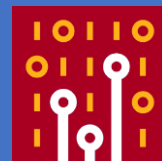




SharkFest '18 US



Point and shoot packet

Shooting point of field in packet analysis

Supplemental trace files are
<http://www.ikeriri.ne.jp/sharkfest/>

Megumi Takeshita

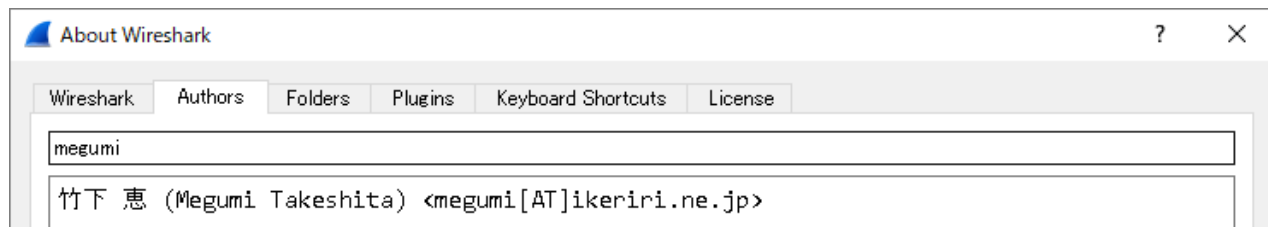
Packet Otaku
ikeriri network service



Megumi Takeshita, ikeriri network service

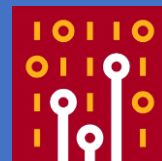


- Founder, ikeriri network service co.,ltd
- Wrote 10+ books about Wireshark
- Reseller of Riverbed Technology (former CACE technologies) in Japan
- Attending all Sharkfest
- Translator of QT Wireshark into Japanese





Point and shoot packet



When you debug, troubleshoot, and inspect security issues using Wireshark, you may just look a glance of trace file, and watch each packet sequentially in detail.

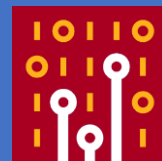
In this session Megumi show you good ways to point and shoot packet, using display filters, graphs and tables. Each layer's header has important fields to analyze trace file. So you know shooting point of field in trace file.

This session show you alternative focus points of packet for your debugging, troubleshooting, and inspection.

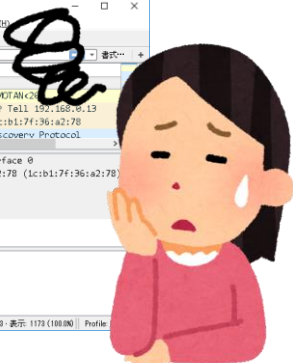
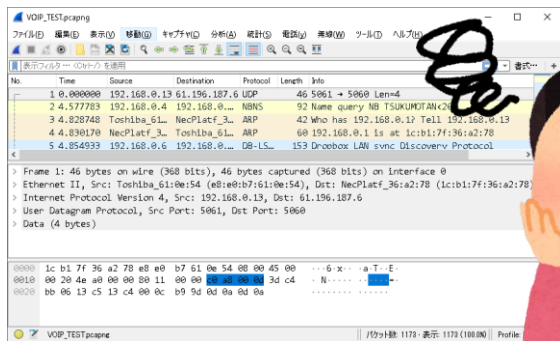




Point and Shoot packet



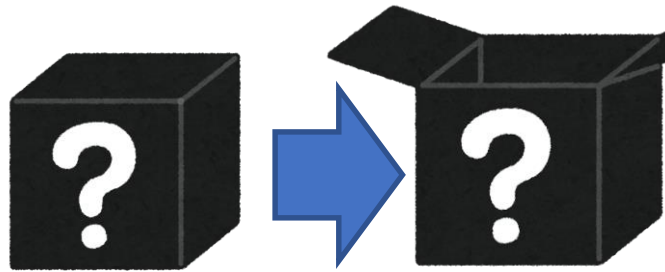
- There are trace files for debugging and troubleshooting, but what, how do we resolve ?
- Now show you the way to point the key in trace files, then shoot the trouble using Wireshark



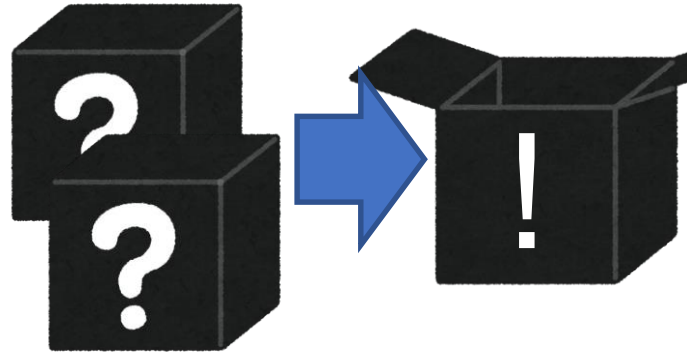
- Note all trace files are modified and anonymized



Collect 2 trace files at least



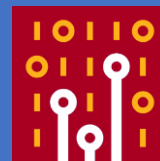
- If you inspect a trace file so deeply, you may not find the key.
- Debugging and troubleshooting are kinds of black box test, so we should point from outside of the system using trace files.



- At least 2 trace files on different conditions are needed for starting



CASE1 CATV Box



- Customer's CATV boxes in the hotel
- Some boxes fails to start up
- Capture 2 trace files
1_DHCP_SUCCESS.pcap
1_DHCP_FAIL.pcap
- Let's start debugging

1_DHCP_FAIL.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

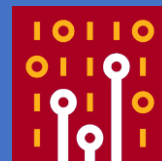
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	0.0.0.0	255.255.255.255	DHCP	590	DHCP Discover - Transaction ID
2	1.044607	10.3.0.1	255.255.255.255	DHCP	367	DHCP Offer - Transaction ID
3	1.057410	10.3.0.1	255.255.255.255	DHCP	367	DHCP ACK - Transaction ID





Open 2 trace files

Recommendation: create Flow Graph of both



1_DHCP_SUCCESS.pcap

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	0.0.0.0	255.255.255.255	DHCP	590	DHCP Discover - Transaction ID 0xfbc0c0
2	0.000512	192.168.2.1	192.168.2.2	DHCP	342	DHCP Offer - Transaction ID 0xfbc0c0
3	0.000926	0.0.0.0	255.255.255.255	DHCP	590	DHCP Request - Transaction ID 0xfbc0c0
4	0.999735	192.168.2.1	192.168.2.2	DHCP	342	DHCP ACK - Transaction ID 0xfbc0c0
5	29.430...	0.0.0.0	255.255.255.255	DHCP	590	DHCP Discover - Transaction ID 0xe1cb5f
6	29.443...	192.168.2.1	192.168.2.2	DHCP	367	DHCP Offer - Transaction ID 0xe1cb5f
7	29.444...	0.0.0.0	255.255.255.255	DHCP	590	DHCP Request - Transaction ID 0xe1cb5f
8	29.458...	192.168.2.1	192.168.2.2	DHCP	367	DHCP ACK - Transaction ID 0xe1cb5f

> Frame 2: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits)
> Ethernet II, Src: 22:22:22:22:22:22 (22:22:22:22:22:22), Dst: Private_11:11:11 (11:11:11:11:11:11)
> Internet Protocol Version 4, Src: 192.168.2.1, Dst: 192.168.2.2
> User Datagram Protocol, Src Port: 67, Dst Port: 68

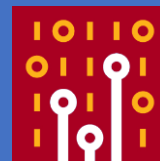
1_DHCP_FAIL.pcap

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	0.0.0.0	255.255.255.255	DHCP	590	DHCP Discover - Transaction ID 0xea928400
2	1.044607	192.168.10.1	255.255.255.255	DHCP	367	DHCP Offer - Transaction ID 0xd03e6d43
3	1.057410	192.168.10.1	255.255.255.255	DHCP	367	DHCP ACK - Transaction ID 0x8044fd8d
4	2.607912	0.0.0.0	255.255.255.255	DHCP	590	DHCP Discover - Transaction ID 0xea928400
5	4.406640	192.168.10.1	255.255.255.255	DHCP	367	DHCP Offer - Transaction ID 0x402ddad8
6	4.421288	192.168.10.1	255.255.255.255	DHCP	367	DHCP ACK - Transaction ID 0x402ddad8

> Frame 1: 590 bytes on wire (4720 bits), 590 bytes captured (4720 bits)
> Ethernet II, Src: Private_11:11:11 (11:11:11:11:11:11), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
> Internet Protocol Version 4, Src: 0.0.0.0, Dst: 255.255.255.255
> User Datagram Protocol, Src Port: 68, Dst Port: 67
> Bootstrap Protocol



Where is the point ?



- DHCP is upgrade version of BOOTP by Microsoft, and RFC defines rough procedures. (Discover-Offer-Request-ACK)
- CATV box, router and small IoT devices uses smaller and incomplete protocol stack.
- Where is the point ? Destination address from packet list ? Umm please look Info and open each packet dissector of Bootstrap protocol.



Transaction ID



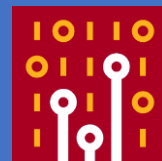
- Transaction ID is the random number chosen by client. And we use the same transaction id in the specific DHCP process.
- Choose Bootstrap Protocol > Transaction ID and right click to Apply as Column
- Check both trace files

```
▼ Bootstrap Protocol (ACK)
  Message type: Boot Reply (2)
  Hardware type: Ethernet (0x01)
  Hardware address length: 6
  Hops: 0
  Transaction ID: 0x8044fd8d
  Seconds elapsed: 0
  > Bootp flags: 0x8000, Broadcast
  Client IP address: 0.0.0.0
  Your (client) IP address: 192.168.1.100
```

Expand Subtrees	Shift+Right
Collapse Subtrees	Shift+Left
Expand All	Ctrl+Right
Collapse All	Ctrl+Left
Apply as Column	Ctrl+Shift+I



Shoot the Bug



1_DHCP_SUCCESS.pcap

No.	Time	Protocol	Length	Transaction ID	Info
1	0.000000	DHCP	590	0xfb0c077c	DHCP Discover
2	0.000512	DHCP	342	0xfb0c077c	DHCP Offer
3	0.000926	DHCP	590	0xfb0c077c	DHCP Request
4	0.999735	DHCP	367	0xfb0c077c	DHCP ACK
5	29.430569	DHCP	590	0xe1cb5e63	DHCP Discover
6	29.443788	DHCP	367	0xe1cb5e63	DHCP Offer
7	29.444266	DHCP	590	0xe1cb5e63	DHCP Request
8	29.458387	DHCP	367	0xe1cb5e63	DHCP ACK

Each DHCP transaction uses the same Transaction ID

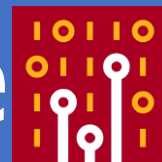
1_DHCP_FAIL.pcap

No.	Time	Protocol	Length	Transaction ID	Info
1	255.255	DHCP	590	0xea928400	DHCP Discover
2	255.255	DHCP	367	0xd03e6d43	DHCP Offer
3	255.255	DHCP	590	0x8044fd8d	DHCP ACK
4	255.255	DHCP	590	0xea928400	DHCP Discover
5	255.255	DHCP	367	0x402ddad8	DHCP Offer
6	255.255	DHCP	367	0x402ddad8	DHCP ACK

DHCP Server returns different Transaction ID that CATV box sent

- DHCP server returns different transaction ID that CATV box sent. The problem is the DHCP server (broadband router) behavior is not appropriate.
- IoT devices uses poor dhcp software implementation

CASE2 Cannot see homepage



- A OSAKA user complains about the trouble some webpage cannot be displayed (the others can). for example, Google OK but Apple NG.
- TOKYO user says he never experienced such trouble in the same environments
- We capture packets in Osaka (2_OSAKA_FAIL_LAN.pcap) and Tokyo (2_TOKYO_SUCCESS_LAN.pcap)





Open 2 trace files

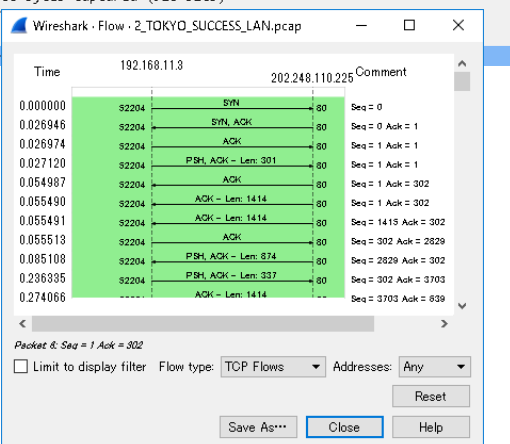
Recommendation: create TCP Flow Graph of both



2_TOKYO_SUCCESS_LAN.pcap

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.11.3	202.248.110.225	TCP	66	52204 → 80 [SYN] Seq=0 Win=8192 Len=0
2	0.026946	202.248.110.225	192.168.11.3	TCP	66	80 → 52204 [SYN, ACK] Seq=0 Ack=1 Win=
3	0.026974	192.168.11.3	202.248.110.225	TCP	54	52204 → 80 [ACK] Seq=1 Ack=1 Win=6645
4	0.027120	192.168.11.3	202.248.110.225	HTTP	355	GET /css/ini.css?121017 HTTP/1.1
5	0.054987	202.248.110.225	192.168.11.3	TCP	60	80 → 52204 [ACK] Seq=1 Ack=302 Win=71
6	0.055490	202.248.110.225	192.168.11.3	TCP	1468	80 → 52204 [ACK] Seq=1 Ack=302 Win=71
7	0.055491	202.248.110.225	192.168.11.3	TCP	1468	80 → 52204 [ACK] Seq=1415 Ack=302 Win=
8	0.055513	192.168.11.3	202.248.110.225	TCP	54	52204 → 80 [ACK] Seq=302 Ack=2829 Win=
9	0.085108	202.248.110.225	192.168.11.3	HTTP	928	HTTP/1.1 200 OK (text/css)

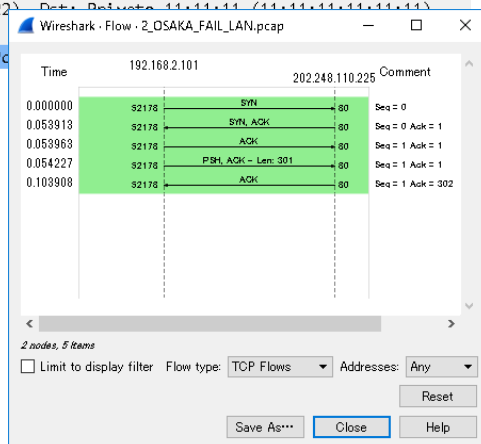
> Frame 1: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
 > Ethernet II, Src: Panasoni_23:e3:20 (7
 > Internet Protocol Version 4, Src: 192.
 > Transmission Control Protocol, Src Port



2_OSAKA_FAIL_LAN.pcap

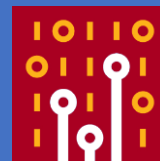
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.2.101	202.248.110.225	TCP	66	52178 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460..
2	0.053913	202.248.110...	192.168.2.101	TCP	66	80 → 52178 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0
3	0.053963	192.168.2.101	202.248.110.225	TCP	54	52178 → 80 [ACK] Seq=1 Ack=1 Win=65700 Len=0
4	0.054227	192.168.2.101	202.248.110.225	HTTP	355	GET /css/ini.css?121017 HTTP/1.1
5	0.103908	202.248.110...	192.168.2.101	TCP	60	80 → 52178 [ACK] Seq=1 Ack=302 Win=7168 Len=0

> Frame 1: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
 > Ethernet II, Src: 22:22:22:22:22:22 (22:22:22:22:22:22)
 > Internet Protocol Version 4, Src: 192.168.2.101, Dst: 202.248.110.225
 > Transmission Control Protocol, Src Port: 52178, Dst Port: 80

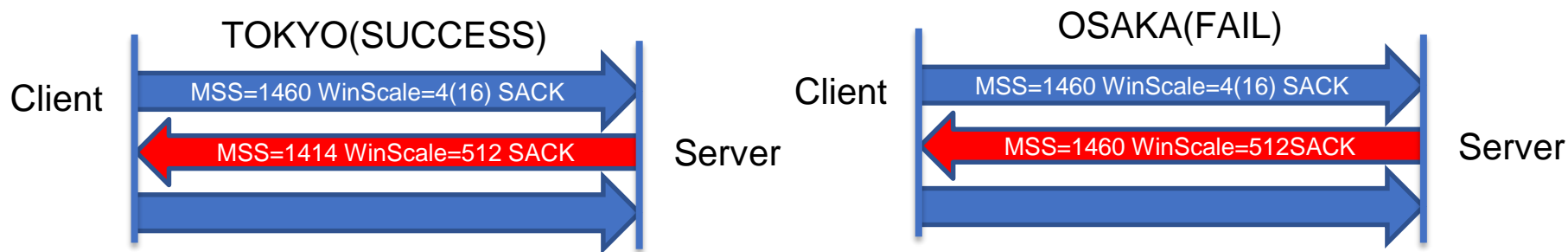




Where is the point ?



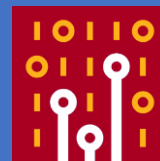
- TCP connection starts with 3 way handshake and exchange parameters (MSS, SACK, Scaling etc.) in first two packet aka TCP negotiation .



- Tokyo's Web Server uses MSS=1414, Osaka 1460



Maximum Segment Size



- MSS(Maximum Segment Size) is calculated
$$\text{MSS} = \text{IP MTU} - \text{IP Header (20)} - \text{TCP Header (20)}$$

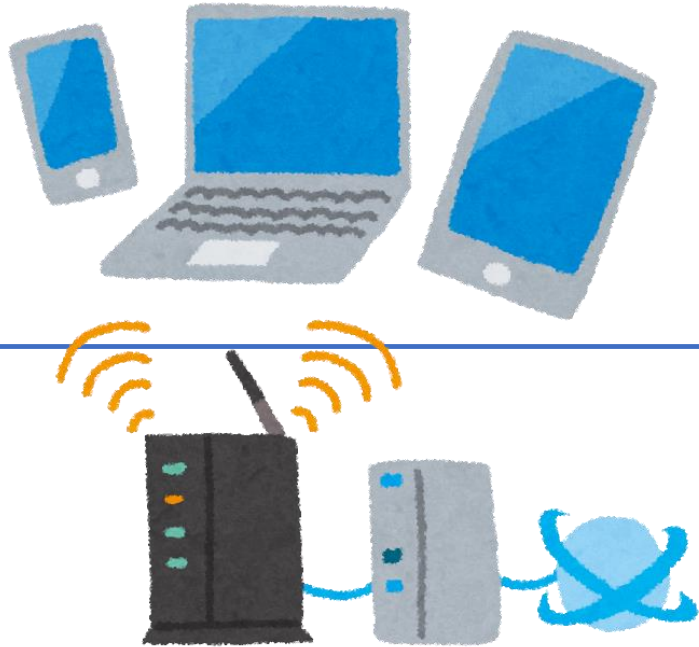
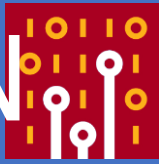
Datalink Header	IP MTU(Maximum Transfer Unit) Size		
	IP Header	TCP Header	MSS (Maximum Segment Size)

each IP and TCP header size without Options is 20

- But why Tokyo is OK and OSAKA is NG ?



Tapping at both LAN and WAN



- To resolve internet connection problem, we need to capture at both boundary point of network. (LAN and WAN side)
- We need router side packets.
- Please add to open additional 2 trace files at the WAN side
2_TOKYO_SUCCESS_WAN.pcap
2_OSAKA_FAIL_WAN.pcap



Open additional 2 trace files

Recommendation: look at TCP negotiation carefully



2_TOKYO_SUCCESS_LAN.pcap

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.11.3	202.248.110.225	TCP	66	52204 → 80 [SYN] Seq=0 Win=8192 Len=0
2	0.026946	202.248.110.225	192.168.11.3	TCP	66	80 → 52204 [SYN, ACK] Seq=0 Ack=1 Win=
3	0.026974	192.168.11.3	202.248.110.225	TCP	54	52204 → 80 [ACK] Seq=1 Ack=1 Win=6645
4	0.027120	192.168.11.3	202.248.110.225	HTTP	355	GET /css/ini.css?121017 HTTP/1.1
5	0.054987	202.248.110.225	192.168.11.3	TCP	60	80 → 52204 [ACK] Seq=1 Ack=302 Win=71
6	0.055490	202.248.110.225	192.168.11.3	TCP	1468	80 → 52204 [ACK] Seq=1 Ack=302 Win=71
7	0.055491	202.248.110.225	192.168.11.3	TCP	1468	80 → 52204 [ACK] Seq=1415 Ack=302 Win=
8	0.055513	192.168.11.3	202.248.110.225	TCP	54	52204 → 80 [ACK] Seq=302 Ack=2829 Win=

2_TOKYO_SUCCESS_WAN.pcap

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	114.167.191.37	202.248.110.225	TCP	74	52204 → 80 [SYN] Seq=0 Win=8192 Len=0
2	0.026314	202.248.110.225	114.167.191.37	TCP	74	80 → 52204 [SYN, ACK] Seq=0 Ack=1 Win=5
3	0.026886	114.167.191.37	202.248.110.225	TCP	62	52204 → 80 [ACK] Seq=1 Ack=1 Win=66456
4	0.027096	114.167.191.37	202.248.110.225	HTTP	363	GET /css/ini.css?121017 HTTP/1.1
5	0.054357	202.248.110.225	114.167.191.37	TCP	62	80 → 52204 [ACK] Seq=1 Ack=302 Win=7168
6	0.054760	202.248.110.225	114.167.191.37	TCP	1476	80 → 52204 [ACK] Seq=1 Ack=302 Win=7168
7	0.054763	202.248.110.225	114.167.191.37	TCP	1476	80 → 52204 [ACK] Seq=1415 Ack=302 Win=7
8	0.055344	114.167.191.37	202.248.110.225	TCP	62	52204 → 80 [ACK] Seq=302 Ack=2829 Win=6
9	0.084476	202.248.110.225	114.167.191.37	HTTP	936	HTTP/1.1 200 OK (text/css)

> Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
 > Ethernet II, Src: Buffalo_03:f2:ff (10:6f:3f:35:f2:ff), Dst: Cisco_99:b5:c1 (00:25:84:99:b5:c1)
 > PPP-over-Ethernet Session
 > Point-to-Point Protocol
 > Internet Protocol Version 4, Src: 114.167.191.37, Dst: 202.248.110.225
 > Transmission Control Protocol, Src Port: 52204, Dst Port: 80, Seq: 0, Len: 0

2_OSAKA_FAIL_LAN.pcap

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.2.101	202.248.110.225	TCP	66	52178 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460..
2	0.053913	202.248.110...	192.168.2.101	TCP	66	80 → 52178 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=
3	0.053963	192.168.2.101	202.248.110.225	TCP	54	52178 → 80 [ACK] Seq=1 Ack=1 Win=65700 Len=0
4	0.054227	192.168.2.101	202.248.110.225	HTTP	355	GET /css/ini.css?121017 HTTP/1.1
5	0.103908	202.248.110...	192.168.2.101	TCP	60	80 → 52178 [ACK] Seq=1 Ack=302 Win=7168 Len=0

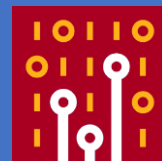
2_OSAKA_FAIL_WAN.pcap

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	180.11.129.64	202.248.110.225	TCP	74	52178 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460
2	0.053524	202.248.110.225	180.11.129.64	TCP	74	80 → 52178 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=
3	0.053885	180.11.129.64	202.248.110.225	TCP	62	52178 → 80 [ACK] Seq=1 Ack=1 Win=65700 Len=0
4	0.054202	180.11.129.64	202.248.110.225	HTTP	363	GET /css/ini.css?121017 HTTP/1.1
5	0.103389	202.248.110.225	180.11.129.64	TCP	62	80 → 52178 [ACK] Seq=1 Ack=302 Win=7168 Len=0

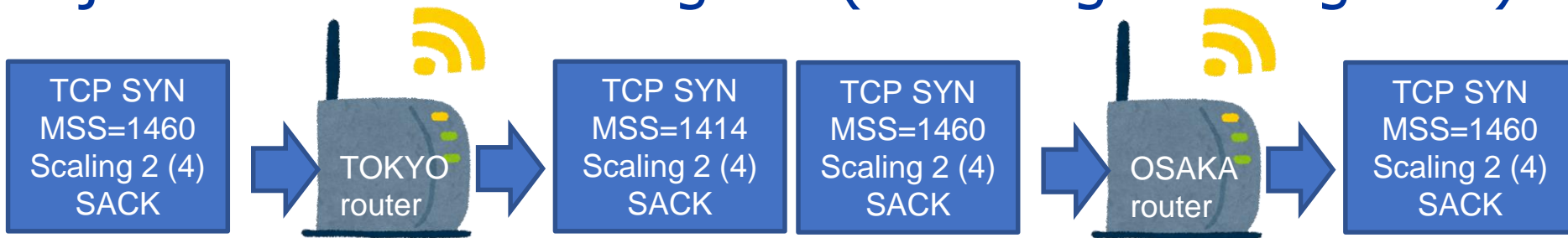
> Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
 > Ethernet II, Src: 44:44:44:44:44:44 (44:44:44:44:44:44), Dst: IPv6mcast_33:33:33:33 (33:33:33:33:33:33)
 > PPP-over-Ethernet Session
 > Point-to-Point Protocol
 > Internet Protocol Version 4, Src: 180.11.129.64, Dst: 202.248.110.225
 > Transmission Control Protocol, Src Port: 52178, Dst Port: 80, Seq: 0, Len: 0



Where is the point ?

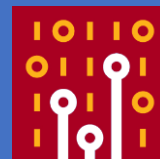


- Both routers in Tokyo and Osaka rewrite Ethernet and IP header (ex. checksum), adds PPPoE and PPP header to connect to access point of the ISP (DSLAM (Digital Subscriber Line Access Multiplexer))
- Osaka router does not adjust MTU and MSS size, just route the IP datagram (including TCP segment)





The Point is in WAN side



#1 in TOKYO LAN Original Client SYN

#1 in TOKYO WAN Router rewrites SYN

Transmission Control Protocol, Src Port: 52204, Dst Port: 80, Seq: 0, Len: 0

Source Port: 52204
Destination Port: 80
[Stream index: 0]
[TCP Segment Len: 0]
Sequence number: 0 (relative sequence number)
[Next sequence number: 0 (relative sequence number)]
Acknowledgment number: 0
1000 = Header Length: 32 bytes (8)

Flags: 0x002 (SYN)

Window size value: 8192
[Calculated window size: 8192]
Checksum: 0x05ac [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0

Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scal

- > TCP Option - Maximum segment size: 1460 bytes
- > TCP Option - No-Operation (NOP)
- > TCP Option - Window scale: 2 (multiply by 4)
- > TCP Option - No-Operation (NOP)
- > TCP Option - No-Operation (NOP)
- > TCP Option - SACK permitted
- > [Timestamps]

Transmission Control Protocol, Src Port: 52204, Dst Port: 80, Seq: 0, Len: 0

Source Port: 52204
Destination Port: 80
[Stream index: 0]
[TCP Segment Len: 0]
Sequence number: 0 (relative sequence number)
[Next sequence number: 0 (relative sequence number)]
Acknowledgment number: 0
1000 = Header Length: 32 bytes (8)

Flags: 0x002 (SYN)

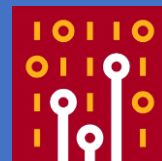
Window size value: 8192
[Calculated window size: 8192]
Checksum: 0xcd52 [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0

Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scal

- > TCP Option - Maximum segment size: 1414 bytes
- > TCP Option - No-Operation (NOP)
- > TCP Option - Window scale: 2 (multiply by 4)
- > TCP Option - No-Operation (NOP)
- > TCP Option - No-Operation (NOP)
- > TCP Option - SACK permitted
- > [Timestamps]



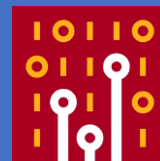
MTU/MSS problem



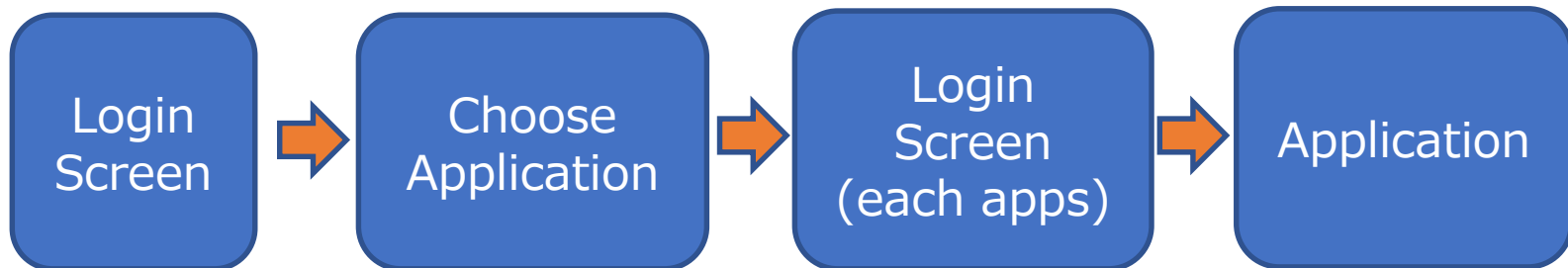
- UDP/IP application uses MTU to fragment each datagrams, and TCP uses MSS to split each segments. TCP sets MSS value in the negotiation.
- MTU and MSS problems are common in Internet connection using PPPoE and PPP datalink.
- We need to modify router (or host) parameters to match ISP's requirements.
- Some application automatically detect and adjust this problem. (ex. PMTU / NDP, and IPv6 !)



CASE3 Slow Single Sign On

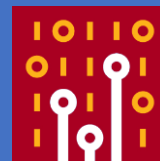


- Many Web based application by 1500 users in enterprise system, each application need to login.
- The customer installed SSO server based on reverse proxy (HTTP proxy authenticates user's application)
- It took 15 seconds to login from remote, but it takes about 100 seconds after SSO, users are angry.



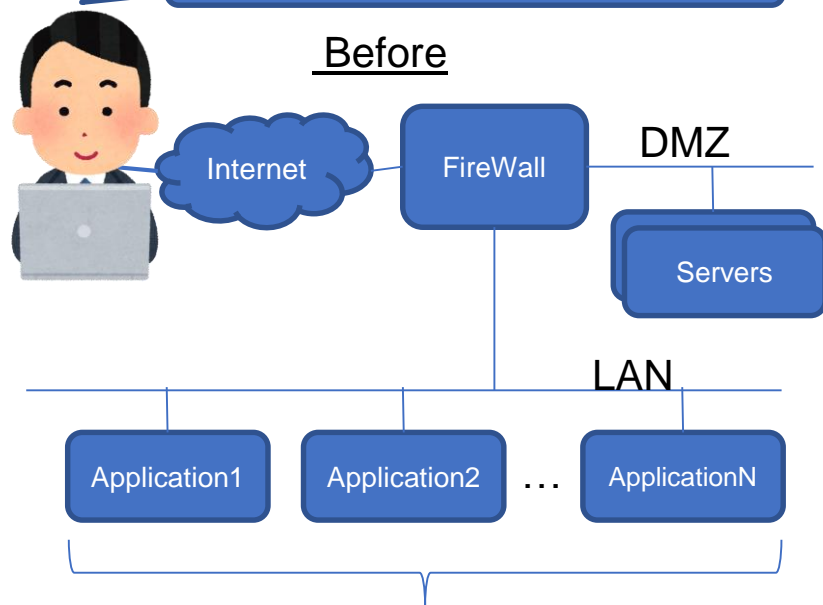


Network diagram



15 seconds to app's login screen

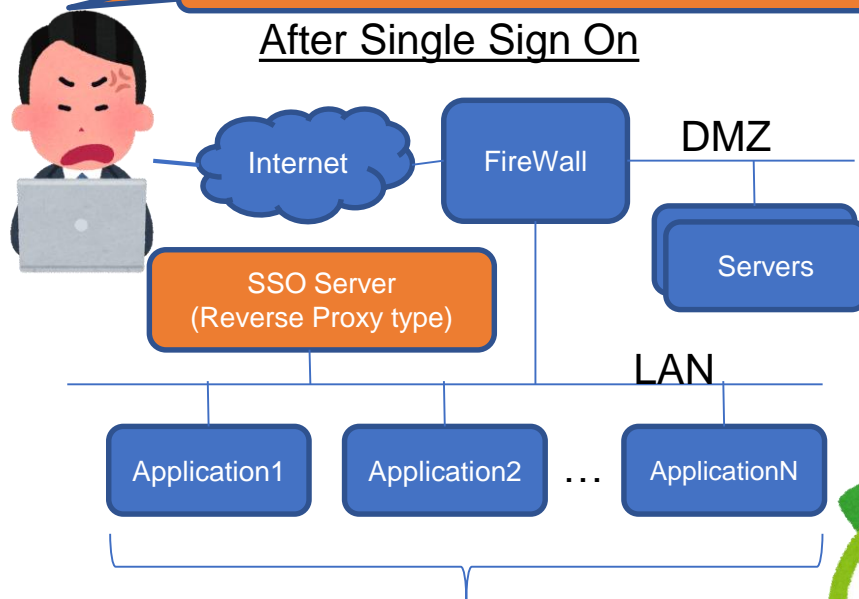
Before



Need to manually and interactive login into each application

100 seconds to app's login screen

After Single Sign On

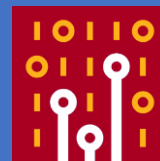


Login at SSO Server once, then you do not need to login into applications !





Capture at user point first

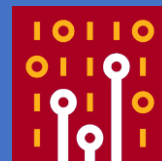


- To check repeatability, we need to capture at user-side point first (sometimes the issue is in the user specific environments, or just user's angry because of another reason.)
- Filter and remove other packet for minimizing and simplifying the trouble, use 2 (Before SSO / After SSO) trace files.
(3_Before.pcap and 3_After.pcap)
NOTE: sorry for inconvenience, all payload data is set to zero using pktanon for security reason
- If you cannot find the key, then we need to check the server-side point, so we can save times and money



Open 2 trace files

Recommendation: create TCP Flow Graph of both



The screenshot shows the Wireshark interface for the file '3_Before.pcap'. The packet list pane displays the following data:

No.	Time	Source	Destination	Protocol	Length	PHY type	Info
1	0.000000	33.203.153.96	173.90.191.95	TCP	66		55515 → 80 [SYN] Seq=0 Win=8192 Len=...
2	0.001286	173.90.191.95	33.203.153.96	TCP	62		80 → 55515 [SYN, ACK] Seq=0 Ack=1 Win=...
3	0.001364	33.203.153.96	173.90.191.95	TCP	60		55515 → 80 [ACK] Seq=1 Ack=65 Win=...
4	0.001852	33.203.153.96	173.90.191.95	TCP	506		55515 → 80 [PSH, ACK] Seq=1 Ack=1 Win=...
5	0.003621	173.90.191.95	33.203.153.96	TCP	1506		80 → 55515 [ACK] Seq=1 Ack=453 Win=...
6	0.003671	173.90.191.95	33.203.153.96	TCP	856		80 → 55515 [PSH, ACK] Seq=1453 Ack=...
7	0.003695	173.90.191.95	33.203.153.96	TCP	60		80 → 55515 [ACK] Seq=1 Ack=453 Win=...
8	0.003745	33.203.153.96	173.90.191.95	TCP	60		55515 → 80 [ACK] Seq=1 Ack=453 Win=...
9	0.003751	33.203.153.96	173.90.191.95	TCP	60		55515 → 80 [ACK] Seq=1 Ack=453 Win=...
10	0.034968	33.203.153.96	173.90.191.95	TCP	60		55515 → 80 [ACK] Seq=1 Ack=453 Win=...
11	0.036068	173.90.191.95	33.203.153.96	TCP	60		80 → 55515 [ACK] Seq=1 Ack=453 Win=...

The packet details pane shows the selected packet (No. 7) with the following details:

- Time: 33.203.153.96
- Destination: 173.90.191.95
- Comment: Seq = 14601 Ack = 444
- Packet 7: Seq = 2295 Ack = 453
- Flow type: TCP Flows
- Addresses: Any

The screenshot shows the Wireshark interface for the file '3_After.pcap'. The packet list pane displays the following data:

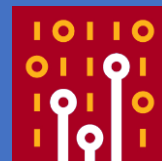
No.	Time	Source	Destination	Protocol	Length	PHY type	Info
1	0.000000	33.203.153.96	173.90.191.18	TCP	66		55860 → 80 [SYN] Seq=0 Win=8192 Len=...
2	0.015084	173.90.191.18	33.203.153.96	TCP	66		80 → 55860 [SYN, ACK] Seq=0 Ack=1 Win=...
3	0.015181	33.203.153.96	173.90.191.18	TCP	60		55860 → 80 [ACK] Seq=1 Ack=1 Win=...
4	0.015595	33.203.153.96	173.90.191.18	TCP	543		55860 → 80 [PSH, ACK] Seq=1 Ack=1 Win=...
5	0.021836	173.90.191.18	33.203.153.96	TCP	1514		80 → 55860 [ACK] Seq=1 Ack=453 Win=...
6	0.021881	173.90.191.18	33.203.153.96	TCP	1514		80 → 55860 [ACK] Seq=1 Ack=453 Win=...
7	0.021907	173.90.191.18	33.203.153.96	TCP	476		80 → 55860 [ACK] Seq=1 Ack=453 Win=...
8	0.021944	33.203.153.96	173.90.191.18	TCP	60		55860 → 80 [ACK] Seq=1 Ack=453 Win=...
9	0.232957	33.203.153.96	173.90.191.18	TCP	60		55860 → 80 [ACK] Seq=1 Ack=453 Win=...
10	2.571910	33.203.153.96	173.90.191.18	TCP	60		55860 → 80 [ACK] Seq=1 Ack=453 Win=...
11	2.573184	173.90.191.18	33.203.153.96	TCP	60		80 → 55860 [ACK] Seq=1 Ack=453 Win=...
12	2.573207	173.90.191.18	33.203.153.96	TCP	60		80 → 55860 [ACK] Seq=1 Ack=453 Win=...
13	2.573278	33.203.153.96	173.90.191.18	TCP	60		55860 → 80 [ACK] Seq=1 Ack=453 Win=...

The packet details pane shows the selected packet (No. 3) with the following details:

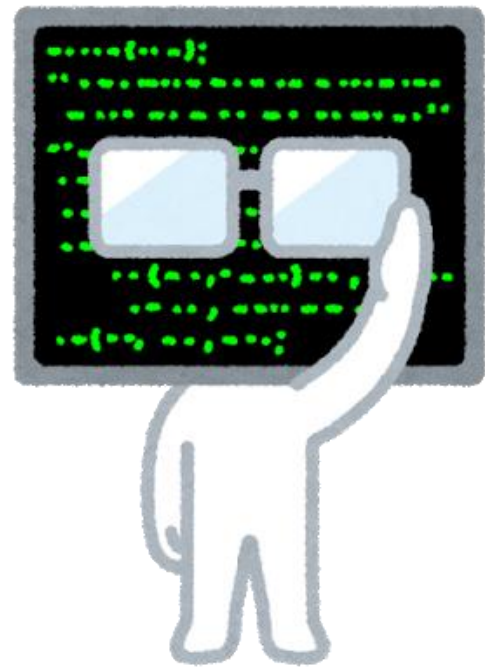
- Time: 33.203.153.96
- Destination: 173.90.191.18
- Comment: Seq = 1074 Ack = 458
- Packet 3: Seq = 1 Ack = 1
- Flow type: TCP Flows
- Addresses: Any



Visualizing TCP Data stream

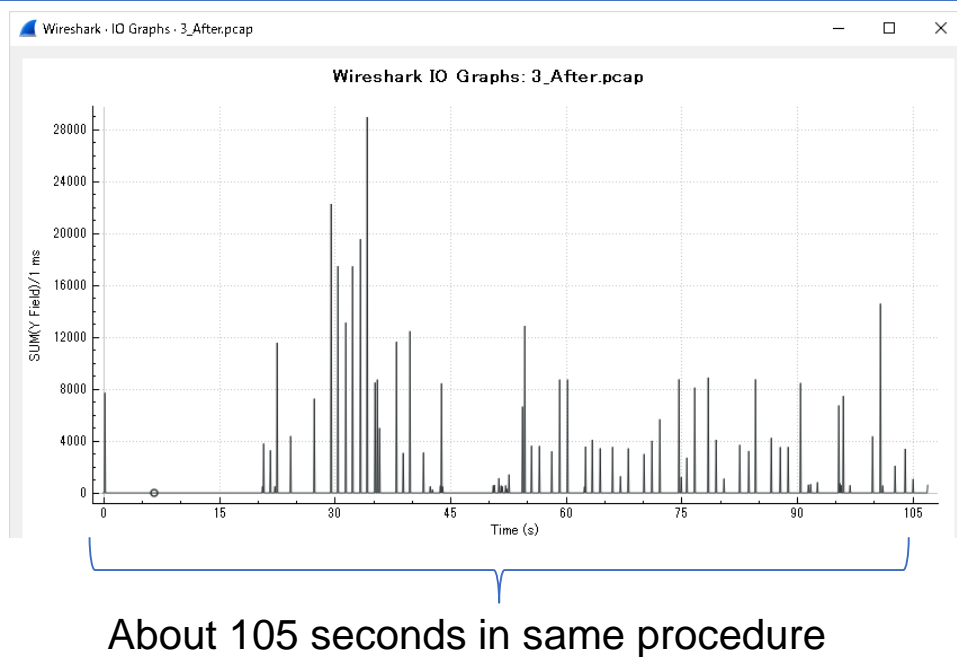
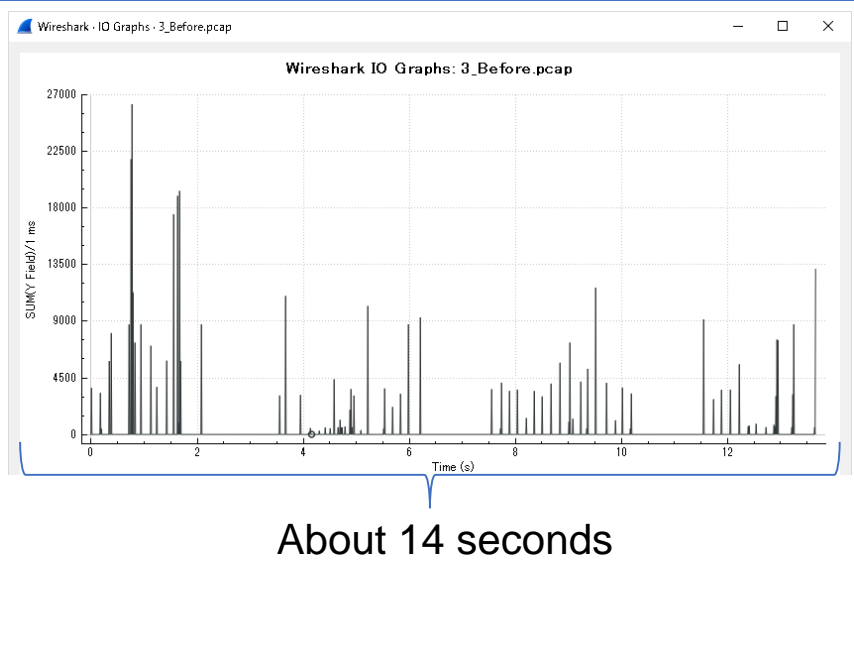


- To understand TCP activities, creating the TCP stream graph is a good idea, but there are many TCP streams in each trace file.
- Using `tcp.analysis.bytes_in_flight` to visualize the TCP by I/O graph (also check TCP pref.)
- Set Y Axis to SUM (Y Field) to set Y Field as `tcp.analysis.bytes_in_flight`
- TCP activity is easy to understand with 1ms interval in common internet infrastructure (Test ping to calculate latency at WAN side)





TCP analysis bytes_in_flight

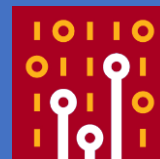


You can visualize the latency and TCP activities, but Why ?



Capture File Properties

Check the Elapsed Time and Bytes



Wireshark · Capture File Properties · 3_Before.pcap

Details

File

Name: C:\Users#megumi\IKERIR\Desktop#SF2018#3_Before.pcap
Length: 662 kB
Format: Wireshark/tcpdump/... - pcap
Encapsulation: Ethernet
Snapshot length: 65535

Time

First packet: 2017-08-23 14:47:07
Last packet: 2017-08-23 14:47:21
Elapsed: 00:00:13

Capture

Hardware: Unknown
OS: Unknown
Application: Unknown

Interfaces

Interface	Dropped packets	Capture filter	Link type	Packet size limit
Unknown	Unknown	Unknown	Ethernet	65535 bytes

Statistics

Measurement	Captured	Displayed	Marked
Packets	1292	495 (38.3%)	-
Time span, s	13.889	13.663	-
Average pps	93.0	36.2	-
Average packet size, B	497	1199	-
Bytes	64205b	59351b (92.4%)	0
Average bytes/s	46 k	43 k	-
Average bits/s	369 k	347 k	-

Wireshark · Capture File Properties · 3_After.pcap

Details

File

Name: C:\Users#megumi\IKERIR\Desktop#SF2018#3_After.pcap
Length: 686 kB
Format: Wireshark/tcpdump/... - pcap
Encapsulation: Ethernet
Snapshot length: 65535

Time

First packet: 2017-08-23 14:58:55
Last packet: 2017-08-23 15:00:42
Elapsed: 00:01:47

Capture

Hardware: Unknown
OS: Unknown
Application: Unknown

Interfaces

Interface	Dropped packets	Capture filter	Link type	Packet size limit
Unknown	Unknown	Unknown	Ethernet	65535 bytes

Statistics

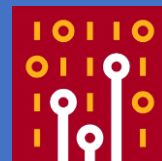
Measurement	Captured	Displayed	Marked
Packets	1433	1433 (100.0%)	-
Time span, s	107.046	107.046	-
Average pps	13.4	13.4	-
Average packet size, B	463	463	-
Bytes	664024	664024 (100.0%)	0
Average bytes/s	6203	6203	-
Average bits/s	49 k	49 k	-

Time, Packets and Bytes are not so different



Where is the point ?

Look into latency of each TCP connection



- Choose Statistics > Conversations and select TCP tab, Look into latency of each TCP connection.
- Sort ascending by Rel Start and check the grey line, the band means the time of each TCP stream like Network tab in Chrome Developer Tool (like Wireshark 3 !)

The screenshot displays a network analysis tool (Wireshark) and a browser's developer tools (Network tab). The Wireshark interface shows a packet capture of a DNS query and response. The Chrome Developer Tools Network tab shows a waterfall chart and a table of resources.

Name	Status	Type	Initiator	Size	Time	Waterfall
www.ikeririne.jp	200	doc...	Other	5.9 ...	202...	
helper.css	200	styl...	www.ike...	(fro...	54 ...	
dropdown.css	200	styl...	www.ike...	(fro...	54 ...	
dropdown.vertical...	200	styl...	www.ike...	(fro...	55 ...	
default.css	200	styl...	www.ike...	(fro...	55 ...	
sitesurvey.css	200	styl...	www.ike...	(fro...	55 ...	



TCP conversation (sort by Rel Start)

Check the grey band and understand each TCP stream



Wireshark · Conversations · 3_Before.pcap

Ethernet · 1		IPv4 · 1		IPv6	TCP · 90				UDP	Rel Start	Duration
Add	Port	Add	Port	Pac	Bytes	Pac	Bytes	Pac	Bytes		
33...	55...	17...	80	11	3356	6	812	5	2544	0.000000	0.0361
33...	55...	17...	80	11	2929	6	799	5	2130	0.161285	0.0305
33...	55...	17...	80	9	1196	5	758	4	438	0.189310	0.0182
33...	55...	17...	80	14	7071	7	908	7	6163	0.329731	0.0813
33...	55...	17...	80	14	6432	7	894	7	5538	0.369324	0.0722
33...	55...	17...	80	15	7173	7	884	8	6289	0.709730	0.0911
33...	55...	17...	80	47	40 k	19	1601	28	38 k	0.740883	0.3101
33...	55...	17...	80	83	77 k	31	2328	52	74 k	0.741147	0.5438
33...	55...	17...	80	34	26 k	15	1368	19	25 k	0.756721	0.2320
33...	55...	17...	80	15	9480	7	889	8	8591	0.756907	0.1052
33...	55...	17...	80	15	8749	7	890	8	7859	0.757097	0.0899
33...	55...	17...	80	12	5157	6	817	6	4340	0.773238	0.0581
33...	55...	17...	80	28	19 k	13	1237	15	18 k	0.773423	0.1829
33...	55...	17...	80	11	3493	6	816	5	2677	0.773671	0.0424
33...	55...	17...	80	15	8787	7	877	8	7910	0.826308	0.0991
33...	55...	17...	80	17	10 k	8	942	9	9544	0.936625	0.1127
33...	55...	17...	80	15	9839	7	886	8	8953	1.123503	0.1137

Wireshark · Conversations · 3_After.pcap

Ethernet · 1		IPv4 · 1		IPv6	TCP · 95				UDP	Rel Start	Duration	Bits/s.
Adc	Por	Adresse	Por	Pac	Bytes	Pac	Byt	Pac	Byte			
3...	5...	173....	80	13	4601	7	909	6	3692	0.000000	2.5733	2825
3...	5...	173....	80	12	3567	6	866	6	2701	20.529745	0.3132	22 k
3...	5...	173....	80	11	2918	6	803	5	2115	21.544630	0.2134	30 k
3...	5...	173....	80	10	1289	6	866	4	423	22.107434	1.7480	3963
3...	5...	173....	80	14	7104	7	956	7	6148	22.432762	1.4353	5328
3...	5...	173....	80	14	6465	7	942	7	5523	24.164929	0.1281	58 k
3...	5...	173....	80	14	7146	7	932	7	6214	27.237261	5.9640	1250
3...	5...	173....	80	50	40 k	21	1...	29	38 k	29.376411	6.2081	2279
3...	5...	173....	80	86	77 k	33	2...	53	74 k	29.376938	6.4103	3114
3...	5...	173....	80	36	26 k	16	1...	20	25 k	30.342772	5.5335	2133
3...	5...	173....	80	17	9633	8	997	9	8636	30.343017	5.1162	1558
3...	5...	173....	80	17	8902	8	998	9	7904	30.343775	5.1019	1564
3...	5...	173....	80	13	5250	7	925	6	4325	31.295568	4.1495	1783
3...	5...	173....	80	30	20 k	14	1...	16	18 k	31.295839	5.5322	1944
3...	5...	173....	80	11	3526	6	864	5	2662	31.296070	4.1486	1666
3...	5...	173....	80	17	8940	8	985	9	7955	32.247220	4.8951	1609
3...	5...	173....	80	19	10 k	9	1...	10	9589	32.247489	4.4750	1877



Each duration of TCP Stream



- Sorting descending by Duration Row, you can also understand the difference of TCP stream.
- Almost over 10 times slower than Before SSO.
- It is the key of the performance problem.

Wireshark · Conversations · 3_Before.pcap

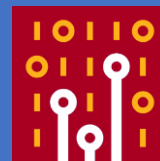
Ethernet · 1		IPv4 · 1		IPv6		TCP · 90		UDP			
Add	Port	Add	Por	Pac	Bytes	Pac	Bytes	Pacl	Bytes	Rel Start	Duration
33...	55...	17...	80	83	77 k	31	2328	52	74 k	0.741147	0.5438
33...	55...	17...	80	46	39 k	19	1599	27	37 k	1.619308	0.3511
33...	55...	17...	80	43	35 k	18	1537	25	34 k	1.549730	0.3290
33...	55...	17...	80	47	40 k	19	1601	28	38 k	0.740883	0.3101
33...	55...	17...	80	34	26 k	15	1368	19	25 k	0.756721	0.2320
33...	55...	17...	80	28	20 k	13	1223	15	19 k	13.658885	0.2299

Wireshark · Conversations · 3_After.pcap

Ethernet · 1		IPv4 · 1		IPv6		TCP · 95		UDP			
Adc	Por	Addr	Por	Pac	Bytes	Pac	Byt	Pac	Byte	Rel Start	Duration
3...	5...	173....	80	86	77 k	33	2...	53	74 k	29.376938	6.4103
3...	5...	173....	80	50	40 k	21	1...	29	38 k	29.376411	6.2081
3...	5...	173....	80	14	7146	7	932	7	6214	27.237261	5.9640
3...	5...	173....	80	36	26 k	16	1...	20	25 k	30.342772	5.5335
3...	5...	173....	80	30	20 k	14	1...	16	18 k	31.295839	5.5322
3...	5...	173....	80	10	1329	6	848	4	481	43.882428	5.5191



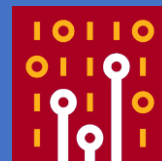
Shoot the trouble



- Increasing Round Trip Time is one of the reasons.
- Sorry All HTTP/HTTPS payloads are anonymized, there is a key in reverse proxy server settings.
- Reverse Proxy SSO server never uses cache, Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0, max-age=0, private, no-transform, proxy-revalidate and Expires: -1
- We need to change reverse proxy server configuration to modify cache settings of SSO server.



CASE 4 Smartphone



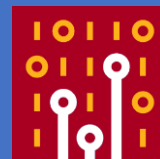
- Only one smartphone fails to connect Wi-Fi, though others can access the network using WPA2-PSK.
- The trouble happens in just onely one client in the entire network, it is a hint to minimize the problem.
- Capture and filter using Smartphone MAC address (22-22-22-22-22-22)
4_WiFi_Fail.pcap (anonymized trace file)





Open trace file 4_WiFi_FAIL

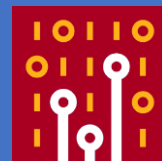
This time please think of layer 2 header



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	Private_11:...	22:22:22:22...	802.11	144	Probe Response, SN=201, FN=0, Flags=....., B
2	0.106669	Private_11:...	22:22:22:22...	802.11	144	Probe Response, SN=203, FN=0, Flags=....., B
3	0.170274	Private_11:...	22:22:22:22...	802.11	144	Probe Response, SN=205, FN=0, Flags=....., B
4	0.182495	Private_11:...	22:22:22:22...	802.11	144	Probe Response, SN=207, FN=0, Flags=....., B
5	0.183849	Private_11:...	22:22:22:22...	802.11	144	Probe Response, SN=207, FN=0, Flags=....R...., B
6	3.147733	22:22:22:22...	Private_11:...	802.11	54	Authentication, SN=11, FN=0, Flags=.....C
7	3.166997	Private_11:...	22:22:22:22...	802.11	54	Authentication, SN=0, FN=0, Flags=.....C
8	3.423790	22:22:22:22...	Private_11:...	802.11	97	Association Request, SN=12, FN=0, Flags=.....
9	3.452944	Private_11:...	22:22:22:22...	802.11	86	Association Response, SN=1, FN=0, Flags=.....
10	7.178897	Private_11:...	22:22:22:22...	802.11	144	Probe Response, SN=287, FN=0, Flags=....., B
11	7.188185	Private_11:...	22:22:22:22...	802.11	144	Probe Response, SN=288, FN=0, Flags=....., B
12	10.003...	22:22:22:22...	Private_11:...	802.11	54	Authentication, SN=15, FN=0, Flags=.....C
13	10.007...	Private_11:...	22:22:22:22...	802.11	54	Authentication, SN=0, FN=0, Flags=.....C
14	10.081...	22:22:22:22...	Private_11:...	802.11	97	Association Request, SN=16, FN=0, Flags=.....
15	10.093...	Private_11:...	22:22:22:22...	802.11	86	Association Response, SN=1, FN=0, Flags=.....
16	13.202...	Private_11:...	22:22:22:22...	802.11	144	Probe Response, SN=349, FN=0, Flags=....., B
17	13.207...	Private_11:...	22:22:22:22...	802.11	144	Probe Response, SN=350, FN=0, Flags=....., B



Association Response



- We need to determine layer 2 or upper. Association Response (`wlan.fc.type_subtype==1`) is a good indicator of Wi-Fi troubleshooting.
- If you find the Association Response, Datalink procedures are almost completed, exchanging Beacon, Probe Request, Probe Response, Authentication x 2, Association Request then link up with Association Response packet
- Let's filter packets with `"wlan.fc.type_subtype==1"`



EAPOL 4 way handshake



- You can find Association Response 3 times ? Why

4_WiFi_FAIL.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

wlan.fc.type_subtype==1

No.	Time	Source	Destination	Protocol	Length	Info
9	3.452944	Private_11:...	22:22:22:22...	802.11	86	Association Response, SN=1, FN=0, Flags=.....C
15	10.093...	Private_11:...	22:22:22:22...	802.11	86	Association Response, SN=1, FN=0, Flags=.....C
21	16.233...	Private_11:...	22:22:22:22...	802.11	86	Association Response, SN=1, FN=0, Flags=.....C

- EAPOL 4 Way handshakes are required in a ordinal WPA-PSK process. Try to filter with "eapol"

4_WiFi_FAIL.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

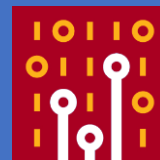
eapol

No.	Time	Source	Destination	Protocol	Length	Info
-----	------	--------	-------------	----------	--------	------



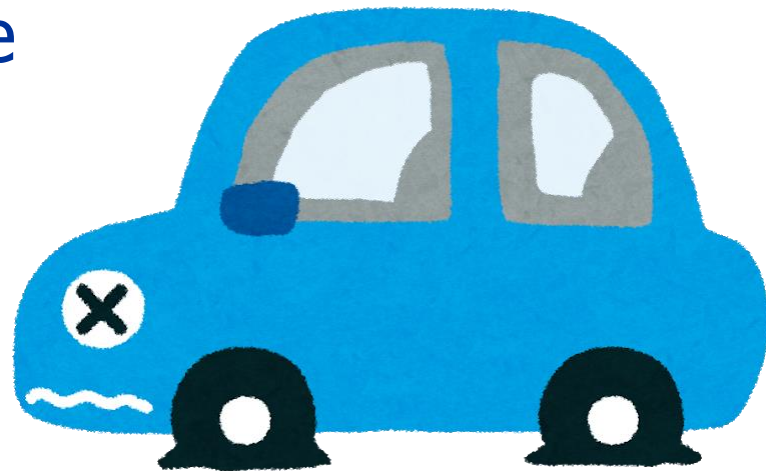
Where is the point ?

Stack point has a problem to shoot the trouble



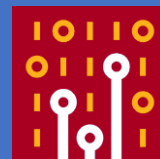
- Finding stack point is important especially in Wireless troubleshooting, in this case, every time the processes are stopped at Association Response.
- Look into Association Response carefully to find the reason

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	Private_11:...	22:22:22:22...	802.11	144	Probe Response, SN=201, FN=0, Flags=....., B
2	0.106669	Private_11:...	22:22:22:22...	802.11	144	Probe Response, SN=203, FN=0, Flags=....., B
3	0.170274	Private_11:...	22:22:22:22...	802.11	144	Probe Response, SN=205, FN=0, Flags=....., B
4	0.182495	Private_11:...	22:22:22:22...	802.11	144	Probe Response, SN=207, FN=0, Flags=....., B
5	0.183849	Private_11:...	22:22:22:22...	802.11	144	Probe Response, SN=207, FN=0, Flags=.....R...., B
6	3.147733	22:22:22:22...	Private_11:...	802.11	54	Authentication, SN=11, FN=0, Flags=.....C
7	3.166997	Private_11:...	22:22:22:22...	802.11	54	Authentication, SN=0, FN=0, Flags=.....C
8	3.423790	22:22:22:22...	Private_11:...	802.11	97	Association Request, SN=12, FN=0, Flags=.....
9	3.457944	Private_11:...	22:22:22:22...	802.11	86	Association Response, SN=1, FN=0, Flags=.....
10	7.178897	Private_11:...	22:22:22:22...	802.11	144	Probe Response, SN=287, FN=0, Flags=....., B
11	7.188185	Private_11:...	22:22:22:22...	802.11	144	Probe Response, SN=288, FN=0, Flags=....., B
12	10.003...	22:22:22:22...	Private_11:...	802.11	54	Authentication, SN=15, FN=0, Flags=.....C
13	10.007...	Private_11:...	22:22:22:22...	802.11	54	Authentication, SN=0, FN=0, Flags=.....C
14	10.081...	22:22:22:22...	Private_11:...	802.11	97	Association Request, SN=16, FN=0, Flags=.....
15	10.093...	Private_11:...	22:22:22:22...	802.11	86	Association Response, SN=1, FN=0, Flags=.....
16	13.202...	Private_11:...	22:22:22:22...	802.11	144	Probe Response, SN=349, FN=0, Flags=....., B
17	13.207...	Private_11:...	22:22:22:22...	802.11	144	Probe Response, SN=350, FN=0, Flags=....., B
18	15.994...	22:22:22:22...	Private_11:...	802.11	54	Authentication, SN=26, FN=0, Flags=.....C





Invalid AKMP (Adaptive Key Management Protocol)



IEEE802.1x authentication method

```
> Radiotap Header v0, Length 20
> 802.11 radio information
> IEEE 802.11 Association Response, Flags: .....C
▼ IEEE 802.11 wireless LAN
  ▼ Fixed parameters (6 bytes)
    > Capabilities Information: 0x0031
    Status code: Invalid AKMP (0x002b)
    ..00 0000 0000 0000 = Association ID: 0x0000
  ▼ Tagged parameters (32 bytes)
    > Tag: Supported Rates 1(B), 2(B), 5.5(B), 11(B), [Mbit/sec]
    > Tag: Vendor Specific: Microsoft Corp.: WMM/WME: Parameter Element
```



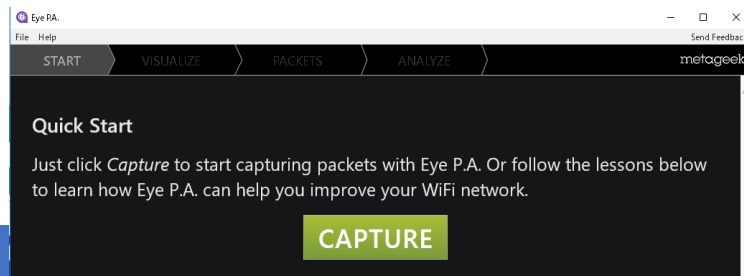
- Invalid AKMP means key exchange settings are mismatch between Access point and Station.
- This time AP use PSK(Pre Shared Key) Passphrase, but STA use IEEE802.1x authentication server.



One more BREAKING NEWS

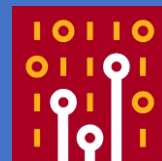


- You may know, Metageek announced Eye P.A. support capturing IEEE802.11n and then IEEE802.11ac in Windows environment !
- After the end of AircapNX era, wireless packet analysis on Windows is limited. (AcrylicWifi and other NDIS based driver never behaves as we wish...)

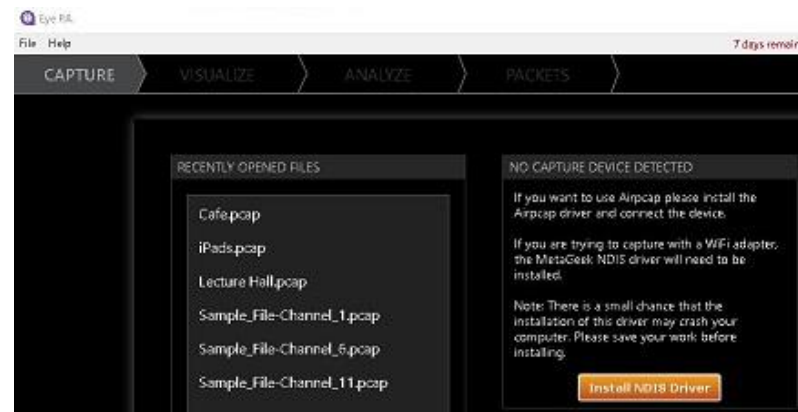




Eye P.A. (Eye Packet Analyzer)

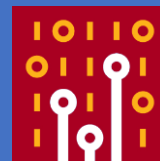


- You need Beta version of Eye P.A. now EyeP.A. supports 11n with v1.12 then 11AC support comes at last week !!
- Start up EyePA with Administrator privilege. Install NDIS Driver
- Then you can capture via **CAPTURE DEVICE CONTROL**
- Off course you can save and send trace to Wireshark !





Compatible Adapters (1)



- Eye P.A. ver1.12 (May 2018) supports 11N
Linksys AE2500, Linksys AE1200, Netgear A6200
(NDIS based driver)
Riverbed AirPcapNX, AirPcapTX
(AirPcap based driver)
- Tarlogic NDIS Monitor Driver
and AirPcap driver are used
when we use 11n in ver1.12

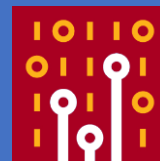
この接続は次の項目を使用します(O):

- Microsoft ネットワーク用クライアント
- VMware Bridge Protocol
- Microsoft ネットワーク用ファイルとプリンター共有
- Npcap Packet Driver (NPCAP)
- QoS パケット スケジューラ
- Tarlogic NDIS Monitor Driver





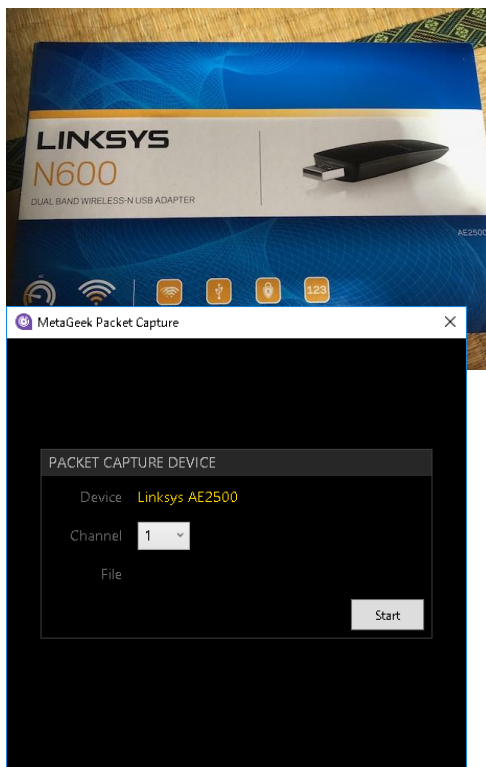
Compatible Adapters (2)



- Eye P.A. Version 1.13 (A.k.a Imperial Eye P.A.)
- Support adapter:
ASUS USB-AC56, ASUS USB-AC68, ALFA Network AWUS1900, Linksys WUSB6300, Amped Wireless ACA1, EnGenius EUB1200AC, D-Link DWA-182 rev C1, D-Link DWA-192, TRENDnet TEW-805UB, TP-LINK Archer T4U v2, TP-LINK Archer T4UH v2, Edimax EW-7822UAC, Edimax EW-7833UAC



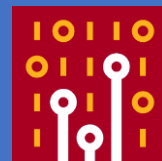
TEST: Linksys AE2500



- Linksys AE2500 Dual Band N USB3 2.4/5GHz USB2 BCM4323 (Broadcom chip)
- CH 1-14, CH 36-64 W52, CH 100-165 W52 W53 in Japan
- Automatically recognized in Eye P.A. version 1.13.0.13
- You can choose Packet Data is truncated or not



TEST: Linksys AE2500



ACTIVE SELECTION

Start: 19:05:24.375 SSIDs: 8
End: 19:06:11.263 Clients: 15
Duration: 46.887 ms Bytes: 1,008,628
Air Time: 1,527 ms Packets: 7,170
Effective Data Rate: 20.4 Mbps Retry Rate: 6%

Data Rates by Percentage

0 Mbps 8667 Mbps

ASSOCIATED DATA

ESSID	BSSIDs	Clients	Air Time	Effective Data Rate	Retry Rate
ikeriri-wim-ax	1	6	501	--	0
at-erm-e3d8b7-a	1	10	263	20.4	10
at-erm-e3d8b7-g	1	3	219	--	0
ikeriri-wim-ax2	1	1	203	--	0
Extender-A-E050	1	2	187	--	0
at-erm-e3d8b7-aw	1	2	150	--	0
UNRESOLVED	1	2	4	--	0
BROADCAST	1	1	0	--	0

PACKETS

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000		NecPlatf_19...	802.11	37	Acknowledgement, Flags=.....C
2	0.026000	Modacom_a8:...	Broadcast	802.11	273	Beacon frame, SN=4085, FN=0, Flags=.....C, BI
3	0.029000	Buffalo_6a:...	PlanexCo_ec...	802.11	46	VHT NDP Announcement, Flags=.....C
4	0.039000	9a:f1:99:19:...	Broadcast	802.11	216	Beacon frame, SN=2811, FN=0, Flags=.....C, BI
5	0.043000	Buffalo_6a:...	Broadcast	802.11	276	Beacon frame, SN=3287, FN=0, Flags=.....C, BI
6	0.044000	Buffalo_6a:...	Broadcast	802.11	326	Beacon frame, SN=3288, FN=0, Flags=.....C, BI

Packet 3 Details:

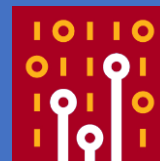
- Frame 3: 46 bytes on wire (368 bits), 46 bytes captured (368 bits)
- Radiotap Header v0, Length 23
- 802.11 radio information
- IEEE 802.11 VHT NDP Announcement, Flags:
- Type/Subtype: VHT NDP Announcement (0x0015)
- Frame Control Field: 0x5400
- .000 0000 0110 0100 = Duration: 100 microseconds
- Receiver address: PlanexCo_ec:ff:38 (00:22:cf:ec:ff:38)

0000 00 00 17 00 2f 00 00 00 6d 74 00 00 00 00 00 00/... mt.....
0010 18 30 3c 14 50 01 c5 54 00 64 00 00 22 cf ec ff .<P>T .d.....
0020 38 88 57 ee 6a e0 55 10 01 00 1d a2 7e fe 8-W-j-U.....

Frame (frame), 46 bytes Packets: 7372 · Displayed: 7372 (100.0%) Profile: Default



Compatible Adapters (3)

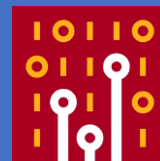


- Eye P.A. ver2 supports 11AC
- Supported adapters (IEEE802.11ac)
Linksys WUSB6300 (recommended), ASUS USB-AC56, ASUS USB-AC68, ALFA Network AWUS1900, Amped Wireless ACA1, EnGenius EUB1200AC, D-Link DWA-182 rev C1, D-Link DWA-192, TRENDnet TEW-805UB, TP-LINK Archer T4U v2, TP-LINK Archer T4UH v2, Edimax EW-7822UAC, Edimax EW-7833UAC
- 802.11n adapters
Linksys AE2500 (recommended), Linksys AE1200, Netgear A6200
NOTE LinksysAE1200 and NetgearA6200 cannot be used on DFSCH
- RiverBed AirPcap NX, AirPcapTX
- Tamosoft (famous as TamoGraph) driver is used.





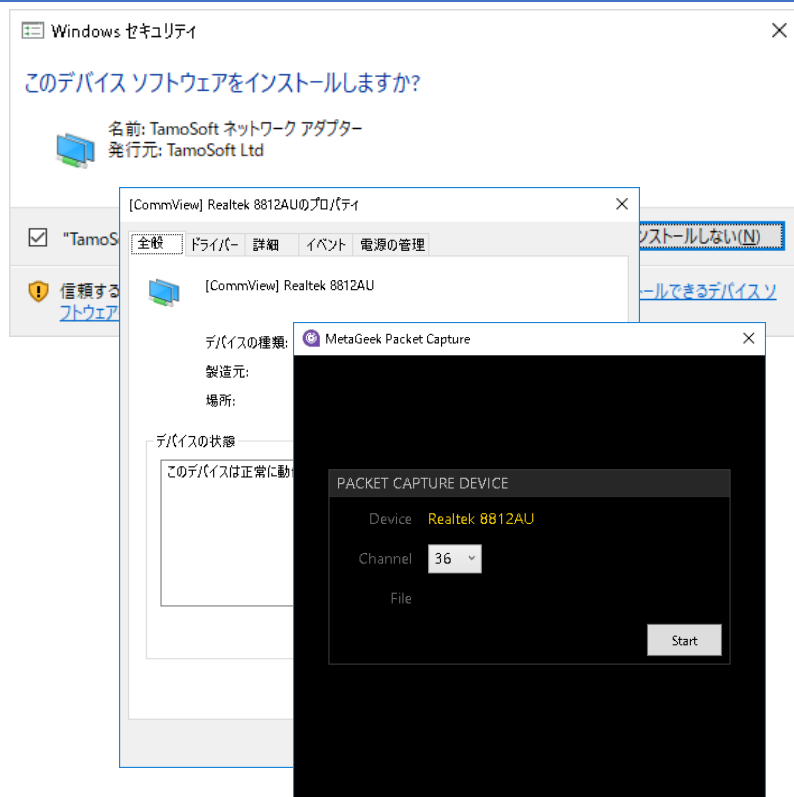
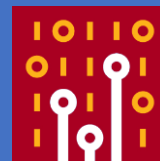
TEST2 : ALFA AWUS036ACH



- Alfa Long-Range Dual-Band AC1200 Wireless USB 3.0 Wi-Fi Adapter w/2x 5dBi External Antennas – 2.4GHz 300Mbps/5GHz 867Mbps – 802.11ac & A, B, G, N
- RTL8812AU Realtek chipset
- CH 1-14, CH 36-64 W52, CH 100-165 W52 W53 in Japan



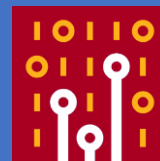
TEST2 : ALFA AWUS036ACH



- This ALFA model is not officially supported, but it try to install specific drivers
- Automatically recognized in Eye P.A. version 1.13.0.13
- You can choose Packet Data is truncated or not



DEMO : ALFA AWUS036ACH



The screenshot displays the Wireshark interface with two windows open. The main window shows the 'PACKETS' pane with a list of captured packets. The 'ASSOCIATED DATA' pane shows a table of wireless networks. The 'AIR TIME' pane shows a graph of air time usage. The 'TREEPIES' pane shows two donut charts representing data and air time distribution.

ASSOCIATED DATA Table:

ESSID	BSSIDs	Clients	Air Time	Effective Data Rate	Retry Rate	
ikeriri-wim-ax		1	8	538	281.2	0
aterm-e3d8b7-a		1	8	156	8.2	1
aterm-e3d8b7-g		1	7	167	8.8	14
ikeriri-wim-ax2		1	1	104	--	0
Extender-A-E050		1	2	96	--	0
aterm-e3d8b7-aw		1	1	73	--	0
UNRESOLVED		1	2	1	--	0

PACKETS Table:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	Buffalo_6a:...	Broadcast	802.11	281	Beacon frame, SN=4059, FN=0, Flags=....., BI=...
2	0.000000	Buffalo_6a:...	Broadcast	802.11	331	Beacon frame, SN=4060, FN=0, Flags=....., BI=...
3	0.017000	RivetNet_e8:...	Modacm_94:...	802.11	148	QoS Data, SN=3945, FN=0, Flags=p.....T[Packet s...
4	0.017000	Modacm_a8:...	RivetNet_e8:...	802.11	60	802.11 Block Ack, Flags=.....
5	0.017000	Modacm_a8:...	Broadcast	802.11	310	Beacon frame, SN=392, FN=0, Flags=....., BI=1...
6	0.017000	Modacm_a8:...	Modacm_a8:...	802.11	42	Acknowledgement, Flags=.....

Packet 1 Details:

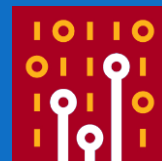
- Frame 1: 281 bytes on wire (2248 bits), 281 bytes captured (2248 bits)
- PPI version 0, 32 bytes
- 802.11 radio information
- IEEE 802.11 Beacon frame, Flags:
- IEEE 802.11 wireless LAN

Packet 1 Hex:

```
0000 00 00 20 00 69 00 00 00 02 00 14 00 c5 75 00 00  ..i...  ....u..
0010 06 79 02 00 00 00 0c 00 3c 14 50 01 00 00 b6 a2  .y..... <P.....
0020 80 00 00 00 ff ff ff ff ff ff 88 57 ee 6a e0 54  .....  ..W.j.T
0030 88 57 ee 6a e0 54 b0 fd a3 20 ba 35 09 00 00 00  .W.j.T  ..5....
```



Thank you for attending !



どうもありがとうございました



Q&A



いけりり★ネットワークサービス

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