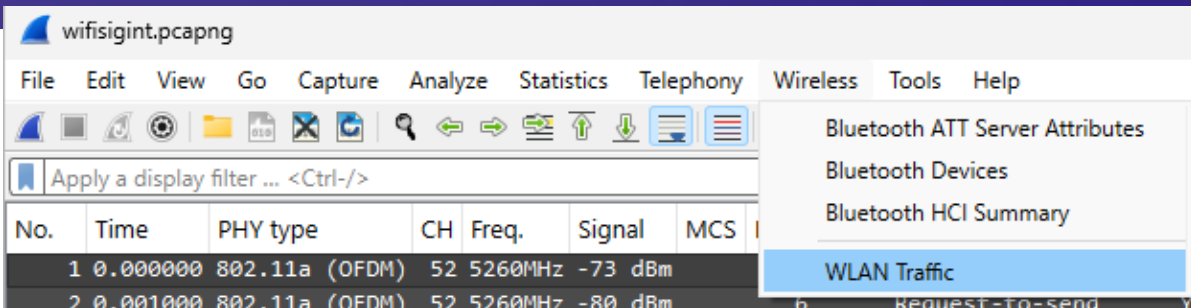


- CEATEC 2025 is one of the biggest IT conference events in Japan.
- Ikeriri booth is located at Hall 6 in the 75,000m² square feet of Makuhari Messe, Chiba prefecture.
- Open wifisigint.pcapng, 11MB trace file at 15:25:21 to 15:25:41 (79 seconds) in 14th, October, the first day of the conference
- Choose Statistics -> Capture File Properties to confirm the number of packets (53809), size of the trace file



Wireless LAN Statistics

- Choose Wireless -> WLAN Traffic to display Wireless LAN Statistics



Wireshark · Wireless LAN Statistics · wifisigint.pcapng

| BSSID | Channel | SSID | Percent Packets | Percent Retry | Retry | Beacons | Data Pkts | Probe Reqs | Probe Resps | Auths | Deauths | Other | Protection |
|---------------------|---------|-----------------|-----------------|---------------|-------|---------|-----------|------------|-------------|-------|---------|-------|------------|
| > | | <Broadcast> | 0.0 | 0.0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | |
| > 00:00:00:00:00:00 | | <Broadcast> | 0.0 | 0.0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | |
| > 00:09:b4:71:01:18 | 132 | 0000softbank | 0.1 | 38.5 | 5 | 4 | 3 | 0 | 6 | 0 | 0 | 0 | Unknown |
| > 00:09:b4:71:07:8c | 112 | 0002softbank | 0.0 | 33.3 | 1 | 0 | 0 | 1 | 2 | 0 | 0 | 0 | |
| > 00:09:b4:71:0a:c0 | 124 | 0000softbank | 0.0 | 75.0 | 3 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | |
| > 00:09:b4:71:13:48 | 116 | ABCeatec_EXT | 0.1 | 6.3 | 1 | 1 | 0 | 11 | 4 | 0 | 0 | 0 | |
| > 00:09:b4:71:15:2f | 11 | 0000softbank | 0.0 | 0.0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | |
| > 00:09:b4:71:15:30 | 140 | 0000softbank | 0.1 | 25.0 | 3 | 3 | 0 | 0 | 9 | 0 | 0 | 0 | |
| > 00:25:00:ff:94:73 | | <Broadcast> | 0.5 | 14.9 | 10 | 0 | 4 | 0 | 0 | 0 | 0 | 63 | |
| > 00:2b:f5:5b:85:2e | 1 | <Broadcast> | 0.0 | 0.0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | |
| > 00:2b:f5:5b:85:30 | 128 | Buffalo-5G-8520 | 0.1 | 0.0 | 0 | 2 | 0 | 0 | 11 | 0 | 0 | 0 | |

Display filter: Enter a display filter ...

Buttons: Copy, Save as..., Close, Help

- We can sort, extract and right-click to filter BSSID, Channel, SSID, Percent Packets, Percent Retry, Retry, Beacons, Data Pkts, Probe Reqs, Probe Resps, Auths, Deauths, Other and Protection Columns

Counting the transmitter of wireless traffic

- tshark is a good tool for counting APs and STAs
- Let's list Transmitter address (wlan.ta) of the wifisigint.pcapng
- Use -T fields -e wlan.ta options to pick up Transmitter Address

```
C:\Users\megumi\Desktop>tshark -T fields -e wlan.ta -r wifisigint.pcapng  
d0:65:78:17:f8:22  
f5:d5:80:1d:01:e0
```

- Redirect the result as TA.txt and cat TA.txt | sort | uniq | wc -l to count up the unique MAC address of the transmitter.

```
C:\Users\megumi\Desktop>tshark -T fields -e wlan.ta -r wifisigint.pcapng >>TA.txt  
  
C:\Users\megumi\Desktop>bash  
[user@area51]~/mnt/c/Users/megumi/Desktop  
$ cat TA.txt | sort | uniq | wc -l  
1344
```

Counting the number of AP and STA

- AP sends Beacons, so counting unique AP by using
-Y wlan.fc.type_subtype==8 and redirect as AP.txt

```
C:\Users\megumi\Desktop>tshark -T fields -e wlan.ta -r wifisigint.pcapng -Y wlan.fc.type_subtype==8 >>AP.txt

C:\Users\megumi\Desktop>bash
[user@area51]~/mnt/c/Users/megumi/Desktop
$ cat AP.txt | sort | uniq | wc -l
215
```

- Counting AP using sort, uniq and wc -l
There are 215 Aps in this trace file
- All MAC addresses 1344 - AP's MAC address 215 = 1129 STAs

- A private MAC Address is a locally administered MAC address, a hardware address that is not globally assigned by a manufacturer (OUI) but instead generated by the device or operating system
- Smart devices such as iOS and Android OS use a Private MAC address as the default.
- ✓ Source: Dell_5d:3c:d4 (cc:48:3a:5d:3c:d4)

```
.....0..... = LG bit: Globally unique address (factory default)  
.....0..... = IG bit: Individual address (unicast)
```
- Private MAC addresses' LG bit (Specifies if this is a locally administered or globally unique (IEEE assigned) is true, that means the 7th bit of the first byte of the MAC Address is 1
- For example, Private MAC address starts with x2, x6, xA, xE, etc.
- We can filter Private MAC address as Display Filter Expression as `eth.src.lg==1` in the Wired network

- If we want to filter the Private MAC Address in a Wireless network, we can filter as Wireshark Display Filter syntax as `wlan.ta[0]&0x02==0x02`
`wlan.ta[0]` means the first byte of the transmitter address
& means bit-based AND calculation
`0x02` is the mask pattern matched for the 7th bit of the first byte
- Using tshark to filter the transmitter MAC address that is private

```
C:\Users\megumi\Desktop>tshark -T fields -e wlan.ta -r wifisigint.pcapng -Y "wlan.ta[0]&0x02==0x02" >>private.txt

C:\Users\megumi\Desktop>bash
[user@area51]~/mnt/c/Users/megumi/Desktop
$ cat private.txt | sort | uniq | wc -l
812
```

- Using `cat`, `sort`, `uniq`, and `wc` commands to count the private MAC
Private MAC STAs: 812, all STAs: 1344
so two-thirds(about 66%) of STAs are smart devices.

Try Display filter as “wlan.ta[0]&0x02==0x02”

wifisigint.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

wlan.ta[0]&0x02==0x02

| No. | Time | PHY type | CH | Freq. | Signal | MCS | Rate | Retry | Type/Subtype | Source | Destination | SSID | BSSID | STA | Info | Private |
|------|----------|------------------|----|---------|---------|-----|------|-------|------------------|-------------------|-----------------------|---------|-------------|---------|--|---------|
| 25 | 0.577000 | 802.11b (HR/D... | 3 | 2422MHz | -66 dBm | | | 1 | Probe Request | a6:d0:9d:65:c8:35 | Broadcast | <MIS... | Broadcast | | Probe Request, SN=2048, FN=0, Flag | Strange |
| 32 | 0.784000 | 802.11b (HR/D... | 4 | 2427MHz | -67 dBm | | | 1 ✓ | Probe Response | 5e:cf:cf:20:fc:12 | Intel_dc:aa:a0 | "AQU... | 5e:cf:cf... | | Probe Response, SN=1086, FN=0, Flags=... | |
| 55 | 1.672000 | 802.11b (HR/D... | 9 | 2452MHz | -68 dBm | | | 1 | Request-to-send | 72:ca:54:e3:b9:af | HuaweiDevice_1f:f0:66 | | | | Request-to-send, Flags=..... | |
| 80 | 2.233000 | 802.11b (HR/D... | 12 | 2467MHz | -67 dBm | | | 1 | Probe Request | 06:bb:3b:ef:10:48 | Broadcast | <MIS... | Broadcast | | Probe Request, SN=2785, FN=0, Flags=... | |
| 1... | 2.674000 | 802.11a (OFDM) | 36 | 5180MHz | -82 dBm | | 24 | | 802.11 Block Ack | 4e:1d:4b:10:28:23 | RuckusWirele_37:23:0c | | | | 802.11 Block Ack, Flags=..... | |
| 1... | 2.683000 | 802.11a (OFDM) | 36 | 5180MHz | -82 dBm | | 6 | | Beacon frame | 22:7c:98:ec:4f:37 | Broadcast | <MIS... | 22:7c:98... | | Beacon frame, SN=2973, FN=0, Flags=..... | |
| 1... | 2.707000 | 802.11a (OFDM) | 36 | 5180MHz | -73 dBm | | 6 | | Probe Request | 7a:fb:e4:7a:8c:eb | Broadcast | <MIS... | Broadcast | | Probe Request, SN=2289, FN=0, Flags=... | |
| 1... | 2.739000 | 802.11a (OFDM) | 36 | 5180MHz | -81 dBm | | 24 | | Request-to-send | 4e:1d:4b:10:28:23 | RuckusWirele_37:23:0c | | | | Request-to-send, Flags=..... | |
| 1... | 2.773000 | 802.11a (OFDM) | 36 | 5180MHz | -83 dBm | | 6 | | Probe Request | 8e:48:67:dc:e9:a9 | Broadcast | <MIS... | Broadcast | | Probe Request, SN=3147, FN=0, Flags=... | |
| 2... | 2.847000 | 802.11a (OFDM) | 40 | 5200MHz | -77 dBm | | 6 | | Probe Response | c6:4e:37:34:2e:d8 | 76:50:a3:da:b2:aa | "Glo... | c6:4e:37... | | Probe Response, SN=140, FN=0, Flags=... | |
| 2... | 2.862000 | 802.11a (OFDM) | 40 | 5200MHz | -80 dBm | | 6 | | Request-to-send | 33:89:d3:c6:92:0a | AskeyCompute_98:35:db | | | | Request-to-send, Flags=..... | |
| 2... | 2.862000 | 802.11a (OFDM) | 40 | 5200MHz | -79 dBm | | 24 | | Request-to-send | 82:97:81:2d:14:70 | ZhongGeSmart_6b:8d:12 | | | | Request-to-send, Flags=..... | |
| 2... | 2.873000 | 802.11a (OFDM) | 40 | 5200MHz | -80 dBm | | 6 | | Request-to-send | 33:89:d3:c6:92:0a | AskeyCompute_98:35:db | | | | Request-to-send, Flags=..... | |
| 2... | 2.891000 | 802.11ax (HE) | 40 | 5200MHz | -77 dBm | | | | QoS Data | f2:d5:65:11:cc:75 | TPLink_72:a5:88 | | TPLink_7... | f2:d... | QoS Data, SN=3442, FN=0, Flags=op....T | |
| 2... | 2.891000 | 802.11ax (HE) | 40 | 5200MHz | -77 dBm | | | | QoS Data | f2:d5:65:11:cc:75 | TPLink_72:a5:88 | | TPLink_7... | f2:d... | QoS Data, SN=3448, FN=0, Flags=op....T | |
| 2... | 2.891000 | 802.11ax (HE) | 40 | 5200MHz | -77 dBm | | | | QoS Data | f2:d5:65:11:cc:75 | TPLink_72:a5:88 | | TPLink_7... | f2:d... | QoS Data, SN=3451, FN=0, Flags=op....T | |
| 2... | 2.891000 | 802.11ax (HE) | 40 | 5200MHz | -77 dBm | | | | QoS Data | f2:d5:65:11:cc:75 | TPLink_72:a5:88 | | TPLink_7... | f2:d... | QoS Data, SN=3455, FN=0, Flags=op....T | |
| 2... | 2.892000 | 802.11ax (HE) | 40 | 5200MHz | -77 dBm | | | | QoS Data | f2:d5:65:11:cc:75 | TPLink_72:a5:88 | | TPLink_7... | f2:d... | QoS Data, SN=3465, FN=0, Flags=op....T | |
| 2... | 2.892000 | 802.11ax (HE) | 40 | 5200MHz | -77 dBm | | | | QoS Data | f2:d5:65:11:cc:75 | TPLink_72:a5:88 | | TPLink_7... | f2:d... | QoS Data, SN=3477, FN=0, Flags=op....T | |
| 2... | 2.915000 | 802.11a (OFDM) | 40 | 5200MHz | -74 dBm | | 6 | | Beacon frame | c6:4e:37:34:2e:d8 | Broadcast | "Glo... | c6:4e:37... | | Beacon frame, SN=141, FN=0, Flags=..... | |
| 2... | 2.947000 | 802.11a (OFDM) | 40 | 5200MHz | -78 dBm | | 12 | | Request-to-send | f2:d5:65:11:cc:75 | TPLink_72:a5:87 | | | | Request-to-send, Flags=..... | |
| 2... | 2.970000 | 802.11a (OFDM) | 40 | 5200MHz | -81 dBm | | 6 | | Request-to-send | 32:89:d3:c6:92:0a | AskeyCompute_98:38:4d | | | | Request-to-send, Flags=..... | |
| 2... | 2.970000 | 802.11a (OFDM) | 40 | 5200MHz | -79 dBm | | 24 | | Request-to-send | 82:97:81:2d:14:70 | ZhongGeSmart_6b:8d:12 | | | | Request-to-send, Flags=..... | |
| 2... | 3.012000 | 802.11a (OFDM) | 44 | 5220MHz | -81 dBm | | 6 | | Probe Request | e6:0c:8b:f3:d9:2c | Broadcast | "Hal... | Broadcast | | Probe Request, SN=3872, FN=0, Flags=... | |
| 2... | 3.012000 | 802.11a (OFDM) | 44 | 5220MHz | -77 dBm | | 24 | | 802.11 Block Ack | 82:97:81:2d:14:70 | ZhongGeSmart_6b:8d:12 | | | | 802.11 Block Ack, Flags=..... | |

- Two-thirds of the STAs are Private MAC addresses in the conference center

Important hint of iOS's Private MAC Address

- iPhone and iPad use a Private MAC address by default, iOS uses randomized MAC address in different locations.
- But iOS has interesting characteristics, iPhone and iPad use the same Private MAC address in the exact locations every time.
- That helps us identify the devices, for example, you watched some Private MAC address on Monday, and you find the same Private MAC address on Tuesday. The device is the same!!



✓ Transmitter address: e2:ea:bc:1d:45:f7 (e2:ea:bc:1d:45:f7)

.....1..... = LG bit: Locally administered address (this is NOT the factory default)

.....0..... = IG bit: Individual address (unicast)

Common pattern of MAC address

マルチSSID(2.4GHz SSID)

SSID:

チャンネル:

暗号化:

BSSID:

2.4GHz BSSID

接続端末台数:

マルチSSID(5GHz SSID)

SSID:

チャンネル:

暗号化:

BSSID:

5GHz BSSID

接続端末台数:

ゲスト-SSID(2.4GHz SSID)

SSID:

チャンネル:

暗号化:

BSSID:

2.4GHz Guest BSSID

接続端末台数:

ゲスト-SSID(5GHz SSID)

SSID:

チャンネル:

暗号化:

BSSID:

5GHz Guest BSSID

戻る

- We experienced the interesting pattern of BSSID in the same AP.
- AP uses a sequential MAC address for different bands' BSSID, for example
B0:BE:76:12:34:56 for 2.4GHz band
B0:BE:76:12:34:57 for 5GHz band
B0:BE:76:12:34:58 for 6GHz band
- Sometimes an AP uses multiple SSIDs with similar BSSIDs
B0:BE:76:12:ab:cd for main SSID
B0:BE:76:12:ab:ce for guest SSID
- First 3 bytes are the same -> same OUI
- First 4 bytes are the same -> same AP

Interesting MAC Address



| wlan.ta in {00-00-00-00-00-00, 11-22-33-44-55-66, de-ad-be-ef-00-00} | | | | | | | | | | | | | | + Management | | Control Frames | Data | Signal | Rate | Retry | EAPOL | PWR |
|--|---------|----------------|----|---------|---------|-----|------|-------|---------------------|-----------|-------------|------|-----------|--------------|---|----------------|------|--------|------|-------|-------|-----|
| No. | Time | PHY type | CH | Freq. | Signal | MCS | Rate | Retry | Type/Subtype | Source | Destination | SSID | BSSID | STA | Info | | | | | | | |
| 2. | 63.4880 | 802.11a (OFDM) | 1. | 5580MHz | -66 dBm | 6 | | 6 | Association Request | 00:00:00_ | 00:00:00_ | | 00:00:00_ | | Association Request, SN=3785, FN=0, Flags=.....[Malformed Packet] | | | | | | | |
| 2. | 64.2440 | 802.11a (OFDM) | 1. | 5580MHz | -67 dBm | 6 | | 6 | Association Request | 00:00:00_ | 00:00:00_ | | 00:00:00_ | | Association Request, SN=3799, FN=0, Flags=.....[Malformed Packet] | | | | | | | |
| 2. | 65.4070 | 802.11a (OFDM) | 1. | 5580MHz | -75 dBm | 6 | | 6 | Association Request | 00:00:00_ | 00:00:00_ | | 00:00:00_ | | Association Request, SN=3445, FN=0, Flags=.....[Malformed Packet] | | | | | | | |
| 3. | 66.7310 | 802.11a (OFDM) | 1. | 5580MHz | -76 dBm | 6 | | 6 | Association Request | 00:00:00_ | 00:00:00_ | | 00:00:00_ | | Association Request, SN=3482, FN=0, Flags=.....[Malformed Packet] | | | | | | | |
| 3. | 69.1370 | 802.11a (OFDM) | 1. | 5580MHz | -81 dBm | 6 | | 6 | Association Request | 00:00:00_ | 00:00:00_ | | 00:00:00_ | | Association Request, SN=3900, FN=0, Flags=.....[Malformed Packet] | | | | | | | |
| 3. | 70.4610 | 802.11a (OFDM) | 1. | 5580MHz | -63 dBm | 6 | | 6 | Association Request | 00:00:00_ | 00:00:00_ | | 00:00:00_ | | Association Request, SN=3933, FN=0, Flags=.....[Malformed Packet] | | | | | | | |
| 4. | 71.8880 | 802.11a (OFDM) | 1. | 5580MHz | -75 dBm | 6 | | 6 | Association Request | 00:00:00_ | 00:00:00_ | | 00:00:00_ | | Association Request, SN=3562, FN=0, Flags=.....[Malformed Packet] | | | | | | | |
| 4. | 72.4130 | 802.11a (OFDM) | 1. | 5580MHz | -74 dBm | 6 | | 6 | Association Request | 00:00:00_ | 00:00:00_ | | 00:00:00_ | | Association Request, SN=3571, FN=0, Flags=.....[Malformed Packet] | | | | | | | |
| 4. | 73.2350 | 802.11a (OFDM) | 1. | 5580MHz | -67 dBm | 6 | | 6 | Association Request | 00:00:00_ | 00:00:00_ | | 00:00:00_ | | Association Request, SN=3982, FN=0, Flags=.....[Malformed Packet] | | | | | | | |
| 5. | 77.0330 | 802.11a (OFDM) | 1. | 5580MHz | -65 dBm | 6 | | 6 | Association Request | 00:00:00_ | 00:00:00_ | | 00:00:00_ | | Association Request, SN=4042, FN=0, Flags=.....[Malformed Packet] | | | | | | | |
| 5. | 77.5470 | 802.11a (OFDM) | 1. | 5580MHz | -75 dBm | 6 | | 6 | Association Request | 00:00:00_ | 00:00:00_ | | 00:00:00_ | | Association Request, SN=3676, FN=0, Flags=.....[Malformed Packet] | | | | | | | |

- Sometimes an attacker uses interesting MAC addresses,
00:11:22:33:44:55 for debugging/testing
DE:AD:BE:EF:00:00 for joking
AA:BB:CC:DD:EE:FF is the default value of the attack tool
00:00:00:00:00:00 is seen when the packet is broken
If you find these MAC addresses on your network, someone is measuring, attacking your WiFi, or packets are broken.
- For example, try display filter as wlan.ta in {00-00-00-00-00-00, 11-22-33-44-55-66, de-ad-be-ef-00-00, aa-bb-cc-dd-ee-ff}

Pick up the certain SSID

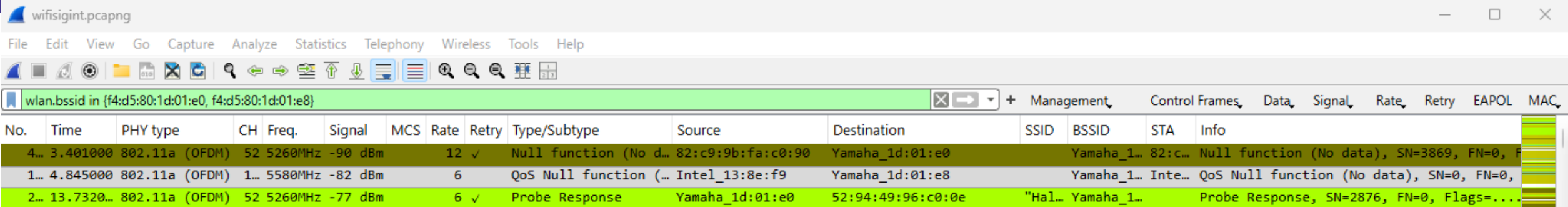


- The capture point is Ikeriri's booth at Hall 6
- Working Lounge is located at Hall 5, about 100 meters away from the capture point.
- Wireless -> WLAN traffic, sort the SSID column, and pick up "Hall5_Working_Lounge_Space" that BSSID is f4:d5:80:1d:01:e0 and f4:d5:80:1d:01:e8

Wireshark · Wireless LAN Statistics · wifisigint.pcapng

| BSSID | Channel | SSID | Percent Packets | Percent Retry | Retry | Beacons | Data Pkts | Probe Reqs | Probe Resp | Auths | Deauths | Other | Protection |
|---------------------|---------|----------------------------|-----------------|---------------|-------|---------|-----------|------------|------------|-------|---------|-------|------------|
| > f4:d5:80:1d:01:e0 | 52 | Hall5_Working_Lounge_Space | 2.3 | 65.6 | 221 | 16 | 215 | 0 | 95 | 2 | 0 | 9 | Unknown |
| > f4:d5:80:1d:01:e8 | 116 | Hall5_Working_Lounge_Space | 4.9 | 51.9 | 379 | 80 | 404 | 0 | 184 | 34 | 5 | 23 | Unknown |

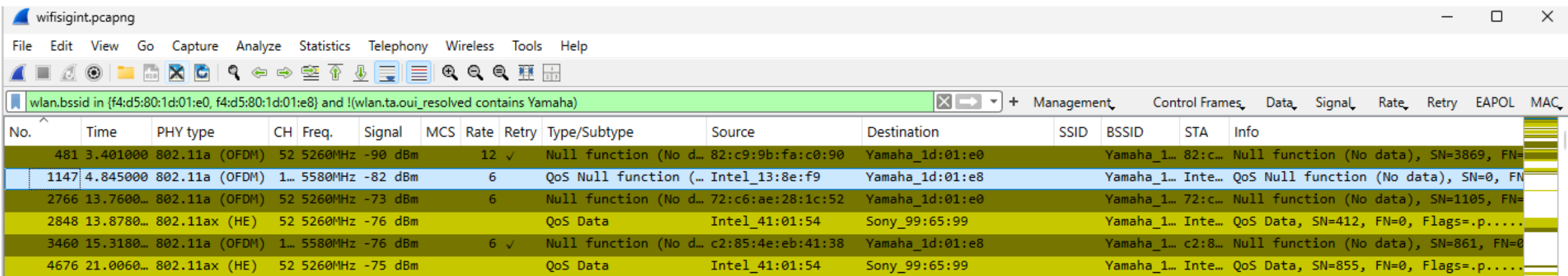
“Hall5_Working_Lounge_Space”



Wireshark capture of wireless traffic. The display filter is set to `wlan.bssid in {f4:d5:80:1d:01:e0, f4:d5:80:1d:01:e8}`. The table shows three packets:

| No. | Time | PHY type | CH | Freq. | Signal | MCS | Rate | Retry | Type/Subtype | Source | Destination | SSID | BSSID | STA | Info |
|------|------------|----------------|------|---------|---------|-----|------|-------|---------------------------------------|-------------------|-------------------|---------|-------------|---------|--|
| 4... | 3.401000 | 802.11a (OFDM) | 52 | 5260MHz | -90 dBm | | 12 | ✓ | Null function (No d... | 82:c9:9b:fa:c0:90 | Yamaha_1d:01:e0 | | Yamaha_1... | 82:c... | Null function (No data), SN=3869, FN=0, F... |
| 1... | 4.845000 | 802.11a (OFDM) | 1... | 5580MHz | -82 dBm | | 6 | | QoS Null function (... Intel_13:8e:f9 | | Yamaha_1d:01:e8 | | Yamaha_1... | Inte... | QoS Null function (No data), SN=0, FN=0, |
| 2... | 13.7320... | 802.11a (OFDM) | 52 | 5260MHz | -77 dBm | | 6 | ✓ | Probe Response | Yamaha_1d:01:e0 | 52:94:49:96:c0:0e | "Hal... | Yamaha_1... | | Probe Response, SN=2876, FN=0, Flags=... |

- Set Display Filter as `wlan.bssid in {f4:d5:80:1d:01:e0, f4:d5:80:1d:01:e8}` to filter “Hall5_Working_Lounge_Space” SSID
- This SSID is operated by two Yamaha APs, so pick up STAs as `wlan.bssid in {f4:d5:80:1d:01:e0, f4:d5:80:1d:01:e8}` and `!(wlan.ta.oui_resolved contains Yamaha)`



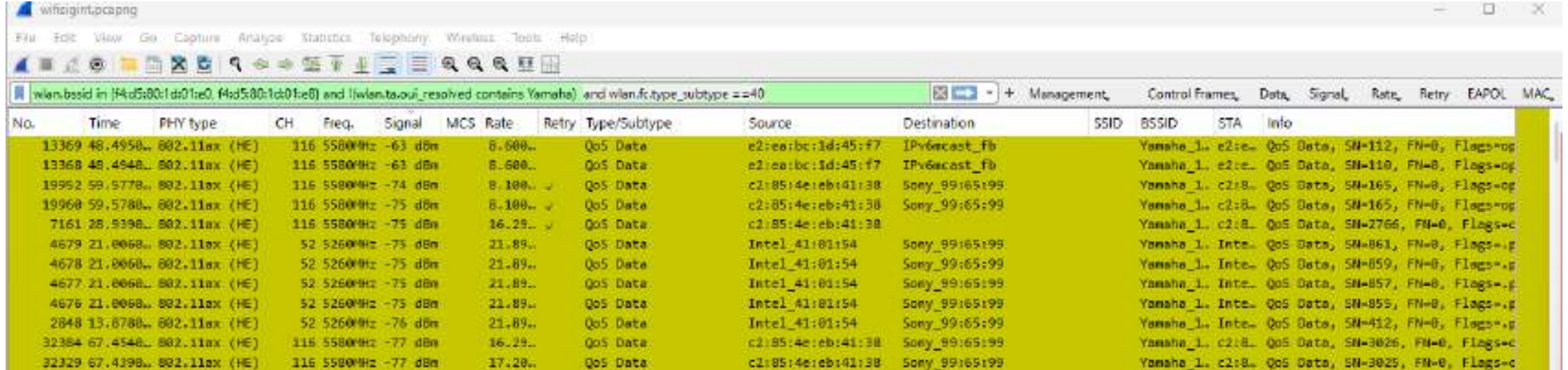
Wireshark capture of wireless traffic. The display filter is set to `wlan.bssid in {f4:d5:80:1d:01:e0, f4:d5:80:1d:01:e8} and !(wlan.ta.oui_resolved contains Yamaha)`. The table shows several packets, with packet 114 highlighted:

| No. | Time | PHY type | CH | Freq. | Signal | MCS | Rate | Retry | Type/Subtype | Source | Destination | SSID | BSSID | STA | Info |
|------|------------|----------------|------|---------|---------|-----|------|-------|---------------------------------------|-------------------|-----------------|------|-------------|---------|---|
| 481 | 3.401000 | 802.11a (OFDM) | 52 | 5260MHz | -90 dBm | | 12 | ✓ | Null function (No d... | 82:c9:9b:fa:c0:90 | Yamaha_1d:01:e0 | | Yamaha_1... | 82:c... | Null function (No data), SN=3869, FN=... |
| 114 | 4.845000 | 802.11a (OFDM) | 1... | 5580MHz | -82 dBm | | 6 | | QoS Null function (... Intel_13:8e:f9 | | Yamaha_1d:01:e8 | | Yamaha_1... | Inte... | QoS Null function (No data), SN=0, FN=... |
| 2766 | 13.7600... | 802.11a (OFDM) | 52 | 5260MHz | -73 dBm | | 6 | | Null function (No d... | 72:c6:ae:28:1c:52 | Yamaha_1d:01:e0 | | Yamaha_1... | 72:c... | Null function (No data), SN=1105, FN=... |
| 2848 | 13.8780... | 802.11ax (HE) | 52 | 5260MHz | -76 dBm | | | | QoS Data | Intel_41:01:54 | Sony_99:65:99 | | Yamaha_1... | Inte... | QoS Data, SN=412, FN=0, Flags=.p.... |
| 3460 | 15.3180... | 802.11a (OFDM) | 1... | 5580MHz | -76 dBm | | 6 | ✓ | Null function (No d... | c2:85:4e:eb:41:38 | Yamaha_1d:01:e8 | | Yamaha_1... | c2:8... | Null function (No data), SN=861, FN=0 |
| 4676 | 21.0060... | 802.11ax (HE) | 52 | 5260MHz | -75 dBm | | | | QoS Data | Intel_41:01:54 | Sony_99:65:99 | | Yamaha_1... | Inte... | QoS Data, SN=855, FN=0, Flags=.p.... |

- Let's think about the Signal Strength of this SSID

“Hall5_Working_Lounge_Space”

- We need to add a filter for the frames Type/Subtype is QoS Data, set Display Filter as wlan.bssid in {f4:d5:80:1d:01:e0, f4:d5:80:1d:01:e8} and and wlan.fc.type_subtype ==40
- Sort the Signal Column descending, from strong to weak signal

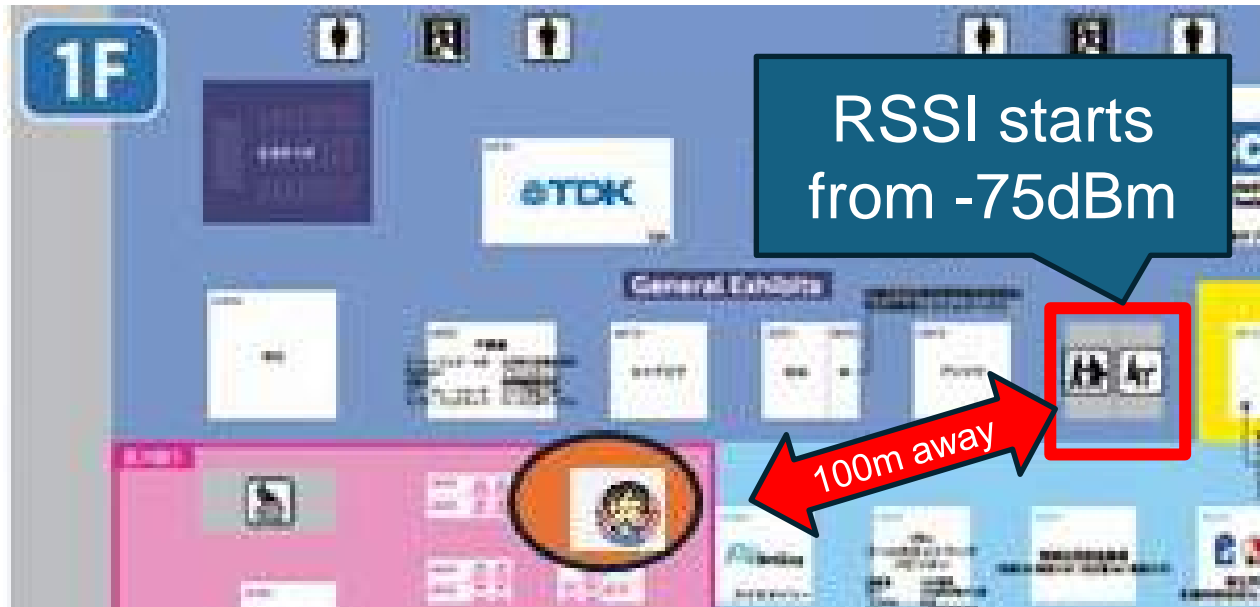


The screenshot shows a Wireshark packet capture window with the following display filter: `wlan.bssid in {f4:d5:80:1d:01:e0, f4:d5:80:1d:01:e8} and wlan.fc.type_subtype ==40`. The packets are sorted by the 'Signal' column in descending order. The table below represents the data visible in the packet list pane.

| No. | Time | PHY type | CH | Freq. | Signal | MCS | Rate | Retry | Type/Subtype | Source | Destination | SSID | BSSID | STA | Info |
|-------|------------|---------------|-----|---------|---------|----------|------|-------|--------------|-------------------|---------------|-----------|----------|-----|----------------------------------|
| 13369 | 48.4958... | 802.11ax (HE) | 116 | 5580MHz | -63 dBm | 8.600... | | | QoS Data | e2:ea:bc:1d:45:f7 | IPv6cast_fb | Yamaha_1. | e2:e... | | QoS Data, SN=112, FN=0, Flags=op |
| 13368 | 48.4948... | 802.11ax (HE) | 116 | 5580MHz | -63 dBm | 8.600... | | | QoS Data | e2:ea:bc:1d:45:f7 | IPv6cast_fb | Yamaha_1. | e2:e... | | QoS Data, SN=110, FN=0, Flags=op |
| 19952 | 59.5778... | 802.11ax (HE) | 116 | 5580MHz | -74 dBm | 8.100... | | | QoS Data | c2:85:4e:eb:41:38 | Sony_99:65:99 | Yamaha_1. | c2:8... | | QoS Data, SN=165, FN=0, Flags=op |
| 19968 | 59.5780... | 802.11ax (HE) | 116 | 5580MHz | -75 dBm | 8.100... | | | QoS Data | c2:85:4e:eb:41:38 | Sony_99:65:99 | Yamaha_1. | c2:8... | | QoS Data, SN=165, FN=0, Flags=op |
| 7161 | 28.9398... | 802.11ax (HE) | 116 | 5580MHz | -75 dBm | 16.29... | | | QoS Data | c2:85:4e:eb:41:38 | Sony_99:65:99 | Yamaha_1. | c2:8... | | QoS Data, SN=2766, FN=0, Flags=c |
| 4679 | 21.0068... | 802.11ax (HE) | 52 | 5260MHz | -75 dBm | 21.89... | | | QoS Data | Intel_41:01:54 | Sony_99:65:99 | Yamaha_1. | Intel... | | QoS Data, SN=861, FN=0, Flags=.p |
| 4678 | 21.0060... | 802.11ax (HE) | 52 | 5260MHz | -75 dBm | 21.89... | | | QoS Data | Intel_41:01:54 | Sony_99:65:99 | Yamaha_1. | Intel... | | QoS Data, SN=859, FN=0, Flags=.p |
| 4677 | 21.0068... | 802.11ax (HE) | 52 | 5260MHz | -75 dBm | 21.89... | | | QoS Data | Intel_41:01:54 | Sony_99:65:99 | Yamaha_1. | Intel... | | QoS Data, SN=857, FN=0, Flags=.p |
| 4676 | 21.0068... | 802.11ax (HE) | 52 | 5260MHz | -75 dBm | 21.89... | | | QoS Data | Intel_41:01:54 | Sony_99:65:99 | Yamaha_1. | Intel... | | QoS Data, SN=855, FN=0, Flags=.p |
| 2848 | 13.8780... | 802.11ax (HE) | 52 | 5260MHz | -76 dBm | 21.89... | | | QoS Data | Intel_41:01:54 | Sony_99:65:99 | Yamaha_1. | Intel... | | QoS Data, SN=412, FN=0, Flags=.p |
| 32384 | 67.4548... | 802.11ax (HE) | 116 | 5580MHz | -77 dBm | 16.29... | | | QoS Data | c2:85:4e:eb:41:38 | Sony_99:65:99 | Yamaha_1. | c2:8... | | QoS Data, SN=3026, FN=0, Flags=c |
| 32329 | 67.4398... | 802.11ax (HE) | 116 | 5580MHz | -77 dBm | 17.20... | | | QoS Data | c2:85:4e:eb:41:38 | Sony_99:65:99 | Yamaha_1. | c2:8... | | QoS Data, SN=3025, FN=0, Flags=c |

- Strongest two packets's Signal Strength(dBm) are -63dBm, But the others start from -74dBm to -92dBm
- 100 meters away's signal starts from -75dBm?

- This trace is captured at a huge conference site, like a stadium, A big open space, and many people visit the booth



- If we touch the wireless capture card with the devices, we get -30dBm in Japan (in Japanese radio law condition)
- Noise floor, the lowest signal is measured from -95dBm to -105 dBm in usual
- How about the office, the office floor is covered by many glass and gypsum board partitions.
- These materials have different attenuation values (in dB) and reflection percentages of 2.4/5/6GHz signal

- In general, attenuations and reflection rates of 2.4/5GHz signal

| Materials | 2.4GHz attenuation | 5GHz attenuation | Reflection Rate |
|-----------------|--------------------|------------------|-----------------|
| Glass partition | About 2-4dB | About 3-6dB | About 15-20% |
| Gypsum board | About 3dB | About 4-5dB | About 3-5% |
| Metal Door | About 16dB | About 28dB | About 100% |
| 20cm concrete | About 20-30dB | About 25-45dB | About 15-25% |

- These parameter helps us to determine the device is on site or off site, if you capture signals at -30dBm, the device locates in just a few meters away, if you captured at -31 to -50, the device may be in the same room, if you capture at -51 to -60, the device may locates in the same floor, if you capture at -61 to -79, the devices may be in the same building...

- Usual threshold of WiFi RSSI(Receive Signal Strength Indicator)

| RSSI(dBm) | Location | Details |
|--------------------|--------------------------------|---|
| From 0 to -30Bm | Same room | A few meters away |
| From -31 to -50dBm | Same room or next room | In the same room or next room divided with thin wall |
| From -51 to -60dBm | Same floor or same building | In the same floor or same building of next floor |
| From -61 to -70dBm | Same building | Reflected signals from 2-3 floors away |
| From -71 to -80dBm | Same building or outside | Reflected signals from outside or concrete wall |
| From -81 to -90dBm | Outside | The lowest signal for data |

- Edit -> Preferences and Choose Filter Buttons

| | | |
|-------------------------------------|------------------------------------|--|
| <input checked="" type="checkbox"/> | Signal//Same room | <code>radiotap.dbm_antsignal <=-0 and radiotap.dbm_antsignal >=-30</code> |
| <input checked="" type="checkbox"/> | Signal//Same Next room | <code>radiotap.dbm_antsignal <=-31 and radiotap.dbm_antsignal >=-50</code> |
| <input checked="" type="checkbox"/> | Signal//Same floor Same building | <code>radiotap.dbm_antsignal <=-51 and radiotap.dbm_antsignal >=-60</code> |
| <input checked="" type="checkbox"/> | Signal//Same building | <code>radiotap.dbm_antsignal <=-61 and radiotap.dbm_antsignal >=-70</code> |
| <input checked="" type="checkbox"/> | Signal//Same bld Outside | <code>radiotap.dbm_antsignal <=-71 and radiotap.dbm_antsignal >=-80</code> |
| <input checked="" type="checkbox"/> | Signal//Outside | <code>radiotap.dbm_antsignal <=-81 and radiotap.dbm_antsignal >=-90</code> |

- Let's start a live demonstration

Part1: Collect WiFi scan packets with CommView for WiFi

Collect WiFi scan packets with Metageek Apps

Part2: Save and merge traces as a pcapng trace file

Start WiFi SIGINT analysis with Wireshark!!

USE WIRESHARK

Thank you for your participation.

Please complete the survey



trace files and Wireshark profiles are here:

https://www.ikeriri.ne.jp/sharkfest/wifi7_sf24eu.zip



ikeriri network service

<http://www.ikeriri.ne.jp>