

## Assignment 1 - Wireshark

COMP 307 - Principles of Web Development

Prof. Joseph Vybihal

Fall 2018

LE, Nhat Hung

McGill ID: 260793376

Date: September 14, 2018

Due date: September 17, 2018

1. Identify the exact locations you were when during your two Wi-Fi packet recording sessions.

Public network: Sherbrooke Tim Hortons.

Home network: At home in my bed.

2. Experiment 1 (at home)

a. Select a packet and include screenshots.

No.	Delta	Time	Source	Destination	Protocol	Length	Info
351	0.015245279	41.763300877	104.208.165.109	192.168.0.103	TCP	74	443 → 44600 [SYN, ACK]
352	0.000046844	41.763347721	192.168.0.103	104.208.165.109	TCP	66	44600 → 443 [ACK] Seq=
353	0.000478701	41.763826422	192.168.0.103	104.208.165.109	TLsv1.2	583	Client Hello
354	0.043456035	41.807282457	104.208.165.109	192.168.0.103	TCP	1514	443 → 44600 [ACK] Seq=
355	0.000041650	41.807324107	192.168.0.103	104.208.165.109	TCP	66	44600 → 443 [ACK] Seq=

Figure 1: Selected packet highlighted in blue.

```
▶ Frame 353: 583 bytes on wire (4664 bits), 583 bytes captured (4664 bits) on interface 0
▶ Ethernet II, Src: HonHaiPr_f4:cf:8b (48:e2:44:f4:cf:8b), Dst: ZyxelCom_d2:5d:ce (b8:ec:a3:d2:5d:ce)
▶ Internet Protocol Version 4, Src: 192.168.0.103, Dst: 104.208.165.109
▶ Transmission Control Protocol, Src Port: 44600, Dst Port: 443, Seq: 1, Ack: 1, Len: 517
▶ Secure Sockets Layer
```

Figure 2: Packet info.

```
0000 b8 ec a3 d2 5d ce 48 e2 44 f4 cf 8b 08 00 45 00 ...] H D ... E
0010 02 39 ec 73 40 00 40 06 7c fe c0 a8 00 67 68 d0 ...9 s @ | ... gh
0020 a5 6d ae 38 01 bb 46 c1 62 97 86 94 06 dc 80 18 ...m 8 F b ...
0030 00 e5 2a a1 00 00 01 01 08 0a c2 9c 98 8a 03 ce ...* ...
0040 99 3e 16 03 01 02 00 01 00 01 fc 03 03 aa 71 2e ...> ... q.
0050 42 04 59 35 32 d4 b9 89 9d e3 39 24 14 1f ed bd B Y52 ... 9$ ...
0060 26 ad 2f f8 8f 56 75 bd 87 ab 84 eb f8 20 72 11 & / V u ... r
0070 00 00 3f d5 e1 23 35 61 76 38 4c 18 b3 31 6e 26 ...? #5a v8L ...1n&
0080 a1 e4 15 fd 05 85 d7 77 91 a8 d6 c1 f3 59 00 22 ...w ... Y.
0090 ea ea 13 01 13 02 13 03 c0 2b c0 2f c0 2c c0 30 ...+ / , 0
00a0 cc a9 cc a8 c0 13 c0 14 00 9c 00 9d 00 2f 00 35 .../ 5
00b0 00 0a 01 00 01 91 2a 2a 00 00 ff 01 00 01 00 00 ...**
00c0 00 00 19 00 17 00 00 14 73 74 61 74 69 63 2e 61 ...static.a
00d0 73 6d 2e 73 6b 79 70 65 2e 63 6f 6d 00 17 00 00 sm.skype .com ...
00e0 00 23 00 00 00 0d 00 14 00 12 04 03 08 04 04 01 ...# ...
00f0 05 03 08 05 05 01 08 06 06 01 02 01 00 05 00 05 ...
0100 01 00 00 00 00 00 12 00 00 00 10 00 0e 00 0c 02 ...
0110 68 32 08 68 74 74 70 2f 31 2e 31 00 0b 00 02 01 h2-http/ 1.1 ...
0120 00 00 33 00 2b 00 29 8a 8a 00 01 00 00 1d 00 20 ...3 + )
0130 96 77 0b 11 fd 91 2f 69 6e 4c 82 35 87 4e db d9 ...w ... /i nL 5 N
0140 cc 10 31 c7 00 5c 0d f4 98 00 4d 1e 37 f5 2c 24 ...1 \ ... M 7 , $
0150 00 2d 00 02 01 01 00 2b 00 0b 0a ca ca 7f 17 03 ...+
0160 03 03 02 03 01 00 0a 00 0a 00 08 8a 8a 00 1d 00 ...
0170 17 00 18 ea ea 00 01 00 00 15 00 cb 0a 00 00 00 ...
0180 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ...
0190 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ...
01a0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ...
01b0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ...
01c0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ...
01d0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ...
01e0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ...
01f0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ...
0200 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ...
0210 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ...
0220 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ...
0230 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ...
0240 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ...
```

Figure 3: Packet's raw data.

b. Identify all the fields contained within the packet. Compare with lecture slides.

Will now compare with the following figures, taken from lecture slides:

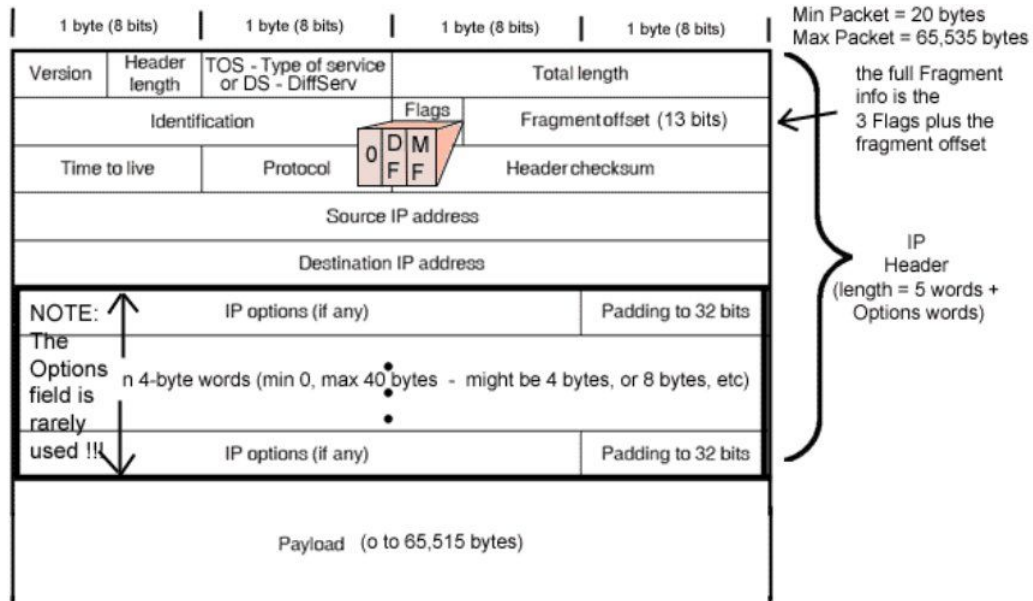


Figure 4: Packet data structure

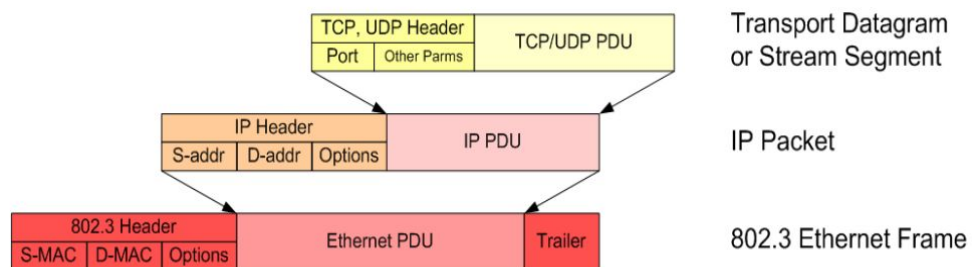


Figure 5: Nested structures of packets

Now identifying fields through the Wireshark packet info pane:

Fields present in figure 4		
IP Header	Version, header length & DiffServ	0100 .... = Version: 4 .... 0101 = Header Length: 20 bytes (5) Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
	Total length	Total Length: 569
	Identification	Identification: 0xec73 (60531)

	Flags & fragment offset	Flags: 0x4000, Don't fragment 0... .. = Reserved bit: Not set .1... .. = Don't fragment: Set ..0... .. = More fragments: Not set ...0 0000 0000 0000 = Fragment offset: 0
	Time to live, protocol & header checksum	Time to live: 64 Protocol: TCP (6) Header checksum: 0x7cfe [validation disabled]
	Source & destination IP addresses	Source: 192.168.0.103 Destination: 104.208.165.109
Payload		TCP payload (517 bytes) Secure Sockets Layer ▼ TLSv1.2 Record Layer: Handshake Protocol: Client Hello Content Type: Handshake (22) Version: TLS 1.0 (0x0301) Length: 512 ► Handshake Protocol: Client Hello
Fields present in figure 5		
802.3 header	Source & destination MAC addresses	Destination: ZyxelCom_d2:5d:ce (b8:ec:a3:d2:5d:ce) Source: HonHaiPr_f4:cf:8b (48:e2:44:f4:cf:8b)
TCP, UDP header	Source & destination ports	Source Port: 44600 Destination Port: 443
	“Other parameters”, which from the Wireshark packet info pane include: <ul style="list-style-type: none"> <li>• TCP segment length</li> <li>• Sequence number</li> <li>• Acknowledgement number</li> <li>• Header length</li> <li>• Flags</li> <li>• Window size</li> <li>• Checksum</li> <li>• Urgent pointer</li> <li>• Various options</li> </ul>	[TCP Segment Len: 517] Sequence number: 1 (relative sequence number) [Next sequence number: 518 (relative sequence number)] Acknowledgment number: 1 (relative ack number) 1000 .... = Header Length: 32 bytes (8) Flags: 0x018 (PSH, ACK) Window size value: 229 [Calculated window size: 29312] [Window size scaling factor: 128] Checksum: 0x2aa1 [unverified] [Checksum Status: Unverified] Urgent pointer: 0 Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps

The packet also contains other fields not represented in the two figures from the lecture slides.

These fields are more specific to the protocol, which in this case is TLSv1.2 (Transport Layer Security, a newer and more secure version of SSL). The basic unit of data in SSL (or TLS) is a **record**. The record in this case is the (TCP) payload represented in the table above.

An SSL record consists of

- A 5 bytes **record header**, containing metadata, similar to TCP headers or IP headers. This packet's SSL record header contains the following fields
  - Content type
  - Protocol version
  - Message length

```

    TLSv1.2 Record Layer: Handshake Protocol: Client Hello
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 512

```

Figure 6: The packet's record header

- The actual data/message. From the content type specified above, we know the data here is a handshake.

```

Handshake Protocol: Client Hello
  Handshake Type: Client Hello (1)
  Length: 508
  Version: TLS 1.2 (0x0303)
  Random: aa712e4204593532d4b9899de33924141fedbd26ad2ff88f...
  Session ID Length: 32
  Session ID: 721100003fd5e123356176384c18b3316e26a1e415fd0585...
  Cipher Suites Length: 34
  Cipher Suites (17 suites)
  Compression Methods Length: 1
  Compression Methods (1 method)
  Extensions Length: 401
  Extension: Reserved (GREASE) (len=0)
  Extension: renegotiation_info (len=1)
  Extension: server_name (len=25)
  Extension: extended_master_secret (len=0)
  Extension: SessionTicket TLS (len=0)
  Extension: signature_algorithms (len=20)
  Extension: status_request (len=5)
  Extension: signed_certificate_timestamp (len=0)
  Extension: application_layer_protocol_negotiation (len=14)
  Extension: ec_point_formats (len=2)
  Extension: key_share (len=43)
  Extension: psk_key_exchange_modes (len=2)
  Extension: supported_versions (len=11)
  Extension: supported_groups (len=10)
  Extension: Reserved (GREASE) (len=1)
  Extension: padding (len=203)

```

Figure 7: The handshake

c. What information does this packet contain?

Where it came from:

The packet's source MAC address, 48:e2:44:f4:cf:8b, matches my pc's MAC address. The packet then came from my own computer.

```

ether 48:e2:44:f4:cf:8b txqueuelen 1000 (Ethernet)

```

Figure 8: MAC address obtained from running ifconfig in command line

Where it's going:

The packet was heading to Microsoft. We can see this through three points.

Exhibit A: Pasting the destination IP address, 104.208.165.109, into Google shows a first page full of IP tracing sites pointing to Microsoft.

Exhibit B: Looking inside the packet data, we find the name the of the destination server: static.asm.skype.com. Skype is a Microsoft product.

```

Server Name Indication extension
  Server Name list length: 23
  Server Name Type: host_name (0)
  Server Name length: 20
  Server Name: static.asm.skype.com

```

Figure 8: Inside the packet's payload

Exhibit C: Wireshark's packet list indicates relationships between captured packets. It shows our selected packet is part of a conversation. The subsequent packets in the conversation show clear traces of Microsoft. Case in point the packet which directly follows our packet is an acknowledgement packet. In it is undeniable proof:

353	0.000478701	41.763826422
354	0.043456035	41.807282457

Figure 9: Our packet, no. 353, is acknowledged by the highlighted packet, as shown by the check mark

```

·U···Washington
1·0···U···Redmond
nd1·0···U···Microsoft Corporation
on1·0···U···Microsoft IT
···Microsoft IT
TLS CA 50···170
82914374 7Z··1908
29143747 Z0·1·0··
·U···static.asm
.skype.com0··"0·

```

Figure 10: Packet no. 354's raw data

```

000100 UUS10U
Washington10URedmond10U
Microsoft Corporation10UMicrosoft IT10UMicrosoft IT TLS CA 50
170829143747Z
190829143747Z010Ustatic.asm.skype.com0"0

```

Figure 11: Enhanced (actually just copy pasted into a text editor)

In conclusion the packet's destination is Microsoft, more specifically for a Skype service.

What security it's using:

The packet uses SSL encryption.

What message it contains:

The packet's record header indicates it's a Client Hello message sent to the server as the start of a handshake protocol. It sends the following information (as shown in figure 7):

- Its protocol version is TLS 1.2
- The session ID the client wishes to use for this connection
- A cipher suite containing the combinations of cryptographic algorithms supported by the client
- Compression methods - a list of compression algorithms supported by the client



In summary, the Client Hello message shares the above information and awaits a responding Server Hello, in order to agree on a protocol version, select cryptographic algorithms, optionally authenticate each other, and use public key encryption techniques in order to generate shared secrets.

This packet is then trying to start a secure connection with Microsoft through a handshake protocol.

### 3. Experiment 2 (at a public hotspot)

a. Select and sort by a sender IP address and try to deduce what they were trying to do.

Now picking sender IP 172.18.20.47.

Source	Destination	Protocol	Length	Info
172.18.20.47	31.13.80.5	TCP	142	39070 → 443 [ACK]
172.18.20.47	31.13.80.5	TCP	158	[TCP Dup ACK 5524]
172.18.20.47	31.13.80.5	TCP	142	[TCP Dup ACK 5524]
172.18.20.47	31.13.80.5	TCP	158	[TCP Dup ACK 5524]

Figure 12: Sorted by source IP - the list goes further

Pasting the destination IP into Google reveals it to be Facebook's. Further down the list is a TLS Client Hello packet, part of the handshake protocol. This packet's payload contains the following line:

Server Name: scontent.fykz2-1.fna.fbcdn.net

Fbcdn.net is a Facebook domain used to serve static content, like media, from a content delivery network (CDN).

Among the following packets, some destination IP's are Google's. Their protocol is TLS, the payload being "application data". A Google service that uses TLS by default is Gmail. We can then guess that the user is also using Gmail.

We can then deduce the user is interacting with Facebook, specifically logging into their account judging by the Client Hello packet, used to exchange security related information and using Gmail at the same time.

b. Include a screenshot of the sorted top summary panel for the above sender IP.

No.	Delta	Time	Source	Destination	Protocol	Length	Info
5524	0.000364	3.789468	172.18.20.47	31.13.80.5	TCP	142	39070 → 443 [ACK]
5525	0.000074	3.789542	172.18.20.47	31.13.80.5	TCP	158	[TCP Dup ACK 5524#]
5528	0.000082	3.789711	172.18.20.47	31.13.80.5	TCP	142	[TCP Dup ACK 5524#]
5529	0.000043	3.789754	172.18.20.47	31.13.80.5	TCP	158	[TCP Dup ACK 5524#]
11665	0.000043	7.552752	172.18.20.47	31.13.80.5	TCP	142	39070 → 443 [ACK]
11666	0.000045	7.552797	172.18.20.47	31.13.80.5	TCP	142	39070 → 443 [ACK]
11667	0.000044	7.552841	172.18.20.47	66.102.1.188	TCP	146	48816 → 443 [ACK]
11690	0.000081	7.559689	172.18.20.47	31.13.71.34	TCP	154	36038 → 443 [ACK]
11691	0.000048	7.559737	172.18.20.47	66.102.1.188	TCP	154	[TCP Dup ACK 11667]
11692	0.000043	7.559780	172.18.20.47	31.13.80.5	TCP	154	[TCP Dup ACK 11666]
11693	0.000044	7.559824	172.18.20.47	31.13.71.34	TCP	158	[TCP Dup ACK 11690]
11709	0.000068	7.564638	172.18.20.47	31.13.80.5	TLSv1.2	181	Application Data
11867	0.000042	7.611210	172.18.20.47	31.13.80.5	TLSv1.2	200	Application Data
12151	0.000043	7.733797	172.18.20.47	31.13.71.34	TLSv1.2	179	Application Data
12458	0.000049	7.944157	172.18.20.47	31.13.80.5	TCP	146	39070 → 443 [ACK]
12803	0.000049	8.109066	172.18.20.47	207.219.36.81	TCP	154	46804 → 443 [SYN]
12840	0.000049	8.119323	172.18.20.47	207.219.36.81	TCP	146	46804 → 443 [ACK]
12858	0.000259	8.125258	172.18.20.47	207.219.36.81	TLSv1.2	341	Client Hello

Figure 13: Sorted top summary panel for 172.18.20.47

c. Include a screenshot of one detailed packet from the sender IP that most helped you identify what that person was doing.

```

Server Name Indication extension
Server Name list length: 33
Server Name Type: host_name (0)
Server Name length: 30
Server Name: scontent.fykz2-1.fna.fbcdn.net

```

Figure 14: Packet 12858, Client Hello message sent to Facebook's fbcdn.net

```

Destination: 31.13.80.5
Transmission Control Protocol, Src Port: 39070, Dst Port: 443, Seq: 1, Ack: 772, Len: 35
Secure Sockets Layer
  TLSv1.2 Record Layer: Application Data Protocol: http-over-tls
    Content Type: Application Data (23)
    Version: TLS 1.2 (0x0303)
    Length: 30
    Encrypted Application Data: 4bcc892094f838ecee023f18be3faebdfc07a2718c8d8b24...

```

Figure 15: Packet 11709, sent to a Google IP - payload is TLS application data

#### 4. Which location was more secure?

We will define security here as higher levels packet protocols. In the public network recording, more than half of the packets have 802.11 protocol which is very low level. On the other hand, the home network recording all had higher level protocols: TCP, UDP, GQUIC, SSL or TLS.

No.	Delta	Time	Source	Destination	Protocol	Length	Info	UTC
59836	0.000266	34.140986	Cisco:c0:94:1f (2)	Apple:35:36:38	802.11	49	802.11 Block Ack Req, Flags:.....C	2018-09-17 23:16:02.709165
59837	0.000556	34.140926	Cisco:c0:94:1f (2)	Apple:35:36:38	802.11	49	802.11 Block Ack Req, Flags:.....R..C	2018-09-17 23:16:02.709721
59838	0.001112	34.142428	Cisco:c0:94:1f (2)	Realtek:80:14:51	802.11	288	Beacon Frame, Subtype: Probe, Flags:.....C, BI=102, SSID=MTLWdF	2018-09-17 23:16:02.729533
59839	0.001734	34.144162	Cisco:c0:94:1f (2)	Apple:36:79:05	802.11	39	Acknowledgement, Flags:.....C	2018-09-17 23:16:02.740267
59840	0.002064	34.152226	Cisco:c0:94:1f (2)	Apple:c4:76:0a	802.11	45	Request-to-send, Flags:.....C	2018-09-17 23:16:02.802121
59841	0.000052	34.151278	Cisco:c0:94:1f (2)	Apple:c4:76:0a	802.11	39	Null function (No data)[Malformed Packet]	2018-09-17 23:16:02.801383
59842	0.000049	34.151277	Cisco:c0:94:1f (2)	Apple:c4:76:0a	802.11	45	Request-to-send, Flags:.....C	2018-09-17 23:16:02.801482
59844	0.000048	34.151425	Cisco:c0:94:1f (2)	Apple:c4:76:0a	802.11	39	Probe Response[Malformed Packet]	2018-09-17 23:16:02.801530
59846	0.000012	34.152118	Cisco:c0:94:1f (2)	Apple:c4:76:0a	802.11	45	Request-to-send, Flags:.....C	2018-09-17 23:16:02.802274
59847	0.000028	34.152112	Cisco:c0:94:1f (2)	Apple:c4:76:0a	802.11	39	Clear-to-send, Flags:.....C	2018-09-17 23:16:02.802281
59848	0.000075	34.152457	Cisco:c0:94:1f (2)	Apple:c4:76:0a	802.11	49	Confirming Report Null, Flags:.....A.F.C	2018-09-17 23:16:02.802452
59849	0.000050	34.154747	Cisco:c0:94:1f (2)	Apple:c4:76:0a	802.11	39	Clear-to-send, Flags:.....C	2018-09-17 23:16:02.804852
59850	0.000052	34.154700	Cisco:c0:94:1f (2)	Apple:c4:76:0a	802.11	57	802.11 Block Ack, Flags:.....C	2018-09-17 23:16:02.804903
59853	0.000063	34.155391	Cisco:c0:94:1f (2)	Apple:28:92:0a	802.11	39	Clear-to-send, Flags:.....C	2018-09-17 23:16:02.805466
59854	0.000053	34.155354	Cisco:c0:94:1f (2)	Apple:28:92:0a	802.11	39	Clear-to-send, Flags:.....C	2018-09-17 23:16:02.805509
59855	0.000000	34.155434	Cisco:c0:94:1f (2)	Apple:28:92:0a	802.11	39	Clear-to-send, Flags:.....C	2018-09-17 23:16:02.806155
59856	0.000016	34.155608	Cisco:c0:94:1f (2)	Apple:28:92:0a	802.11	39	Clear-to-send, Flags:.....C	2018-09-17 23:16:02.806478
59857	0.000023	34.156373	Cisco:c0:94:1f (2)	Apple:28:92:0a	802.11	39	Clear-to-send, Flags:.....C	2018-09-17 23:16:02.806702
59858	0.000224	34.156597	Cisco:c0:94:1f (2)	Apple:28:92:0a	802.11	39	Clear-to-send, Flags:.....C	2018-09-17 23:16:02.806774
59859	0.000072	34.156609	Cisco:c0:94:1f (2)	Apple:28:92:0a	802.11	57	802.11 Block Ack, Flags:.....C	2018-09-17 23:16:02.806852
59860	0.000001	34.156706	Cisco:c0:94:1f (2)	Apple:28:92:0a	802.11	39	Acknowledgement, Flags:.....C	2018-09-17 23:16:02.806905
59862	0.000051	34.156768	Cisco:c0:94:1f (2)	Apple:c4:76:0a	802.11	39	Clear-to-send, Flags:.....C	2018-09-17 23:16:02.806973
59863	0.000050	34.158188	Cisco:c0:94:1f (2)	Apple:c4:76:0a	802.11	57	802.11 Block Ack, Flags:.....C	2018-09-17 23:16:02.808893
59864	0.000205	34.164923	Cisco:c0:94:1f (2)	Apple:58:14:2e	802.11	39	Acknowledgement, Flags:.....C	2018-09-17 23:16:02.814128
59865	0.000218	34.165041	Cisco:c0:94:1f (2)	Apple:58:14:2e	802.11	269	Probe Response, Subtype: Probe, Flags:.....R..C, BI=102, SSID=MTLWdF	2018-09-17 23:16:02.817146
59866	0.000248	34.167187	Cisco:c0:94:1f (2)	Apple:56:79:05	802.11	269	Probe Response, Subtype: Probe, Flags:.....R..C, BI=102, SSID=MTLWdF	2018-09-17 23:16:02.817292
59867	0.000471	34.173958	Cisco:c0:94:1f (2)	Apple:08:3c:59	802.11	45	Request-to-send, Flags:.....C	2018-09-17 23:16:02.823763
59868	0.000073	34.173729	Cisco:c0:94:1f (2)	Apple:08:3c:59	802.11	45	Request-to-send, Flags:.....C	2018-09-17 23:16:02.823854
59871	0.000486	34.175303	Cisco:c0:94:1f (2)	Apple:08:3c:59	802.11	45	Request-to-send, Flags:.....C	2018-09-17 23:16:02.825988
59877	0.000263	34.178028	Cisco:c0:94:1f (2)	Apple:08:3c:59	802.11	45	Request-to-send, Flags:.....C	2018-09-17 23:16:02.832832
59879	0.000096	34.182744	Cisco:c0:94:1f (2)	Apple:a5:a5:45	802.11	39	Acknowledgement, Flags:.....C	2018-09-17 23:16:02.832849
59880	0.000157	34.182851	Cisco:c0:94:1f (2)	Apple:08:3c:59	802.11	45	Request-to-send, Flags:.....C	2018-09-17 23:16:02.832966
59882	0.000090	34.187769	Cisco:c0:94:1f (2)	Apple:08:3c:59	802.11	45	Request-to-send, Flags:.....C	2018-09-17 23:16:02.837874
59884	0.000040	34.189770	Cisco:c0:94:1f (2)	Apple:08:3c:59	802.11	45	Request-to-send, Flags:.....C	2018-09-17 23:16:02.839075
59888	0.000092	34.194529	Cisco:c0:94:1f (2)	Apple:08:3c:59	802.11	39	Clear-to-send, Flags:.....C	2018-09-17 23:16:02.844734
59889	0.000032	34.194726	Cisco:c0:94:1f (2)	Apple:08:3c:59	802.11	57	802.11 Block Ack, Flags:.....C	2018-09-17 23:16:02.844825
59890	0.000078	34.194808	Cisco:c0:94:1f (2)	Apple:08:3c:59	802.11	45	Request-to-send, Flags:.....C	2018-09-17 23:16:02.844913
59892	0.000094	34.199883	Cisco:c0:94:1f (2)	Apple:08:3c:59	802.11	39	Clear-to-send, Flags:.....C	2018-09-17 23:16:02.849989
59894	0.000098	34.204082	Cisco:c0:94:1f (2)	Apple:08:3c:59	802.11	39	Clear-to-send, Flags:.....C	2018-09-17 23:16:02.854737
59895	0.000121	34.204893	Cisco:c0:94:1f (2)	Apple:08:3c:59	802.11	57	802.11 Block Ack, Flags:.....C	2018-09-17 23:16:02.854988
59896	0.000099	34.205082	Cisco:c0:94:1f (2)	Apple:08:3c:59	802.11	45	Request-to-send, Flags:.....C	2018-09-17 23:16:02.856087
59898	0.000078	34.205529	Cisco:c0:94:1f (2)	Apple:08:3c:59	802.11	45	Request-to-send, Flags:.....C	2018-09-17 23:16:02.856824
59899	0.000069	34.207591	Cisco:c0:94:1f (2)	Apple:08:3c:59	802.11	45	Request-to-send, Flags:.....C	2018-09-17 23:16:02.857606
59902	0.000082	34.209721	Cisco:c0:94:1f (2)	Apple:58:14:2e	802.11	39	Acknowledgement, Flags:.....C	2018-09-17 23:16:02.859826
59903	0.000040	34.209770	Cisco:c0:94:1f (2)	Apple:58:14:2e	802.11	39	Acknowledgement, Flags:.....C	2018-09-17 23:16:02.859875
59904	0.000083	34.209853	Cisco:c0:94:1f (2)	Apple:58:14:2e	802.11	39	Acknowledgement, Flags:.....C	2018-09-17 23:16:02.859928
59905	0.000056	34.210499	Cisco:c0:94:1f (2)	Apple:58:14:2e	802.11	39	Acknowledgement, Flags:.....C	2018-09-17 23:16:02.860014
59906	0.000105	34.210514	Cisco:c0:94:1f (2)	Apple:08:3c:59	802.11	45	Request-to-send, Flags:.....C	2018-09-17 23:16:02.862019
59909	0.000084	34.212193	Cisco:c0:94:1f (2)	Apple:08:3c:59	802.11	39	Clear-to-send, Flags:.....C	2018-09-17 23:16:02.862258
59910	0.000096	34.212253	Cisco:c0:94:1f (2)	Apple:08:3c:59	802.11	57	802.11 Block Ack, Flags:.....C	2018-09-17 23:16:02.862258
59911	0.000180	34.212253	Cisco:c0:94:1f (2)	Apple:08:3c:59	802.11	45	Request-to-send, Flags:.....C	2018-09-17 23:16:02.862258
59917	0.000072	34.214984	Cisco:c0:94:1f (2)	Apple:08:3c:59	802.11	45	Request-to-send, Flags:.....C	2018-09-17 23:16:02.865059
59918	0.000073	34.215987	Cisco:c0:94:1f (2)	Apple:08:3c:59	802.11	45	Request-to-send, Flags:.....C	2018-09-17 23:16:02.865172
180	0.000087	0.000212	Vmware:90:2e:f0	Tp-Link:00:10:10	ADP	140	who has 172.19.0.10? Tell 172.19.0.10	2018-09-17 23:15:28.160217
189	0.000069	0.000281	Vmware:90:2e:f0	Tp-Link:00:10:10	ADP	140	who has 172.19.0.55? Tell 172.19.0.10	2018-09-17 23:15:28.160386
119	0.000047	0.000529	Vmware:a3:3a:15	Tp-Link:00:10:10	ADP	140	who has 172.19.0.55? Tell 172.19.0.10	2018-09-17 23:15:28.160738
200	0.000060	0.000922	Vmware:a3:3a:15	Tp-Link:00:10:10	ADP	140	172.19.0.10 is at 00:10:10:a3:3a:15	2018-09-17 23:15:28.164667
203	0.000069	0.000373	Vmware:a3:3a:15	Tp-Link:00:10:10	ADP	140	172.19.0.10 is at 00:10:10:a3:3a:15	2018-09-17 23:15:28.164478
211	0.000043	0.000924	Vmware:a3:3a:15	Tp-Link:00:10:10	ADP	140	172.19.0.10 is at 00:10:10:a3:3a:15	2018-09-17 23:15:28.160159

Figure 16: Public network recording - all white packets are in 802.11.

The white packets go back to the beginning of the list.

We can then conclude that the home network was secure.