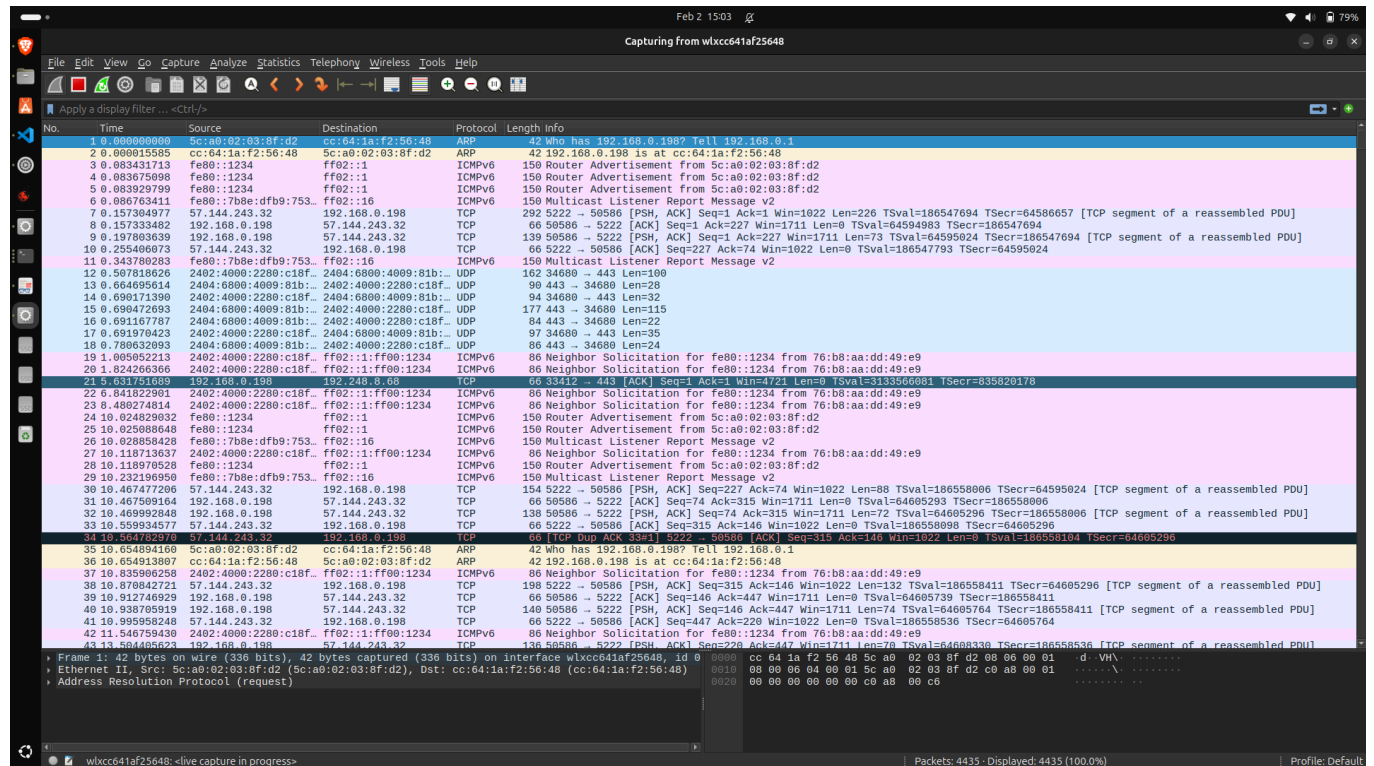


230415H
Mudaliarachchi N.S
Take Home Assignment
02/02/2026

1. Capture of Some PDUs



2. Five protocols and its functions

Protocol	Name	Function
ARP	Address Resolution Protocol	Translates logical ip address(Layer 3) into physical mac address(Layer 2)
ICMPv6	Internet control message protocol version 6	Error reporting and network diagnostics, handles Neighbour discovery, Stateless address configuration.
TCP	Transmission control protocol	Ensure reliable, ordered and

		error checked delivery of data over networks(Braking data in to packets, number them and reassembling them correctly at the destination)
UDP	User Datagram Protocol	Sends data independently packets(Data Grams) without establishing a connection(Fast and Connection less)
DNS	Domain Name System	Acts as an Internet phone book. Translating human readable domain names into machine readable IP addresses.

3. Application Layer Protocol : HTTP

The image shows a Wireshark network traffic capture. The top pane displays a list of network packets. The middle pane shows the details of the selected packet (No. 1000), and the bottom pane shows the raw packet data in hexadecimal and ASCII.

Packet List:

No.	Time	Source	Destination	Protocol	Length	Info
6989	268.392866242	192.168.0.198	91.189.91.48	HTTP	154	GET / HTTP/1.1
6992	268.656632555	91.189.91.48	192.168.0.198	HTTP	255	HTTP/1.1 204 No Content
7034	278.333773712	2402:4000:2280:c18f...	2620:2d:4000:1::96	HTTP	174	GET / HTTP/1.1
13960	568.400592899	192.168.0.198	91.189.91.48	HTTP	154	GET / HTTP/1.1
13961	568.664061358	91.189.91.48	192.168.0.198	HTTP	255	HTTP/1.1 204 No Content
13998	578.342574736	2402:4000:2280:c18f...	2620:2d:4000:1::96	HTTP	174	GET / HTTP/1.1
20456	868.289237103	192.168.0.198	185.125.190.17	HTTP	154	GET / HTTP/1.1
20561	894.849706649	2402:4000:2280:c18f...	2620:2d:4000:1::197	HTTP	174	GET / HTTP/1.1
20567	895.440764528	2402:4000:2280:c18f...	2620:2d:4000:1::96	HTTP	174	GET / HTTP/1.1
25794	1168.3170731...	192.168.0.198	185.125.190.49	HTTP	154	GET / HTTP/1.1
27261	1195.4939751...	2402:4000:2280:c18f...	2620:2d:4000:1::22	HTTP	174	GET / HTTP/1.1

Packet Details (No. 1000):

- Total Length: 140
- Identification: 0x0799 (1945)
- 010 ... = Flags: 0x2, Don't fragment
- ...0 0000 0000 = Fragment Offset: 0
- Time to Live: 64
- Protocol: TCP (6)
- Header Checksum: 0xba77 [validation disabled]
- [Header checksum status: Unverified]
- Source Address: 192.168.0.198
- Destination Address: 91.189.91.48
- Transmission Control Protocol, Src Port: 42228, Dst Port: 80, Seq: 1, Ack: 1, Len: 88
- Source Port: 42228
- Destination Port: 80
- [Stream index: 0]
- [Conversation completeness: Complete, WITH_DATA (31)]
- [TCP Segment Len: 88]
- Sequence Number: 1 (relative sequence number)
- Sequence Number (raw): 2752647194
- [Next Sequence Number: 89 (relative sequence number)]
- Acknowledgment Number: 1 (relative ack number)
- Acknowledgment number (raw): 3037150324
- 1000 ... = Header Length: 32 bytes (8)
- Flags: 0x010 (PSH, ACK)
- Window: 582
- [Calculated window size: 64256]
- [Window size scaling factor: 128]
- Checksum: 0x2e54 [unverified]
- [Checksum Status: Unverified]
- Urgent Pointer: 0
- Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps

Raw Packet Data (Hex/ASCII):

```

0000 5c a0 02 03 8f d2 cc 64 1a f2 56 48 08 00 45 00  \.....d..VH..E:
0010 00 8c 07 99 40 09 40 06 ba 77 c9 a0 00 c6 5b bd  @..w...[
0020 5b 39 a4 f4 00 50 a4 12 10 1a b5 07 3c 74 80 18  [0...P...<...
0030 01 f6 2e 54 00 00 01 01 00 0a 27 48 97 a1 b5 02  .T...H...
0040 83 b1 47 45 54 20 2f 20 48 54 54 50 2f 31 2e 31  .GET / HTTP/1.1
0050 0d 0a 48 6f 73 74 3a 20 63 6f 6e 6e 65 63 74 69  .Host: connecti
0060 76 69 74 79 2d 63 68 65 63 6b 2e 75 62 75 6e 74  vity-che ck.ubunt
0070 75 2e 63 6f 6d 2e 0d 0a 41 63 63 65 70 74 3a 29  u.com... Accept:
0080 2a 2f 2a 0d 0a 43 6f 6e 6e 65 63 74 69 6f 6e 3a  /*...Con nectio:
0090 20 63 6c 6f 73 65 0d 0a 0d 0a                  close...

```

Consider about the HTTP GET method:

6992	268.65632559	91.189.91.48	192.168.0.198	HTTP	255 HTTP/1.1 204 No Content
7634	278.33377312	2402:4000:2280:c18f_	2620:2d:4000:1::96	HTTP	174 GET / HTTP/1.1
13960	568.408592899	192.168.0.198	91.189.91.48	HTTP	154 GET / HTTP/1.1
13961	568.664061358	91.189.91.48	192.168.0.198	HTTP	255 HTTP/1.1 204 No Content
13998	578.342574736	2402:4000:2280:c18f_	2620:2d:4000:1::96	HTTP	174 GET / HTTP/1.1
20456	868.289237193	192.168.0.198	185.125.190.17	HTTP	154 GET / HTTP/1.1
20561	894.849706649	2402:4000:2280:c18f_	2620:2d:4000:1::197	HTTP	174 GET / HTTP/1.1
20567	895.440764528	2402:4000:2280:c18f_	2620:2d:4000:1::96	HTTP	174 GET / HTTP/1.1
25794	1168.3178731_	192.168.0.198	185.125.190.49	HTTP	154 GET / HTTP/1.1
27261	1195.4939751_	2402:4000:2280:c18f_	2620:2d:4000:1::22	HTTP	174 GET / HTTP/1.1
32654	1468.4004561_	192.168.0.198	91.189.91.48	HTTP	154 GET / HTTP/1.1
32655	1468.6802726_	91.189.91.48	192.168.0.198	HTTP	255 HTTP/1.1 204 No Content
33288	1495.5677883_	2402:4000:2280:c18f_	2620:2d:4000:1::2a	HTTP	174 GET / HTTP/1.1

Lets look at the single Application layer PDU like HTTP GET request.(Consider the tcp IP model)

- Application Layer(PDU name : Message or Data):

```

Wireshark - Packet 7034 - wlxc641af25648
+ Frame 7034: 174 bytes on wire (1392 bits), 174 bytes captured (1392 bits) on interface wlxc641af25648, id 0
+ Ethernet II, Src: cc:64:1a:f2:56:48 (cc:64:1a:f2:56:48), Dst: 5c:a0:02:03:8f:d2 (5c:a0:02:03:8f:d2)
+ Internet Protocol Version 6, Src: 2402:4000:2280:c18f:5592:aac0:8b5:d984, Dst: 2620:2d:4000:1::96
+ Transmission Control Protocol, Src Port: 57476, Dst Port: 80, Seq: 1, Ack: 1, Len: 68
+ Hypertext Transfer Protocol
  GET / HTTP/1.1\r\n
  Host: connectivity-check.ubuntu.com.\r\n
  Accept: */*\r\n
  Connection: close\r\n
  \r\n
  [Full request URI: http://connectivity-check.ubuntu.com/]
  [HTTP request 1/1]

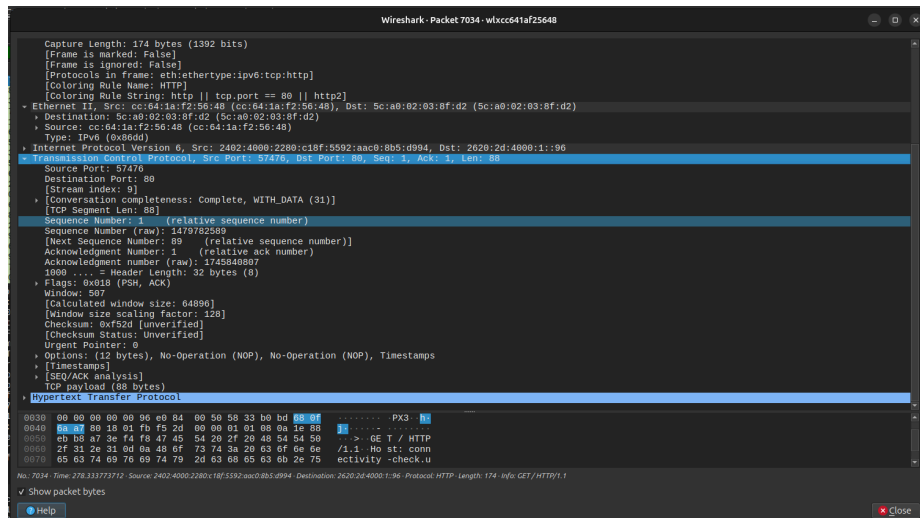
0000  5c a0 02 03 8f d2 5c 64 1a f2 56 48 86 dd 60 02  \....d.VH...
0010  36 7f 00 78 06 48 24 02 40 00 22 80 c1 8f 55 92  6-x0S0"...U.
0020  aa c9 08 05 09 94 20 20 00 2d 40 00 01 00 00    ...A-0...
0030  00 00 00 00 00 9e e8 84 00 50 58 33 b0 bd 68 6f  ....PX3..h.
0040  6a a7 80 18 01 fb f5 2d 00 00 01 01 08 0a 1e 88  j.....
0050  eb 80 a7 3e f4 f8 47 45 54 28 2f 29 a8 84 54 59  ->GET / HTTP
0060  2f 31 2e 31 6d 0a 48 6f 73 74 3a 20 43 6f 6e 6e  /1.1. Host: conn
0070  65 63 74 69 76 69 74 79 2d 63 68 65 63 6b 2e 75  ectivity-check.u
0080  62 75 6e 74 75 2e 63 6f 6d 2e 0d 0a 41 63 63 65  buntu.co m..Acce
0090  70 74 0a 20 2a 2f 2a 0d 0a 43 6f 6e 6e 65 63 74  pt: "/" -Connect
00a0  69 6f 6e 3a 20 63 6c 6f 73 65 0d 0a 0d 0a     ion: clo se...

No: 7034 - Time: 278.33377312 - Source: 2402:4000:2280:c18f:5592:aac0:8b5:d984 - Destination: 2620:2d:4000:1::96 - Protocol: HTTP - Length: 174 - Info: GET / HTTP/1.1
v Show packet bytes
[ Help ] [ Close ]

```

This layer combines the three layers of the OSI model. As the screenshot shows, the Ubuntu connectivity check using the get method. The HTTP protocol formats the request accordingly.

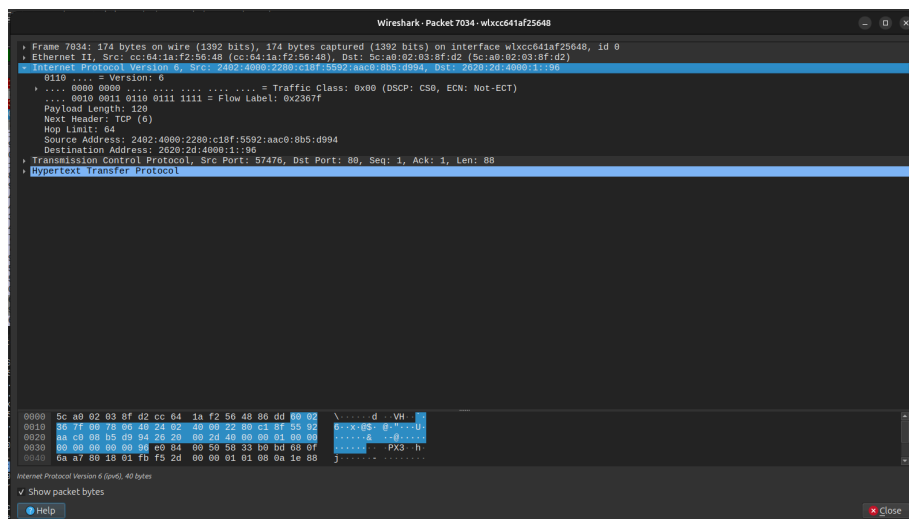
- Transport Layer (PDU name: Segment or Datagram)



This layer is responsible for host to host communication and error recovery. Data chopped into smaller chunks. If using TCP, a header is added that includes Source/Destination Ports(57476 and 80) and Sequence numbers(1), which ensures the pieces are put back together in the right order at the destination.

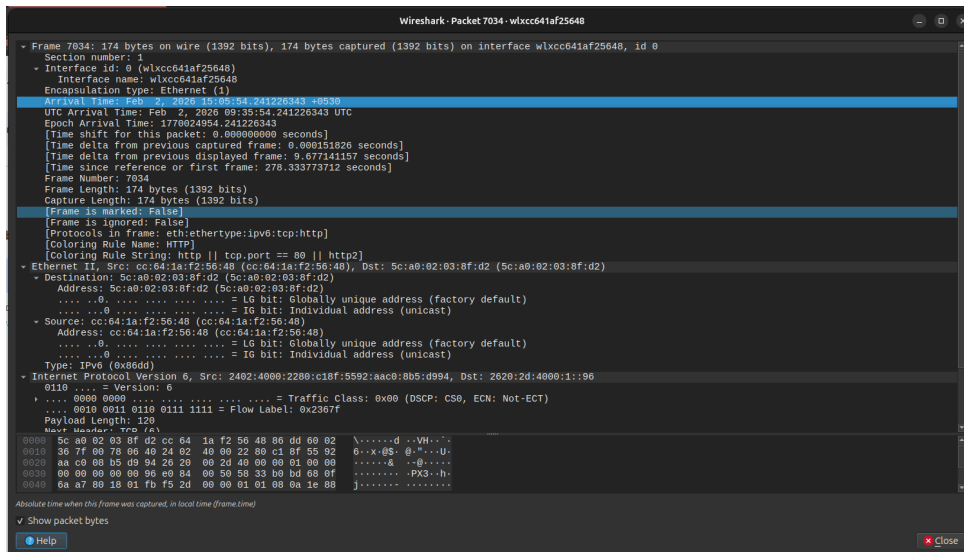
- Internet Layer(PDU name : Packets)

This is the routing layer. The primary job is to move packets from host to host.



The ip header is attached. This includes the Source and Destination IP addresses. Routes and deciding the best path to for your packet to travel.

- Network access Layer (PDU name : Frame)



The packet is wrapped with a MAC address. Also add a “Frame Check Sequence “ (FCS) at the end to detect if any bits were corrupted during the physical transmission. Finally It converts the frame into electrical or radio signals.