Wireless Troubleshooting Tips using AirPcaps: DFS & Module Debugging

Megumi Takeshita
ikeriri network service co.,ltd

Megumi Takeshita, ikeriri network service a.k.a. packet otaku since first Sharkfest

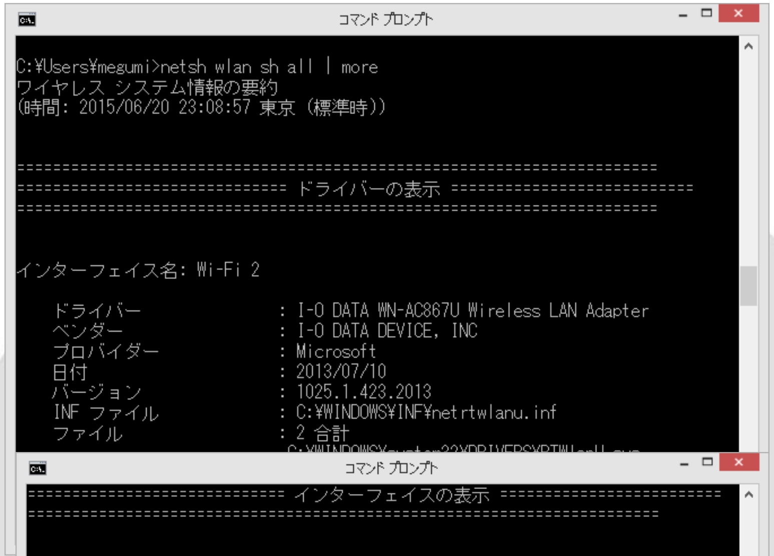
- Founder, ikeriri network service co.,ltd I am network troubleshooter and debugger using packet analysis.
- Wrote 10+ books of packet capturing
- Reseller of Riverbed Technology (former CACE technologies) and Metageek, Dualcomm etc. in Japan
- Attending all Sharkfest and first translator of QT Wireshark into Japanese ! 日本語ワイヤーシャーク

Wireless troubleshooting TIPS using AirPcaps: DFS & Module Debugging

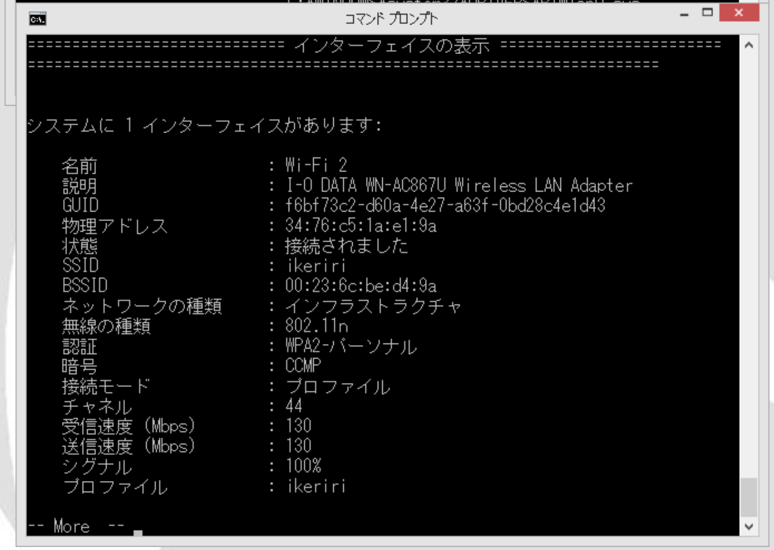
- Now I talk about 20 TIPS and troubleshooting in wireless environment
- AirPcap(s) is necessary for debugging in Windows environment.
- Please ask me if you have some question.

#1 Collecting host / AP info (Windows)

- “netsh wlan sh all | more “
  Driver description
  Driver version (important)
  INF file name
  Interface name
  MAC address
  SSID / BSSID
  authentication/encryption
  Channel / speed /signal
- Demonstration



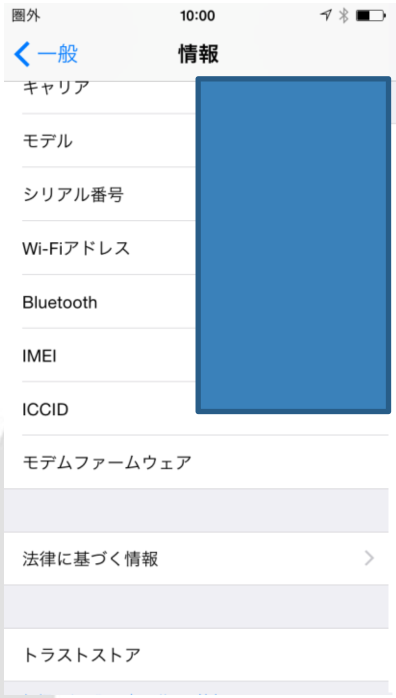
```
C:¥Users¥megumi>netsh wlan sh all | more
ワイヤレス システム情報の要約
(時間: 2015/06/20 23:08:57 東京 (標準時))


=======================================================================
========================== ドライバーの表示 ============================
=======================================================================


インターフェイス名: Wi-Fi 2

    ドライバー             : I-O DATA WN-AC867U Wireless LAN Adapter
    ベンダー               : I-O DATA DEVICE, INC
    プロバイダー           : Microsoft
    日付                   : 2013/07/10
    バージョン             : 1025.1.423.2013
    INF ファイル           : C:¥WINDOWS¥INF¥netrtwlanu.inf
    ファイル               : 2 合計
```



```
========================== インターフェイスの表示 =======================
=======================================================================


システムに 1 インターフェイスがあります:

    名前                   : Wi-Fi 2
    説明                   : I-O DATA WN-AC867U Wireless LAN Adapter
    GUID                   : f6bf73c2-d60a-4e27-a63f-0bd28c4e1d43
    物理アドレス           : 34:76:c5:1a:e1:9a
    状態                   : 接続されました
    SSID                   : ikeriri
    BSSID                  : 00:23:6c:be:d4:9a
    ネットワークの種類     : インフラストラクチャ
    無線の種類             : 802.11n
    認証                   : WPA2-パーソナル
    暗号                   : CCMP
    接続モード             : プロファイル
    チャネル               : 44
    受信速度 (Mbps)        : 130
    送信速度 (Mbps)        : 130
    シグナル               : 100%
    プロファイル           : ikeriri

-- More  --
```

#1 Collecting host / AP info (iOS)

- Setting>General>Info
  “MAC address”
- Setting>Privacy>Location
  if “disabled” and no carrier setting may causes randomize
  MAC address (iOS8)
- Setting>Wi-Fi
  SSID / IP address / mask / gateway / DNS…


圏外 10:00

< 一般 情報

キャリア [redacted]

モデル

シリアル番号

Wi-Fiアドレス

Bluetooth

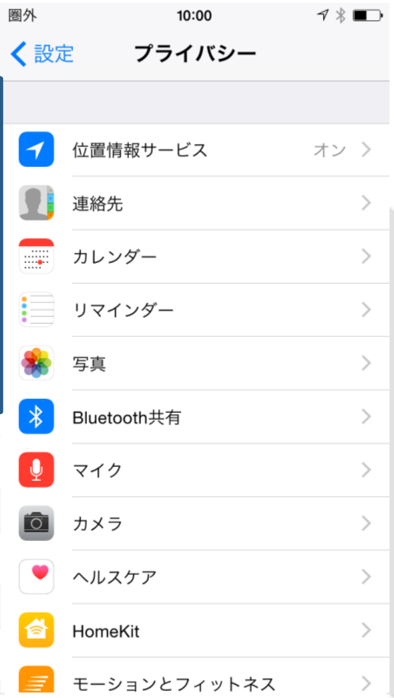
IMEI

ICCID

モデムファームウェア

法律に基づく情報 >

トラストストア


圏外 10:00

< 設定 プライバシー

位置情報サービス オン >

連絡先 >

カレンダー >

リマインダー >

写真 >

Bluetooth共有 >

マイク >

カメラ >

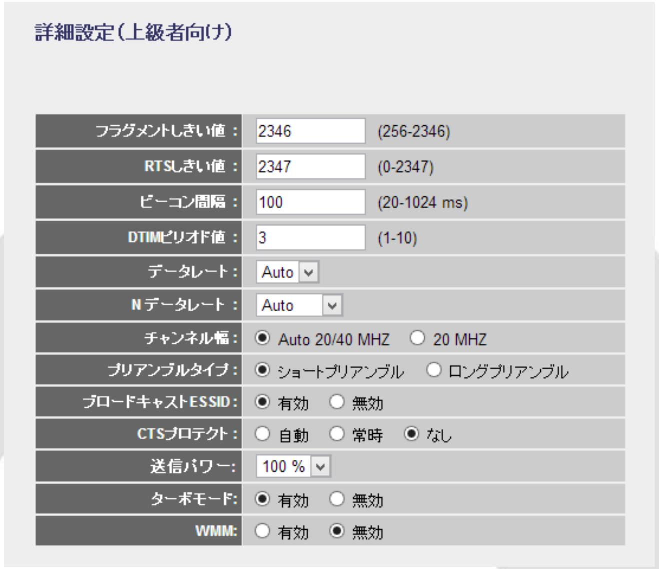
ヘルスケア >

HomeKit >

モーションとフィットネス >

#1 Collecting host / AP info (AP Side)

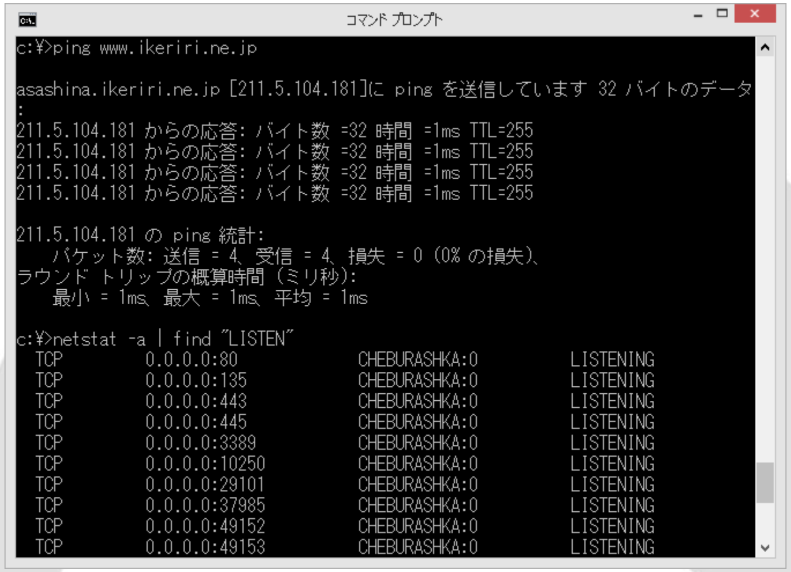
- SSID / BSSID / Channel / Channel bandwidth connection speed/mode encryption type / SSID etc.
- Also check the controller settings (if user use),
- Short Guard Interval 20 and Greenfield mode (High Throughput) are not supported by AirPcap series.


詳細設定（上級者向け）

フラグメントしきい値：	2346 (256-2346)
RTSしきい値：	2347 (0-2347)
ビーコン間隔：	100 (20-1024 ms)
DTIMピリオド値：	3 (1-10)
データレート：	Auto
Nデータレート：	Auto
チャンネル幅：	[✓ Auto 20/40 MHZ] 20 MHZ
プリアンブルタイプ：	[✓ ショートプリアンブル] ロングプリアンブル
ブロードキャストESSID：	[✓ 有効] 無効
CTSプロテクト：	自動 常時 [✓ なし]
送信パワー：	100 %
ターボモード：	[✓ 有効] 無効
WMM：	有効 [✓ 無効]

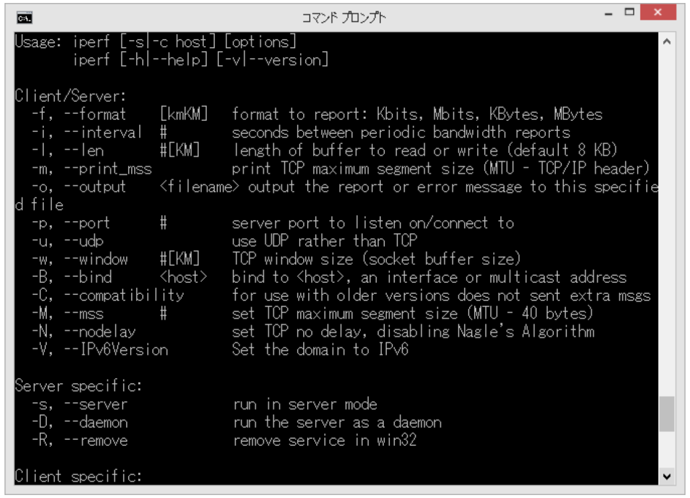
#2 Collecting Baseline of network

- Latency and lost
  of Ping command
- tracert and pathping
- netstat –a | find “LISTEN”
- Iperf (throughput test)



コマンド プロンプト

```
c:¥>ping www.ikeriri.ne.jp

asashina.ikeriri.ne.jp [211.5.104.181]に ping を送信しています 32 バイトのデータ
:
211.5.104.181 からの応答: バイト数 =32 時間 =1ms TTL=255
211.5.104.181 からの応答: バイト数 =32 時間 =1ms TTL=255
211.5.104.181 からの応答: バイト数 =32 時間 =1ms TTL=255
211.5.104.181 からの応答: バイト数 =32 時間 =1ms TTL=255

211.5.104.181 の ping 統計:
    パケット数: 送信 = 4、受信 = 4、損失 = 0 (0% の損失)、
ラウンド トリップの概算時間 (ミリ秒):
    最小 = 1ms、最大 = 1ms、平均 = 1ms

c:¥>netstat -a | find "LISTEN"
  TCP         0.0.0.0:80             CHEBURASHKA:0          LISTENING
  TCP         0.0.0.0:135            CHEBURASHKA:0          LISTENING
  TCP         0.0.0.0:443            CHEBURASHKA:0          LISTENING
  TCP         0.0.0.0:445            CHEBURASHKA:0          LISTENING
  TCP         0.0.0.0:3389           CHEBURASHKA:0          LISTENING
  TCP         0.0.0.0:10250          CHEBURASHKA:0          LISTENING
  TCP         0.0.0.0:29101          CHEBURASHKA:0          LISTENING
  TCP         0.0.0.0:37985          CHEBURASHKA:0          LISTENING
  TCP         0.0.0.0:49152          CHEBURASHKA:0          LISTENING
  TCP         0.0.0.0:49153          CHEBURASHKA:0          LISTENING
```

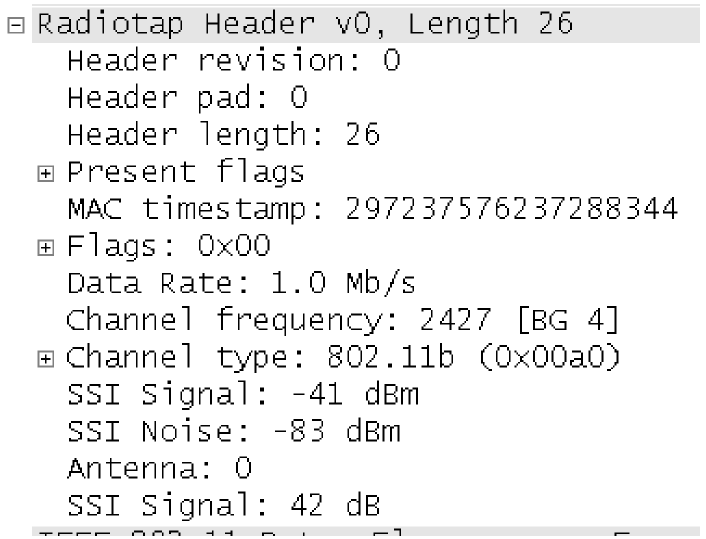
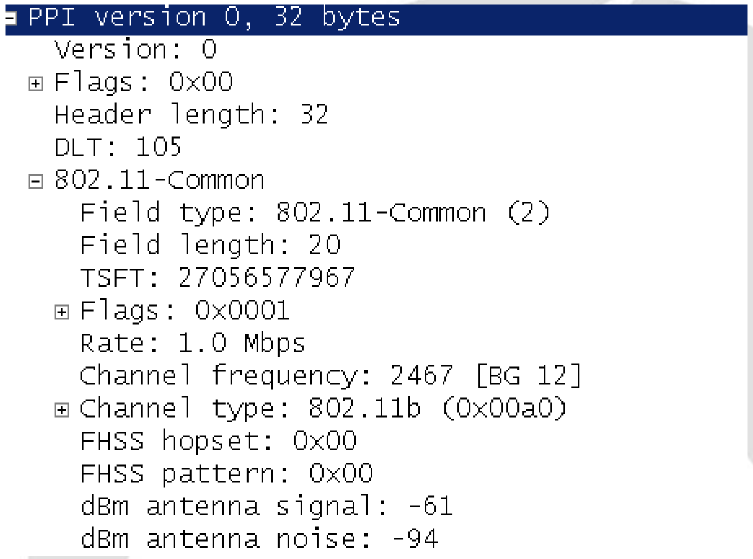


コマンド プロンプト

```
Usage: iperf [-s|-c host] [options]
       iperf [-h|--help] [-v|--version]

Client/Server:
  -f, --format    [kmKM]   format to report: Kbits, Mbits, KBytes, MBytes
  -i, --interval  #        seconds between periodic bandwidth reports
  -l, --len       #[KM]    length of buffer to read or write (default 8 KB)
  -m, --print_mss          print TCP maximum segment size (MTU - TCP/IP header)
  -o, --output    <filename> output the report or error message to this specifie
d file
  -p, --port      #        server port to listen on/connect to
  -u, --udp                use UDP rather than TCP
  -w, --window    #[KM]    TCP window size (socket buffer size)
  -B, --bind      <host>   bind to <host>, an interface or multicast address
  -C, --compatibility      for use with older versions does not sent extra msgs
  -M, --mss       #        set TCP maximum segment size (MTU - 40 bytes)
  -N, --nodelay            set TCP no delay, disabling Nagle's Algorithm
  -V, --IPv6Version        Set the domain to IPv6

Server specific:
  -s, --server             run in server mode
  -D, --daemon             run the server as a daemon
  -R, --remove             remove service in win32

Client specific:
```

- Demonstration

#3 Choosing Physical header type

Type	Radiotap	PPI
Packet		

```
⊟ Radiotap Header v0, Length 26
    Header revision: 0
    Header pad: 0
    Header length: 26
  ⊞ Present flags
    MAC timestamp: 297237576237288344
  ⊞ Flags: 0x00
    Data Rate: 1.0 Mb/s
    Channel frequency: 2427 [BG 4]
  ⊞ Channel type: 802.11b (0x00a0)
    SSI Signal: -41 dBm
    SSI Noise: -83 dBm
    Antenna: 0
    SSI Signal: 42 dB
```

```
⊟ PPI version 0, 32 bytes
    Version: 0
  ⊞ Flags: 0x00
    Header length: 32
    DLT: 105
  ⊟ 802.11-Common
      Field type: 802.11-Common (2)
      Field length: 20
      TSFT: 27056577967
    ⊞ Flags: 0x0001
      Rate: 1.0 Mbps
      Channel frequency: 2467 [BG 12]
    ⊞ Channel type: 802.11b (0x00a0)
      FHSS hopset: 0x00
      FHSS pattern: 0x00
      dBm antenna signal: -61
      dBm antenna noise: -94
```

We can capture wireless frames as 2 kinds of frame format in Physical layer using AirPcap and Wireshark

#3 Choosing Physical header type

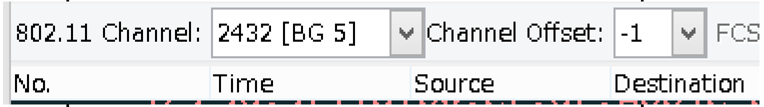
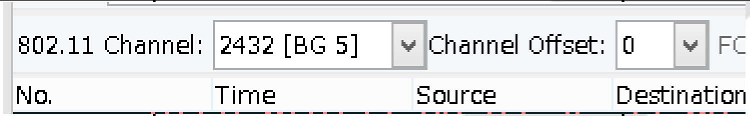
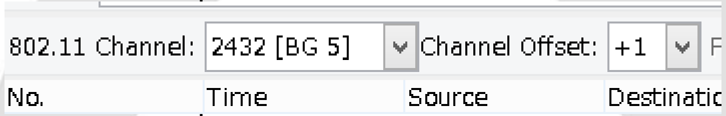
Type	Radiotap	PPI
GOOD	• Easy to read, simple • Fixed format • Easy filter radiotap.dbm_antsignal	• Extensible format future info 11ac, etc • Includes multiple antenna information
BAD	• Cannot collect multiple anntena information	• Hard to read, complex • Long filter ppi.80211n-mac-phy.dbmant0.signal

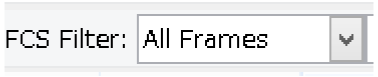
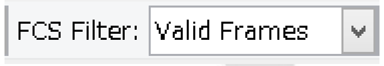
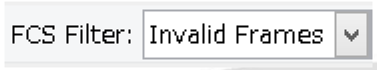
- RECOMMEND Radiotap in 11a/b/g/n(20MHz)
- Demonstration Wireless toolbar> setting

#4 Using AirPcap(s)

- Using multiple AirPcaps tell us a different discovery of target devices (multiple channel info)
- We can use different PC with an AirPcap capturing specific channel (then merge pcap files)
- Trying 3 times or more sometimes AirPcap could not capture the packet.

#4 Using AirPcap(s)

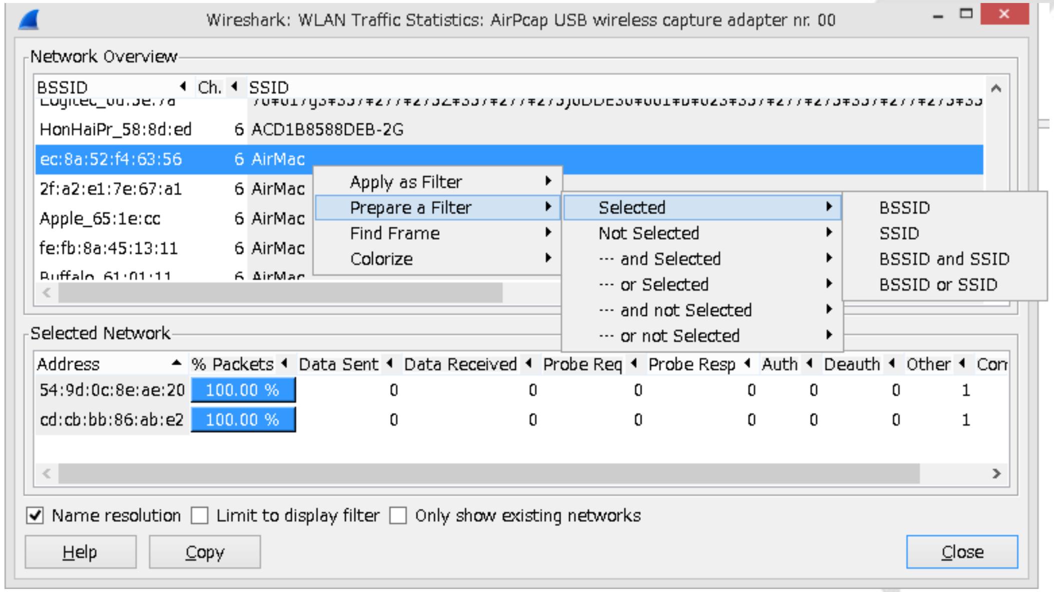
Setting	Offset -1	Offset 0	Offset +1
Channel	Main Channel 5 + Sub 1 1+5(40MHz)	Channel 5 (20MHz)	Main Channel 5 + Sub 9 5+9(40MHz)
	 802.11 Channel: 2432 [BG 5] Channel Offset: -1 FCS No. Time Source Destination	 802.11 Channel: 2432 [BG 5] Channel Offset: 0 FC No. Time Source Destination	 802.11 Channel: 2432 [BG 5] Channel Offset: +1 F No. Time Source Destinatic

Setting	All Frame	Valid Frame	Invalid Frame
	 FCS Filter: All Frames	 FCS Filter: Valid Frames	 FCS Filter: Invalid Frames

- Demonstration

#5 Filtering packet in rough

- Wireless trace file is big,
  Connected wireless trace files are huge.
- Using Statistics>WLAN Traffic
  is the best way to filter packet in rough



Wireshark: WLAN Traffic Statistics: AirPcap USB wireless capture adapter nr. 00

Network Overview

BSSID	Ch.	SSID
[illegible]		[illegible]
HonHaiPr_58:8d:ed	6	ACD1B8588DEB-2G
ec:8a:52:f4:63:56	6	AirMac
2f:a2:e1:7e:67:a1	6	AirMac
Apple_65:1e:cc	6	AirMac
fe:fb:8a:45:13:11	6	AirMac
[illegible]	6	AirMac

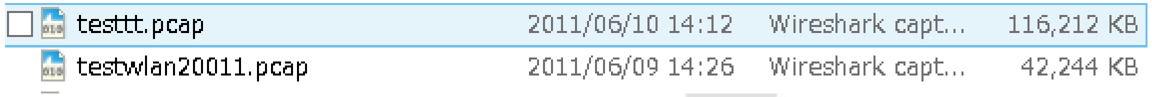
Selected Network

Address	% Packets	Data Sent	Data Received	Probe Req	Probe Resp	Auth	Deauth	Other	Con
54:9d:0c:8e:ae:20	100.00 %	0	0	0	0	0	0	1	
cd:cb:bb:86:ab:e2	100.00 %	0	0	0	0	0	0	1	

[✓] Name resolution ☐ Limit to display filter ☐ Only show existing networks

#5 Filtering packet in rough

- Once filter, or Mark packets or something, then File>Export specified packets.
- Iteration of exporting trace file,
  we can go back, look up the IO Graph, filtered packets at the moment.
- Small trace file is also good to open and read

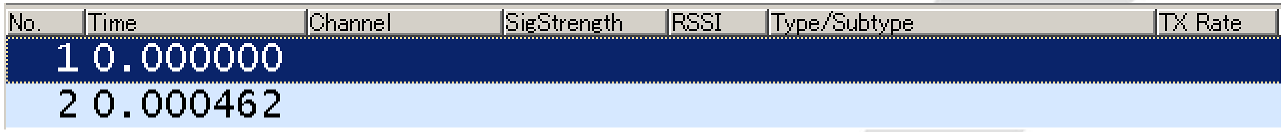


```
testtt.pcap          2011/06/10 14:12   Wireshark capt...   116,212 KB
testwlan20011.pcap   2011/06/09 14:26   Wireshark capt...    42,244 KB
```

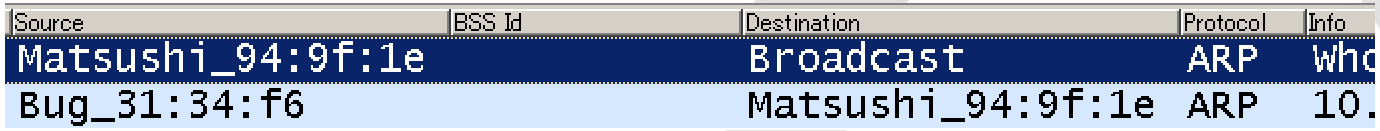
- Demonstration

#6 customizing summary pane

- Summary pane is the first chance to find the important packet
- Choosing field, right click to Apply as Column



No.	Time	Channel	SigStrength	RSSI	Type/Subtype	TX Rate
1	0.000000					
2	0.000462					

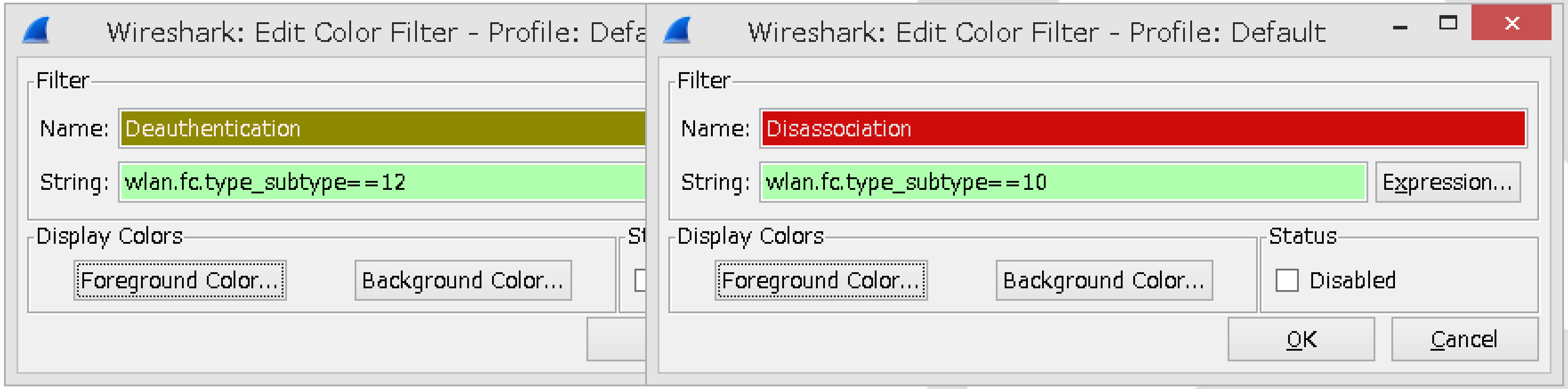


Source	BSS Id	Destination	Protocol	Info
Matsushi_94:9f:1e		Broadcast	ARP	Wh
Bug_31:34:f6		Matsushi_94:9f:1e	ARP	10.

- Type/Subtype … absolutely Apply as Column
  Channel / RSSI / SigStrength / TX Rate …

#7 customizing coloring rules

- typical troublesome packet
  Deauthentication   from AP or from Client
  wlan.fc.type_subtype==12
  Disassociation   from AP or from Client
  wlan.fc.type_subtype==10



Wireshark: Edit Color Filter - Profile: Defa

Filter

Name: Deauthentication

String: wlan.fc.type_subtype==12

Display Colors

Wireshark: Edit Color Filter - Profile: Default

Filter

Name: Disassociation

String: wlan.fc.type_subtype==10

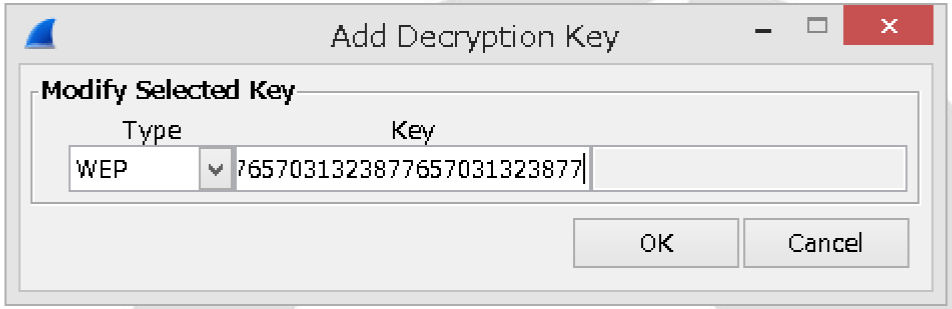
Display Colors

Status

☐ Disabled

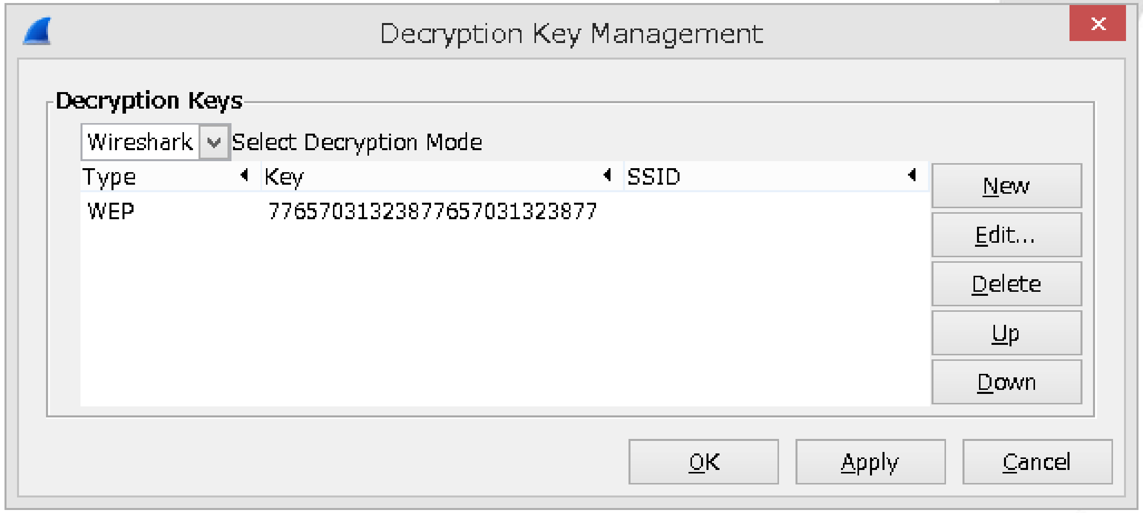
#8 Setting WEP Key

- WEP decryption in Wireshark is easy.
- Any AP, any Client any data frame can be decrypted if the key is correct



Modify Selected Key

Type	Key
WEP	765703132387765703132387 7



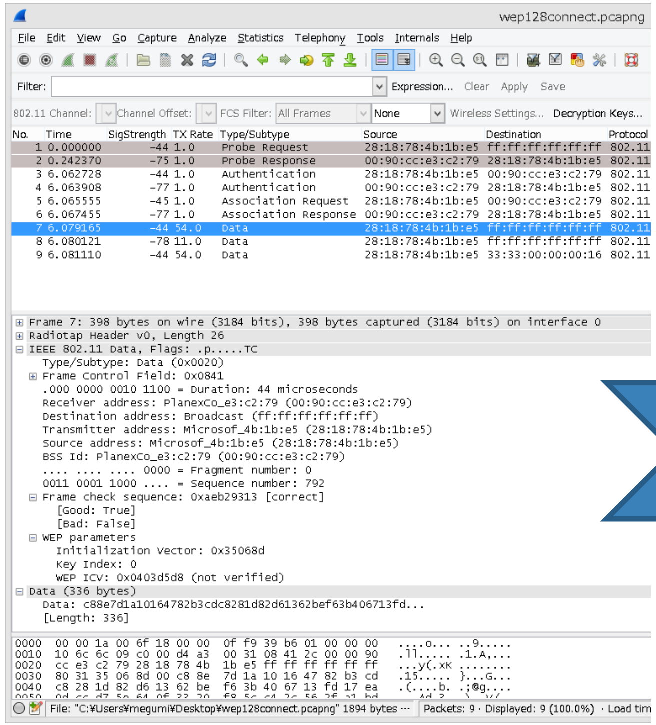
Decryption Keys

Wireshark Select Decryption Mode

Type	Key	SSID
WEP	776570313238776570313238 77	

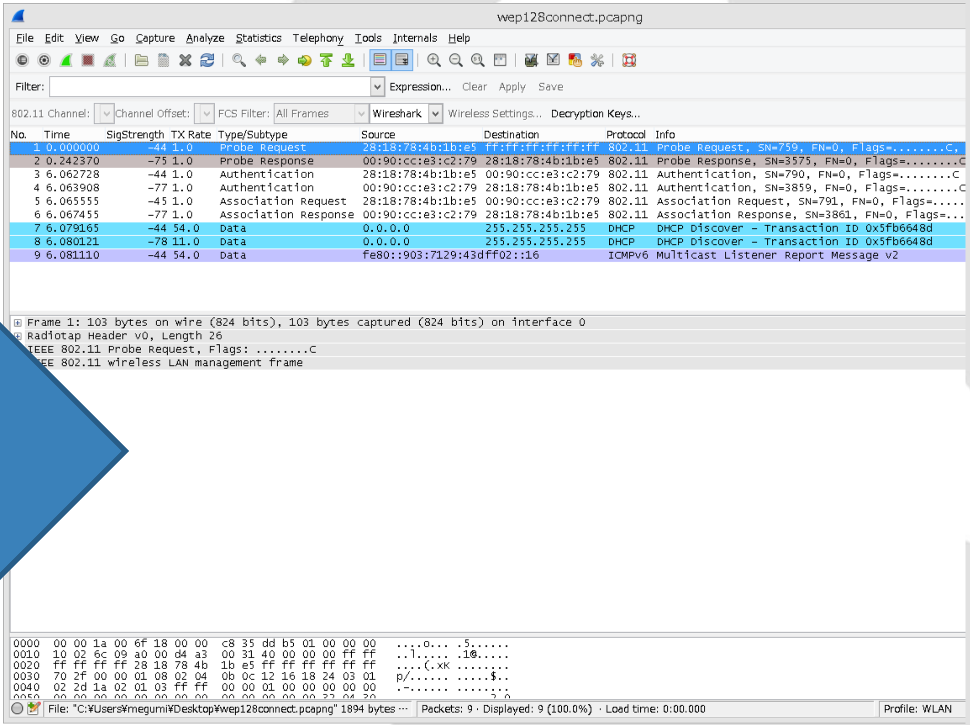
#8 Setting WEP Key

- Remember to enter the key in ASCII format
  wep128wep128w
  77 65 70 31 32 38 77 65 70 31 32 38 77


wep128connect.pcapng
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: Expression... Clear Apply Save
802.11 Channel: Channel Offset: FCS Filter: All Frames None Wireless Settings... Decryption Keys...

```
No. Time      SigStrength TX Rate Type/Subtype          Source            Destination       Protocol
  1 0.000000      -44 1.0   Probe Request         28:18:78:4b:1b:e5 ff:ff:ff:ff:ff:ff 802.11
  2 0.242370      -75 1.0   Probe Response        00:90:cc:e3:c2:79 28:18:78:4b:1b:e5 802.11
  3 6.062728      -44 1.0   Authentication        28:18:78:4b:1b:e5 00:90:cc:e3:c2:79 802.11
  4 6.063908      -77 1.0   Authentication        00:90:cc:e3:c2:79 28:18:78:4b:1b:e5 802.11
  5 6.065555      -45 1.0   Association Request   28:18:78:4b:1b:e5 00:90:cc:e3:c2:79 802.11
  6 6.067455      -77 1.0   Association Response  00:90:cc:e3:c2:79 28:18:78:4b:1b:e5 802.11
  7 6.079165      -44 54.0  Data                  28:18:78:4b:1b:e5 ff:ff:ff:ff:ff:ff 802.11
  8 6.080121      -78 11.0  Data                  28:18:78:4b:1b:e5 ff:ff:ff:ff:ff:ff 802.11
  9 6.081110      -44 54.0  Data                  28:18:78:4b:1b:e5 33:33:00:00:00:16 802.11

⊞ Frame 7: 398 bytes on wire (3184 bits), 398 bytes captured (3184 bits) on interface 0
⊞ Radiotap Header v0, Length 26
⊟ IEEE 802.11 Data, Flags: .p.....TC
    Type/Subtype: Data (0x0020)
  ⊞ Frame Control Field: 0x0841
    .000 0000 0010 1100 = Duration: 44 microseconds
    Receiver address: PlanexCo_e3:c2:79 (00:90:cc:e3:c2:79)
    Destination address: Broadcast (ff:ff:ff:ff:ff:ff)
    Transmitter address: Microsof_4b:1b:e5 (28:18:78:4b:1b:e5)
    Source address: Microsof_4b:1b:e5 (28:18:78:4b:1b:e5)
    BSS Id: PlanexCo_e3:c2:79 (00:90:cc:e3:c2:79)
    .... .... .... 0000 = Fragment number: 0
    0011 0001 1000 .... = Sequence number: 792
  ⊟ Frame check sequence: 0xaeb29313 [correct]
      [Good: True]
      [Bad: False]
  ⊟ WEP parameters
      Initialization Vector: 0x35068d
      Key Index: 0
      WEP ICV: 0x0403d5d8 (not verified)
⊟ Data (336 bytes)
    Data: c88e7d1a10164782b3cdc8281d82d61362bef63b406713fd...
    [Length: 336]

0000  00 00 1a 00 6f 18 00 00  0f f9 39 b6 01 00 00 00   ....o... ..9.....
0010  10 6c 6c 09 c0 00 d4 a3  00 31 08 41 2c 00 00 90   .ll..... .1.A,...
0020  cc e3 c2 79 28 18 78 4b  1b e5 ff ff ff ff ff ff   ...y(.xK ........
0030  80 31 35 06 8d 00 c8 8e  7d 1a 10 16 47 82 b3 cd   .15..... }...G...
0040  c8 28 1d 82 d6 13 62 be  f6 3b 40 67 13 fd 17 ea   .(....b. .;@g....
```

File: "C:¥Users¥megumi¥Desktop¥wep128connect.pcapng" 1894 bytes ··· Packets: 9 · Displayed: 9 (100.0%) · Load tim

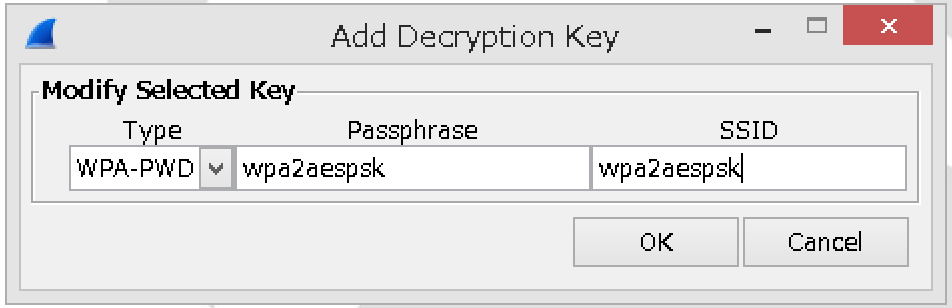

wep128connect.pcapng
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: Expression... Clear Apply Save
802.11 Channel: Channel Offset: FCS Filter: All Frames Wireshark Wireless Settings... Decryption Keys...

```
No. Time      SigStrength TX Rate Type/Subtype          Source            Destination       Protocol Info
  1 0.000000      -44 1.0   Probe Request         28:18:78:4b:1b:e5 ff:ff:ff:ff:ff:ff 802.11  Probe Request, SN=759, FN=0, Flags=........C,
  2 0.242370      -75 1.0   Probe Response        00:90:cc:e3:c2:79 28:18:78:4b:1b:e5 802.11  Probe Response, SN=3575, FN=0, Flags=........C
  3 6.062728      -44 1.0   Authentication        28:18:78:4b:1b:e5 00:90:cc:e3:c2:79 802.11  Authentication, SN=790, FN=0, Flags=........C
  4 6.063908      -77 1.0   Authentication        00:90:cc:e3:c2:79 28:18:78:4b:1b:e5 802.11  Authentication, SN=3859, FN=0, Flags=........C
  5 6.065555      -45 1.0   Association Request   28:18:78:4b:1b:e5 00:90:cc:e3:c2:79 802.11  Association Request, SN=791, FN=0, Flags=.....
  6 6.067455      -77 1.0   Association Response  00:90:cc:e3:c2:79 28:18:78:4b:1b:e5 802.11  Association Response, SN=3861, FN=0, Flags=...
  7 6.079165      -44 54.0  Data                  0.0.0.0           255.255.255.255   DHCP    DHCP Discover - Transaction ID 0x5fb6648d
  8 6.080121      -78 11.0  Data                  0.0.0.0           255.255.255.255   DHCP    DHCP Discover - Transaction ID 0x5fb6648d
  9 6.081110      -44 54.0  Data                  fe80::903:7129:43dff02::16          ICMPv6  Multicast Listener Report Message v2

⊞ Frame 1: 103 bytes on wire (824 bits), 103 bytes captured (824 bits) on interface 0
⊞ Radiotap Header v0, Length 26
  IEEE 802.11 Probe Request, Flags: ........C
  EE 802.11 wireless LAN management frame

0000  00 00 1a 00 6f 18 00 00  c8 35 dd b5 01 00 00 00   ....o... .5......
0010  10 02 6c 09 a0 00 d4 a3  00 31 40 00 00 00 ff ff   ..l..... .1@.....
0020  ff ff ff ff 28 18 78 4b  1b e5 ff ff ff ff ff ff   ....(.xK ........
0030  70 2f 00 00 01 08 02 04  0b 0c 12 16 18 24 03 01   p/...... .....$..
0040  02 2d 1a 02 01 03 ff ff  00 00 01 00 00 00 00 00   .-...... ........
```

File: "C:¥Users¥megumi¥Desktop¥wep128connect.pcapng" 1894 bytes ··· Packets: 9 · Displayed: 9 (100.0%) · Load time: 0:00.000 Profile: WLAN

#9 Setting WPA/WPA2 Key

- WPA 1/2 needs both Passphrase and SSID key input in alphabet format. (or PMK 256bit Hex)
- The difficulties lies in EAPOL 4-way handshake. The complete 4 packet of a series of handshake is necessary for decryption.
- Note some Windows and IOS use **the cache information** of the past connection to the AP, in this case, decryption fails.

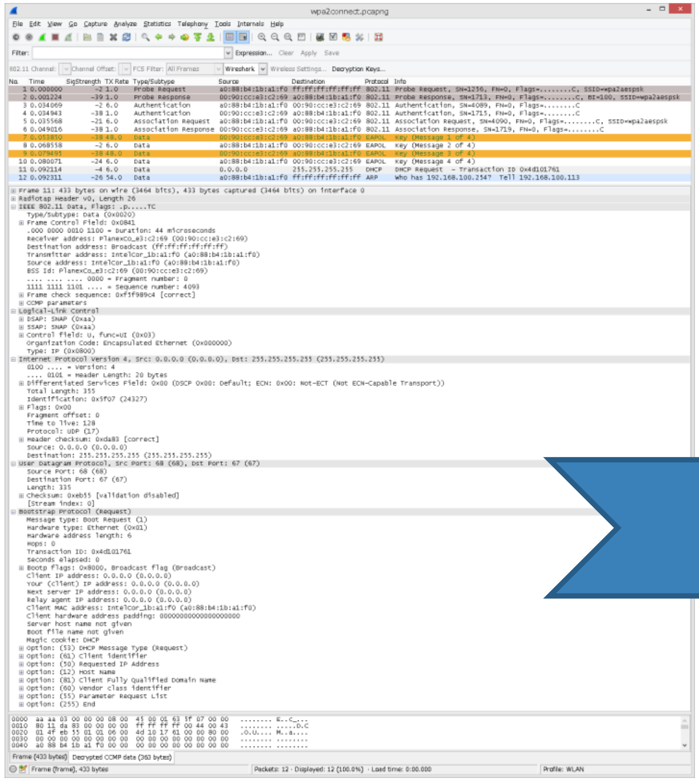

Add Decryption Key

Modify Selected Key

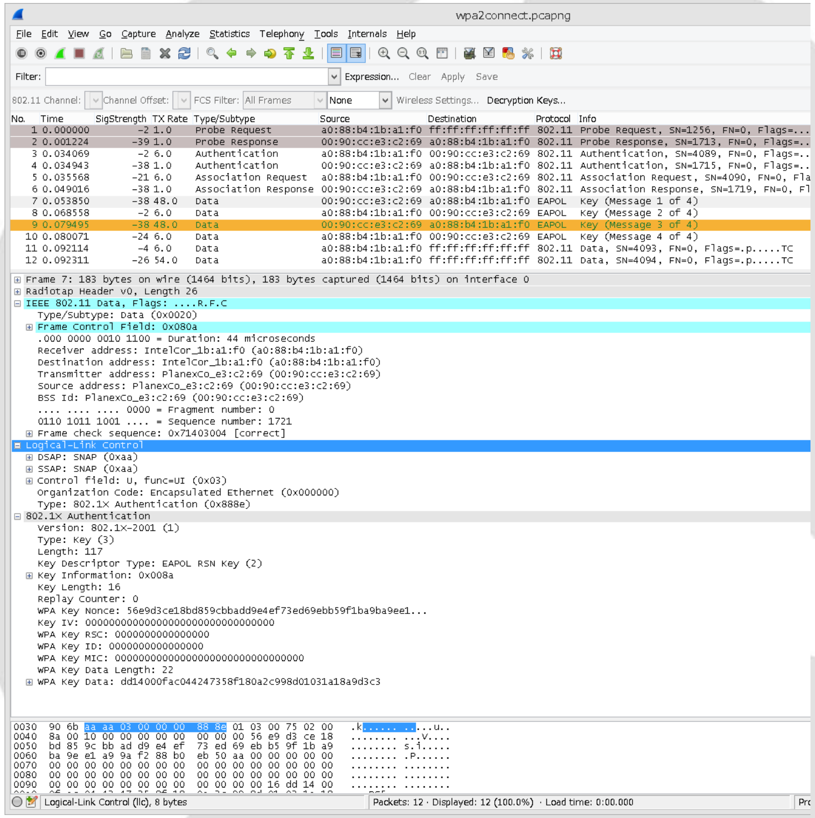
Type	Passphrase	SSID
WPA-PWD	wpa2aespsk	wpa2aespsk

#9 Setting WPA/WPA2 Key

- Please note the complete 4 way handshake
- Key/SSID wpa2aespsk


wpa2connect.pcapng
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: Expression... Clear Apply Save
802.11 Channel: Channel Offset: FCS Filter: All Frames Wireshark Wireless Settings... Decryption Keys...

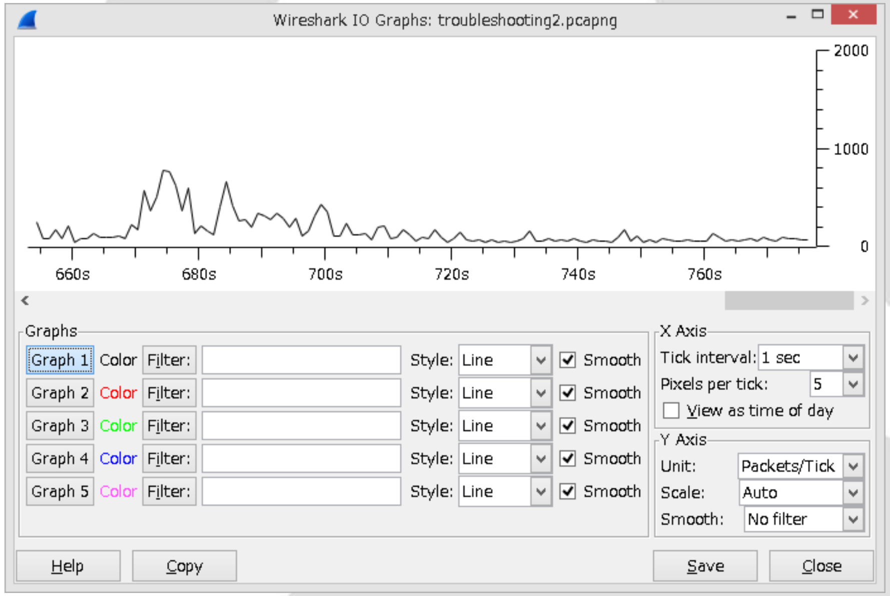
```
No. Time      SigStrength TX Rate Type/Subtype         Source            Destination       Protocol Info
  1 0.000000          -2 1.0  Probe Request        a0:88:b4:1b:a1:f0 ff:ff:ff:ff:ff:ff 802.11 Probe Request, SN=1256, FN=0, Flags=........C, SSID=wpa2aespsk
  2 0.001224         -39 1.0  Probe Response       00:90:cc:e3:c2:69 a0:88:b4:1b:a1:f0 802.11 Probe Response, SN=1713, FN=0, Flags=........C, BI=100, SSID=wpa2aespsk
  3 0.034069          -2 6.0  Authentication       a0:88:b4:1b:a1:f0 00:90:cc:e3:c2:69 802.11 Authentication, SN=4089, FN=0, Flags=........C
  4 0.034943         -38 1.0  Authentication       00:90:cc:e3:c2:69 a0:88:b4:1b:a1:f0 802.11 Authentication, SN=1715, FN=0, Flags=........C
  5 0.035568         -21 6.0  Association Request  a0:88:b4:1b:a1:f0 00:90:cc:e3:c2:69 802.11 Association Request, SN=4090, FN=0, Flags=........C, SSID=wpa2aespsk
  6 0.049016         -38 1.0  Association Response 00:90:cc:e3:c2:69 a0:88:b4:1b:a1:f0 802.11 Association Response, SN=1719, FN=0, Flags=........C
  7 0.053850         -38 48.0 Data                 00:90:cc:e3:c2:69 a0:88:b4:1b:a1:f0 EAPOL  Key (Message 1 of 4)
  8 0.068558          -2 6.0  Data                 a0:88:b4:1b:a1:f0 00:90:cc:e3:c2:69 EAPOL  Key (Message 2 of 4)
  9 0.079495         -38 48.0 Data                 00:90:cc:e3:c2:69 a0:88:b4:1b:a1:f0 EAPOL  Key (Message 3 of 4)
 10 0.080071         -24 6.0  Data                 a0:88:b4:1b:a1:f0 00:90:cc:e3:c2:69 EAPOL  Key (Message 4 of 4)
 11 0.092114          -4 6.0  Data                 0.0.0.0           255.255.255.255   DHCP   DHCP Request  - Transaction ID 0x4d101761
 12 0.092311         -26 54.0 Data                 a0:88:b4:1b:a1:f0 ff:ff:ff:ff:ff:ff ARP    Who has 192.168.100.254?  Tell 192.168.100.113

⊞ Frame 11: 433 bytes on wire (3464 bits), 433 bytes captured (3464 bits) on interface 0
⊞ Radiotap Header v0, Length 26
⊟ IEEE 802.11 Data, Flags: .p.....TC
    Type/Subtype: Data (0x0020)
  ⊞ Frame Control Field: 0x0841
    .000 0000 0010 1100 = Duration: 44 microseconds
    Receiver address: PlanexCo_e3:c2:69 (00:90:cc:e3:c2:69)
    Destination address: Broadcast (ff:ff:ff:ff:ff:ff)
    Transmitter address: IntelCor_1b:a1:f0 (a0:88:b4:1b:a1:f0)
    Source address: IntelCor_1b:a1:f0 (a0:88:b4:1b:a1:f0)
    BSS Id: PlanexCo_e3:c2:69 (00:90:cc:e3:c2:69)
    .... .... .... 0000 = Fragment number: 0
    1111 1111 1101 .... = Sequence number: 4093
  ⊞ Frame check sequence: 0xf5f989c4 [correct]
  ⊞ CCMP parameters
⊟ Logical-Link Control
  ⊞ DSAP: SNAP (0xaa)
  ⊞ SSAP: SNAP (0xaa)
  ⊞ Control field: U, func=UI (0x03)
    Organization Code: Encapsulated Ethernet (0x000000)
    Type: IP (0x0800)
⊟ Internet Protocol Version 4, Src: 0.0.0.0 (0.0.0.0), Dst: 255.255.255.255 (255.255.255.255)
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
    Total Length: 355
    Identification: 0x5f07 (24327)
  ⊞ Flags: 0x00
    Fragment offset: 0
    Time to live: 128
    Protocol: UDP (17)
  ⊞ Header checksum: 0xda83 [correct]
    Source: 0.0.0.0 (0.0.0.0)
    Destination: 255.255.255.255 (255.255.255.255)
⊟ User Datagram Protocol, Src Port: 68 (68), Dst Port: 67 (67)
    Source Port: 68 (68)
    Destination Port: 67 (67)
    Length: 335
  ⊞ Checksum: 0xeb55 [validation disabled]
    [Stream index: 0]
⊟ Bootstrap Protocol (Request)
    Message type: Boot Request (1)
    Hardware type: Ethernet (0x01)
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0x4d101761
    Seconds elapsed: 0
  ⊞ Bootp flags: 0x8000, Broadcast flag (Broadcast)
    Client IP address: 0.0.0.0 (0.0.0.0)
    Your (client) IP address: 0.0.0.0 (0.0.0.0)
    Next server IP address: 0.0.0.0 (0.0.0.0)
    Relay agent IP address: 0.0.0.0 (0.0.0.0)
    Client MAC address: IntelCor_1b:a1:f0 (a0:88:b4:1b:a1:f0)
    Client hardware address padding: 00000000000000000000
    Server host name not given
    Boot file name not given
    Magic cookie: DHCP
  ⊞ Option: (53) DHCP Message Type (Request)
  ⊞ Option: (61) Client identifier
  ⊞ Option: (50) Requested IP Address
  ⊞ Option: (12) Host Name
  ⊞ Option: (81) Client Fully Qualified Domain Name
  ⊞ Option: (60) Vendor class identifier
  ⊞ Option: (55) Parameter Request List
  ⊞ Option: (255) End

0000  aa aa 03 00 00 00 08 00  45 00 01 63 5f 07 00 00   ........ E..c_...
0010  80 11 da 83 00 00 00 00  ff ff ff ff 00 44 00 43   ........ .....D.C
0020  01 4f eb 55 01 01 06 00  4d 10 17 61 00 00 80 00   .O.U.... M..a....
0030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0040  a0 88 b4 1b a1 f0 00 00  00 00 00 00 00 00 00 00   ........ ........

Frame (433 bytes)  Decrypted CCMP data (363 bytes)
Frame (frame), 433 bytes    Packets: 12 · Displayed: 12 (100.0%) · Load time: 0:00.000    Profile: WLAN
```


wpa2connect.pcapng
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: Expression... Clear Apply Save
802.11 Channel: Channel Offset: FCS Filter: All Frames None Wireless Settings... Decryption Keys...

```
No. Time      SigStrength TX Rate Type/Subtype         Source            Destination       Protocol Info
  1 0.000000          -2 1.0  Probe Request        a0:88:b4:1b:a1:f0 ff:ff:ff:ff:ff:ff 802.11 Probe Request, SN=1256, FN=0, Flags=...
  2 0.001224         -39 1.0  Probe Response       00:90:cc:e3:c2:69 a0:88:b4:1b:a1:f0 802.11 Probe Response, SN=1713, FN=0, Flags=..
  3 0.034069          -2 6.0  Authentication       a0:88:b4:1b:a1:f0 00:90:cc:e3:c2:69 802.11 Authentication, SN=4089, FN=0, Flags=..
  4 0.034943         -38 1.0  Authentication       00:90:cc:e3:c2:69 a0:88:b4:1b:a1:f0 802.11 Authentication, SN=1715, FN=0, Flags=..
  5 0.035568         -21 6.0  Association Request  a0:88:b4:1b:a1:f0 00:90:cc:e3:c2:69 802.11 Association Request, SN=4090, FN=0, Fla
  6 0.049016         -38 1.0  Association Response 00:90:cc:e3:c2:69 a0:88:b4:1b:a1:f0 802.11 Association Response, SN=1719, FN=0, Fl
  7 0.053850         -38 48.0 Data                 00:90:cc:e3:c2:69 a0:88:b4:1b:a1:f0 EAPOL  Key (Message 1 of 4)
  8 0.068558          -2 6.0  Data                 a0:88:b4:1b:a1:f0 00:90:cc:e3:c2:69 EAPOL  Key (Message 2 of 4)
  9 0.079495         -38 48.0 Data                 00:90:cc:e3:c2:69 a0:88:b4:1b:a1:f0 EAPOL  Key (Message 3 of 4)
 10 0.080071         -24 6.0  Data                 a0:88:b4:1b:a1:f0 00:90:cc:e3:c2:69 EAPOL  Key (Message 4 of 4)
 11 0.092114          -4 6.0  Data                 a0:88:b4:1b:a1:f0 ff:ff:ff:ff:ff:ff 802.11 Data, SN=4093, FN=0, Flags=.p.....TC
 12 0.092311         -26 54.0 Data                 a0:88:b4:1b:a1:f0 ff:ff:ff:ff:ff:ff 802.11 Data, SN=4094, FN=0, Flags=.p.....TC

⊞ Frame 7: 183 bytes on wire (1464 bits), 183 bytes captured (1464 bits) on interface 0
⊞ Radiotap Header v0, Length 26
⊟ IEEE 802.11 Data, Flags: ....R.F.C
    Type/Subtype: Data (0x0020)
  ⊞ Frame Control Field: 0x080a
    .000 0000 0010 1100 = Duration: 44 microseconds
    Receiver address: IntelCor_1b:a1:f0 (a0:88:b4:1b:a1:f0)
    Destination address: IntelCor_1b:a1:f0 (a0:88:b4:1b:a1:f0)
    Transmitter address: PlanexCo_e3:c2:69 (00:90:cc:e3:c2:69)
    Source address: PlanexCo_e3:c2:69 (00:90:cc:e3:c2:69)
    BSS Id: PlanexCo_e3:c2:69 (00:90:cc:e3:c2:69)
    .... .... .... 0000 = Fragment number: 0
    0110 1011 1001 .... = Sequence number: 1721
  ⊞ Frame check sequence: 0x71403004 [correct]
⊟ Logical-Link Control
  ⊞ DSAP: SNAP (0xaa)
  ⊞ SSAP: SNAP (0xaa)
  ⊞ Control field: U, func=UI (0x03)
    Organization Code: Encapsulated Ethernet (0x000000)
    Type: 802.1X Authentication (0x888e)
⊟ 802.1X Authentication
    Version: 802.1X-2001 (1)
    Type: Key (3)
    Length: 117
    Key Descriptor Type: EAPOL RSN Key (2)
  ⊞ Key Information: 0x008a
    Key Length: 16
    Replay Counter: 0
    WPA Key Nonce: 56e9d3ce18bd859cbbadd9e4ef73ed69ebb59f1ba9ba9ee1...
    Key IV: 00000000000000000000000000000000
    WPA Key RSC: 0000000000000000
    WPA Key ID: 0000000000000000
    WPA Key MIC: 00000000000000000000000000000000
    WPA Key Data Length: 22
  ⊞ WPA Key Data: dd14000fac044247358f180a2c998d01031a18a9d3c3

0030  90 6b aa aa 03 00 00 00  88 8e 01 03 00 75 02 00   .k...... ....u..
0040  8a 00 10 00 00 00 00 00  00 00 00 56 e9 d3 ce 18   ........ ...V....
0050  bd 85 9c bb ad d9 e4 ef  73 ed 69 eb b5 9f 1b a9   ........ s.i.....
0060  ba 9e e1 a9 9a f2 88 b0  eb 50 aa 00 00 00 00 00   ........ .P......
0070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0080  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0090  00 00 00 00 00 00 00 00  00 00 00 00 16 dd 14 00   ........ ........

Logical-Link Control (llc), 8 bytes    Packets: 12 · Displayed: 12 (100.0%) · Load time: 0:00.000
```

#10 Visualization (1) Retry

- Easy way to check the CSMA/CA status.
- We can check the retry packet rate, as well as the throughput of data frame.
- Filter packet within the specified AP or Client
- Statistics>IO Graph
  Retry rate graph
  Y/X axis -> packet/sec
  Throughput graph
  Y/X axis -> bit/sec



Wireshark IO Graphs: troubleshooting2.pcapng

Graphs

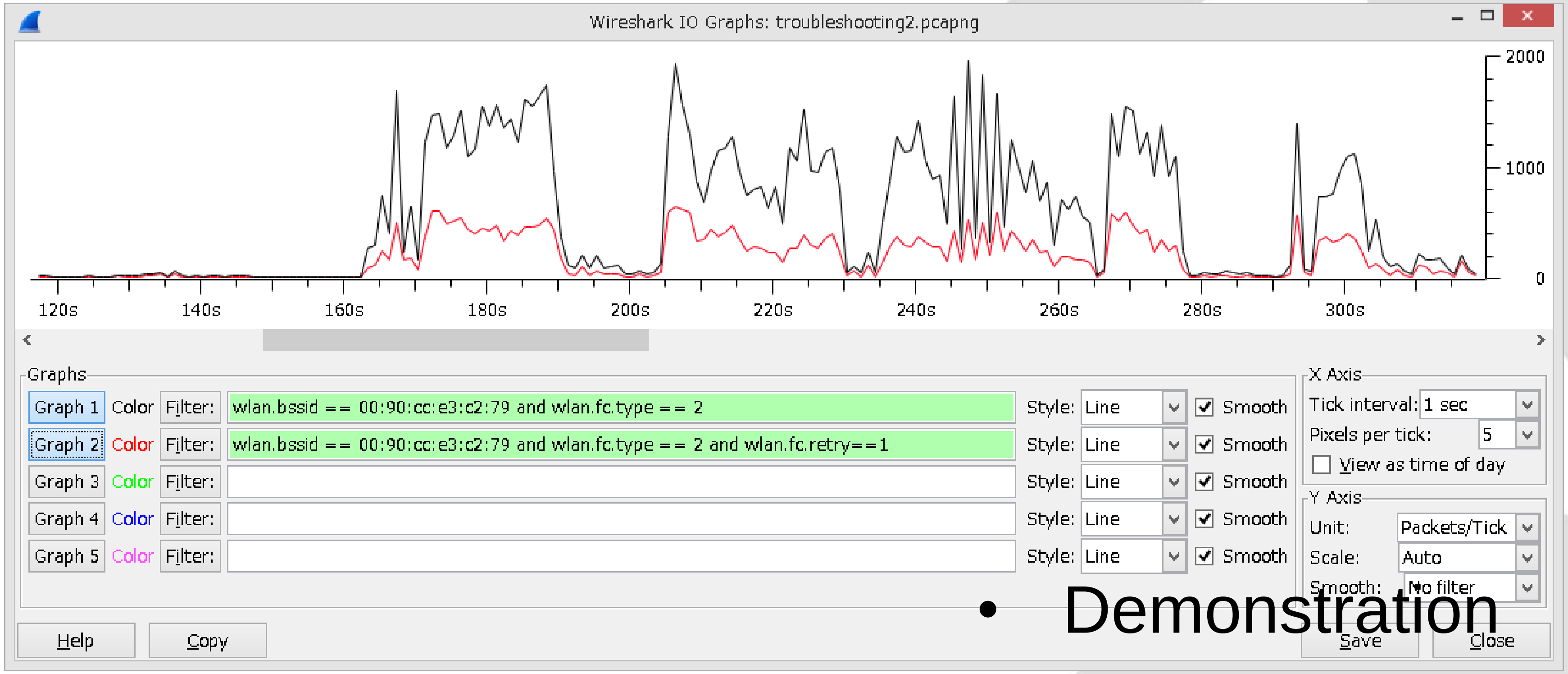
Graph 1	Color	Filter:		Style: Line	☑ Smooth
Graph 2	Color	Filter:		Style: Line	☑ Smooth
Graph 3	Color	Filter:		Style: Line	☑ Smooth
Graph 4	Color	Filter:		Style: Line	☑ Smooth
Graph 5	Color	Filter:		Style: Line	☑ Smooth

X Axis
Tick interval: 1 sec
Pixels per tick: 5
☐ View as time of day

Y Axis
Unit: Packets/Tick
Scale: Auto
Smooth: No filter

#10 Visualization (1) Retry

- Graph1: specified BSSID and data frame
- Graph2: the same with Graph1 and **“wlan.fc.retry==1”**



Graphs

Graph 1	Color	Filter:	wlan.bssid == 00:90:cc:e3:c2:79 and wlan.fc.type == 2	Style:	Line	✓ Smooth
Graph 2	Color	Filter:	wlan.bssid == 00:90:cc:e3:c2:79 and wlan.fc.type == 2 and wlan.fc.retry==1	Style:	Line	✓ Smooth
Graph 3	Color	Filter:		Style:	Line	✓ Smooth
Graph 4	Color	Filter:		Style:	Line	✓ Smooth
Graph 5	Color	Filter:		Style:	Line	✓ Smooth

X Axis
Tick interval: 1 sec
Pixels per tick: 5
☐ View as time of day

Y Axis
Unit: Packets/Tick
Scale: Auto
Smooth: No filter

- Demonstration

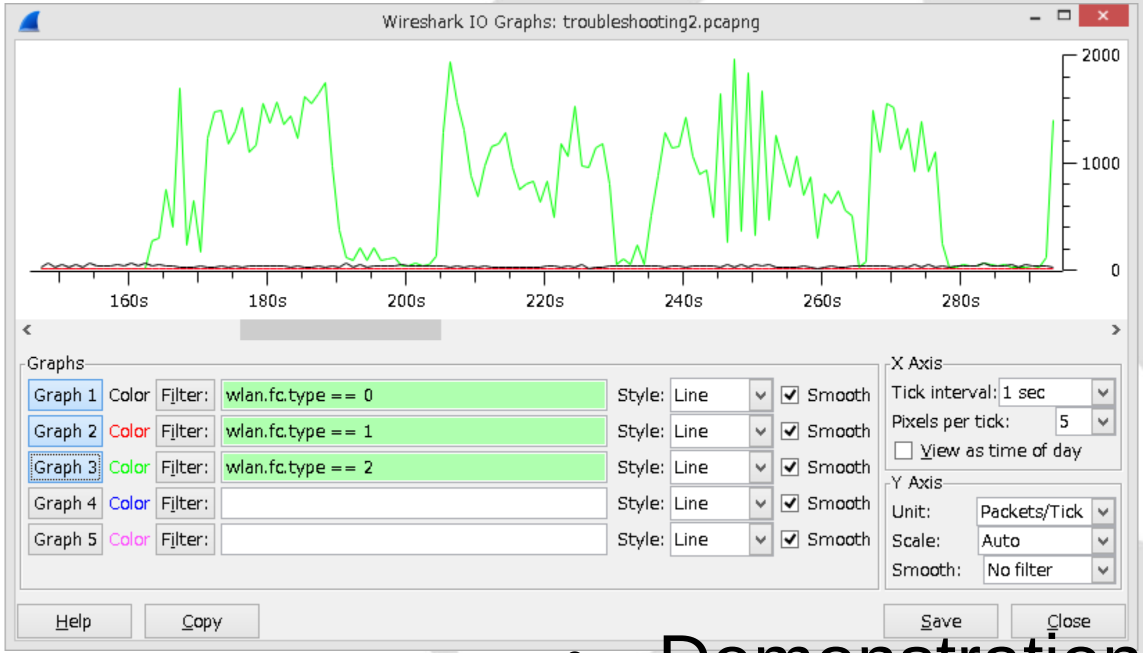
#11 Visualization (2) Frame type

- What type of IEEE802.11 frames in RF is important in analysis the compose of frame tells us the status of RF

Status	Management	Control	Data
IDLE	Many	Few	Few
BUSY (GOOD)	Few	Many same as Data	Many same as Control
BUSY (BAD)	Few	Many less than Data	Many more than Control
RTS/CTS (protect mode)	Few	Many more than Data	Many less than Control

#11 Visualization (2) Frame type

- Management frame wlan.fc.type==0
  Control Frame wlan.fc.type==1
  Data Frame (includes NULL) wlan.fc.type==2
- Statistics>
  IO Graph
  Y/X Axis ->
  packets / sec
- This time is
  BAD RF
  (many retry)



Graphs

Graph 1	Color	Filter: wlan.fc.type == 0	Style: Line	✓ Smooth
Graph 2	Color	Filter: wlan.fc.type == 1	Style: Line	✓ Smooth
Graph 3	Color	Filter: wlan.fc.type == 2	Style: Line	✓ Smooth
Graph 4	Color	Filter:	Style: Line	✓ Smooth
Graph 5	Color	Filter:	Style: Line	✓ Smooth

X Axis
Tick interval: 1 sec
Pixels per tick: 5
☐ View as time of day

Y Axis
Unit: Packets/Tick
Scale: Auto
Smooth: No filter

- Demonstration

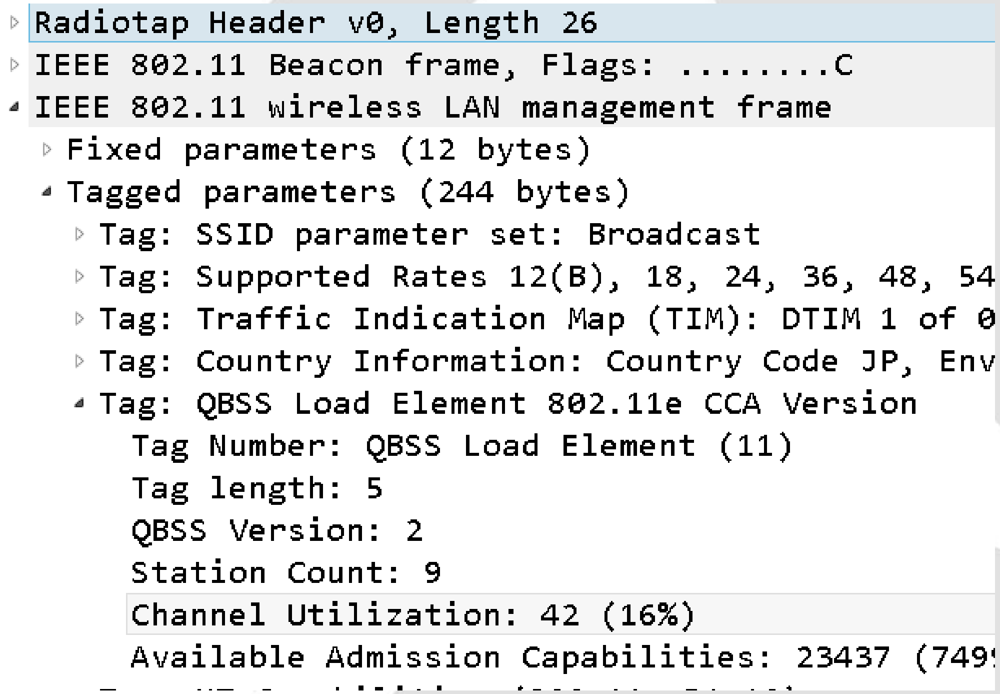
#12 Visualization (3) management

- Management frame contains many good information for debugging and troubleshooting.
- Some AP sends important information in management frame.
- IEEE802.11e has QBSS (QoS Based Service Set) CCA (Clear Channel Assignment) information that contains the number of the connected station and utilization of the channel.

#12 Visualization (3) management

IEEE802.11e Beacon frame contains QBSS Tag
QBS Load Element CCA has the number of the Station and Channel Utilization

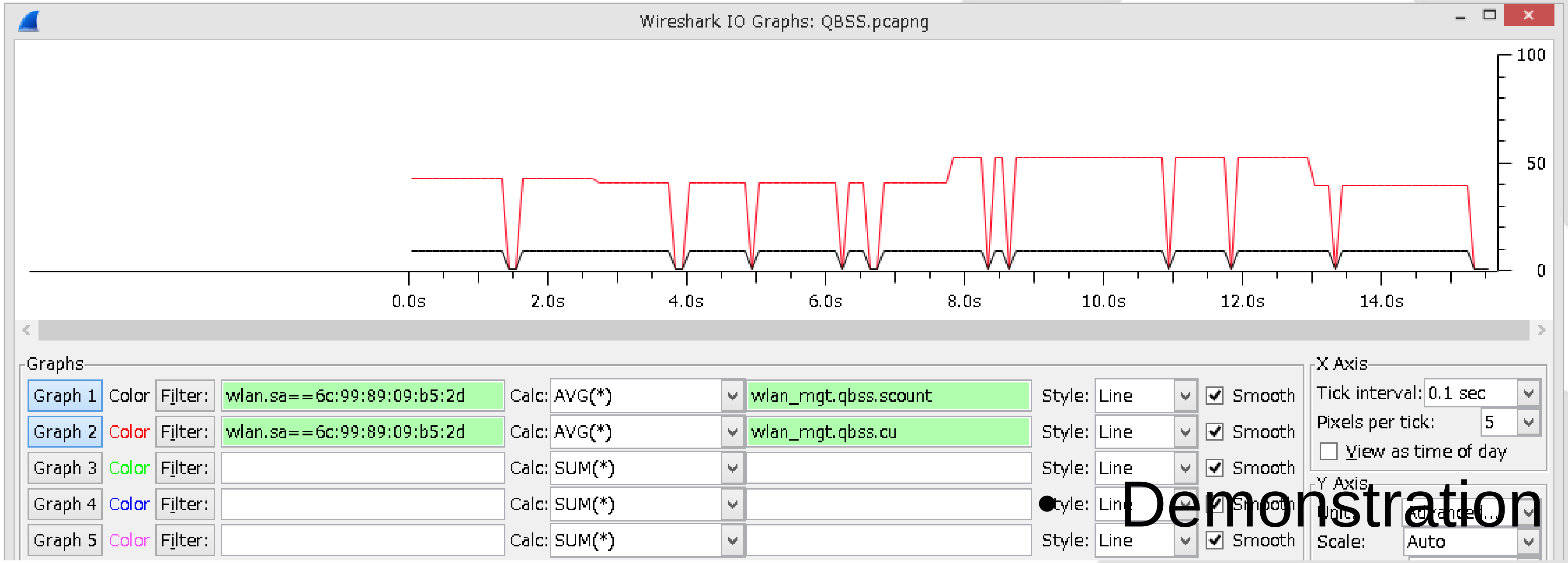
Station Count
wlan_mgt.qbss.scount
Channel Utilization
 wlan_mgt.qbss.cu



```
▷ Radiotap Header v0, Length 26
▷ IEEE 802.11 Beacon frame, Flags: ........C
◢ IEEE 802.11 wireless LAN management frame
  ▷ Fixed parameters (12 bytes)
  ◢ Tagged parameters (244 bytes)
    ▷ Tag: SSID parameter set: Broadcast
    ▷ Tag: Supported Rates 12(B), 18, 24, 36, 48, 54
    ▷ Tag: Traffic Indication Map (TIM): DTIM 1 of 0
    ▷ Tag: Country Information: Country Code JP, Env
    ◢ Tag: QBSS Load Element 802.11e CCA Version
        Tag Number: QBSS Load Element (11)
        Tag length: 5
        QBSS Version: 2
        Station Count: 9
        Channel Utilization: 42 (16%)
        Available Admission Capabilities: 23437 (749
```

#12 Visualization (3) management Visualizing Station and Utilization

- Statistics>IO Graph and set Y Axis to advanced filtering specified AP and use AVG(*) and counting Station(Black) / Utilization (Red)



Wireshark IO Graphs: QBSS.pcapng

Graphs

Graph 1	Color	Filter:	wlan.sa==6c:99:89:09:b5:2d	Calc:	AVG(*)	wlan_mgt.qbss.scount	Style:	Line	✓	Smooth
Graph 2	Color	Filter:	wlan.sa==6c:99:89:09:b5:2d	Calc:	AVG(*)	wlan_mgt.qbss.cu	Style:	Line	✓	Smooth
Graph 3	Color	Filter:		Calc:	SUM(*)		Style:	Line	✓	Smooth
Graph 4	Color	Filter:		Calc:	SUM(*)		Style:	Line	✓	Smooth
Graph 5	Color	Filter:		Calc:	SUM(*)		Style:	Line	✓	Smooth

X Axis

Tick interval: 0.1 sec

Pixels per tick: 5

☐ View as time of day

Y Axis

Unit: Advanced...

Scale: Auto

Demonstration

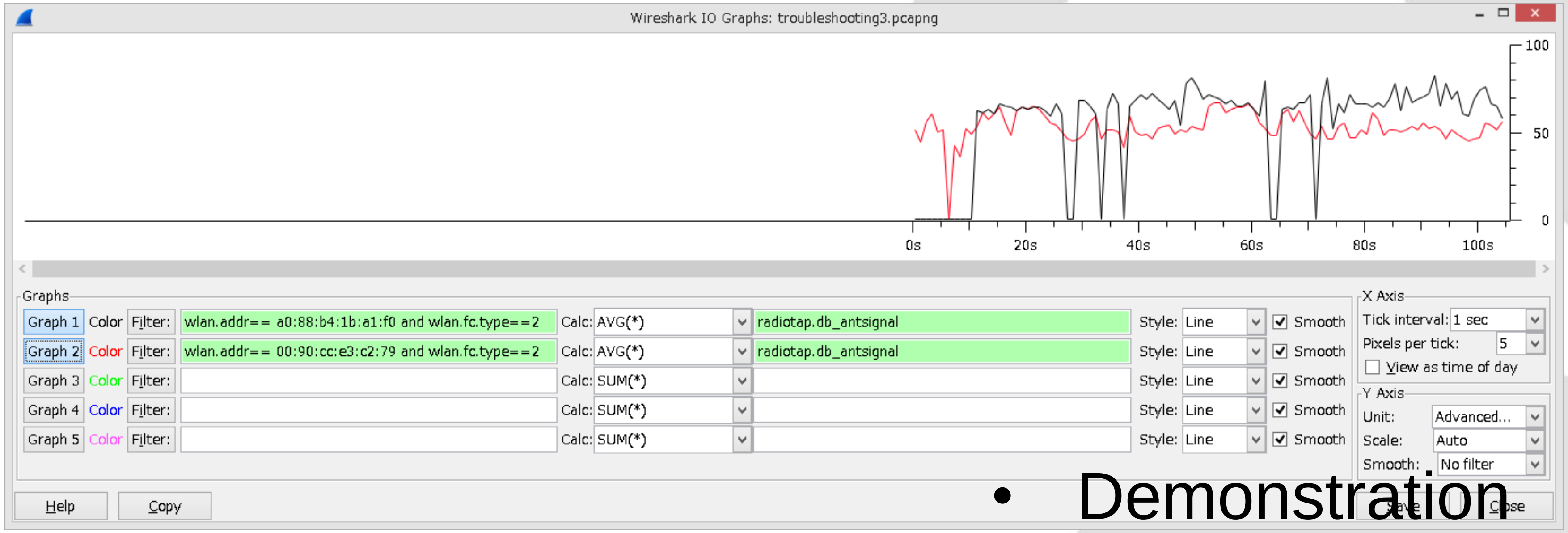
#13 Visualization (4) signal

- Signal / Noise ratio is useful, and good ratio is 20 (signal is 10 times louder than noise)
  20x log 10/1 = 20dB
- AirPcap collect signal info and display filter is radiotap.db_antsignal

dB	multiple
1	1.122018
2	1.258925
3	1.412538
4	1.584893
5	1.778279
6	1.995262
7	2.238721
8	2.511886
9	2.818383
10	3.162278
11	3.548134
12	3.981072
13	4.466836
14	5.011872
15	5.623413
16	6.309573
17	7.079458
18	7.943282
19	8.912509
20	10
21	11.22018
22	12.58925
23	14.12538
24	15.84893
25	17.78279
26	19.95262
27	22.38721
28	25.11886
29	28.18383
30	31.62278
31	35.48134
32	39.81072
33	44.66836
34	50.11872
35	56.23413
36	63.09573
37	70.79458
38	79.43282
39	89.12509
40	100
41	112.2018
42	125.8925
43	141.2538
44	158.4893
45	177.8279
46	199.5262
47	223.8721
48	251.1886
49	281.8383
50	316.2278

#13 Visualization (4) signal

- Statistics > IO Graph and filter AP (Graph1) and filter Client (Graph2) and set Y axis to advanced, then counting AVG(*) of radiotap.db_antsignal



Graphs

Graph 1	Color	Filter:	wlan.addr== a0:88:b4:1b:a1:f0 and wlan.fc.type==2	Calc:	AVG(*)	radiotap.db_antsignal	Style:	Line	✓ Smooth
Graph 2	Color	Filter:	wlan.addr== 00:90:cc:e3:c2:79 and wlan.fc.type==2	Calc:	AVG(*)	radiotap.db_antsignal	Style:	Line	✓ Smooth
Graph 3	Color	Filter:		Calc:	SUM(*)		Style:	Line	✓ Smooth
Graph 4	Color	Filter:		Calc:	SUM(*)		Style:	Line	✓ Smooth
Graph 5	Color	Filter:		Calc:	SUM(*)		Style:	Line	✓ Smooth

X Axis

Tick interval: 1 sec
Pixels per tick: 5
☐ View as time of day

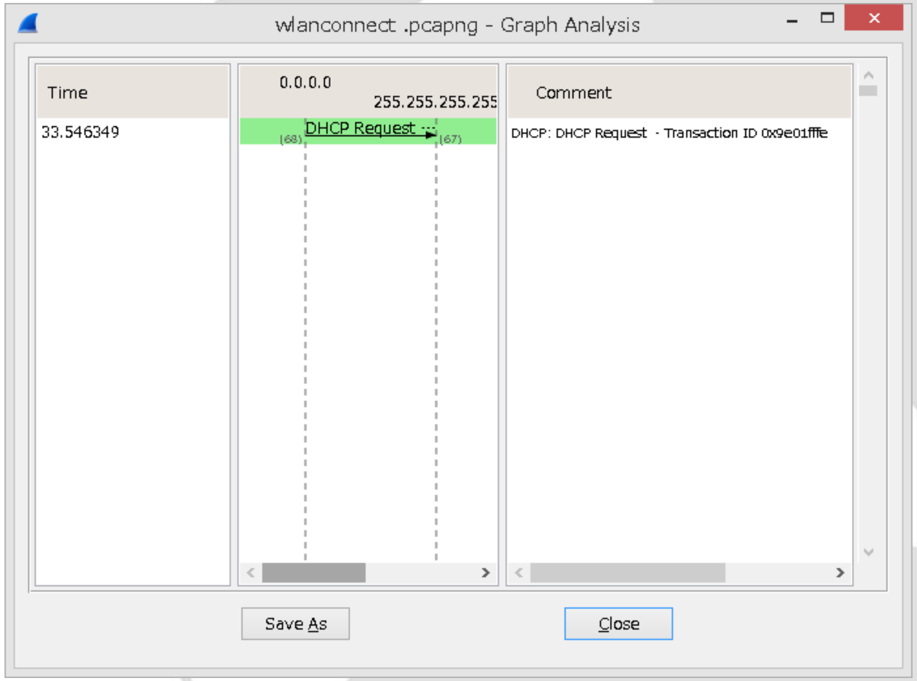
Y Axis

Unit: Advanced...
Scale: Auto
Smooth: No filter

- Demonstration

#14 using flow graph

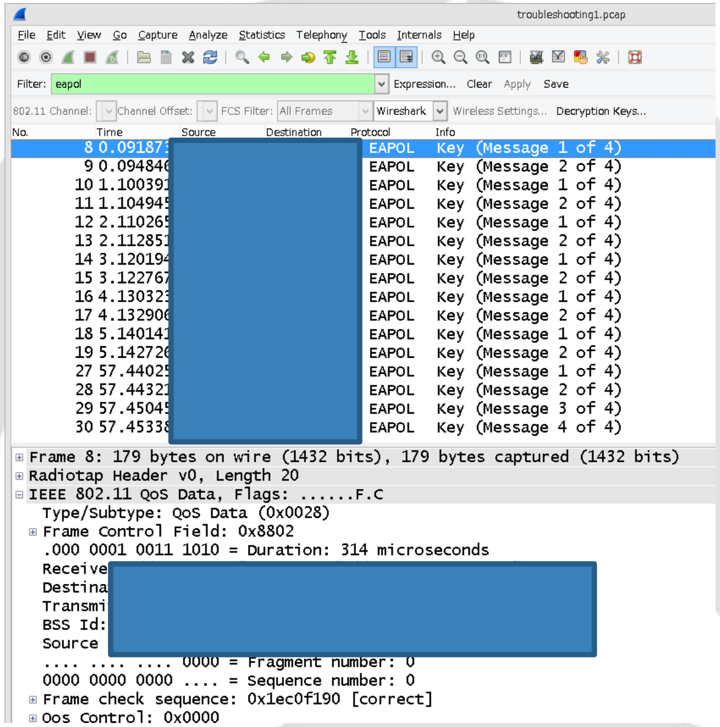
- Flow graph is good Obtusely trouble packet
- If you need to draw Flow Graph under layer2 old version of Wireshark is good.
- Use Wireshark1.6 or older
- Statistics> Flow Graph



Time	0.0.0.0 255.255.255.255	Comment
33.546349	(68) DHCP Request (67)	DHCP: DHCP Request - Transaction ID 0x9e01fffe

#15 Repetition of packets (iOS)

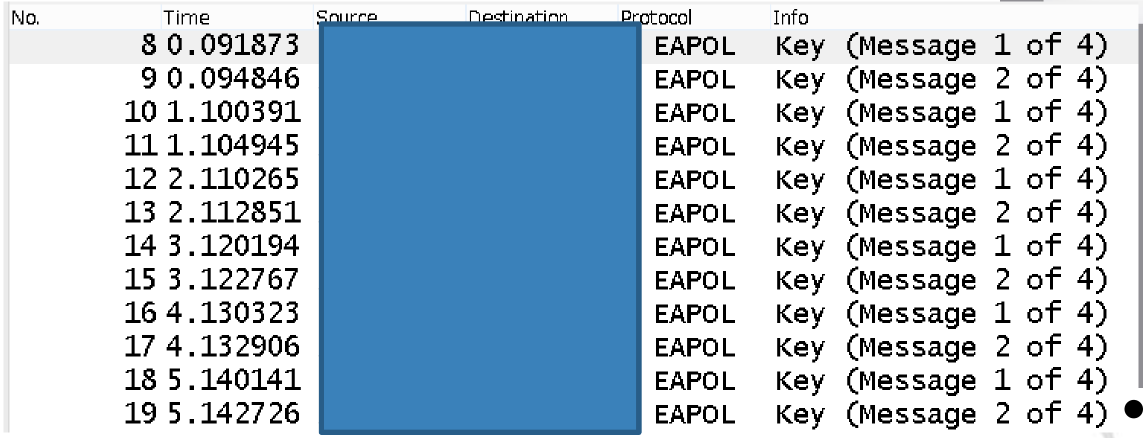
- Repetition of a series of the packet gives us the hint for debugging, troubleshooting.
- This packet contains the repetition that EAPOL(mes1/4) EAPOL(mes2/4) counts 6 times !
- The troubles lies in here.


troubleshooting1.pcap
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: eapol
Expression... Clear Apply Save
802.11 Channel: Channel Offset: FCS Filter: All Frames Wireshark Wireless Settings... Decryption Keys...

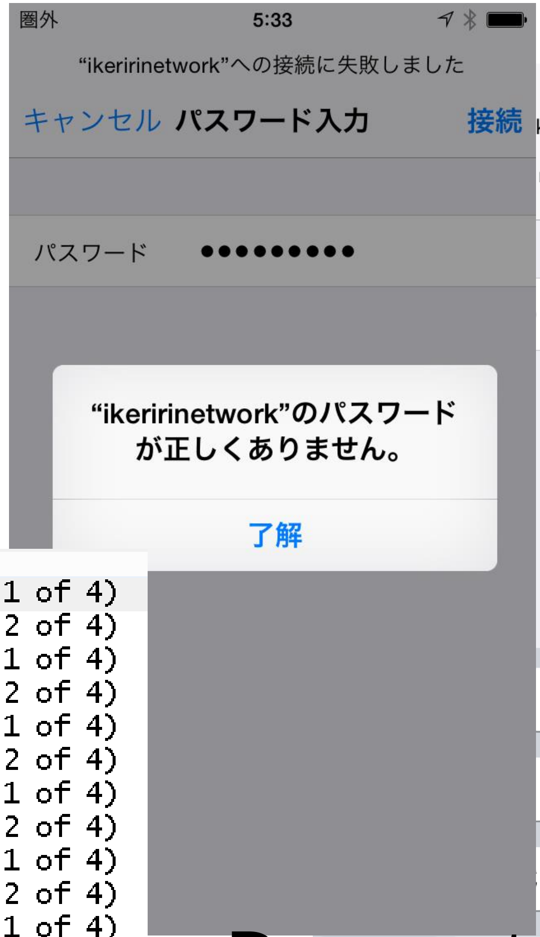
```
No.   Time       Source  Destination  Protocol  Info
 8 0.091873  [redacted]               EAPOL     Key (Message 1 of 4)
 9 0.094846                           EAPOL     Key (Message 2 of 4)
10 1.100391                           EAPOL     Key (Message 1 of 4)
11 1.104945                           EAPOL     Key (Message 2 of 4)
12 2.110265                           EAPOL     Key (Message 1 of 4)
13 2.112851                           EAPOL     Key (Message 2 of 4)
14 3.120194                           EAPOL     Key (Message 1 of 4)
15 3.122767                           EAPOL     Key (Message 2 of 4)
16 4.130323                           EAPOL     Key (Message 1 of 4)
17 4.132906                           EAPOL     Key (Message 2 of 4)
18 5.140141                           EAPOL     Key (Message 1 of 4)
19 5.142726                           EAPOL     Key (Message 2 of 4)
27 57.44025                           EAPOL     Key (Message 1 of 4)
28 57.44321                           EAPOL     Key (Message 2 of 4)
29 57.45045                           EAPOL     Key (Message 3 of 4)
30 57.45338                           EAPOL     Key (Message 4 of 4)

⊞ Frame 8: 179 bytes on wire (1432 bits), 179 bytes captured (1432 bits)
⊞ Radiotap Header v0, Length 20
⊟ IEEE 802.11 QoS Data, Flags: ......F.C
    Type/Subtype: QoS Data (0x0028)
  ⊞ Frame Control Field: 0x8802
    .000 0001 0011 1010 = Duration: 314 microseconds
    Receive [redacted]
    Destina
    Transmi
    BSS Id:
    Source
    .... .... .... 0000 = Fragment number: 0
    0000 0000 0000 .... = Sequence number: 0
  ⊞ Frame check sequence: 0x1ec0f190 [correct]
  ⊞ Qos Control: 0x0000
```

#15 Repetition of packets (iOS)

- Wrong passphrase causes network error of EAPOL 4-way handshake.
- iOS tried 6 times.



No.	Time	Source	Destination	Protocol	Info
8	0.091873	[redacted]	[redacted]	EAPOL	Key (Message 1 of 4)
9	0.094846	[redacted]	[redacted]	EAPOL	Key (Message 2 of 4)
10	1.100391	[redacted]	[redacted]	EAPOL	Key (Message 1 of 4)
11	1.104945	[redacted]	[redacted]	EAPOL	Key (Message 2 of 4)
12	2.110265	[redacted]	[redacted]	EAPOL	Key (Message 1 of 4)
13	2.112851	[redacted]	[redacted]	EAPOL	Key (Message 2 of 4)
14	3.120194	[redacted]	[redacted]	EAPOL	Key (Message 1 of 4)
15	3.122767	[redacted]	[redacted]	EAPOL	Key (Message 2 of 4)
16	4.130323	[redacted]	[redacted]	EAPOL	Key (Message 1 of 4)
17	4.132906	[redacted]	[redacted]	EAPOL	Key (Message 2 of 4)
18	5.140141	[redacted]	[redacted]	EAPOL	Key (Message 1 of 4)
19	5.142726	[redacted]	[redacted]	EAPOL	Key (Message 2 of 4)



圏外 5:33

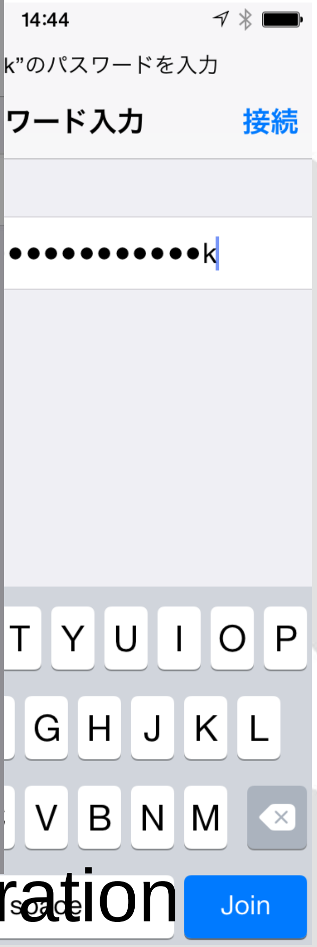
“ikeririnetwork”への接続に失敗しました

キャンセル パスワード入力 接続

パスワード ●●●●●●●●●●

“ikeririnetwork”のパスワードが正しくありません。

了解



14:44

k”のパスワードを入力

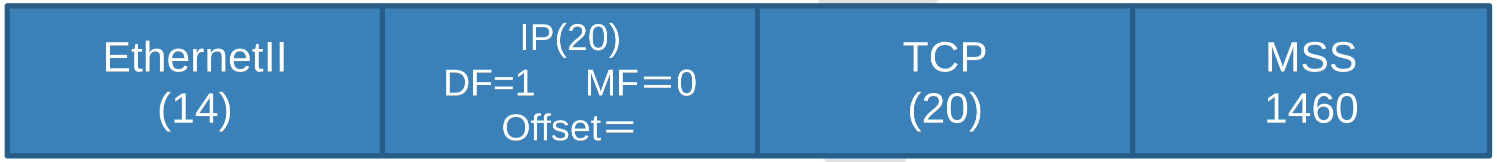
ワード入力 接続

●●●●●●●●●●●●k

- Demonstration

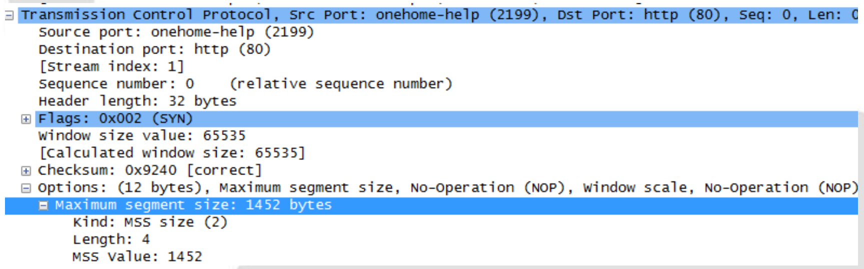
#16 Wireless Router's MTU/MSS

- Some user says they cannot see specific website. (ex. Google OK Yahoo NG)
- When MTU 1454 (default), we cannot see But MTU 1414, and we CAN SEE



#16 Wireless Router’s MTU/MSS

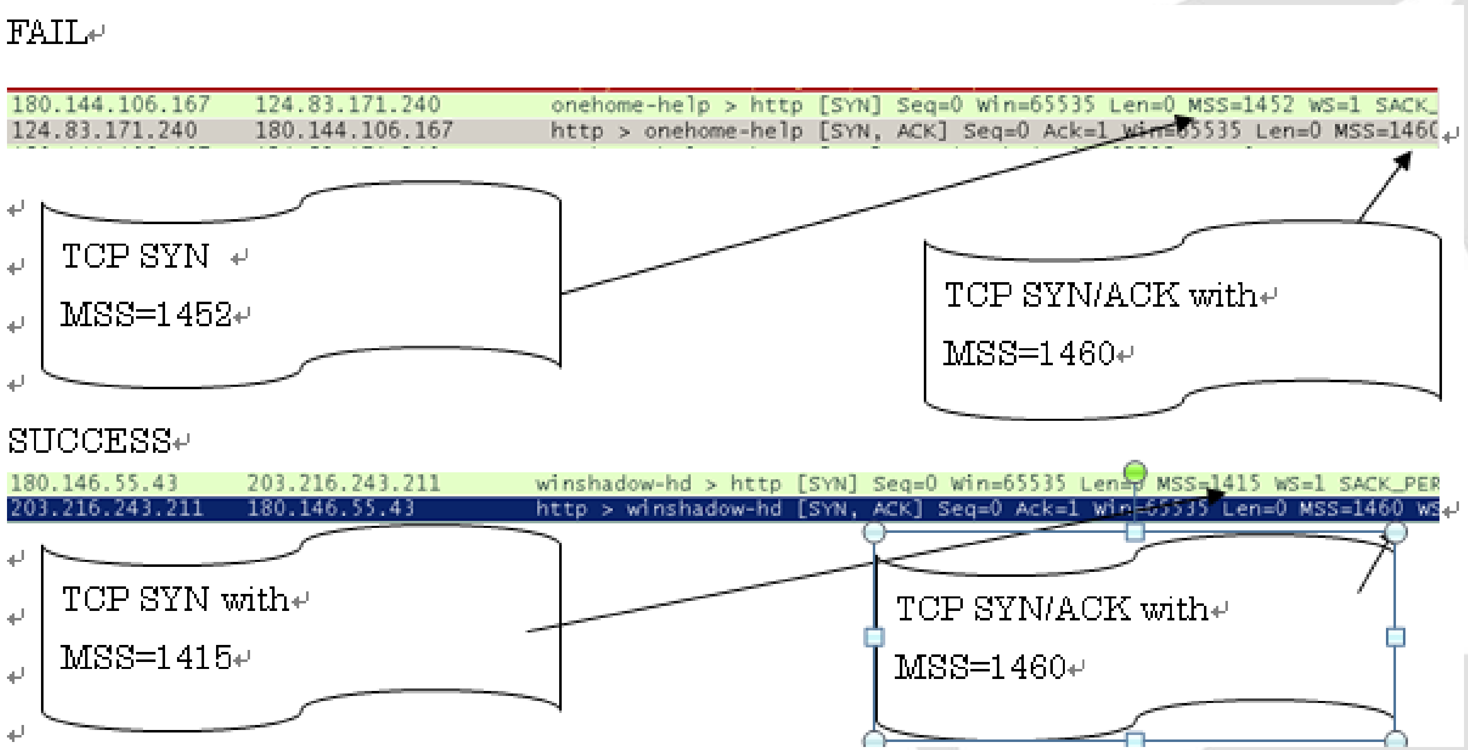
- PPPoE(FTTH) is popular in Japan.
- NTT west’s MTU is 1454 (Ethernet(1518)-EthernetHeader+FCS(14+4)-IP(20)-UDP(20)-L2TP(16)-PPPheader(2))
- NTT east optical fiber network’s MTU is 1438 （ MSS 1398）
- MSS value is determined in TCP negotiation, SYN/SYN-ACK packet in 3 way handshake



```
⊟ Transmission Control Protocol, Src Port: onehome-help (2199), Dst Port: http (80), Seq: 0, Len: 0
    Source port: onehome-help (2199)
    Destination port: http (80)
    [Stream index: 1]
    Sequence number: 0    (relative sequence number)
    Header length: 32 bytes
  ⊞ Flags: 0x002 (SYN)
    Window size value: 65535
    [Calculated window size: 65535]
  ⊞ Checksum: 0x9240 [correct]
  ⊟ Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP)
    ⊟ Maximum segment size: 1452 bytes
        Kind: MSS size (2)
        Length: 4
        MSS Value: 1452
```

#16 Wireless Router's MTU/MSS

- MSS values are not the same in the debug.



FAIL

```
180.144.106.167    124.83.171.240     onehome-help > http [SYN] Seq=0 Win=65535 Len=0 MSS=1452 WS=1 SACK_
124.83.171.240     180.144.106.167    http > onehome-help [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
```

TCP SYN

MSS=1452

TCP SYN/ACK with

MSS=1460

SUCCESS

```
180.146.55.43      203.216.243.211    winshadow-hd > http [SYN] Seq=0 Win=65535 Len=0 MSS=1415 WS=1 SACK_PER
203.216.243.211    180.146.55.43      http > winshadow-hd [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS
```

TCP SYN with

MSS=1415

TCP SYN/ACK with

MSS=1460

#17 WPS debugging

- Push button connection of WPS between wireless router and client fails in 40MHz mode, but it works in 20MHz mode.
- IEEE defines WPS but not in detail implements

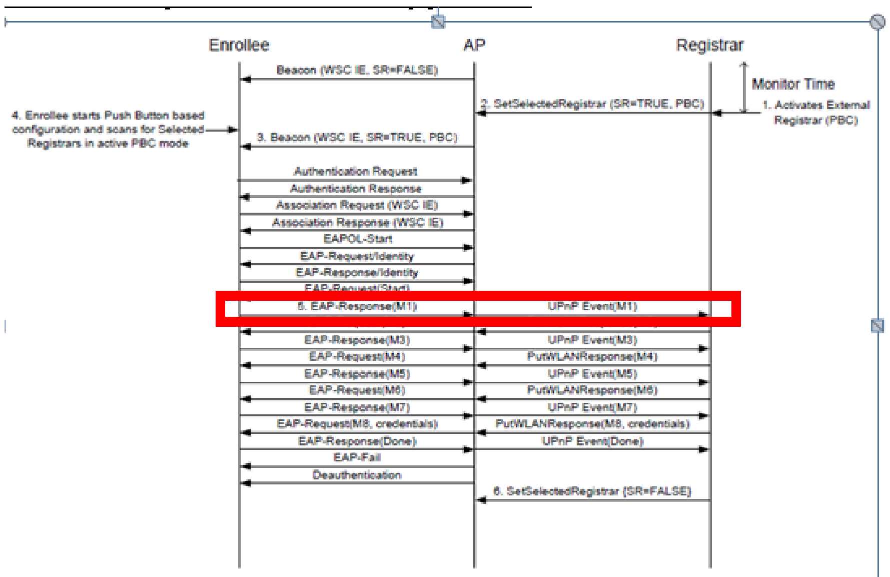
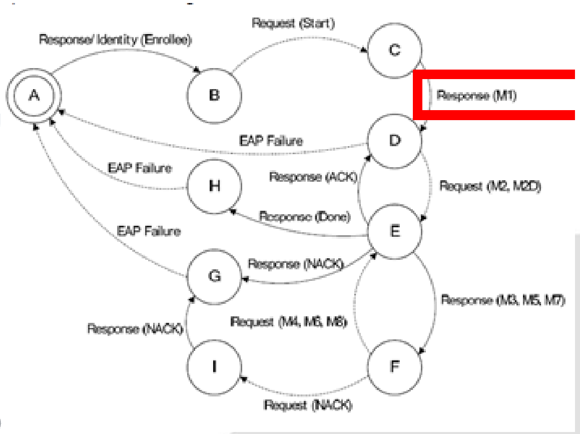
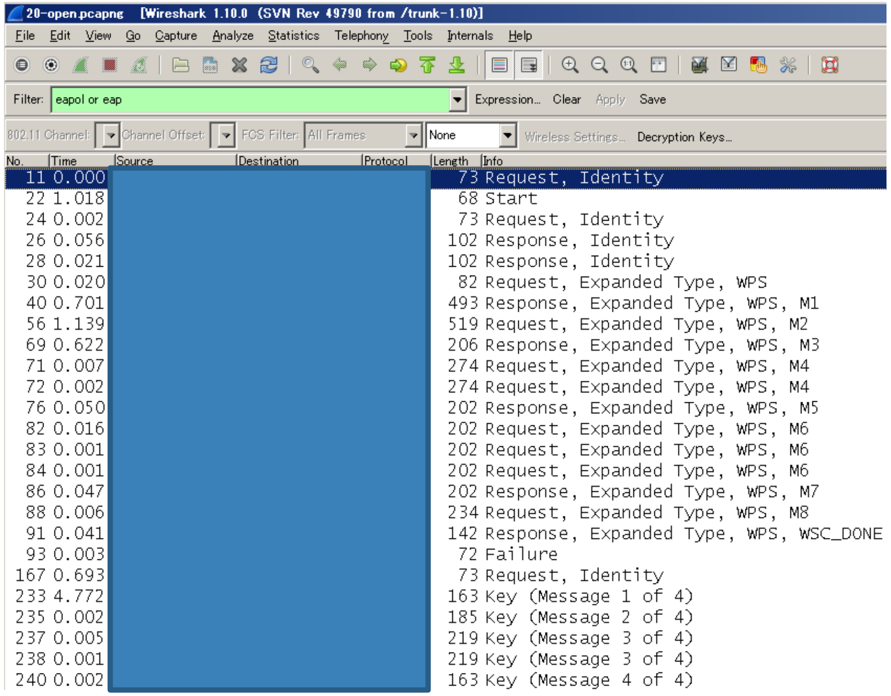


Figure 6 – PBC based setup – External Registrar trigger first



#17 WPS debugging

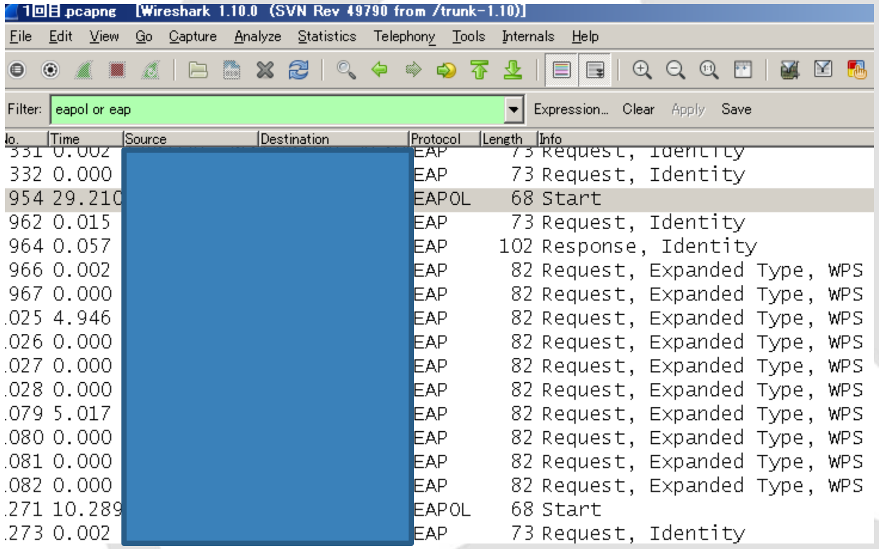
- AP sends Request Expand Type,
  but Client never response and stacked after
  ten times tries, so need to fix the one.



20-open.pcapng [Wireshark 1.10.0 (SVN Rev 49790 from /trunk-1.10)]

Filter: eapol or eap

```
No.  Time   Source  Destination  Protocol  Length  Info
[redacted]
 11 0.000    73 Request, Identity
 22 1.018    68 Start
 24 0.002    73 Request, Identity
 26 0.056   102 Response, Identity
 28 0.021   102 Response, Identity
 30 0.020    82 Request, Expanded Type, WPS
 40 0.701   493 Response, Expanded Type, WPS, M1
 56 1.139   519 Request, Expanded Type, WPS, M2
 69 0.622   206 Response, Expanded Type, WPS, M3
 71 0.007   274 Request, Expanded Type, WPS, M4
 72 0.002   274 Request, Expanded Type, WPS, M4
 76 0.050   202 Response, Expanded Type, WPS, M5
 82 0.016   202 Request, Expanded Type, WPS, M6
 83 0.001   202 Request, Expanded Type, WPS, M6
 84 0.001   202 Request, Expanded Type, WPS, M6
 86 0.047   202 Response, Expanded Type, WPS, M7
 88 0.006   234 Request, Expanded Type, WPS, M8
 91 0.041   142 Response, Expanded Type, WPS, WSC_DONE
 93 0.003    72 Failure
167 0.693    73 Request, Identity
233 4.772   163 Key (Message 1 of 4)
235 0.002   185 Key (Message 2 of 4)
237 0.005   219 Key (Message 3 of 4)
238 0.001   219 Key (Message 3 of 4)
240 0.002   163 Key (Message 4 of 4)
```



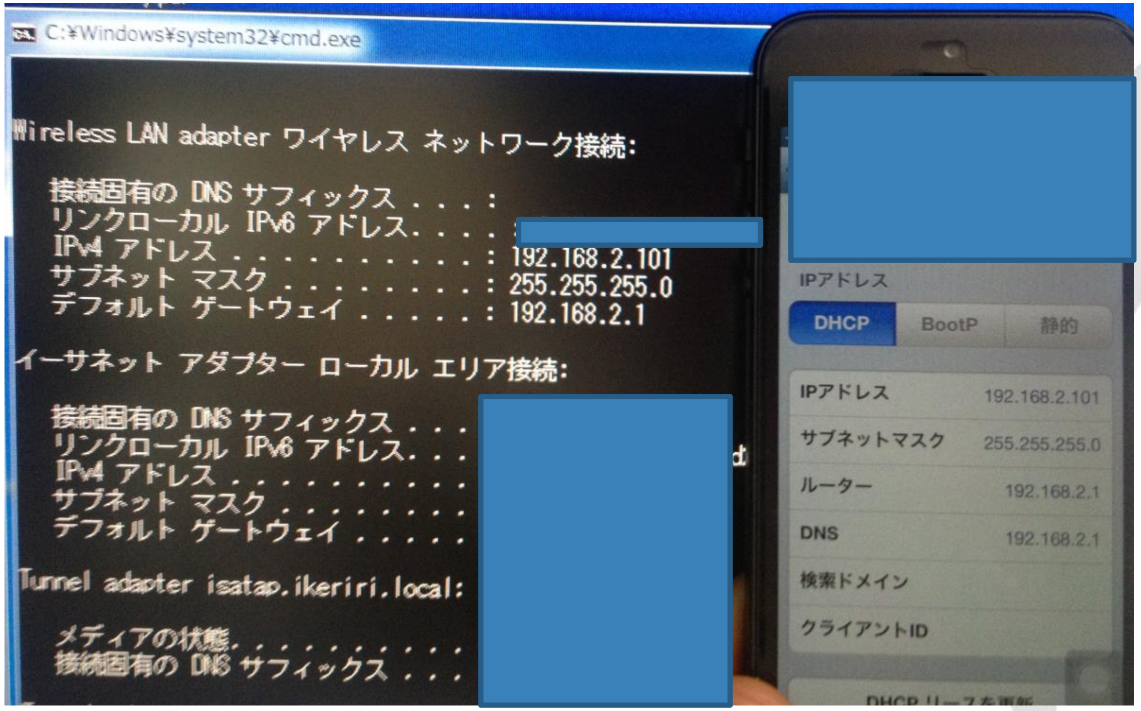
1回目.pcapng [Wireshark 1.10.0 (SVN Rev 49790 from /trunk-1.10)]

Filter: eapol or eap

```
No.  Time     Source  Destination  Protocol  Length  Info
[redacted]
 331 0.002    EAP     73 Request, Identity
 332 0.000    EAP     73 Request, Identity
 954 29.210   EAPOL   68 Start
 962 0.015    EAP     73 Request, Identity
 964 0.057    EAP    102 Response, Identity
 966 0.002    EAP     82 Request, Expanded Type, WPS
 967 0.000    EAP     82 Request, Expanded Type, WPS
.025 4.946    EAP     82 Request, Expanded Type, WPS
.026 0.000    EAP     82 Request, Expanded Type, WPS
.027 0.000    EAP     82 Request, Expanded Type, WPS
.028 0.000    EAP     82 Request, Expanded Type, WPS
.079 5.017    EAP     82 Request, Expanded Type, WPS
.080 0.000    EAP     82 Request, Expanded Type, WPS
.081 0.000    EAP     82 Request, Expanded Type, WPS
.082 0.000    EAP     82 Request, Expanded Type, WPS
.271 10.289   EAPOL   68 Start
.273 0.002    EAP     73 Request, Identity
```

#18 wireless router’s DHCP issue

- The wireless router provides same IP address to another PC and smartphone in same SSID.



```
C:¥Windows¥system32¥cmd.exe

Wireless LAN adapter ワイヤレス ネットワーク接続:

   接続固有の DNS サフィックス . . . . :
   リンクローカル IPv6 アドレス. . . . : [redacted]
   IPv4 アドレス . . . . . . . . . . . : 192.168.2.101
   サブネット マスク . . . . . . . . . : 255.255.255.0
   デフォルト ゲートウェイ . . . . . . : 192.168.2.1

イーサネット アダプター ローカル エリア接続:

   接続固有の DNS サフィックス . . . [redacted]
   リンクローカル IPv6 アドレス. . .
   IPv4 アドレス . . . . . . . . . .
   サブネット マスク . . . . . . . .
   デフォルト ゲートウェイ . . . . .

Tunnel adapter isatap.ikeriri.local:

   メディアの状態. . . . . . . . . . :
   接続固有の DNS サフィックス . . . :
```

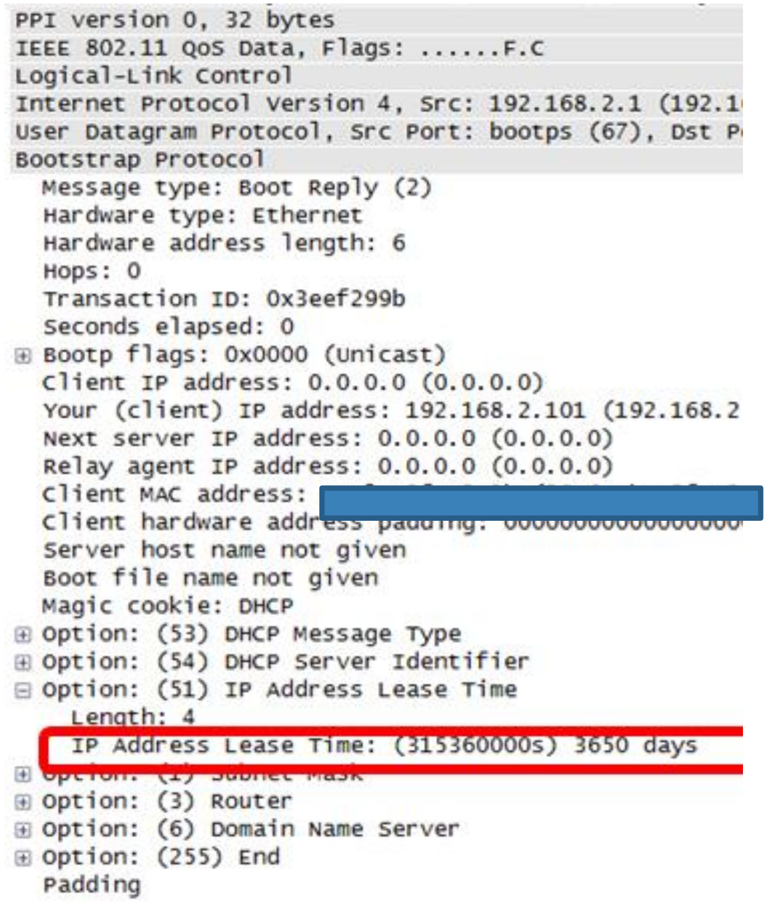
[redacted]

IPアドレス

IPアドレス	192.168.2.101
サブネットマスク	255.255.255.0
ルーター	192.168.2.1
DNS	192.168.2.1
検索ドメイン	
クライアントID	

#18 wireless router's DHCP issue

- The wireless router sends DHCP-ACK with 31536000 seconds (3650 days) of lease time

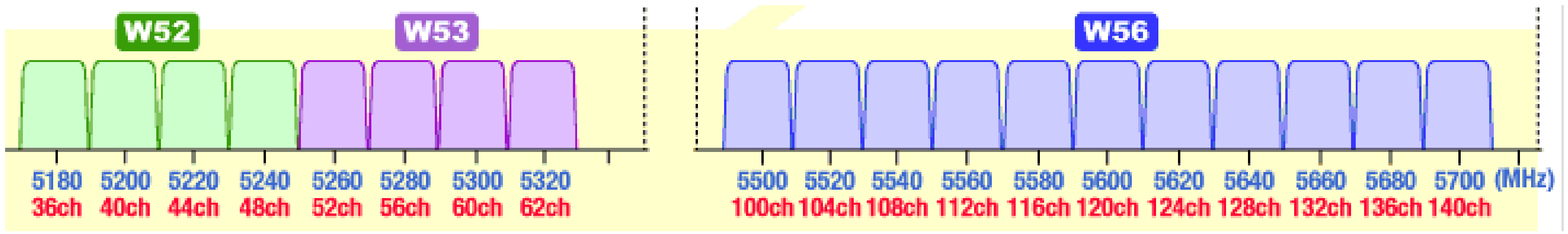


```
PPI version 0, 32 bytes
IEEE 802.11 QoS Data, Flags: ......F.C
Logical-Link Control
Internet Protocol Version 4, Src: 192.168.2.1 (192.1
User Datagram Protocol, Src Port: bootps (67), Dst P
Bootstrap Protocol
   Message type: Boot Reply (2)
   Hardware type: Ethernet
   Hardware address length: 6
   Hops: 0
   Transaction ID: 0x3eef299b
   Seconds elapsed: 0
⊞ Bootp flags: 0x0000 (Unicast)
   Client IP address: 0.0.0.0 (0.0.0.0)
   Your (client) IP address: 192.168.2.101 (192.168.2
   Next server IP address: 0.0.0.0 (0.0.0.0)
   Relay agent IP address: 0.0.0.0 (0.0.0.0)
   Client MAC address: [redacted]
   Client hardware address padding: 00000000000000000
   Server host name not given
   Boot file name not given
   Magic cookie: DHCP
⊞ Option: (53) DHCP Message Type
⊞ Option: (54) DHCP Server Identifier
⊟ Option: (51) IP Address Lease Time
     Length: 4
     IP Address Lease Time: (31536000s) 3650 days
⊞ Option: (1) Subnet Mask
⊞ Option: (3) Router
⊞ Option: (6) Domain Name Server
⊞ Option: (255) End
   Padding
```

- Both Windows and smartphone accepted, but smartphone changes lease time value into 90 days (selfishly)
- So IP duplicated.

#19 DFS debugging

- There are tons of RF signals in Tokyo central. 2.4GHz bands are worthless, so companies tends to use 5GHz (W53, W54, W56 channel)
- W58 bandwidth is prohibited in Japanese law



W52
W53
W56
5180 5200 5220 5240 5260 5280 5300 5320
36ch 40ch 44ch 48ch 52ch 56ch 60ch 62ch
5500 5520 5540 5560 5580 5600 5620 5640 5660 5680 5700 (MHz)
100ch 104ch 108ch 112ch 116ch 120ch 124ch 128ch 132ch 136ch 140ch

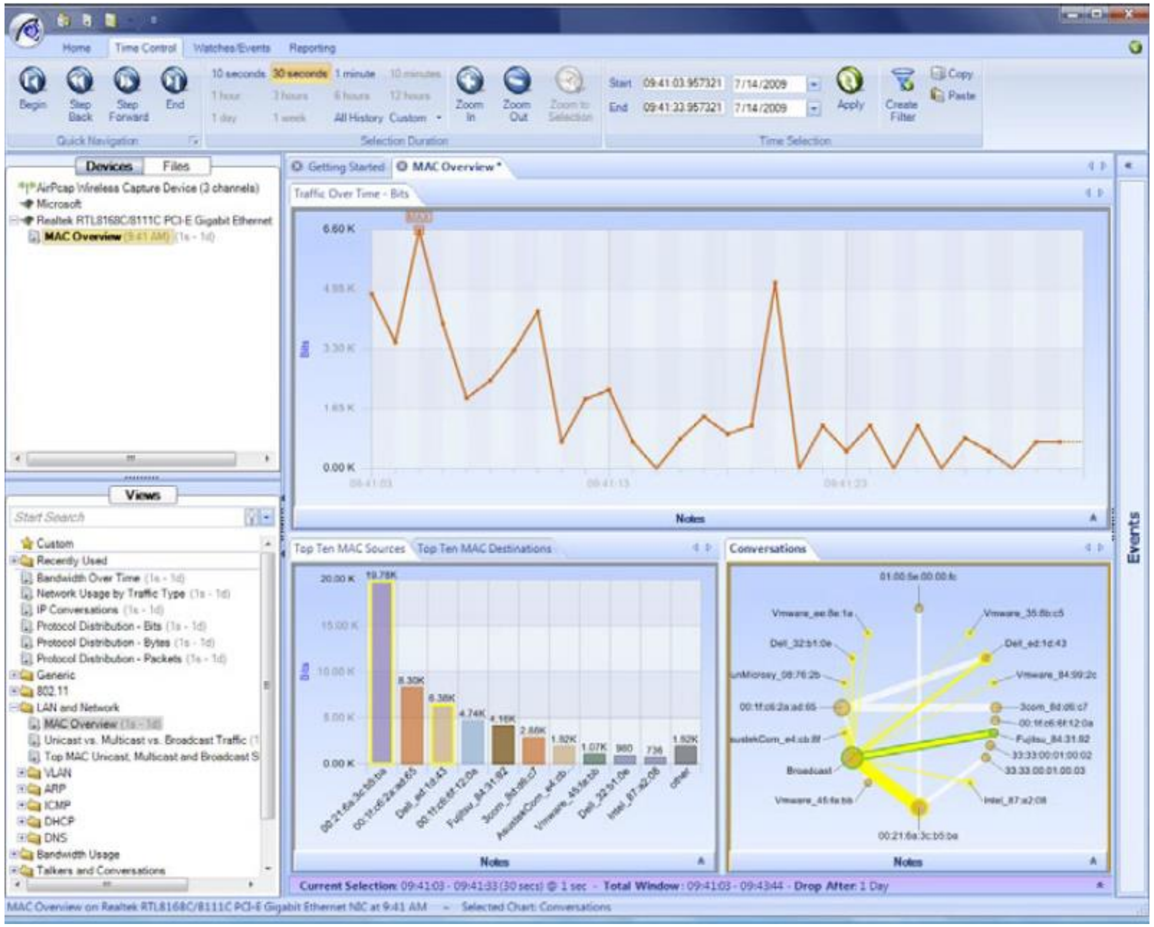
- In case of indoor office, DFS comes and stack the communication 30 minutes, no fallback.
- Failed in automatically channel changing, so the customer have to re-connect manually.

#19 DFS Debugging

- Using “tshark –i interface –b filesize:XXX –w filename.pcapng” and capture for long time.
- We uses 8 PCs with 8 AirPcapNX with 8 different CHs W53 (52 / 56 /60 / 64) and W56 (100 / 104 / 108 / 112) channel.
- Capture and wait like fishing, lurk in silence, until DFS comes (3 days …)

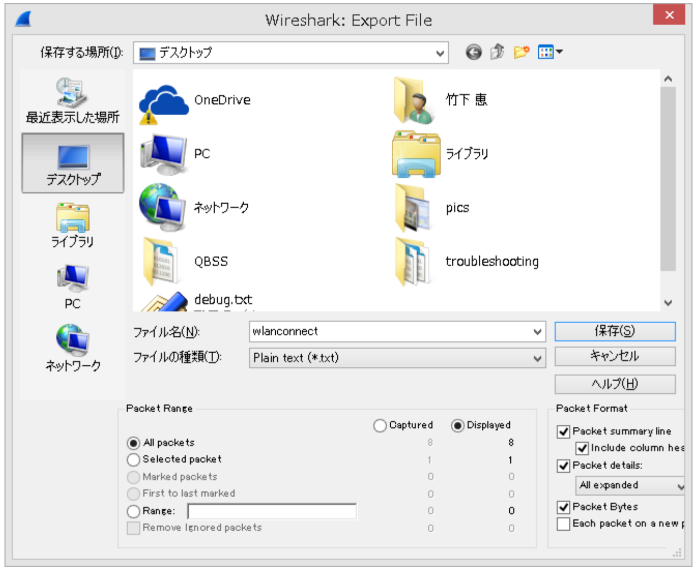
#19 DFS debugging

- If you have SteelCentral Packet Analyzer, you are lucky !
- If trace file size is 10GB, it is easy to create many graph, charts under 1 minutes


Home
Time Control
Watches/Events
Reporting
Begin
Step Back
Step Forward
End
Quick Navigation
Selection Duration
Zoom In
Zoom Out
Zoom to Selection
Start 09:41:03.957321 7/14/2009
End 09:41:33.957321 7/14/2009
Apply
Create Filter
Copy
Paste
Time Selection
Devices
Files
AirPcap Wireless Capture Device (3 channels)
Microsoft
Realtek RTL8168C/8111C PCI-E Gigabit Ethernet
MAC Overview
Views
Getting Started
MAC Overview
Traffic Over Time - Bits
Top Ten MAC Sources
Top Ten MAC Destinations
Conversations
Events
Notes
Current Selection: 09:41:03 - 09:41:33 (30 secs) @ 1 sec - Total Window: 09:41:03 - 09:43:44 - Drop After: 1 Day
MAC Overview on Realtek RTL8168C/8111C PCI-E Gigabit Ethernet NIC at 9:41 AM - Selected Chart: Conversations

#19 DFS debugging

- In deep and complex debugging, we have to collect a lot of data, and have to combine a lot of data in text.
- File>Export Packet Dissections>as “Plain Text”

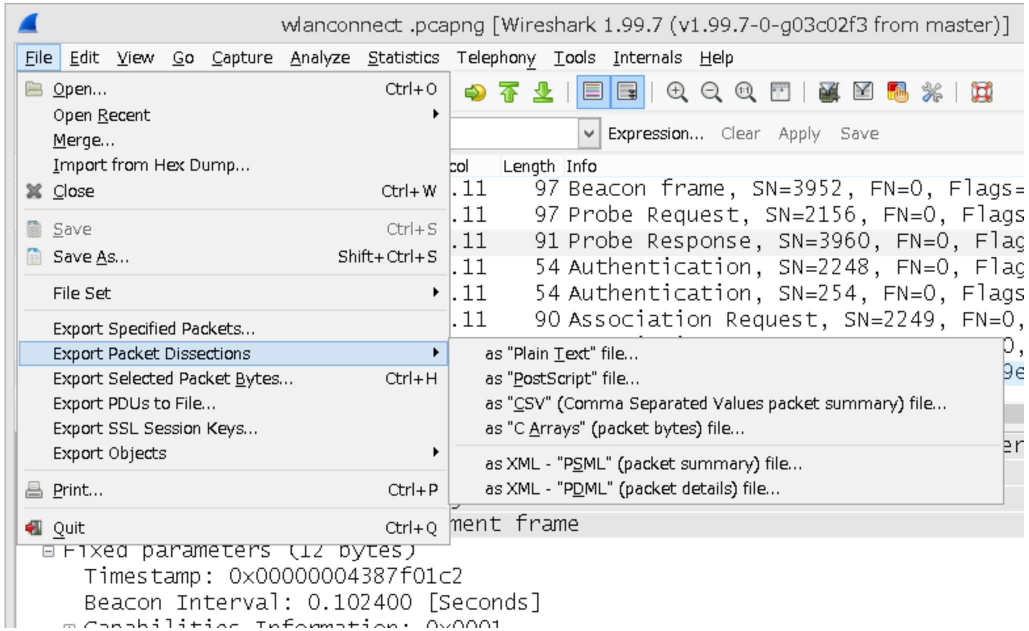


保存する場所(I): デスクトップ
ファイル名(N): wlanconnect
ファイルの種類(T): Plain text (*.txt)

Packet Range

	○ Captured	◉ Displayed
◉ All packets	8	8
○ Selected packet	1	1
○ Marked packets	0	0
○ First to last marked	0	0
○ Range:	0	0
☐ Remove Ignored packets	0	0

Packet Format
- ☑ Packet summary line
- ☑ Include column hea
- ☑ Packet details:
- All expanded
- ☑ Packet Bytes
- ☐ Each packet on a new p

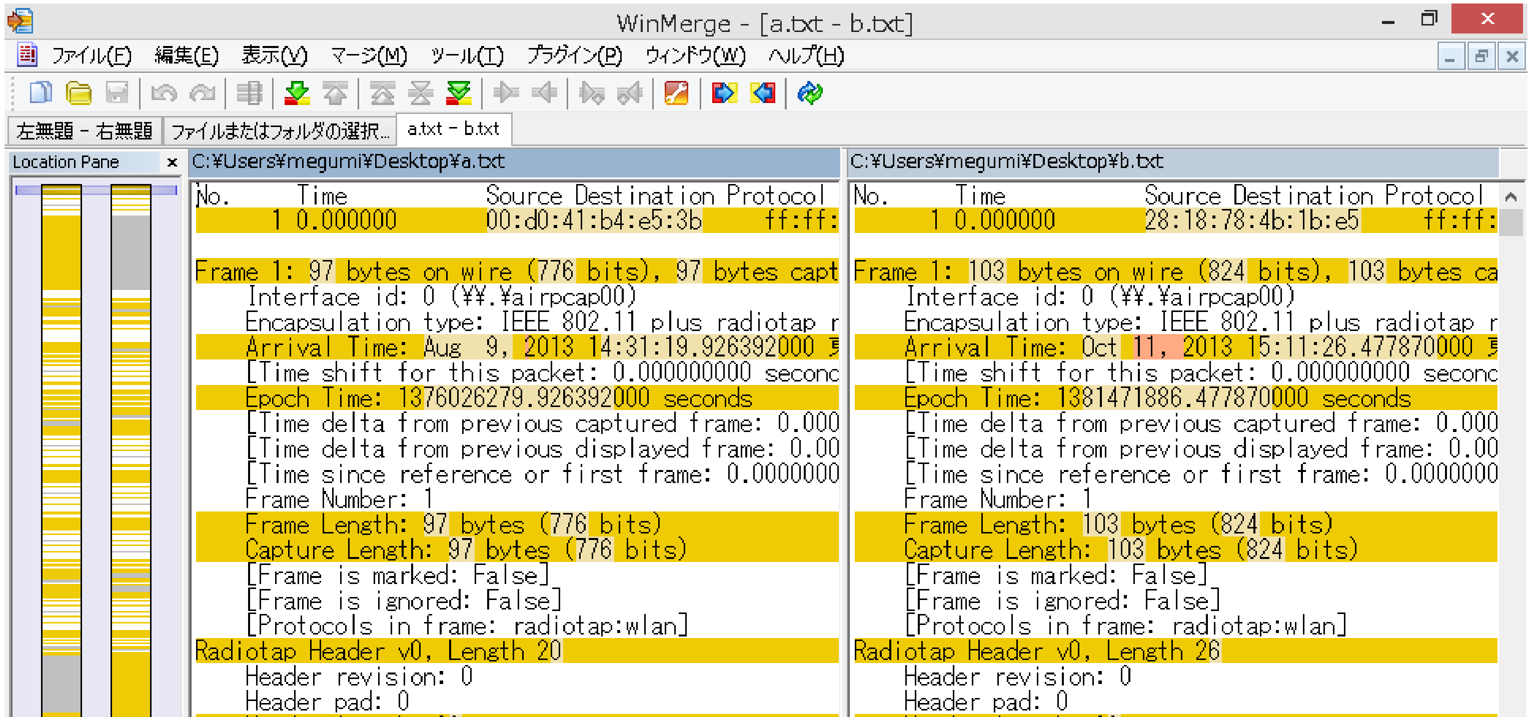

wlanconnect .pcapng [Wireshark 1.99.7 (v1.99.7-0-g03c02f3 from master)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Open... Ctrl+O
Open Recent
Merge...
Import from Hex Dump...
Close Ctrl+W
Save Ctrl+S
Save As... Shift+Ctrl+S
File Set
Export Specified Packets...
Export Packet Dissections
Export Selected Packet Bytes... Ctrl+H
Export PDUs to File...
Export SSL Session Keys...
Export Objects
Print... Ctrl+P
Quit Ctrl+Q
as "Plain Text" file...
as "PostScript" file...
as "CSV" (Comma Separated Values packet summary) file...
as "C Arrays" (packet bytes) file...
as XML - "PSML" (packet summary) file...
as XML - "PDML" (packet details) file...
Expression... Clear Apply Save

```
col   Length Info
.11      97 Beacon frame, SN=3952, FN=0, Flags=
.11      97 Probe Request, SN=2156, FN=0, Flags
.11      91 Probe Response, SN=3960, FN=0, Flag
.11      54 Authentication, SN=2248, FN=0, Flag
.11      54 Authentication, SN=254, FN=0, Flags
.11      90 Association Request, SN=2249, FN=0,
```

```
ment frame
⊟ Fixed parameters (12 bytes)
    Timestamp: 0x00000004387f01c2
    Beacon Interval: 0.102400 [Seconds]
```

#19 DFS debugging

- Text based debug is the last resort.
- check a pair of the text translated trace file. Use the WinMerge


WinMerge - [a.txt - b.txt]
ファイル(F) 編集(E) 表示(V) マージ(M) ツール(T) プラグイン(P) ウィンドウ(W) ヘルプ(H)
左無題 - 右無題
ファイルまたはフォルダの選択...
a.txt - b.txt
Location Pane
C:¥Users¥megumi¥Desktop¥a.txt
C:¥Users¥megumi¥Desktop¥b.txt

```
No.     Time               Source Destination Protocol
      1 0.000000           00:d0:41:b4:e5:3b        ff:ff:

Frame 1: 97 bytes on wire (776 bits), 97 bytes capt
    Interface id: 0 (¥¥.¥airpcap00)
    Encapsulation type: IEEE 802.11 plus radiotap r
    Arrival Time: Aug  9, 2013 14:31:19.926392000 東
    [Time shift for this packet: 0.000000000 seconc
    Epoch Time: 1376026279.926392000 seconds
    [Time delta from previous captured frame: 0.000
    [Time delta from previous displayed frame: 0.00
    [Time since reference or first frame: 0.0000000
    Frame Number: 1
    Frame Length: 97 bytes (776 bits)
    Capture Length: 97 bytes (776 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: radiotap:wlan]
Radiotap Header v0, Length 20
    Header revision: 0
    Header pad: 0
```

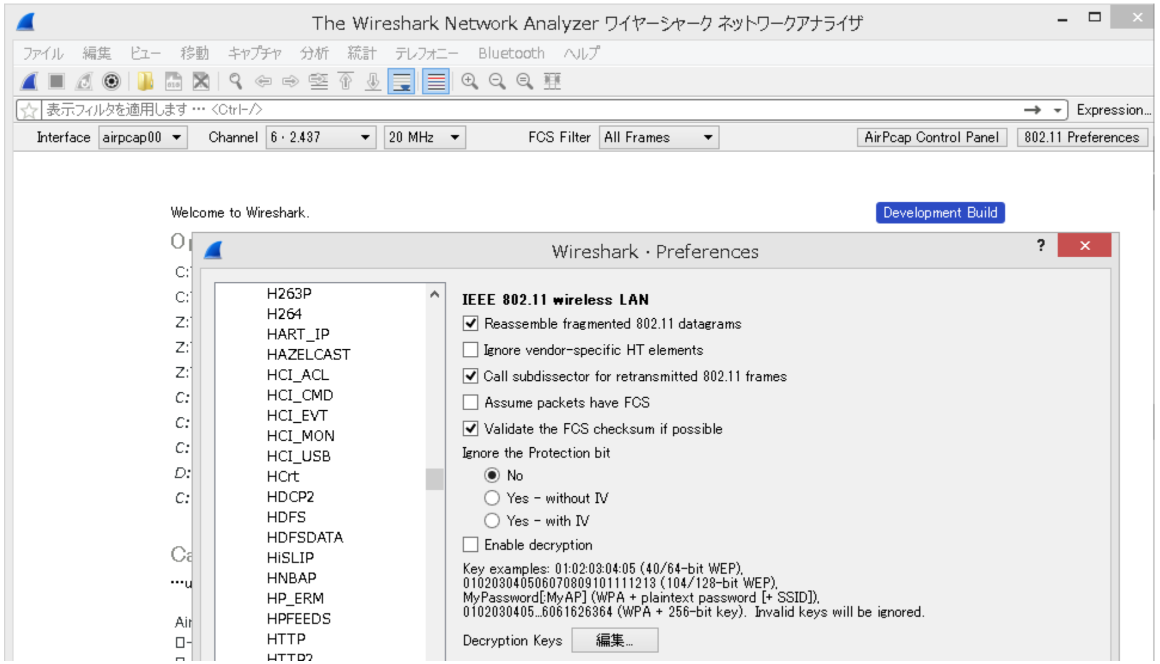
```
No.     Time               Source Destination Protocol
      1 0.000000           28:18:78:4b:1b:e5        ff:ff:

Frame 1: 103 bytes on wire (824 bits), 103 bytes ca
    Interface id: 0 (¥¥.¥airpcap00)
    Encapsulation type: IEEE 802.11 plus radiotap r
    Arrival Time: Oct 11, 2013 15:11:26.477870000 東
    [Time shift for this packet: 0.000000000 seconc
    Epoch Time: 1381471886.477870000 seconds
    [Time delta from previous captured frame: 0.000
    [Time delta from previous displayed frame: 0.00
    [Time since reference or first frame: 0.0000000
    Frame Number: 1
    Frame Length: 103 bytes (824 bits)
    Capture Length: 103 bytes (824 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: radiotap:wlan]
Radiotap Header v0, Length 26
    Header revision: 0
    Header pad: 0
```

#19 DFS debugging

- We found strange management packet at result.
- Sometimes vender may not admit,
  After many months, the fixed patch was released.
- And the wrong detection bug causes the trouble of the stack and non-recovery problem.

#20 Use Wireshark !

- Wireshark help us finding many bugs and troubles in debugging and troubleshooting
- Use Wireshark !



The Wireshark Network Analyzer ワイヤーシャーク ネットワークアナライザ

ファイル 編集 ビュー 移動 キャプチャ 分析 統計 テレフォニー Bluetooth ヘルプ
表示フィルタを適用します … <Ctrl-/>
Expression…
AirPcap Control Panel
802.11 Preferences
Development Build

Interface airpcap00 Channel 6 · 2.437 20 MHz FCS Filter All Frames

Welcome to Wireshark.

Wireshark · Preferences

H263P
H264
HART_IP
HAZELCAST
HCI_ACL
HCI_CMD
HCI_EVT
HCI_MON
HCI_USB
HCrt
HDCP2
HDFS
HDFSDATA
HiSLIP
HNBAP
HP_ERM
HPFEEDS
HTTP

IEEE 802.11 wireless LAN

- [✓] Reassemble fragmented 802.11 datagrams
- [ ] Ignore vendor-specific HT elements
- [✓] Call subdissector for retransmitted 802.11 frames
- [ ] Assume packets have FCS
- [✓] Validate the FCS checksum if possible

Ignore the Protection bit

[✓ No] Yes - without IV Yes - with IV

- [ ] Enable decryption

Key examples: 01:02:03:04:05 (40/64-bit WEP),
010203040506070809101111213 (104/128-bit WEP),
MyPassword[:MyAP] (WPA + plaintext password [+ SSID]),
0102030405…6061626364 (WPA + 256-bit key). Invalid keys will be ignored.

Decryption Keys 編集…

Thank you !
どうもありがとうございます！