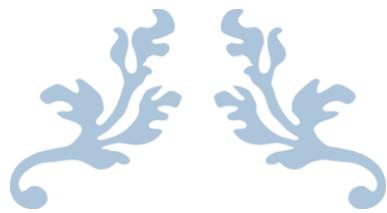


«باسم‌هه تعالی»



---

گزارش آزمایش هفته‌ی سوم  
آشنایی با وایرشارک

---

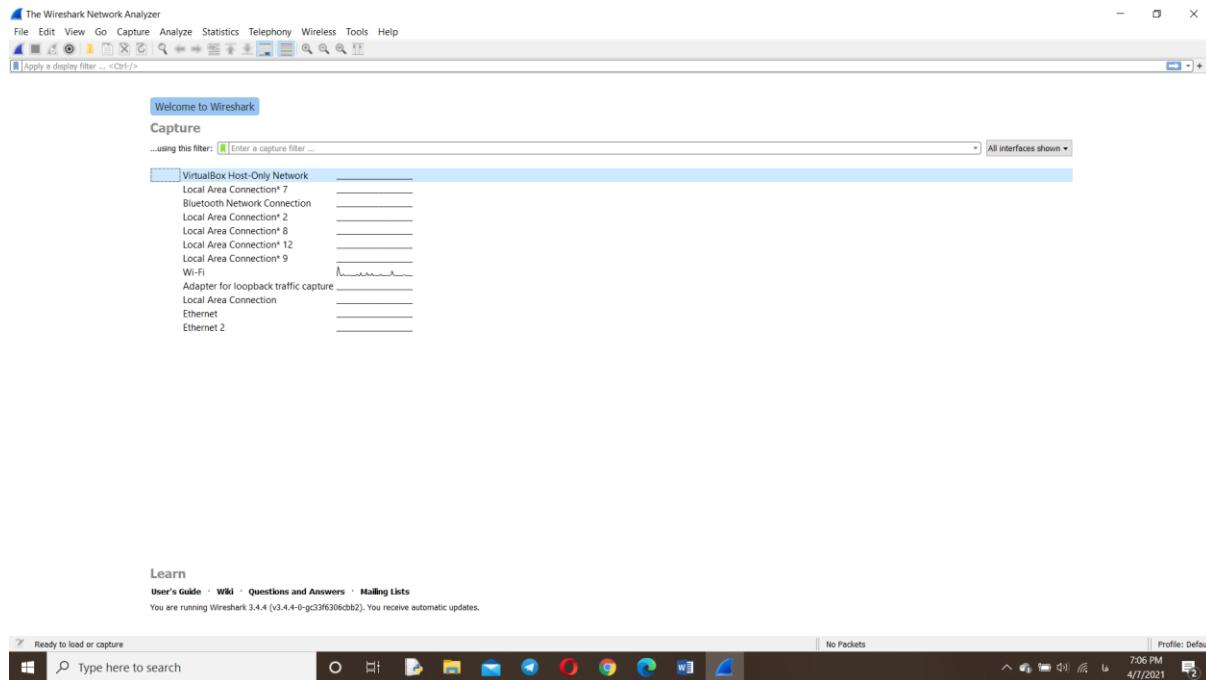


طراحی و تدوین:

مهدی رحمانی

9731701

واسطی که با آن دسترسی به اینترنت داریم را ابتدا انتخاب میکنیم که برای بنده اتصال از طریق Wi-Fi میباشد و در تمام بخش های آزمایش برای شنود بسته از آن استفاده میکنم.



## قسمت اول آزمایش

### «لایه بندی پروتکل ها»

ابتدا شروع به شنود بسته ها میکنیم. در حدود 3 دقیقه وب گردی میکنیم به سایت هایی مثل گوگل و یاهو و کوئری و سایت دانشگاه امیرکبیر و فرادرس مراجعه میکنیم و سپس آن را متوقف میکنیم و به سراغ سوالات میرویم:

**سوال 1:** به یک بخش دلخواه از بسته های شنود شده مراجعه کنید. چه پروتکل هایی را مشاهده می کنید. لیست آن ها را یادداشت کنید.

No.	Time	Source	Destination	Protocol	Length	Info
488 11.310580	119.161.43.175	119.161.43.175	TCP	66	[TCP Previous segment not captured] BB + 0x125 [ACK] Seq=1 Ack=1 Win=570 Len=0	
488 11.310580	192.168.43.175	119.161.43.175	TCP	54	61590 + 443 [ACK] Seq=2 Ack=2 Win=370 Len=0	
489 11.313081	216.58.208.226	192.168.43.175	QUIC	67	Protected Payload (K99)	
490 11.398340	119.161.43.176	192.168.43.175	TCP	54	61590 + 61590 [FIN, ACK] Seq=1 Ack=2 Win=370 Len=0	
491 11.398386	192.168.43.175	119.161.43.176	TCP	54	61590 + 443 [ACK] Seq=2 Ack=2 Win=511 Len=0	
492 11.427579	138.199.44.86	192.168.43.175	TCP	66	[TCP Previous segment not captured] BB + 0x125 [ACK] Seq=2 Ack=1 Win=501 Len=0 SLE=1	
493 11.715221	192.168.43.175	142.250.181.78	QUIC	1392	Initial, SCID=63835c3a536ee15, PNL: 1, CRYPTO, PADDING	
494 12.021480	142.250.181.78	192.168.43.175	QUIC	1392	Initial, SCID=63835c3a536ee15, PNL: 1, ACK, PADDING	
495 12.089593	142.250.181.78	192.168.43.175	QUIC	1392	Initial, SCID=63835c3a536ee15, PNL: 2, CRYPTO, PADDING	
496 12.089593	142.250.181.78	192.168.43.175	QUIC	276	Handshake, SCID=63835c3a536ee15	
497 12.089593	142.250.181.78	192.168.43.175	QUIC	100	Protected Payload (K99)	
498 12.086135	192.168.43.175	142.250.181.78	QUIC	187	Protected Payload (K99), DCID=63835c3a536ee15	
499 12.086638	192.168.43.175	142.250.181.78	QUIC	1388	Protected Payload (K99), DCID=63835c3a536ee15	
500 12.086713	192.168.43.175	142.250.181.78	QUIC	1388	Protected Payload (K99), DCID=63835c3a536ee15	
501 12.086755	192.168.43.175	142.250.181.78	QUIC	406	Protected Payload (K99), DCID=63835c3a536ee15	
502 12.127414	157.240.20.52	192.168.43.175	TLSv1.2	78	Application Data	
503 12.127570	157.240.20.52	192.168.43.175	TCP	54	443 + 61463 [RST, ACK] Seq=25 Ack=1 Win=412 Len=0	
504 12.146981	192.168.43.175	192.168.43.1	DNS	76	Standard query 0x3910 A web.whatsapp.com	
505 12.200999	192.168.43.175	192.168.43.1	DNS	76	Standard query 0x3910 A web.whatsapp.com	
506 12.217150	192.168.43.1	192.168.43.175	DNS	129	Standard query response 0x3910 A web.whatsapp.com CHNAME mnx-ds.cdn.whatsapp.net A 157.240.20.52	
507 12.231388	157.240.20.52	192.168.43.175	TCP	60	66.443 - 63573 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1	
508 12.331442	192.168.43.175	157.240.20.52	TCP	54	61573 + 443 [ACK] Seq=1 Ack=1 Win=32096 Len=0	

همانطور که در اسکرین شات مشخص است طبق خواسته‌ی سوال یک بخش دلخواه از این لیست بسته‌های شنود شده را درنظر میگیریم. حال به ستون پروتکل این لیست مراجعه میکنیم و لیست پروتکل هایی را که مشاهده میکنیم را مینویسیم که به شرح زیر است:

TCP و QUIC و TLSv1.2 و DNS

همچنین در طول این ۳ دقیقه بسته‌های زیادی شنود شده اند که اگه بخواهیم همه را درنظر بگیریم لیست پروتکل‌های مشاهده شده به صورت زیر میباشد:

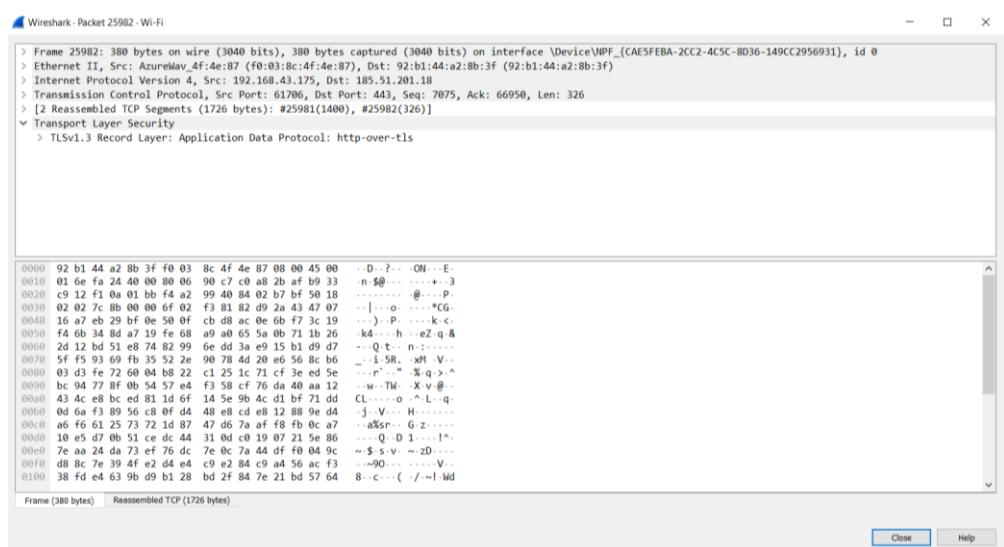
TLSv1.3 و TCP و DNS و NBNS و LLMNR و SSDP و ARP و QUIC و TLSv1.2 و DNS و MDNS و TCP

**سوال 2:** یک بسته را به دلخواه انتخاب کنید. مشخص کنید که چه پروتکل هایی در لایه های مختلف آن استفاده شده است. ترتیب قرارگیری بیت‌ها داخل بسته با لایه های مختلف دارد؟ اندازه فریم لایه دو این بسته چقدر است؟ اندازه بسته لایه ۳ چقدر است؟

ابتدا برای مثال یک بسته را انتخاب میکنیم:

25980 119.736527	192.168.43.175	104.16.87.20	QUIC	1392 Initial, DCID=638bc1894e46daf0, PKN: 1, CRYPTO, PADDING
25981 119.737466	192.168.43.175	185.51.201.18	TCP	1454 61706 → 443 [ACK] Seq=5675 Ack=66950 Win=131584 Len=1400 [TCP segment of a reassembled PDU]
25982 119.737466	192.168.43.175	185.51.201.18	[TLSv1.3]	388 Application Data
25983 119.782885	192.168.43.175	239.255.255.250	SSDP	216 M-SEARCH * HTTP/1.1
25984 119.782885	192.168.43.175	239.255.255.250	SSDP	215 M-SEARCH * HTTP/1.1

پس از اینکه روی آن کلیک کنیم اطلاعات مختلف آن نمایش داده میشود:



براساس مدل ۵ لایه‌ای TCP/IP میباشد ولی اکثر اوقات اطلاعات لایه‌ی physical را نمیفرستد.

در لایه‌ی application این بسته از پروتکل http استفاده شده.

در لایه‌ی Transport این بسته از پروتکل TCP استفاده شده.

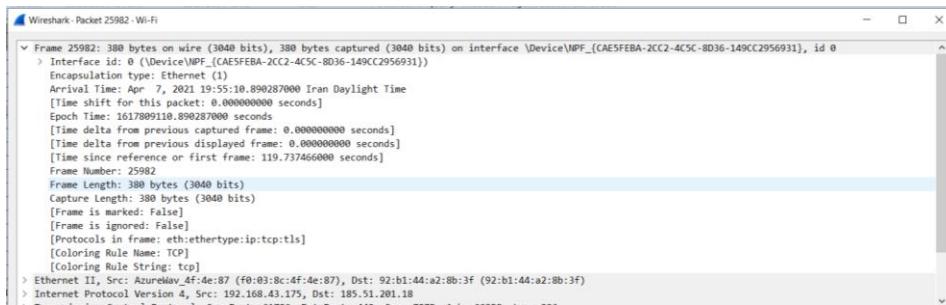
در لایه‌ی Network آن از پروتکل IP یا به طور دقیق‌تر IPv4 استفاده شده است.

در لایه‌ی Data Link این بسته نیز از پروتکل Ethernet II استفاده شده است.

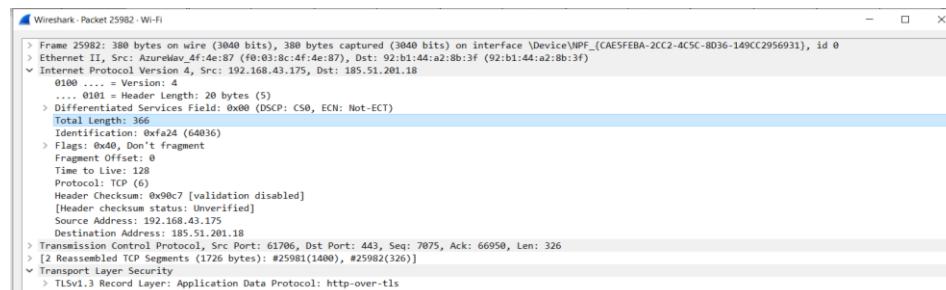
(اطلاعات مربوط به لایه‌ی فیزیکال نیست ولی فکر می‌کنم با توجه به اینترفیس ما، Wi-Fi می‌باشد)

ترتیب قرارگیری بیت‌ها اینگونه است که اگر اطلاعات لایه‌ی physical باشد که غالباً نیست، در صورت وجود در ابتدای بسته قرار می‌گیرد. سپس بیت‌های مربوط به لایه‌ی Data Link و بعد بیت‌های مربوط به لایه‌ی Network یا همان لایه‌ی اینترنت و سپس Application و در نهایت بیت‌های لایه‌ی Transport در بسته قرار می‌گیرند. این موضوع را می‌توان از روی این فهمید که هر وقت روى اطلاعات human readable مربوط به بسته کلیک می‌کنیم منتظر با آن در قسمت پایین که کدهای hex وجود دارند سلکت می‌شوند. می‌توان ارتباط آن‌ها را اینگونه درک کرد.

لایه‌ی دوم میدانیم که لایه‌ی Data Link می‌باشد. طبق سوالی که در تلگرام از استاد پرسیدم برای این قسمت منظور همان اندازه‌ی frame می‌باشد که در همان ابتدای اسکرین شات هم قابل مشاهده می‌باشد. بنابراین اندازه فریم لایه‌ی دوی این بسته طبق اسکرین شات زیر برابر با 380 بايت یا 3040 بیت می‌باشد.



لایه‌ی سوم Network یا همان لایه‌ی اینترنت می‌باشد. پس به این لایه مراجعه کرده و در قسمت Total Length می‌توان اندازه‌ی بسته‌ی این لایه را دید. اندازه‌ی بسته‌ی لایه‌ی سوم طبق اسکرین شات زیر برابر با 366 بايت می‌باشد.

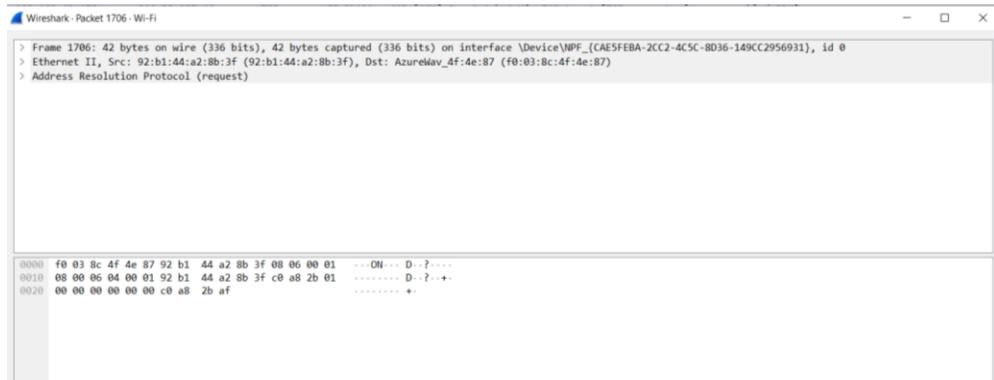


### سوال 3: آیا می توانید بسته هایی را پیدا کنید که بدون پروتکل های لایه های Transport ، Network و Application باشند؟ این بسته ها از چه پروتکلی استفاده کرده اند؟

بله - برای مثال بسته‌ی زیر را انتخاب کردم و لایه‌های گفته شده را نداشت:

1705 27.620962	192.168.43.175	185.211.88.131	TCP	66 61593 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
1706 27.621685	92:b1:44:a2:8b:3f	AzureWav_Af:4e:87	ARP	42 Who has 192.168.43.175? Tell 192.168.43.1
1707 27.621708	AzureWav_Af:4e:87	92:b1:44:a2:8b:3f	ARP	42 192.168.43.175 is at f0:03:8c:4f:4e:87
1708 27.645590	185.211.88.131	192.168.43.175	TCP	62 443 → 61593 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1400 SACK_PERM=1
1709 27.645595	192.168.43.175	185.211.88.131	TCP	54 61593 → 443 [ACK] Seq=1 Ack=1 Win=64240 Len=0
1710 27.645874	192.168.43.175	185.211.88.131	TLSv1.2	571 Client Hello

اگر روی آن کلیک کنیم میتوانیم ببینیم که پروتکل‌های لایه‌های Network و Transport و Application در این بسته نمیباشند:



همانطور هم که مشاهده میشود این بسته‌ها از پروتکل ARP استفاده میکنند. وظیفه‌ی آن تبدیل آدرس IP به آدرس فیزیکی یا همان MAC Address میباشد.

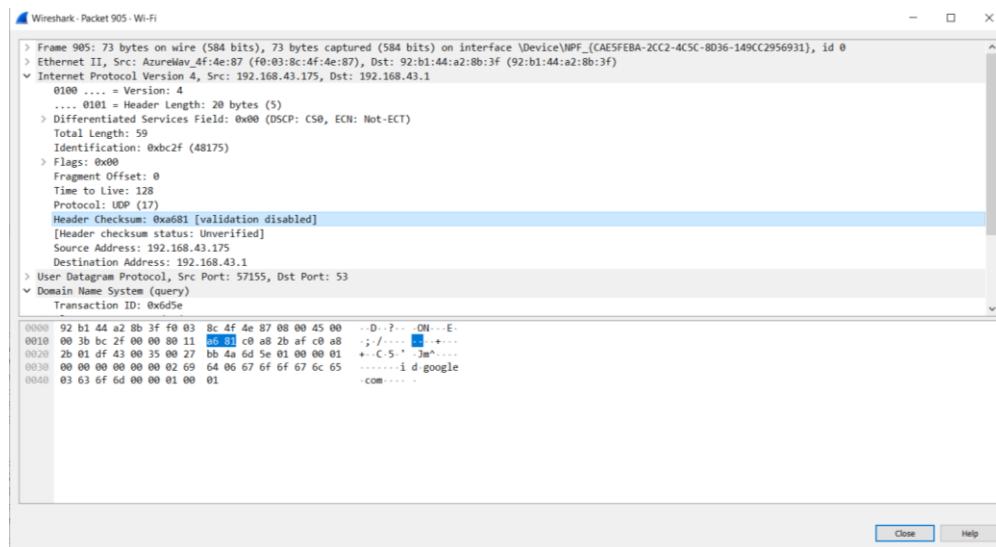
سوال 4: از یکی از بسته‌ها بخش مربوط به پروتکل Internet Protocol (IP) را پیدا کنید.

پروتکل IP را پیدا کنید و آن را یادداشت کنید.

ابتدا یکی از بسته‌ها را انتخاب می‌کنیم. برای مثال بسته‌ی زیر را انتخاب کردم:

903 16.710579	172.217.21.36	192.168.43.175	QUIC	312 Protected Payload (KPO)
904 16.710689	172.217.21.36	192.168.43.175	QUIC	214 Protected Payload (KPO)
905 16.711610	192.168.43.175	192.168.43.1	DNS	73 Standard query 0x6d5e A id.google.com
906 16.715075	192.168.43.175	172.217.21.36	QUIC	225 Protected Payload (KPO), DCID=7b5d415e20a3c37
907 16.743148	192.168.43.175	192.168.43.1	DNS	73 Standard query 0x6d5e A id.google.com

حال اگر روی آن کلیک کنیم و به بخش پروتکل IP برویم، طبق اسکرین شات زیر می‌توان checksum را مشاهده کرد که برابر است با: 0xa681



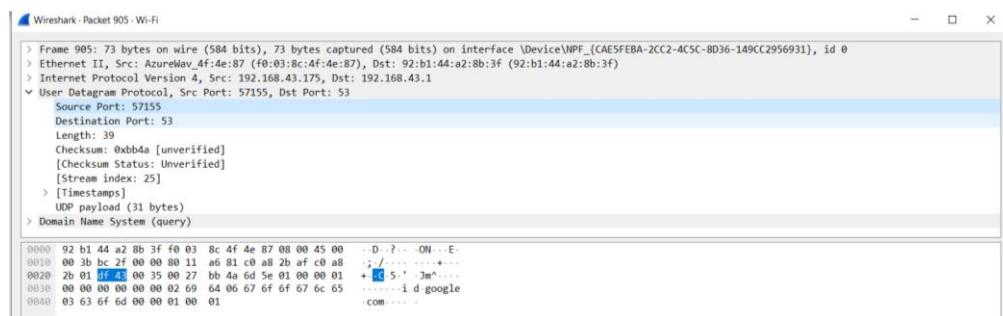
**سوال 5:** از یکی از بسته‌ها بخش مربوط به پروتکل **(TCP)** و یا **User Transport Control Protocol (TCP)** را پیدا کنید. عدد مربوط به پورت مبدأ و مقصد را یادداشت کنید. به نظر شما این اعداد در مبدأ و مقصد چه چیزی را مشخص می‌کند؟ checksum مربوط به پروتکل‌های **TCP** و **UDP** را مشخص کنید.

برای این قسمت هم بسته‌ای پیدا می‌کنیم که از TCP استفاده کرده باشے و هم یک بسته که از UDP استفاده کرده باشد.

برای مثال بسته‌ی زیر را انتخاب می‌کنیم که دارای بخش UDP می‌باشد:

903 16.710579	172.217.21.36	192.168.43.175	QUIC	312 Protected Payload (KPO)
904 16.710689	172.217.21.36	192.168.43.175	QUIC	214 Protected Payload (KPO)
905 16.711610	192.168.43.175	192.168.43.1	DNS	73 Standard query 0x6d5e A id.google.com
906 16.715075	192.168.43.175	172.217.21.36	QUIC	225 Protected Payload (KPO) A DCID=7b5d415e20a3c37
907 16.743148	192.168.43.175	192.168.43.1	DNS	73 Standard query 0x6d5e A id.google.com

حال به بخش User Datagram Protocol می‌رویم و عدد مربوط به پورت مبدأ و مقصد را می‌ایم:



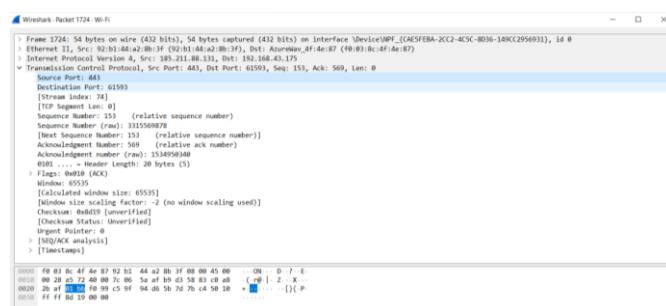
شماره پورت مبدأ: 57155      شماره پورت مقصد: 53

همچنین checksum مربوط به آن هم در اسکرین شات مشخص است که برابر با: 0xbbb4a می‌باشد.

به عنوان مثال دیگر بسته‌ی زیر را انتخاب می‌کنیم که دارای بخش TCP می‌باشد:

1721 27.701381	192.168.43.175	185.211.88.131	TLSv1.2	105 Change Cipher Spec, Encrypted Handshake Message
1722 27.716313	40.119.211.203	192.168.43.175	TLSv1.2	225 Application Data
1723 27.756947	192.168.43.175	40.119.211.203	TCP	54.61207 → 443 [ACK] Seq=102 Ack=172 Win=513 Len=0
1724 27.757816	185.211.88.131	192.168.43.175	TCP	54.61207 → 443 [ACK] Seq=153 Ack=569 Win=65535 Len=0
1725 27.786668	185.211.88.131	192.168.43.175	TCP	54.61207 → 443 [ACK] Seq=153 Ack=1969 Win=65535 Len=0
1726 27.801723	185.211.88.131	192.168.43.175	TCP	54.61207 → 443 [ACK] Seq=153 Ack=2549 Win=65535 Len=0

حال به بخش Transport Control Protocol می‌رویم و عدد مربوط به پورت مبدأ و مقصد را می‌ایم:



همچنین checksum مربوط به آن هم در اسکرین شات مشخص است که برابر با: 0x8d19 میباشد.

حال به توضیح این موضوع میپردازیم که این اعداد در مبدا و مقصد چه چیزی را مشخص میکنند.

میدانیم که یک پراسس یا برنامه‌ی کاربردی میتواند یک یا چند سوکت داشته باشد و گفتم که سوکت دریچه‌ای است که یک پراسس داده‌های خود را از آنجا به شبکه میدهد یا داده‌های دریافتی را از شبکه میگیرد. از آنجایی که host گیرنده میتواند بیش از یک سوکت داشته باشد، هر سوکت یک شناسه‌ی منحصر به فرد دارد. در واقع ما به کمک IP Address میتوانیم سیستم مورد نظر را در شبکه پیدا کنیم و سپس به کمک شماره پورت اون پراسس که یا درخواست داده است یا میخواهد سرویس دهد را، میتوان پیدا کرد. وقتی هر سوکت یک شناسه‌ی منحصر به فرد داشته باشد، لایه‌ی انتقال میتواند با نگاه کردن به شماره‌ی پورت مقصد در بسته‌های ورودی، سوکت مقصد آن‌ها را پیدا کند و داده‌های داخل هر قطعه را به سوکت متناظر بفرستد. در مقصد هم وقتی بخواهد جواب را به همان سوکت مبدا بفرستد به شماره‌ی پورت مبدا نگاه میکند و سوکت مبدا را شناسایی میکند.

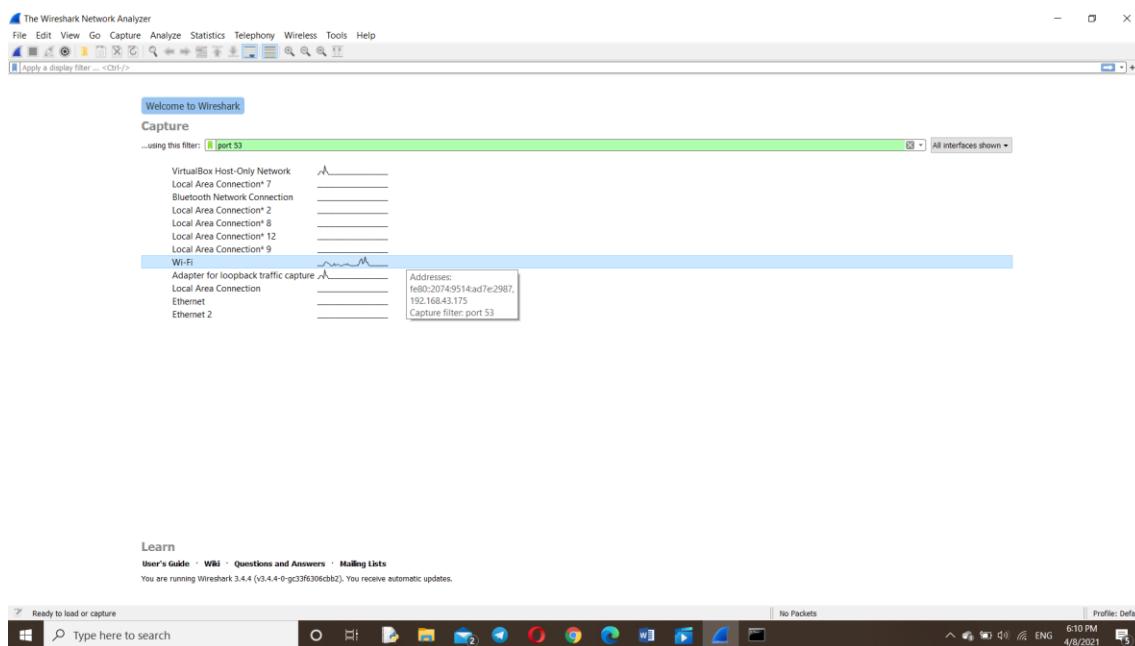
این شماره‌های پورت مبدأ و مقصد هرکدام یک عدد 16 بیتی (از 0 تا 65535) است. شماره‌های 0 تا 1023 که به well-known port numbers معروفند، برای پروتکل‌های کاربردی معروف اینترنت مانند HTTP که از پورت 80 استفاده میکند و FTP (که از پورت 21 استفاده میکند) کnar گذاشته شده اند.

وقتی یک سوکت UDP یا TCP ایجاد میکنیم لایه‌ی انتقال معمولاً به طور خودکار و رندوم یک شماره‌ی پورت به آن نسبت میدهد. شماره‌ای که لایه‌ی انتقال به این سوکت میدهد شماره‌ای بین 1024 تا 65535 که از قبل رزرو نشده اند، میباشد. پس در این حالت شماره پورت مبدا این شماره‌ی رندوم است و شماره‌ی پورت مقصد احتمالاً مربوط به یکی از سوروها و سرویس دهنده‌ها میباشد که یکی از همان پورت‌های معروف و رزرو شده میباشد (البته میتواند هم از آن‌ها نباشد لزوماً) و درواقع اگر برنامه نویس بخواهد کد سمت سرویس دهنده‌ی یکی از «پروتکل‌های شناخته شده و رزرو شده» را بنویسد باید شماره‌ی پورت شناخته شده برای آن پروتکل را برای سوکت سرویس دهنده استفاده کند. به طور معمول برنامه‌های سمت client اجازه میدهند تا لایه‌ی انتقال به طور خودکار شماره‌ی پورت را به سوکت‌ها تخصیص دهد درحالی که شماره‌ی پورت برنامه‌های سمت سرویس دهنده به طور مشخص توسط برنامه نویس تعیین میشوند.

## قسمت دوم آزمایش

### «capture filter» کار با

ابتدا به صفحه اول برنامه رفته و در قسمت port 53 capture file را مینویسیم. درنهایت چون از Wi-Fi استفاده میکنیم، اینترفیسی که با آن به اینترنت دسترسی دارم همان Wi-Fi میباشد و آن را انتخاب میکنیم. سپس به DNS را بازمیکنیم و دستور ipconfig /flushdns اجرا میکنیم که cache مربوط به DNS را در سیستم ما پاک کند.



سپس طبق گفته‌ی دستور کار ping را google.com میکنیم و سپس دستور nslookup 1.1.1.1 را هم وارد میکنیم.

```
C:\ Command Prompt
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\W R>ipconfig /flushdns

Windows IP Configuration

Successfully flushed the DNS Resolver Cache.

C:\Users\W R>ping google.com

Pinging google.com [172.217.19.14] with 32 bytes of data:
Reply from 172.217.19.14: bytes=32 time=69ms TTL=106
Reply from 172.217.19.14: bytes=32 time=205ms TTL=106
Reply from 172.217.19.14: bytes=32 time=125ms TTL=106
Reply from 172.217.19.14: bytes=32 time=66ms TTL=106

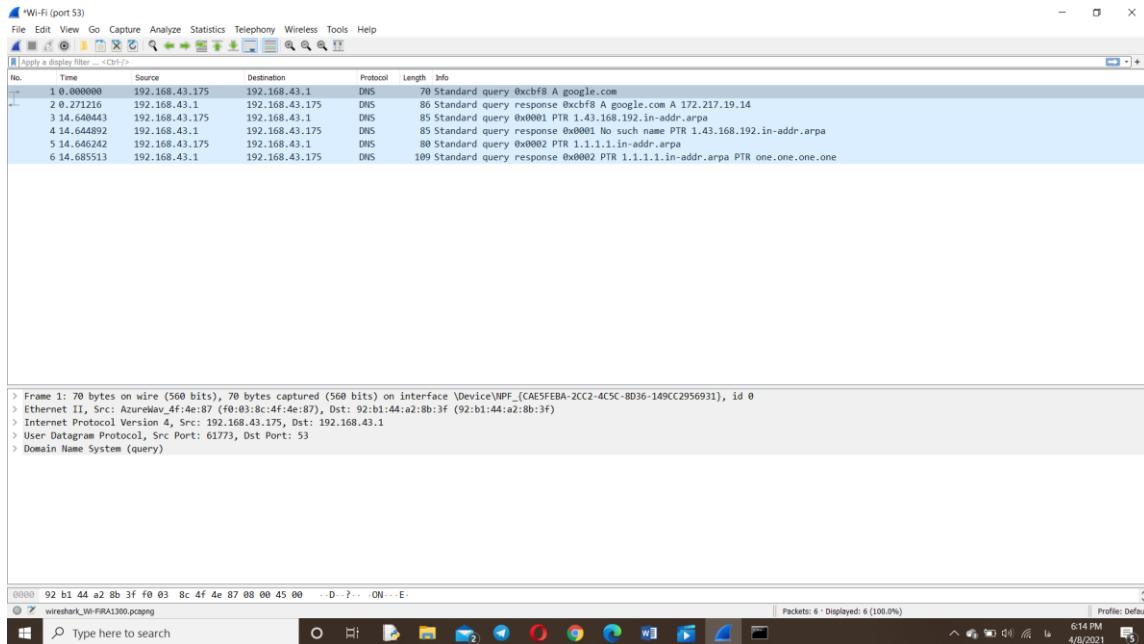
Ping statistics for 172.217.19.14:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 66ms, Maximum = 205ms, Average = 116ms

C:\Users\W R>nslookup 1.1.1.1
Server: Unknown
Address: 192.168.43.1

Name: one.one.one.one
Address: 1.1.1.1

C:\Users\W R>
```

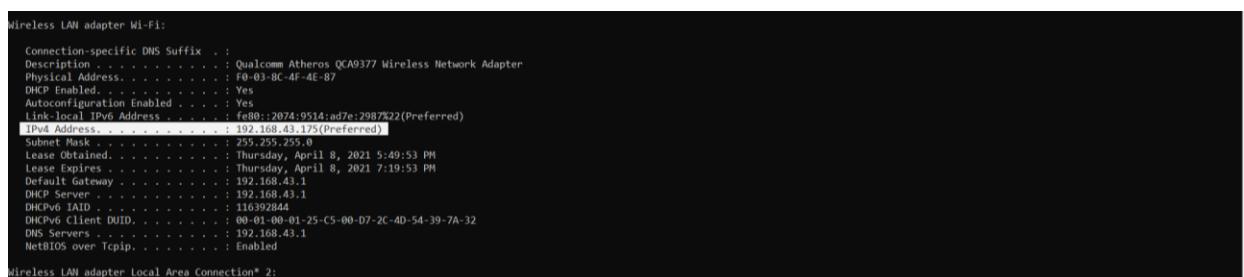
حال شنود بسته‌ها را متوقف می‌کنیم و اکنون صرفاً بسته‌های پروتکل DNS را در Wireshark مشاهده می‌کنیم.



**سوال 6:** یکی از بسته‌ها که از سیستم شما ارسال شده است را انتخاب کنید. پروتکل لایه Transport چیست؟

آدرس IP مقصد چیست؟ سرایند لایه دوم را انتخاب کنید. آدرس مبدأ و مقصد را پادداشت کنید.

برای اینکه بفهمیم کدام بسته از سیستم ما ارسال شده است لازم است ابتدا در cmd دستور ipconfig /all را بزنیم که تمام اطلاعات مربوط به کارت شبکه‌های سیستم ما را میدهد. حال چون از Wi-Fi استفاده میکنم به مشخصات آن بخش میروم که طبق آن IPv4 من برابر است با: 192.168.43.175.



حال در لیست packet های دریافت شده هر کدام از packet هایی که در ستون source این آدرس IP برای آن ثبت شده باشد، از سیستم ما ارسال شده است.

حال برای مثال بسته‌ی select شده‌ی زیر را انتخاب میکنیم:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.43.175	192.168.43.1	DNS	70	Standard query 0xcbf8 A google.com
2	0.271216	192.168.43.1	192.168.43.175	DNS	86	Standard query response 0xcbf8 A google.com A 172.217.19.14
3	14.640443	192.168.43.175	192.168.43.1	DNS	85	Standard query 0x0001 PTR 1.43.168.192.in-addr.arpa
4	14.644892	192.168.43.1	192.168.43.175	DNS	85	Standard query response 0x0001 No such name PTR 1.43.168.192.in-addr.arpa
5	14.646242	192.168.43.175	192.168.43.1	DNS	80	Standard query 0x0002 PTR 1.1.1.1.in-addr.arpa
6	14.685513	192.168.43.1	192.168.43.175	DNS	109	Standard query response 0x0002 PTR 1.1.1.1.in-addr.arpa PTR one.one.one.one

پروتکل لایه‌ی آن طبق اسکرین شات زیر Transport User Datagram Protocol(UDP) میباشد. برای تمامی بسته‌های DNS پروتکل این لایه UDP میباشد.

آدرس IP مقصد را هم میتوان در Destination Address در بخش اطلاعات مربوط به لایه‌ی Network و هم در ستون packet مربوط به آن در لیست Destination مشاهده کرد که برابر است با: 192.168.43.1



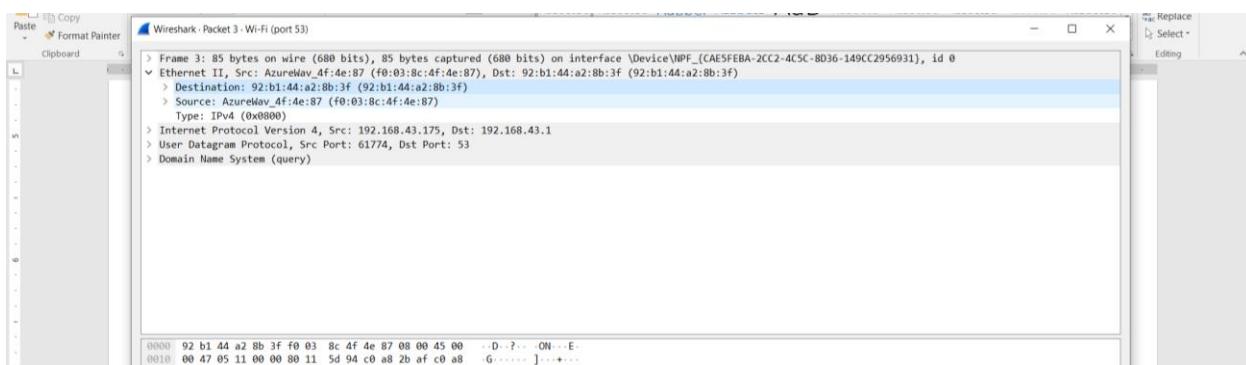
لایه‌ی دوم که همان لایه‌ی Data Link میباشد را انتخاب میکنیم. همانطور که در تصویر زیر دیده میشود:

آدرس مبدأ: f0:03:8c:4f:4e:87 (AzureWav\_4f:4e:87) همچنین به صورت AzureWav\_4f:4e:87 هم نشان داده شده

آدرس مقصد: 92:b1:44:a2:8b:3f

(این ها همان آدرس‌های فیزیکی میباشند)

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.43.175	192.168.43.1	DNS	70	Standard query 0xcbf8 A google.com
2	0.271216	192.168.43.1	192.168.43.175	DNS	86	Standard query response 0xcbf8 A google.com A 172.217.19.14
3	14.640443	192.168.43.175	192.168.43.1	DNS	85	Standard query 0x0001 PTR 1.43.168.192.in-addr.arpa
4	14.644892	192.168.43.1	192.168.43.175	DNS	85	Standard query response 0x0001 No such name PTR 1.43.168.192.in-addr.arpa
5	14.646242	192.168.43.175	192.168.43.1	DNS	80	Standard query 0x0002 PTR 1.1.1.1.in-addr.arpa



## سوال 7: کدام یک از آدرس های پیدا کرده در بخش قبل را میتوانید در خروجی دستور ipconfig /all مشاهده

کنید؟

اگر دستور ipconfig /all را در cmd اجرا کنیم و به بخش Wi-Fi مراجعه کنیم، آنگاه خواهیم داشت:

```
Wireless LAN adapter Wi-Fi:
Connection-specific DNS Suffix . :
Description . . . . . : Qualcomm Atheros QCA9377 Wireless Network Adapter
Physical Address . . . . . : 00:03:8C:4F:4E:87
DHCP Enabled . . . . . : Yes
Autoconfiguration Enabled . . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::2074:951a:ad7e:298%22(Preferred)
IPv4 Address . . . . . : 192.168.43.175(PREFERRED)
Subnet Mask . . . . . : 255.255.255.0
Lease Obtained . . . . . : Thursday, April 8, 2021 5:49:53 PM
Lease Expires . . . . . : Thursday, April 8, 2021 7:19:53 PM
Default Gateway . . . . . : 192.168.43.1
DHCP Server . . . . . : 192.168.43.1
DHCPv6 IAID . . . . . : 116392844
DHCPv6 Client UUID . . . . . : 00-01-00-01-25-C5-00-07-2C-4D-54-39-7A-32
DNS Servers . . . . . : 192.168.43.1
NetBIOS over Tcpip . . . . . : Enabled
```

همانطور که مشاهده میشود همان آدرس مبدأ در لایه‌ی Transport میباشد) یا همان آدرسی که از روی آن در لیست packet های شنود شده فهمیدیم کدام packet از سیستم ما ارسال شده است). مقدار آن برابر با : 192.168.43.175 میباشد.

همچنین آدرس های Default Gateway و DHCP Server نیز برابر با همان آدرس مقصد در لایه‌ی Destination میباشد) یا همان آدرسی که از روی آن در لیست packet های شنود شده درستون Transport برای بسته‌ی انتخاب شده دیدیم)

همچنین Physical Address نوشته شده نیز برابر با همان آدرس source نوشته شده در لایه‌ی دوم یعنی Data Link میباشد که برابر با : f0:03:8c:4f:4e:87 میباشد.

(البته در خروجی ipconfig /all مقدار آدرس فیزیکی gateway نبود. ولی با زدن دستور arp -a در CMD میتوان مشاهده کرد که آدرس destination (مقصد) نوشته شده در لایه‌ی دوم همان آدرس فیزیکی میباشد. در اسکرین شات زیر میتوان آن را دید.)

The screenshot shows a Command Prompt window with the following text:

```
Command Prompt
Ping statistics for 192.168.43.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 3ms, Maximum = 45ms, Average = 13ms

C:\Users\...\Rarp -a

Interface: 192.168.56.1 --- 0xc
Internet Address      Physical Address      Type
192.168.56.255        ff-ff-ff-ff-ff-ff  static
224.0.0.22             01-00-5e-00-00-16  static
224.0.0.251            01-00-5e-00-00-fb  static
224.0.0.252            01-00-5e-00-00-fc  static
239.255.255.250        01-00-5e-7f-ff-fa  static

Interface: 192.168.43.175 --- 0x16
Internet Address      Physical Address      Type
192.168.43.1           02-b1-44-a2-8b-3f  dynamic
192.168.43.255         ff-ff-ff-ff-ff-ff  static
224.0.0.22             01-00-5e-00-00-16  static
224.0.0.251            01-00-5e-00-00-fb  static
224.0.0.252            01-00-5e-00-00-fc  static
239.255.255.250        01-00-5e-7f-ff-fa  static
255.255.255.255        ff-ff-ff-ff-ff-ff  static
```

سوال 8: یک بسته مربوط به دستور Ping را انتخاب کنید و به بخش مربوط به پروتکل DNS در آن بروید. به بخش Queries بروید. چه ای انتخاب شده است؟ به نظر شما این درخواست DNS برای چه کاری استفاده شده است؟

بسته select شده زیر مربوط به دستور ping میباشد. آن را انتخاب میکنیم :

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.43.175	192.168.43.1	DNS	70	Standard query 0xcbf8 A google.com
2	0.271216	192.168.43.1	192.168.43.175	DNS	86	Standard query response 0xcbf8 A google.com A 172.217.19.14
3	14.640443	192.168.43.175	192.168.43.1	DNS	85	Standard query response 0x0001 PTR 1.43.168.192.in-addr.arpa
4	14.644892	192.168.43.1	192.168.43.175	DNS	85	Standard query response 0x0001 No such name PTR 1.43.168.192.in-addr.arpa
5	14.646242	192.168.43.175	192.168.43.1	DNS	80	Standard query response 0x0002 PTR 1.1.1.1.in-addr.arpa
6	14.685513	192.168.43.1	192.168.43.175	DNS	109	Standard query response 0x0002 PTR 1.1.1.1.in-addr.arpa PTR one.one.one.one

حال به بخش مربوط به پروتکل DNS در آن میرویم، سپس به بخش Queries میرویم:

Wireshark - Packet 1 - Wi-Fi (port 53)						
> Frame 1: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface \Device\NPF_{CAE5FEB...}, id 0						
> Ethernet II, Src: AzureWave_4f:4e:87 (f0:03:8c:4f:4e:87), Dst: 92:b1:44:a2:8b:3f (92:b1:44:a2:8b:3f)						
> Internet Protocol Version 4, Src: 192.168.43.175, Dst: 192.168.43.1						
> User Datagram Protocol, Src Port: 61773, Dst Port: 53						
▼ Domain Name System (query)						
Transaction ID: 0xcbf8						
Flags: 0x0100 Standard query						
Questions: 1						
Answer RRs: 0						
Authority RRs: 0						
Additional RRs: 0						
▼ Queries						
▼ google.com: type A, class IN						
Name: google.com						
[Name Length: 10]						
[Label Count: 2]						
Type: A (Host Address) (1)						
Class: IN (0x0001)						
[Response In: 2]						

همانطور که مشاهده میشود type مربوط به آن A میباشد.

سرورهای DNS یک رکورد DNS ایجاد می کنند تا اطلاعات مهم در مورد یک دامنه یا hostname ، به ویژه آدرس IP فعلی آن را ارائه دهند.

سرвис دهنده های DNS همگی با هم یک پایگاهداده توزیع شده DNS را پیاده سازی می کنند که رکوردهای مرجع (Resource Record-RR) شامل رکوردهای نگاشت hostname به IP، را در خود ذخیره میکند. هر پیام پاسخ DNS یک یا چند رکورد مرجع را در خود حمل می کند. هر رکورد مرجع از چهار فیلد زیر تشکیل شده است:

(Name, Value, Type, TTL)

TTL طول عمر رکورد مرجع را مشخص می کند، یعنی زمانی که این رکورد باید از حافظه نهان سرورهایی که آن را ذخیره کرده اند، پاک شود. در مثال های انواع رکوردها که آمده این فیلد را نادیده میگیریم. معنای فیلدهای value و Name به فیلد Type بستگی دارد.

اگر آنگاه **Type=A** همان **Name** همان **hostname** مربوط به آن **IP address** است و **Value** همان **hostname** است. بنابراین، رکورد نوع A همان نگاشت استاندارد **hostname** یا درواقع **Domain Name** به **IP** است. برای مثال، رکوردي به شکل (A , 147.37.93.126 , relay1.bar.foo.com ) یک رکورد **Address** نوع A میباشد. در ضمن در این منظور از آدرس همان آدرس IPv4 میباشد.

پس درواقع در اینجا که ما بسته‌ی اول از ping را هم انتخاب کردیم DNS یک درخواست تایپ A داده تا **IP** مربوط به آدرس دامنه‌ی google.com را پیدا کند و ادامه‌ی کار را به واسطه‌ی این آدرس **Address** پیش ببرد.

**سوال 9:** یک بسته مربوط به دستور nslookup را انتخاب کنید و به بخش مربوط به پروتکل DNS در آن **Queries** برای چه type ای انتخاب شده است؟ به نظر شما این درخواست DNS برای چه کاری استفاده شده است؟

بسته select شده زیر مربوط به دستور nslookup میباشد. آن را انتخاب میکنیم :

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.43.175	192.168.43.1	DNS	70	Standard query 0xcbf8 A google.com
2	0.271216	192.168.43.1	192.168.43.175	DNS	86	Standard query response 0xcbf8 A google.com A 172.217.19.14
3	14.640443	192.168.43.175	192.168.43.1	DNS	85	Standard query 0x0001 PTR 1.43.168.192.in-addr.arpa
4	14.644892	192.168.43.1	192.168.43.175	DNS	85	Standard query response 0x0003 No such name PTR 1.43.168.192.in-addr.arpa
5	14.646242	192.168.43.175	192.168.43.1	DNS	80	Standard query 0x0002 PTR 1.1.1.1.in-addr.arpa
6	14.685513	192.168.43.1	192.168.43.175	DNS	109	Standard query response 0x0002 PTR 1.1.1.1.in-addr.arpa PTR one.one.one.one

حال به بخش مربوط به پروتکل DNS در آن میرویم، سپس به بخش Queries میرویم:

Wireshark - Packet 5 - Wi-Fi (port 53)	
> Frame 5: 80 bytes on wire (640 bits), 80 bytes captured (640 bits) on interface \Device\NPF_{CAE5FEB0-2CC2-4C5C-8D36-149CC2956931}, id 0	
> Ethernet II, Src: AzureWave_4f:4e:87 (f0:03:8c:4f:4e:87), Dst: 92:b1:44:a2:8b:3f (92:b1:44:a2:8b:3f)	
> Internet Protocol Version 4, Src: 192.168.43.175, Dst: 192.168.43.1	
> User Datagram Protocol, Src Port: 61775, Dst Port: 53	
▼ Domain Name System (query)	
Transaction ID: 0x0002	
Flags: 0x0100 Standard query	
Questions: 1	
Answer RRs: 0	
Authority RRs: 0	
Additional RRs: 0	
Queries	
1.1.1.1.in-addr.arpa: type PTR, class IN	
Name: 1.1.1.1.in-addr.arpa	
[Name Length: 20]	
[Label Count: 6]	
Type: PTR (domain name PoinTeR) (12)	
Class: IN (0x0001)	
[Response In: 6]	

همانطور که مشاهده میشود type PTR مربوط به آن میباشد.

**Type=PTR:** رکورد DNS که یک رکورد DNS معکوس نامیده میشود، یک IP Address را به یک نام دامنه ارجاع می‌دهد. دقیقاً برعکس همان کاری که رکورد A انجام می‌دهد. در واقع این رکورد یک اتصال صحیح بین دامنه و آی پی برقرار می‌کند تا درخواست‌ها اشتباها به سرورهای دیگر ارسال نشود.

درواقع در این بسته از دستور nslookup که در نظر گرفته ایم، DNS یک درخواست تایپ PTR داده است تا نام دامنه مربوط به IP Address ای که زدیم یعنی 1.1.1.1 را بیابد و این دو را به هم نگاشت کند.

برای توضیح بیشتر در رابطه با کمکی که به ما میکند میتوان گفت: از این رکورد برای ایمیل سرور استفاده می‌شود در واقع اطمینان از صحت اتصالات IP و hostname یا همان دامنه به واسطه این رکورد میسر میگردد. گاهی ممکن است خروجی های ایمیل بدون وجود این رکورد مارک شوند، به عنوان اسپم در نظر گرفته شوند و یا اینکه ریجکت گردند. پس برای موقعی مفید است که شما IP اختصاصی داشته باشید. این رکورد بررسی می کند که آیا دامنه شما به آدرس IP به صورت صحیح متصل شده است یا خیر. از کاربردهای این رکورد معمولاً در استفاده از وب سرور یا از سرور ابری می باشد، زیرا این رکورد به سرور شما در مورد مسائل امنیتی جهت اتصال به Mail Server قابل توجهی خواهد نمود.

## سوال 10: به نظر شما چه type های دیگری ممکن است وجود داشته باشد؟ سه مورد را یادداشت کنید.

Type های زیادی وجود دارد که بنده در ادامه سه مورد را معرفی کرده و توضیح داده ام.(لازم به ذکر است منبع مورد استفاده‌ی بنده کتاب جیمز کورووس و کیت راس میباشد) طبق توضیحی که در سوال 8 دادم هر پیام پاسخ یک یا چند رکورد مرجع را در خود حمل می کند. هر رکورد مرجع از چهار فیلد زیر تشکیل شده است:

(Name, Value, Type, TTL)

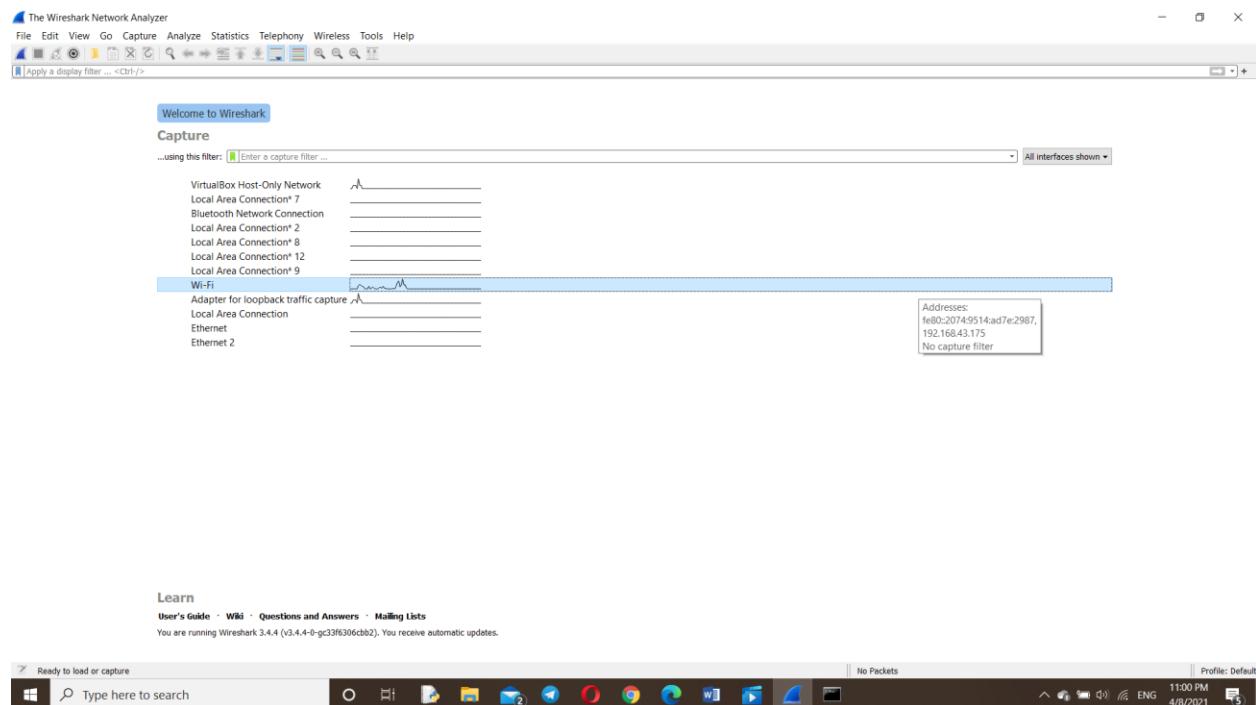
- اگر **Type=AAAA** ، آنگاه این رکورد همانند رکورد نوع A میباشد با این تفاوت که آدرس IP که استفاده میشود همان IPv6 یک دامنه میباشد.
- اگر **Type=NS** ، آنگاه Name همان نام دامنه (مانند foo.com) و Value درواقع همان hostname مربوط به یک authoritative DNS server است که میداند چگونه به آدرس‌های IP مربوط به host های این دامنه برسیم. از این رکورد برای هدایت پرس و جوهای DNS در زنجیره‌ی پرس و جو(query chain) استفاده میشود. درواقع به صورت خلاصه میتوان گفت این رکورد مشخص‌کننده‌ی DNS server معتبری است که می‌تواند به درخواست‌های DNS مربوط به یک دامنه‌ی خاص و بعضی زیردامنه‌های آن پاسخ بدهد. برای مثال، رکوردي به شکل (NS , dns.foo.com , foo.com) یک رکورد نوع NS است.
- اگر **Type=CNAME** ، آنگاه canonical hostname همان Value مربوط به alias hostname (نام مستعار) است. این رکورد می‌تواند نام متعارف(canonical name) یک host را در اختیار یک پرسو جو کننده یا درواقع query قرار دهد. به عبارت دیگر از رکورد cname برای هدایت اتومات یک نام به نام دامنه دیگر استفاده می‌شود. شناخته شده‌ترین رکورد CNAME همان www می‌باشد که www.yourdomain.com به آدرس yourdomain.com ارجاع می‌دهد و باعث می‌شود هر دو آدرس یک محتوا را نمایش دهند. یا مثلا (foo.com, relay1.bar.foo.com, CNAME) هم یک رکورد نوع CNAME میباشد.
- اگر **Type=MX** آنگاه canonical name همان Value مربوط به یک mail server با alias hostname (نام مستعار) است. برای مثال رکوردي به شکل (MX , mail.bar.foo.com , foo.com) یک MX نوع رکورد است. این رکورد اجازه می‌دهد تا mail server ها نام‌های مستعار ساده داشته باشند. به عبارتی میتوان گفت که این رکورد ایمیل ها را به یک mail server مخصوص هدایت میکند. توجه کنید که با این رکورد به شرکت‌ها اجازه میدهد از یک نام مستعار واحد همزمان برای mail server و برای یکی از سرورهای دیگرش مانند Web Server استفاده کنید. اگر یک DNS client بخواهد canonical name

یک mail server را بداند باید برای رکورد MX آن دامنه درخواست دهد، برای یافتن canonical name سرورهای دیگر باید برای رکورد CNAME درخواست دهد.

## قسمت سوم آزمایش

### «Display Filter» کار با

دوباره به صفحه‌ی اول برنامه میرویم. این بار اینترفیس را بدون هیچ Capture Filter ای انتخاب میکنیم.



در CMD دستور tracert p30download.com را وارد میکنیم و منتظر میمانیم تا کار دستور تمام شود:

```
Microsoft Windows [Version 10.0.18363.1256]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Rx> tracert p30download.com

Tracing route to p30download.com [5.144.130.115]
over a maximum of 30 hops:
1 * * * Request timed out.
2 * * * Request timed out.
3 29 ms 29 ms 36 ms 10.222.211.65
4 84 ms 30 ms 46 ms 10.222.212.23
5 * * * Request timed out.
6 73 ms 36 ms 23 ms 10.222.212.65
7 * * * Request timed out.
8 72 ms 26 ms 31 ms 10.221.57.142
9 42 ms 40 ms 33 ms 10.222.119.1
10 25 ms 40 ms 29 ms 172.17.132.9
11 * * * Request timed out.
12 72 ms 39 ms 42 ms 10.202.1.5
13 * * * Request timed out.
14 44 ms 44 ms 36 ms 5.144.130.115.static.hostiran.name [5.144.130.115]

Trace complete.

C:\Users\Rx>
```

بدون اینکه شنود بسته را متوقف کنیم در قسمت display filter مقدار dns را تایپ میکنیم:

No.	dns	Source	Destination	Protocol	Length	Info
593	dnsserver	108.177.15.188	192.168.43.175	TCP	66	[TCP Keep-Alive ACK] 5228 + 65175 [ACK] Seq=1 Ack=2 Win=265 Len=0 SRE=1
582	106.656461	185.211.88.218	192.168.43.175	TLSv1.3	593	Application Data
583	106.697205	185.168.43.175	185.211.88.218	TCP	54	549818 → 443 [ACK] Seq=2630 Ack=859 Win=130560 Len=0
584	162.160318	212.16.77.188	192.168.43.175	TLSv1.2	82	Application Data
585	162.162576	192.168.43.175	212.16.77.188	TLSv1.2	86	Application Data
586	162.190608	212.16.77.188	192.168.43.175	TCP	54	5443 → 65356 [ACK] Seq=265 Ack=305 Win=47 Len=0
587	106.356832	138.199.14.82	192.168.43.175	TCP	54	[TCP Keep-Alive] 88 → 65202 [ACK] Seq=1 Ack=1 Win=501 Len=0
588	106.356972	192.168.43.175	138.199.14.82	TCP	54	[TCP Keep-Alive ACK] 65202 → 88 [ACK] Seq=1 Ack=2 Win=513 Len=0
589	106.115488	185.60.216.53	192.168.43.175	TLSv1.2	160	Application Data
590	106.156215	192.168.43.175	185.60.216.53	TCP	54	58131 → 443 [ACK] Seq=218 Ack=479 Win=514 Len=0
591	106.159673	138.199.14.82	192.168.43.175	TCP	54	[TCP Keep-Alive] 88 → 65202 [ACK] Seq=1 Ack=1 Win=501 Len=0
592	106.159680	192.168.43.175	138.199.14.82	TCP	54	[TCP Keep-Alive ACK] 65202 → 88 [ACK] Seq=1 Ack=2 Win=513 Len=0
593	178.181216	192.168.43.175	142.250.181.106	TCP	55	[TCP Keep-Alive] 50877 → 443 [ACK] Seq=1 Ack=131 Win=510 Len=1
594	178.218505	142.250.181.106	192.168.43.175	TCP	66	[TCP Keep-Alive ACK] 443 → 50877 [ACK] Seq=131 Ack=2 Win=370 Len=0 SLE=1
595	179.142592	192.168.43.175	13.94.251.244	TCP	55	[TCP Keep-Alive] 50915 → 443 [ACK] Seq=2134 Ack=6276 Win=130816 Len=1
596	179.171717	13.94.251.244	192.168.43.175	TCP	66	[TCP Keep-Alive ACK] 443 → 50915 [ACK] Seq=6276 Ack=2135 Win=98304 Len=0 SLE=2134 SRE=2135
597	179.625592	192.168.43.175	68.232.34.200	TCP	55	[TCP Keep-Alive] 50916 → 443 [ACK] Seq=1824 Ack=8546 Win=130304 Len=1
598	179.665195	68.232.34.200	192.168.43.175	TCP	66	[TCP Keep-Alive ACK] 443 → 50916 [ACK] Seq=8546 Ack=1825 Win=97280 Len=0 SLE=1824 SRE=1825
599	181.097738	192.168.43.175	185.60.216.53	TLSv1.2	85	Application Data
600	181.121350	185.60.216.53	192.168.43.175	TCP	54	443 → 50131 [ACK] Seq=479 Ack=249 Win=502 Len=0
601	181.290017	185.60.216.53	192.168.43.175	TLSv1.2	92	Application Data
602	181.331448	192.168.43.175	185.60.216.53	TCP	54	58131 → 443 [ACK] Seq=249 Ack=517 Win=514 Len=0

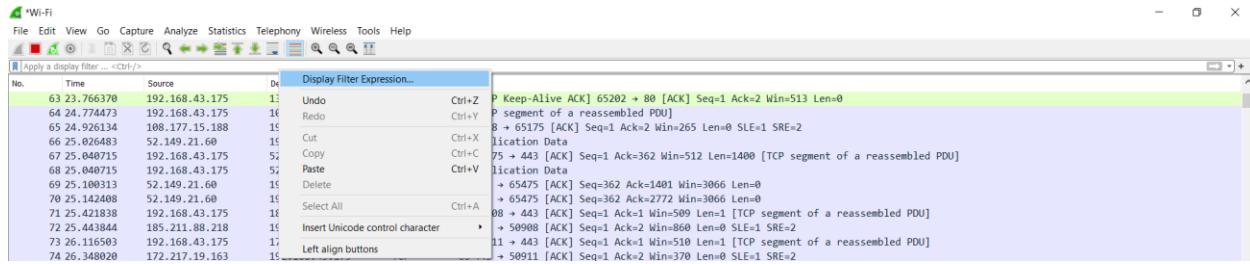
> Frame 1: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface \Device\NPF\_{CAE5FEB0-2CC2-4C5C-8D36-149CC2956931}, id 0  
> Ethernet II, Src: 92:b1:44:a2:8b:3f (92:b1:44:a2:8b:3f), Dst: AzureNav\_4f:4e:87 (f0:03:c8:4f:4e:87)  
> Internet Protocol Version 4, Src: 138.199.14.82, Dst: 192.168.43.175  
> Transmission Control Protocol, Src Port: 80, Dst Port: 65202, Seq: 1, Ack: 1, Len: 0

سپس اینتر میزینیم و مشاهده میکنیم که صرفا بسته های مربوط به پروتکل DNS انتخاب شدند در حالی که سایر بسته ها نیز در حال دریافت شدن از گرداننده کارت شبکه هستند.

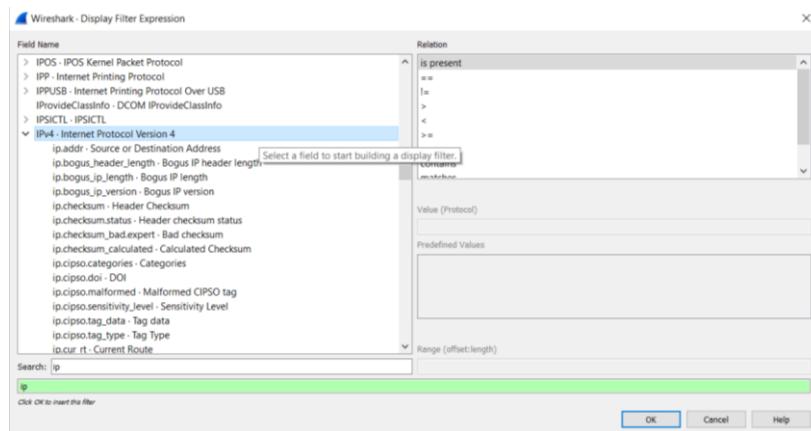
No.	dns	Source	Destination	Protocol	Length	Info
94	dns	192.168.43.175	192.168.43.1	DNS	75	Standard query 0x23ba A p30download.com
156	64.110187	192.168.43.175	192.168.43.1	DNS	91	Standard query response 0x23ba A p30download.com A 5.144.130.115
157	64.209665	192.168.43.175	192.168.43.1	DNS	80	Standard query 0x1bd9 A translate.google.com
158	64.369661	192.168.43.1	192.168.43.175	DNS	117	Standard query response 0x1bd9 A translate.google.com CNAME www3.l.google.com A 216.58.207.110
198	68.893211	192.168.43.175	192.168.43.1	DNS	86	Standard query 0x7d77 PTR 65.211.222.10.in-addr.arpa
200	68.938016	192.168.43.1	192.168.43.175	DNS	163	Standard query response 0xd727 No such name PTR 65.211.222.10.in-addr.arpa SOA prisoner.iana.org
225	74.596273	192.168.43.175	192.168.43.1	DNS	86	Standard query 0x4489 PTR 23.212.222.10.in-addr.arpa
226	74.645877	192.168.43.1	192.168.43.175	DNS	163	Standard query response 0x4486 No such name PTR 23.212.222.10.in-addr.arpa SOA prisoner.iana.org
264	91.874996	192.168.43.175	192.168.43.1	DNS	86	Standard query 0x2ab9 PTR 65.212.222.10.in-addr.arpa
265	91.974832	192.168.43.175	192.168.43.1	DNS	86	Standard query 0x2ab9 PTR 65.212.222.10.in-addr.arpa
266	92.014545	192.168.43.175	192.168.43.1	DNS	163	Standard query response 0x2ab9 No such name PTR 65.212.222.10.in-addr.arpa SOA prisoner.iana.org
267	92.014545	192.168.43.1	192.168.43.175	DNS	86	Standard query response 0x2ab9 No such name PTR 65.212.222.10.in-addr.arpa
280	94.827469	192.168.43.175	192.168.43.1	DNS	81	Standard query 0x9082 A api.intelsa.intel.com
284	94.923301	192.168.43.175	192.168.43.1	DNS	81	Standard query 0x9082 A api.intelsa.intel.com
290	95.064538	192.168.43.175	192.168.43.1	DNS	81	Standard query response 0x9082 No such name A api.intelsa.intel.com
291	95.064688	192.168.43.1	192.168.43.175	DNS	81	Standard query response 0x9082 No such name A api.intelsa.intel.com
334	109.369949	192.168.43.175	192.168.43.1	DNS	86	Standard query 0x44b9 PTR 142.57.221.10.in-addr.arpa
335	109.463317	192.168.43.175	192.168.43.1	DNS	86	Standard query 0x44b9 PTR 142.57.221.10.in-addr.arpa
336	109.506661	192.168.43.1	192.168.43.175	DNS	163	Standard query response 0x44b9 No such name PTR 142.57.221.10.in-addr.arpa SOA prisoner.iana.org
354	115.120361	192.168.43.175	192.168.43.1	DNS	85	Standard query 0x33aa PTR 1.119.222.10.in-addr.arpa
355	115.164707	192.168.43.1	192.168.43.175	DNS	162	Standard query response 0x33aa No such name PTR 1.119.222.10.in-addr.arpa SOA prisoner.iana.org
374	120.764158	192.168.43.175	192.168.43.1	DNS	85	Standard query 0x725e PTR 9.132.17.172.in-addr.arpa

> Frame 93: 75 bytes on wire (600 bits), 75 bytes captured (600 bits) on interface \Device\NPF\_{CAE5FEB0-2CC2-4C5C-8D36-149CC2956931}, id 0  
> Ethernet II, Src: AzurNav\_4f:4e:87 (f0:03:c8:4f:4e:87), Dst: 92:b1:44:a2:8b:3f (92:b1:44:a2:8b:3f)  
> Internet Protocol Version 4, Src: 192.168.43.175, Dst: 192.168.43.1  
> User Datagram Protocol, Src Port: 60236, Dst Port: 53  
> Domain Name System (query)  
Transaction ID: 0x23ba  
Flags: 0x0100 Standard query  
Questions: 1  
Answer RRs: 0  
Authority RRs: 0  
Additional RRs: 0  
Queries  
p30download.com type A, class IN  
Name: p30download.com  
[Name Length: 15]  
0000 92 b1 44 a2 8b 3f f0 03 8c 4f 4e 87 08 00 45 00 -D- -? -ON -E -  
Packets: 616 - Displayed: 36 (5.8%)  
Domain Name System: Protocol  
Type here to search  
Windows 10 Profil: Default  
11:03 PM 4/8/2021

سپس در قسمت Display Filter Expression کلیک راست کرده و برروی Display Filter میکنیم.



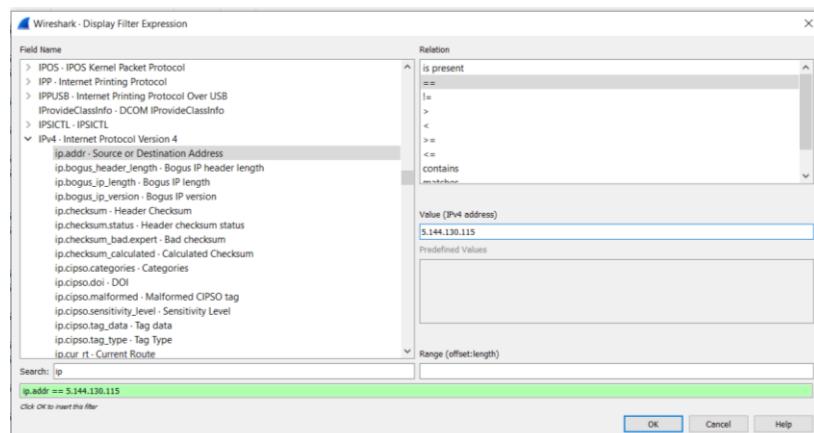
سپس IP را جست وجو میکنیم و IPv4 را از ستون سمت چپ انتخاب میکنیم:



آدرس IP که از دستور tracert به ما گزارش شده است برابر با: 5.144.130.115 میباشد.

```
C:\Users\M R> tracert p30download.com
Tracing route to p30download.com [5.144.130.115]
```

حال از زیربخش های IPv4، بخش ip.addr را انتخاب میکنیم. سپس از بخش relation ==، مقدار == را انتخاب کرده و در بخش Value IP که بالاتر به دست آمد را مینویسیم.



## سوال 11: بعد از کلیک کردن بر روی OK چه اتفاقی می‌افتد؟ در بسته‌هایی که مشخص شده‌اند چه پروتکل‌های را مشاهده می‌کنید؟

بعد از OK کردن در قسمت ip.addr == 5.144.130.115 نوشته می‌شود و به رنگ سبز در می‌آید.

No.	Time	Source	Destination	Protocol	Length	Info
63	23.76637	192.168.43.175	138.199.14.82	TCP	54	54 [TCP Keep-Alive ACK] 65202 → 88 [ACK] Seq=1 Ack=2 Win=513 Len=0
64	24.774473	192.168.43.175	188.177.15.188	TCP	55	[TCP segment of a reassembled PDU]
65	24.926134	188.177.15.188	192.168.43.175	TCP	66	5228 → 65175 [ACK] Seq=1 Ack=2 Win=265 Len=0 SLE=1 SRE=2
66	25.026483	52.149.21.60	192.168.43.175	TLSv1.2	415	Application Data
67	25.040975	192.168.43.175	52.149.21.60	TCP	1454	65475 → 443 [ACK] Seq=1 Ack=362 Win=512 Len=1400 [TCP segment of a reassembled PDU]
68	25.040975	192.168.43.175	52.149.21.60	TLSv1.2	1425	Application Data
69	25.108913	52.149.21.60	192.168.43.175	TCP	54	443 → 65475 [ACK] Seq=362 Ack=1401 Win=3066 Len=0
70	25.142408	52.149.21.60	192.168.43.175	TCP	54	443 → 65475 [ACK] Seq=362 Ack=2772 Win=3066 Len=0
71	25.421838	192.168.43.175	185.211.88.218	TCP	55	50908 → 443 [ACK] Seq=1 Ack=1 Win=509 Len=1 [TCP segment of a reassembled PDU]
72	25.443844	185.211.88.218	192.168.43.175	TCP	66	443 → 50908 [ACK] Seq=1 Ack=2 Win=860 Len=0 SLE=1 SRE=2
73	26.116593	192.168.43.175	172.217.19.163	TCP	55	50911 → 443 [ACK] Seq=1 Ack=1 Win=510 Len=1 [TCP segment of a reassembled PDU]
74	26.348020	172.217.19.163	192.168.43.175	TCP	66	443 → 50911 [ACK] Seq=1 Ack=2 Win=370 Len=0 SLE=1 SRE=2

سپس با زدن اینتر یک سری از بسته‌ها از بین تمامی بسته‌های شنود شده جدا می‌شوند. در واقع تمامی بسته‌های که آدرس IP مبدأ یا مقصد آن‌ها مقدار 5.144.130.115 که همان IP Address مربوط به سایت p30download.com میباشد لیست می‌شوند. در اسکرین شات زیر میتوانید این موضوع را بینید:

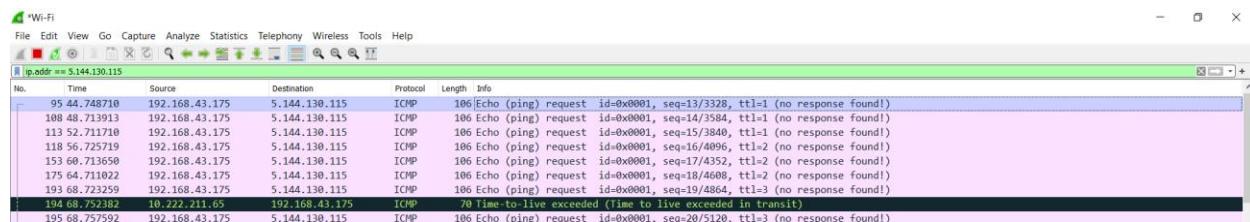
No.	Time	Source	Destination	Protocol	Length	Info
95	44.748710	192.168.43.175	5.144.130.115	ICMP	106	Echo (ping) request id=0x0001, seq=13/3328, ttl=1 (no response found!)
108	48.713913	192.168.43.175	5.144.130.115	ICMP	106	Echo (ping) request id=0x0001, seq=14/3584, ttl=1 (no response found!)
113	52.717110	192.168.43.175	5.144.130.115	ICMP	106	Echo (ping) request id=0x0001, seq=15/3840, ttl=1 (no response found!)
118	56.725719	192.168.43.175	5.144.130.115	ICMP	106	Echo (ping) request id=0x0001, seq=16/4096, ttl=2 (no response found!)
153	60.713650	192.168.43.175	5.144.130.115	ICMP	106	Echo (ping) request id=0x0001, seq=17/4352, ttl=2 (no response found!)
175	64.711022	192.168.43.175	5.144.130.115	ICMP	106	Echo (ping) request id=0x0001, seq=18/4608, ttl=2 (no response found!)
193	68.723259	192.168.43.175	5.144.130.115	ICMP	106	Echo (ping) request id=0x0001, seq=19/4864, ttl=3 (no response found!)
194	68.752382	10.222.211.65	192.168.43.175	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
198	68.757592	192.168.43.175	5.144.130.115	ICMP	106	Echo (ping) request id=0x0001, seq=20/5120, ttl=3 (no response found!)
199	68.787239	10.222.211.65	192.168.43.175	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
200	68.792270	192.168.43.175	5.144.130.115	ICMP	106	Echo (ping) request id=0x0001, seq=21/5376, ttl=3 (no response found!)
204	68.828747	10.222.211.65	192.168.43.175	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
219	74.418117	192.168.43.175	5.144.130.115	ICMP	106	Echo (ping) request id=0x0001, seq=22/5632, ttl=4 (no response found!)
220	74.502228	10.222.212.23	192.168.43.175	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
221	74.507690	192.168.43.175	5.144.130.115	ICMP	106	Echo (ping) request id=0x0001, seq=23/5888, ttl=4 (no response found!)
222	74.538242	10.222.212.23	192.168.43.175	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
223	74.533118	192.168.43.175	5.144.130.115	ICMP	106	Echo (ping) request id=0x0001, seq=24/6144, ttl=4 (no response found!)
224	74.589457	10.222.212.23	192.168.43.175	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
243	80.123792	192.168.43.175	5.144.130.115	ICMP	106	Echo (ping) request id=0x0001, seq=25/6400, ttl=5 (no response found!)
251	83.713027	192.168.43.175	5.144.130.115	ICMP	106	Echo (ping) request id=0x0001, seq=26/6656, ttl=5 (no response found!)
255	83.713465	192.168.43.175	5.144.130.115	ICMP	106	Echo (ping) request id=0x0001, seq=27/6912, ttl=5 (no response found!)
258	91.724593	192.168.43.175	5.144.130.115	ICMP	106	Echo (ping) request id=0x0001, seq=28/7168, ttl=5 (no response found!)
259	91.797927	10.222.212.65	192.168.43.175	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
260	91.803553	192.168.43.175	5.144.130.115	ICMP	106	Echo (ping) request id=0x0001, seq=29/7424, ttl=6 (no response found!)
261	91.839916	10.222.212.65	192.168.43.175	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
262	91.844839	192.168.43.175	5.144.130.115	ICMP	106	Echo (ping) request id=0x0001, seq=30/7680, ttl=6 (no response found!)
263	91.868231	10.222.212.65	192.168.43.175	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
302	97.511089	192.168.43.175	5.144.130.115	ICMP	106	Echo (ping) request id=0x0001, seq=31/7936, ttl=7 (no response found!)
309	101.213881	192.168.43.175	5.144.130.115	ICMP	106	Echo (ping) request id=0x0001, seq=32/8192, ttl=7 (no response found!)
318	105.211553	192.168.43.175	5.144.130.115	ICMP	106	Echo (ping) request id=0x0001, seq=33/8448, ttl=7 (no response found!)
328	109.223982	192.168.43.175	5.144.130.115	ICMP	106	Echo (ping) request id=0x0001, seq=34/8704, ttl=8 (no response found!)
329	109.296076	10.221.57.142	192.168.43.175	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
330	109.301273	192.168.43.175	5.144.130.115	ICMP	106	Echo (ping) request id=0x0001, seq=35/8960, ttl=8 (no response found!)
331	109.327588	10.221.57.142	192.168.43.175	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)

در بسته‌های مشخص شده پروتکل ICMP را میتوان مشاهده کرد.

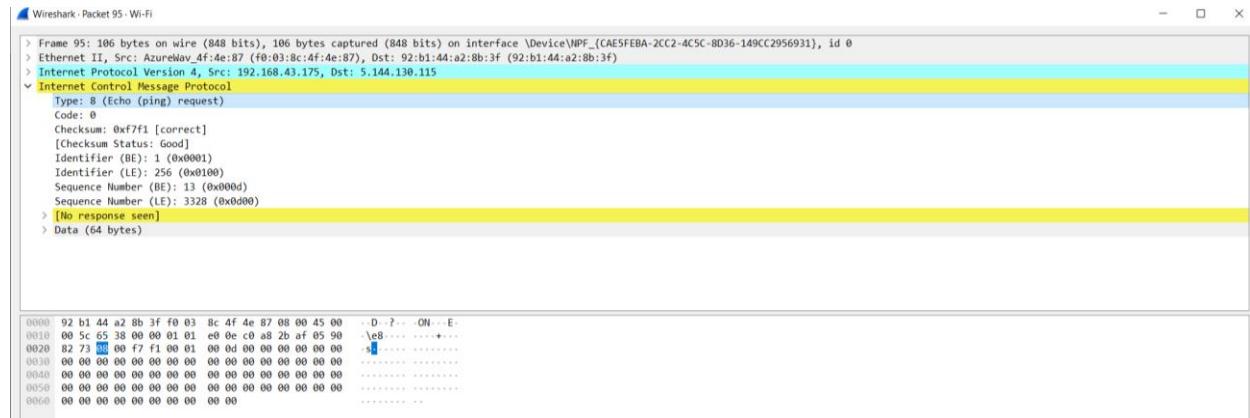
## سوال 12: اولین بسته را انتخاب کنید. به بخش پروتکل Internet Control Message Protocol بروید.

مقدار type را مشخص کنید. به بخش مربوط به پروتکل IP بروید و مقدار TTL را یادداشت کنید.

به صورت زیر ابتدا اولین بسته را انتخاب میکنیم:



سپس به بخش Internet Control Message Protocol را مشخص میکنیم که طبق تصویر زیر مقدار آن 8 میباشد که به عنوان توضیحات در پرانتز گفته شده تایپ آن همان (request) میباشد.



حال به بخش مربوط به پروتکل IP میرویم و مقدار TTL را یادداشت میکنیم که طبق تصویر زیر مقدار آن برابر با 1 میباشد.



در مراحل قبل فهمیدیم که IP Address 192.168.43.175 مربوط به دستگاه ما برابر با که مبدأ آن‌ها ماسن مقدار TTL را یادداشت می‌کنیم. این مقدار در حال تغییر(درحال افزایش) است.

برای مشاهده مقدار TTL هم می‌توان به پروتکل IP هرسن IP و آن مقدار برویم و آن مقدار را یادداشت کنیم. البته راه دیگر آن است که در همان لیست بسته‌های شنود شده، در ستون info این مقدار را برای هر بسته مشاهده کنیم:

## سوال 13: به نظر شما هدف از تغییر این مقدار چیست؟ می‌توانید با مراجعه به هدف دستور tracert آن را شرح دهید.

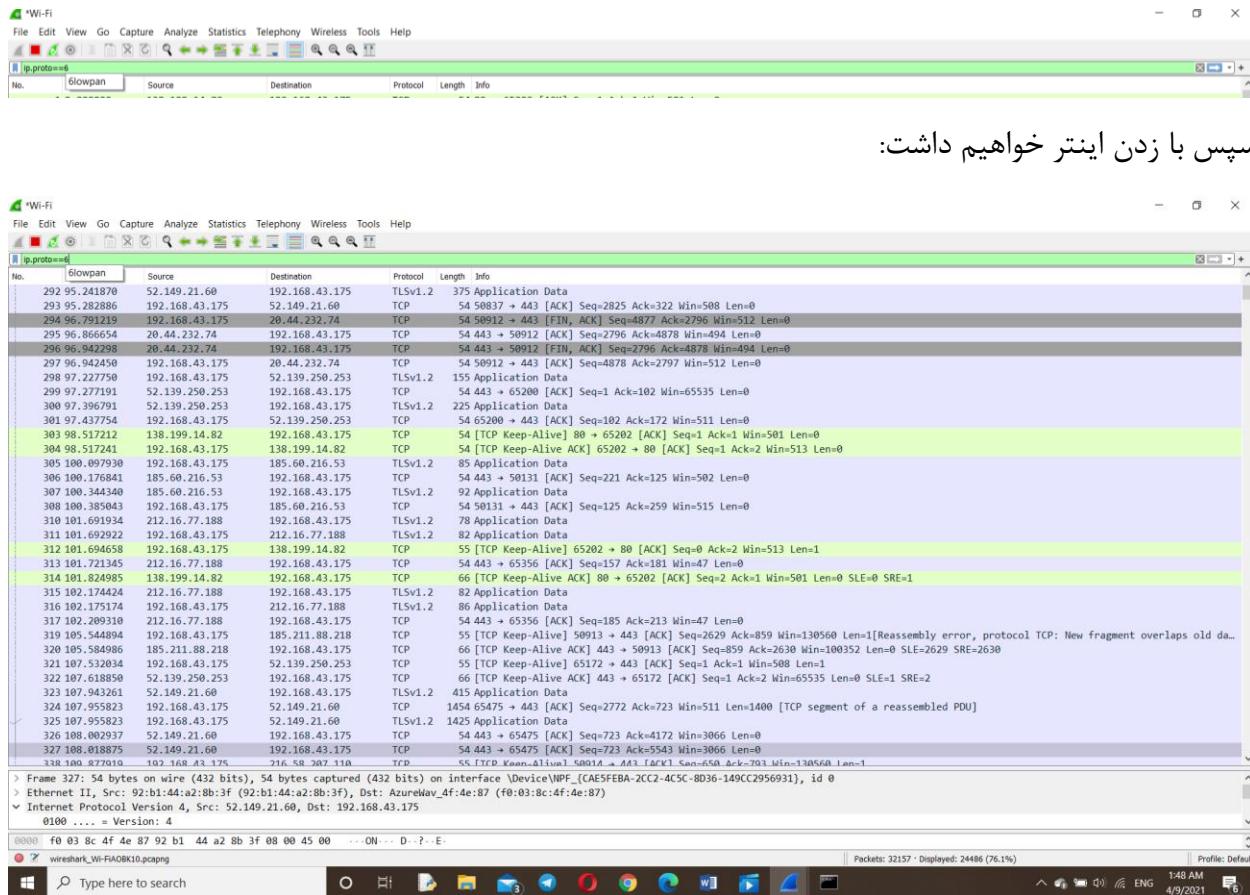
Time To Live (TTL) یا IP Protocol قرار می‌گیرد در واقع یک مقدار است که در بسته اطلاعاتی Internet Protocol (IP) می‌تواند باقی بماند و اگر به یک روتر شبکه می‌گوید که چه مدت زمان این بسته اطلاعاتی حق دارد در شبکه‌ی شما باقی بماند و اگر بیشتر از زمان مقرر بسته اطلاعاتی در شبکه ماند بسته توسط روتر منهدم می‌شود. به دلایل بسیار یک بسته اطلاعاتی ممکن است به مقصد مورد نظر در زمان مورد نظر نرسد. وقتی سیستمی بسته ای به یک سیستم دیگر ارسال می‌کند در ip header TTL ای برای آن تعیین می‌کند. نسبت به نوع سیستم عامل‌ها این عدد متفاوت است مثلاً در ویندوز ۱۰ بخواهید بسته icmp echo-request به یک سیستم در اینترنت بفرستید TTL بسته را ۶۴ قرار میدهد. حالا این بسته از هر روتری که در زیرساخت اینترنت عبور کنه از مقدار آن ۱ واحد کم می‌شود. اگر ttl بسته به صفر برسد و به مقصد نرسد آن بسته توسط روتر مربوطه drop می‌شود.

TTL صرفا برای جلوگیری از ایجاد loop در لایه سه استفاده می‌شود و باعث می‌شود بسته تا بی‌نهایت در اینترنت سرگردان نشود و نهایتاً ۲۵۵ hop را بتواند زنده بماند. یکی از استفاده‌هایی که از این TTL می‌شود مشخص کردن مسیر عبوری بسته هست که برای اینکار از دستور traceroute یا تو ویندوز tracert استفاده می‌شود. روش کار به این صورت است:

مقدار TTL بسته را برابر با ۱ قرار میدهد و به مقصد می‌فرستد، در اولین روتر چون مقدار این TTL به صفر کاهش پیدا می‌کند بسته drop می‌شود و روتر بسته icmp ttl exceeded را به فرستنده برمی‌گرداند. دفعه‌ی بعد مقدار TTL را ۲ قرار میدهد بسته در روتر دوم drop می‌شود. به همین ترتیب این TTL افزایش داده می‌شود تا بسته به مقصد برسد. این روش ممکن است در مواردی مثل غیر فعال بودن یا فیلتر بودن بسته‌های icmp یا پاسخ ICMP با IP جعلی (مثلاً آدرس یکی از اینترفیس‌های دیگر روتر مثل اینترفیس ۰ loopback) کارساز نباشد. حتی ممکن است بسته از زیرساخت mpls رد بشود و TTL را کاری کنند ثابت بماند و چندین روتر از نظر فرستنده پنهان بماند

به طور خلاصه می‌توان گفت: ابزار Tracert از مقدار TTL برای رسیدن و یا تست کردن مسیر ارتباطی مبدأ به مقصد استفاده می‌کند Tracert یک بسته اطلاعاتی با مقدار TTL کم در شبکه ارسال می‌کند و با رسیدن به هر روتر بسته با توجه به مقدار TTL ای که داشت از بین می‌رود و اطلاعات روتر مورد نظر برای مبدأ ارسال می‌شود. مدت زمانی که بین ارسال بسته و دریافت آن توسط پروتکل ICMP از مبدأ به مقصد اعلام می‌شود مدت زمان Hop Travel گفته می‌شود.

حال از بخش فیلتر، مقدار فیلتر را به دستور ip.proto==6 تغییر میدهیم.



## سوال 14: این فیلتر چه کاری انجام میدهد؟

در Internet Protocol ورژن 4 یک فیلد به نام protocol وجود دارد که پروتکل لایه‌ی بعدی را تعیین می‌کند. این فیلد 8 بیتی می‌باشد. هم چنین در Internet Protocol ورژن 6 به این فیلد Next Header می‌گویند.

حال در این فیلد شماره‌ی آن پروتکل نگهداری می‌شود. اگر دروایرشارک به لایه‌ی Network که پروتکل IP ورژن 4 دارد مراجعه کنیم می‌توانیم این فیلد را ببینیم که در آن نام یک پروتکل به همراه شماره‌ی آن نوشته شده است که به آن شماره Protocol Number می‌گویند و در آن قسمت از وایرشارک به صورت یک عدد دسیمال که در محدوده‌ی 0 تا 255 است نشان داده می‌شود.

برای مثال اگر بسته‌ی select شده زیر را انتخاب کنیم و به قسمت آن برویم خواهیم داشت:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	138.199.14.82	192.168.43.175	TCP	54	80 → 65202 [ACK] Seq=1 Ack=1 Win=501 Len=0
2	0.000101	192.168.43.175	138.199.14.82	TCP	54	[TCP ACKED unseen segment] 65202 → 80 [ACK] Seq=1 Ack=2 Win=513 Len=0
3	0.099206	192.168.43.175	185.60.216.53	TLSv1.2	85	Application Data
4	0.161855	185.60.216.53	192.168.43.175	TCP	54	443 → 50131 [ACK] Seq=1 Ack=32 Win=502 Len=0
5	0.328459	185.60.216.53	192.168.43.175	TLSv1.2	92	Application Data

Wireshark - Packet 4 - Wi-Fi	
> Frame 4: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface \Device\NPF_{CAE5FEB4-2CC2-4C5C-BD36-149CC2956931}, id 0	
> Ethernet II, Src: 92:b1:44:a2:8b:3f (92:b1:44:a2:8b:3f), Dst: AzureWave_4f:4e:87 (f0:03:8c:4f:4e:87)	
▼ Internet Protocol Version 4, Src: 185.60.216.53, Dst: 192.168.43.175	
0100 .... = Version: 4	
.... 0101 = Header Length: 20 bytes (5)	
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)	
Total Length: 40	
Identification: 0x0d1a (54234)	
Flags: 0x00, Don't fragment	
Fragment Offset: 0	
Time to Live: 124	
Protocol: TCP (6)	
Header Checksum: 0xad2b [validation disabled]	
[Header checksum status: Unverified]	
Source Address: 185.60.216.53	
Destination Address: 192.168.43.175	
Transmission Control Protocol, Src Port: 443, Dst Port: 50131, Seq: 1, Ack: 32, Len: 0	
0000 f0 03 8c 4f 4e 87 92 b1 44 a2 8b 3f 08 00 45 00 .. ON .. D .. ? .. E ..	
0010 00 28 d3 da 40 00 7c 06 ad 2b b9 3c d8 35 c0 a8 ..(.-@.1) ..+..<.5..	
0020 2b af 01 bb c3 d3 8f bf 1e 6d be 91 49 00 50 10 ..+..... m..1 P ..	
0030 01 f6 b4 be 00 00 ..	

همانطور که از تصویر هم می‌توان فهمید این فیلتر همه‌ی بسته‌های شنود شده‌ای که شماره‌ی پروتکل لایه‌ی آن ها برابر با 6 می‌باشد را فیلتر و جدا می‌کند. همچنین توجه داشته باشید که شماره‌ی 6 شماره‌ی Transport پروتکل مربوط به پروتکل TCP می‌باشد. پس درواقع آن بسته‌هایی که پروتکل لایه‌ی آن ها TCP می‌باشد را جدا می‌کند.