# WLAN Pi PCAP Capture Guide

Field reference for 802.11 frame capture, verification, and analysis

WLAN Pi OS 3.x · Wi-Fi 7 / 802.11be · macOS + Wireshark · 16 Sections shankarwifi.com | May 2026

| Audience | Scope | Validated on |
|---|---|---|
| Wi-Fi engineers using a WLAN Pi (R4 / Pro / M4 / Go) with a Mac analysis workstation. | First-boot setup, channel selection, monitor mode, verification, transfer, Wireshark, Wi-Fi 7 EHT capture, iOS workflow. | WLAN Pi OS 3.x, built-in Wi-Fi adapter, USB-C to macOS. Wi-Fi 7 section tested kernel 6.18. |

## TABLE OF CONTENTS

## 16. Working Notes

## 1. Understand the Hardware

Different WLAN Pi models have different radio capabilities. Capture quality depends entirely on what your radio can see. Confirm your model before starting.

| Model | Radio | Can Capture | 6 GHz |
|---|---|---|---|
| WLAN Pi R4 | MT7612U (USB) | Wi-Fi 5 (802.11ac) -- 2.4 / 5 GHz | No |
| WLAN Pi R4 + A9000 | MT7612U + MT7925U (USB) | Wi-Fi 7 (802.11be) -- 2.4 / 5 / 6 GHz | Yes |
| WLAN Pi Pro (EoS) | 2x internal Wi-Fi 6 | Wi-Fi 6 (802.11ax) -- 2.4 / 5 / 6 GHz | Yes |
| WLAN Pi M4 | MT7921K / MT7922 | Wi-Fi 6E (802.11ax) | Yes |
| WLAN Pi Go | Intel BE200 (native) | Wi-Fi 7 (802.11be) -- 2.4 / 5 / 6 GHz, up to 320 MHz | Yes |

**■ TIP**

If you plan to capture HE (Wi-Fi 6) or EHT (Wi-Fi 7) frames, your radio must support that standard. An MT7612U cannot decode HE rate info even if it sees the frames. See Section 15 for Wi-Fi 7 upgrade options.

## 2. What You Need

- WLAN Pi (any supported model) with WLAN Pi OS 3.x flashed and booted
- USB-C cable (Pi to Mac) OR Ethernet cable (Pi to LAN)
- macOS with Wireshark installed -- brew install --cask wireshark
- Terminal app (default macOS Terminal or iTerm2)
- Optional: Airtool 2 from Intuitibits ($30, 3-day free trial) -- multi-channel captures and one-click Wireshark integration
- Optional (WLAN Pi Go users): Airtool Pi iOS app -- capture directly from iPhone/iPad without a laptop

# 3. Connect to the WLAN Pi

Three ways to reach the Pi. Pick whichever fits your setup.

## 3.1 USB-C (recommended for desk work)

Plug the Pi into your Mac with a USB-C cable. The Pi exposes itself as a virtual network interface.

```
Pi IP address (USB-C): 198.18.42.1
```

## 3.2 Ethernet (best for stability)

Connect Pi to your LAN. Find the IP from:

- Pi front panel LCD: Menu -> Network -> Ethernet
- Router's DHCP lease table (look for hostname starting with 'wlanpi')
- Run arp -a on your Mac and look for the Pi's MAC OUI

## 3.3 SSH from Your Mac

```
ssh wlanpi@198.18.42.1
```

Default password: wlanpi (or Wlanpi! on newer images). Nothing appears as you type -- that's normal SSH behavior.

**■ HEADS UP**

Watch your shell prompt. yourname@MacBook-Pro means you're on your Mac. wlanpi@wlanpi-xxx means you're SSH'd into the Pi. Most capture failures are commands run on the wrong machine.

---

# 4. Verify the Capture Interface

Modern WLAN Pi OS pre-creates a dedicated monitor-mode interface called wlanpi0. Confirm it is there:

```
iw dev
```

Expected output:

```
phy#0
    Interface wlanpi0
        ifindex 6
```

```
        type monitor
        channel 1 (2412 MHz), width: 20 MHz
    Interface wlan0
        type managed
```

**wlanpi0** is your capture interface -- stays in monitor mode permanently. **wlan0** is the management interface -- leave it alone.

**■ TIP**

On older WLAN Pi OS versions, wlanpi0 may not exist. In that case put wlan0 into monitor mode manually with airmon-ng + iw. Stick to modern OS for the simpler workflow.

---

## 5. Set the Channel and Width

| Use Case | Command |
|---|---|
| 2.4 GHz, ch 6, 20 MHz | `sudo iw dev wlanpi0 set channel 6 HT20` |
| 5 GHz, ch 36, 80 MHz | `sudo iw dev wlanpi0 set channel 36 80MHz` |
| 5 GHz, ch 149, 80 MHz | `sudo iw dev wlanpi0 set channel 149 80MHz` |
| 6 GHz, ch 37, 80 MHz | `sudo iw dev wlanpi0 set channel 37 80MHz` |

### Width Syntax Cheat Sheet

| Spec | Meaning |
|---|---|
| `HT20` | 20 MHz |
| `HT40+` | 40 MHz, secondary channel above primary |
| `HT40-` | 40 MHz, secondary channel below primary |
| `80MHz` | 80 MHz VHT / HE / EHT |
| `160MHz` | 160 MHz (radio support varies) |
| `320MHz` | 320 MHz EHT -- WLAN Pi Go (BE200) only |

### Verify the Channel Took

```
iw dev wlanpi0 info
```

Confirm: correct channel, width, and type monitor. Always verify before capturing -- silent channel changes are the #1 cause of empty PCAPs.

■ HEADS UP

If you get 'command failed: Invalid argument (-22)', the channel + width combination is invalid. Drop to HT20 to confirm the radio works, then widen.

---

## 6. Run the Capture

ON THE PI

```
mkdir -p ~/captures
sudo tcpdump -i wlanpi0 -nn -s 0 \
  -w ~/captures/test_ch36_$(date +%Y%m%d_%H%M%S).pcap
```

You should see **IEEE802_11_RADIO** in the output -- this confirms radiotap is being captured (RSSI, MCS, PHY info included).

| Flag | Meaning |
|---|---|
| `-i wlanpi0` | Capture interface |
| `-nn` | Skip name resolution (faster) |
| `-s 0` | Capture full frames (no truncation) |
| `-w` | Write to file (no on-screen output) |
| `-U` | Packet-buffered output -- useful for live streaming to Wireshark |

Stop with **Ctrl+C**. Zero dropped packets is what you want. If "dropped by kernel" is non-zero, narrow the channel width or shorten the capture.

### Management-Only Capture (disk-saving)

If you only need handshakes and management frames, exclude data frames entirely. Reduces file size by 80-95% on busy channels:

MANAGEMENT + CONTROL FRAMES ONLY

```
sudo tcpdump -i wlanpi0 -nn -s 0 \
  -w ~/captures/mgmt_$(date +%Y%m%d_%H%M%S).pcap \
  type mgt or type ctl
```

### Long Unattended Captures

```
sudo tcpdump -i wlanpi0 -nn -s 0 -C 100 -W 20 \
    -w ~/captures/longrun_$(date +%Y%m%d_%H%M%S).pcap
```

-C 100 rotates every 100 MB. -W 20 keeps 20 files (2 GB ring buffer).

---

## 7. Verify the Capture

```
ls -lh ~/captures/
capinfos ~/captures/test_ch36_*.pcap
```

Confirm: encapsulation = IEEE 802.11 plus radiotap, packet count > 0 matching tcpdump summary, capture duration matches runtime, strict time order = True.

### Frame-Type Sanity Check

```
tshark -r ~/captures/test_ch36_*.pcap -q -z io,phs | head -30
```

Shows protocol hierarchy. If you see no 802.11 frames at all, monitor mode is not working. If only beacons and no data, the target AP/client may not be active.

---

## 8. Transfer the PCAP to Your Mac

**■ HEADS UP**

scp runs on your Mac, not on the Pi. Type 'exit' to drop back to your Mac prompt before continuing.

ON THE PI -- EXIT FIRST

```
exit
```

ON YOUR MAC

```
mkdir -p ~/Documents/PCAPs
scp wlanpi@198.18.42.1:/home/wlanpi/captures/*.pcap ~/Documents/PCAPs/
```

## 9. Wireshark Filters

| Filter | What It Shows |
|---|---|
| `wlan.fc.type_subtype == 8` | Beacon frames only |
| `wlan.fc.type_subtype == 4` | Probe requests |
| `wlan.fc.type_subtype == 5` | Probe responses |
| `wlan.fc.type_subtype == 0` | Association requests |
| `wlan.fc.type_subtype == 11` | Authentication frames |
| `eapol` | 4-way handshake frames |
| `wlan_radio.phy == 12` | EHT (802.11be / Wi-Fi 7) frames |
| `wlan_radio.phy == 11` | HE (802.11ax / Wi-Fi 6/6E) frames |
| `wlan_radio.phy == 9` | VHT (802.11ac / Wi-Fi 5) frames |
| `radiotap.dbm_antsignal < -75` | Weak RSSI frames (< -75 dBm) |
| `wlan.fc.retry == 1` | Retransmitted frames |
| `wlan.fc.protected == 0 && wlan.fc.type == 0` | Unprotected management frames (PMF check) |
| `awdl` | Apple Wireless Direct Link traffic |
| `wlan.tag.number == 255 && wlan.ext_tag.number == 107` | Multi-Link Element (MLO IE) |

## 10. Cleanup (Optional)

Only needed if you flipped wlan0 (not wlanpi0) into monitor mode manually:

```
sudo ip link set wlan0 down
sudo iw dev wlan0 set type managed
sudo ip link set wlan0 up
sudo systemctl restart NetworkManager
```

If you only used wlanpi0, no cleanup needed -- wlan0 was never touched.

---

## 11. Common Gotchas

| Symptom | Cause / Fix |
|---|---|
| zsh: parse error near '<' | Placeholders like typed literally. Replace with real values: 'set channel 36'. |
| bash: open: command not found | open is macOS only. Type 'exit' to return to your Mac first. |
| Empty PCAP after long run | Radio dropped to managed silently. Add iw dev wlanpi0 info check at start of capture script. |
| File grows but no beacons | Wrong channel or no APs nearby. Verify with iw dev wlanpi0 info and tshark hierarchy. |
| 'Operation not supported' on channel set | Width invalid for that channel. Try HT20 first, then widen. |
| 'Device or resource busy' | NetworkManager holding the adapter. Run sudo airmon-ng check kill. |
| Decryption fails in Wireshark | Missed the 4-way handshake. Start capture BEFORE client associates to the SSID. |
| Frames present but no HE/EHT info | Radio doesn't support that standard. MT7612U is 11ac only. See Section 15. |
| Disk full mid-capture | Check df -h / before long captures. Use -C and -W flags for ring-buffer rotation. |
| capinfos shows 0 packets | tcpdump was stopped too quickly or the channel was wrong. Re-verify iw dev wlanpi0 info before next run. |

---

## 12. Quick Reference Card

MAC -- START

```
ssh wlanpi@198.18.42.1
```

PI -- CAPTURE

```
sudo iw dev wlanpi0 set channel 36 HT20
iw dev wlanpi0 info                              # verify
mkdir -p ~/captures
sudo tcpdump -i wlanpi0 -nn -s 0 \
    -w ~/captures/label_$(date +%Y%m%d_%H%M%S).pcap
# Ctrl+C to stop
capinfos ~/captures/label_*.pcap                 # verify
exit
```

MAC -- TRANSFER & OPEN

```
scp wlanpi@198.18.42.1:/home/wlanpi/captures/label_*.pcap ~/Documents/PCAPs/
open -a Wireshark ~/Documents/PCAPs/label_*.pcap
```

**TIP**

Replace 'label' with something descriptive: kitchen_ch36, mlo_test, roaming_lab. Replace '36' and 'HT20' with your target channel and width.

# 13. Beyond Basic Capture

## 13.1 Wireshark Live Remote (sshdump)

Stream frames live to your Mac instead of writing to a file on the Pi.

- Open Wireshark, scroll interface list to SSH remote capture: sshdump
- Click the gear icon, set Server = 198.18.42.1, port 22
- Authentication tab: username wlanpi, password wlanpi
- Capture tab: paste remote command (one line):

```
sudo iw dev wlanpi0 set channel 36; sudo tcpdump -i wlanpi0 -U -w -
```

Click Start. Frames stream live to Wireshark. Apply display filters in real-time.

## 13.2 Airtool 2 (macOS, multi-channel)

Intuitibits Airtool 2 ($30) -- macOS menu-bar app, auto-discovers WLAN Pi sensors on LAN, multi-source capture across multiple Pis into one merged pcapng with per-frame sensor annotation. Works as remote sensor for both WLAN Pi R4 and WLAN Pi Go.

## 13.3 Airtool Pi (iOS/iPadOS) -- NEW

**Airtool Pi** is a separate iOS app (free download, Intuitibits) that turns your iPhone or iPad into a packet capture controller using a WLAN Pi Go as the radio sensor. No Mac or laptop needed.

- Connect WLAN Pi Go to iPhone via USB-C
- Open Airtool Pi -- auto-discovers the Go in seconds
- Select interface, channel (2.4 / 5 / 6 GHz), channel width (20/40/80/160/320 MHz)
- Tap Start -- capture runs on the Go, PCAP saved to iPhone
- Share PCAP via AirDrop to Mac for Wireshark analysis
- Multi-channel capture: up to 4 adapters on one sensor, channel hopping supported
- Output format: pcapng (preserves per-packet metadata)

**■ AIRTOOL PI VS AIRTOOL 2**

Airtool Pi = iOS app (iPhone/iPad) for mobile/field use with WLAN Pi Go. Airtool 2 = macOS menu-bar app for desk/lab use with any WLAN Pi model. Both are from Intuitibits. Pick the one that matches your workflow.

### 13.4 WiFi Explorer Pi (iOS Scanning)

**WiFi Explorer Pi** is an iOS companion app for WLAN Pi Go that provides real-time Wi-Fi scanning and analysis -- AP inventory, RSSI, channel utilization, security mode, and more. Scan without a Mac.

### 13.5 WLAN Pi Profiler

Built-in tool that builds a synthetic AP and profiles client capabilities.

```
sudo profiler -c 36               # 5 GHz channel 36
sudo profiler --listen-only -c 36 # passive mode
sudo profiler --pcap capture.pcap # analyze existing capture
```

## 14. Multi-Band Capture (2.4 / 5 / 6 GHz)

The WLAN Pi R4 has one radio -- one channel at a time without an additional USB adapter.

| You Want | Best Method | Trade-off |
|---|---|---|
| All bands simultaneously (parallel) | R4 + A9000 USB adapter, or two WLAN Pis + Airtool 2 | Needs second radio or paid tool |
| All bands sequentially | Scripted hop + capture loop | Best for test corpus building |
| Mobile capture (no laptop) | WLAN Pi Go + Airtool Pi on iPhone | Requires WLAN Pi Go |
| Device discovery survey | Kismet (auto channel hopping) | Misses most data frames |

### 14.1 Sequential Captures

```
# 2.4 GHz
sudo iw dev wlanpi0 set channel 6 HT20
sudo tcpdump -i wlanpi0 -nn -s 0 -w ~/captures/24ghz_$(date +%H%M%S).pcap
# Ctrl+C after 60s

# 5 GHz UNII-1
sudo iw dev wlanpi0 set channel 36 80MHz
sudo tcpdump -i wlanpi0 -nn -s 0 -w ~/captures/5ghz_low_$(date +%H%M%S).pcap

# 5 GHz UNII-3
sudo iw dev wlanpi0 set channel 149 80MHz
sudo tcpdump -i wlanpi0 -nn -s 0 -w ~/captures/5ghz_high_$(date +%H%M%S).pcap
```

## 14.2 Scripted Multi-Band Capture

SAVE AS ~/CAPTURES/MULTIBAND_CAPTURE.SH

```
#!/bin/bash
# Multi-band capture loop for WLAN Pi
DUR=60   # seconds per channel

CHANNELS=(
  "6 HT20"      # 2.4 GHz
  "36 HT20"     # 5 GHz UNII-1
  "149 HT20"    # 5 GHz UNII-3
)

mkdir -p ~/captures
TS=$(date +%Y%m%d_%H%M%S)

for ch_spec in "${CHANNELS[@]}"; do
  ch=$(echo $ch_spec | awk '{print $1}')
  width=$(echo $ch_spec | awk '{print $2}')
  echo "--- Capturing channel $ch ($width) for $DUR seconds..."
  sudo iw dev wlanpi0 set channel $ch $width
  sudo iw dev wlanpi0 info | grep channel
  sudo timeout $DUR tcpdump -i wlanpi0 -nn -s 0 \
    -w ~/captures/multiband_${TS}_ch${ch}.pcap
done

echo "--- Done:"
ls -lh ~/captures/multiband_${TS}_*.pcap
```

```
chmod +x ~/captures/multiband_capture.sh && ~/captures/multiband_capture.sh
```

**■ TIP**

Add 6 GHz channels (e.g., "37 80MHz") to the CHANNELS array if your radio supports it. The R4 MT7612U is 2.4/5 GHz only -- see Section 15 for 6 GHz options.

## 14.3 Transfer All Files

```
scp wlanpi@198.18.42.1:/home/wlanpi/captures/multiband_*.pcap ~/Documents/PCAPs/
```

## 14.4 Merge Files (macOS)

```
mergecap -w combined.pcapng 5ghz_ch149.pcap 6ghz_ch37.pcap
```

## 14.5 Kismet (Survey Mode)

```
sudo kismet -c wlanpi0
```

Browse to http://198.18.42.1:2501/ on your Mac. Kismet hops channels automatically. Good for device discovery. Poor for protocol analysis -- misses ~90% of traffic on any single channel.

# 15. Wi-Fi 7 (802.11be) Capture

■ NEW SECTION

Two paths to 802.11be EHT frame capture: adding a Netgear A9000 (MT7925U) to an existing WLAN Pi R4, or using a WLAN Pi Go with its native Intel BE200.

To capture Wi-Fi 7 EHT frames -- TWT, OFDMA trigger frames, MLO IE, BSS Color, 6 GHz operation -- you need a radio that supports 802.11be. The R4 built-in MT7612U is 802.11ac only.

## 15.1 WLAN Pi R4 + Netgear A9000 (MT7925U)

The **Netgear Nighthawk A9000** is a $89 USB 3.0 Wi-Fi 7 adapter based on the MediaTek MT7925U chipset. Plugged into a WLAN Pi R4, it adds a second independent Wi-Fi 7 radio alongside the existing MT7612U.

| Spec | Value |
|---|---|
| Chipset | MediaTek MT7925U |
| USB VID/PID | 0846:9072 |
| Bands | 2.4 / 5 / 6 GHz |
| Max channel width | 160 MHz (no 320 MHz) |
| MIMO | 2x2:2 |
| Linux driver | mt7925u (kernel >= 6.7; plug-and-play on >= 6.17) |
| Monitor mode | Yes -- stable on kernel 6.18 (tested Dec 2025) |
| MLO as client (STR) | No -- single-band association only |

### Prerequisites: wlanpi-kernel

The R4 must run the custom wlanpi-kernel (kernel >= 6.12 LTS). Follow Bill Bushong's guide:

```
cd /usr/src
sudo git clone https://github.com/WLAN-Pi/wlanpi-kernel.git
cd wlanpi-kernel
```

```
sudo ./build-kernel.sh       # takes several hours
sudo dpkg -i output/wlanpi-kernel-bookworm-v8_*.deb
sudo reboot
```

### Verify A9000 is Detected

```
lsusb | grep 0846:9072        # confirms USB detection
modinfo mt7925u                # confirms driver is loaded
iw dev                         # shows new interface (wlan1 or similar)
```

### Set Monitor Mode on A9000

```
sudo ip link set wlan1 down
sudo iw dev wlan1 set type monitor
sudo ip link set wlan1 up
iw dev wlan1 info              # verify type monitor
```

### Capture 802.11be on 6 GHz

```
sudo iw dev wlan1 set channel 37 80MHz
iw dev wlan1 info              # verify
sudo tcpdump -i wlan1 -nn -s 0 \
  -w ~/captures/wifi7_6ghz_$(date +%Y%m%d_%H%M%S).pcap
```

Confirm EHT frames in Wireshark: **wlan_radio.phy == 12** (PHY type 12 = EHT / 802.11be).

**■ A9000 LIMITATIONS**

Maximum channel width is 160 MHz -- 320 MHz EHT not supported. Cannot negotiate STR MLO as a client -- single-band association only. For MLO client validation use Intel BE200/BE210 or a native Wi-Fi 7 laptop.

## 15.2 WLAN Pi Go -- Intel BE200 (Native Wi-Fi 7)

The WLAN Pi Go ships with an **Intel BE200** -- the same chip in M4 MacBook Pro and current-generation Wi-Fi 7 laptops. No additional hardware or kernel compilation required. The Go also supports USB-C connection to an iPhone (powered by the phone) and captures via Airtool Pi.

| Spec | Value |
|---|---|
| Chipset | Intel BE200 |
| Bands | 2.4 / 5 / 6 GHz |
| Max channel width | 320 MHz (full EHT) |

| Spec | Value |
|---|---|
| Linux driver | iwlwifi (in-kernel, no patches needed) |
| Monitor mode | Yes |
| MLO as client | EMLSR (enhanced multi-link, single radio) |
| Mobile workflow | Airtool Pi (iOS) via USB-C -- no laptop needed |
| Spectrum analysis | Wi-Spy Lucid clip-on (accessory port) |

### Capture 802.11be on 6 GHz (WLAN Pi Go)

WLAN PI GO -- VIA SSH FROM MAC

```
# wlanpi0 is already in monitor mode on the Go
sudo iw dev wlanpi0 set channel 37 80MHz
iw dev wlanpi0 info            # verify
sudo tcpdump -i wlanpi0 -nn -s 0 \
  -w ~/captures/go_6ghz_$(date +%Y%m%d_%H%M%S).pcap
```

### Capture via Airtool Pi (iPhone -- no Mac needed)

- Connect WLAN Pi Go to iPhone via USB-C
- Open Airtool Pi on iPhone -- auto-discovers Go
- Select band, channel, width (up to 320 MHz supported)
- Tap Start Capture
- Share PCAP from iPhone to Mac via AirDrop for Wireshark

**■ BE200 VS A9000 FOR CAPTURE**

BE200 supports 320 MHz channels and has a more mature Linux driver (iwlwifi vs mt7925u). For protocol analysis and PCAP corpus building, the Go is the better platform. The A9000 is the right choice only if you need to upgrade an R4 you already own.

## 15.3 Dual-Radio Setup -- R4 Built-in + A9000

With R4 built-in MT7612U and A9000 plugged in, capture two independent channels simultaneously -- lowest-cost path to parallel dual-band capture without buying a second Pi.

TWO TERMINALS RUNNING SIMULTANEOUSLY ON THE PI

```
# Terminal 1 -- built-in MT7612U on 5 GHz (wlanpi0)
sudo iw dev wlanpi0 set channel 149 80MHz
sudo tcpdump -i wlanpi0 -nn -s 0 \
  -w ~/captures/dual_5ghz_ch149_$(date +%Y%m%d_%H%M%S).pcap

# Terminal 2 -- A9000 on 6 GHz (wlan1)
sudo iw dev wlan1 set channel 37 80MHz
sudo tcpdump -i wlan1 -nn -s 0 \
  -w ~/captures/dual_6ghz_ch37_$(date +%Y%m%d_%H%M%S).pcap
```

MAC -- MERGE BOTH FILES

```
mergecap -w combined_dual.pcapng \
  dual_5ghz_ch149_*.pcap \
  dual_6ghz_ch37_*.pcap
```

**■ TIME ALIGNMENT**

Both interfaces share the Pi system clock, so NTP sync is sufficient for accurate frame timestamp correlation. No additional time source needed for single-Pi dual-radio setup.

# 16. Working Notes

## 16.1 Capture Density

On a busy 5 GHz channel (149 with active mesh + AWDL), expect 3,000-5,000 packets/sec. A 45-second capture can easily exceed 100,000 frames and 70 MB. Plan accordingly if your analysis tool has frame-count or file-size caps.

## 16.2 Channel Width vs. Fidelity

For protocol analysis (RSN IEs, EAPOL handshakes, beacon parsing), 20 MHz is almost always sufficient and gives the cleanest captures. Use 80/160 MHz only when you specifically need PHY-level detail across the bonded channel.

## 16.3 Capture Before Associating

If you want decrypted data frames in Wireshark, you must capture the 4-way handshake. Start the capture **before** the client associates to the SSID. Provide the PSK in Wireshark preferences: Edit -> Preferences -> Protocols -> IEEE 802.11 -> Decryption keys.

## 16.4 pcapng vs. pcap

Modern Wireshark prefers pcapng (Next Generation) -- preserves per-packet metadata, multi-interface annotation, and capture comments. For most workflows pcap is fine. Switch to pcapng with dumpcap instead of tcpdump when you need those features. Airtool Pi automatically uses pcapng.

## 16.5 Wi-Fi 7 Capture Notes

- EHT filter: wlan_radio.phy == 12 in Wireshark confirms 802.11be frames. PHY type 11 = HE (Wi-Fi 6/6E).
- MLO IE: IE ID 255, extension ID 107 in beacon/probe frames -- Multi-Link element carrying affiliated link MACs.
- TWT frames: Action frames subtype 0x0D. Filter: wlan.fixed.action_code == 26.
- BSS Color: Inside HE Operation IE. Filter: wlan.ext_tag.he_bss_color.
- 320 MHz channels: Only capturable with Intel BE200 (WLAN Pi Go). MT7925U (A9000) max is 160 MHz.
- PMF protected management: wlan.fc.protected == 0 && wlan.fc.type == 0 to find unprotected management frames.

## 16.6 Recommended Tool Stack by Workflow

| Goal | Recommended Tool |
|---|---|
| Protocol analysis at desk | WLAN Pi (any) + tcpdump + Wireshark (macOS) |
| Quick single-band capture | ssh + tcpdump (Section 6) |
| Build multi-band PCAP corpus | Sequential script (Section 14.2) |
| MLO time-aligned capture | Two radios (R4+A9000 or R4+Go) + mergecap |
| Mobile capture (field, no laptop) | WLAN Pi Go + Airtool Pi (iPhone) |
| Real-time live analysis | sshdump in Wireshark (Section 13.1) |
| Multi-source parallel capture | Airtool 2 or Airtool Pi multi-channel mode |
| Site survey / device discovery | Kismet (Section 14.5) |
| Client capability profiling | WLAN Pi Profiler (Section 13.5) |

v1.1 -- May 2026. Original v1.0 validated on WLAN Pi OS 3.x, USB-C to macOS. Section 15 (Wi-Fi 7) added May 2026 based on MT7925U kernel 6.18 testing and Intel BE200 field experience. Airtool Pi / WiFi Explorer Pi / WLAN Pi Go mobile workflow added May 2026. Part of the 802.11 Protocol Library at shankarwifi.com.