

Comp416 Project 2 Report
Burak Yıldırım 72849

Be aware that

```
private final String SERVER_KEYSTORE_FILE = "/Users/burak/Downloads/PC_4/tls-server/keystore.jks";
private final String KEY_STORE_NAME = "/Users/burak/Downloads/PC_4/tls-client/clientkeystore";
```

You should specify the path for keystore with your own path before running the codes!

1)

There is only 1 query for a single nslookup command as shown below. I did many tests but only one single query shows up in each trial. Total 2 packets by one of them is a query and other one is a query response.

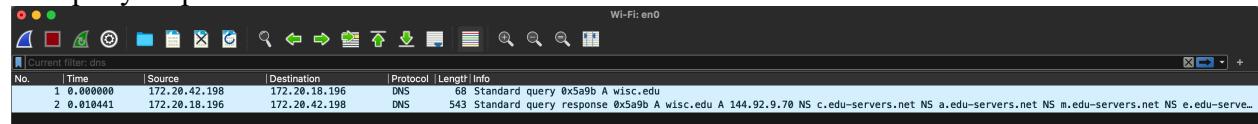


Figure 1



Figure 2

2)

Stream index corresponds for source address, source port, destination address, and destination port. Stream index remains same for the packets that are sent in the same stream. Using stream index we can distinguish a certain stream.

3)

The IP address being resolved is 144.92.9.70

Yes, we get the same result from the nslookup by using IP address and wisc.edu. Also I noticed that we can get to the wisc.edu by searching its address on a search engine.

```

burak@BURAK-MacBook-Pro ~ % nslookup 144.92.9.70
Server:      172.20.18.196
Address:     172.20.18.196#53

Non-authoritative answer:
70.9.92.144.in-addr.arpa      name = wisc.edu.

Authoritative answers can be found from:
in-addr.arpa    nameserver = c.in-addr-servers.arpa.
in-addr.arpa    nameserver = b.in-addr-servers.arpa.
in-addr.arpa    nameserver = a.in-addr-servers.arpa.
in-addr.arpa    nameserver = f.in-addr-servers.arpa.
in-addr.arpa    nameserver = d.in-addr-servers.arpa.
in-addr.arpa    nameserver = e.in-addr-servers.arpa.
a.in-addr-servers.arpa  internet address = 199.180.182.53
a.in-addr-servers.arpa  has AAAA address 2620:37:000::53
b.in-addr-servers.arpa  internet address = 199.253.183.183
b.in-addr-servers.arpa  has AAAA address 2001:500:87::87
c.in-addr-servers.arpa  internet address = 196.216.169.10
c.in-addr-servers.arpa  has AAAA address 2001:43f8:110::10
d.in-addr-servers.arpa  internet address = 200.10.60.53
d.in-addr-servers.arpa  has AAAA address 2001:13c7:7010::53
e.in-addr-servers.arpa  internet address = 203.119.86.101
e.in-addr-servers.arpa  has AAAA address 2001:dd8:6::101
f.in-addr-servers.arpa  internet address = 193.0.9.1
f.in-addr-servers.arpa  has AAAA address 2001:67c:e0::1

```

Figure 3

4)

The purpose of authoritative flag is to check whether the server is the authority for domain. In other words, the server can set the final IP address and it has full authority to adjust the DNS record.

The purpose of recursion desired flag is to determine whether the query should be done with recursion or not. The purpose of recursion available flag is to check whether server can do recursive queries or not.

These flags are used to execute the DNS queries appropriately and resolve domain names to IP addresses.

5)

The types of DNS records are A, NS, MX, and, CNAME.

```

> Flags: 0x8180 Standard query response, No error
Questions: 1
Answer RRs: 1
Authority RRs: 13
Additional RRs: 11
< Queries
> wisc.edu: type A, class IN
< Answers
> wisc.edu: type A, class IN, addr 144.92.9.70
< Authoritative nameservers
> edu: type NS, class IN, ns c.edu-servers.net
> edu: type NS, class IN, ns l.edu-servers.net
> edu: type NS, class IN, ns f.edu-servers.net
> edu: type NS, class IN, ns e.edu-servers.net
> edu: type NS, class IN, ns g.edu-servers.net
> edu: type NS, class IN, ns d.edu-servers.net
> edu: type NS, class IN, ns a.edu-servers.net
> edu: type NS, class IN, ns m.edu-servers.net
> edu: type NS, class IN, ns i.edu-servers.net
> edu: type NS, class IN, ns k.edu-servers.net
> edu: type NS, class IN, ns h.edu-servers.net
> edu: type NS, class IN, ns b.edu-servers.net
> edu: type NS, class IN, ns j.edu-servers.net
< Additional records
> a.edu-servers.net: type A, class IN, addr 192.5.6.30
> a.edu-servers.net: type AAAA, class IN, addr 2001:503:
> b.edu-servers.net: type A, class IN, addr 192.33.14.30
> b.edu-servers.net: type AAAA, class IN, addr 2001:503:
> c.edu-servers.net: type A, class IN, addr 192.26.92.30
> c.edu-servers.net: type AAAA, class IN, addr 2001:503:
> d.edu-servers.net: type A, class IN, addr 192.31.80.30
> d.edu-servers.net: type AAAA, class IN, addr 2001:500:
> e.edu-servers.net: type A, class IN, addr 192.12.94.30
> e.edu-servers.net: type AAAA, class IN, addr 2001:502:
> f.edu-servers.net: type A, class IN, addr 192.35.51.30

```

Figure 4

In nslookup, the type can be specified as follows:

nslookup -type=<DNS_RECORD_TYPE> wisc.edu

```
burak@BURAK-MacBook-Pro ~ % nslookup -type=NS wisc.edu
Server:      172.20.18.196
Address:     172.20.18.196#53

Non-authoritative answer:
wisc.edu      nameserver = adns2.doit.wisc.edu.
wisc.edu      nameserver = adns3.doit.wisc.edu.
wisc.edu      nameserver = adns1.doit.wisc.edu.
wisc.edu      nameserver = adns4.doit.wisc.edu.

Authoritative answers can be found from:
adns1.doit.wisc.edu    internet address = 144.92.9.21
adns1.doit.wisc.edu    has AAAA address 2607:f388::a53:1
adns2.doit.wisc.edu    internet address = 144.92.20.99
adns2.doit.wisc.edu    has AAAA address 2607:f388::a53:2
adns3.doit.wisc.edu    internet address = 144.92.104.21
adns3.doit.wisc.edu    has AAAA address 2607:f388::a53:3
adns4.doit.wisc.edu    internet address = 128.6.1.132
adns4.doit.wisc.edu    has AAAA address 2620:0:d60:6::12
```

Figure 5

```
burak@BURAK-MacBook-Pro ~ % nslookup -type=CNAME wisc.edu
Server:      172.20.18.196
Address:     172.20.18.196#53

Non-authoritative answer:
*** Can't find wisc.edu: No answer

Authoritative answers can be found from:
wisc.edu
      origin = ipam-cssc.doit.wisc.edu
      mail addr = hostmaster.doit.wisc.edu
      serial = 2018647395
      refresh = 7200
      retry = 720
      expire = 3628800
      minimum = 600
```

Figure 6

```
burak@BURAK-MacBook-Pro ~ % nslookup -type=MX wisc.edu
Server:      172.20.18.196
Address:     172.20.18.196#53

Non-authoritative answer:
wisc.edu      mail exchanger = 10 mta4.wisemail.wisc.edu.
wisc.edu      mail exchanger = 10 mta5.wisemail.wisc.edu.
wisc.edu      mail exchanger = 5 smtp.wisemail.wisc.edu.
wisc.edu      mail exchanger = 10 mta2.wisemail.wisc.edu.
wisc.edu      mail exchanger = 10 mta1.wisemail.wisc.edu.
wisc.edu      mail exchanger = 10 mta3.wisemail.wisc.edu.

Authoritative answers can be found from:
wisc.edu      nameserver = adns4.doit.wisc.edu.
wisc.edu      nameserver = adns2.doit.wisc.edu.
wisc.edu      nameserver = adns3.doit.wisc.edu.
wisc.edu      nameserver = adns1.doit.wisc.edu.
adns1.doit.wisc.edu    internet address = 144.92.9.21
adns1.doit.wisc.edu    has AAAA address 2607:f388::a53:1
adns2.doit.wisc.edu    internet address = 144.92.20.99
adns2.doit.wisc.edu    has AAAA address 2607:f388::a53:2
adns3.doit.wisc.edu    internet address = 144.92.104.21
adns3.doit.wisc.edu    has AAAA address 2607:f388::a53:3
adns4.doit.wisc.edu    internet address = 128.6.1.132
adns4.doit.wisc.edu    has AAAA address 2620:0:d60:6::12
```

Figure 7

6)

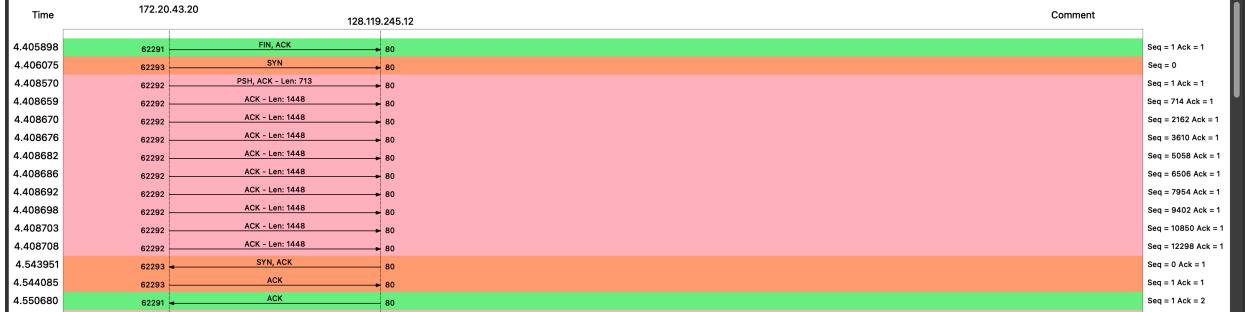


Figure 8

Green ones are for the termination. Red ones are for the handshake. Pink ones for the data exchange. PSH,ACK is getting data from previous packets and transmits more packets.

For the details of a packet that is used for exchanging information are given below.

Figure 9

Start-End Time	Source Socket	Destination Socket	Sequence Number	Number of Packets in the stage	Stream ID
4.840345-4.481421	172.20.43.20:62292	128.119.245.12:80	511417948	107	2

7)

Stream index displays unique number for each stream such as one for the first stream and two for the second stream, etc. 3-way handshake is the start of a stream as we seen in the class. Then, after the 3-way handshake successfully occurs data transmit part starts. In the end, data exchange process terminates with the following procedure: FIN from client, ACK from server, client waiting, FIN from server, and ACK from client.

No, the packets being transmitted during the experiment don't belong to the same stream index.

Stream index 1 is used for the handshake. Stream index 0 is used for the termination process.

Stream index 2 is used for the data exchange process

8)

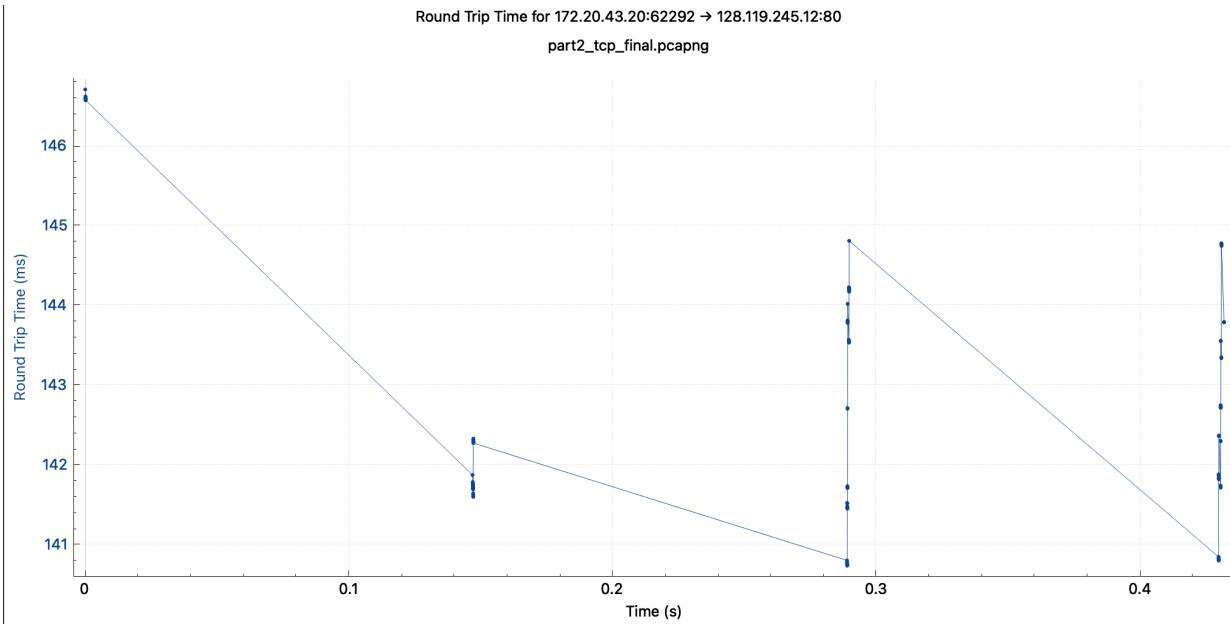


Figure 10

RTT graph is shown above.

```
burak@BURAK-MacBook-Pro ~ Downloads % tshark -Y "tcp" -r part2_tcp_final.pcapng -Tfields -e "tcp.analysis.ack_rtt"
0.137874000
0.080340000
0.144782000
0.146718000
0.133140000
0.146574000

0.141744000
0.141642000
0.141600000
```

Figure 11

In the terminal, I executed the following command to get the RTT values for the entire communication. `tshark -Y "tcp" -r part2_tcp_final.pcapng -Tfields -e "tcp.analysis.ack_rtt"`. Then, I imported these values into an excel sheet and calculated the average by using AVERAGE function of Excel.

0.137876	Average
0.000134	0.1377239474
0.144782	
0.14671	
0.146599	
0.146574	
0.141744	
0.141642	
0.1416	
0.142283	
0.142275	
0.140759	
0.140737	
0.141523	
0.141479	
0.141468	
0.14146	
0.141452	
0.141733	
0.141718	
0.14171	
0.142713	
0.142706	
0.143804	
0.143796	
0.143789	
0.143781	
0.144021	
0.143541	
0.143534	
0.144206	

As it can be seen in the excel spreadsheet, the average of RTT values is 0.1377239474.

Figure 12

9)

No.	Time	Source	Destination	Protocol	Length	Info
219	4.988216	128.119.245.12	172.28.43.20	TCP	66 80 + 62292 [ACK] Seq=1 Ack=115186 Win=1432 Len=0 Tsv=271996899 Tsec=r=1188891424	
228	4.988216	128.119.245.12	172.28.43.20	TCP	66 80 + 62292 [ACK] Seq=1 Ack=116554 Win=1426 Len=0 Tsv=271996899 Tsec=r=1188891424	
221	4.988770	128.119.245.12	172.28.43.20	TCP	66 80 + 62292 [ACK] Seq=1 Ack=120898 Win=1432 Len=0 Tsv=271996899 Tsec=r=1188891424	
222	4.988771	128.119.245.12	172.28.43.20	TCP	66 80 + 62292 [ACK] Seq=1 Ack=122346 Win=1454 Len=0 Tsv=271996899 Tsec=r=1188891424	
223	4.988771	128.119.245.12	172.28.43.20	TCP	66 80 + 62292 [ACK] Seq=1 Ack=123794 Win=1477 Len=0 Tsv=271996899 Tsec=r=1188891424	
224	4.981364	128.119.245.12	172.28.43.20	TCP	66 80 + 62292 [ACK] Seq=1 Ack=126698 Win=1522 Len=0 Tsv=271996899 Tsec=r=1188891424	
225	4.981821	128.119.245.12	172.28.43.20	TCP	66 80 + 62292 [ACK] Seq=1 Ack=133938 Win=1635 Len=0 Tsv=271996899 Tsec=r=1188891424	
226	4.982672	128.119.245.12	172.28.43.20	TCP	66 80 + 62292 [ACK] Seq=1 Ack=136026 Win=1635 Len=0 Tsv=271996899 Tsec=r=1188891424	
227	4.982673	128.119.245.12	172.28.43.20	TCP	66 80 + 62292 [ACK] Seq=1 Ack=136183 Win=1635 Len=0 Tsv=271996899 Tsec=r=1188891424	
228	4.982673	128.119.245.12	172.28.43.20	TCP	66 80 + 62292 [ACK] Seq=1 Ack=139723 Win=1700 Len=0 Tsv=271996899 Tsec=r=1188891425	
229	4.984133	128.119.245.12	172.28.43.20	TCP	66 80 + 62292 [ACK] Seq=1 Ack=146063 Win=1859 Len=0 Tsv=271996899 Tsec=r=1188891425	
230	4.984134	128.119.245.12	172.28.43.20	TCP	66 80 + 62292 [ACK] Seq=1 Ack=149419 Win=1884 Len=0 Tsv=271996899 Tsec=r=1188891425	
231	4.984134	128.119.245.12	172.28.43.20	TCP	66 80 + 62292 [ACK] Seq=1 Ack=149859 Win=1884 Len=0 Tsv=271996899 Tsec=r=1188891426	
232	4.984134	128.119.245.12	172.28.43.20	TCP	66 80 + 62292 [ACK] Seq=1 Ack=151396 Win=1987 Len=0 Tsv=271996899 Tsec=r=1188891426	
233	4.984135	128.119.245.12	172.28.43.20	TCP	66 80 + 62292 [ACK] Seq=1 Ack=153835 Win=1934 Len=0 Tsv=271996899 Tsec=r=1188891426	
234	4.984135	128.119.245.12	172.28.43.20	HTTP	843 HTTP/1.1 200 OK (text/html)	

Figure 13

If we observe the selected TCP packet, its properties are as follows:

TCP Segment Length: 0

Sequence Number (raw): 1876458969

Relative Sequence Number: 1

Acknowledgment Number: 511415052

Relative Acknowledgment Number: 149858

Nonce Flag:

Congestion Window Reduce (CWR) Flag: 0

10)

No.	Time	Source	Destination	Protocol	Length	Info
41	2.541068	127.0.0.1	127.0.0.1	TLSv1...	520	Client Hello
43	2.562956	127.0.0.1	127.0.0.1	TLSv1...	183	Server Hello
45	2.568672	127.0.0.1	127.0.0.1	TLSv1...	62	Change Cipher Spec
47	2.570857	127.0.0.1	127.0.0.1	TLSv1...	126	Application Data
49	2.571620	127.0.0.1	127.0.0.1	TLSv1...	62	Change Cipher Spec
51	2.572819	127.0.0.1	127.0.0.1	TLSv1...	980	Application Data
53	2.595178	127.0.0.1	127.0.0.1	TLSv1...	358	Application Data
55	2.596227	127.0.0.1	127.0.0.1	TLSv1...	146	Application Data
57	2.602213	127.0.0.1	127.0.0.1	TLSv1...	146	Application Data
59	2.606461	127.0.0.1	127.0.0.1	TLSv1...	1244	Application Data
61	3.543776	127.0.0.1	127.0.0.1	TLSv1...	96	Application Data
63	3.544533	127.0.0.1	127.0.0.1	TLSv1...	95	Application Data
65	3.545867	127.0.0.1	127.0.0.1	TLSv1...	96	Application Data
Session ID: 802584db7b1c1f62906769fc92f70e6af765ccbc1111dd8289ff07e93cb45ae8						
Cipher Suites Length: 98						
▼ Cipher Suites (49 suites)						
Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)						
Cipher Suite: TLS_AES_128_GCM_SHA256 (0x1301)						
Cipher Suite: TLS_CHACHA20_POLY1305_SHA256 (0x1303)						
Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c)						
Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b)						
Cipher Suite: TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 (0xcc9a)						
Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)						
Cipher Suite: TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xccaa8)						
Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)						
Cipher Suite: TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (0x009f)						
Cipher Suite: TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xccaa)						
Cipher Suite: TLS_DHE_DSS_WITH_AES_256_GCM_SHA384 (0x00a3)						
Cipher Suite: TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 (0x009e)						
Cipher Suite: TLS_DHE_DSS_WITH_AES_128_GCM_SHA256 (0x00a2)						
Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 (0xc024)						
Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xc028)						
Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 (0xc023)						
Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xc027)						
Cipher Suite: TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 (0x006b)						
Cipher Suite: TLS_DHE_DSS_WITH_AES_256_CBC_SHA256 (0x006a)						

Figure 14

Client sends 49 Cipher Suites shown above.

No.	Time	Source	Destination	Protocol	Length	Info
41	2.541068	127.0.0.1	127.0.0.1	TLSv1...	520	Client Hello
43	2.562956	127.0.0.1	127.0.0.1	TLSv1...	183	Server Hello
45	2.568672	127.0.0.1	127.0.0.1	TLSv1...	62	Change Cipher Spec
47	2.570857	127.0.0.1	127.0.0.1	TLSv1...	126	Application Data
49	2.571620	127.0.0.1	127.0.0.1	TLSv1...	62	Change Cipher Spec
51	2.572819	127.0.0.1	127.0.0.1	TLSv1...	980	Application Data
53	2.595178	127.0.0.1	127.0.0.1	TLSv1...	358	Application Data
55	2.596227	127.0.0.1	127.0.0.1	TLSv1...	146	Application Data
57	2.602213	127.0.0.1	127.0.0.1	TLSv1...	146	Application Data
59	2.606461	127.0.0.1	127.0.0.1	TLSv1...	1244	Application Data
61	3.543776	127.0.0.1	127.0.0.1	TLSv1...	96	Application Data
63	3.544533	127.0.0.1	127.0.0.1	TLSv1...	95	Application Data
65	3.545867	127.0.0.1	127.0.0.1	TLSv1...	96	Application Data
> Frame 43: 183 bytes on wire (1464 bits), 183 bytes captured (1464 bits) on interface lo0,						
> Null/Loopback						
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1						
> Transmission Control Protocol, Src Port: 4444, Dst Port: 54764, Seq: 1, Ack: 465, Len: 127						
▼ Transport Layer Security						
▼ TLSv1.3 Record Layer: Handshake Protocol: Server Hello						
Content Type: Handshake (22)						
Version: TLS 1.2 (0x0303)						
Length: 122						
▼ Handshake Protocol: Server Hello						
Handshake Type: Server Hello (2)						
Length: 118						
Version: TLS 1.2 (0x0303)						
Random: dfb077bd365e0718af1c53f0468b7b36b74ad256b6af8af6977edb41517dfb40						
Session ID Length: 32						
Session ID: 802584db7b1c1f62906769fc92f70e6af765ccbc1111dd8289ff07e93cb45ae8						
Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)						

Figure 15

Server choose Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)

11)

It uses TLS version 1.3. I had done many research and I saw that it doesn't show the certificates in Wireshark. Hence, I couldn't find the certificates.

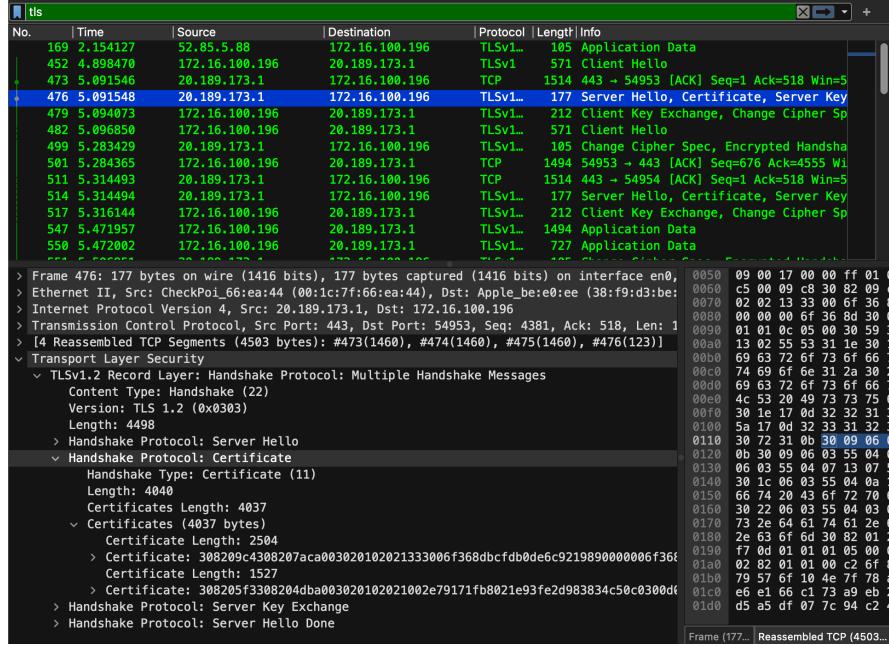


Figure 16

When I capture packet using WIFI instead of loopback. I got the certificates. There are 2 certificates sent by the server as seen in Figure 16. It's due to the its using 2-way SSL handshake.

12)

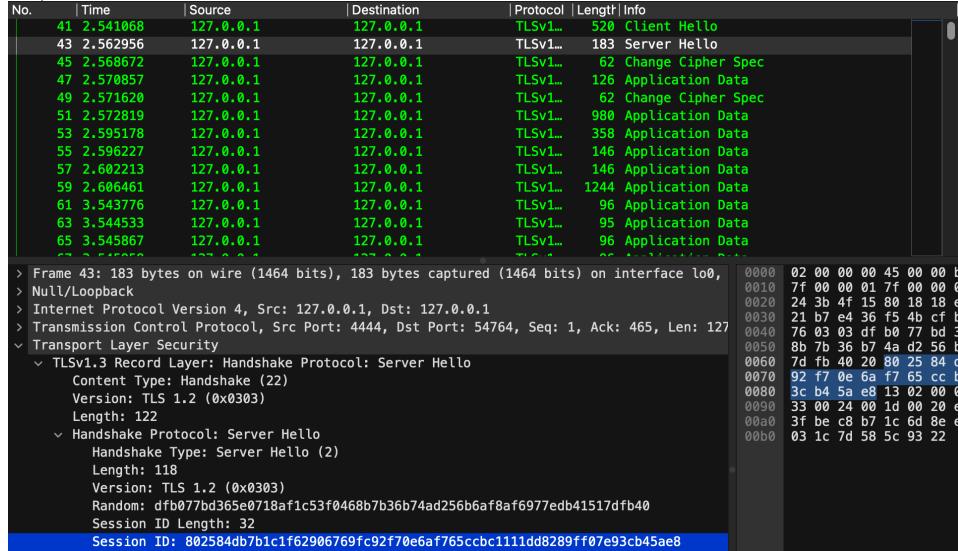


Figure 17

Session ID: 802584db7b1c1f62906769fc92f70e6af765ccbc1111dd8289ff07e93cb45ae8

The purpose of specifying a session ID is used to distinguish a certain session among SSL sessions. It's used to encrypt the communication between the client and server to have a secure connection. Using session ID server can recognize the client and the data is sent to the correct client.

13)

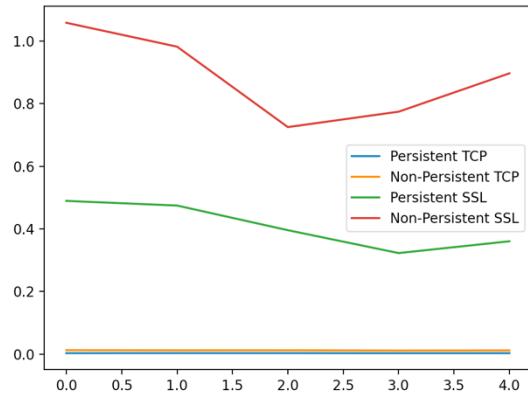


Figure 18

From doing 5 experiments for each protocol and persistency option, I got a graph as shown. Persistent communications are faster than non-persistent ones because one handshake is sufficient to exchange the whole data whereas in non-consistent communication handshake should be repeated for each data. TCP communication is faster than SSL communication because SSL communication uses encryption whereas TCP directly communicates with the client by IP and port. In the graph, x values are experiment numbers and y values are measured delays.

Persistent TCP = [0.002920,0.003130,0.003039,0.002833,0.002874]

Non-Persistent TCP = [0.012517,0.012036,0.012110,0.011287,0.011691]

Persistent SSL = [0.489279,0.474389,0.39568,0.322609,0.360253]

Non-Persistent SSL = [1.05819,0.981875,0.725004,0.774202,0.896675]

14)

Yes, they have stream index. Since the stream index is different for the handshake, data exchange, and termination processes. They contain stream index values to indicate those processes.