

OSI Model, TCP/IP Model & Wireshark Packet Analysis

PART A – OSI Model

1. OSI Layer Explanations

Layer 1 – Physical Layer

The Physical layer handles how raw data moves across a medium, using things like electric signals or light flashes instead of smart processing. It sets rules for wires, plugs, voltage levels, and pin setups - including stuff such as Ethernet cords, fiber lines, or wireless radio waves. Think of it like a highway: just as cars need roads to drive, data needs this layer to go anywhere. There's no routing or labels here - just basic bit transfer from one point to another.

Layer 2 – Data Link Layer

The Data Link layer helps data flow between devices on a single local network. Rather than just passing raw bits, it wraps them into organized frames while handling physical addresses, catching mistakes, or controlling when each device can transmit. Take Ethernet, Wi-Fi (IEEE 802.11), and address lookup - they all work at this stage. Without these rules, many gadgets could talk at once, leading to chaos. Imagine it like a bouncer at an entrance checking IDs before allowing entry.

Layer 3 – Network Layer

The Network layer sends data from one network to another. Instead it picks the most efficient route using IP addresses to guide the way. Devices like routers work right here, handling traffic just like mail centers sorting

parcels. Protocols such as IPv4, ICMP, OSPF, and IPv6 run on this level without exception. Think of it like a delivery system planning how packages move between distant places. Unlike Layer 2, which stays inside local zones, Layer 3 links far-off locations together.

Layer 4 – Transport Layer

The Transport layer manages how data moves from one device to another across a network. Instead it takes care of splitting info into chunks, putting them back together, managing speed, and making sure nothing gets lost. On this level you've got TCP and UDP - TCP works like a confirmed delivery with tracking, whereas UDP skips the paperwork for quicker drops. Meanwhile ports such as 21, 53, or 443 are tied right here, helping direct traffic. Think of it like a mail runner who doesn't just drop off a package at the door but makes sure it goes to the exact room using number tags (ports).

Layer 5 – Session Layer

The Session layer handles how connections between apps begin, stay running, or close. Instead it watches conversations already in motion - like signing into an account, chatting on video, or moving files - and makes sure things don't drop or drift out of sync. Tools like remote procedure calls, NetBIOS, or digital session tags work here. Think about making a call: you dial, talk while linked, then stop once someone disconnects.

Layer 6 – Presentation Layer

The Presentation layer shifts data from app format to network form - or back again. It handles tasks such as encrypting info, unscrambling it later, squeezing files down, or turning them into readable code. Things like JPEG images, MP3 audio, PNG graphics, secure connections via SSL/TLS, plus text systems including ASCII and Unicode fit right in this space. Think of someone translating speech so folks who speak differently can actually get

what's being said. What matters here is how information looks - transmission details? Not its job.

Layer 7 – Application Layer

The Application layer lets people work with network tools up close. It handles ways to surf websites, send emails, move files around while chatting online. You'll find stuff like HTTP, HTTPS, DNS, along with FTP, SMTP, or even DHCP living right here. Think of it like a reception area at an office - folks show up there asking for help. That doesn't refer to mobile apps, rather the backend functions those programs depend on.

2. OSI Mnemonic

Mnemonic:

“Please Do Not Throw Sausage Pizza Away”

This phrase is a popular and easy way to remember the seven layers of the OSI (Open Systems Interconnection) model from bottom to top. Each word represents the first letter of a layer, helping students quickly recall the correct order.

P – Physical (Layer 1)

This is the lowest layer of the OSI model. The Physical Layer deals with raw data transmission as electrical, optical, or radio signals. It defines cables, connectors, voltage levels, timings, light pulses, and frequencies. Anything related to the physical hardware and actual movement of bits happens here—for example Ethernet cables, Wi-Fi radio waves, fiber optics, and network interface cards (NICs). It does not understand packets or addresses; it only moves bits (0s and 1s).

D – Data Link (Layer 2)

The Data Link Layer ensures that data transferred over the physical medium is error-free, reliable, and properly framed. It organizes bits into frames, manages MAC addresses, and handles error detection using checksums. It also decides who can send data at a time using protocols like CSMA/CD or CSMA/CA. Devices like switches operate at this layer. It is divided into two sublayers: LLC (Logical Link Control) and MAC (Media Access Control).

N – Network (Layer 3)

The Network Layer handles routing and logical addressing, meaning it decides the best path for data to travel across networks. It uses IP addresses to identify devices across different networks and performs packet forwarding, fragmentation, and traffic control. Routers work at this layer. Protocols such as IPv4, IPv6, ICMP, and ARP function here. The focus is on moving packets from source to destination across multiple networks.

T – Transport (Layer 4)

The Transport Layer ensures end-to-end communication between devices. It provides reliable or unreliable data delivery depending on the protocol. TCP ensures reliable communication through error correction, sequencing, acknowledgment, and congestion control. UDP provides faster but connectionless communication for real-time applications like gaming and VoIP. This layer breaks data into segments and ensures that the data arrives completely and correctly.

S – Session (Layer 5)

The Session Layer manages the start, control, and end of communication sessions between devices or applications. It keeps track of connections so that two applications can continue their conversation without interruption.

It also handles session checkpoints, so if a connection fails midway, communication can resume from the last saved point. Examples include online meetings, remote logins, and continuous data exchange.

P – Presentation (Layer 6)

The Presentation Layer acts as the translator of the OSI model. It makes sure the data sent from one application can be understood by another. It handles data formatting, encryption, compression, and conversion into standard formats. This is where tasks like converting text to ASCII or Unicode, encrypting data using SSL/TLS, or compressing files (ZIP, JPEG, MPEG) occur. Essentially, it prepares data for the Application Layer and vice versa.

A – Application (Layer 7)

The Application Layer is the closest to the user. It provides user interfaces and network services directly used by applications such as browsers, email clients, file transfer tools, and messaging apps. It enables activities such as web browsing, sending emails, watching videos, and downloading files. Protocols like HTTP, HTTPS, FTP, SMTP, and DNS operate here. This is where interaction between humans and applications happens.

3. OSI vs TCP/IP Model Comparison

The OSI and TCP/IP models both describe how data moves from a sender to a receiver, but they were created for different purposes. The OSI Model is a **theoretical reference model** with 7 separate layers. It was mainly designed to help people understand the flow of data and to standardize networking concepts. Each layer has a very specific job, and the OSI model separates responsibilities in a very detailed manner. Because of this, it is widely used in education and documentation.

The TCP/IP Model, in contrast, is a **practical model** developed based on the actual protocols used on the internet. It has only 4 layers, because it combines some of the OSI layers into broader, more functional layers. TCP/IP focuses on real-world communication, which is why routers, servers, and the entire internet still depend on it. While OSI is better for learning, TCP/IP is what actually runs modern networks. Both models explain the same process but at different levels of abstraction.

Mapping Table

OSI Layer	TCP/IP Layer
Application / Presentation / Session (L7/L6/L5)	Application
Transport (L4)	Transport
Network (L3)	Internet
Data Link / Physical (L2/L1)	Network Access

4. Protocol Data Units (PDUs)

Every layer of the OSI model has a special name for the data it processes. This name represents what kind of information the layer adds to the data. As data moves downward from the Application layer to the Physical layer, each layer adds its own header (and sometimes trailer). As the data moves upward, each layer removes its corresponding header and interprets it.

Layer 4 – Transport

At the Transport layer, the data unit is called a Segment if it is using TCP (because TCP is connection-oriented and breaks data into numbered segments). For UDP, the PDU is called a Datagram because UDP sends data in lightweight, connectionless chunks.

Layer 3 – Network

The PDU here is a Packet. This layer adds logical addressing (IP addresses) and routing information. The network layer's job is to move packets from one network to another.

Layer 2 – Data Link

Here the PDU is called a Frame. This layer adds MAC addresses and prepares data for local delivery within the same network. It also handles error detection via frame checksums.

Layer 1 – Physical

The PDU is simply Bits. These are electrical signals, radio waves, or light pulses that represent binary 1s and 0s.

PDU Table

OSI Layer	PDU Name
Layer 4 – Transport	Segment (TCP) / Datagram (UDP)
Layer 3 – Network	Packet
Layer 2 – Data Link	Frame
Layer 1 – Physical	Bits

5. Addressing Concepts

Addressing plays a vital role in how data is delivered across networks. Each type of address works at a different OSI layer and serves a different purpose.

MAC Address – Used at Layer 2 (Data Link Layer)

A MAC address is a unique, fixed, hardware-based address assigned to a device's network interface card (like your Wi-Fi or Ethernet adapter). It is expressed in hexadecimal format (e.g., A4-5E-60-1B-2C-7D). MAC addresses are used only within the local network, meaning your device uses them to communicate with routers, switches, or nearby devices. Layer 2 frames rely on MAC addresses for delivery on a LAN. This type of addressing does not change unless the hardware is replaced.

IP Address – Used at Layer 3 (Network Layer)

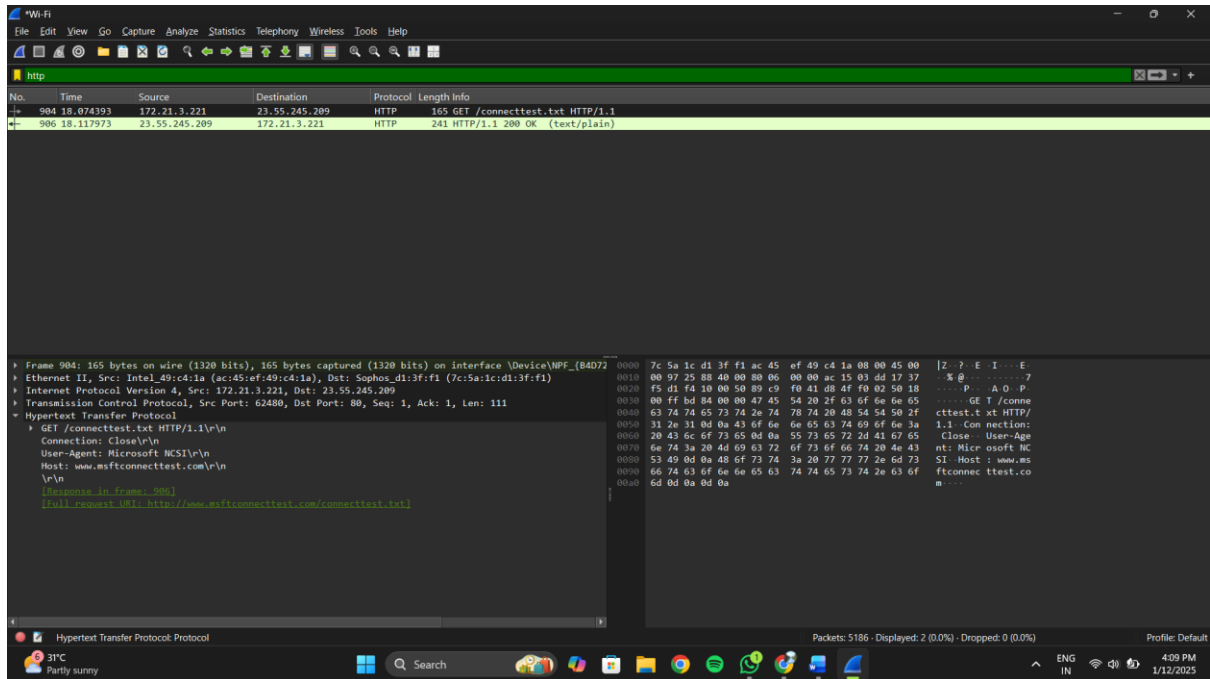
An IP address is a logical, software-assigned address that identifies a device on a global network. Unlike MAC addresses, IP addresses can change when you switch networks, restart devices, or use DHCP. IP addresses help routers make decisions about the best path to reach a destination. For example, your laptop might have 192.168.1.10 on a home network but a completely different IP when connected to campus Wi-Fi. IP addressing allows communication beyond the local network.

Port Number – Used at Layer 4 (Transport Layer)

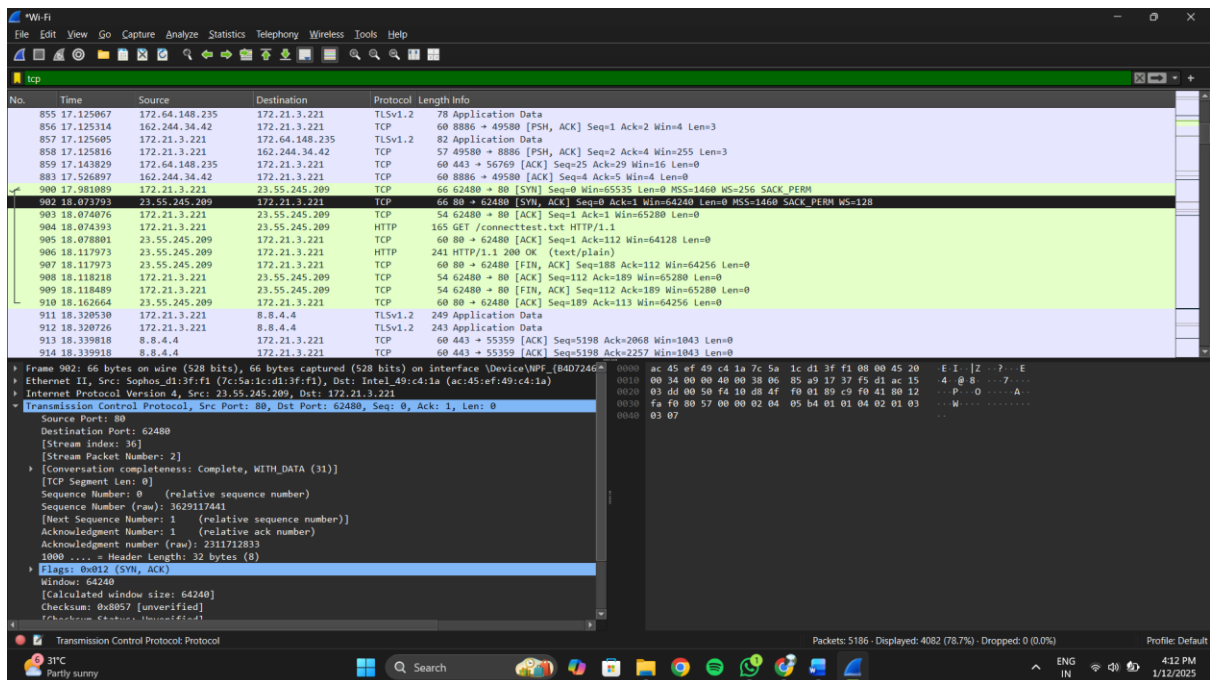
A port number identifies specific applications or services running on a device. While IP addresses identify *where* a device is, port numbers identify *what* service the device wants to access. For example, visiting a website uses TCP port 80 (HTTP) or 443 (HTTPS), while DNS uses UDP port 53. Multiple applications can run on the same device using different ports, allowing many network services to operate simultaneously. Ports, combined with IP addresses, create a socket, such as:
192.168.1.10:443

Part B – Wireshark Practical

1. HTTP Traffic



2. TCP Packets



3. UDP Packets

The screenshot shows a Wireshark capture of UDP traffic. The packet list on the left shows a series of packets, with packet 897 selected. The packet details pane on the right shows the structure of the selected packet, which is a DNS query. The packet bytes pane on the right shows the raw data of the packet, including the Ethernet II header, Internet Protocol Version 4 header, and User Datagram Protocol header. The packet is a standard query for the domain 'www.msftncsi.com.edgesuite.net'.

No.	Time	Source	Destination	Protocol	Length	Info
889	17.573396	172.21.0.12	255.255.255.255	UDP	554	9999 → 9999 Len=512
890	17.573396	172.21.0.31	255.255.255.255	UDP	554	9999 → 9999 Len=512
891	17.607496	172.21.0.17	255.255.255.255	UDP	554	9999 → 9999 Len=512
892	17.620937	172.21.0.30	255.255.255.255	UDP	554	9999 → 9999 Len=512
893	17.627879	172.21.0.56	255.255.255.255	UDP	554	9999 → 9999 Len=512
894	17.665948	172.21.0.38	255.255.255.255	UDP	554	9999 → 9999 Len=512
896	17.884122	172.21.3.238	239.255.255.250	SSDP	212	M-SEARCH * HTTP/1.1
897	17.899083	172.21.0.22	172.21.255.255	UDP	102	32919 → 7788 Len=60
898	17.917243	172.21.3.221	8.8.8.8	DNS	83	Standard query 0xb203 A www.msftconnecttest.com
899	17.919370	8.8.8.8	172.21.3.221	DNS	227	Standard query response 0xb203 A www.msftconnecttest.com CNAME ncsi-geo.trafficmanager.net CNAME www.msftncsi.com.edgesuite.net...
927	18.383998	172.21.5.110	224.0.0.251	MDNS	87	Standard query 0x0000 PTR _spotify-connect._tcp.local, "QM" question
928	18.383998	fe80::e191:7dd6:7cb...ff02::fb	ff02::fb	MDNS	107	Standard query 0x0000 PTR _spotify-connect._tcp.local, "QM" question
934	18.427882	172.21.4.214	224.0.0.251	MDNS	340	Standard query response 0x0000 PTR Spotify Desktop Launcher._spotify-connect._tcp.local SRV 0 0 55102 DESKTOP-HAISC19.local TXT...
935	18.441107	172.21.6.100	172.21.255.255	UDP	82	57621 → 57621 Len=40
948	18.516142	172.21.6.100	172.21.255.255	UDP	82	57621 → 57621 Len=40
951	18.616460	172.21.3.238	239.255.255.250	SSDP	212	M-SEARCH * HTTP/1.1
952	18.630500	172.21.5.110	239.255.255.250	SSDP	167	M-SEARCH * HTTP/1.1
989	18.730667	172.21.3.221	142.251.43.67	QUIC	1292	Initial, DCID=7ca5323a3c819607, PKN: 1, PING, PING, PADDING, PING, PING, PING, PING, CRYPTO, CRYPTO, CRYPTO, CRYPTO, PING, PING...
990	18.730818	172.21.3.221	142.251.43.67	QUIC	1292	Initial, DCID=7ca5323a3c819607, PKN: 2, CRYPTO, PING, PING, PADDING, CRYPTO, CRYPTO, CRYPTO, PING, PING, CRYPTO, CRYPTO, PADDING...
1014	18.898752	172.21.0.11	255.255.255.255	UDP	554	9999 → 9999 Len=512

Frame 897: 102 bytes on wire (816 bits), 102 bytes captured (816 bits) on interface \Device\NPF... (8AD7246-2...)
Ethernet II, Src: ASUSTekCOMP0_08:e4:60 (50:eb:f6:08:e4:60), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Internet Protocol Version 4, Src: 172.21.0.22, Dst: 172.21.255.255
User Datagram Protocol, Src Port: 32919, Dst Port: 7788
Source Port: 32919
Destination Port: 7788
Length: 68
Checksum: 0xd9b6 [unverified]
[Checksum Status: Unverified]
[Stream index: 93]
[Stream Packet Number: 1]
[Timestamps]
UDP payload (60 bytes)
Data (60 bytes)

4. ICMP Packets (Ping)

(REQUEST)

The screenshot shows a Wireshark capture of ICMP traffic. The packet list on the left shows a series of packets, with packet 95 selected. The packet details pane on the right shows the structure of the selected packet, which is an ICMP Echo (ping) request. The packet bytes pane on the right shows the raw data of the packet, including the Ethernet II header, Internet Protocol Version 4 header, and Internet Control Message Protocol header. The packet is a standard Echo (ping) request to the destination 8.8.8.8.

No.	Time	Source	Destination	Protocol	Length	Info
95	1.911631	172.21.3.221	8.8.8.8	ICMP	74	Echo (ping) request id=0x0001, seq=5/1280, ttl=128 (reply in 96)
96	1.943099	8.8.8.8	172.21.3.221	ICMP	74	Echo (ping) reply id=0x0001, seq=5/1280, ttl=128 (request in 95)
122	2.940705	172.21.3.221	8.8.8.8	ICMP	74	Echo (ping) request id=0x0001, seq=6/1536, ttl=128 (reply in 125)
125	2.975572	8.8.8.8	172.21.3.221	ICMP	74	Echo (ping) reply id=0x0001, seq=6/1536, ttl=128 (request in 122)
147	3.970373	172.21.3.221	8.8.8.8	ICMP	74	Echo (ping) request id=0x0001, seq=7/1792, ttl=128 (reply in 148)
148	3.993989	8.8.8.8	172.21.3.221	ICMP	74	Echo (ping) reply id=0x0001, seq=7/1792, ttl=128 (request in 147)
166	4.983392	172.21.3.221	8.8.8.8	ICMP	74	Echo (ping) request id=0x0001, seq=8/2048, ttl=128 (reply in 168)
168	5.043042	8.8.8.8	172.21.3.221	ICMP	74	Echo (ping) reply id=0x0001, seq=8/2048, ttl=128 (request in 166)

Frame 95: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF... (8AD7246-2...)
Ethernet II, Src: Intel_49:c4:1a (ac:45:ef:49:c4:1a), Dst: Sophos_d1:3f:f1 (7c:5a:1c:d1:3f:f1)
Internet Protocol Version 4, Src: 172.21.3.221, Dst: 8.8.8.8
Internet Control Message Protocol
Type: 8 (Echo (ping) request)
Codes: 0
Checksum: 0x4d56 [correct]
[Checksum Status: Good]
Identifier (BE): 1 (0x0001)
Identifier (LE): 256 (0x0100)
Sequence Number (BE): 5 (0x0005)
Sequence Number (LE): 1280 (0x0500)
[Response Frame: 96]
Data (32 bytes)

(REPLY)

The screenshot shows a Wireshark capture of ICMP Echo (ping) replies. The packet list pane displays several packets, with packet 96 selected. The packet details pane shows the structure of the ICMP Echo (ping) reply, including the Echo (ping) request ID, sequence number, and TTL. The packet bytes pane shows the raw data of the packet, including the Ethernet II header, Internet Protocol Version 4 header, and Internet Control Message Protocol header.

No.	Time	Source	Destination	Protocol	Length	Info
95	1.911631	172.21.3.221	8.8.8.8	ICMP	74	Echo (ping) request id=0x0001, seq=5/1280, ttl=128 (reply in 96)
96	1.943099	8.8.8.8	172.21.3.221	ICMP	74	Echo (ping) reply id=0x0001, seq=5/1280, ttl=128 (request in 95)
122	2.940705	172.21.3.221	8.8.8.8	ICMP	74	Echo (ping) request id=0x0001, seq=6/1536, ttl=128 (reply in 125)
125	2.975572	8.8.8.8	172.21.3.221	ICMP	74	Echo (ping) reply id=0x0001, seq=6/1536, ttl=128 (request in 122)
147	3.970373	172.21.3.221	8.8.8.8	ICMP	74	Echo (ping) request id=0x0001, seq=7/1792, ttl=128 (reply in 148)
148	3.993989	8.8.8.8	172.21.3.221	ICMP	74	Echo (ping) reply id=0x0001, seq=7/1792, ttl=128 (request in 147)
166	4.983392	172.21.3.221	8.8.8.8	ICMP	74	Echo (ping) request id=0x0001, seq=8/2048, ttl=128 (reply in 168)
168	5.043842	8.8.8.8	172.21.3.221	ICMP	74	Echo (ping) reply id=0x0001, seq=8/2048, ttl=128 (request in 166)

Frame 96: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{B4D72460-2} Ethernet II, Src: Sophos_d1:3f:f1 (7c:5a:1c:d1:3f:f1), Dst: Intel_49:c4:1a (ac:45:ef:49:c4:1a) Internet Protocol Version 4, Src: 8.8.8.8, Dst: 172.21.3.221 Internet Control Message Protocol Type: 0 (Echo (ping) reply) Code: 0 Checksum: 0x5556 [correct] [Checksum Status: Good] Identifier (BE): 1 (0x0001) Identifier (LE): 256 (0x0100) Sequence Number (BE): 5 (0x0005) Sequence Number (LE): 1280 (0x0500) [Response time: 31.468 ms] Data (32 bytes)

5. ARP Frames

(REQUEST)

The screenshot shows a Wireshark capture of ARP request frames. The packet list pane displays several packets, with packet 150 selected. The packet details pane shows the structure of the ARP request, including the Hardware type, Protocol type, Hardware size, Protocol size, Opcode, Sender MAC address, Sender IP address, Target MAC address, and Target IP address. The packet bytes pane shows the raw data of the packet, including the Ethernet II header, Internet Protocol Version 4 header, and Address Resolution Protocol header.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	HewlettPacka_cc:c7:...	Broadcast	ARP	60	Who has 172.21.0.1? Tell 172.21.5.121
4	0.000155	CloudNetwork_81:eb:...	Broadcast	ARP	60	ARP Announcement for 172.21.3.218
9	0.927080	Sophos_d1:3f:f1	Broadcast	ARP	60	Who has 172.21.3.175? Tell 172.21.0.1
47	1.557419	AzureNaveTec_51:82:...	Broadcast	ARP	60	Who has 172.21.2.158? (ARP Probe)
72	1.662901	AzureNaveTec_51:82:...	Broadcast	ARP	60	Who has 172.21.0.1? Tell 172.21.2.158
80	1.764864	Sophos_d1:3f:f1	Broadcast	ARP	60	Who has 172.21.3.175? Tell 172.21.0.1
100	2.169135	7c:d9:d1:90:29:d4	Broadcast	ARP	60	Gratuitous ARP for 172.21.1.17 (Reply)
105	2.459424	AzureNaveTec_51:82:...	Broadcast	ARP	60	Who has 172.21.2.158? (ARP Probe)
119	2.790184	Sophos_d1:3f:f1	Broadcast	ARP	60	Who has 172.21.3.175? Tell 172.21.0.1
138	3.486917	AzureNaveTec_51:82:...	Broadcast	ARP	60	Who has 172.21.2.158? (ARP Probe)
150	4.154552	AzureNaveTec_f0:81:a9	Broadcast	ARP	60	Who has 172.21.0.14? Tell 172.21.5.130
152	4.400460	Sophos_d1:3f:f1	Broadcast	ARP	60	Who has 172.21.3.175? Tell 172.21.0.1
154	4.502098	AzureNaveTec_51:82:...	Broadcast	ARP	60	ARP Announcement for 172.21.2.158
175	5.284518	ee:a0:2b:0c:92:d6	Broadcast	ARP	60	Gratuitous ARP for 172.21.8.31 (Reply)
176	5.421007	Sophos_d1:3f:f1	Broadcast	ARP	60	Who has 172.21.3.175? Tell 172.21.0.1
200	6.450339	AzureNaveTec_51:82:...	Broadcast	ARP	60	ARP Announcement for 172.21.2.158
201	6.471276	Sophos_d1:3f:f1	Broadcast	ARP	60	Who has 172.21.3.175? Tell 172.21.0.1
205	6.781052	ee:a0:2b:0c:92:d6	Broadcast	ARP	60	Who has 172.21.0.1? Tell 172.21.8.31
224	7.365736	AzureNaveTec_51:82:...	Broadcast	ARP	60	Who has 172.21.0.14? Tell 172.21.2.158
226	7.372040	HewlettPacka_cc:c7:...	Broadcast	ARP	60	Who has 172.21.0.1? Tell 172.21.5.130

Frame 150: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{B4D72460-2} Ethernet II, Src: AzureNaveTec_f0:81:a9 (1c:ce:51:f0:81:a9), Dst: Broadcast (ff:ff:ff:ff:ff:ff) Address Resolution Protocol (request) Hardware type: Ethernet (1) Protocol type: IPv4 (0x0800) Hardware size: 6 Protocol size: 4 Opcode: request (1) Sender MAC address: AzureNaveTec_f0:81:a9 (1c:ce:51:f0:81:a9) Sender IP address: 172.21.1.2 Target MAC address: 00:00:00:00:00:00 (00:00:00:00:00:00) Target IP address: 172.21.0.1

(REPLY)

Wireshark packet capture showing ARP traffic. The selected packet is a gratuitous ARP reply from 172.21.0.17 to 172.21.0.1.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	HewlettPacka_cc:c7:...	Broadcast	ARP	60	Who has 172.21.0.1? Tell 172.21.5.121
4	0.080155	Cloudnetwork_81:eb:...	Broadcast	ARP	60	ARP Announcement for 172.21.3.218
9	0.927080	Sophos_d1:3f:f1	Broadcast	ARP	60	Who has 172.21.3.175? Tell 172.21.0.1
47	1.557419	AzureNaveTec_51:82:...	Broadcast	ARP	60	Who has 172.21.2.158? (ARP Probe)
72	1.662901	AzureNaveTec_51:82:...	Broadcast	ARP	60	Who has 172.21.0.1? Tell 172.21.2.158
80	1.764864	Sophos_d1:3f:f1	Broadcast	ARP	60	Who has 172.21.3.175? Tell 172.21.0.1
100	2.169136	7e:d9:41:90:29:d4	Broadcast	ARP	60	Gratuitous ARP for 172.21.1.17 (Reply)
105	2.459424	AzureNaveTec_51:82:...	Broadcast	ARP	60	Who has 172.21.2.158? (ARP Probe)
119	2.790184	Sophos_d1:3f:f1	Broadcast	ARP	60	Who has 172.21.3.175? Tell 172.21.0.1
130	3.486917	AzureNaveTec_51:82:...	Broadcast	ARP	60	Who has 172.21.2.158? (ARP Probe)
150	4.194059	AzureNaveTec_f0:81:...	Broadcast	ARP	60	Who has 172.21.0.1? Tell 172.21.1.2
152	4.400460	Sophos_d1:3f:f1	Broadcast	ARP	60	Who has 172.21.3.175? Tell 172.21.0.1
154	4.502098	AzureNaveTec_51:82:...	Broadcast	ARP	60	ARP Announcement for 172.21.2.158
172	5.927451	172.21.0.17	172.21.0.1	ARP	60	Gratuitous ARP for 172.21.0.1 (Reply)
176	5.421007	Sophos_d1:3f:f1	Broadcast	ARP	60	Who has 172.21.3.175? Tell 172.21.0.1
200	6.459339	AzureNaveTec_51:82:...	Broadcast	ARP	60	ARP Announcement for 172.21.2.158
201	6.471276	Sophos_d1:3f:f1	Broadcast	ARP	60	Who has 172.21.3.175? Tell 172.21.0.1
205	6.783052	ee:a0:2b:0c:92:d6	Broadcast	ARP	60	Who has 172.21.0.1? Tell 172.21.8.31
224	7.365736	AzureNaveTec_51:82:...	Broadcast	ARP	60	Who has 172.21.0.14? Tell 172.21.2.158
226	7.372940	HewlettPacka_ca:6a:...	Broadcast	ARP	60	Who has 172.21.0.1? Tell 172.21.5.130

Frame 175: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{B4D72460-0000-0000-0000-000000000000}

Ethernet II, Src: ee:a0:2b:0c:92:d6 (ee:a0:2b:0c:92:d6), Dst: Broadcast (ff:ff:ff:ff:ff:ff)

Address Resolution Protocol (reply/gratuitous ARP)

Hardware type: Ethernet (1)

Protocol type: IPv4 (0x0800)

Hardware size: 6

Protocol size: 4

Opcode: reply (2)

[Is gratuitous: true]

Sender MAC address: ee:a0:2b:0c:92:d6 (ee:a0:2b:0c:92:d6)

Sender IP address: 172.21.8.31

Target MAC address: Broadcast (ff:ff:ff:ff:ff:ff)

Target IP address: 172.21.8.31

Packets: 291 - Displayed: 26 (8.9%) - Dropped: 0 (0.0%)

Profile: Default

Hot days ahead 30°C

Search

ENG IN 4:20 PM 1/12/2025