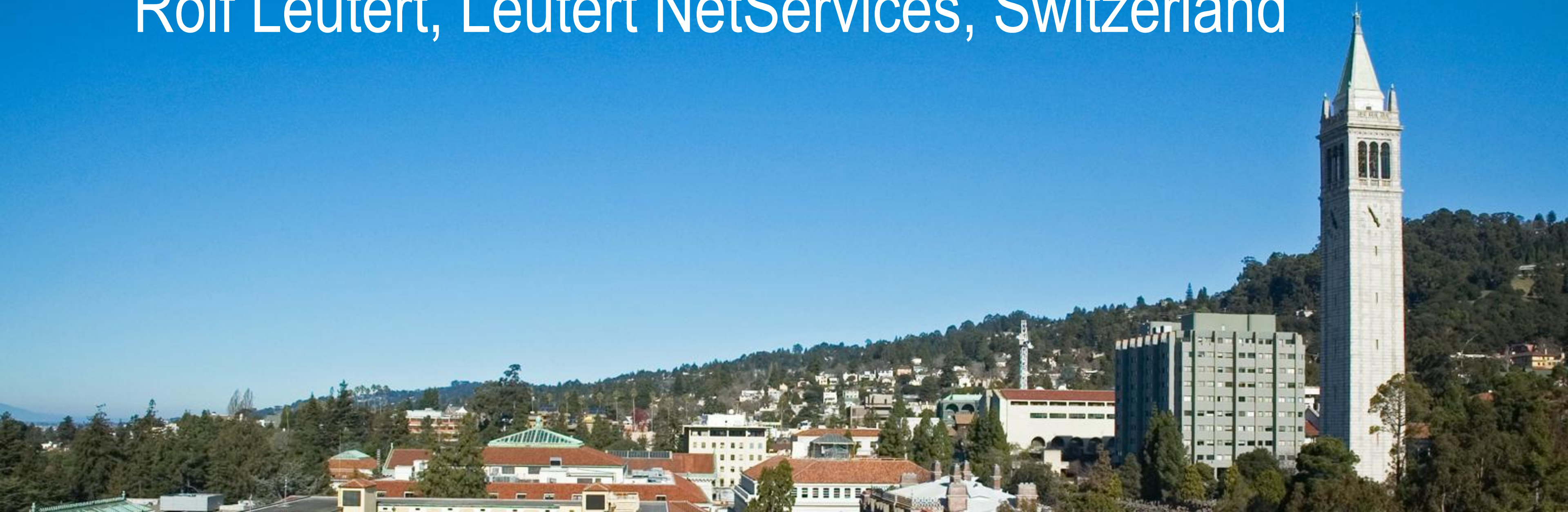
SHARKFEST '13

Wireshark Developer and User Conference

PA-12 WLAN Troubleshooting with Wireshark and AirPcap

Rolf Leutert, Leutert NetServices, Switzerland

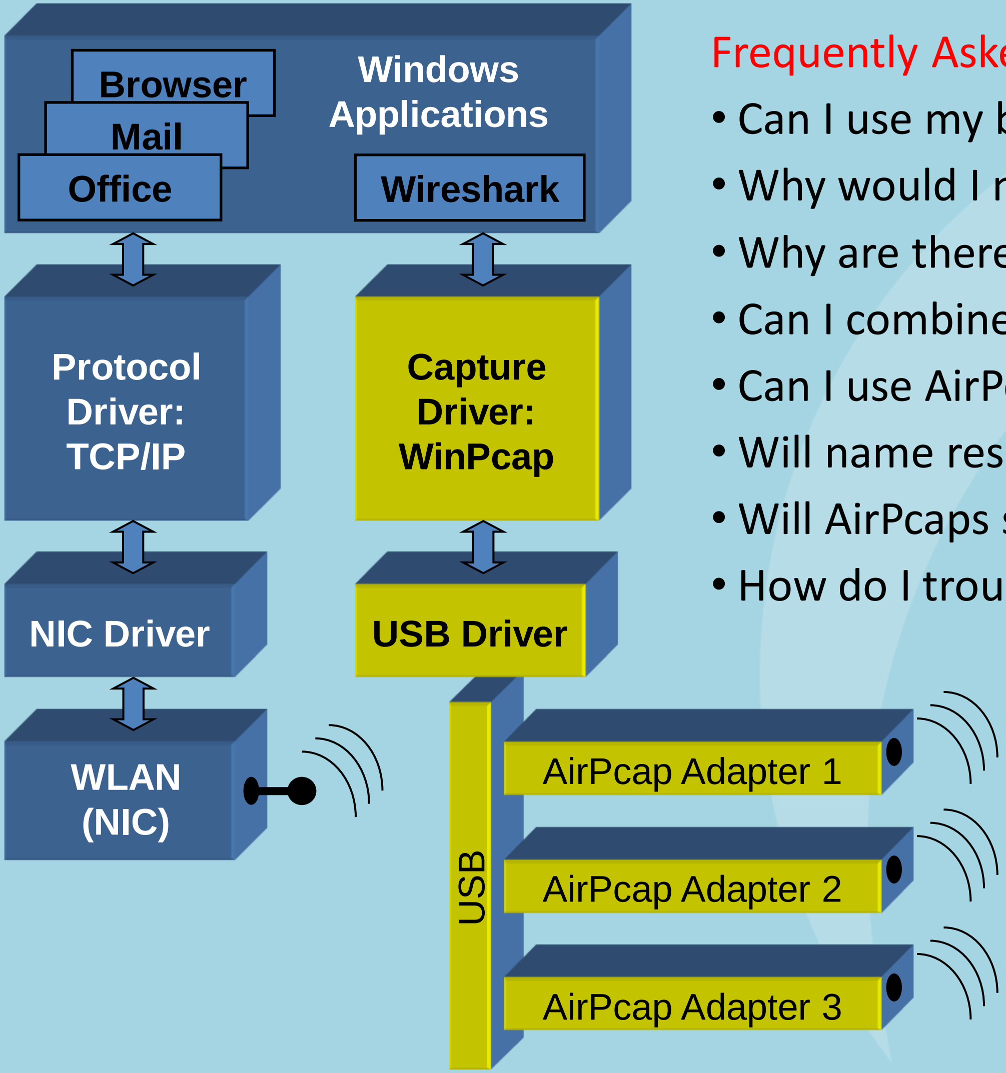


WLAN Troubleshooting with Wireshark and AirPcap

Troubleshooting WLAN problems can be a very challenging task. The wireless media is known to be unreliable. Signal interferences, low signal areas or overloaded cells are just a few of possible issues.
In addition, the compatibility between the different IEEE standards and the vendor's way of implementation is not always granted.

Having so many factors potentially impacting the performance of a wireless LAN, a systematic root-cause analysis will be more promising than the trial and error method.

WLAN Troubleshooting with Wireshark and AirPcap

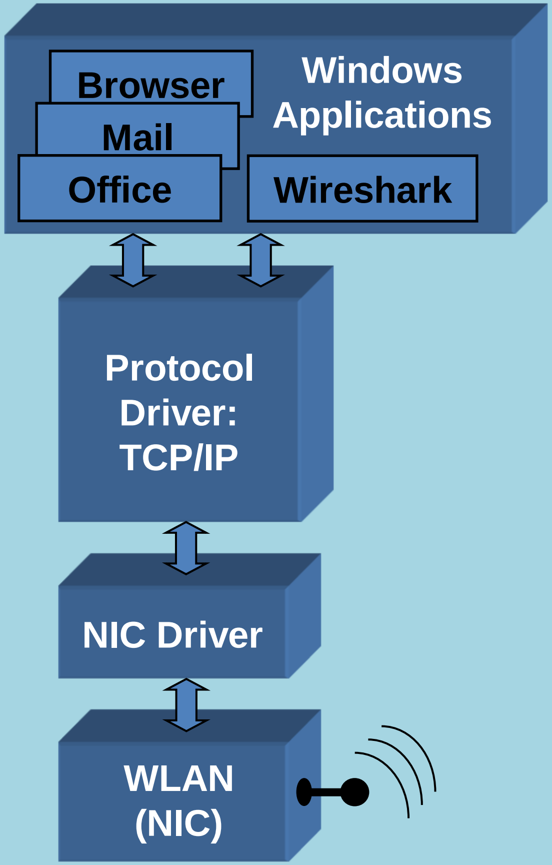


Frequently Asked Questions:

- Can I use my built-in WLAN NIC with Wireshark?
- Why would I need AirPcaps to analyze WLAN?
- Why are there different types of AirPcaps?
- Can I combine different types of AirPcaps?
- Can I use AirPcaps to join a WLAN?
- Will name resolution work with AirPcaps?
- Will AirPcaps show me Radio Interferences?
- How do I troubleshoot encrypted WLANs?

WLAN Troubleshooting with Wireshark and AirPcap

Capturing with built-in WLAN card



Frequently Given Answers:

- Yes you can use the built in WLAN NIC with Wireshark!

But with a lot of restrictions:

- No promiscuous mode, only the own traffic visible
- Frames will be displayed in Ethernet format
- No radio information like SNR, channel no, speed etc.
- One channel only, not suitable for roaming analysis

And the biggest limitation:

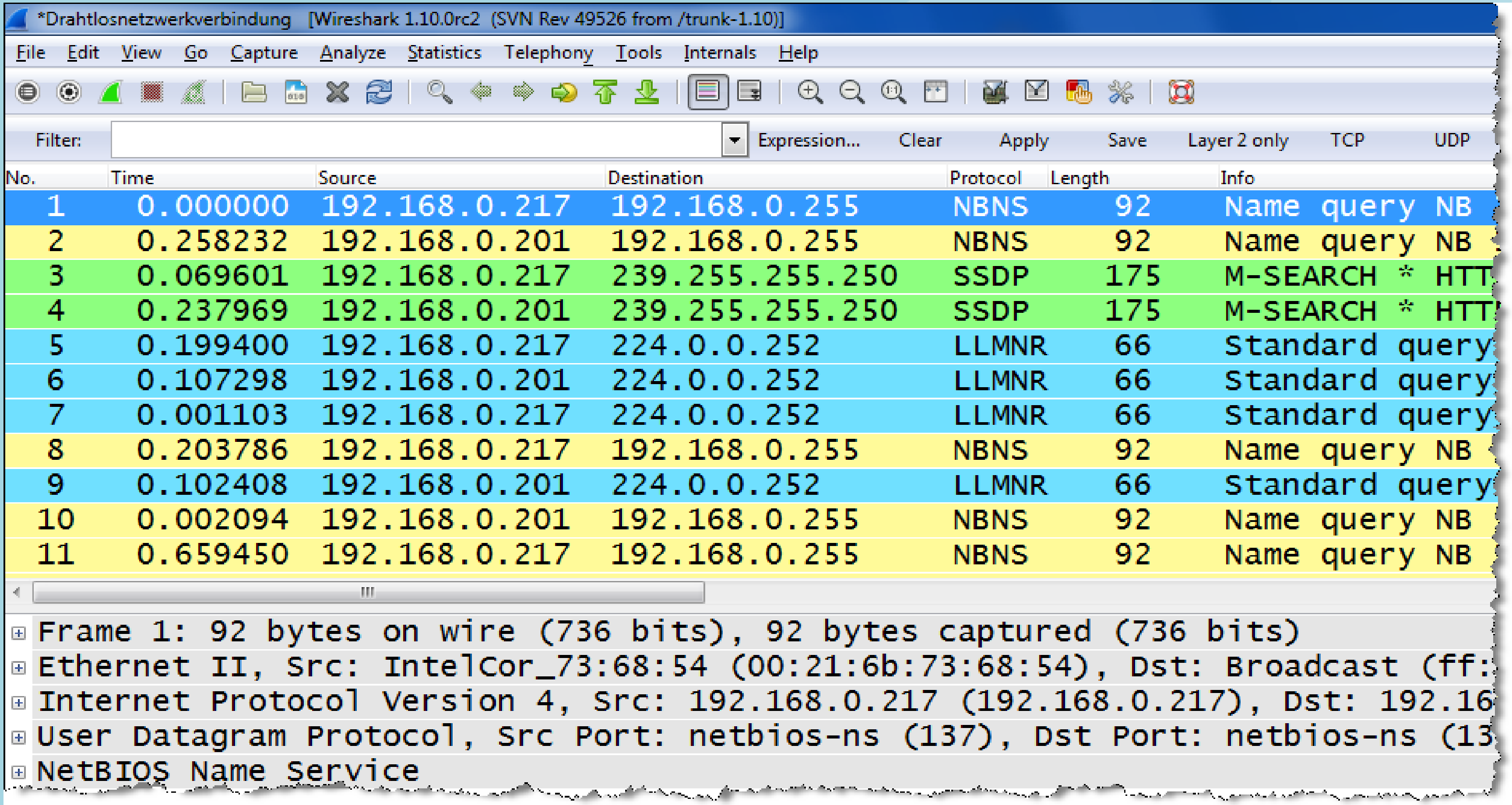
- No management or control frames visible!
- But these are the ones you need for troubleshooting

(Exception: under Linux some NICs support more features)

WLAN Troubleshooting with Wireshark and AirPcap

Capturing with built-in WLAN card

- Capturing on built in WLAN NIC will display Ethernet like frames
- Only Data frames and no Radio or WLAN header will be seen

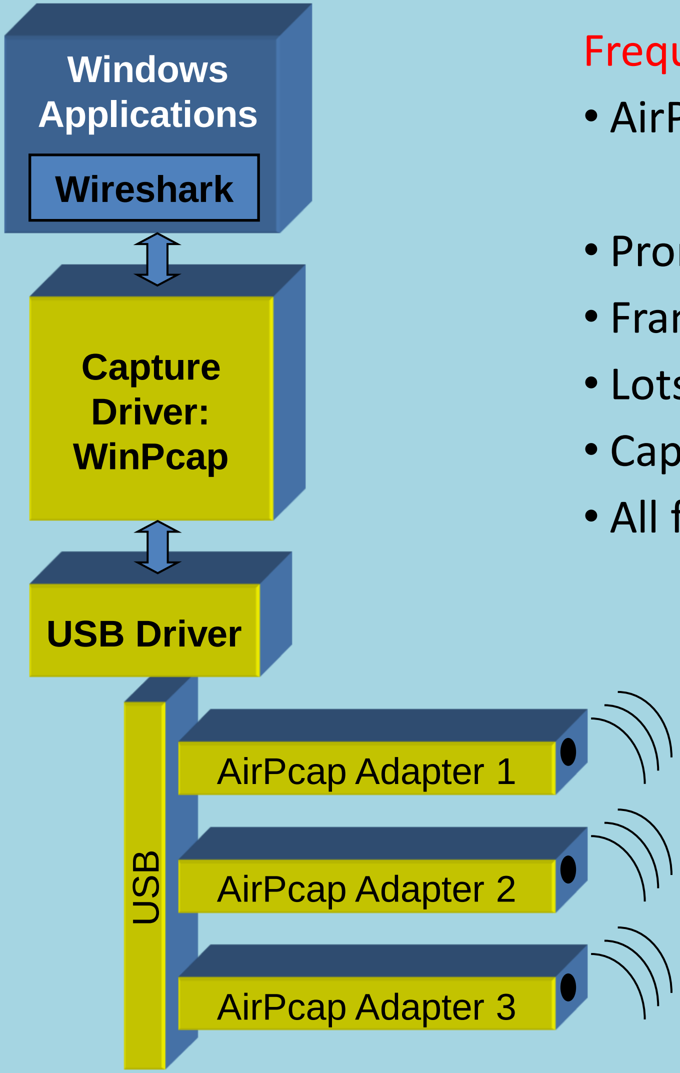


No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.0.217	192.168.0.255	NBNS	92	Name query NB
2	0.258232	192.168.0.201	192.168.0.255	NBNS	92	Name query NB
3	0.069601	192.168.0.217	239.255.255.250	SSDP	175	M-SEARCH * HTT
4	0.237969	192.168.0.201	239.255.255.250	SSDP	175	M-SEARCH * HTT
5	0.199400	192.168.0.217	224.0.0.252	LLMNR	66	Standard query
6	0.107298	192.168.0.201	224.0.0.252	LLMNR	66	Standard query
7	0.001103	192.168.0.217	224.0.0.252	LLMNR	66	Standard query
8	0.203786	192.168.0.217	192.168.0.255	NBNS	92	Name query NB
9	0.102408	192.168.0.201	224.0.0.252	LLMNR	66	Standard query
10	0.002094	192.168.0.201	192.168.0.255	NBNS	92	Name query NB
11	0.659450	192.168.0.217	192.168.0.255	NBNS	92	Name query NB

```
⊞ Frame 1: 92 bytes on wire (736 bits), 92 bytes captured (736 bits)
⊞ Ethernet II, Src: IntelCor_73:68:54 (00:21:6b:73:68:54), Dst: Broadcast (ff:
⊞ Internet Protocol Version 4, Src: 192.168.0.217 (192.168.0.217), Dst: 192.16
⊞ User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (13
⊞ NetBIOS Name Service
```

WLAN Troubleshooting with Wireshark and AirPcap

Capturing with AirPcap Adapters



Frequently Given Answers:

- AirPcaps support the following features:

- Promiscuous mode, all traffic in a radio cell visible
- Frames will be displayed original WLAN format
- Lots of radio information like SNR, channel no, speed etc.
- Capturing in multiple channels with multiple adapters
- All frame types visible (Data, Management and Control)

WLAN Troubleshooting with Wireshark and AirPcap

Different AirPcap Adapters

AirPcap Classic
802.11b/g

AirPcap TX
802.11b/g
+ Frame injection

AirPcap NX
802.11a/b/g/n

Frequently Given Answers:

- Different AirPcaps for different 802.11 standards
- Different features at different costs
- Different AirPcaps can be combined together
- AirPcaps can not join a WLAN, are for capturing only
- Name resolution will not work for above reason
- Radio interferences can not be detected directly with AirPcaps
- Supported by all popular Windows versions up to Win7

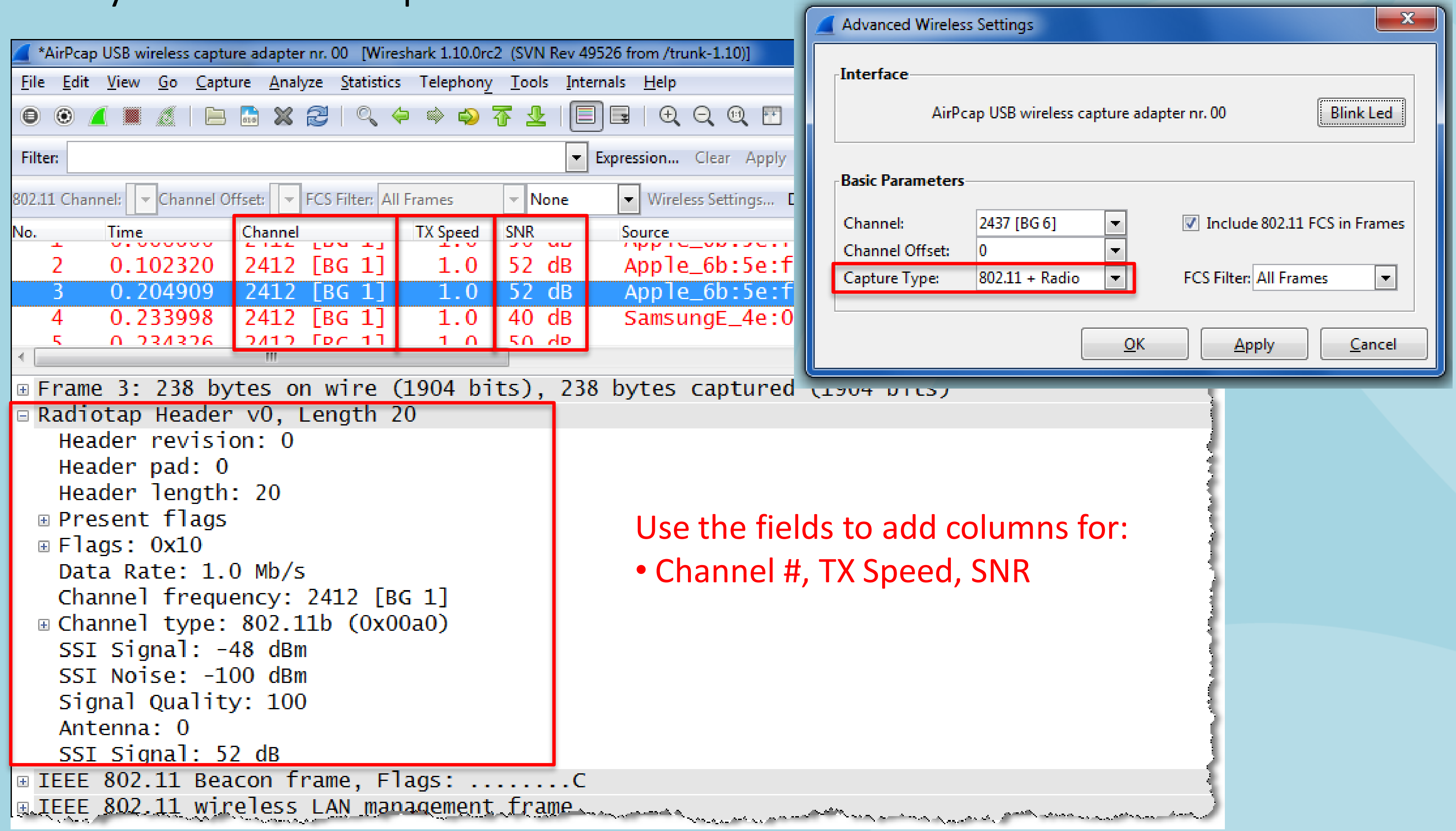
New features within near future:

- 802.11ac standard support
- Win 8 drivers
- USB 3.0 support for NX (Classic and TX today)

WLAN Troubleshooting with Wireshark and AirPcap

Additional Wireshark Columns

- AirPcaps add a Radiotap Header with useful information to each captured frame
- Verify that the Radio option is turned on



*AirPcap USB wireless capture adapter nr. 00 [Wireshark 1.10.0rc2 (SVN Rev 49526 from /trunk-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply
802.11 Channel: Channel Offset: FCS Filter: All Frames None Wireless Settings...

No.	Time	Channel	TX Speed	SNR	Source
2	0.102320	2412 [BG 1]	1.0	52 dB	Apple_6b:5e:f
3	0.204909	2412 [BG 1]	1.0	52 dB	Apple_6b:5e:f
4	0.233998	2412 [BG 1]	1.0	40 dB	SamsungE_4e:0

```
⊞ Frame 3: 238 bytes on wire (1904 bits), 238 bytes captured (1904 bits)
⊟ Radiotap Header v0, Length 20
    Header revision: 0
    Header pad: 0
    Header length: 20
  ⊞ Present flags
  ⊞ Flags: 0x10
    Data Rate: 1.0 Mb/s
    Channel frequency: 2412 [BG 1]
  ⊞ Channel type: 802.11b (0x00a0)
    SSI Signal: -48 dBm
    SSI Noise: -100 dBm
    Signal Quality: 100
    Antenna: 0
    SSI Signal: 52 dB
⊞ IEEE 802.11 Beacon frame, Flags: ........C
⊞ IEEE 802.11 wireless LAN management frame
```

AirPcap USB wireless capture adapter nr. 00

Channel: 2437 [BG 6]
Channel Offset: 0
Capture Type: 802.11 + Radio
☑ Include 802.11 FCS in Frames
FCS Filter: All Frames

Use the fields to add columns for:

- Channel #, TX Speed, SNR

WLAN Troubleshooting with Wireshark and AirPcap

Frame Types Overview

The Management Frames:

- Beacon
- Probe Request & Response
- Authentication & Deauthentication
- Association & Disassociation
- Reassociation Request & Response
- Action

The Control Frames:

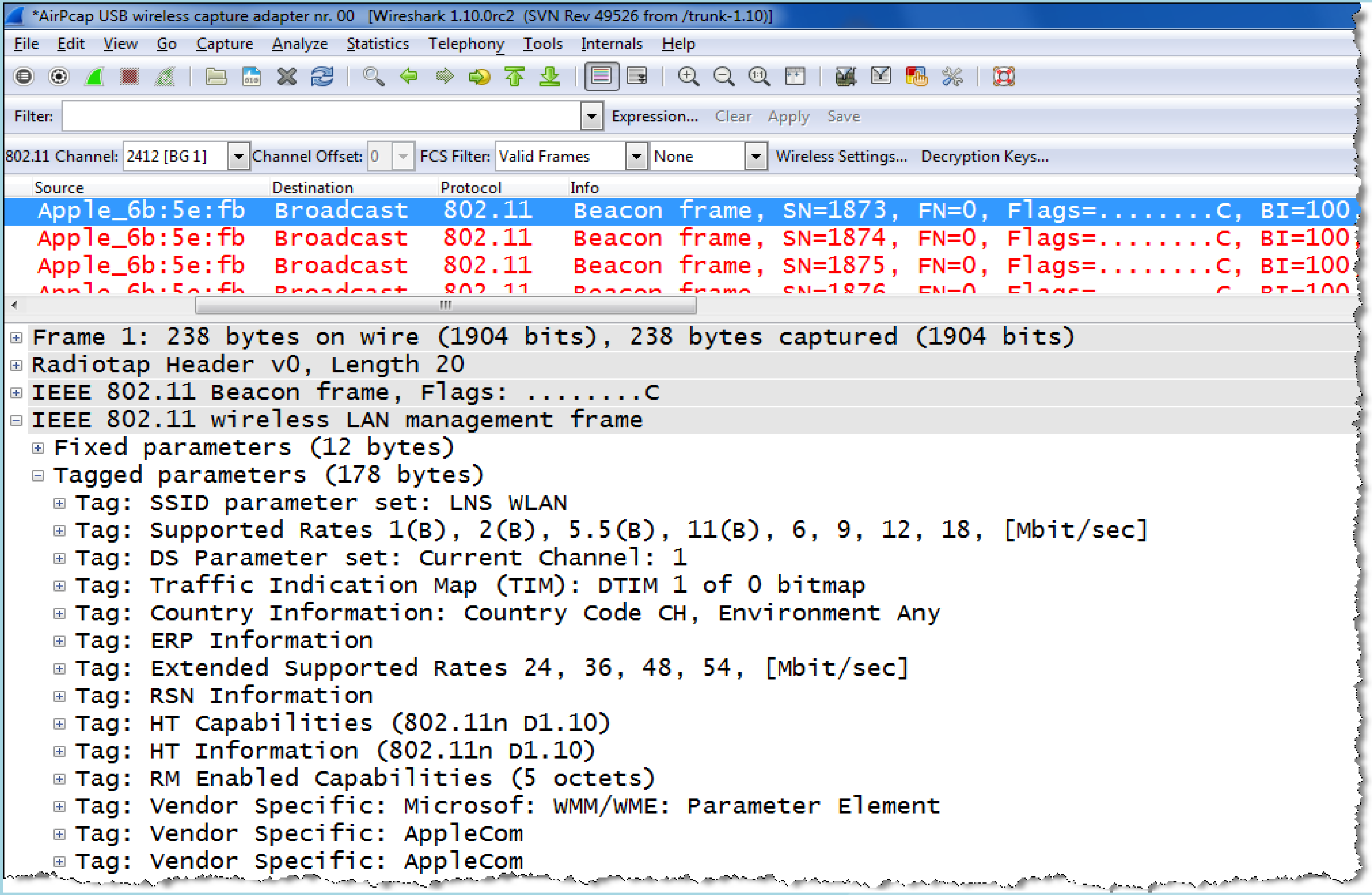
- Request to Send (RTS)
- Clear to Send (CTS)
- Acknowledge / Block Acknowledge Request / Block Acknowledge
- Power Save Poll

The Data Frames:

- Data
- Null Function

WLAN Troubleshooting with Wireshark and AirPcap

Frame Type: Beacon


*AirPcap USB wireless capture adapter nr. 00 [Wireshark 1.10.0rc2 (SVN Rev 49526 from /trunk-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save
Wireless Settings... Decryption Keys...

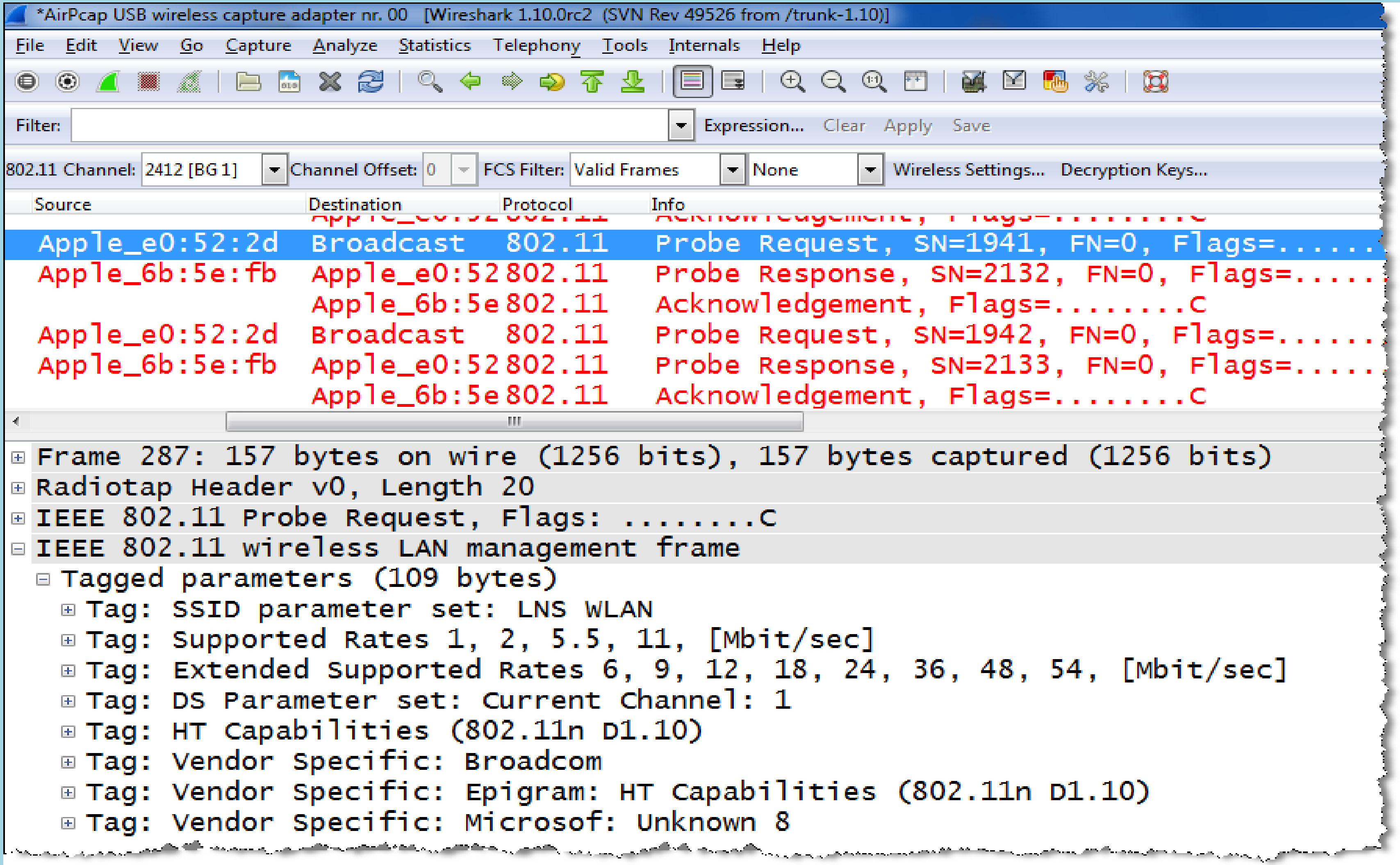
802.11 Channel: 2412 [BG 1]
Channel Offset: 0
FCS Filter: Valid Frames
None

```
Source            Destination   Protocol  Info
Apple_6b:5e:fb    Broadcast     802.11    Beacon frame, SN=1873, FN=0, Flags=........C, BI=100
Apple_6b:5e:fb    Broadcast     802.11    Beacon frame, SN=1874, FN=0, Flags=........C, BI=100
Apple_6b:5e:fb    Broadcast     802.11    Beacon frame, SN=1875, FN=0, Flags=........C, BI=100
Apple_6b:5e:fb    Broadcast     802.11    Beacon frame, SN=1876, FN=0, Flags=........C, BI=100
```

```
⊞ Frame 1: 238 bytes on wire (1904 bits), 238 bytes captured (1904 bits)
⊞ Radiotap Header v0, Length 20
⊞ IEEE 802.11 Beacon frame, Flags: ........C
⊟ IEEE 802.11 wireless LAN management frame
  ⊞ Fixed parameters (12 bytes)
  ⊟ Tagged parameters (178 bytes)
    ⊞ Tag: SSID parameter set: LNS WLAN
    ⊞ Tag: Supported Rates 1(B), 2(B), 5.5(B), 11(B), 6, 9, 12, 18, [Mbit/sec]
    ⊞ Tag: DS Parameter set: Current Channel: 1
    ⊞ Tag: Traffic Indication Map (TIM): DTIM 1 of 0 bitmap
    ⊞ Tag: Country Information: Country Code CH, Environment Any
    ⊞ Tag: ERP Information
    ⊞ Tag: Extended Supported Rates 24, 36, 48, 54, [Mbit/sec]
    ⊞ Tag: RSN Information
    ⊞ Tag: HT Capabilities (802.11n D1.10)
    ⊞ Tag: HT Information (802.11n D1.10)
    ⊞ Tag: RM Enabled Capabilities (5 octets)
    ⊞ Tag: Vendor Specific: Microsof: WMM/WME: Parameter Element
    ⊞ Tag: Vendor Specific: AppleCom
    ⊞ Tag: Vendor Specific: AppleCom
```

WLAN Troubleshooting with Wireshark and AirPcap

Frame Types: Probe Request / Probe Response


*AirPcap USB wireless capture adapter nr. 00 [Wireshark 1.10.0rc2 (SVN Rev 49526 from /trunk-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save
Wireless Settings... Decryption Keys...

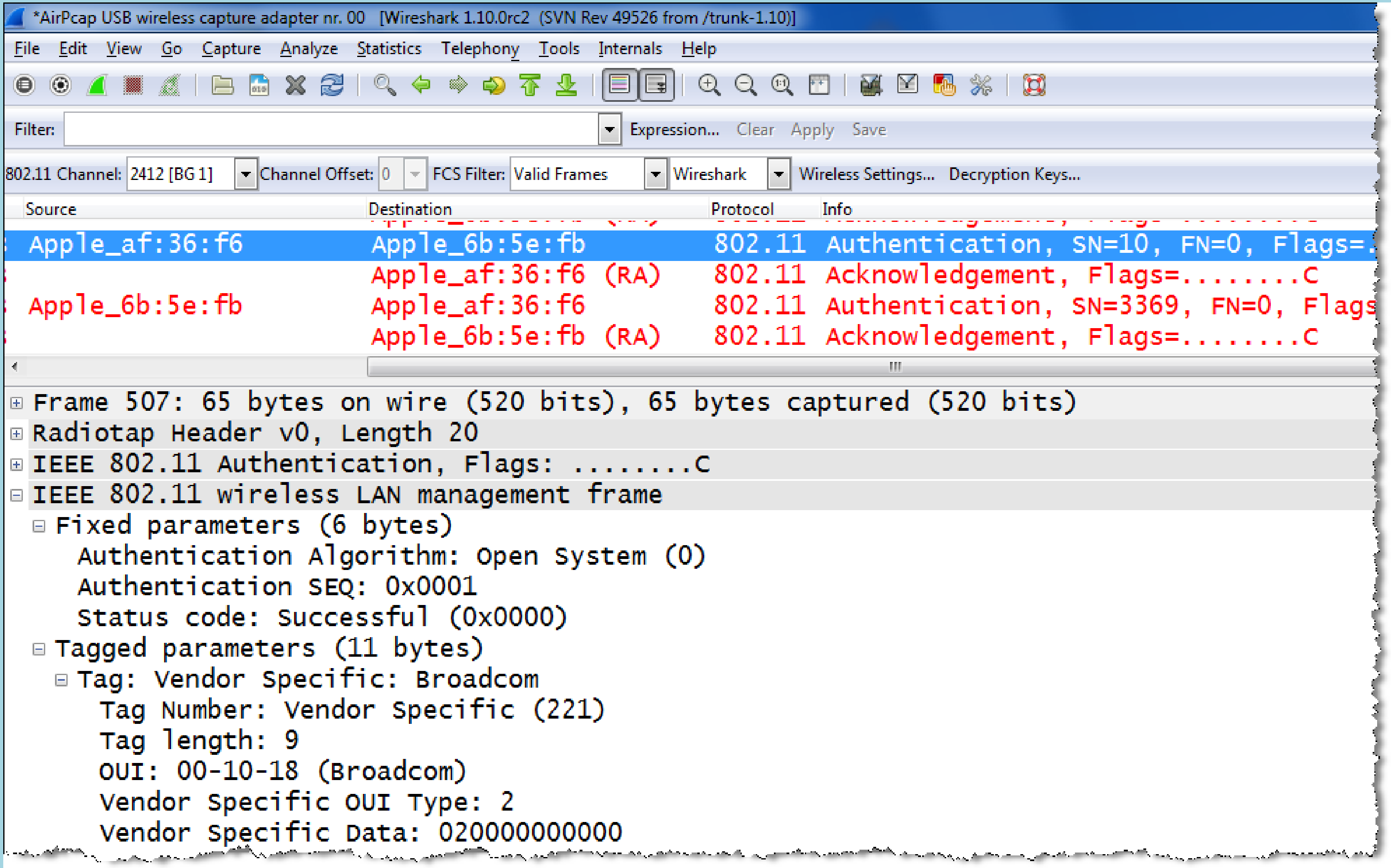
802.11 Channel: 2412 [BG 1]
Channel Offset: 0
FCS Filter: Valid Frames
None

```
Source          Destination    Protocol  Info
Apple_e0:52:2d  Broadcast      802.11    Probe Request, SN=1941, FN=0, Flags=......
Apple_6b:5e:fb  Apple_e0:52    802.11    Probe Response, SN=2132, FN=0, Flags=.....
                Apple_6b:5e    802.11    Acknowledgement, Flags=........C
Apple_e0:52:2d  Broadcast      802.11    Probe Request, SN=1942, FN=0, Flags=......
Apple_6b:5e:fb  Apple_e0:52    802.11    Probe Response, SN=2133, FN=0, Flags=.....
                Apple_6b:5e    802.11    Acknowledgement, Flags=........C
```

```
⊞ Frame 287: 157 bytes on wire (1256 bits), 157 bytes captured (1256 bits)
⊞ Radiotap Header v0, Length 20
⊞ IEEE 802.11 Probe Request, Flags: ........C
⊟ IEEE 802.11 wireless LAN management frame
  ⊟ Tagged parameters (109 bytes)
    ⊞ Tag: SSID parameter set: LNS WLAN
    ⊞ Tag: Supported Rates 1, 2, 5.5, 11, [Mbit/sec]
    ⊞ Tag: Extended Supported Rates 6, 9, 12, 18, 24, 36, 48, 54, [Mbit/sec]
    ⊞ Tag: DS Parameter set: Current Channel: 1
    ⊞ Tag: HT Capabilities (802.11n D1.10)
    ⊞ Tag: Vendor Specific: Broadcom
    ⊞ Tag: Vendor Specific: Epigram: HT Capabilities (802.11n D1.10)
    ⊞ Tag: Vendor Specific: Microsof: Unknown 8
```

WLAN Troubleshooting with Wireshark and AirPcap

Frame Types: Authentication Request / Authentication Response

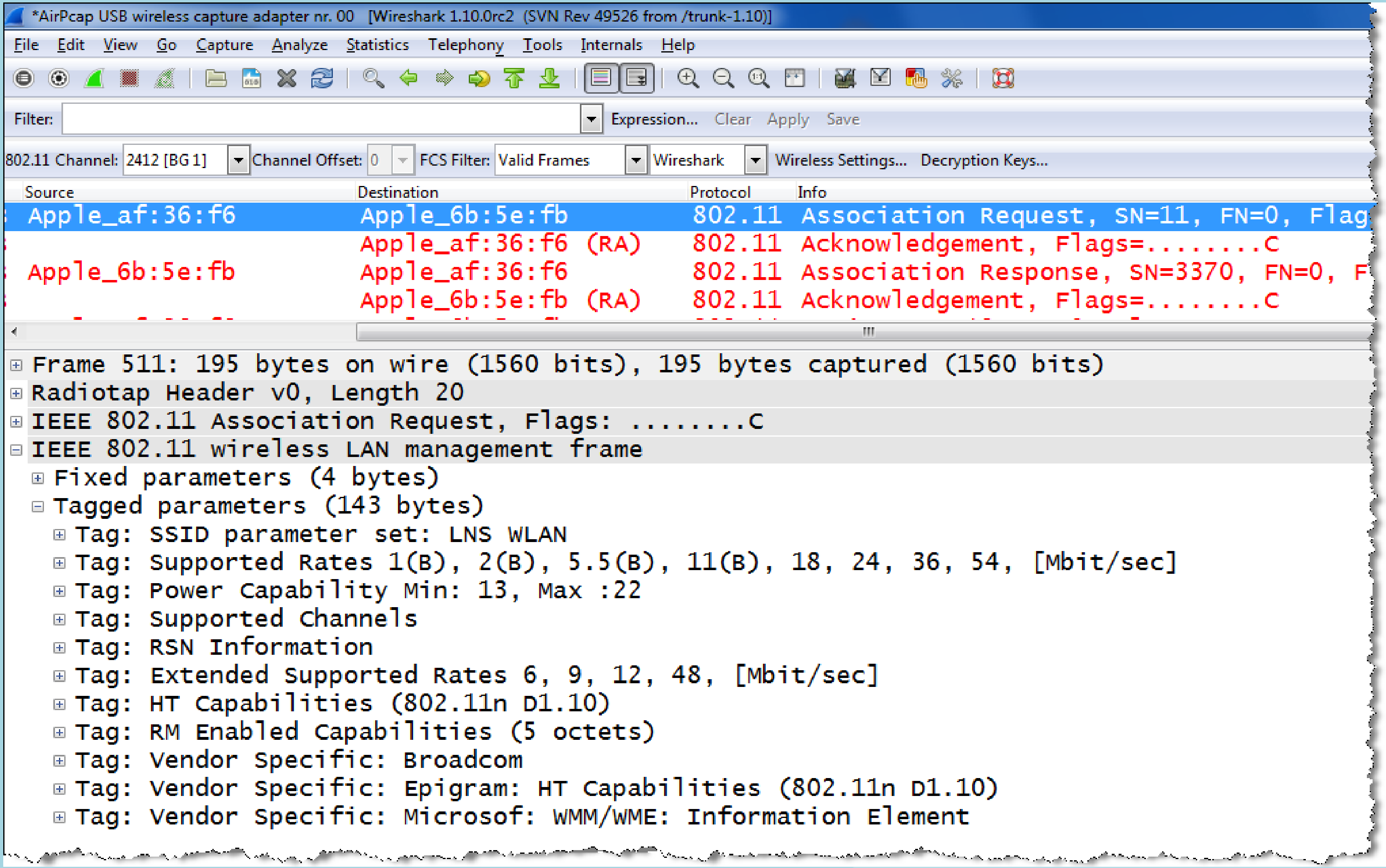

*AirPcap USB wireless capture adapter nr. 00 [Wireshark 1.10.0rc2 (SVN Rev 49526 from /trunk-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save
802.11 Channel: 2412 [BG 1]
Channel Offset: 0
FCS Filter: Valid Frames
Wireshark
Wireless Settings... Decryption Keys...

```
Source            Destination              Protocol  Info
Apple_af:36:f6    Apple_6b:5e:fb           802.11    Authentication, SN=10, FN=0, Flags=.
                  Apple_af:36:f6 (RA)      802.11    Acknowledgement, Flags=........C
Apple_6b:5e:fb    Apple_af:36:f6           802.11    Authentication, SN=3369, FN=0, Flags
                  Apple_6b:5e:fb (RA)      802.11    Acknowledgement, Flags=........C
```

```
⊞ Frame 507: 65 bytes on wire (520 bits), 65 bytes captured (520 bits)
⊞ Radiotap Header v0, Length 20
⊞ IEEE 802.11 Authentication, Flags: ........C
⊟ IEEE 802.11 wireless LAN management frame
  ⊟ Fixed parameters (6 bytes)
      Authentication Algorithm: Open System (0)
      Authentication SEQ: 0x0001
      Status code: Successful (0x0000)
  ⊟ Tagged parameters (11 bytes)
    ⊟ Tag: Vendor Specific: Broadcom
        Tag Number: Vendor Specific (221)
        Tag length: 9
        OUI: 00-10-18 (Broadcom)
        Vendor Specific OUI Type: 2
        Vendor Specific Data: 020000000000
```

WLAN Troubleshooting with Wireshark and AirPcap

Frame Types: Association Request / Association Response



*AirPcap USB wireless capture adapter nr. 00   [Wireshark 1.10.0rc2 (SVN Rev 49526 from /trunk-1.10)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save

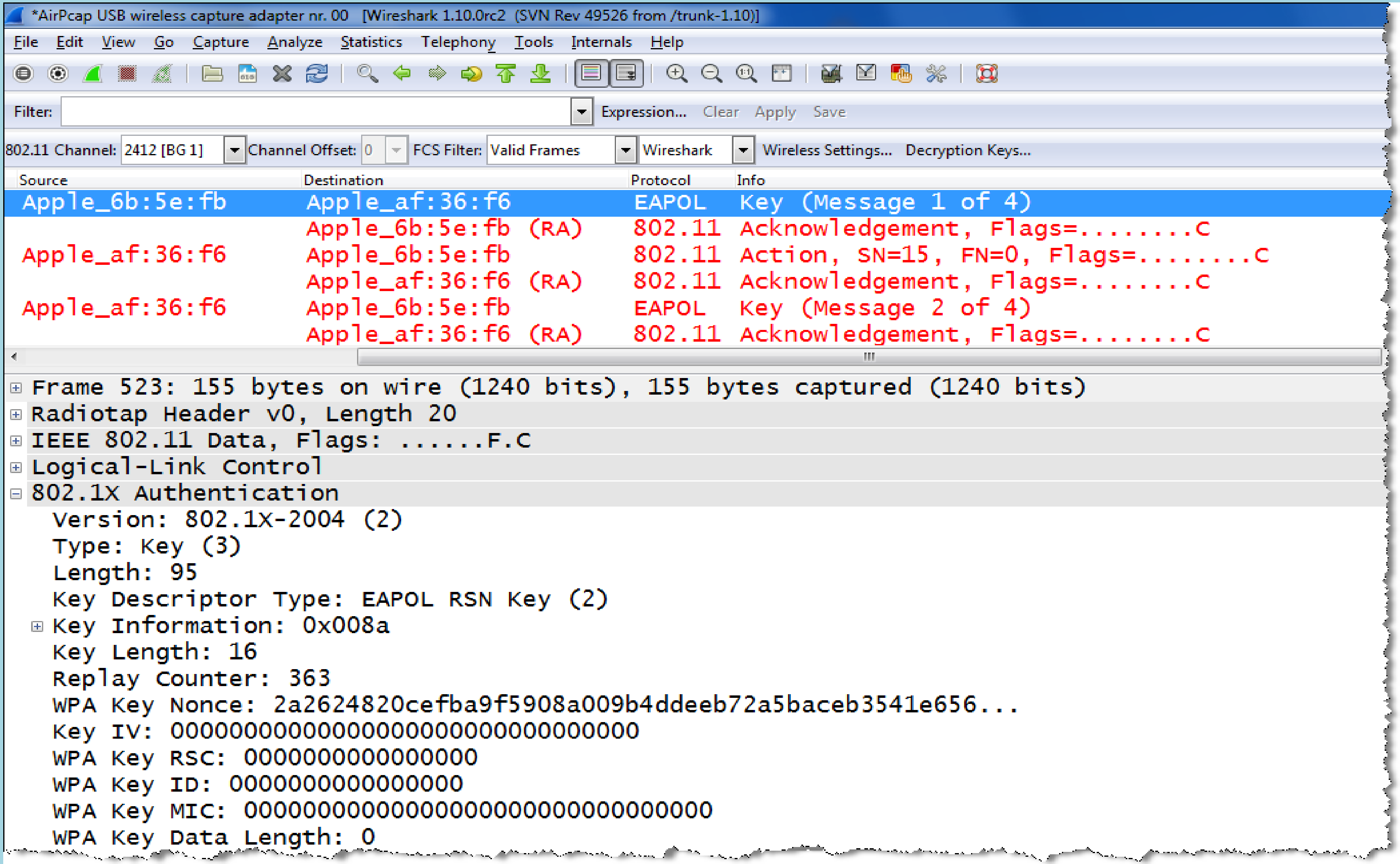
802.11 Channel: 2412 [BG 1] Channel Offset: 0 FCS Filter: Valid Frames Wireshark Wireless Settings... Decryption Keys...

```
Source              Destination              Protocol  Info
Apple_af:36:f6      Apple_6b:5e:fb           802.11    Association Request, SN=11, FN=0, Flag
                    Apple_af:36:f6 (RA)      802.11    Acknowledgement, Flags=........C
Apple_6b:5e:fb      Apple_af:36:f6           802.11    Association Response, SN=3370, FN=0, F
                    Apple_6b:5e:fb (RA)      802.11    Acknowledgement, Flags=........C
```

```
⊞ Frame 511: 195 bytes on wire (1560 bits), 195 bytes captured (1560 bits)
⊞ Radiotap Header v0, Length 20
⊞ IEEE 802.11 Association Request, Flags: ........C
⊟ IEEE 802.11 wireless LAN management frame
  ⊞ Fixed parameters (4 bytes)
  ⊟ Tagged parameters (143 bytes)
    ⊞ Tag: SSID parameter set: LNS WLAN
    ⊞ Tag: Supported Rates 1(B), 2(B), 5.5(B), 11(B), 18, 24, 36, 54, [Mbit/sec]
    ⊞ Tag: Power Capability Min: 13, Max :22
    ⊞ Tag: Supported Channels
    ⊞ Tag: RSN Information
    ⊞ Tag: Extended Supported Rates 6, 9, 12, 48, [Mbit/sec]
    ⊞ Tag: HT Capabilities (802.11n D1.10)
    ⊞ Tag: RM Enabled Capabilities (5 octets)
    ⊞ Tag: Vendor Specific: Broadcom
    ⊞ Tag: Vendor Specific: Epigram: HT Capabilities (802.11n D1.10)
    ⊞ Tag: Vendor Specific: Microsof: WMM/WME: Information Element
```

WLAN Troubleshooting with Wireshark and AirPcap

Frame Types: EAPOL Key Messages



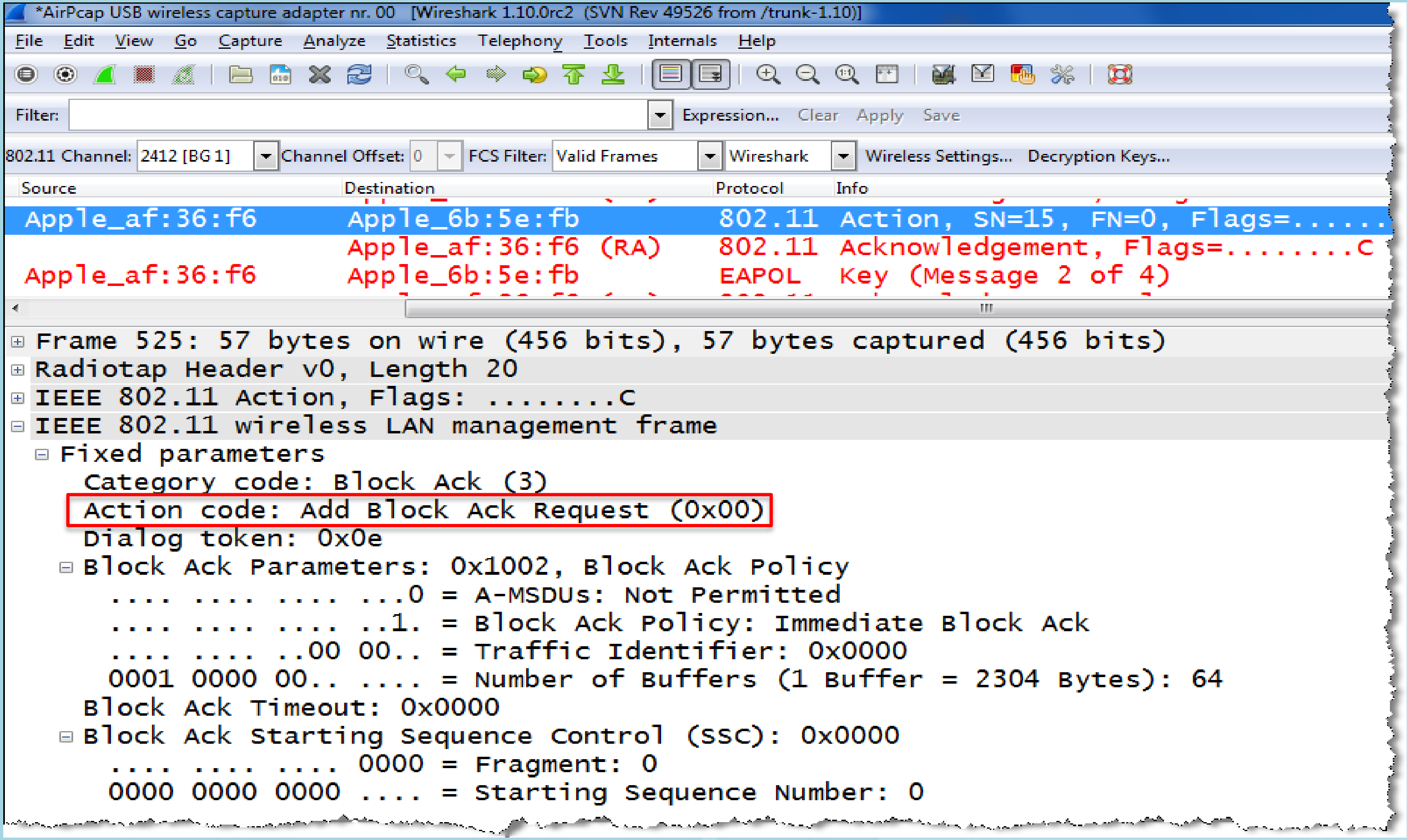
*AirPcap USB wireless capture adapter nr. 00 [Wireshark 1.10.0rc2 (SVN Rev 49526 from /trunk-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save
802.11 Channel: 2412 [BG 1]
Channel Offset: 0
FCS Filter: Valid Frames
Wireshark
Wireless Settings... Decryption Keys...

```
Source              Destination              Protocol  Info
Apple_6b:5e:fb      Apple_af:36:f6           EAPOL     Key (Message 1 of 4)
                    Apple_6b:5e:fb (RA)      802.11    Acknowledgement, Flags=........C
Apple_af:36:f6      Apple_6b:5e:fb           802.11    Action, SN=15, FN=0, Flags=........C
                    Apple_af:36:f6 (RA)      802.11    Acknowledgement, Flags=........C
Apple_af:36:f6      Apple_6b:5e:fb           EAPOL     Key (Message 2 of 4)
                    Apple_af:36:f6 (RA)      802.11    Acknowledgement, Flags=........C
```

```
⊞ Frame 523: 155 bytes on wire (1240 bits), 155 bytes captured (1240 bits)
⊞ Radiotap Header v0, Length 20
⊞ IEEE 802.11 Data, Flags: ......F.C
⊞ Logical-Link Control
⊟ 802.1X Authentication
    Version: 802.1X-2004 (2)
    Type: Key (3)
    Length: 95
    Key Descriptor Type: EAPOL RSN Key (2)
  ⊞ Key Information: 0x008a
    Key Length: 16
    Replay Counter: 363
    WPA Key Nonce: 2a2624820cefba9f5908a009b4ddeeb72a5baceb3541e656...
    Key IV: 00000000000000000000000000000000
    WPA Key RSC: 0000000000000000
    WPA Key ID: 0000000000000000
    WPA Key MIC: 00000000000000000000000000000000
    WPA Key Data Length: 0
```

WLAN Troubleshooting with Wireshark and AirPcap

Frame Type: Action


*AirPcap USB wireless capture adapter nr. 00   [Wireshark 1.10.0rc2 (SVN Rev 49526 from /trunk-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save
Wireless Settings... Decryption Keys...

802.11 Channel: 2412 [BG 1] Channel Offset: 0 FCS Filter: Valid Frames Wireshark

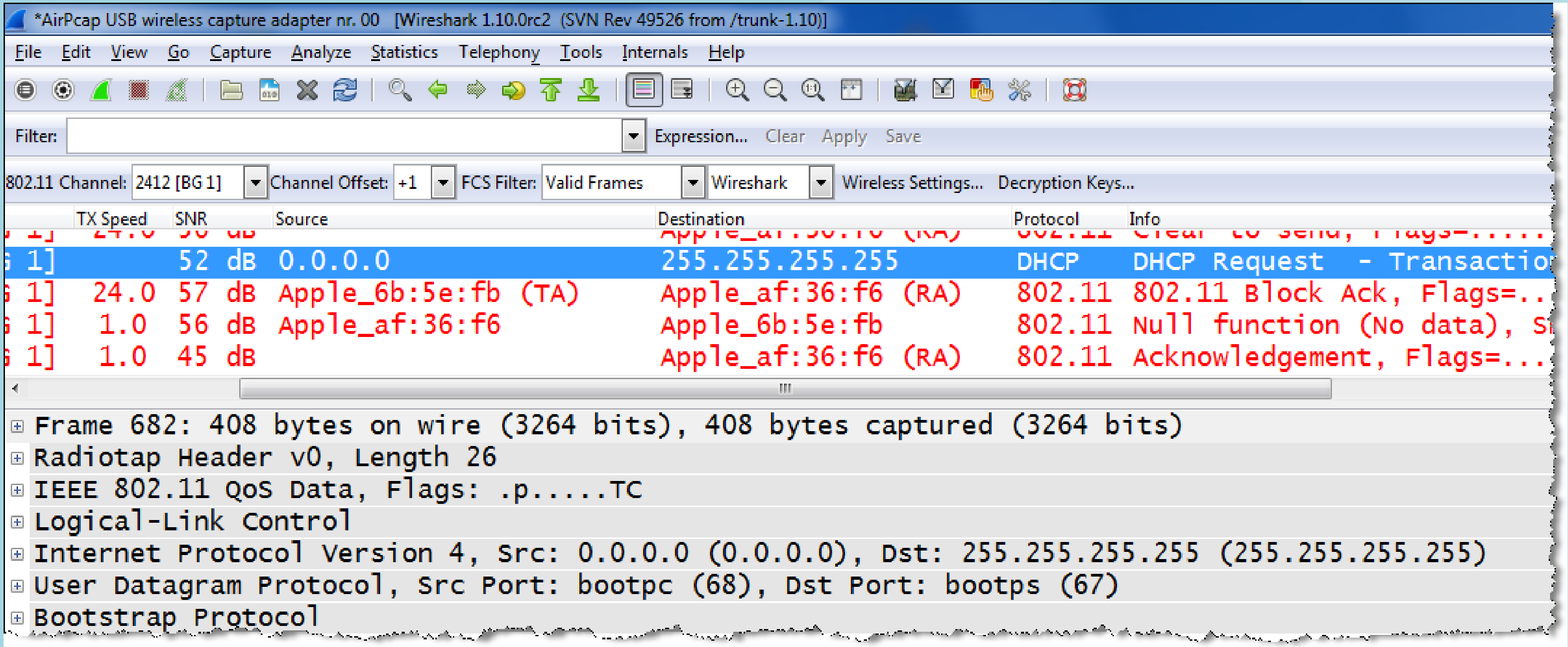
```
Source              Destination              Protocol   Info
Apple_af:36:f6      Apple_6b:5e:fb           802.11     Action, SN=15, FN=0, Flags=......
                    Apple_af:36:f6 (RA)      802.11     Acknowledgement, Flags=........C
Apple_af:36:f6      Apple_6b:5e:fb           EAPOL      Key (Message 2 of 4)
```

```
⊞ Frame 525: 57 bytes on wire (456 bits), 57 bytes captured (456 bits)
⊞ Radiotap Header v0, Length 20
⊞ IEEE 802.11 Action, Flags: ........C
⊟ IEEE 802.11 wireless LAN management frame
   ⊟ Fixed parameters
        Category code: Block Ack (3)
        Action code: Add Block Ack Request (0x00)
        Dialog token: 0x0e
      ⊟ Block Ack Parameters: 0x1002, Block Ack Policy
           .... .... .... ...0 = A-MSDUs: Not Permitted
           .... .... .... ..1. = Block Ack Policy: Immediate Block Ack
           .... .... ..00 00.. = Traffic Identifier: 0x0000
           0001 0000 00.. .... = Number of Buffers (1 Buffer = 2304 Bytes): 64
        Block Ack Timeout: 0x0000
      ⊟ Block Ack Starting Sequence Control (SSC): 0x0000
           .... .... .... 0000 = Fragment: 0
           0000 0000 0000 .... = Starting Sequence Number: 0
```

WLAN Troubleshooting with Wireshark and AirPcap

Frame Types: Decrypted Data Frame followed by Block Acknowledge

- WEP and WPA1/2 personal mode (shared key) can be decrypted by Wireshark
- To enable WPA decryption, the key negotiation process must be captured too
- Shared Key decryptions is possible during capturing or offline from a stored file



*AirPcap USB wireless capture adapter nr. 00 [Wireshark 1.10.0rc2 (SVN Rev 49526 from /trunk-1.10)]

Filter:

802.11 Channel: 2412 [BG 1] Channel Offset: +1 FCS Filter: Valid Frames Wireshark

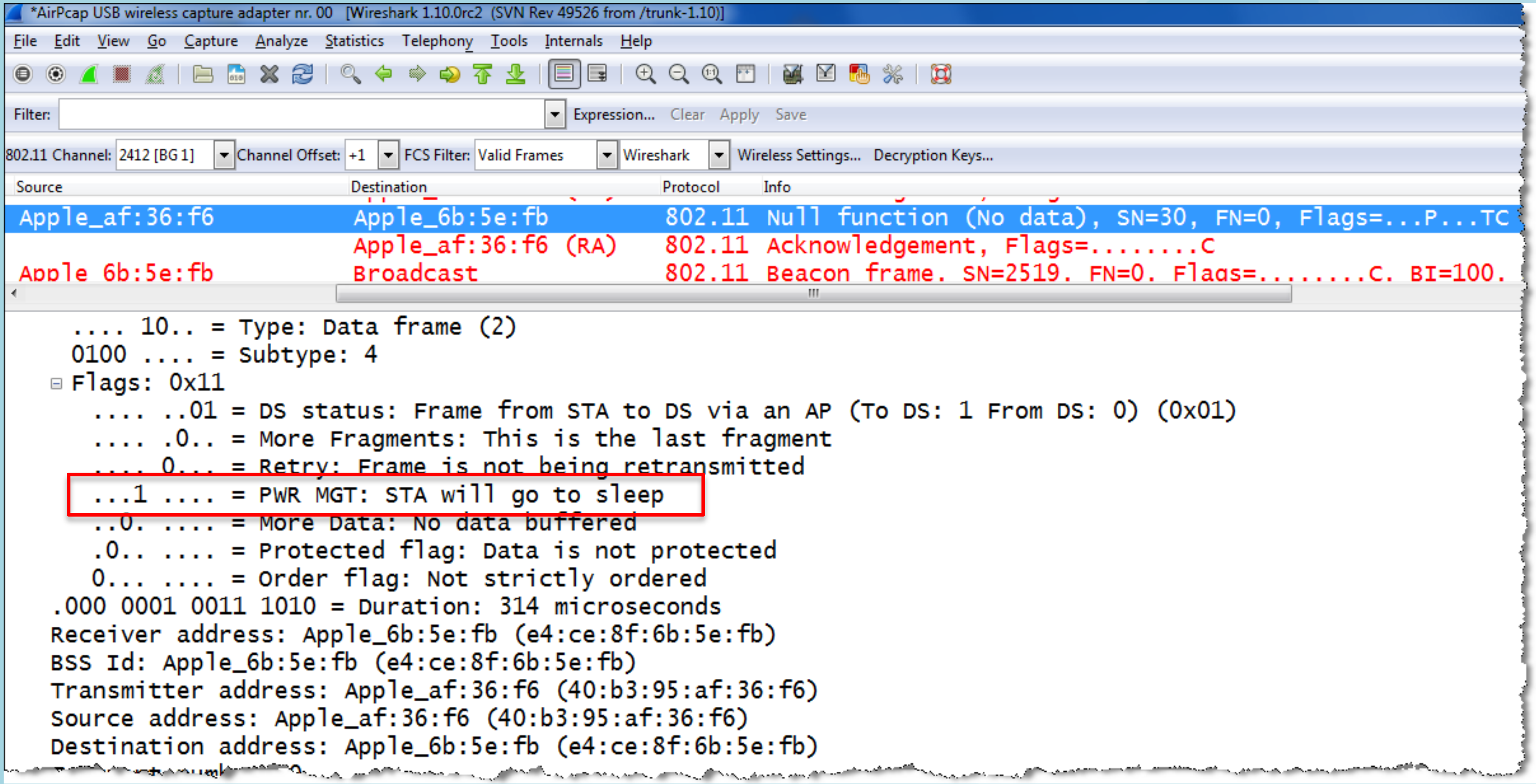
```
    TX Speed  SNR     Source                   Destination             Protocol   Info
 1]   24.0    50 dB                            Apple_af:36:f6 (RA)     802.11     Clear to send, Flags=.......
 1]           52 dB   0.0.0.0                  255.255.255.255         DHCP       DHCP Request  - Transactio
 1]   24.0    57 dB   Apple_6b:5e:fb (TA)      Apple_af:36:f6 (RA)     802.11     802.11 Block Ack, Flags=..
 1]    1.0    56 dB   Apple_af:36:f6           Apple_6b:5e:fb          802.11     Null function (No data), SN
 1]    1.0    45 dB                            Apple_af:36:f6 (RA)     802.11     Acknowledgement, Flags=...
```

```
⊞ Frame 682: 408 bytes on wire (3264 bits), 408 bytes captured (3264 bits)
⊞ Radiotap Header v0, Length 26
⊞ IEEE 802.11 QoS Data, Flags: .p.....TC
⊞ Logical-Link Control
⊞ Internet Protocol Version 4, Src: 0.0.0.0 (0.0.0.0), Dst: 255.255.255.255 (255.255.255.255)
⊞ User Datagram Protocol, Src Port: bootpc (68), Dst Port: bootps (67)
⊞ Bootstrap Protocol
```

WLAN Troubleshooting with Wireshark and AirPcap

Frame Types: Null Function Data followed by Acknowledge

- The Null Function frame is often used as keep-alive message from the client
- Another purpose is to inform the AP if the client is changing the power save status



*AirPcap USB wireless capture adapter nr. 00 [Wireshark 1.10.0rc2 (SVN Rev 49526 from /trunk-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save
802.11 Channel: 2412 [BG 1]
Channel Offset: +1
FCS Filter: Valid Frames
Wireshark
Wireless Settings... Decryption Keys...

```
Source              Destination            Protocol  Info
Apple_af:36:f6      Apple_6b:5e:fb         802.11    Null function (No data), SN=30, FN=0, Flags=...P...TC
                    Apple_af:36:f6 (RA)    802.11    Acknowledgement, Flags=........C
Apple_6b:5e:fb      Broadcast              802.11    Beacon frame, SN=2519, FN=0, Flags=........C, BI=100,
```

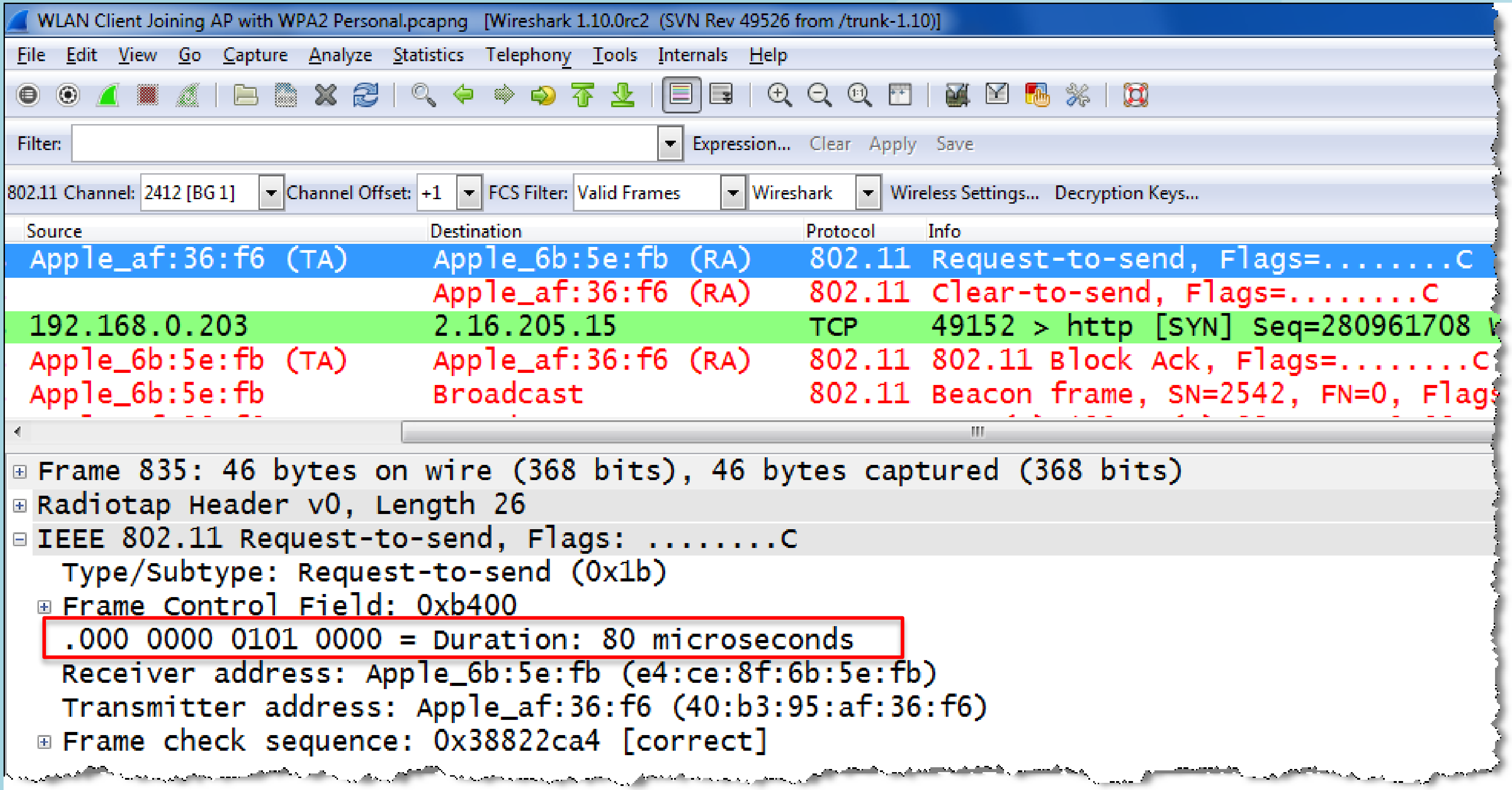
```
   .... 10.. = Type: Data frame (2)
   0100 .... = Subtype: 4
 ⊟ Flags: 0x11
     .... ..01 = DS status: Frame from STA to DS via an AP (To DS: 1 From DS: 0) (0x01)
     .... .0.. = More Fragments: This is the last fragment
     .... 0... = Retry: Frame is not being retransmitted
     ...1 .... = PWR MGT: STA will go to sleep
     ..0. .... = More Data: No data buffered
     .0.. .... = Protected flag: Data is not protected
     0... .... = Order flag: Not strictly ordered
  .000 0001 0011 1010 = Duration: 314 microseconds
  Receiver address: Apple_6b:5e:fb (e4:ce:8f:6b:5e:fb)
  BSS Id: Apple_6b:5e:fb (e4:ce:8f:6b:5e:fb)
  Transmitter address: Apple_af:36:f6 (40:b3:95:af:36:f6)
  Source address: Apple_af:36:f6 (40:b3:95:af:36:f6)
  Destination address: Apple_6b:5e:fb (e4:ce:8f:6b:5e:fb)
```

© Leutert NetServices 2013

WLAN Troubleshooting with Wireshark and AirPcap

Frame Types: Request-to-send (RTS) and Clear-to-send (CTS)

- RTS /CTS are used to reserve airtime in hidden node situations or busy networks
- Another purpose is to hinder old clients from interfering with clients of new standards



WLAN Client Joining AP with WPA2 Personal.pcapng [Wireshark 1.10.0rc2 (SVN Rev 49526 from /trunk-1.10)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save
Wireless Settings... Decryption Keys...

802.11 Channel: 2412 [BG 1] Channel Offset: +1 FCS Filter: Valid Frames Wireshark

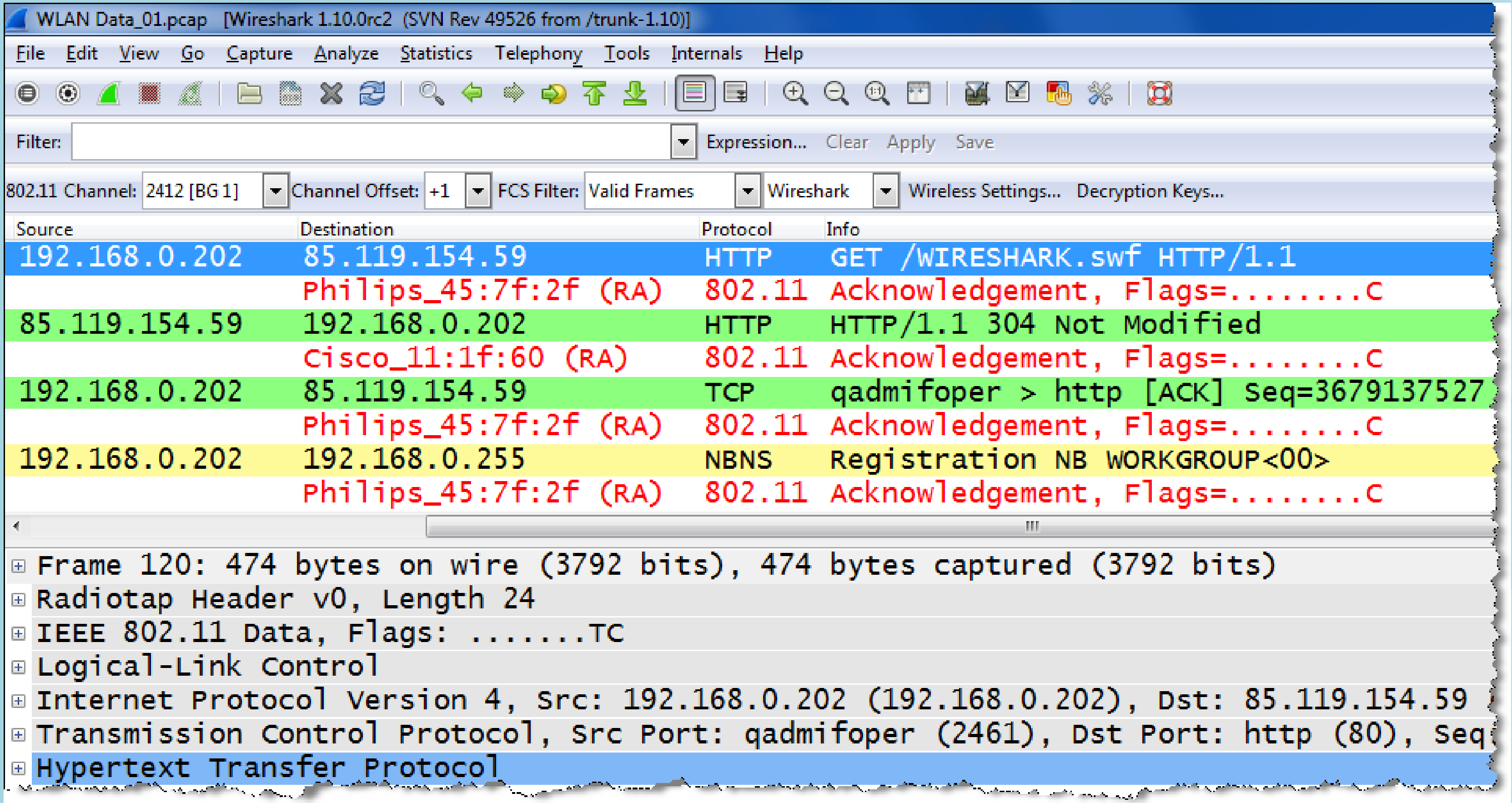
```
Source                  Destination             Protocol  Info
Apple_af:36:f6 (TA)     Apple_6b:5e:fb (RA)     802.11    Request-to-send, Flags=........C
                        Apple_af:36:f6 (RA)     802.11    Clear-to-send, Flags=........C
192.168.0.203           2.16.205.15             TCP       49152 > http [SYN] Seq=280961708
Apple_6b:5e:fb (TA)     Apple_af:36:f6 (RA)     802.11    802.11 Block Ack, Flags=........C
Apple_6b:5e:fb          Broadcast               802.11    Beacon frame, SN=2542, FN=0, Flag
```

```
⊞ Frame 835: 46 bytes on wire (368 bits), 46 bytes captured (368 bits)
⊞ Radiotap Header v0, Length 26
⊟ IEEE 802.11 Request-to-send, Flags: ........C
    Type/Subtype: Request-to-send (0x1b)
  ⊞ Frame Control Field: 0xb400
    .000 0000 0101 0000 = Duration: 80 microseconds
    Receiver address: Apple_6b:5e:fb (e4:ce:8f:6b:5e:fb)
    Transmitter address: Apple_af:36:f6 (40:b3:95:af:36:f6)
  ⊞ Frame check sequence: 0x38822ca4 [correct]
```

WLAN Troubleshooting with Wireshark and AirPcap

Frame Types: Data and Acknowledges

- In the air, every Data frame is acknowledged or otherwise retransmitted
- 802.11 a/b/g every single Data frame is acknowledged. 802.11n introduced Block Acks
- Single Acks must follow immediately after a Data frame and have no source address



WLAN Data_01.pcap [Wireshark 1.10.0rc2 (SVN Rev 49526 from /trunk-1.10)]

802.11 Channel: 2412 [BG 1]
Channel Offset: +1
FCS Filter: Valid Frames
Wireshark

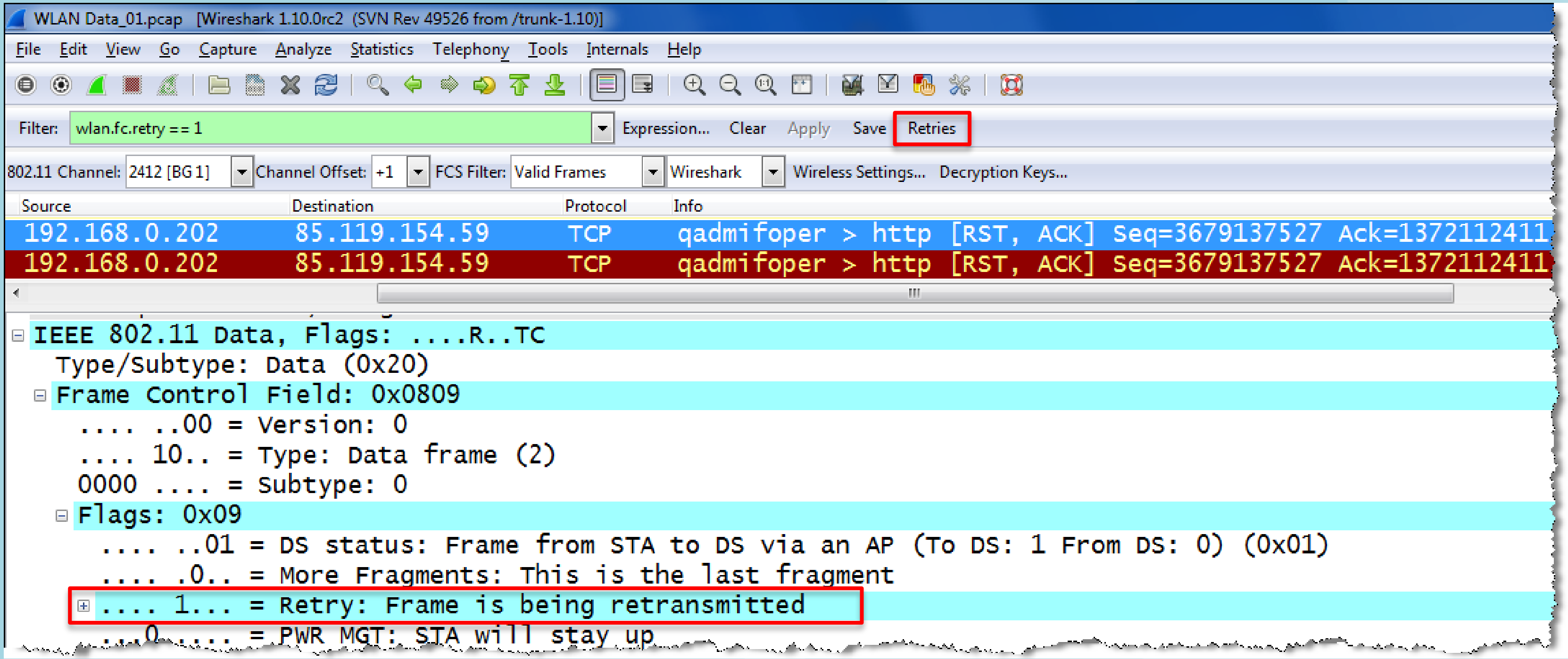
```
Source           Destination              Protocol  Info
192.168.0.202    85.119.154.59            HTTP      GET /WIRESHARK.swf HTTP/1.1
                 Philips_45:7f:2f (RA)    802.11    Acknowledgement, Flags=........C
85.119.154.59    192.168.0.202            HTTP      HTTP/1.1 304 Not Modified
                 Cisco_11:1f:60 (RA)      802.11    Acknowledgement, Flags=........C
192.168.0.202    85.119.154.59            TCP       qadmifoper > http [ACK] Seq=3679137527
                 Philips_45:7f:2f (RA)    802.11    Acknowledgement, Flags=........C
192.168.0.202    192.168.0.255            NBNS      Registration NB WORKGROUP<00>
                 Philips_45:7f:2f (RA)    802.11    Acknowledgement, Flags=........C
```

```
⊞ Frame 120: 474 bytes on wire (3792 bits), 474 bytes captured (3792 bits)
⊞ Radiotap Header v0, Length 24
⊞ IEEE 802.11 Data, Flags: .......TC
⊞ Logical-Link Control
⊞ Internet Protocol Version 4, Src: 192.168.0.202 (192.168.0.202), Dst: 85.119.154.59
⊞ Transmission Control Protocol, Src Port: qadmifoper (2461), Dst Port: http (80), Seq
⊞ Hypertext Transfer Protocol
```

WLAN Troubleshooting with Wireshark and AirPcap

Filter on Retransmitted frames

- Retransmitted frames are marked with the Retry Bit by the sender
- Create a Display Filter on retransmitted frames and save it as a Quick Filter Button
- Watch the percentage of retransmitted versus original frames in the bottom line



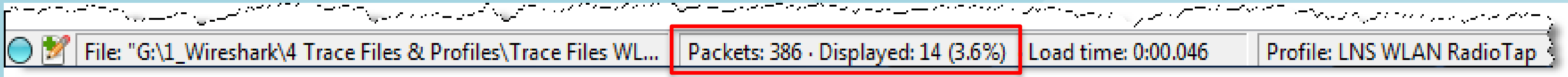
WLAN Data_01.pcap [Wireshark 1.10.0rc2 (SVN Rev 49526 from /trunk-1.10)]

Filter: wlan.fc.retry == 1

802.11 Channel: 2412 [BG 1] Channel Offset: +1 FCS Filter: Valid Frames Wireshark

Source	Destination	Protocol	Info
192.168.0.202	85.119.154.59	TCP	qadmifoper > http [RST, ACK] Seq=3679137527 Ack=1372112411
192.168.0.202	85.119.154.59	TCP	qadmifoper > http [RST, ACK] Seq=3679137527 Ack=1372112411

```
⊟ IEEE 802.11 Data, Flags: ....R..TC
    Type/Subtype: Data (0x20)
  ⊟ Frame Control Field: 0x0809
      .... ..00 = Version: 0
      .... 10.. = Type: Data frame (2)
      0000 .... = Subtype: 0
    ⊟ Flags: 0x09
        .... ..01 = DS status: Frame from STA to DS via an AP (To DS: 1 From DS: 0) (0x01)
        .... .0.. = More Fragments: This is the last fragment
      ⊞ .... 1... = Retry: Frame is being retransmitted
        ...0 .... = PWR MGT: STA will stay up
```



File: "G:\1_Wireshark\4 Trace Files & Profiles\Trace Files WL... | Packets: 386 · Displayed: 14 (3.6%) | Load time: 0:00.046 | Profile: LNS WLAN RadioTap

WLAN Troubleshooting with Wireshark and AirPcap

Where to capture WLAN frames

- The physical location within a radio cell is relevant for your capturing results

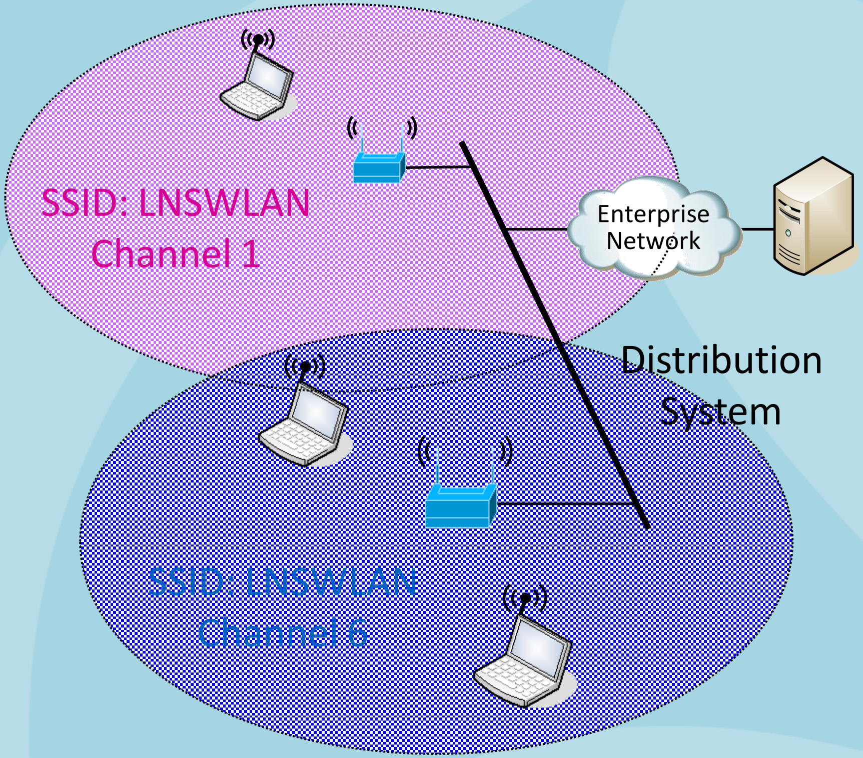
Rules of thumb

For analyzing problems in a single cell:

- Stay near the Access Point
- All traffic flows through the AP
- Clients must not hear each other

For analyzing roaming problems:

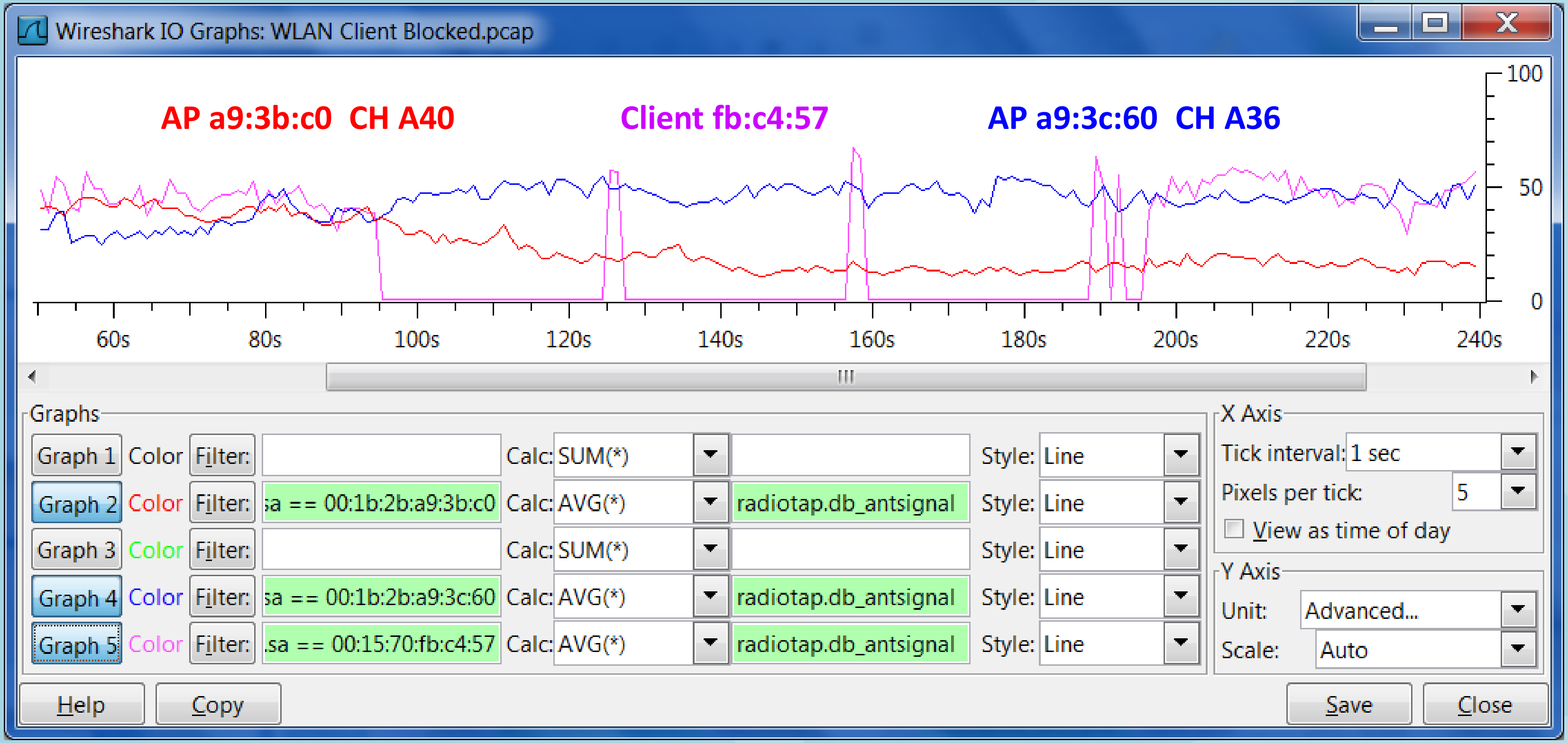
- Stay near the roaming client
- Capture with multiple AirPcaps
- Use Beacons to define your location



WLAN Troubleshooting with Wireshark and AirPcap

Graphical presentation of Radio Signal Strength with Wireshark IO Graphs

- Using the field radiotap.db_antsignal from two AirPcap NX tuned in two channels



Graphs

Graph 1	Color	Filter:		Calc: SUM(*)		Style: Line
Graph 2	Color	Filter:	sa == 00:1b:2b:a9:3b:c0	Calc: AVG(*)	radiotap.db_antsignal	Style: Line
Graph 3	Color	Filter:		Calc: SUM(*)		Style: Line
Graph 4	Color	Filter:	sa == 00:1b:2b:a9:3c:60	Calc: AVG(*)	radiotap.db_antsignal	Style: Line
Graph 5	Color	Filter:	sa == 00:15:70:fb:c4:57	Calc: AVG(*)	radiotap.db_antsignal	Style: Line

X Axis

Tick interval: 1 sec

Pixels per tick: 5

☐ View as time of day

Y Axis

Unit: Advanced...

Scale: Auto

Graph 2 Color Filter: wlan.sa == 00:1b:2b:a9:3b:c0 → Access Point in Channel A40

Graph 4 Color Filter: wlan.sa == 00:1b:2b:a9:3c:60 → Access Point in Channel A36

Graph 5 Color Filter: wlan.sa == 00:15:70:fb:c4:57 → Mobile Client followed with Wireshark

WLAN Troubleshooting with Wireshark and AirPcap

Overview of WLAN standards

Mbps	Coding	Modulation	Description	
1 2	Barker Barker	DBPSK	**802.11** **DSSS (Clause 15)** with ‚Long Preamble‘	
5.5 11	CCK CCK	DQPSK	**802.11b** **HR/DSSS (Clause 18)** with ‚Short Preamble‘	
6, 9 12, 18 24, 36 48, 54	OFDM OFDM OFDM OFDM	BPSK QPSK 16-QAM 64-QAM	**802.11g** **Extended Rate PHY** **(ERP)**	**802.11a**
7.2-72.2 14.4-144.4	OFDM OFDM	MCS 0-7 MCS 8-15	1 Stream 2 Streams **802.11n** **High Troughput (HT)** **Extensions**	

2.4 GHz

5 GHz

CCK = Complementary Code Keying
DBPSK = Differential Binary Phase-Shift Keying
DQPSK = Differential Quadrature Phase-Shift Keying
OFDM = Orthogonal Frequency Division Multiplexing

BPSK = Binary Phase-Shift Keying
QPSK = Quadrature Phase-Shift Keying
QAM = Quadrature Amplitude Modul.
MCS = Modulation Coding Scheme

WLAN Troubleshooting with Wireshark and AirPcap

Outlook to WLAN products and standards

- 802.11n products using 4 streams will go up to 600 Mbps (PHY data rate)
- 802.11n products using Beamforming to focus RF energy and improve radio signal
- 802.11z Direct Link Setup to allow direct client to client communication
- 802.11w Management Frame Protection to increase security level against intruders

- 802.11ac 5G WiFi is an improvement to 802.11n. Uses 5GHz band and defines up to a maximum of 6.93 Gbps with up to 8 streams and up to 8 bonded channels (160 MHz)

802.11ac 5G WiFi logo

802.11ad WiGig logo

- 802.11ad WiGig for short range WLANs using 60GHz band with up to 7Gbps

Thank you for your attention

Hope you learned something useful

Rolf Leutert, Leutert NetServices, www.wireshark.ch