


SHARKFEST 2015

WIRESHARK DEVELOPER AND USER CONFERENCE



Wireless Troubleshooting Tips using AirPcaps: DFS & Module Debugging

 COMPUTER HISTORY MUSEUM

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 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 Collect host / AP info (Windows)

- chcp 437 (English codeset)
 - “netsh wlan sh all | more “
- Driver description
Driver version (important)
INF file name
MAC address
SSID / BSSID
authentication/encryption
Channel / speed /signal
- Demonstration



The image shows two screenshots of a Windows command prompt window. The top screenshot shows the command 'netsh wlan sh all | more' being executed, resulting in a list of driver information for the 'Wi-Fi 2' interface. The bottom screenshot shows the command 'netsh wlan sh all' being executed, resulting in a detailed list of interface information for the 'Wi-Fi 2' interface.

```
C:\Users\Yreiguni>netsh wlan sh all | more
ワイヤレス システム情報の表示
(時間: 2015/08/20 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          : I-O DATA MN-AC887U 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\netrtwlana.inf
ファイル          : 2.88版

C:\Users\Yreiguni>netsh wlan sh all
ワイヤレス システム情報の表示
(時間: 2015/08/20 23:09:00 東京 (標準時))

.....
..... 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 MN-AC887U Wireless LAN Adapter
GUID              : f86f73c2-d00a-4e27-a631-0bd29c4e1d43
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:d4:0a
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
報告              : OAMP
接続モード        : プロファイル
チャンネル        : 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 Collect 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...



#1 Collect 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)

フラグメントサイズ値	2346	(256-2346)
HTサイズ値	2347	(0-2347)
ピーコン間隔	100	(20-1024 ms)
DTIMクリオド値	3	(1-10)
データレート	Auto	
ビデオレート	Auto	
チャンネル幅	<input checked="" type="radio"/> Auto 20/40 MHz <input type="radio"/> 20 MHz	
プリアンブルタイプ	<input checked="" type="radio"/> ショートプリアンブル <input type="radio"/> ロングプリアンブル	
ブロードキャストESSID	<input checked="" type="radio"/> 有効 <input type="radio"/> 無効	
CTSプロテクト	<input type="radio"/> 自動 <input type="radio"/> 常時 <input checked="" type="radio"/> なし	
送信パワー	100 %	
ターボモード	<input checked="" type="radio"/> 有効 <input type="radio"/> 無効	
WMM	<input type="radio"/> 有効 <input checked="" type="radio"/> 無効	



#2 Collect Baseline of network

- Latency and lost of Ping command
- tracer and pathping
- netstat -a | find "LISTEN"
- Iperf (throughput test)

```
コマンドプロンプト
c:\>ping www.iberiri.ne.jp

asashina.iberiri.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
base> iperf [-s|-c host] [-options]
iperf [-h|-help] [-v|-version]

Client/Server:
-t, --format [unit]      format to report: Kbits, Mbits, KBytes, MBytes
-i, --interval #        seconds between periodic bandwidth reports
-l, --len #[10M]        length of buffer to read or write (default 8 KB)
-m, --print_max          print TCP maximum segment size (MTU - TCP/IP header)
-o, --output <filename> output the report or error message to this specific file
-p, --port #            server port to listen or/connect to
-u, --udp                use UDP rather than TCP
-w, --window #[10M]     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 send extra wmsg
-M, --max #            set TCP maximum segment size (MTU - 4) 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, --reuse             reuse service in win32

Client specific:
```

- Demonstration

#3 Choose Physical header type

Type	Radiotap	PPI
Packet	<ul style="list-style-type: none">▣ Radiotap Header v0, Length 26<ul style="list-style-type: none">Header revision: 0Header pad: 0Header length: 26▣ Present flags<ul style="list-style-type: none">MAC timestamp: 297237576237288344▣ Flags: 0x00<ul style="list-style-type: none">Data Rate: 1.0 Mb/sChannel frequency: 2427 [BG 4]▣ Channel type: 802.11b (0x00a0)<ul style="list-style-type: none">SSI Signal: -41 dBmSSI Noise: -83 dBmAntenna: 0SSI Signal: 42 dB	<ul style="list-style-type: none">▣ PPI version 0, 32 bytes<ul style="list-style-type: none">Version: 0▣ Flags: 0x00<ul style="list-style-type: none">Header length: 32DLT: 105▣ 802.11-Common<ul style="list-style-type: none">Field type: 802.11-Common (2)Field length: 20TSFT: 27056577967▣ Flags: 0x0001<ul style="list-style-type: none">Rate: 1.0 MbpsChannel frequency: 2467 [BG 12]▣ Channel type: 802.11b (0x00a0)<ul style="list-style-type: none">FHSS hopset: 0x00FHSS pattern: 0x00dBm antenna signal: -61dBm antenna noise: -94

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

#3 Choose Physical header type

Type	Radiotap	PPI
GOOD	<ul style="list-style-type: none">• Easy to read, simple• Fixed format• Easy filter radiotap.dbm_antsignal	<ul style="list-style-type: none">• Extensible format future info 11ac, etc• Includes multiple antenna information
BAD	<ul style="list-style-type: none">• Cannot collect multiple antenna information	<ul style="list-style-type: none">• 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 Use 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 Use AirPcap(s)

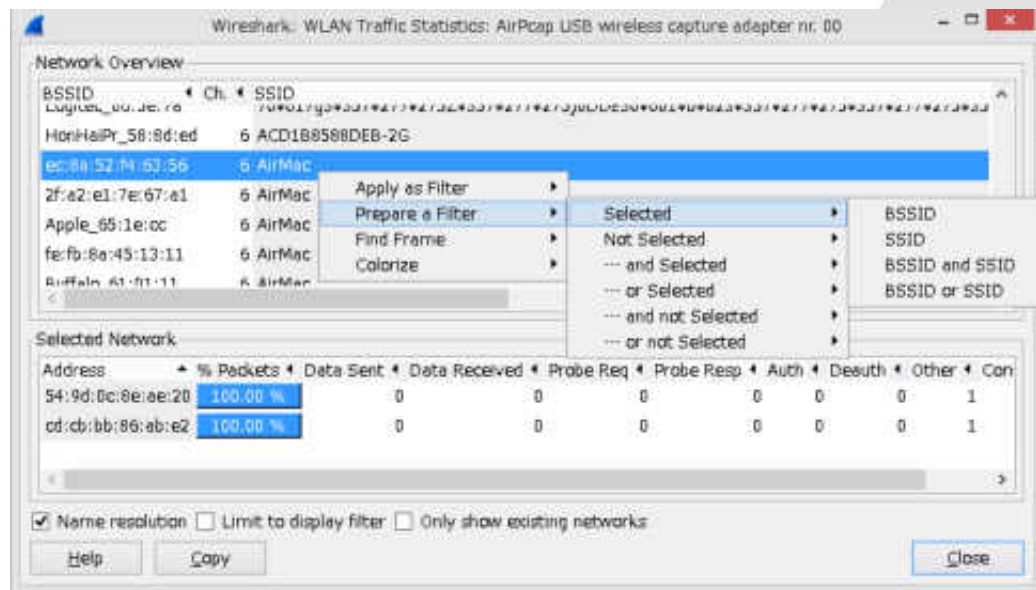
Setting	Offset -1	Offset 0	Offset +1
Channel	Main Channel 5 + Sub 1 1+5(40MHz)	Channel 5 (20MHz)	Main Channel 5 + Sub 9 5+9(40MHz)

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

- Demonstration

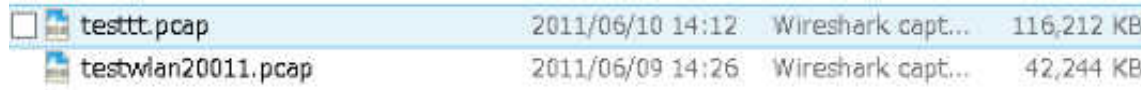
#5 Filter packet in rough



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



#5 Filter 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 Customize 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	who
Bug_31:34:f6		Matsushi_94:9f:1e	ARP	10.

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

#7 Customize coloring rules

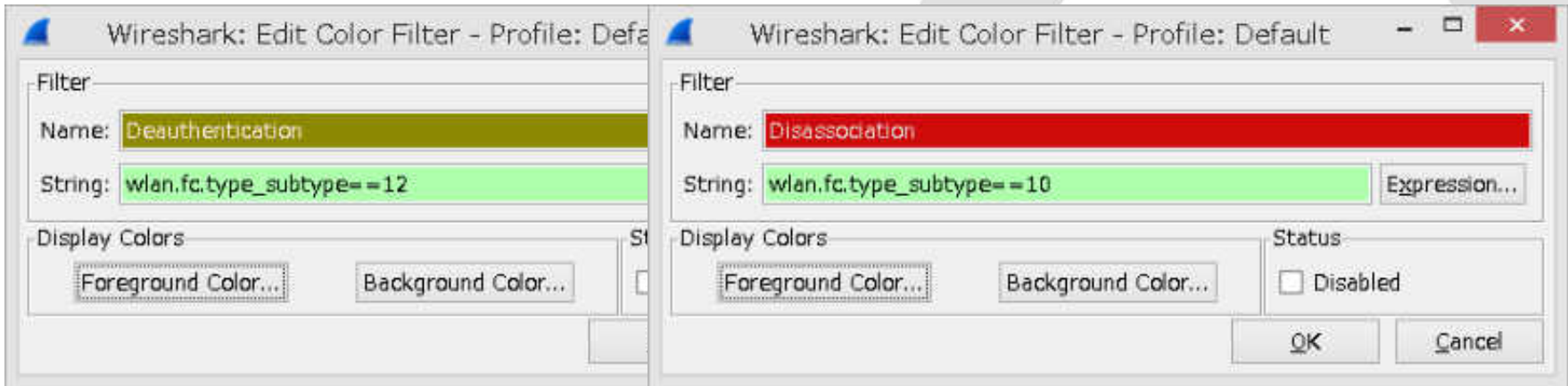
- typical troublesome packet

Deauthentication from AP or from Client

wlan.fc.type_subtype==12

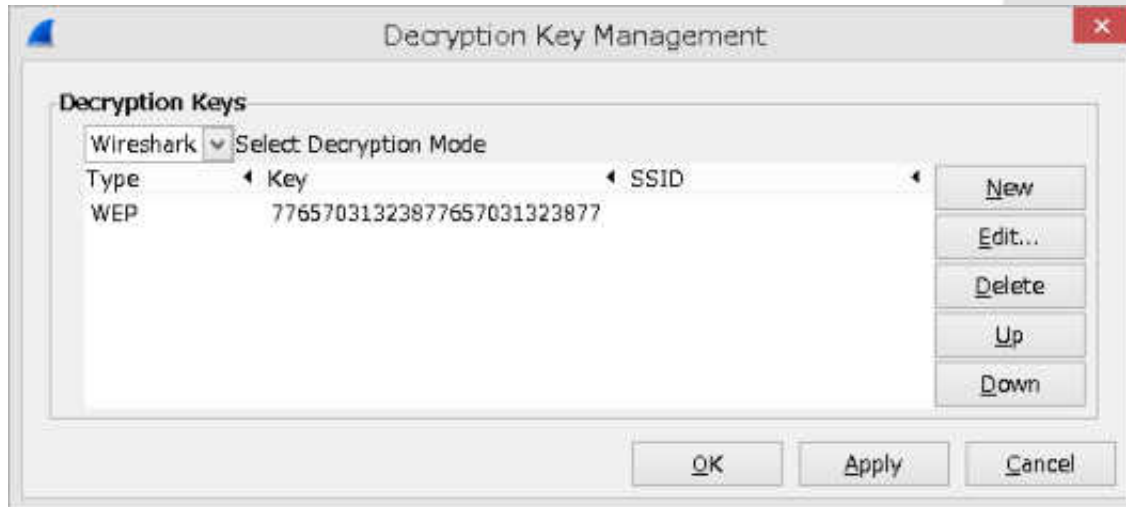
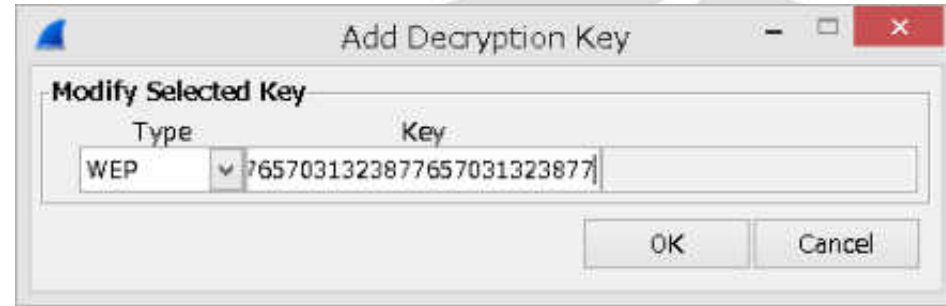
Disassociation from AP or from Client

wlan.fc.type_subtype==10



#8 Set WEP Key

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



#8 Set 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 Display Tools Internet Help
Filter:
Expression... Clear Apply Save
Packet List
No. Time Signture Length Code Type Code Info Source Destination Encrypted
1 0.000000 44 L D Probe Request 28:18:78:4b:18:e5 ff:ff:ff:ff:ff:ff 802.11
2 0.042370 -77 L D Probe Response 00:90:cc:c8:c2:79 28:18:78:4b:18:e5 802.11
3 0.062728 -44 L D Authentication 28:18:78:4b:18:e5 00:90:cc:c8:c2:79 802.11
4 0.063908 -77 L D Authentication 00:90:cc:c8:c2:79 28:18:78:4b:18:e5 802.11
5 0.065559 -45 L D Association Request 28:18:78:4b:18:e5 00:90:cc:c8:c2:79 802.11
6 0.067455 -77 L D Association Response 00:90:cc:c8:c2:79 28:18:78:4b:18:e5 802.11
7 0.070601 -44 B D Data 00:90:cc:c8:c2:79 28:18:78:4b:18:e5 802.11
8 0.080211 -78 L D Data 28:18:78:4b:18:e5 ff:ff:ff:ff:ff:ff 802.11
9 0.082110 -44 S L D Data 28:18:78:4b:18:e5 33:33:33:33:33:33 802.11

# Frame 7: 308 bytes on wire (3184 bits), 308 bytes captured (3184 bits) on Interface 0
# Ethernet II, Src: Intel(R) Wi-Fi, Dst: Intel(R) Wi-Fi
# IEEE 802.11 Data, Flags: [Data]...TC
  Type/Subtype: data (0x002d)
  # frame control field: 0x0041
    0001 0000 0010 1100 = Duration: 44 microseconds
    receiver address: ff:ff:ff:ff:ff:ff (00:00:00:00:00:00)
    transmitter address: ff:ff:ff:ff:ff:ff (ff:ff:ff:ff:ff:ff)
    source address: 28:18:78:4b:18:e5 (28:18:78:4b:18:e5)
    destination address: 00:90:cc:c8:c2:79 (00:90:cc:c8:c2:79)
    BSS ID: ff:ff:ff:ff:ff:ff (00:00:00:00:00:00)
    ... .. 0000 = fragment number: 0
    0011 0001 1000 .... = Sequence number: 790
  # frame check sequence: 0x0e520113 [correct]
    [Good: True]
    [Bad: False]
  # WEP parameters
    Initialization vector: 0x310666
    Key Index: 0
    WEP SCV: 0x04031000 (not verified)
  # Data (308 bytes)
    Data: 0090cc c8c279 281878 4b18e5 ff ff ff ff ff ff
    [Length: 308]

0000 00 00 1a 00 4f 18 00 00 00 31 32 38 77 65 70 31 32 38 77
0010 10 0c 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0020 00 03 02 79 18 18 78 4b 18 e5 ff ff ff ff ff ff ff ff ff
0030 00 11 25 0c 88 00 c8 8e 7f 1a 10 2b 47 81 03 02 10 00 00
0040 c8 28 1d 02 01 03 4f ff 33 00 01 00 00 00 00 00 00 00 00
0050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```



```
wep128connect.pcapng
File Edit View Go Capture Analyze Statistics Display Tools Internet Help
Filter:
Expression... Clear Apply Save
Packet List
No. Time Signture Length Code Type Code Info Source Destination Encrypted
1 0.000000 44 L D Probe Request 28:18:78:4b:18:e5 ff:ff:ff:ff:ff:ff 802.11
2 0.042370 -77 L D Probe Response 00:90:cc:c8:c2:79 28:18:78:4b:18:e5 802.11
3 0.062728 -44 L D Authentication 28:18:78:4b:18:e5 00:90:cc:c8:c2:79 802.11
4 0.063908 -77 L D Authentication 00:90:cc:c8:c2:79 28:18:78:4b:18:e5 802.11
5 0.065559 -45 L D Association Request 28:18:78:4b:18:e5 00:90:cc:c8:c2:79 802.11
6 0.067455 -77 L D Association Response 00:90:cc:c8:c2:79 28:18:78:4b:18:e5 802.11
7 0.070601 -44 B D Data 00:90:cc:c8:c2:79 28:18:78:4b:18:e5 802.11
8 0.080211 -78 L D Data 00:00:00 205:255:215:215 DHCP DHCP DISCOVER - Transaction ID 0a1f706886
9 0.082110 -44 S L D Data FABB19281229430FF01118 JCMPW Multiple Systems Report Message 02

# Frame 1: 103 bytes on wire (1034 bits), 103 bytes captured (1034 bits) on Interface 0
# Ethernet II, Src: Intel(R) Wi-Fi, Dst: Intel(R) Wi-Fi
# IEEE 802.11 Probe Request, Flags: [Data]...C
# IEEE 802.11 Wireless LAN Management Frame

0000 00 00 1a 00 4f 18 00 00 00 31 32 38 77 65 70 31 32 38 77
0010 10 0c 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0020 ff ff ff ff 28 18 78 4b 18 e5 ff ff ff ff ff ff ff ff ff
0030 00 11 25 0c 88 00 c8 8e 7f 1a 10 2b 47 81 03 02 10 00 00
0040 c8 28 1d 02 01 03 4f ff 33 00 01 00 00 00 00 00 00 00 00
0050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

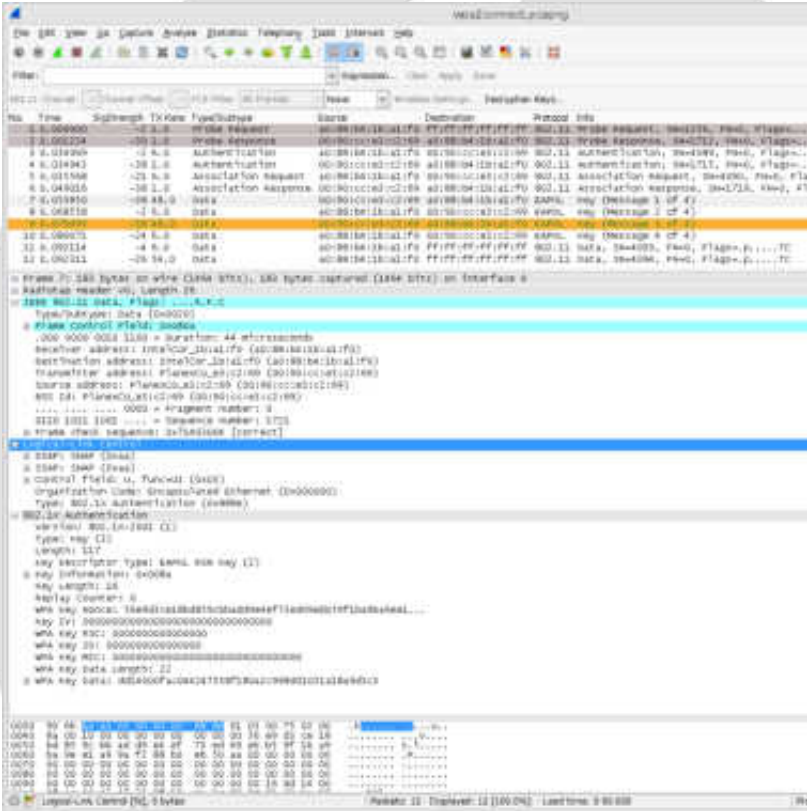
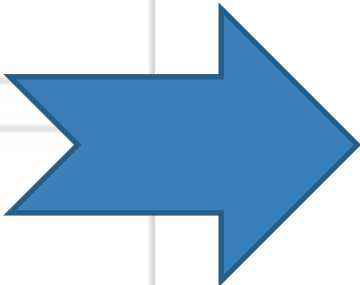
#9 Set WPA/WPA2 Key

- 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.



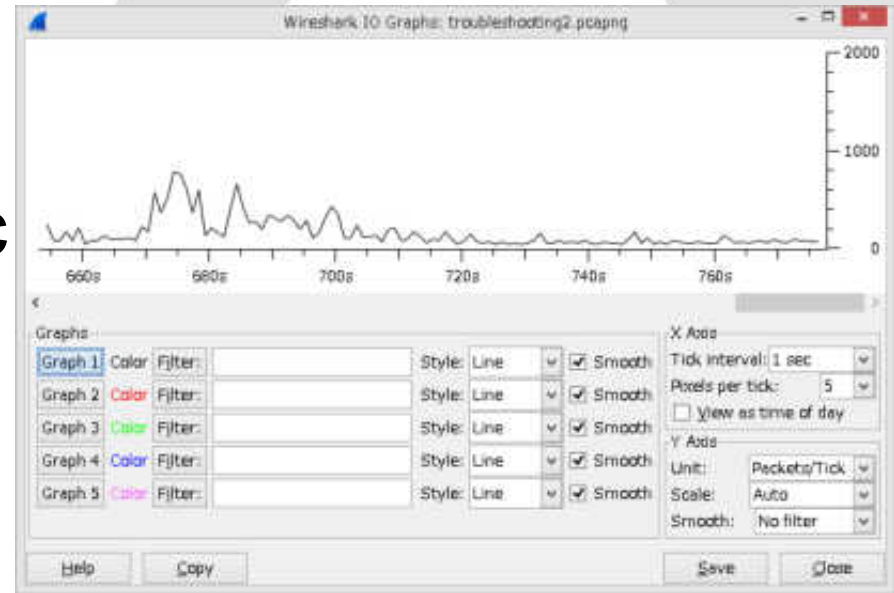
#9 Set WPA/WPA2 Key

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



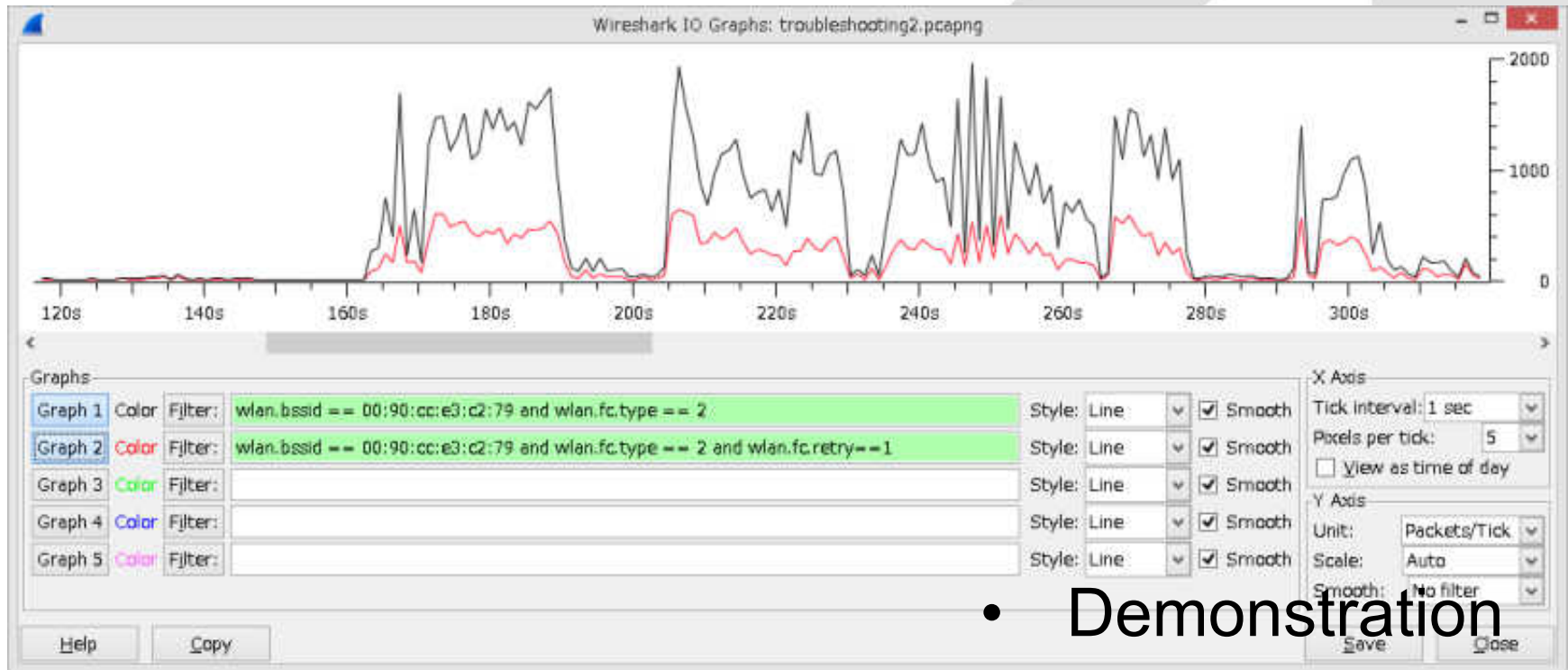
#10 Visualize (1) Retry

- 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



#10 Visualize (1) Retry

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



- Demonstration

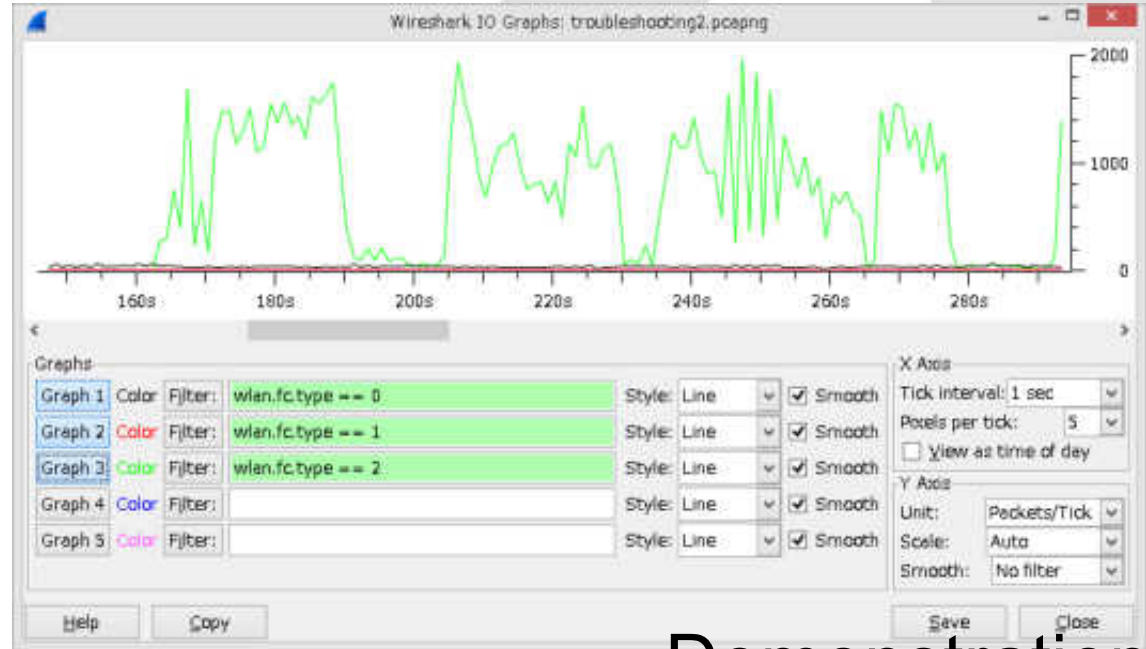
#11 Visualize (2) Frame type

- 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 Visualize (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)



• Demonstration

#12 Visualize (3) management frames

- 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 Visualize (3) management frames

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

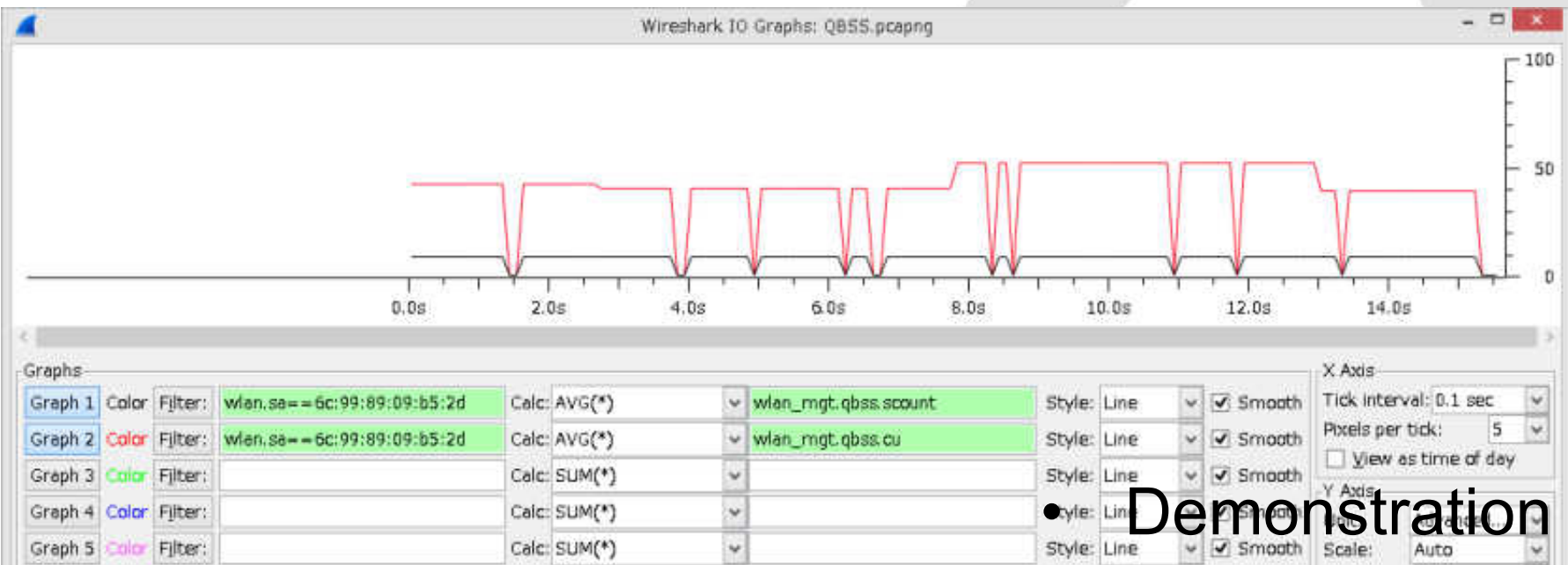
Station Count
wlan_mgt.qbss.scount
Channel Utilization
wlan_mgt.qbss.cu

```
▸ 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 Visualize (3) management frames

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)



#13 Visualize (4) signal

- Signal / Noise ratio is useful, and good ratio is 20 (signal is 10 times louder than noise)
 $20x \log 10/1 = 20\text{dB}$
- 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

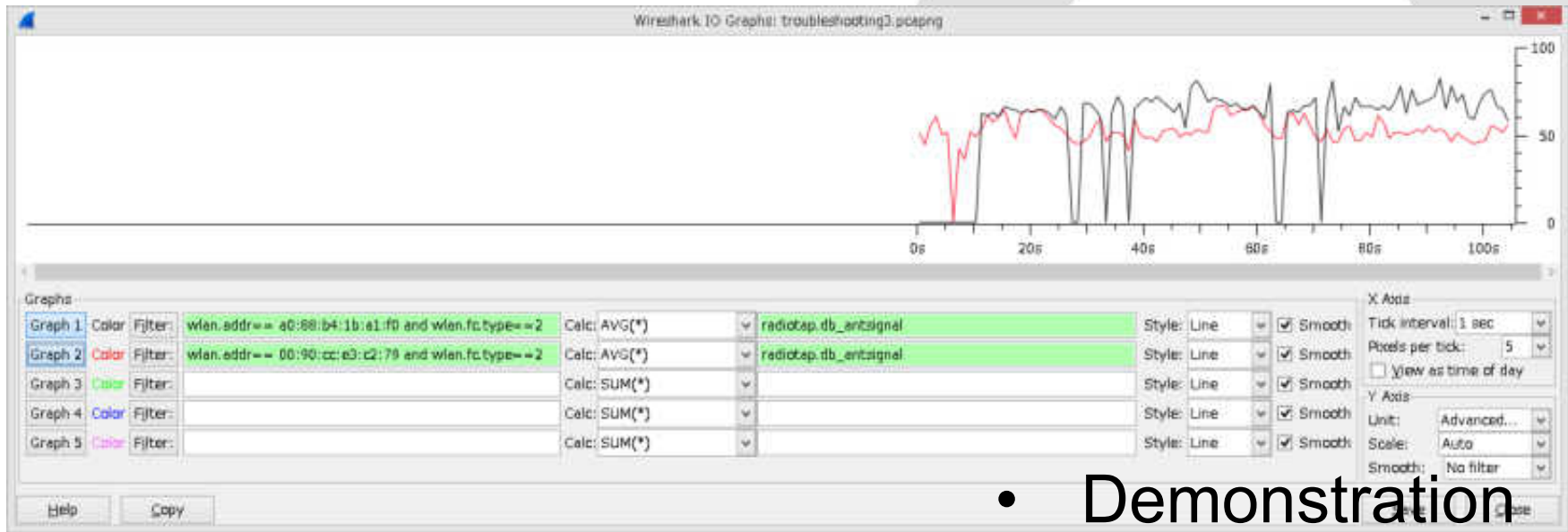
16	6.309573
17	7.079458
18	7.943282
19	8.912509
20	10
21	11.22018
22	12.58925
23	14.12538
24	15.84893
25	17.78279
26	19.95262
27	22.38721
28	25.11886
29	28.18383
30	31.62278

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 Visualize (4) signal

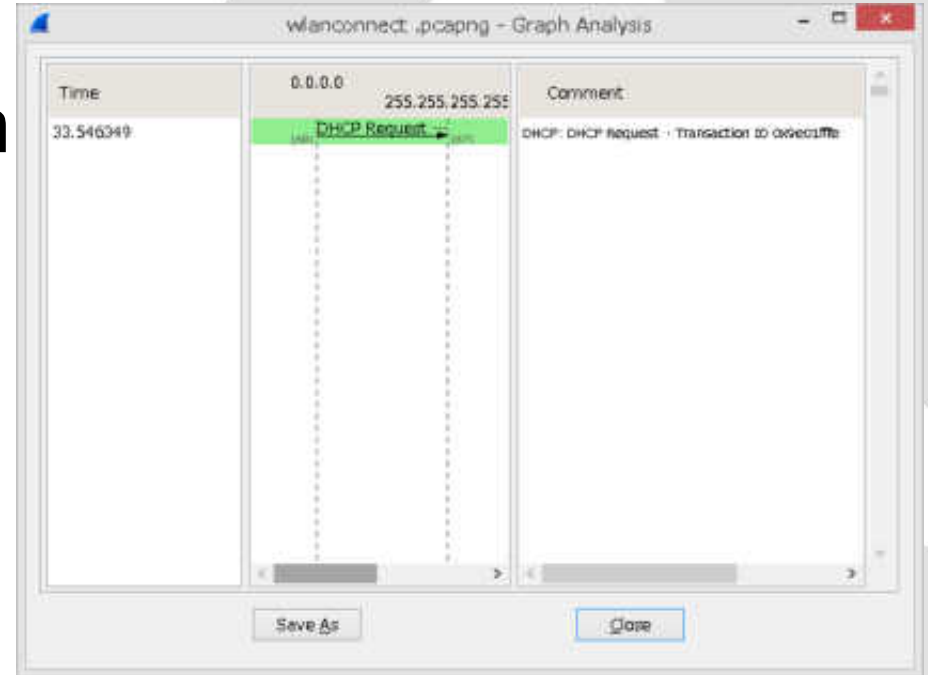
- Statistics > IO Graph and filter AP (Graph1) and filter Client (Graph2) and set Y axis to advanced, then counting AVG(*) of radiotap.db_antsignal



- Demonstration

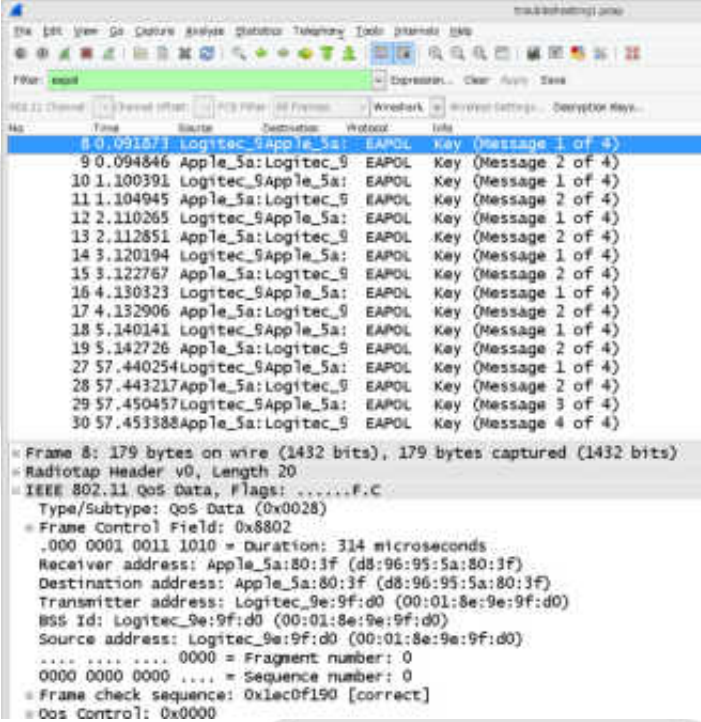
#14 Use flow graph

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



#15 Repetition of packets (iOS)

- Repetition of packets 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.



The image shows a Wireshark packet capture window. The top part displays a list of captured packets. The 8th packet is highlighted in blue. Below the list, the packet details pane shows the structure of the 8th packet, which is an IEEE 802.11 QoS Data frame containing an EAPOL Key message.

No.	Time	Source	Destination	Protocol	Info
8	0.091673	Logitec_9Apple_5a	Apple_5a:Logitec_9	EAPOL	Key (Message 1 of 4)
9	0.094846	Apple_5a:Logitec_9	Logitec_9Apple_5a	EAPOL	Key (Message 2 of 4)
10	1.100391	Logitec_9Apple_5a	Apple_5a:Logitec_9	EAPOL	Key (Message 1 of 4)
11	1.104945	Apple_5a:Logitec_9	Logitec_9Apple_5a	EAPOL	Key (Message 2 of 4)
12	2.110265	Logitec_9Apple_5a	Apple_5a:Logitec_9	EAPOL	Key (Message 1 of 4)
13	2.112851	Apple_5a:Logitec_9	Logitec_9Apple_5a	EAPOL	Key (Message 2 of 4)
14	3.120194	Logitec_9Apple_5a	Apple_5a:Logitec_9	EAPOL	Key (Message 1 of 4)
15	3.122767	Apple_5a:Logitec_9	Logitec_9Apple_5a	EAPOL	Key (Message 2 of 4)
16	4.130323	Logitec_9Apple_5a	Apple_5a:Logitec_9	EAPOL	Key (Message 1 of 4)
17	4.132906	Apple_5a:Logitec_9	Logitec_9Apple_5a	EAPOL	Key (Message 2 of 4)
18	5.140141	Logitec_9Apple_5a	Apple_5a:Logitec_9	EAPOL	Key (Message 1 of 4)
19	5.142726	Apple_5a:Logitec_9	Logitec_9Apple_5a	EAPOL	Key (Message 2 of 4)
27	57.440254	Logitec_9Apple_5a	Apple_5a:Logitec_9	EAPOL	Key (Message 1 of 4)
28	57.443217	Apple_5a:Logitec_9	Logitec_9Apple_5a	EAPOL	Key (Message 2 of 4)
29	57.450457	Logitec_9Apple_5a	Apple_5a:Logitec_9	EAPOL	Key (Message 3 of 4)
30	57.453388	Apple_5a:Logitec_9	Logitec_9Apple_5a	EAPOL	Key (Message 4 of 4)

```
Frame 8: 179 bytes on wire (1432 bits), 179 bytes captured (1432 bits) on interface 0
Ethernet II, Src: Logitec_9e:9f:d0 (00:01:8e:9e:9f:d0), Dst: Apple_5a:80:3f (d8:96:95:5a:80:3f)
IEEE 802.11 QoS Data, Flags: .....F.C
Type/Subtype: QoS Data (0x0028)
Frame Control Field: 0x8802
Duration: 314 microseconds
Receiver address: Apple_5a:80:3f (d8:96:95:5a:80:3f)
Destination address: Apple_5a:80:3f (d8:96:95:5a:80:3f)
Transmitter address: Logitec_9e:9f:d0 (00:01:8e:9e:9f:d0)
BSS Id: Logitec_9e:9f:d0 (00:01:8e:9e:9f:d0)
Source address: Logitec_9e:9f:d0 (00:01:8e:9e:9f:d0)
.....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	Logitec_9	Apple_5a:	EAPOL	Key (Message 1 of 4)
9	0.094846	Apple_5a:	Logitec_9	EAPOL	Key (Message 2 of 4)
10	1.100391	Logitec_9	Apple_5a:	EAPOL	Key (Message 1 of 4)
11	1.104945	Apple_5a:	Logitec_9	EAPOL	Key (Message 2 of 4)
12	2.110265	Logitec_9	Apple_5a:	EAPOL	Key (Message 1 of 4)
13	2.112851	Apple_5a:	Logitec_9	EAPOL	Key (Message 2 of 4)
14	3.120194	Logitec_9	Apple_5a:	EAPOL	Key (Message 1 of 4)
15	3.122767	Apple_5a:	Logitec_9	EAPOL	Key (Message 2 of 4)
16	4.130323	Logitec_9	Apple_5a:	EAPOL	Key (Message 1 of 4)
17	4.132906	Apple_5a:	Logitec_9	EAPOL	Key (Message 2 of 4)
18	5.140141	Logitec_9	Apple_5a:	EAPOL	Key (Message 1 of 4)
19	5.142726	Apple_5a:	Logitec_9	EAPOL	Key (Message 2 of 4)

• 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

EthernetII (14)	IP(20) DF=1 MF=0 Offset=	TCP (20)	MSS 1460
--------------------	--------------------------------	-------------	-------------

#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

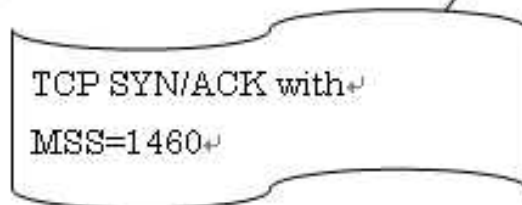
```
Transmission Control Protocol, Src Port: onehome-help (2199), Dst Port: http (80), Seq: 0, Len: 4
Source port: onehome-help (2199)
Destination port: http (80)
[stream index: 1]
sequence number: 0 (relative sequence number)
header length: 32 bytes
Flags: 0x0002 (SYN)
window size value: 65535
[calculated 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.

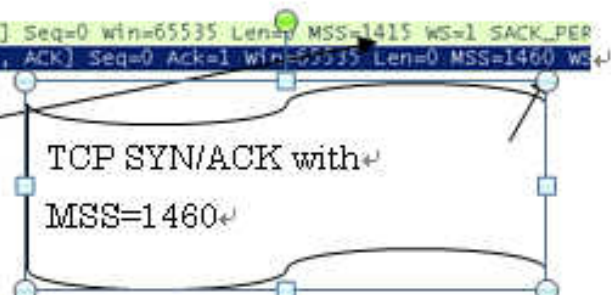
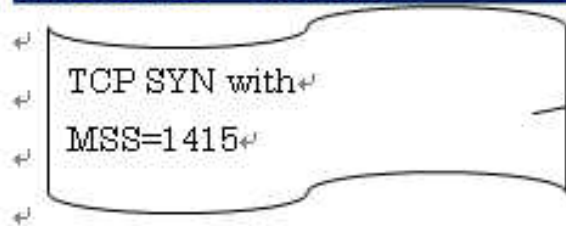
FAIL ↵

```
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 ↵
```



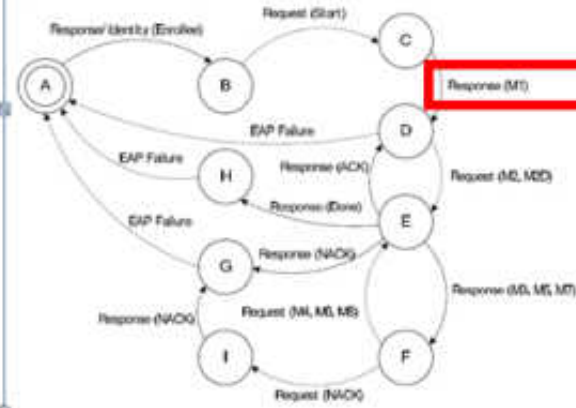
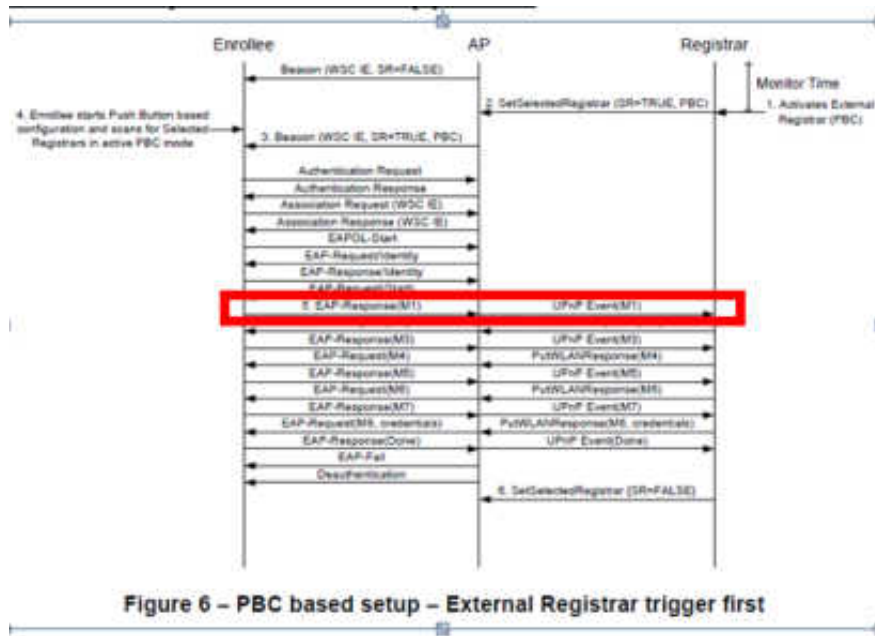
SUCCESS ↵

```
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 ↵
```



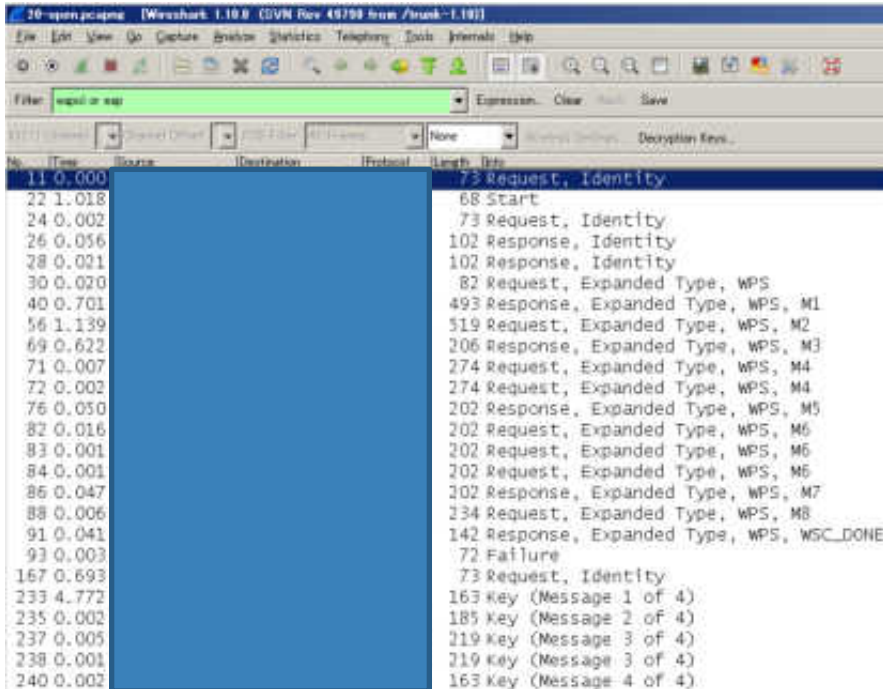
#17 WPS debugging

- 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

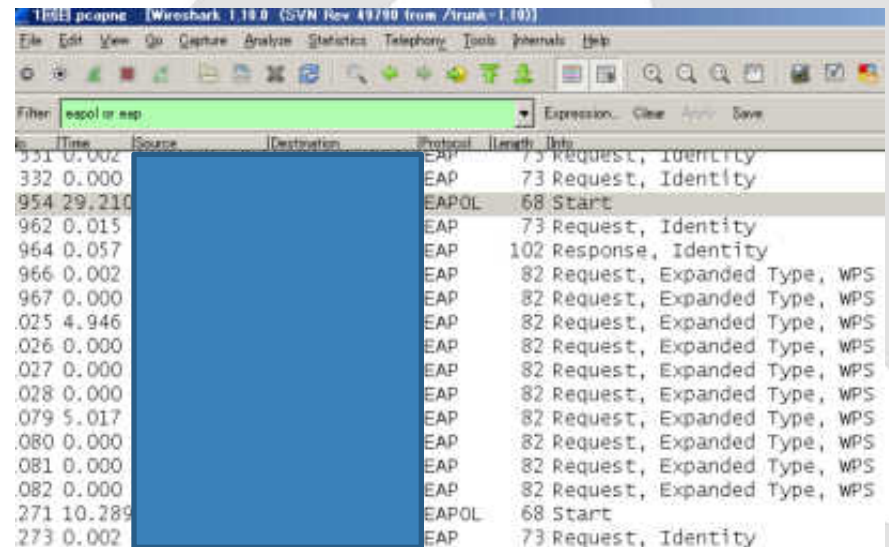


#17 WPS debugging

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



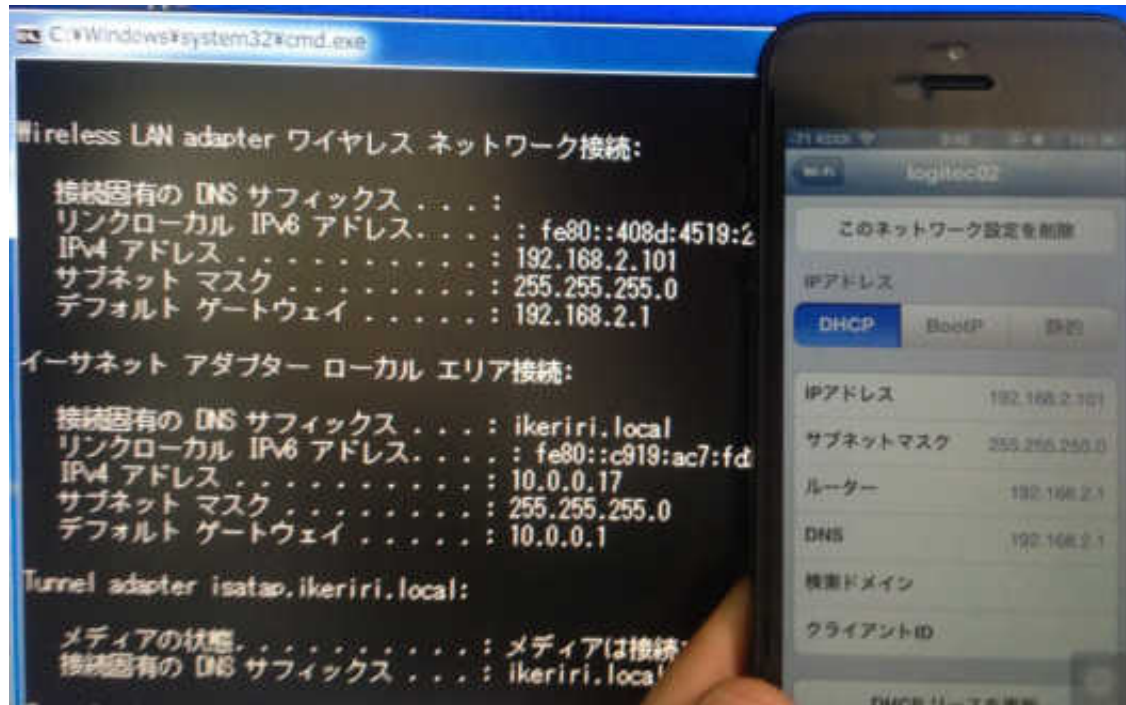
No.	Time	Source	Destination	Protocol	Length	Info
11	0.000			73		Request, Identity
22	1.018			68		Start
24	0.002			73		Request, Identity
26	0.056			102		Response, Identity
28	0.021			102		Response, Identity
30	0.020			82		Request, Expanded Type, WPS
40	0.701			493		Response, Expanded Type, WPS, M1
56	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			302		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)



No.	Time	Source	Destination	Protocol	Length	Info
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.



#18 wireless router's DHCP issue

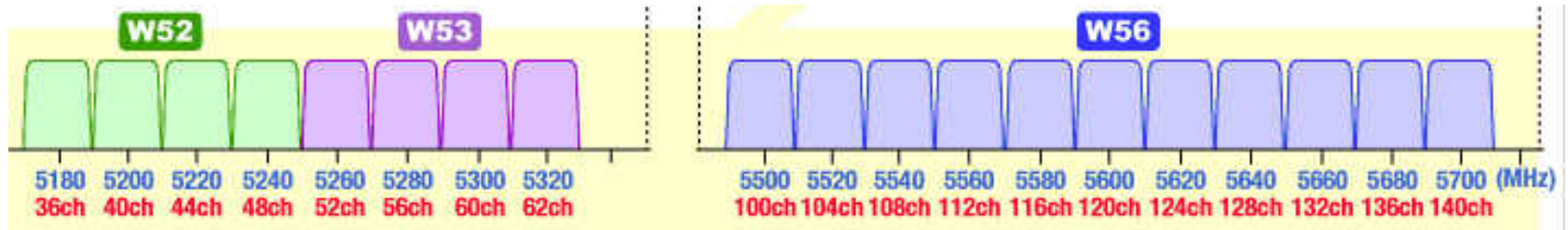
- The wireless router sends DHCP-ACK with 31536000 seconds (3650 days) of lease time

```
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 (192.168.2.1)
User Datagram Protocol, Src Port: bootps (67), Dst Port: bootpc (68)
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 (192.168.2.101)
  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: 0000000000000000
  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: (315360000s) 3650 days
  Option: (2) 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



- 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.

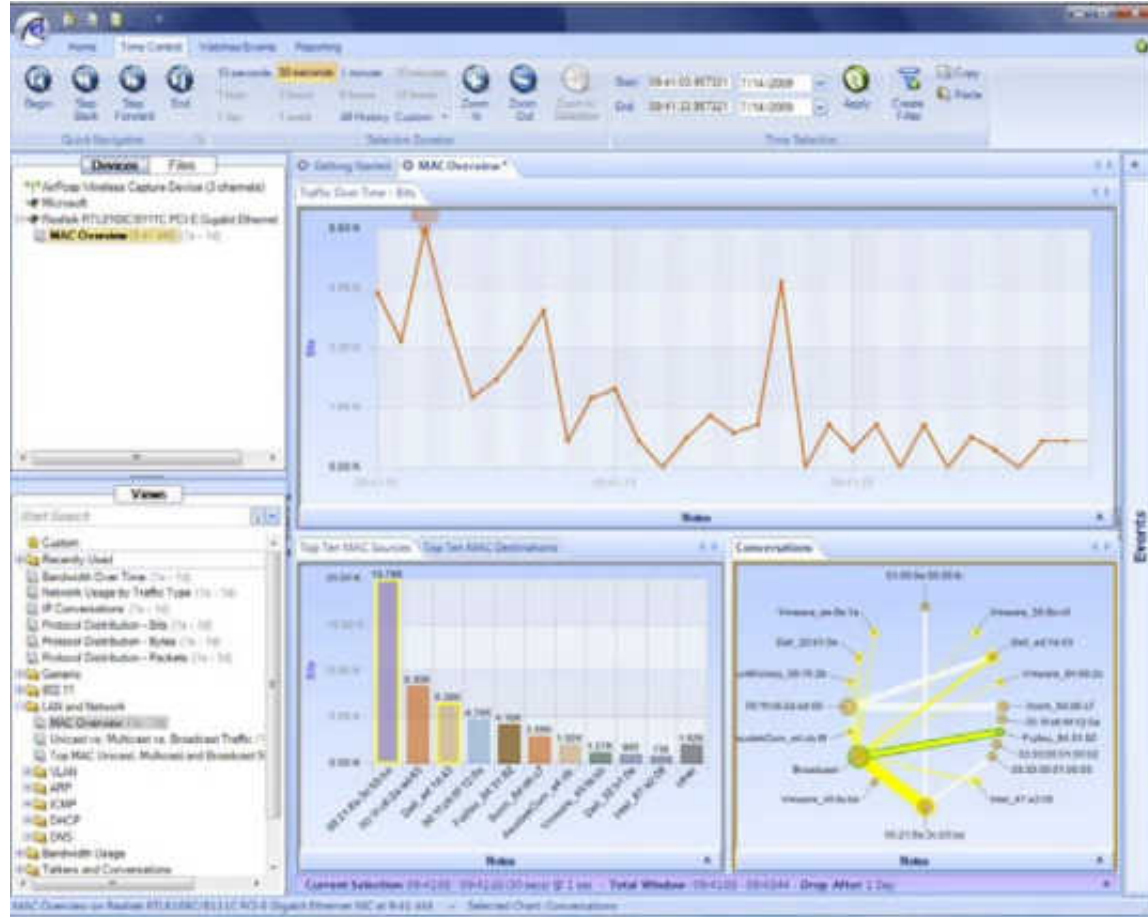
#19 DFS Debugging

- 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 ...)



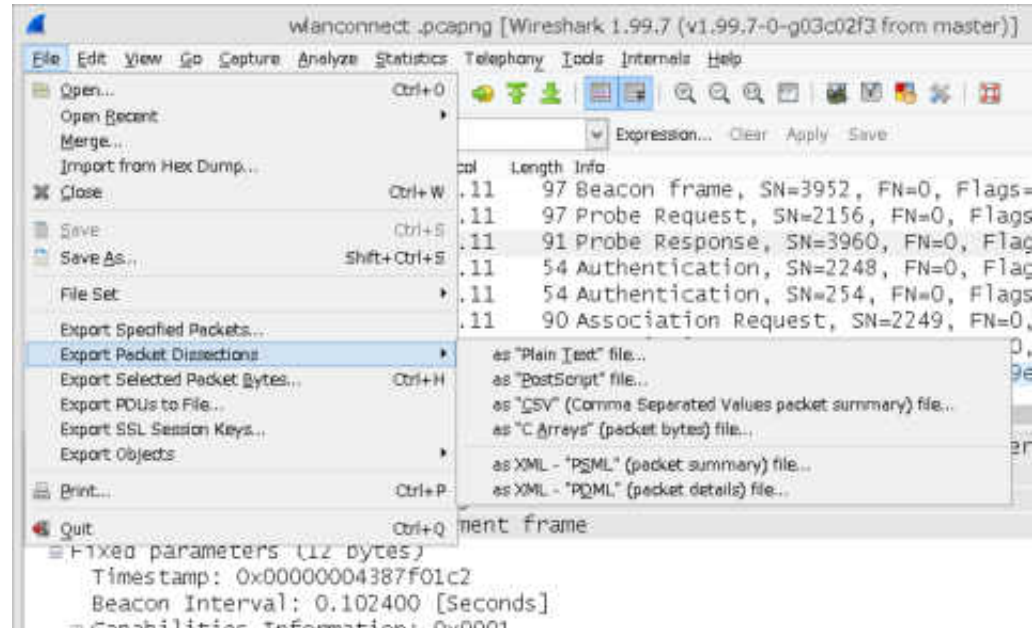
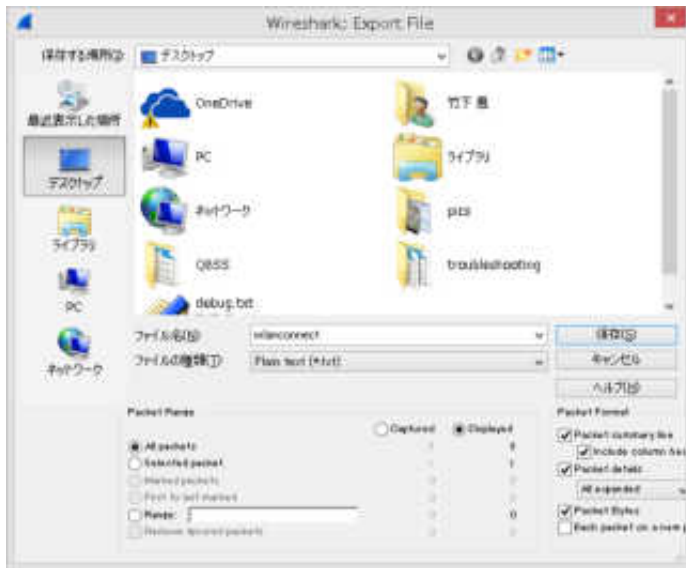
#19 DFS debugging

- 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



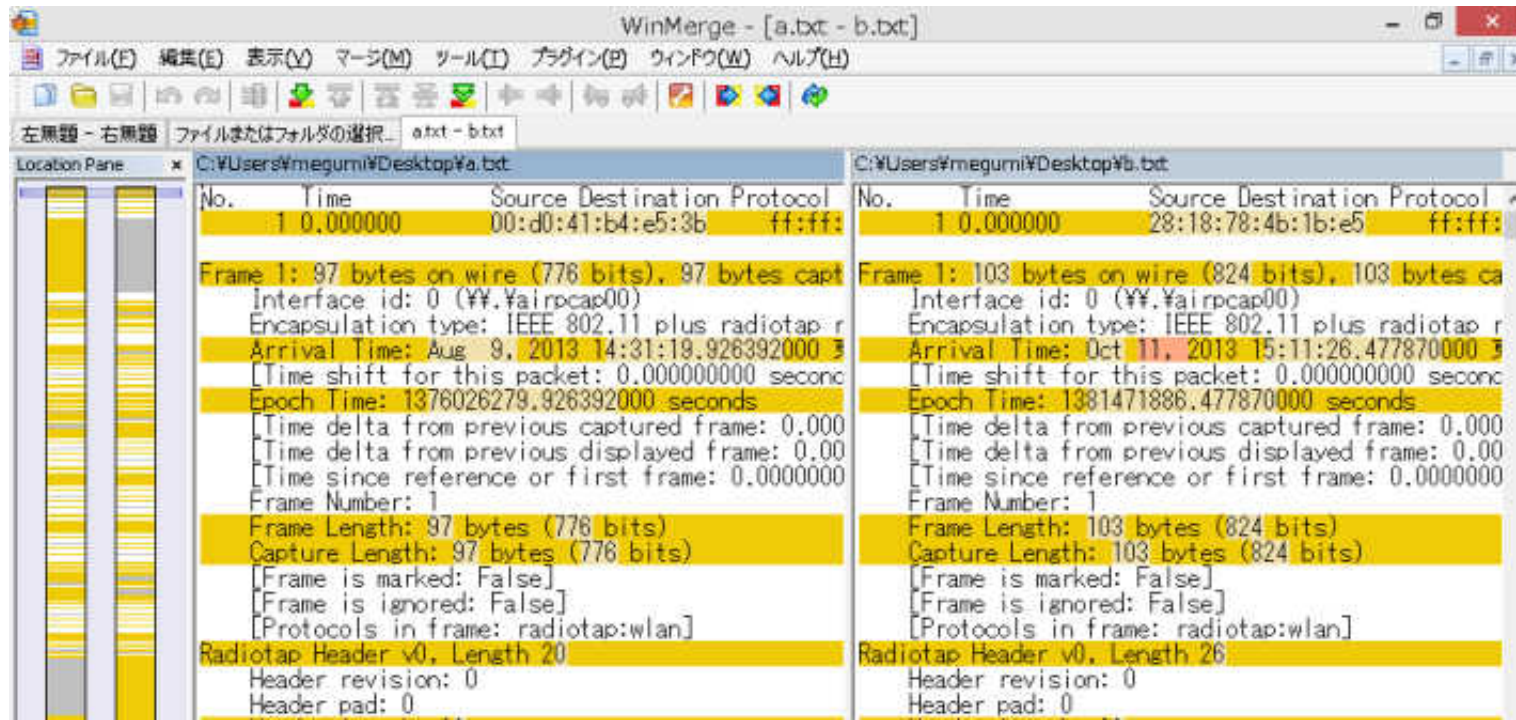
#19 DFS debugging

- 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”



#19 DFS debugging

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

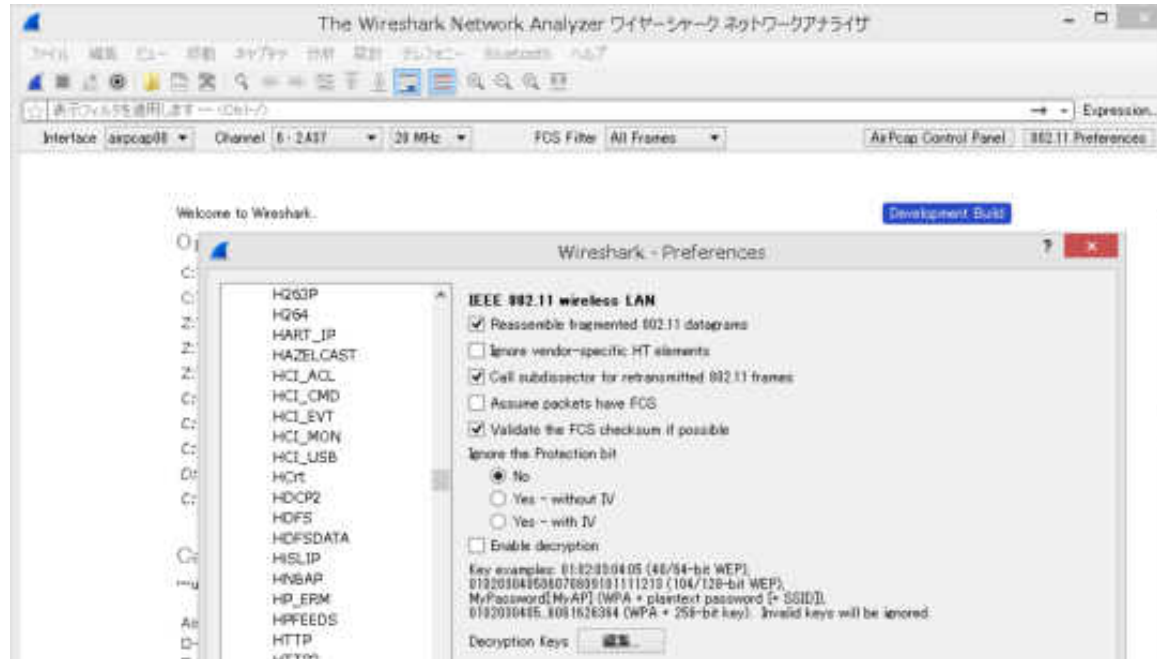


#19 DFS debugging

- 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 !



Thank you !
どうもありがとうございます !

