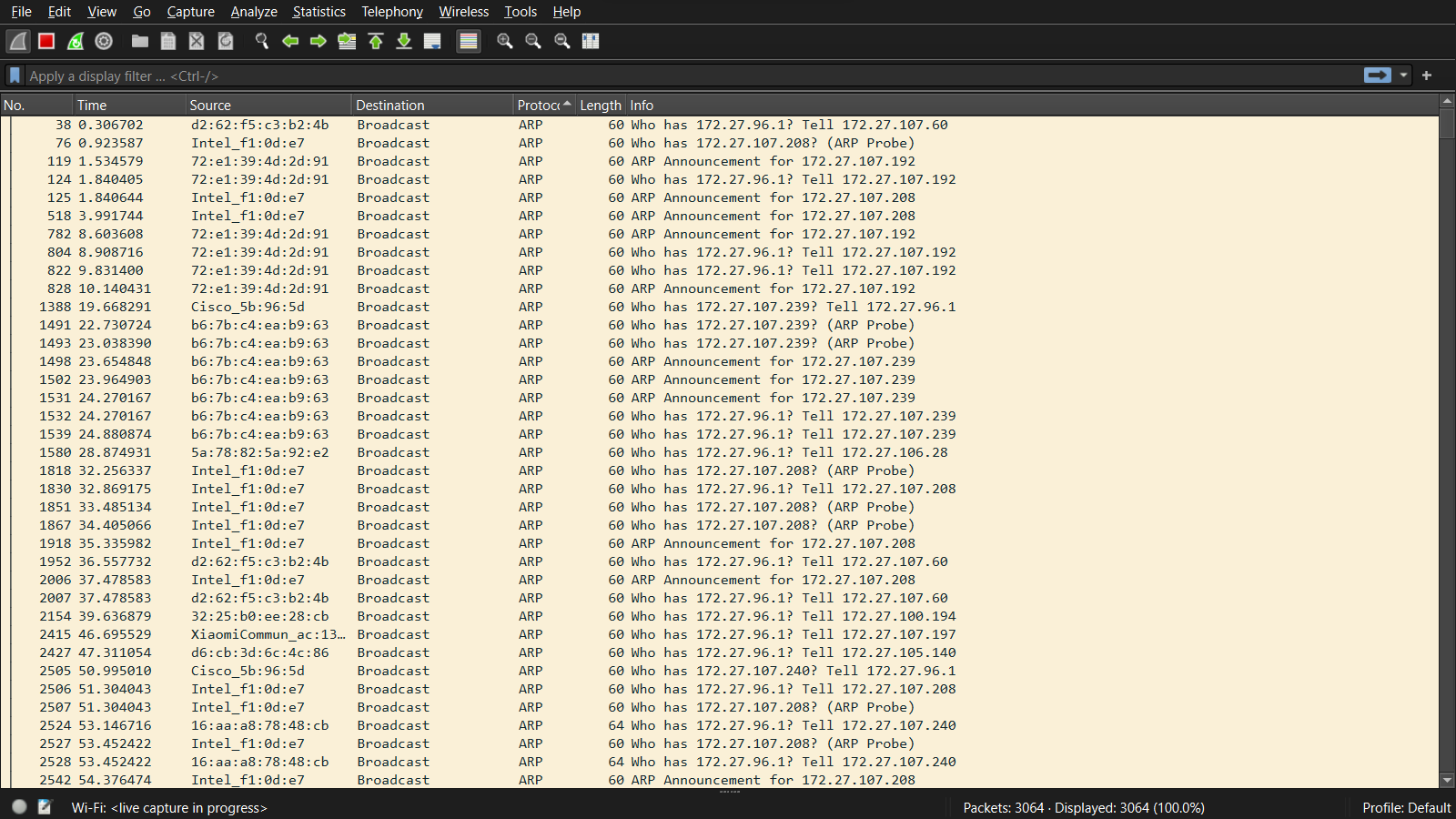
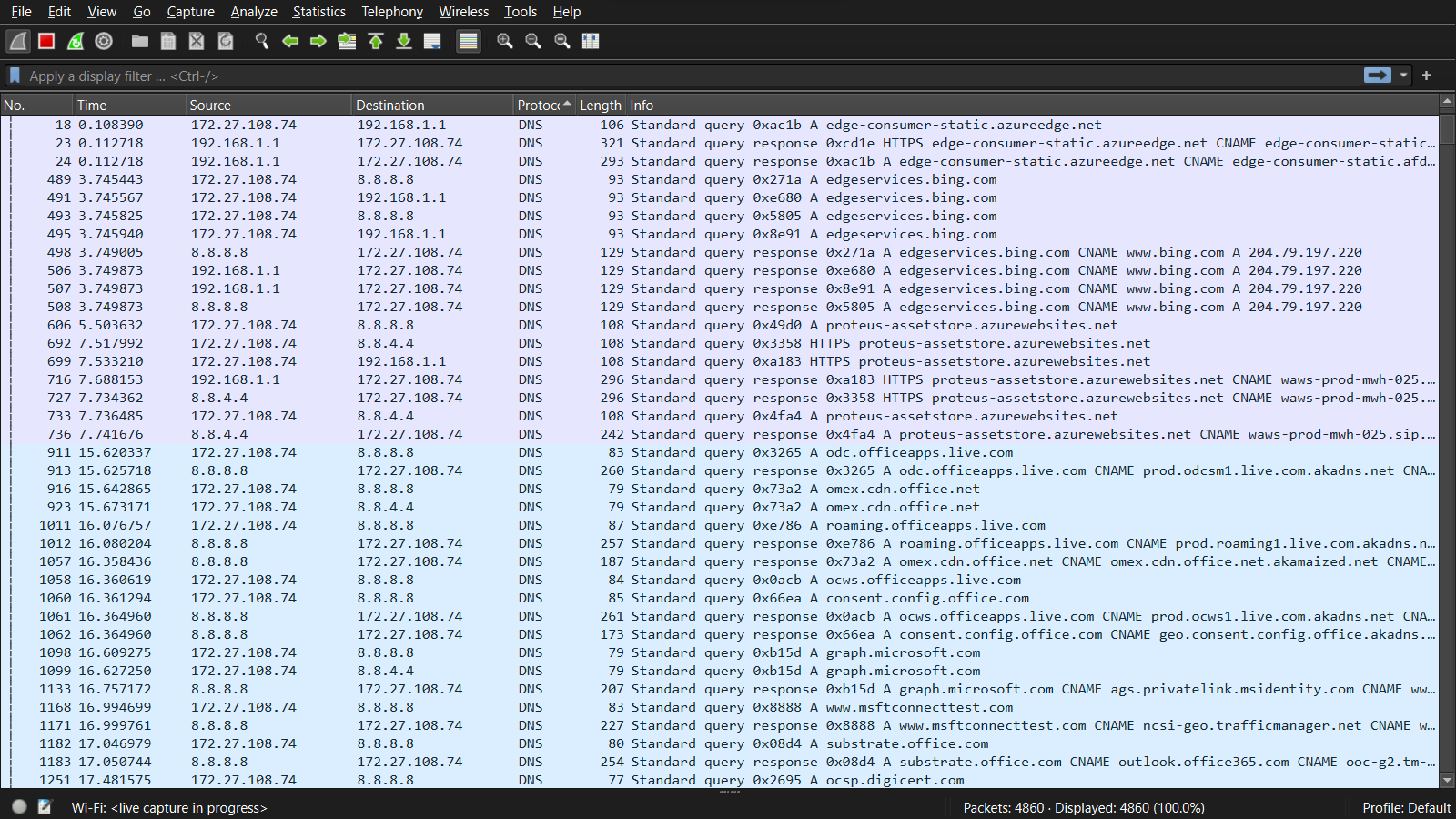
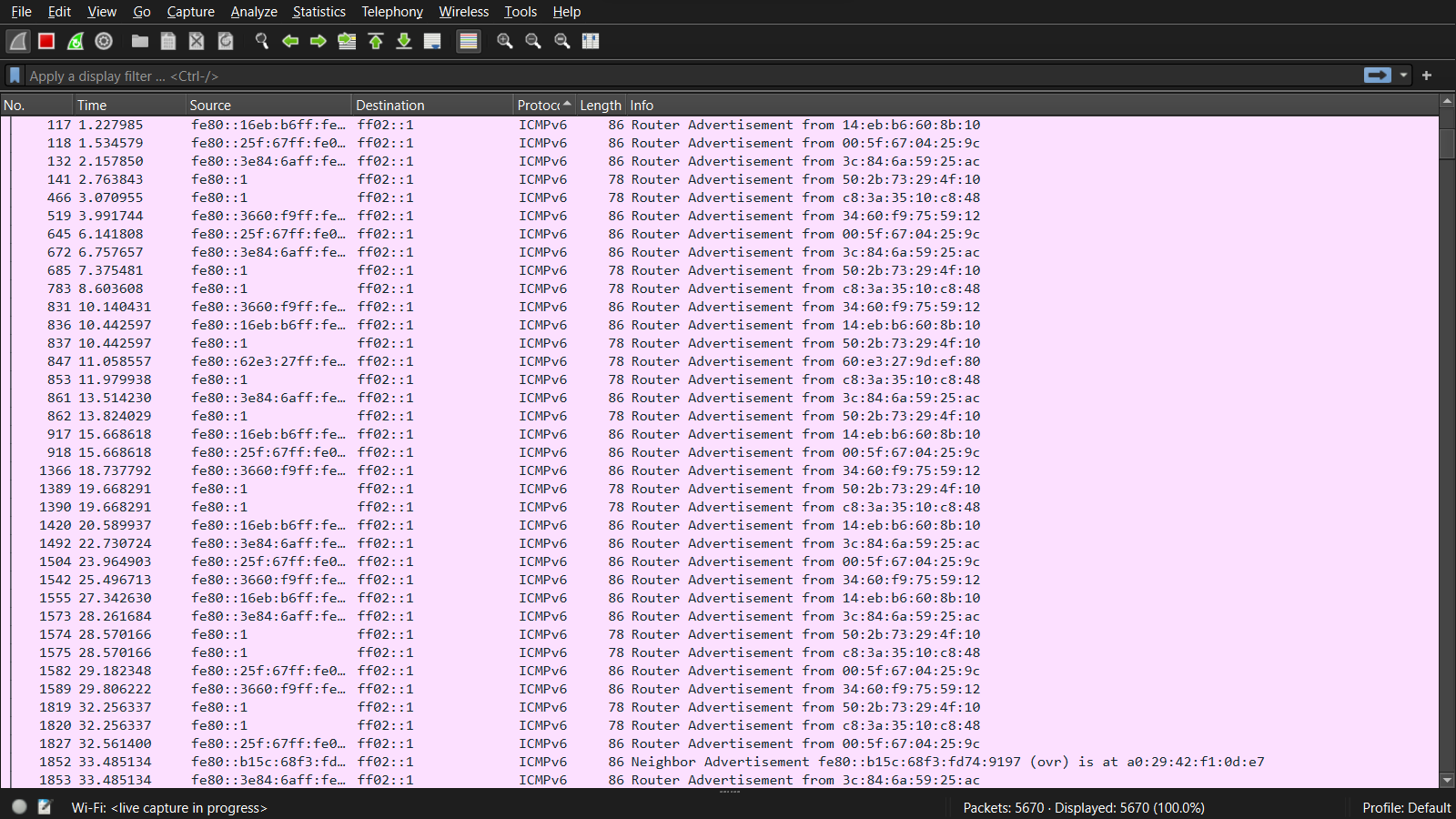
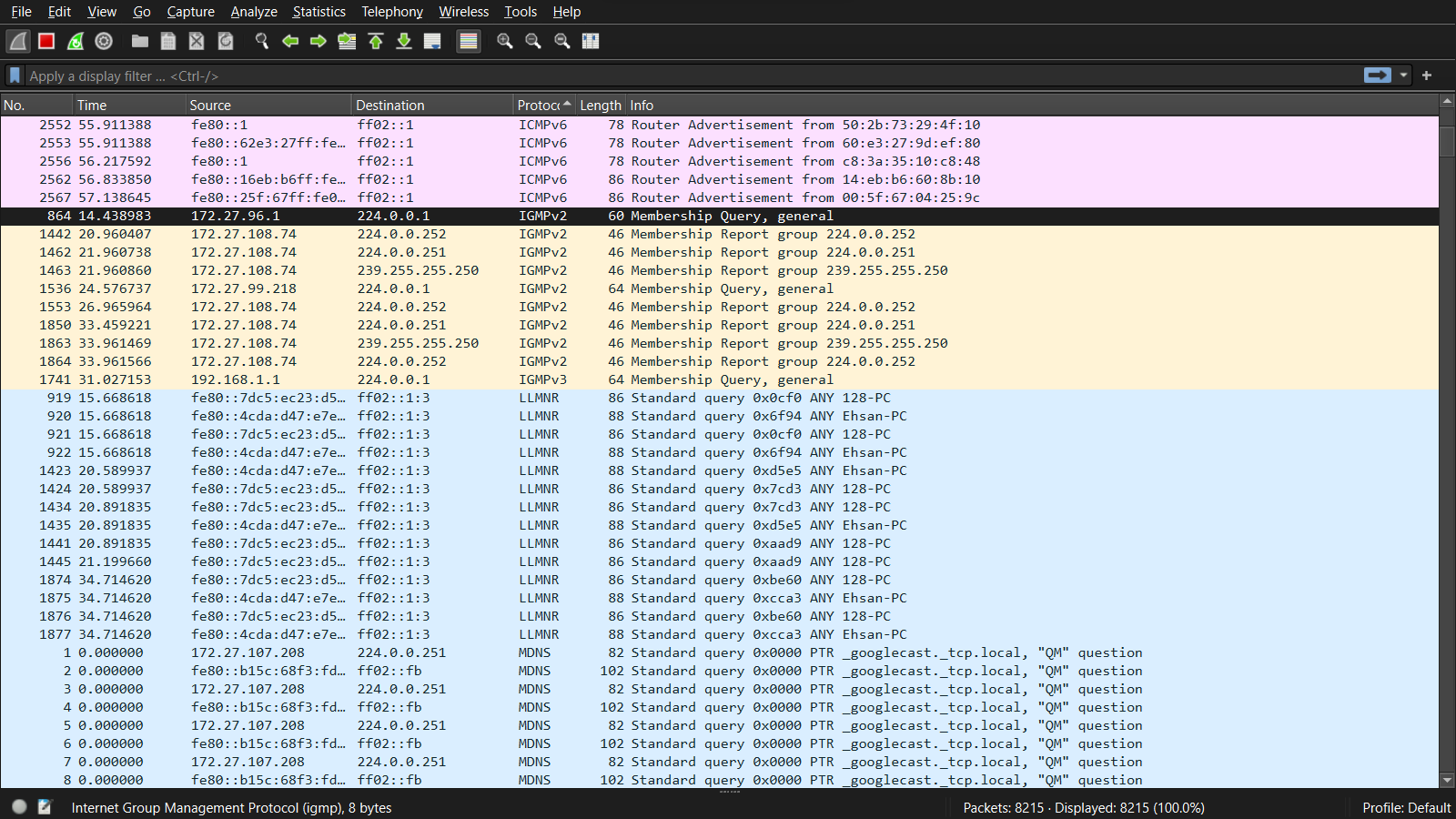
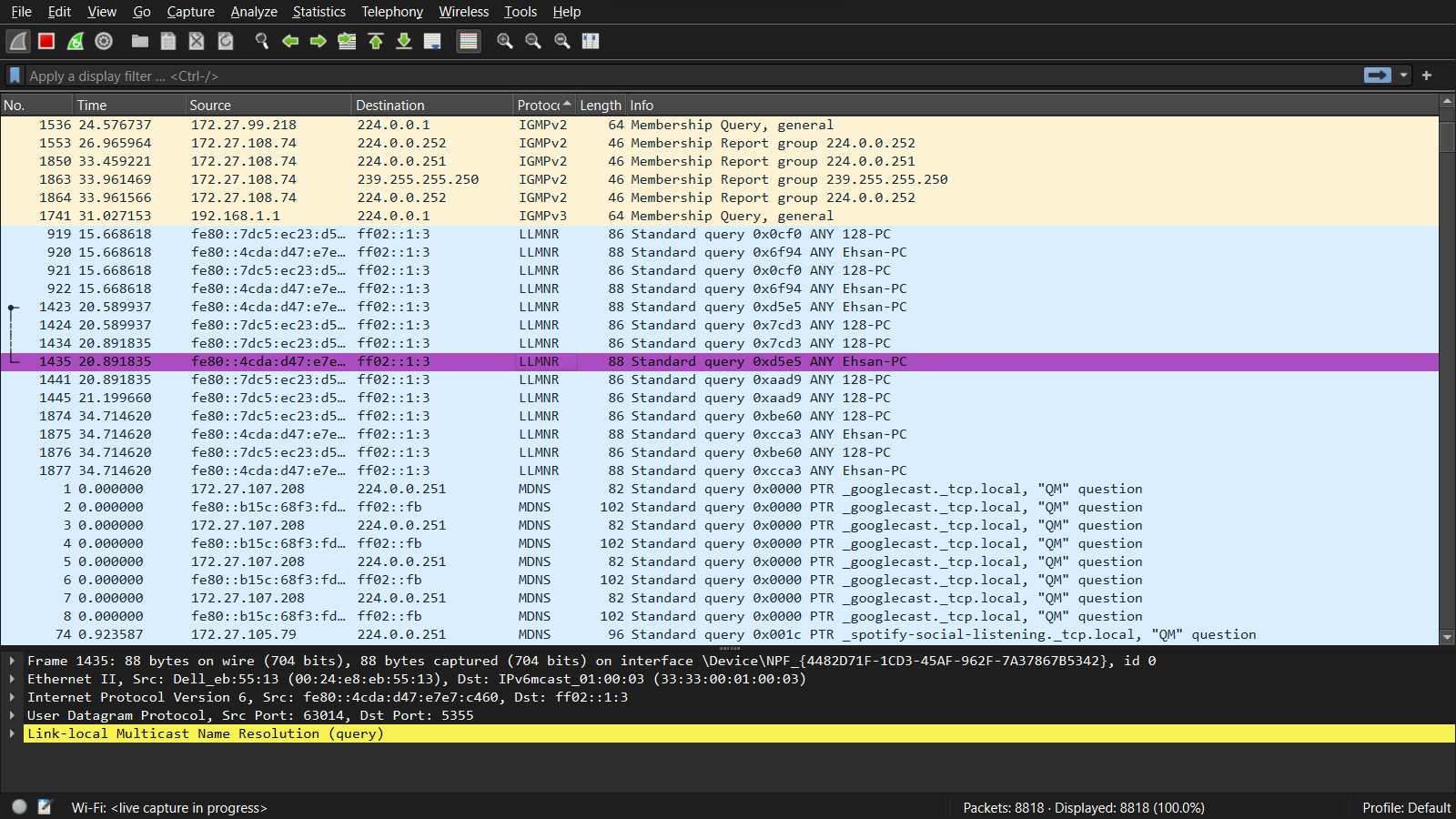
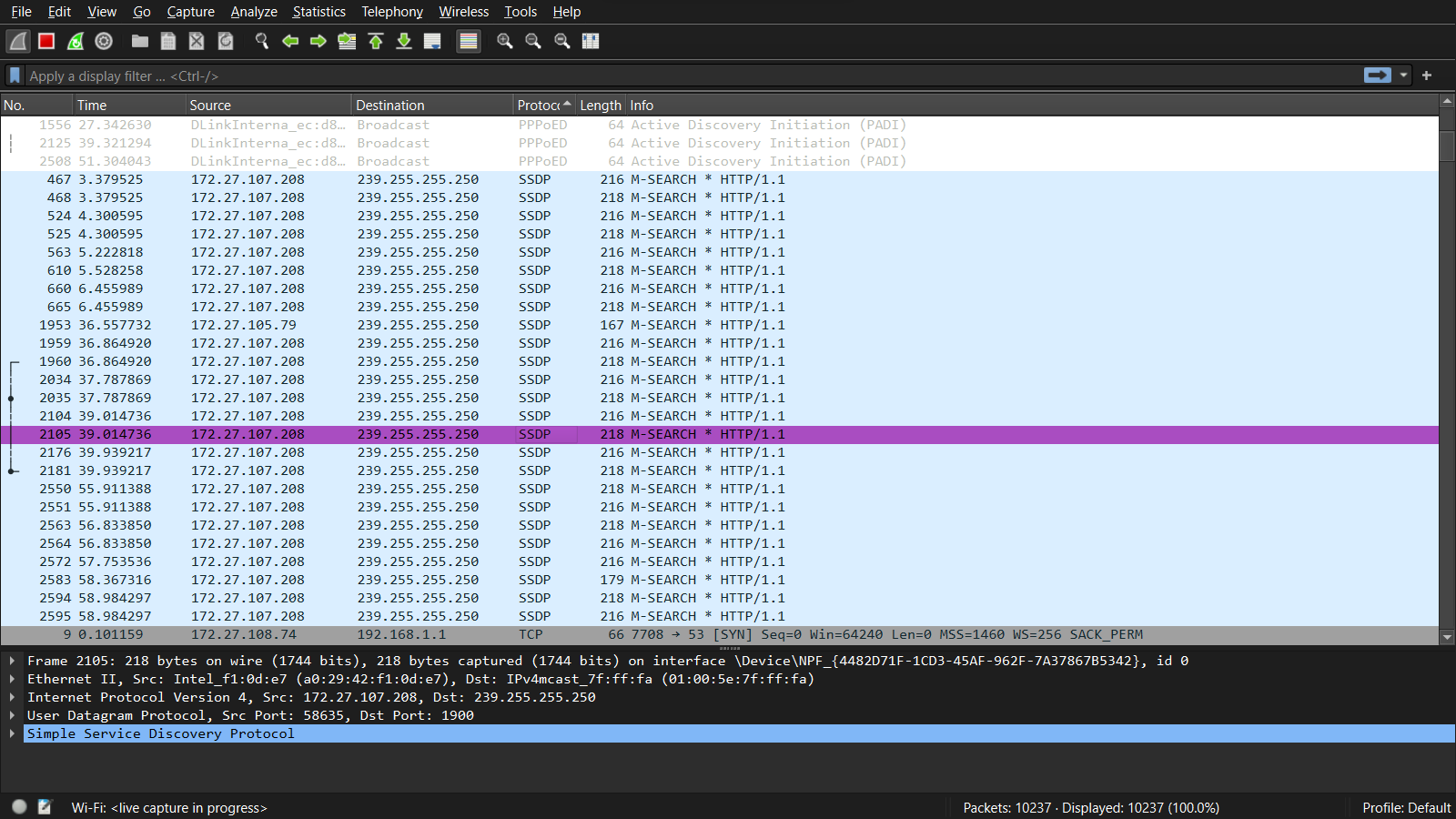
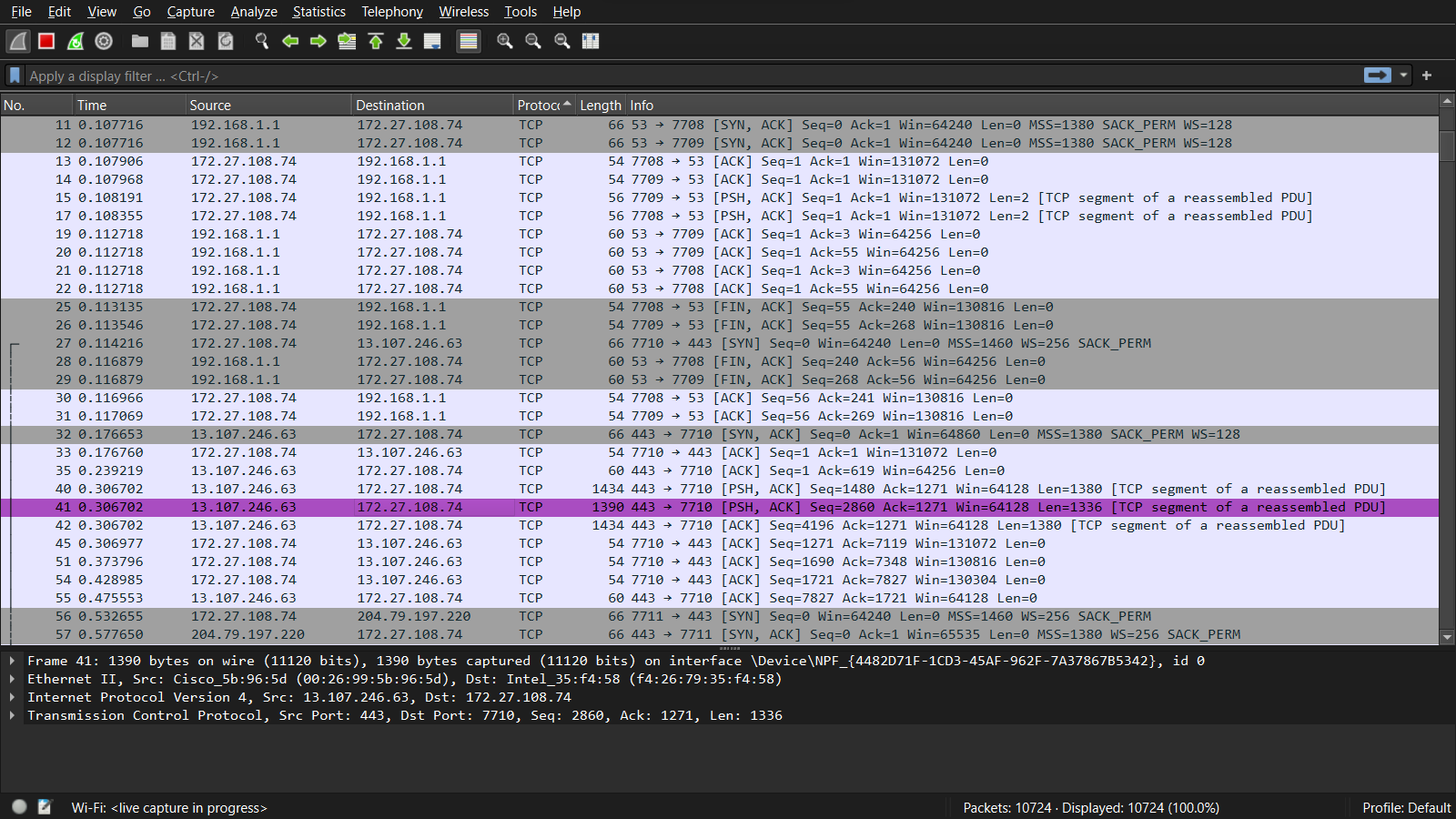
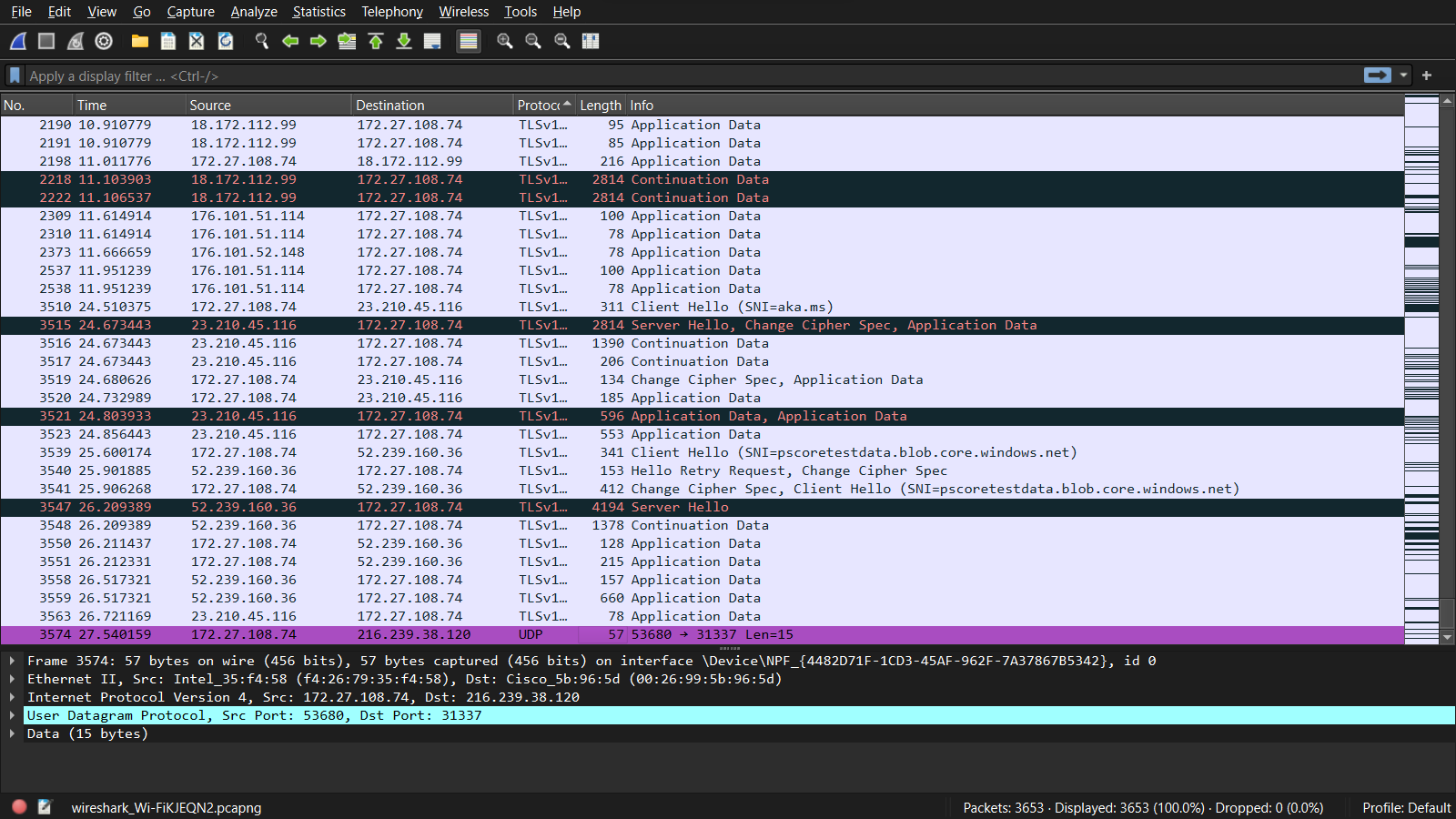
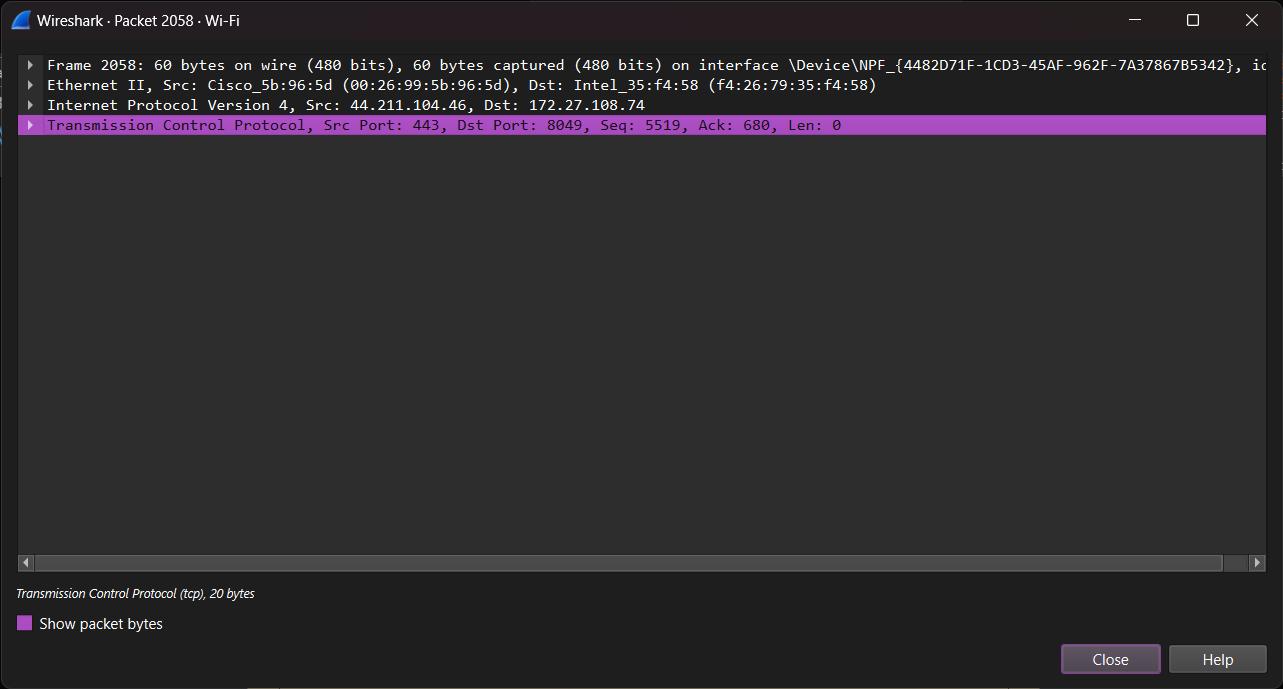
1. * 1. Address Resolution Protocol (ARP Announcement):
     2. Domain Name System (response)(DNS):
     3. Internet Control Message Protocol v6 (ICMPv6):
     4. Internet Group Management Protocol:

* + 1. Link-local Multicast Name Resolution:
    2. Simple Service Discovery Protocol:

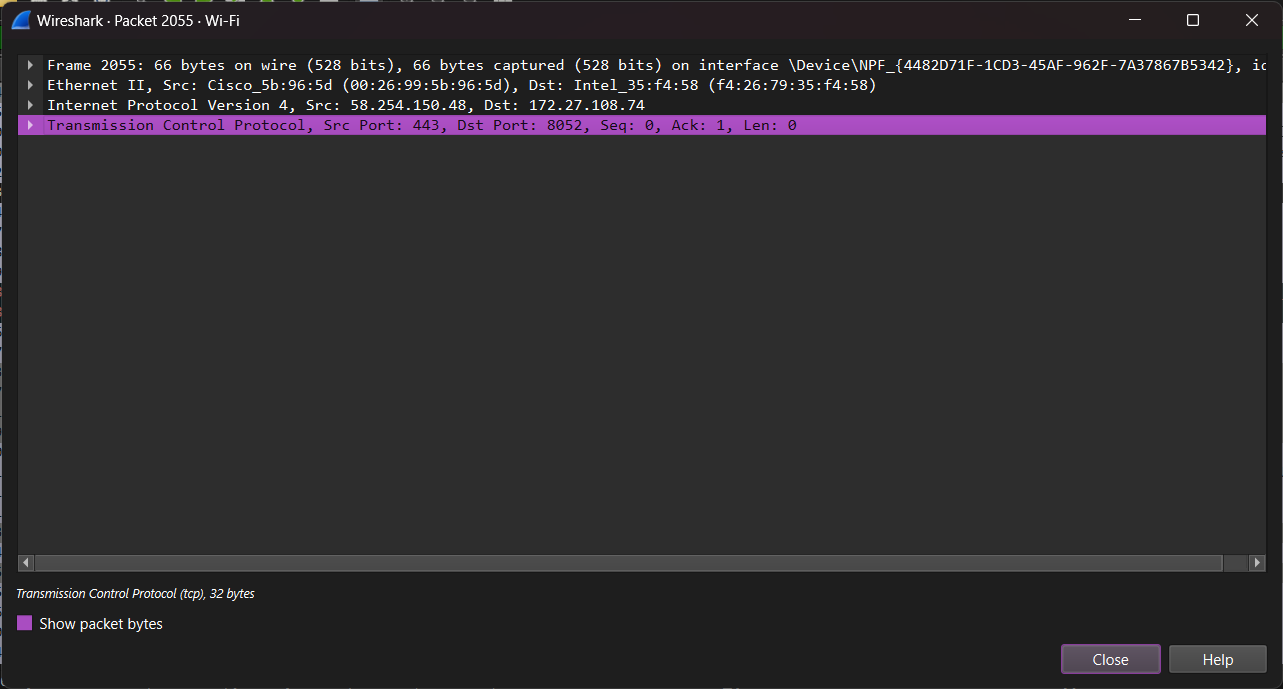
* + 1. Transmission Control Protocol (TCP):
    2. User Datagram Protocol (UDP):

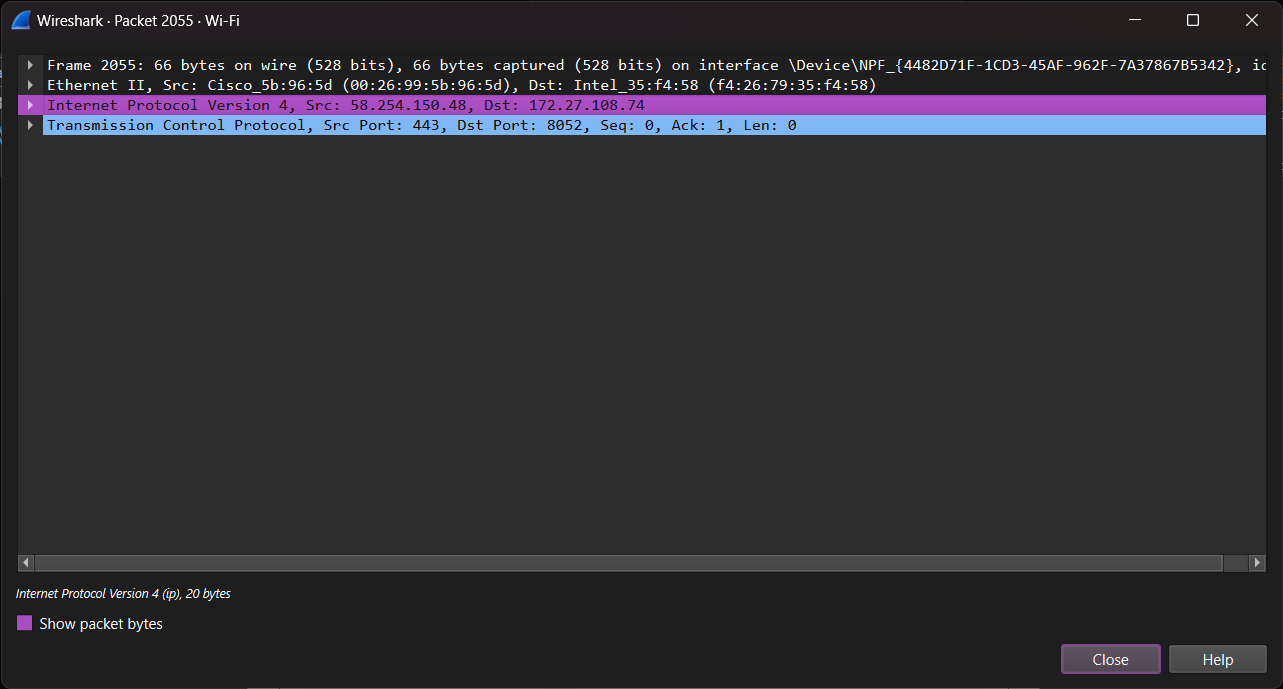
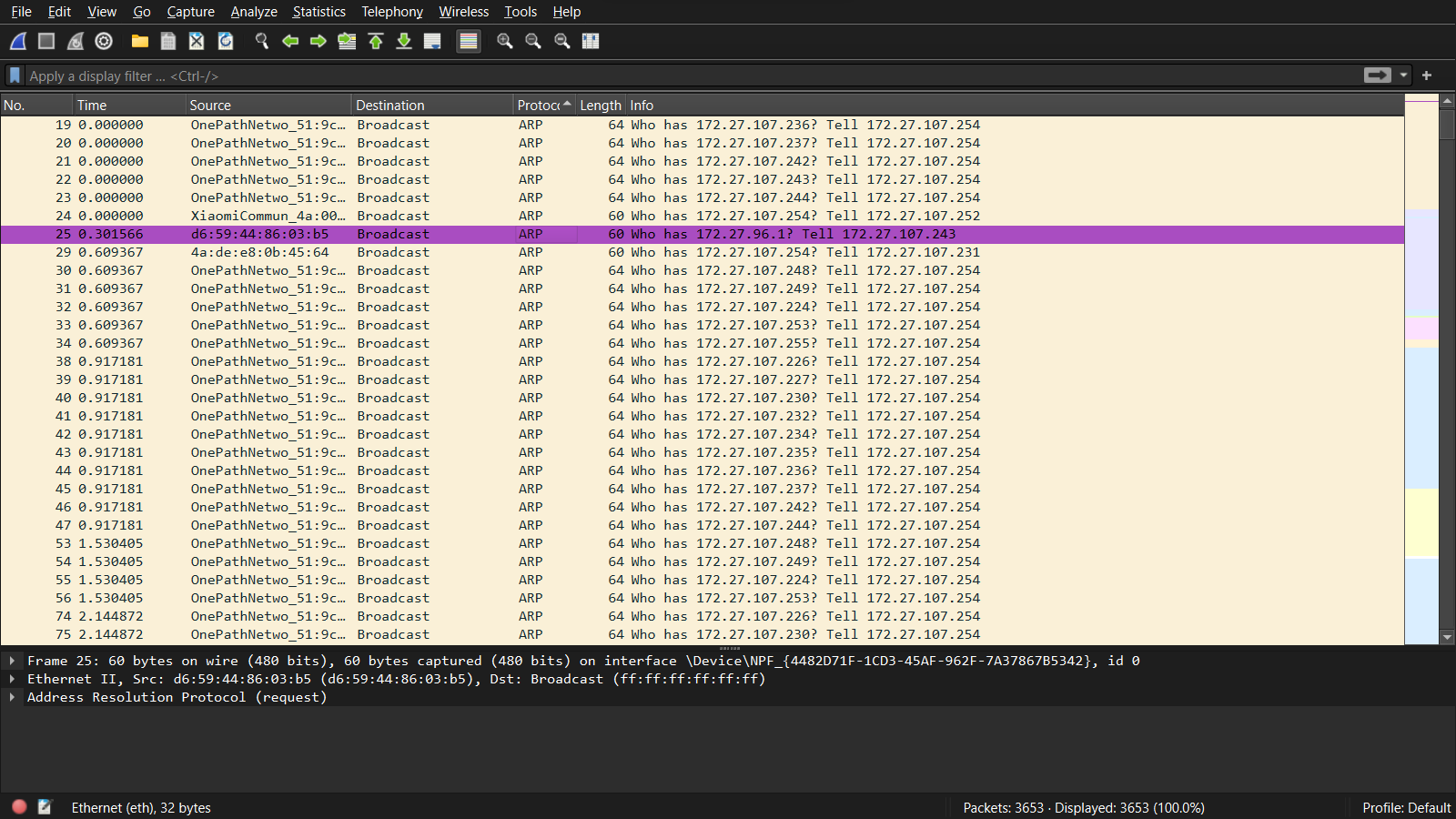


* + 1. Physical Layer: Bit
    2. Link Layer: Ethernet 2
    3. Network Layer: Internet Protocol Version 4 (IPV4)
    4. Transport Layer: Transmission Control Protocol (TCP)

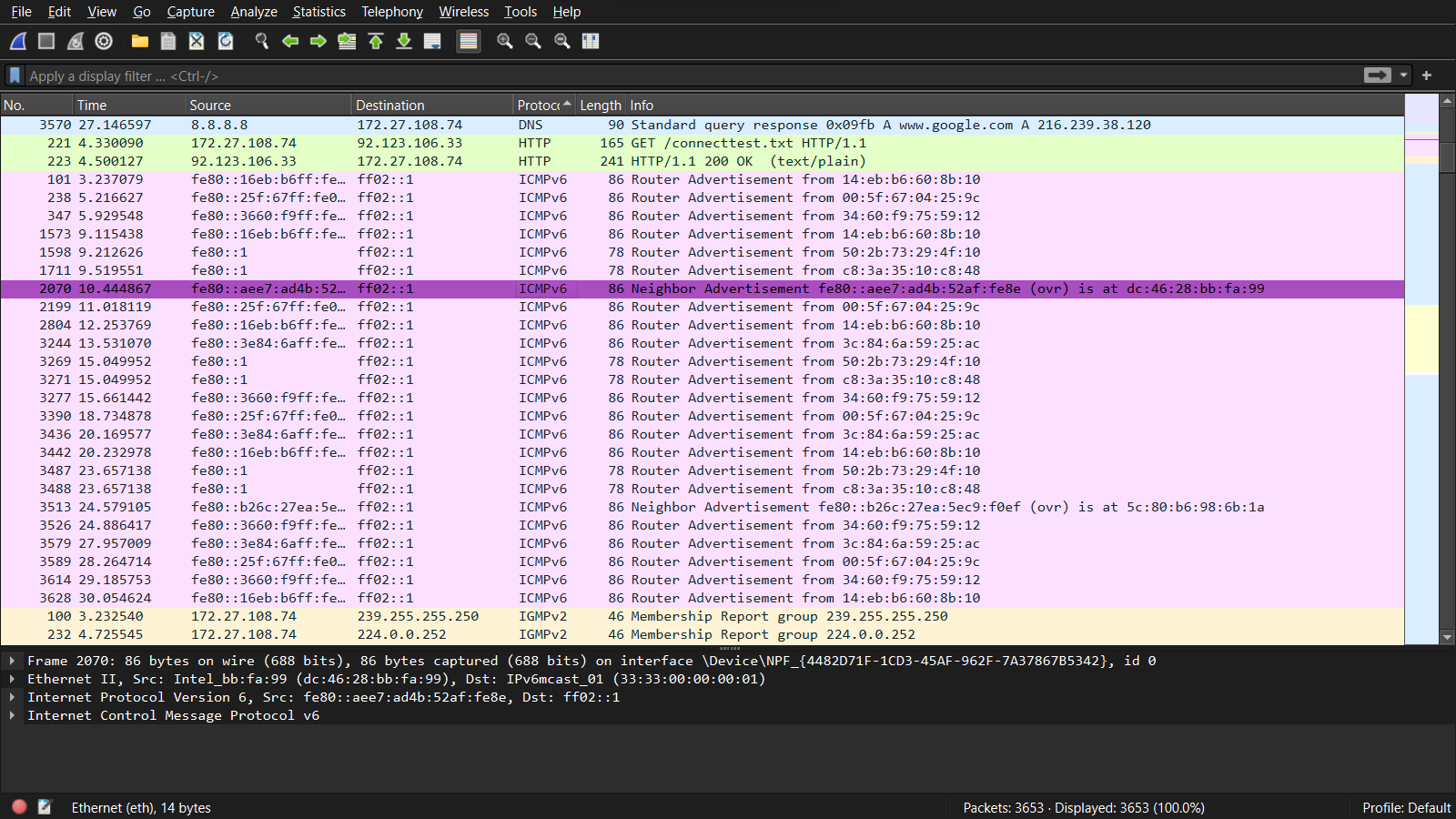
کل بیت ها بوسیله لایه فیزیکی انتقال پیدا می کنند و ابتدای بیت ها مربوط به link layer است ،بعد network layer ،بعد transport layer و در آخر هم بیت های مربوط به application layer قرار می گیرند.

* 1. 32 bytes یا 256 bits

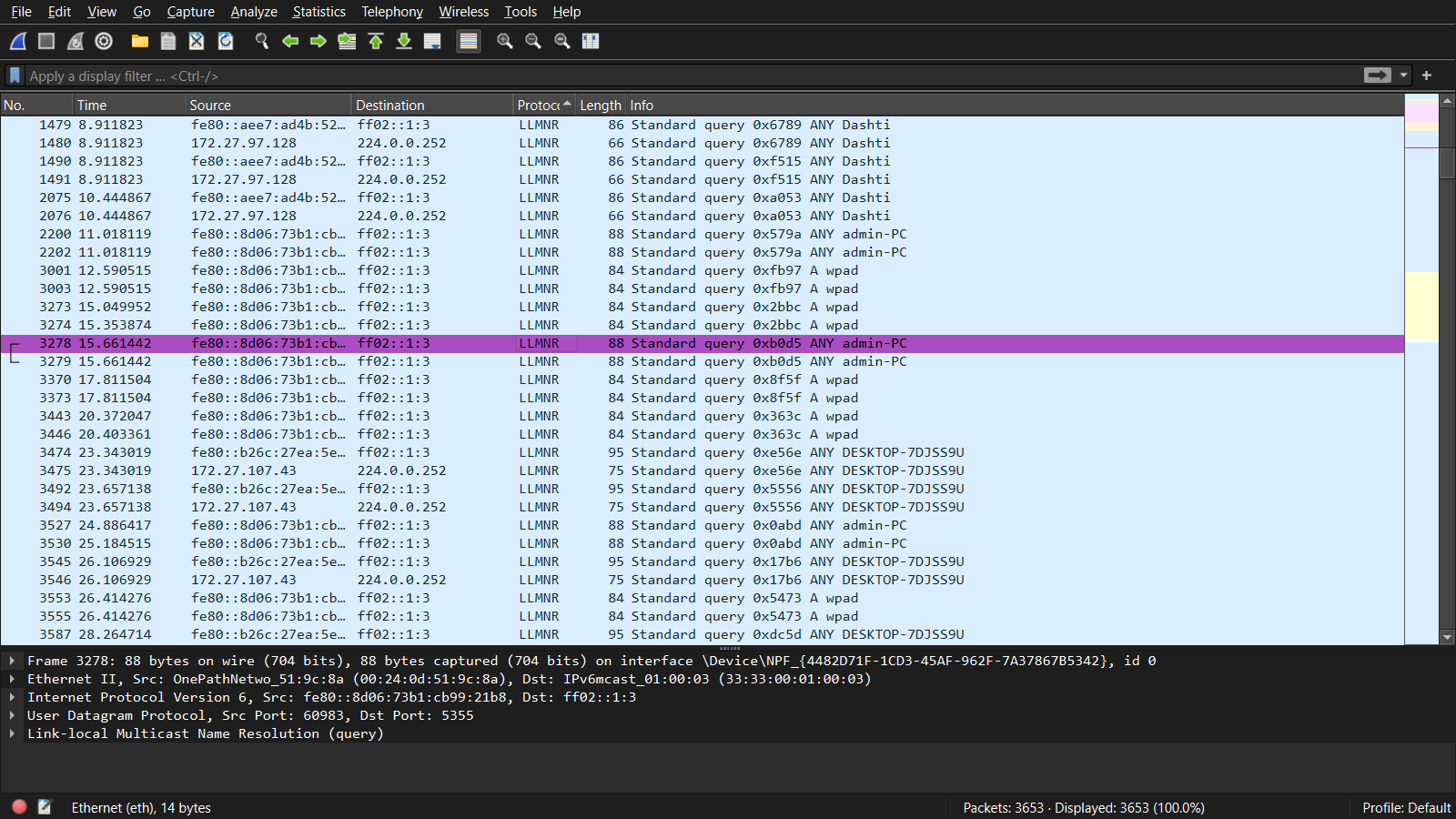


* 1.  20 bytes یا 160 bits
  2. بله –
  3. 1. ARP:

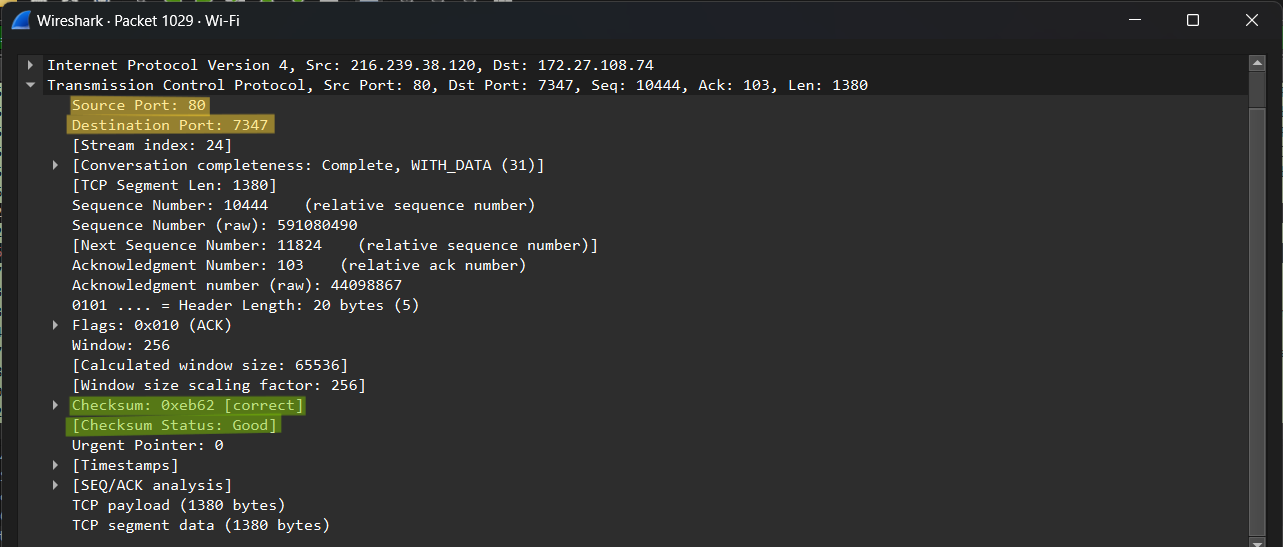
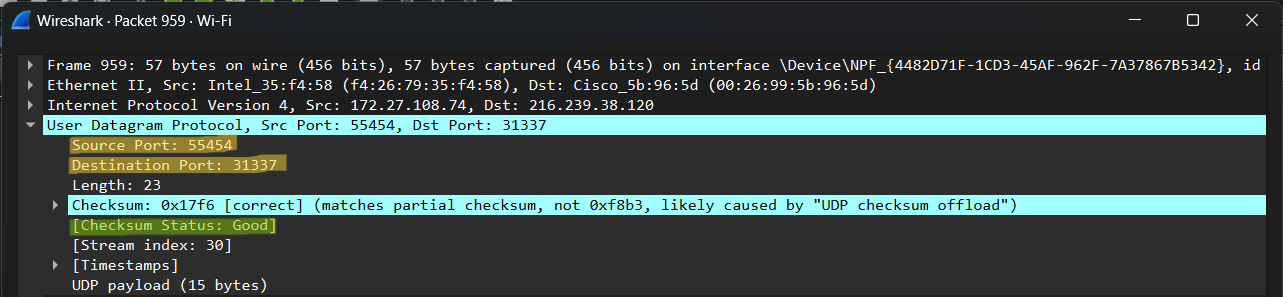
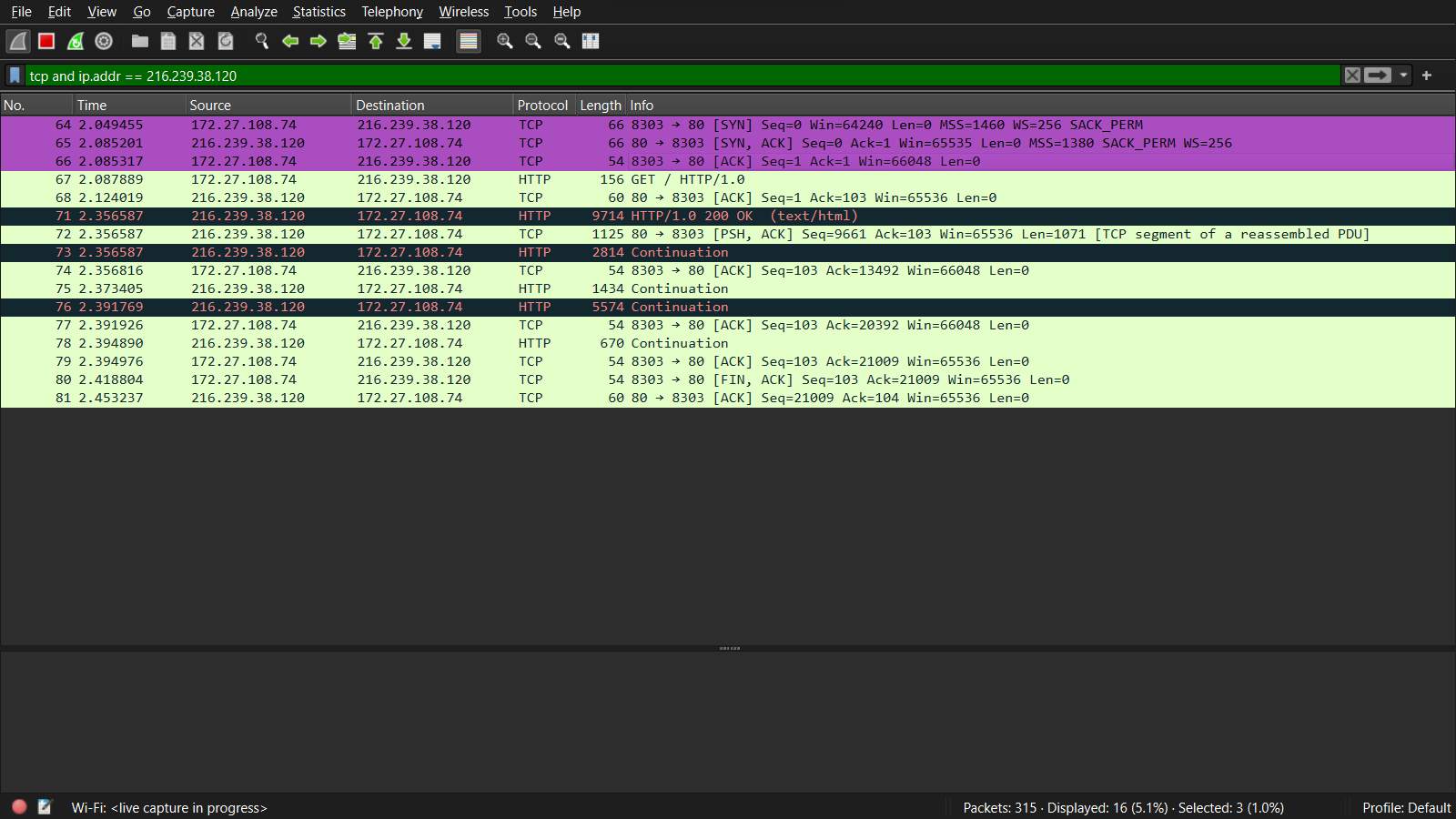
فاقد لایه های application layer ، transport layer و network layer است.

* + 1. ICMPv6, IGMPv2: 

فاقد لایه های application layer و transport layer است.

* + 1. Link-local Multicast Name Resolution, Multicast Domain Name System: 

فاقد لایه ی application layer است.

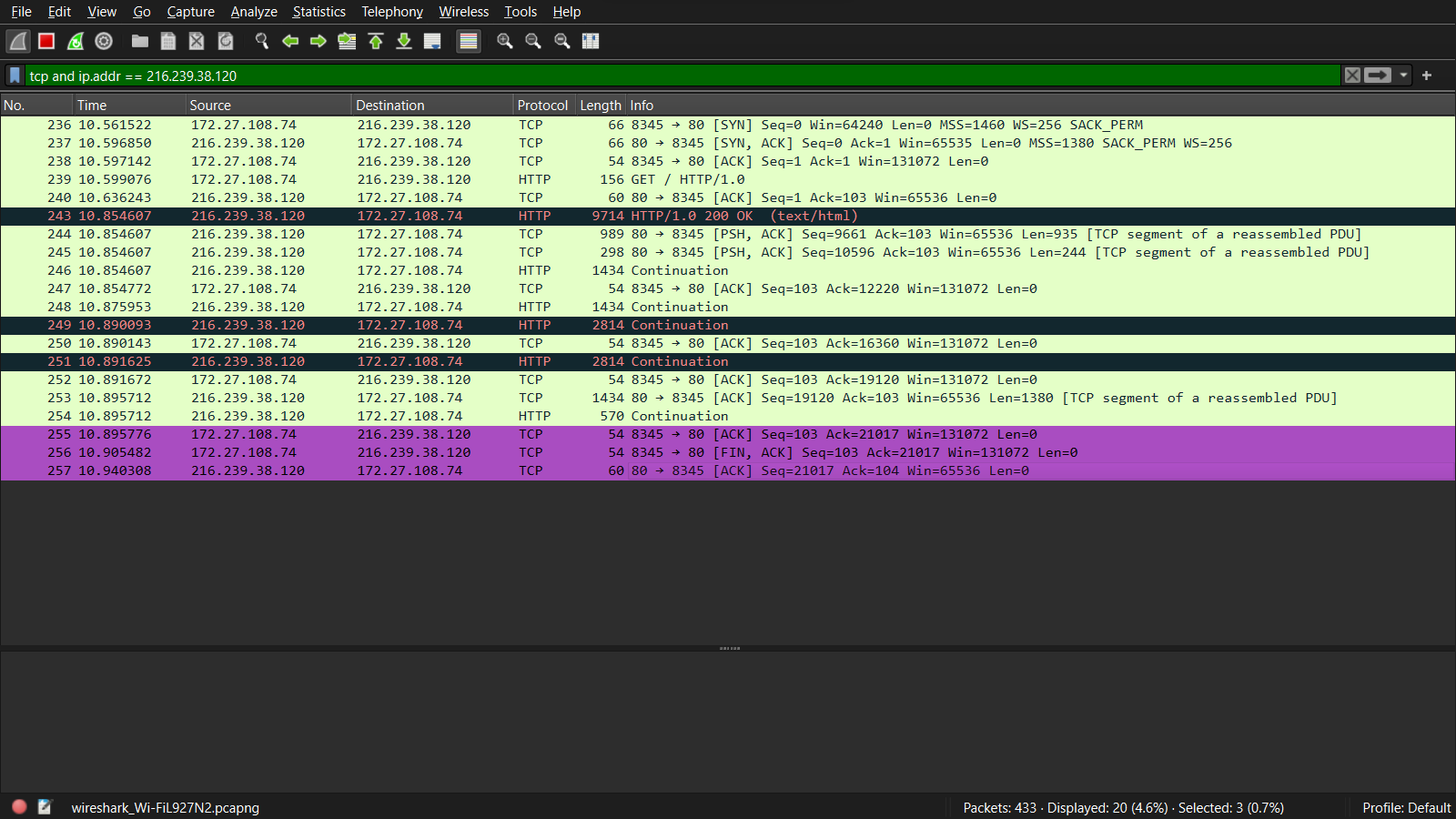
1. 1. TCP:
   2. UDP:
   3. Handshake: 
      1. Client 🡪Server: Seq number: 0, Ack number: 0, Window size: 64240, tcp.flags.syn = 1, tcp.flags.ack = 0, tcp.flags.fin = 0;
      2. Server🡪Client: Seq number: 0, Ack number: 1, Window size: 65535, tcp.flags.syn = 1,

tcp.flags.ack = 1, tcp.flags.fin = 0;

* + 1. Client 🡪Server: Seq number: 1, Ack number: 1, Window size: 512 tcp.flags.syn = 0,

tcp.flags.ack = 1, tcp.flags.fin = 0;

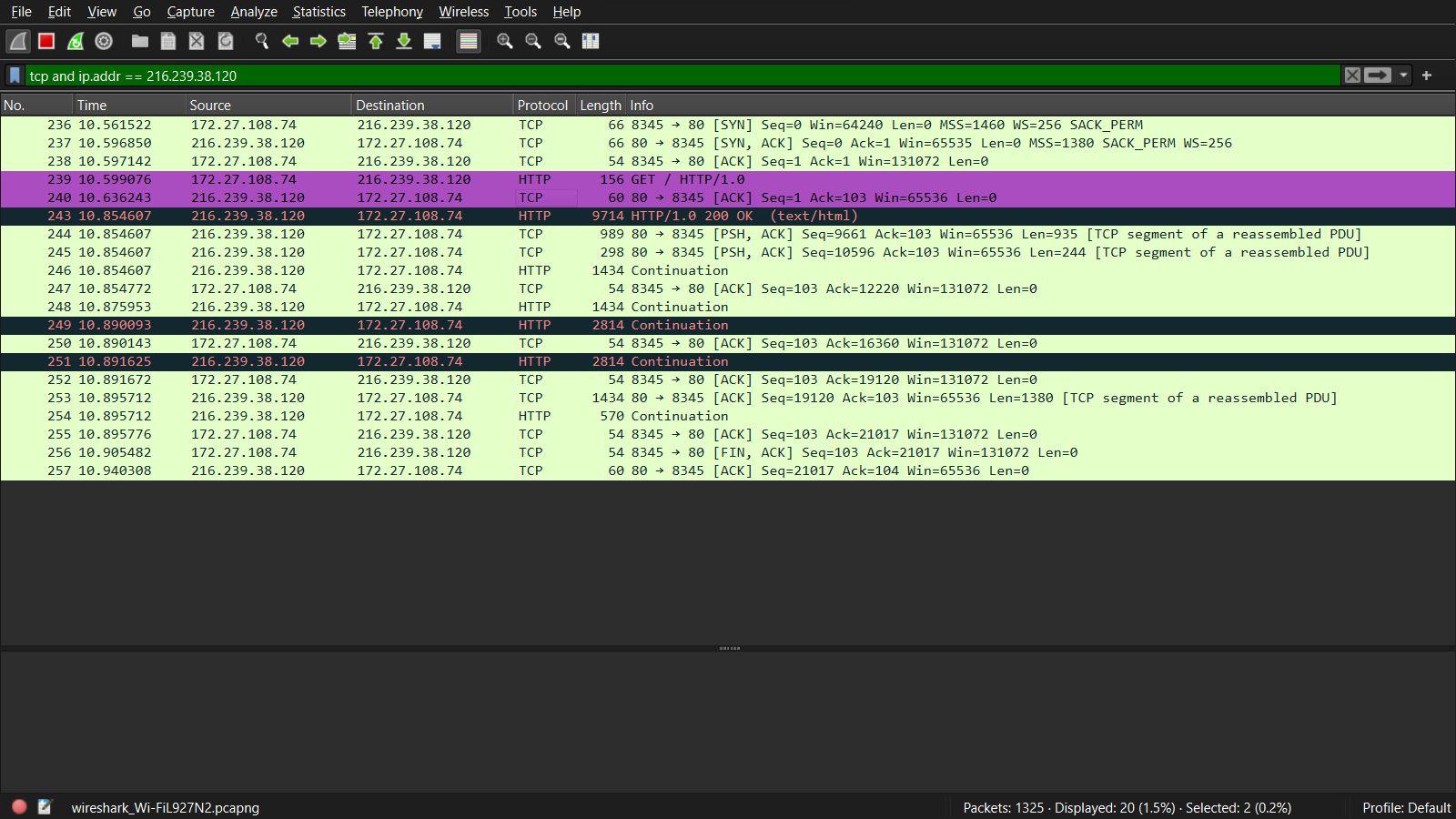
ابتدا کلاینت به سرور درخواست سینک شدن می فرستد با فعال کردن tcp.flags.syn بعد سرور درخواست کلاینت را با قبول کردن شماره سکونس می پزیرد و بیت های مریوط به tcp.flags.ack و tcp.flags.syn را یک می کنه و برای کلاینت ارسال می کند و کلاینت هم با قبول کردن سکونس نامبر قبلی درخاستش را پاسخ می دهد.

* 1. Connection Closer: 
     1. Client 🡪Server: Seq number = 20104, Ack number = 103, Window size = 256,

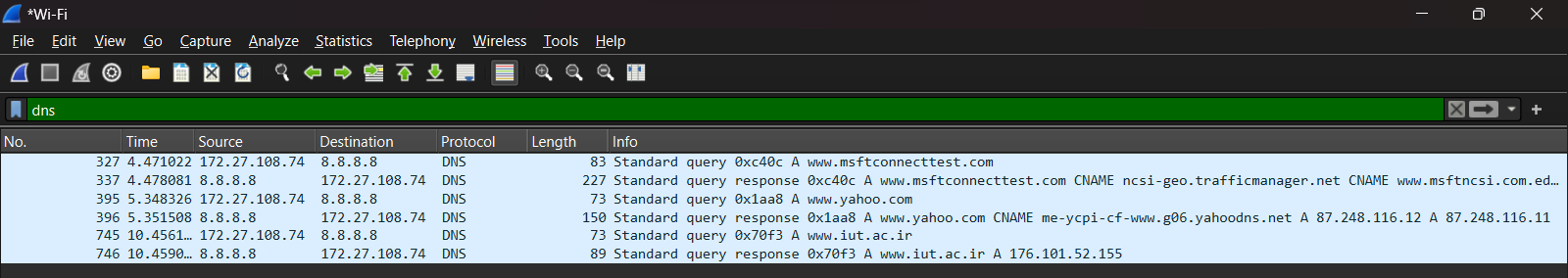
tcp.flags.syn = 0, tcp.flags.ack = 1, tcp.flags.fin = 1;

* + 1. Server🡪Client: Seq number: 103, Ack number: 21068, Window size: 508, Tcp.flags.syn = 0, Tcp.flags.ack = 1, tcp.flags.fin = 0;
    2. Server🡪Client: Seq number: 103, Ack number: 21068, Window size: 508, Tcp.flags.syn = 0, Tcp.flags.ack = 1, tcp.flags.fin = 1;
    3. Client 🡪Server: Seq number: 21068, Ack number: 104, Window size: 256, Tcp.flags.syn = 0, Tcp.flags.ack = 1, tcp.flags.fin = 0;

ابتدا کلاینت درخواست fin را به سرور می دهد و منتظر پاسخ سرور می ایستد بعد سرو fin و ack قبلی را در جواب کلاینت پشت سر هم می فرستد و کلاینت هم با ack کردن درخواست سرور ارتباط را می بندد.

* 1. Connection Duration: 
     1. Client 🡪Server: Seq number: 1, Ack number: 1, Window size: 512, Tcp.flags.syn = 0, Tcp.flags.ack = 1, tcp.flags.fin = 0;
     2. Server🡪Client: Seq number: 1, Ack number: 103, Window size: 256, Tcp.flags.syn = 0, Tcp.flags.ack = 1, tcp.flags.fin = 0;

کلاینت درخواست خودش را با سکونس نامبری به سمت سرور ارسال می کند و سرور با اک نامبر سکونس بعدی و سکونس نامبر خودش به درخواست کلاینت پاسخ می دهد.

1. 

Response time [www.yahoo.com](http://www.yahoo.com): [Time: 0.003182000 seconds];

Response time [www.iut.ac.ir](http://www.iut.ac.ir): [Time: 0.002912000 seconds]

Protocol Hierarchy:

