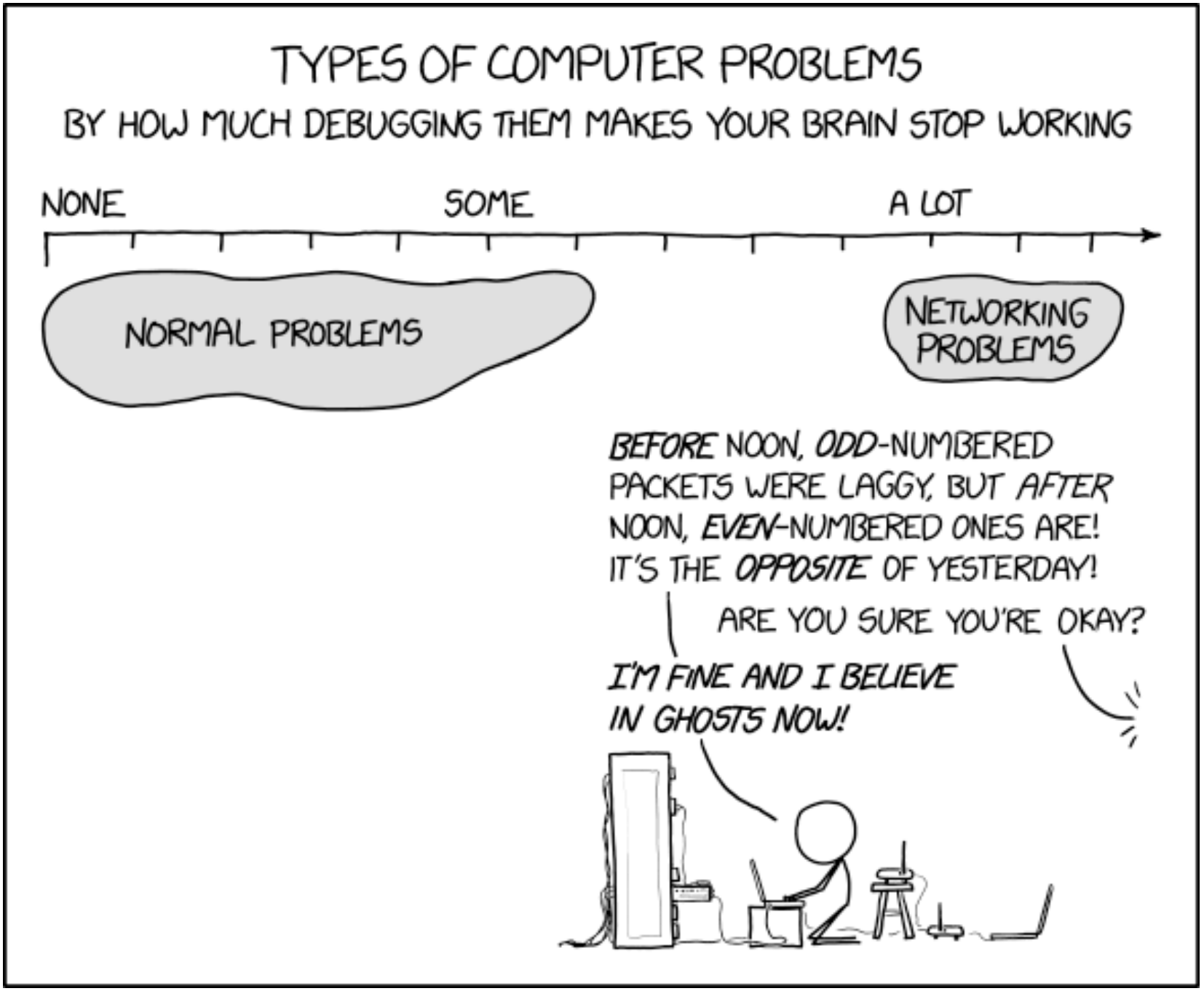


https://xkcd.com/2259

CS 60:
Computer Networks

Network layers

Review from last class

Networks are "unreasonably" fast! (200 ms to Australia vs 100 ms at speed of light)
The Internet is a "network of networks"
Components:



Hub: most basic connecting device **Layer 1**

- Broadcasts data to all hub-connected devices
- Outdated for most modern networks due to their inefficiency
- Found mainly in small home LANs (if you can find one!)



Switch: more intelligent than hubs **Layer 2**

- Directs traffic based on MAC addresses
- Improves network performance and creates multiple collision domains



Router: connects networks **Layer 3**

- Manages traffic between networks using IP addresses
- Enables Internet connectivity
- Can provide additional services such as firewalls and DHCP



Wireless Access Point (AP): connects devices using radio frequencies (RF)

- Commonly acts like a router for mobile and wireless devices

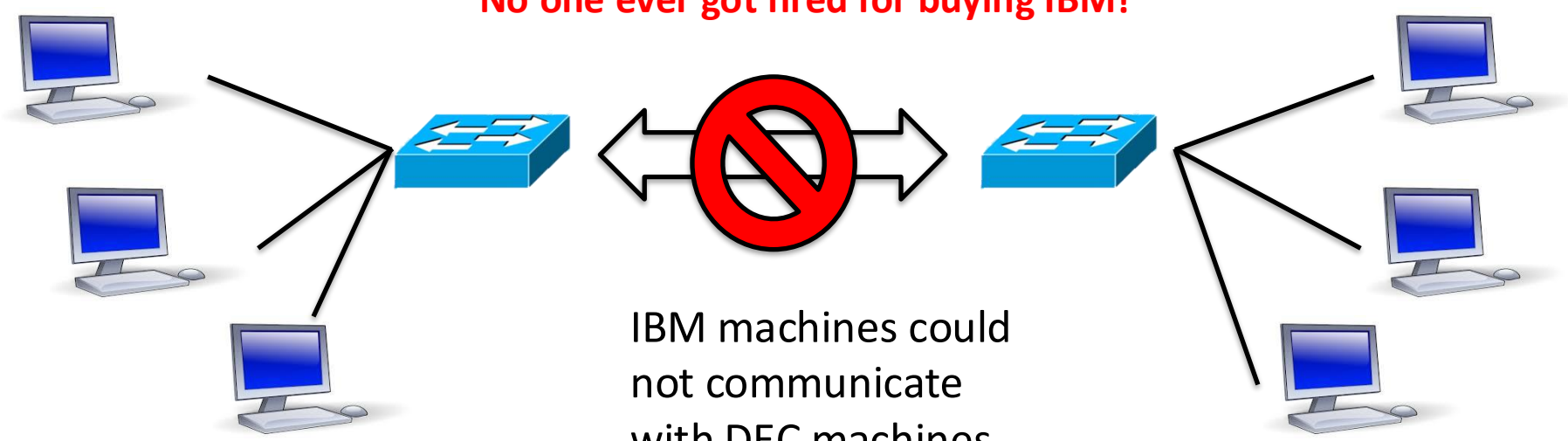
Layer 1, 2, and 3

Agenda



1. Network layer overview
2. DHCP
3. Ping, traceroute, and ARP
4. Exercises

In the old days, networking protocols were proprietary to each manufacturer



IBM machines could not communicate with DEC machines

IBM had protocols for communicating with other IBM machines

Example: SNA

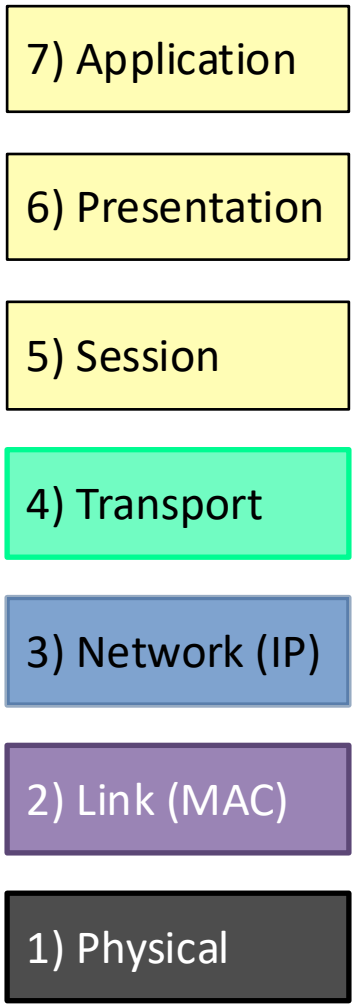
Incompatibility problems!

DEC had protocols for communicating with other DEC machines

Example: DECnet

Two frameworks were developed to remedy incompatibility problems

Open Systems Interconnection (OSI)



https://www.ibm.com/think/topics/osi-model

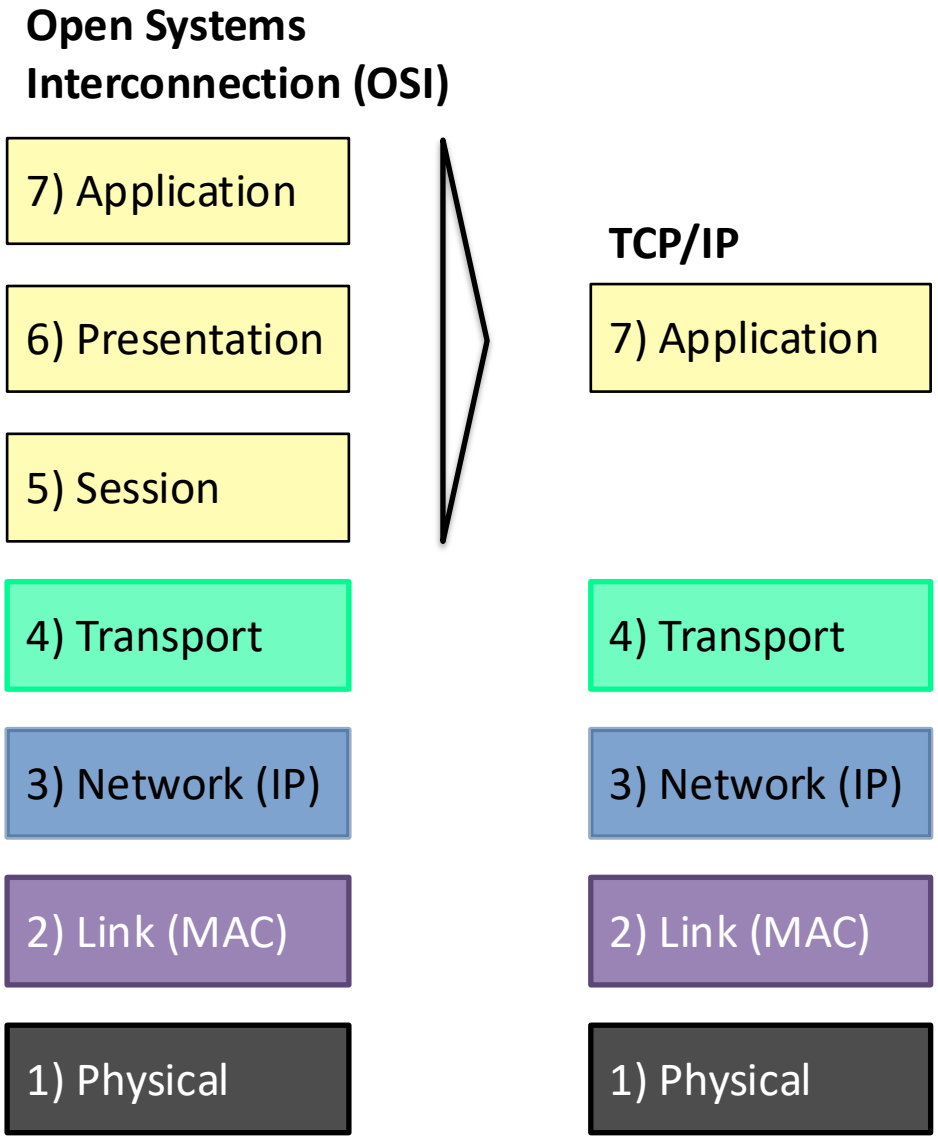
Goal: define each layer of network from the physical up to applications to standardize how communications work

International Organization for Standardization (ISO) published OSI model in 1984

Key point: Upper layers still function the same, even if lower layers change

For example: Network layer and above work the same regardless if using RF (Wi-Fi) or electrical (Ethernet cables)

Two frameworks were developed to remedy incompatibility problems



https://www.ibm.com/think/topics/osi-model

TCP/IP Model

Five layers by combining Application layers (sometimes Physical and Link layers combined to four-layer model)

We still call Application layer, Layer 7!

TCP/IP model sometimes called:

- Network model
- Internet model
- Protocol stack
- Reference model

Maddeningly, I may use all these 5-layer terms!

The OSI model's primary value lies in its educational utility and its role as a conceptual framework for designing new protocols to ensure compatibility

"TCP/IP model's practical focus and real-world applicability have made it the backbone of modern networking"

Physical layer transports bits

Conceptual network layers

7) Application

4) Transport

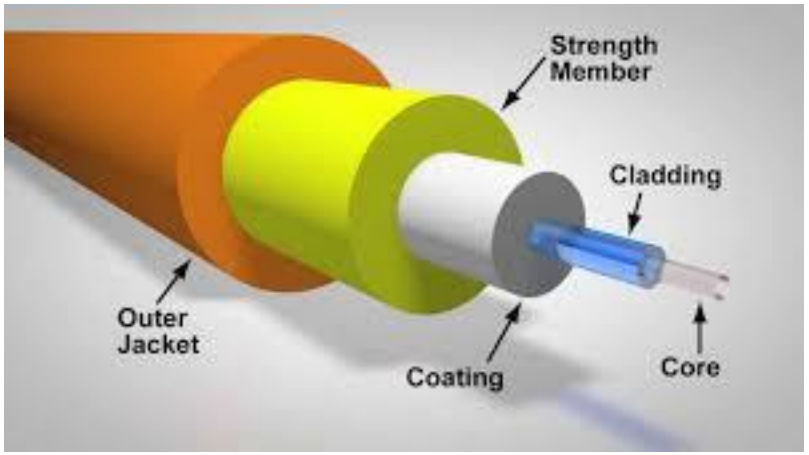
3) Network (IP)

2) Link (MAC)

1) Physical

DO NOT BEND FIBER CABLES They will break if you bend them too far!

Fiber optic cable



Glass or plastic core channels light (from lasers or LEDs)
At a given time: light pulse = 1 , no light pulse = 0
Two types

- Single mode – one light frequency, small core (~9 microns), long-range (~40 km), useful for long haul
- Multimode – multiple freqs, large core (~60 microns), short range (~100m), useful in building/campus
- Theoretical speed in petabits (practical about 800Gb/sec)

How data is physically transmitted

- Transmitter converts logical 1 and 0 ***bits*** to electrical/light pulses or phase/amplitude of radio frequency (RF) and sends down wire or over air
- Receiver converts electrical/light or RF back to logical 1 and 0 bits

Physical layer transports bits

Conceptual network layers

7) Application

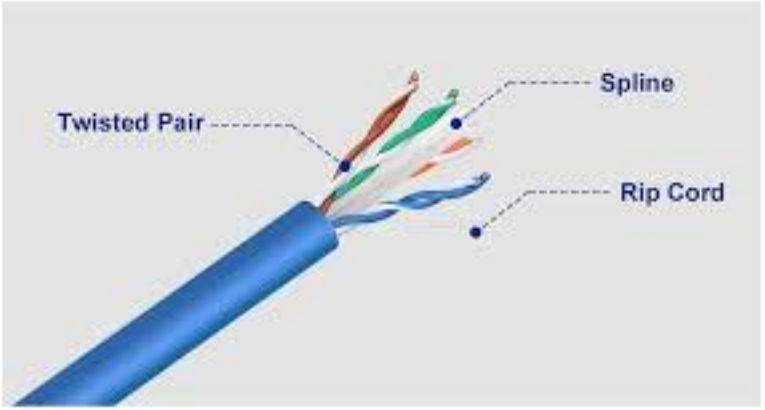
4) Transport

3) Network (IP)

2) Link (MAC)

1) Physical

Ethernet cable



Twisted pairs of copper wire
Unshielded twisted pair (UTP) or shielded twisted pair (STP)
Category 5e (1 Gb/sec) to Category 8 (40 Gb/sec)
Goes about 100m
Two kinds:

- Straight through (computer to switch/router)
- Crossover (computer to computer)

How data is physically transmitted

- Transmitter converts logical 1 and 0 ***bits*** to electrical/light pulses or phase/amplitude of radio frequency (RF) and sends down wire or over air
- Receiver converts electrical/light or RF back to logical 1 and 0 bits

Physical layer transports bits

Conceptual network layers

7) Application

4) Transport

3) Network (IP)

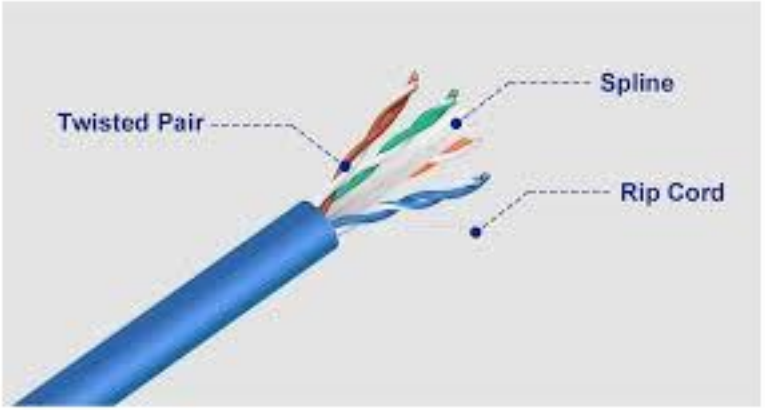
2) Link (MAC)

1) Physical

How would you tap an ethernet cable?

- **Use a hub!**
- **But you'll need to cut the wire**
- **Let's do that**

Ethernet cable



Twisted pairs of copper wire
Unshielded twisted pair (UTP) or shielded twisted pair (STP)
Category 5e (1 Gb/sec) to Category 8 (40 Gb/sec)
Goes about 100m
Two kinds:

- Straight through (computer to switch/router)
- Crossover (computer to computer)

How data is physically transmitted

- Transmitter converts logical 1 and 0 ***bits*** to electrical/light pulses or phase/amplitude of radio frequency (RF) and sends down wire or over air
- Receiver converts electrical/light or RF back to logical 1 and 0 bits

https://www.ibm.com/think/topics/osi-model

Physical layer transports bits

Conceptual network layers

7) Application

4) Transport

3) Network (IP)

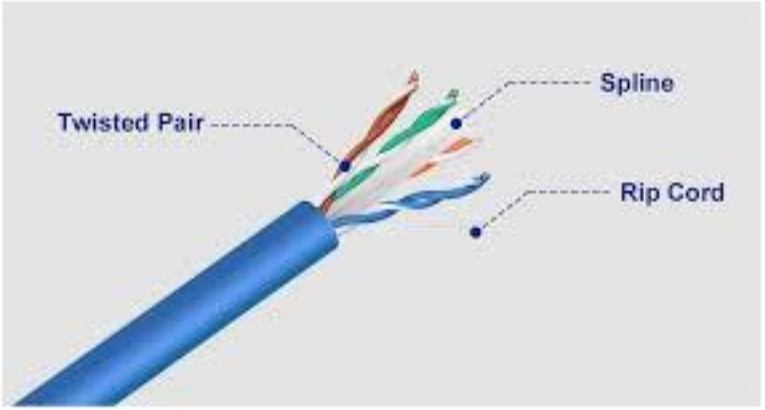
2) Link (MAC)

1) Physical

How would you tap an ethernet cable?

- **Use a hub!**
- **But you'll need to cut the wire**
- **Let's do that**

Ethernet cable



Twisted pairs of copper wire
Unshielded twisted pair (UTP) or shielded tw
Category 5e (1 Gb/sec) to Category 8 (40 Gb/
Goes about 100m
Two kinds:

- Straight through (computer to switch/rou
- Crossover (computer to computer)

T568A:
1-White/Green
2-Green
3-White/Orange
4-Blue
5-White/Blue
6-Orange
7-White/Brown
8-Brown

T568B:
1-White/Orange
2-Orange
3-White/Green
4-Blue
5-White/Blue
6-Green
7-White/Brown
8-Brown

Straight (B to B)
Crossover(A to B)

How data is physically transmitted

- Transmitter converts logical 1 and 0 ***bits*** to electrical/light pulses or phase/amplitude of radio frequency (RF) and sends down wire or over air
- Receiver converts electrical/light or RF back to logical 1 and 0 bits

Physical layer transports bits

Conceptual network layers

- 7) Application
- 4) Transport
- 3) Network (IP)
- 2) Link (MAC)
- 1) Physical

How would you tap an ethernet cable?

- **Use a hub!**
- **But you'll need to cut the wire**
- **Let's do that**
- **Could also use a Throwing Star instead (good if you don't have power available for the hub)**

https://greatscottgadgets.com/throwingstar/

Ethernet cable



Note: these days NICs auto sense straight through vs cross over

Twisted pairs of copper wire
Unshielded twisted pair (UTP) or shielded tw
Category 5e (1 Gb/sec) to Category 8 (40 Gb/
Goes about 100m
Two kinds:

- Straight through (computer to switch/rou
- Crossover (computer to computer)

T568A:
1-White/Green
2-Green
3-White/Orange
4-Blue
5-White/Blue
6-Orange
7-White/Brown
8-Brown

T568B:
1-White/Orange
2-Orange
3-White/Green
4-Blue
5-White/Blue
6-Green
7-White/Brown
8-Brown

Straight (B to B)
Crossover(A to B)

How data is physically transmitted

- Transmitter converts logical 1 and 0 ***bits*** to electrical/light pulses or phase/amplitude of radio frequency (RF) and sends down wire or over air
- Receiver converts electrical/light or RF back to logical 1 and 0 bits

Physical layer transports bits

Conceptual network layers

7) Application

4) Transport

3) Network (IP)

2) Link (MAC)

1) Physical

Regardless of how physical layer works, received *bits* are decoded and sent to Layer 2 (Link Layer)

Radio Frequency (RF)



Sends data over the air using radio frequency
Examples: Wi-Fi, Bluetooth, Cellular, Satellite
Transmitter sends at known phase and amplitude
Receiver converts RF phase/amplitude into 1 and 0 bits
Wi-Fi ~100m, Bluetooth ~10m
Theoretical max Wi-Fi speed about 46 Gb/sec (802.11be)

How data is physically transmitted

- Transmitter converts logical 1 and 0 ***bits*** to electrical/light pulses or phase/amplitude of radio frequency (RF) and sends down wire or over air
- Receiver converts electrical/light or RF back to logical 1 and 0 bits

Link layer connects computers in a Local Area Network (LAN)

Conceptual network layers

7) Application

4) Transport

3) Network (IP)

2) Link (MAC)

1) Physical

https://www.ibm.com/think/topics/osi-model

Each NIC has a unique MAC address burned into NIC's ROM by the manufacturer, does not (normally) change



Wired Network Interface Card

- Common on non-mobile devices such as desktops, IP phones, servers
- Network cable plugs into NIC



Wireless Wi-Fi Interface Card

- Common on mobile devices such as smart phones, tablets, IoT devices
- Connects to Wi-Fi Access Point (which normally acts as a router) via RF

Moves ***frames*** within a local area network (switching)
Each computer identified by a MAC address on its Network Interface Card (NIC)
Also called Layer 2, MAC layer, Data Link layer, or Ethernet layer

Does the Link Layer care about how it got the bits from the PHY layer? No!

How data is physically transmitted

- Transmitter converts logical 1 and 0 ***bits*** to electrical/light pulses or phase/amplitude of radio frequency (RF) and sends down wire or over air
- Receiver converts electrical/light or RF back to logical 1 and 0 bits

Link layer connects computers in a Local Area Network (LAN)

Conceptual network layers

7) Application

4) Transport

3) Network (IP)

2) Link (MAC)

1) Physical

Find your MAC address with **ifconfig** (**i**fconfig Mac/Linux, i**p**config Windows)

```
% ifconfig
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=6460<TSO4,TSO6,CHANNEL_IO,PARTIAL_CSUM,ZEROINVERT_CSUM>
        ether 5c:e9:1e:a2:4d:4f
        inet6 fe80::1421:7489:af6b:6a95%en0 prefixlen 64 secured scopeid 0xf
        inet 10.0.0.64 netmask 0xffffff00 broadcast 10.0.0.255
        inet6 2601:19e:8200:80d0:81b:df78:32be:289 prefixlen 64 autoconf secured
        inet6 2601:19e:8200:80d0:1cbd:e651:1f3f:8753 prefixlen 64 autoconf temporary
        inet6 2601:19e:8200:80d0::8ec8 prefixlen 64 dynamic
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect
        status: active
```

MAC address uniquely identifies each host

Example: two iPhones of same model will have different MACs

Moves ***frames*** within a local area network (switching)
Each computer identified by a MAC address on its Network Interface Card (NIC)
Also called Layer 2, MAC layer, Data Link layer, or Ethernet layer

How data is physically transmitted

- Transmitter converts logical 1 and 0 ***bits*** to electrical/light pulses or phase/amplitude of radio frequency (RF) and sends down wire or over air
- Receiver converts electrical/light or RF back to logical 1 and 0 bits

https://www.ibm.com/think/topics/osi-model

Link layer connects computers in a Local Area Network (LAN)

Conceptual network layers

7) Application

4) Transport

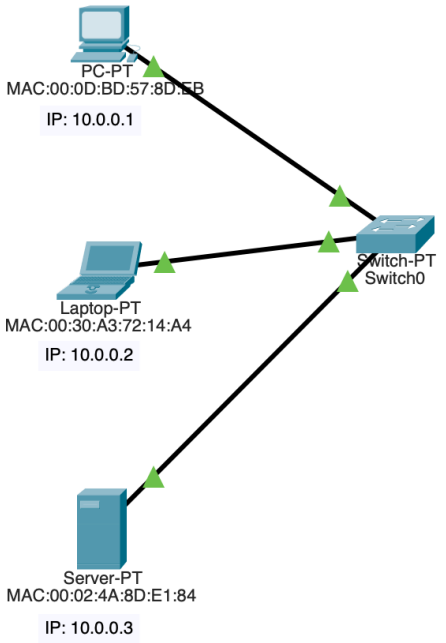
3) Network (IP)

2) Link (MAC)

1) Physical

Switches:

- Connect devices in a local area
- Send traffic to devices based on their MAC
- Switch learns the MAC address of a device when it first transmits



Moves ***frames*** within a local area network (switching)
Each computer identified by a MAC address on its Network Interface Card (NIC)
Also called Layer 2, MAC layer, Data Link layer, or Ethernet layer

How data is physically transmitted

- Transmitter converts logical 1 and 0 ***bits*** to electrical/light pulses or phase/amplitude of radio frequency (RF) and sends down wire or over air
- Receiver converts electrical/light or RF back to logical 1 and 0 bits

https://www.ibm.com/think/topics/osi-model

Link layer connects computers in a Local Area Network (LAN)

Conceptual network layers

7) Application

4) Transport

3) Network (IP)

2) Link (MAC)

1) Physical

https://www.ibm.com/think/topics/osi-model

Simplified table

Port	MAC	Timeout (secs)
1	00:1a:2b:3c:4d:5e	300
2	00:aa:bb:cc:dd:ee	121.43

Switches:

- Connect devices in a local area
- Send traffic to devices based on their MAC
- Switch learns the MAC address of a device when it first transmits
- Keep a table of which MAC is plugged into which port
- Switches do not know about higher layers (only Layer 2)



Switch port

Key point: Layer 2 deals with computers in the *same* LAN

No routing!

No IP addresses

Moves ***frames*** within a local area network (switching)
Each computer identified by a MAC address on its Network Interface Card (NIC)
Also called Layer 2, MAC layer, Data Link layer, or Ethernet layer

How data is physically transmitted

- Transmitter converts logical 1 and 0 ***bits*** to electrical/light pulses or phase/amplitude of radio frequency (RF) and sends down wire or over air
- Receiver converts electrical/light or RF back to logical 1 and 0 bits

Network layer moves packets across networks

Conceptual network layers

7) Application

4) Transport

3) Network (IP)

Moves ***packets*** between local area networks (routing)
Each computer on the Internet identified by an IP address (IP v4 or v6)
Also called Layer 3 or IP layer (ICMP Ping is here)

2) Link (MAC)

Moves ***frames*** within a local area network (switching)
Each computer identified by a MAC address on its Network Interface Card (NIC)
Also called Layer 2, MAC layer, Data Link layer, or Ethernet layer

1) Physical

How data is physically transmitted

- Transmitter converts logical 1 and 0 ***bits*** to electrical/light pulses or phase/amplitude of radio frequency (RF) and sends down wire or over air
- Receiver converts electrical/light or RF back to logical 1 and 0 bits

https://www.ibm.com/think/topics/osi-model

Network layer moves packets across networks

Conceptual network layers

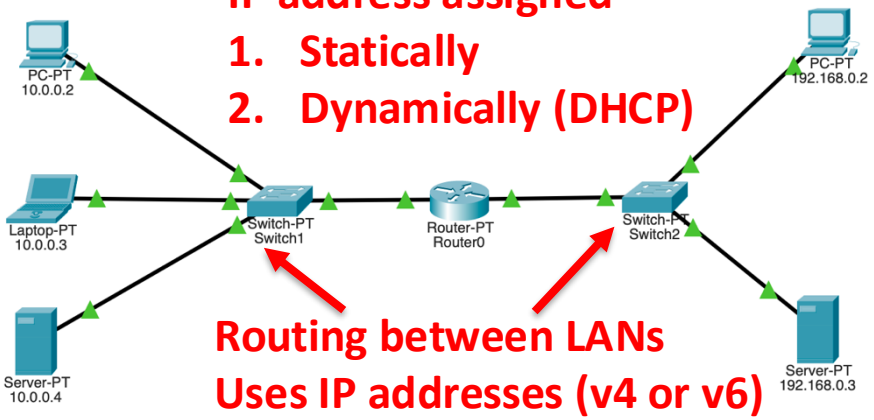
7) Application

4) Transport

3) Network (IP)

2) Link (MAC)

1) Physical



Moves ***packets*** between local area networks (routing)
Each computer on the Internet identified by an IP address (IP v4 or v6)
Also called Layer 3 or IP layer (ICMP Ping is here)

Moves ***frames*** within a local area network (switching)
Each computer identified by a MAC address on its Network Interface Card (NIC)
Also called Layer 2, MAC layer, Data Link layer, or Ethernet layer

How data is physically transmitted

- Transmitter converts logical 1 and 0 ***bits*** to electrical/light pulses or phase/amplitude of radio frequency (RF) and sends down wire or over air
- Receiver converts electrical/light or RF back to logical 1 and 0 bits

https://www.ibm.com/think/topics/osi-model

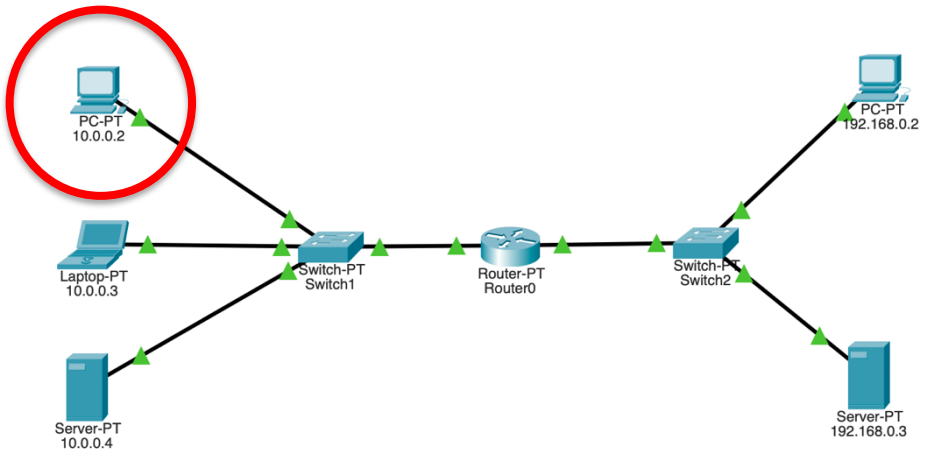
Network layer moves packets across networks

Conceptual network layers

7) Application

4) Transport

Layer 3 finds a route between LANs Say from this PC…



3) Network (IP)

Moves ***packets*** between local area networks (routing)
Each computer on the Internet identified by an IP address (IP v4 or v6)
Also called Layer 3 or IP layer (ICMP Ping is here)

2) Link (MAC)

Moves ***frames*** within a local area network (switching)
Each computer identified by a MAC address on its Network Interface Card (NIC)
Also called Layer 2, MAC layer, Data Link layer, or Ethernet layer

1) Physical

How data is physically transmitted

- Transmitter converts logical 1 and 0 ***bits*** to electrical/light pulses or phase/amplitude of radio frequency (RF) and sends down wire or over air
- Receiver converts electrical/light or RF back to logical 1 and 0 bits

https://www.ibm.com/think/topics/osi-model

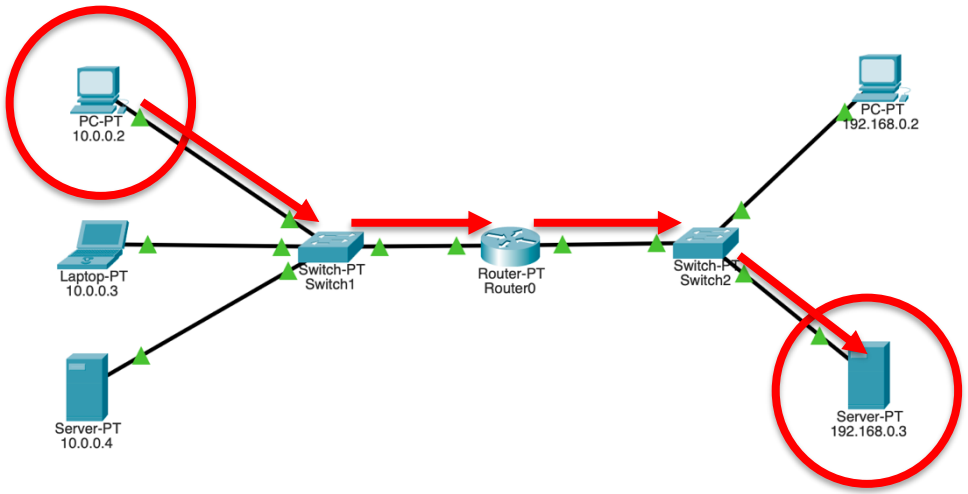
Network layer moves packets across networks

Conceptual network layers

7) Application

4) Transport

Layer 3 finds a route between LANs Say from this PC…



…to this server

3) Network (IP)

Moves ***packets*** between local area networks (routing)
Each computer on the Internet identified by an IP address (IP v4 or v6)
Also called Layer 3 or IP layer (ICMP Ping is here)

2) Link (MAC)

Moves ***frames*** within a local area network (switching)
Each computer identified by a MAC address on its Network Interface Card (NIC)
Also called Layer 2, MAC layer, Data Link layer, or Ethernet layer

1) Physical

How data is physically transmitted

- Transmitter converts logical 1 and 0 ***bits*** to electrical/light pulses or phase/amplitude of radio frequency (RF) and sends down wire or over air
- Receiver converts electrical/light or RF back to logical 1 and 0 bits

https://www.ibm.com/think/topics/osi-model

Network layer moves packets across networks

Conceptual network layers

7) Application

4) Transport

3) Network (IP)

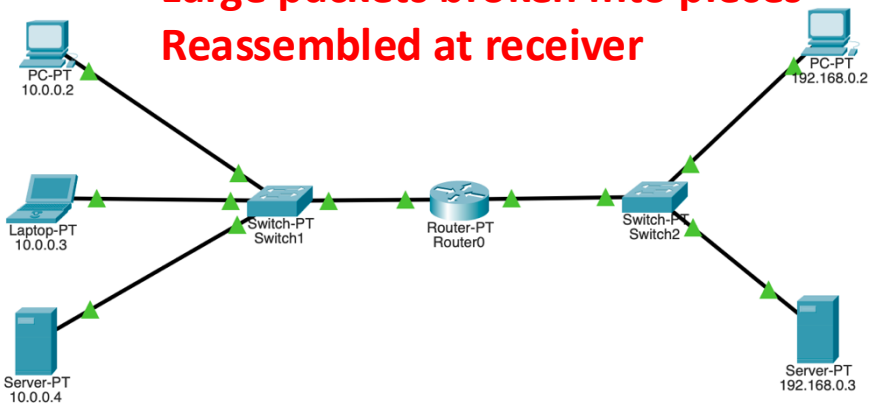
2) Link (MAC)

1) Physical

Packet delivery is not guaranteed at Layer 3

Routing finds efficient routes between LANs

Large packets broken into pieces
Reassembled at receiver



Moves ***packets*** between local area networks (routing)
Each computer on the Internet identified by an IP address (IP v4 or v6)
Also called Layer 3 or IP layer (ICMP Ping is here)

Moves ***frames*** within a local area network (switching)
Each computer identified by a MAC address on its Network Interface Card (NIC)
Also called Layer 2, MAC layer, Data Link layer, or Ethernet layer

How data is physically transmitted

- Transmitter converts logical 1 and 0 ***bits*** to electrical/light pulses or phase/amplitude of radio frequency (RF) and sends down wire or over air
- Receiver converts electrical/light or RF back to logical 1 and 0 bits

https://www.ibm.com/think/topics/osi-model

Transport layer moves segments or datagrams

Conceptual network layers

7) Application

Can you make a reliable channel over an unreliable channel? How?
TCP provides "guaranteed" delivery, UDP does not
Provides port to make sure data gets to right application (layer above)
Port at Layer 4 is a number, not a physical port like Layer 1

4) Transport

Moves ***segments*** *or* ***datagrams***
May provide error control, flow control, application addressing (ports)
Examples: TCP (connection-oriented), UDP (connectionless)
TCP provides sequencing, dropped packet resend, traffic congestion control

3) Network (IP)

Moves ***packets*** between local area networks (routing)
Each computer on the Internet identified by an IP address (IP v4 or v6)
Also called Layer 3 or IP layer (ICMP Ping is here)

2) Link (MAC)

Moves ***frames*** within a local area network (switching)
Each computer identified by a MAC address on its Network Interface Card (NIC)
Also called Layer 2, MAC layer, Data Link layer, or Ethernet layer

1) Physical

How data is physically transmitted

- Transmitter converts logical 1 and 0 ***bits*** to electrical/light pulses or phase/amplitude of radio frequency (RF) and sends down wire or over air
- Receiver converts electrical/light or RF back to logical 1 and 0 bits

https://www.ibm.com/think/topics/osi-model

Application layer moves messages between applications

Conceptual network layers

7) Application

Interacts with application programs to send ***messages***
Applications assigned a port, multiple instances can run (many browser pages)
Examples: HTTP, SSH, FTP, SMTP, DNS

4) Transport

Moves ***segments or datagrams***
May provide error control, flow control, application addressing (ports)
Examples: TCP (connection-oriented), UDP (connectionless)
TCP provides sequencing, dropped packet resend, traffic congestion control

3) Network (IP)

Moves ***packets*** between local area networks (routing)
Each computer on the Internet identified by an IP address (IP v4 or v6)
Also called Layer 3 or IP layer (ICMP Ping is here)

2) Link (MAC)

Moves ***frames*** within a local area network (switching)
Each computer identified by a MAC address on its Network Interface Card (NIC)
Also called Layer 2, MAC layer, Data Link layer, or Ethernet layer

1) Physical

How data is physically transmitted

- Transmitter converts logical 1 and 0 ***bits*** to electrical/light pulses or phase/amplitude of radio frequency (RF) and sends down wire or over air
- Receiver converts electrical/light or RF back to logical 1 and 0 bits

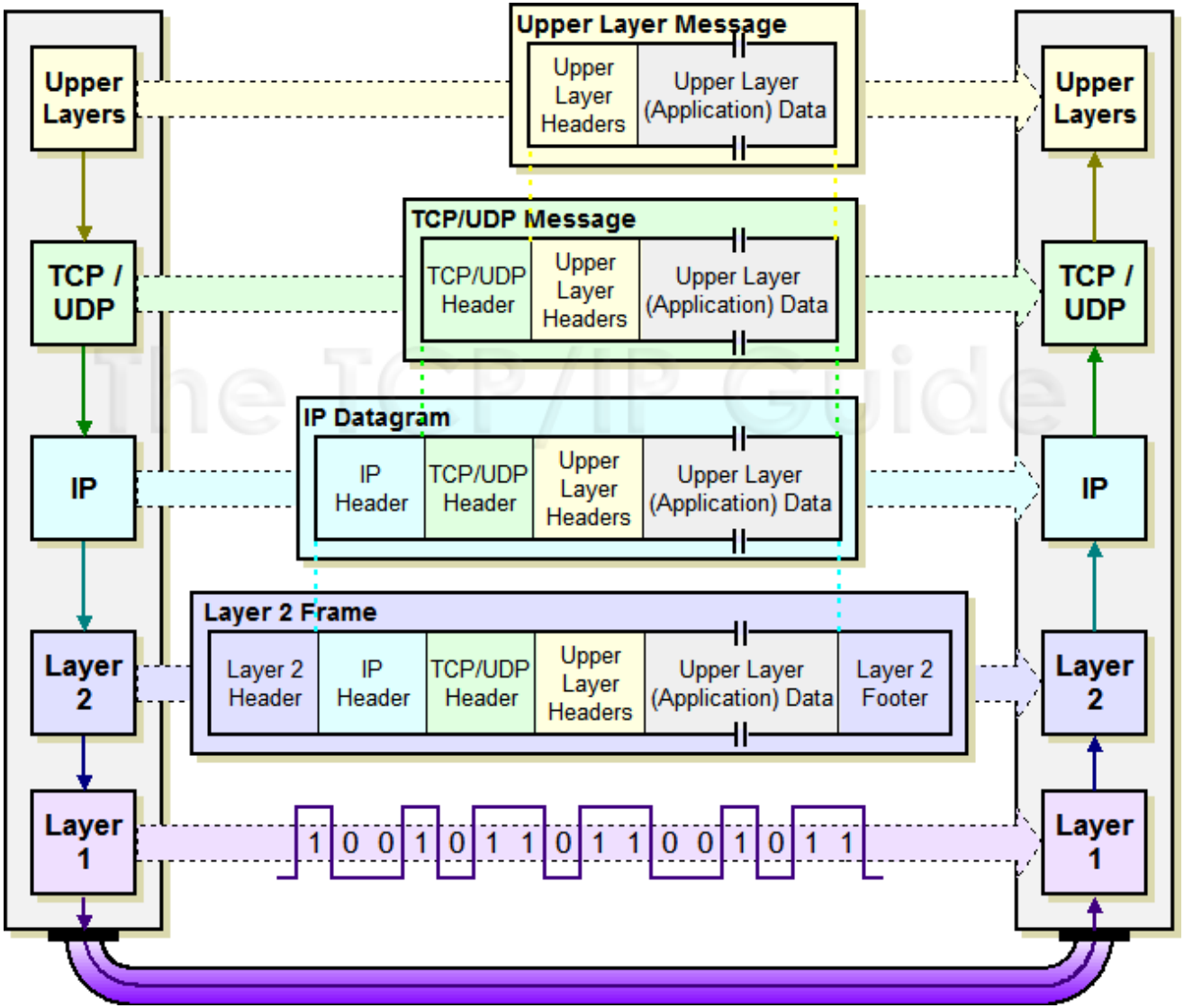
https://www.ibm.com/think/topics/osi-model

Messages travel down the stack from the transmitter and up the stack at the receiver

Transmitter
Wants to send message to receiver

Message starts at top and is encapsulated by each layer down the stack

Each layer adds it own headers



Receiver
Message starts at the physical layer and moves upward

Each layer strips off headers from prior layer

Adapted from http://www.tcpipguide.com/free/t_IPDatagramEncapsulation.htm

Agenda

1. Network layer overview
2. DHCP
3. Ping, traceroute, and ARP
4. Exercises

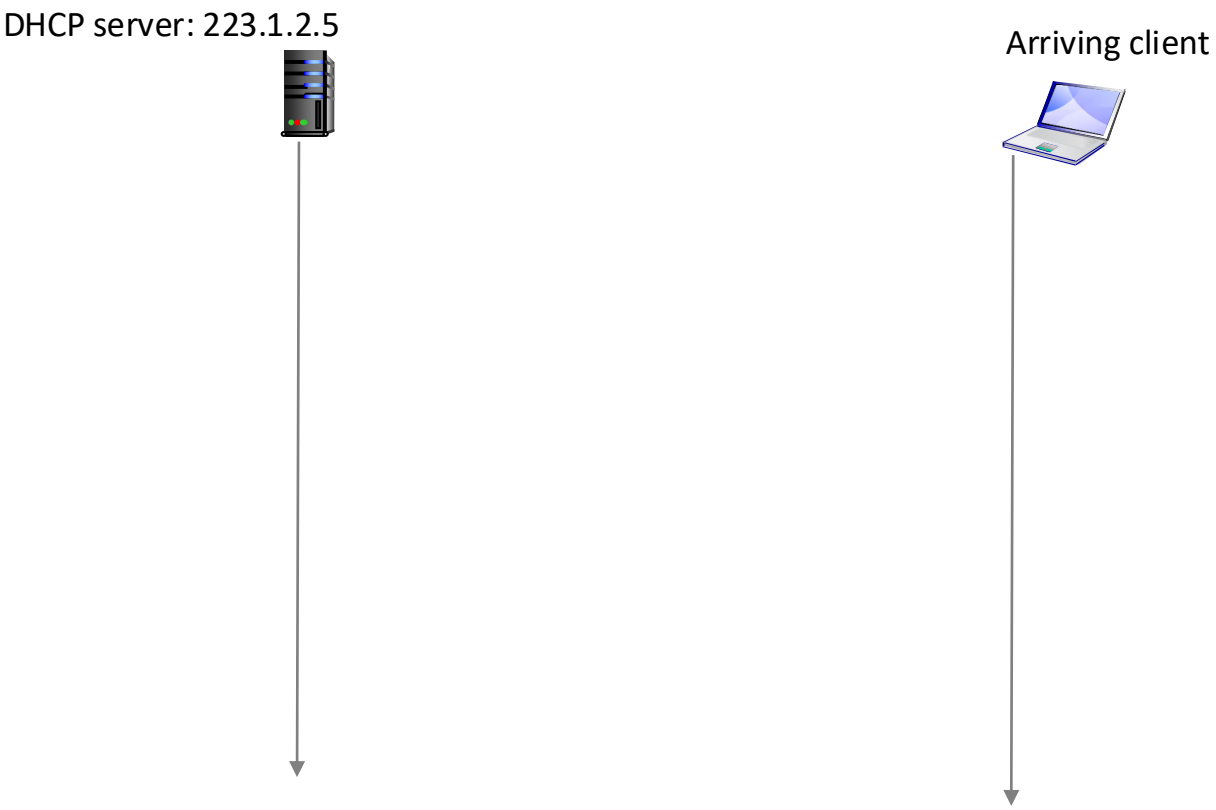
Two ways to get an IP address: static or dynamic (DHCP)

Two ways to get an IP address:

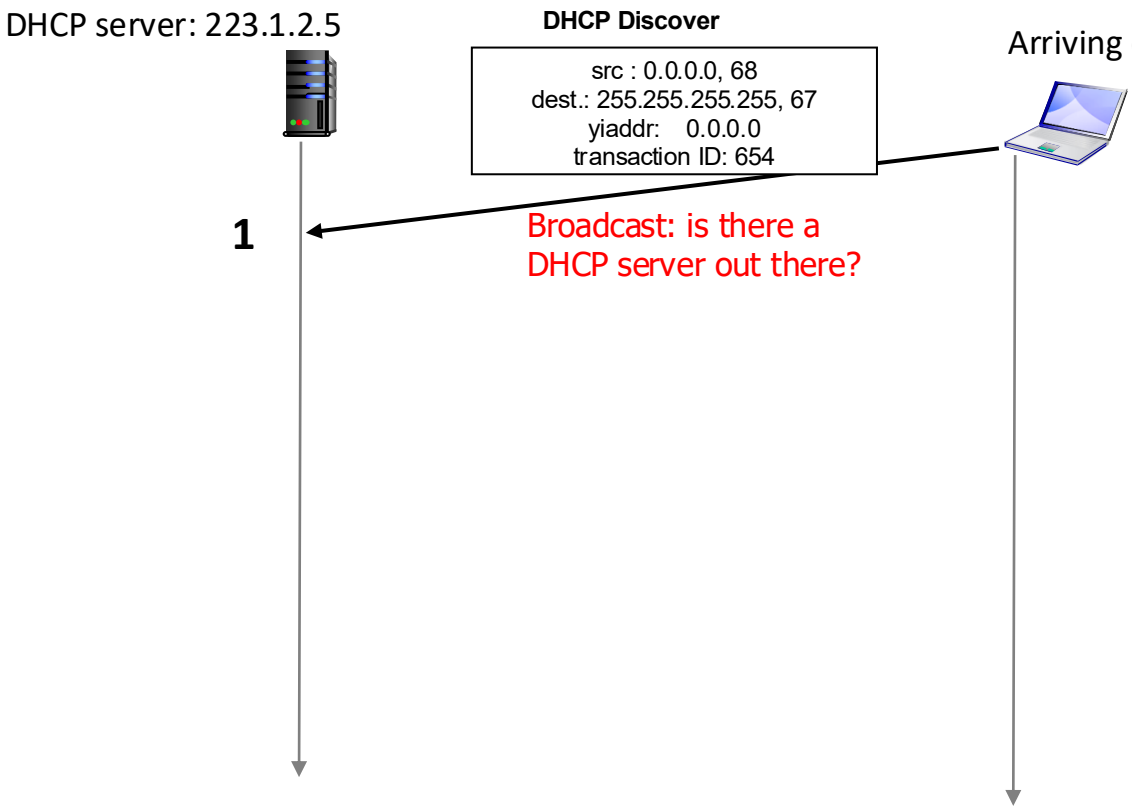
1. Manually set a static IP
   - In the old days, a network admin manually set the IP for each host
   - Sometimes done today for things that shouldn't change IP address like servers

2. Use Dynamic Host Configuration Protocol (DHCP)
   - Assign a server to give out IP addresses from a pool
   - When hosts join the network, DHCP server gives it an IP address not already in use
   - Each IP has a lease with an expiration time so the IP address assignment will expire (in case a device leaves the network)
   - Hosts can renew their lease

DHCP: Client wants to get an IP address, but doesn't know about DHCP server



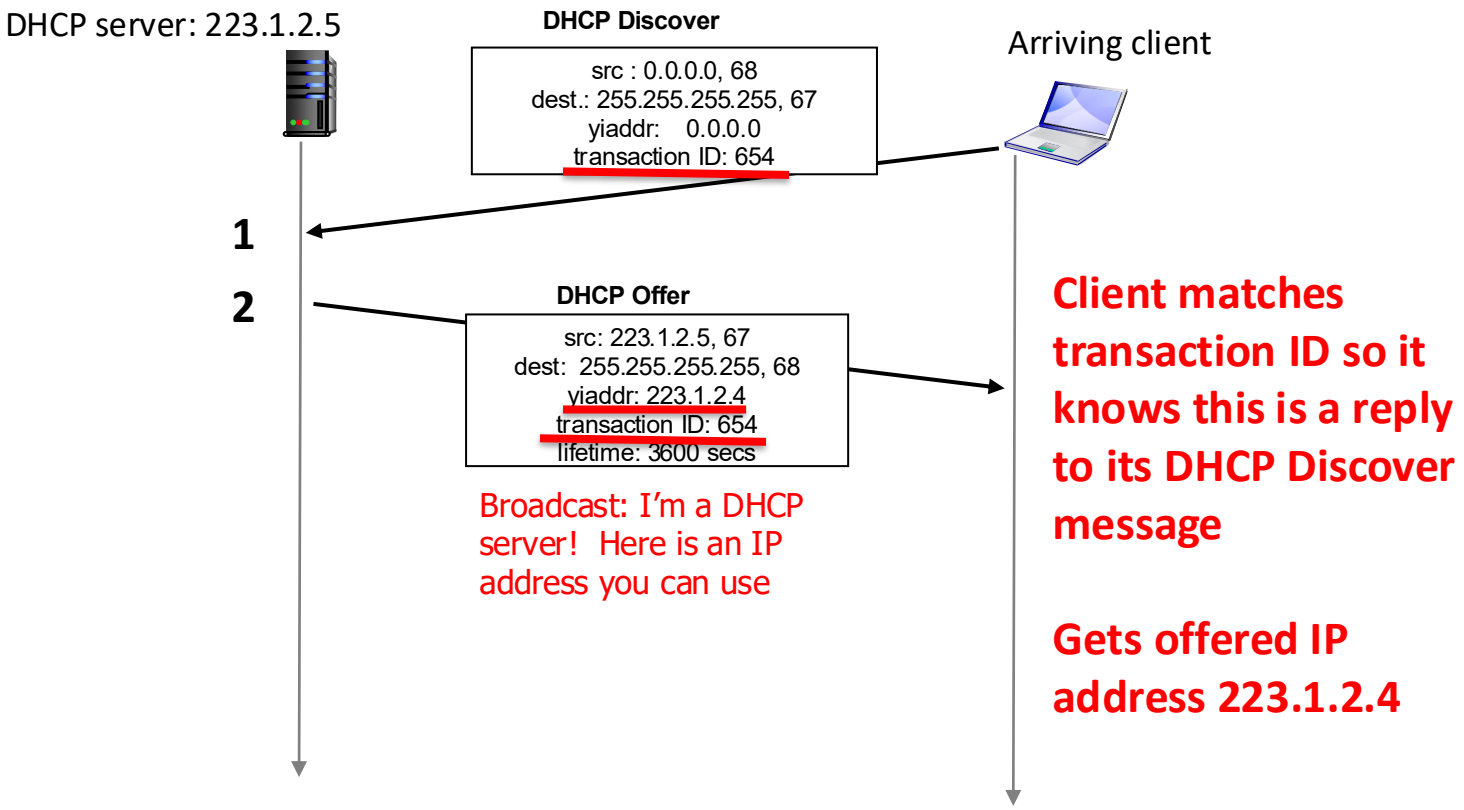
DHCP: 1) Arriving client sends DHCP Discover packet to broadcast



Why is src IP address 0.0.0.0? Why is dst 255.255.255.255?

Adapted from Kurose and Ross: Computer Networking: A Top-Down Approach

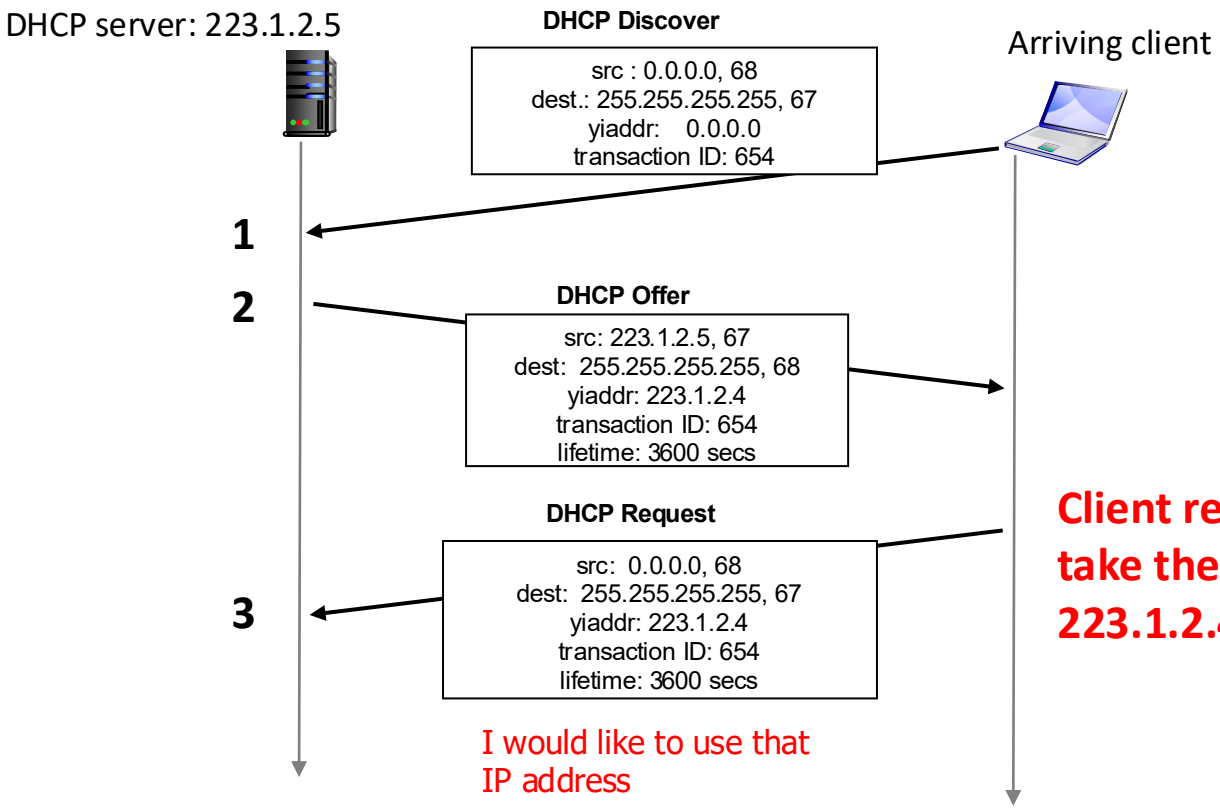
DHCP: 2) DHCP server replies with DHCP Offer packet



Note: clients use port 68, servers use port 67
More on ports soon

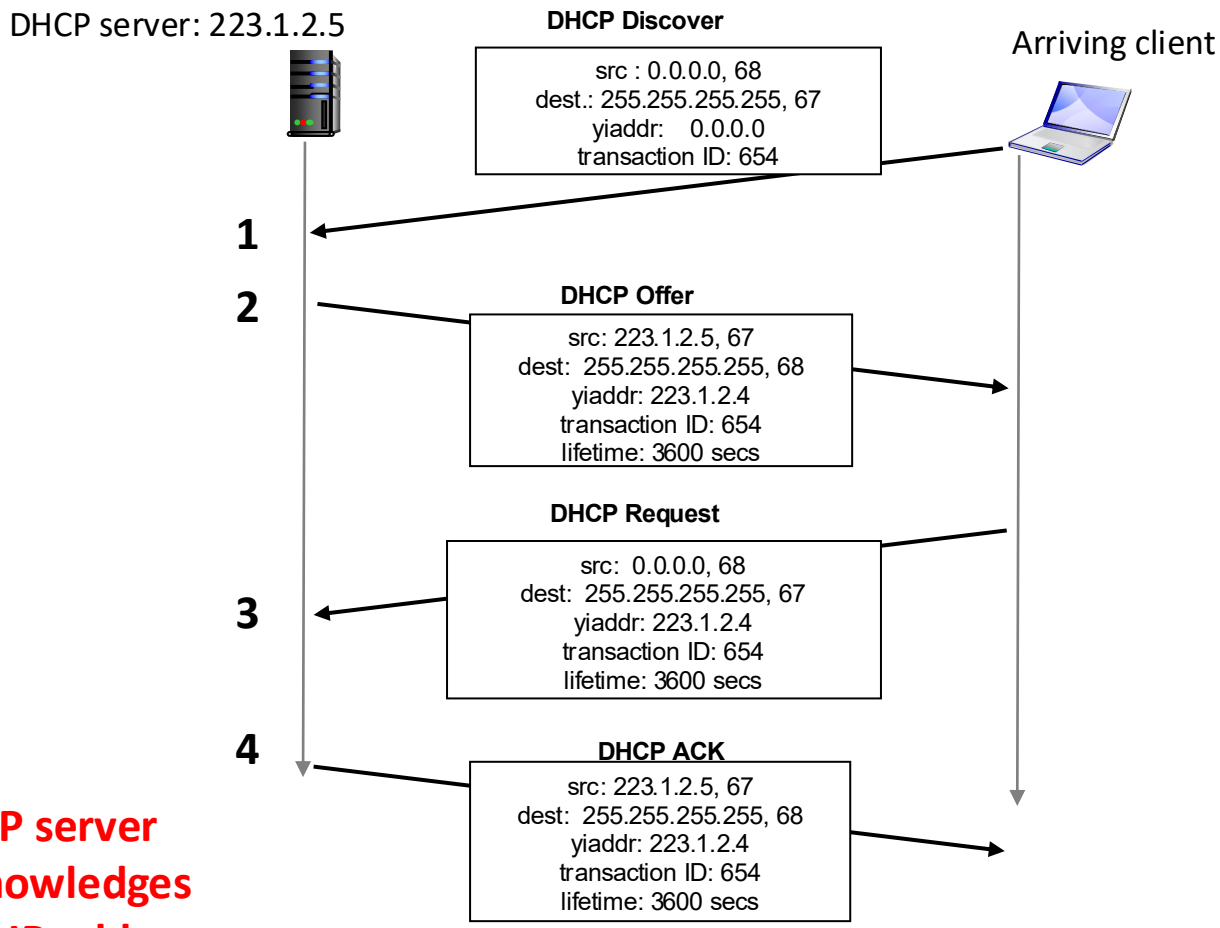
Adapted from Kurose and Ross: Computer Networking: A Top-Down Approach

DHCP: 3) Client replies with DHCP Request to take the offered IP address



Adapted from Kurose and Ross: Computer Networking: A Top-Down Approach

DHCP: 4) DHCP acknowledges client has new IP address with DHCP ACK

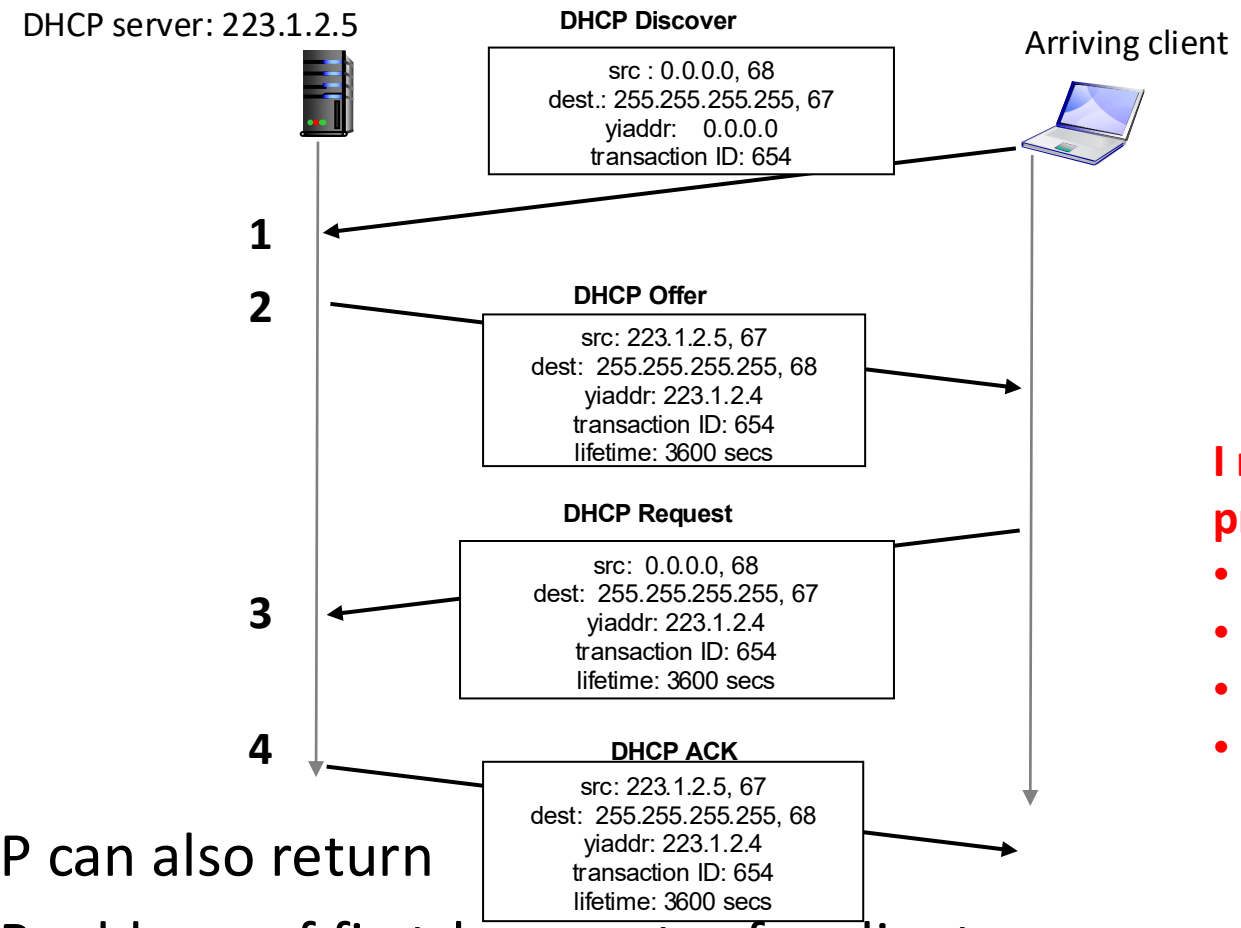


DHCP server acknowledges new IP address

Ok, you've got that IP address

Adapted from Kurose and Ross: Computer Networking: A Top-Down Approach

DHCP: 4) DHCP acknowledges client has new IP address with DHCP ACK



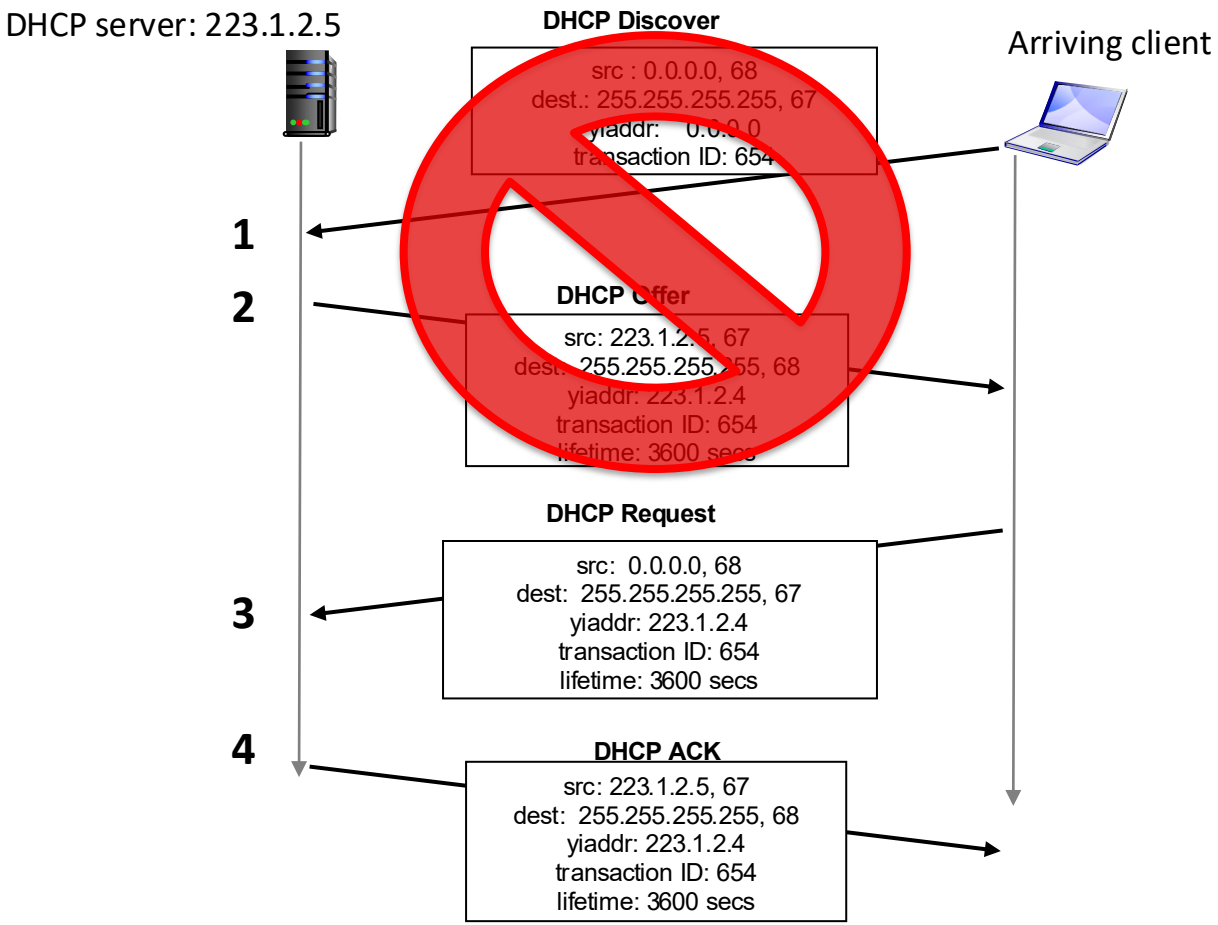
I remember this process as DORA:

- **Discover**
- **Offer**
- **Request**
- **Acknowledge**

DHCP can also return

- IP address of first-hop router for client
- Name and IP address of DNS server
- Network mask (indicating network versus host portion of address)

DHCP: 4) DHCP acknowledges client has new IP address with DHCP ACK



Step 1 and 2 can be skipped "if a client remembers and wishes to reuse a previously allocated network address" [RFC 2131]
Dartmouth remembers your MAC

Demo: DHCP capture with Wireshark

Start Wireshark running

$ **wireshark**

Start new terminal

//find network interface

$ **ifconfig**

```
ens160: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.159.133 netmask 255.255.255.0  broadcast 192.168.159.255
        inet6 fe80::20c:29ff:fe40:521  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:40:05:21  txqueuelen 1000  (Ethernet)
        RX packets 1079351  bytes 1560052499 (1.5 GB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 118582  bytes 20873061 (20.8 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 44  memory 0x3fe00000-3fe20000
```

On the VM, interface is ens160
Layer 2 MAC is 00:0c:29:40:05:21
(yours will likely be different)

Demo: DHCP capture with Wireshark

Start Wireshark running

$ **wireshark**

Start new terminal

//find network interface

$ **ifconfig**

```
ens160: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.159.133 netmask 255.255.255.0  broadcast 192.168.159.255
        inet6 fe80::20c:29ff:fe40:521  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:40:05:21  txqueuelen 1000  (Ethernet)
        RX packets 1079351  bytes 1560052499 (1.5 GB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 118582  bytes 20873061 (20.8 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 44  memory 0x3fe00000-3fe20000
```

//force new dhcp negotiation

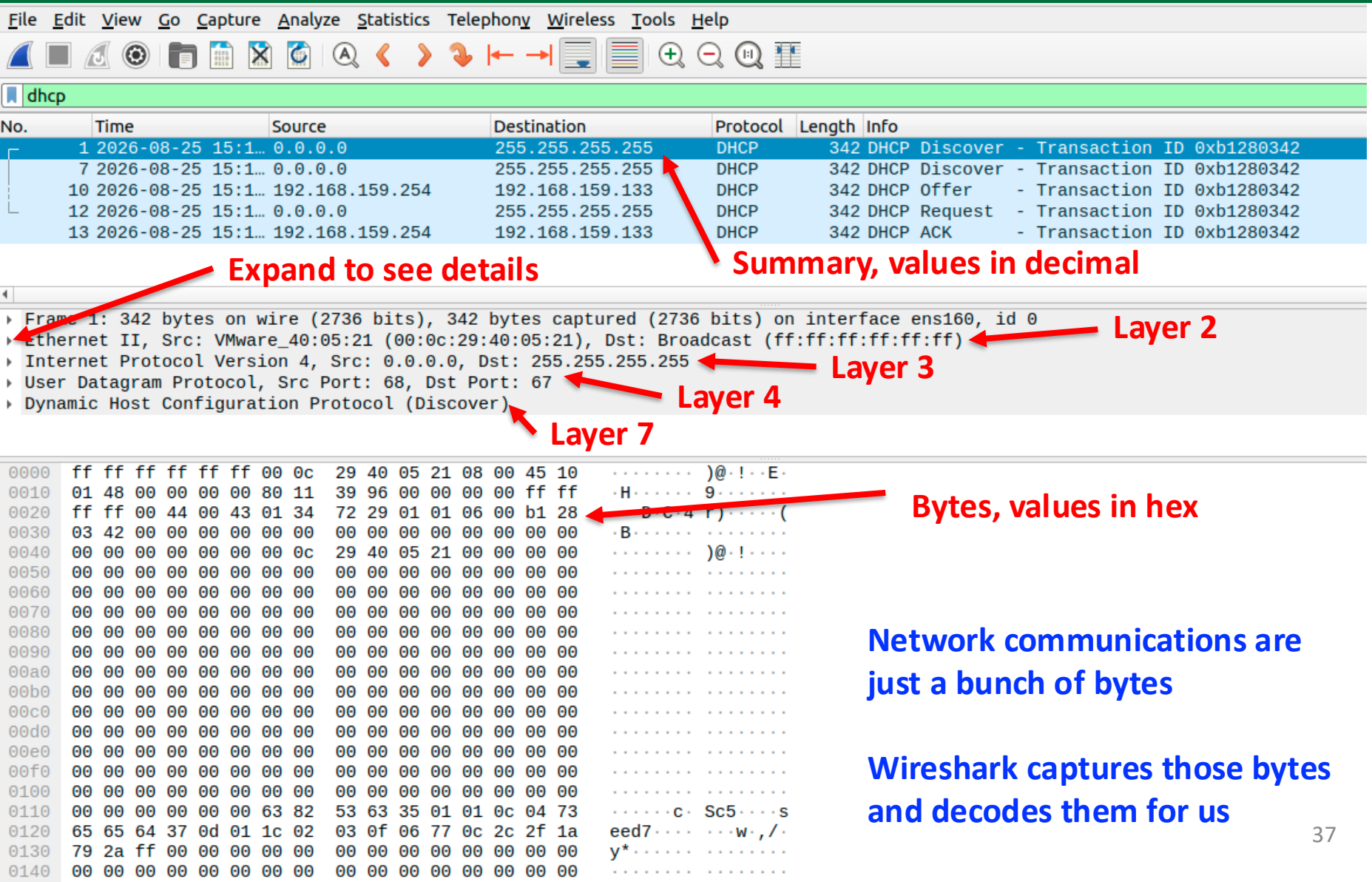
$ **sudo dhclient -r ens160** //drop lease

$ **sudo dhclient ens160** //get new lease



Force laptop to drop existing lease
Then get a new DHCP lease

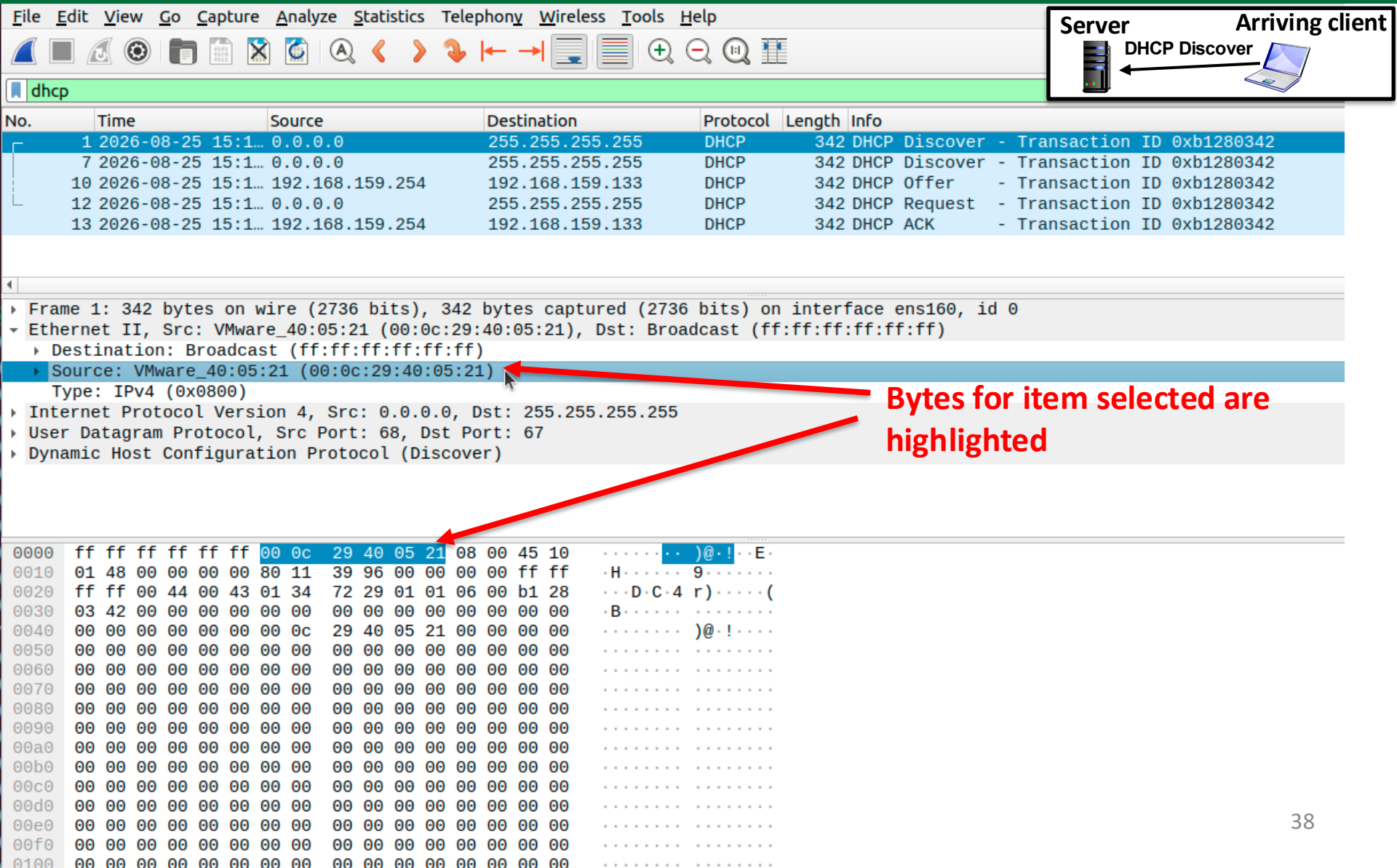
Demo: DHCP capture with Wireshark



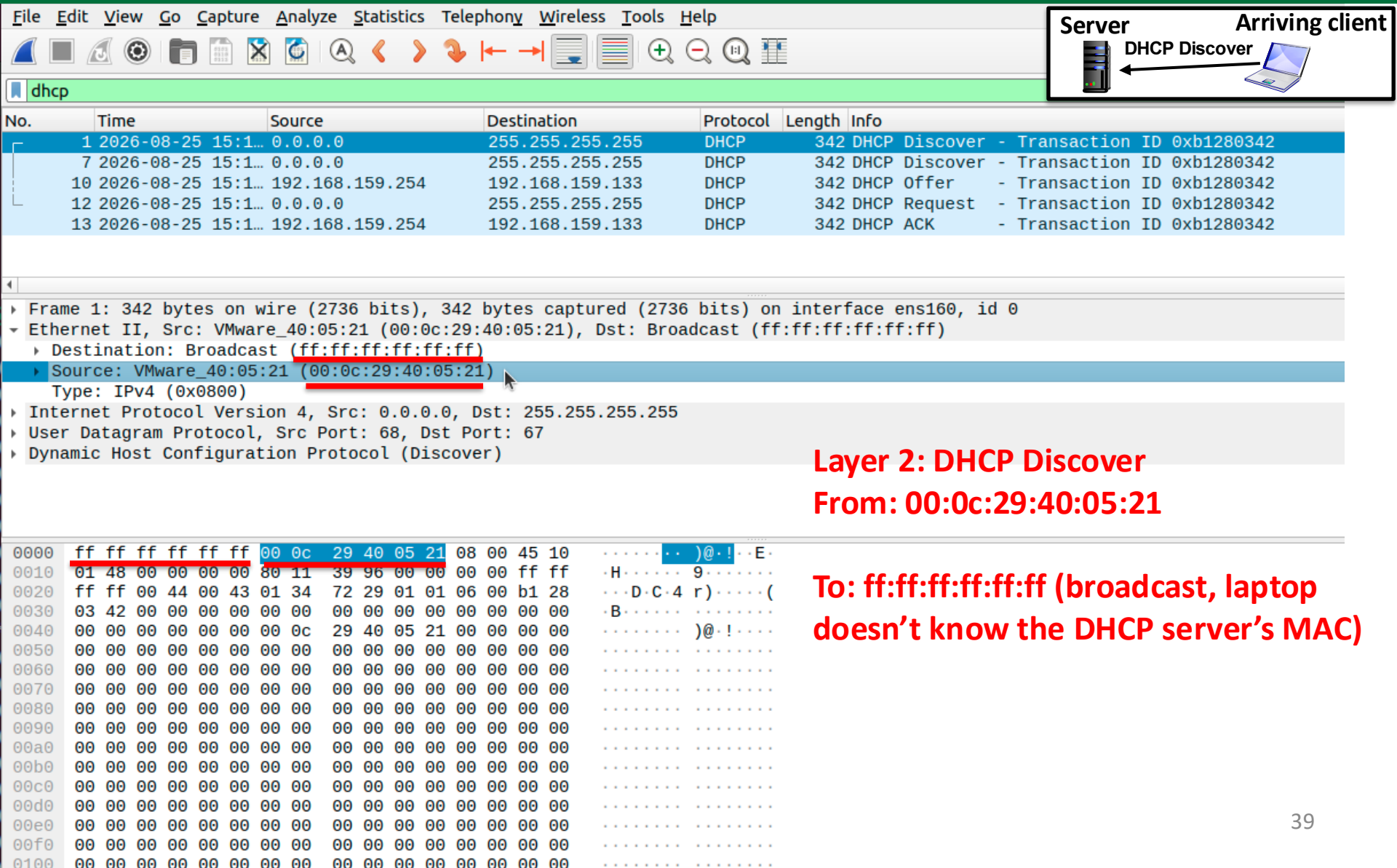
Network communications are just a bunch of bytes

Wireshark captures those bytes and decodes them for us

Demo: DHCP capture with Wireshark



Demo: DHCP capture with Wireshark



Layer 2: DHCP Discover
From: 00:0c:29:40:05:21

To: ff:ff:ff:ff:ff:ff (broadcast, laptop doesn't know the DHCP server's MAC)

Demo: DHCP capture with Wireshark



[SEED Labs] *ens160

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

dhcp

No.	Time	Source	Destination	Protocol	Length	Info
1	2026-08-25 20:2…	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0xf93f5051
4	2026-08-25 20:2…	192.168.159.254	192.168.159.133	DHCP	342	DHCP Offer - Transaction ID 0xf93f5051
5	2026-08-25 20:2…	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0xf93f5051
6	2026-08-25 20:2…	192.168.159.254	192.168.159.133	DHCP	342	DHCP ACK - Transaction ID 0xf93f5051

```
▸ Ethernet II, Src: VMware_40:05:21 (00:0c:29:40:05:21), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▾ Internet Protocol Version 4, Src: 0.0.0.0, Dst: 255.255.255.255
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▸ Differentiated Services Field: 0x10 (DSCP: Unknown, ECN: Not-ECT)
    Total Length: 328
    Identification: 0x0000 (0)
  ▸ Flags: 0x00
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 128
    Protocol: UDP (17)
    Header Checksum: 0x3996 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 0.0.0.0
    Destination Address: 255.255.255.255
```

```
0000  ff ff ff ff ff ff 00 0c  29 40 05 21 08 00 45 10   ........ )@.!..E.
0010  01 48 00 00 00 00 80 11  39 96 00 00 00 00 ff ff   .H...... 9.......
0020  ff ff 00 44 00 43 01 34  aa 70 01 01 06 00 f9 3f   ...D.C.4 .p.....?
0030  50 51 00 00 00 00 00 00  00 00 00 00 00 00 00 00   PQ...... ........
0040  00 00 00 00 00 00 00 0c  29 40 05 21 00 00 00 00   ........ )@.!....
0050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0080  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0090  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
```

Layer 3: DHCP Discover
From: 0.0.0.0 (laptop does not have an IP address yet!)

To: 255.255.255.255 (broadcast,0xff:ff:ff:ff, laptop doesn't know the DHCP server's IP address yet)

Demo: DHCP capture with Wireshark



[SEED Labs] *ens160

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

dhcp

No.	Time	Source	Destination	Protocol	Length	Info
1	2026-08-25 20:2…	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0xf93f5051
4	2026-08-25 20:2…	192.168.159.254	192.168.159.133	DHCP	342	DHCP Offer - Transaction ID 0xf93f5051
5	2026-08-25 20:2…	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0xf93f5051
6	2026-08-25 20:2…	192.168.159.254	192.168.159.133	DHCP	342	DHCP ACK - Transaction ID 0xf93f5051

```
▸ Ethernet II, Src: VMware_40:05:21 (00:0c:29:40:05:21), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▾ Internet Protocol Version 4, Src: 0.0.0.0, Dst: 255.255.255.255
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▸ Differentiated Services Field: 0x10 (DSCP: Unknown, ECN: Not-ECT)
    Total Length: 328
    Identification: 0x0000 (0)
  ▸ Flags: 0x00
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 128
    Protocol: UDP (17)
    Header Checksum: 0x3996 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 0.0.0.0
    Destination Address: 255.255.255.255
```

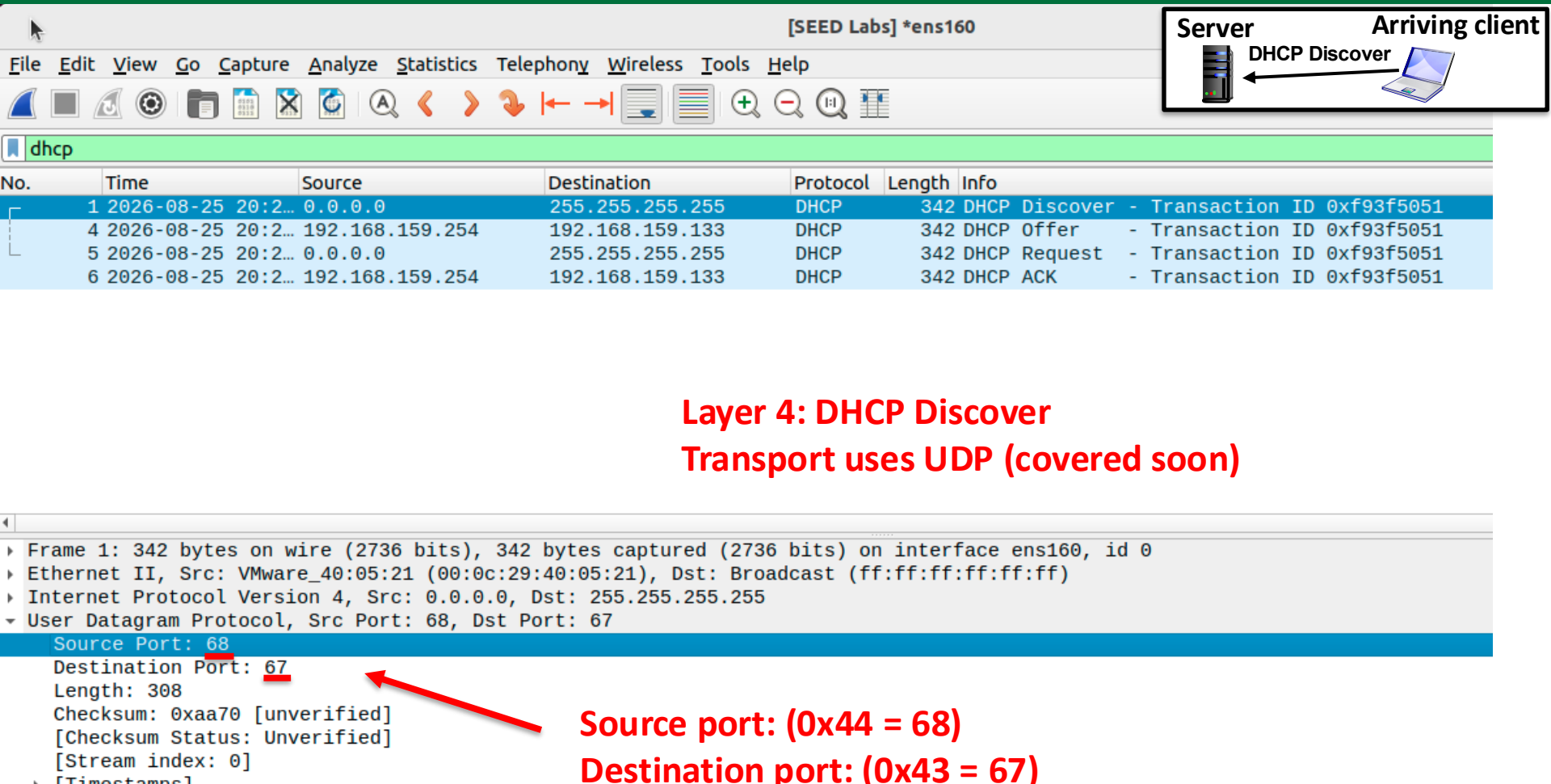
```
0000  ff ff ff ff ff ff 00 0c  29 40 05 21 08 00 45 10   ........ )@.!..E.
0010  01 48 00 00 00 00 80 11  39 96 00 00 00 00 ff ff   .H...... 9.......
0020  ff ff 00 44 00 43 01 34  aa 70 01 01 06 00 f9 3f   ...D.C.4 .p.....?
0030  50 51 00 00 00 00 00 00  00 00 00 00 00 00 00 00   PQ...... ........
0040  00 00 00 00 00 00 00 0c  29 40 05 21 00 00 00 00   ........ )@.!....
0050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0080  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0090  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
```

Layer 3: DHCP Discover

Time to live 128 hops (128 = 0x80)

Layer 4 contains a UDP packet (17 = 0x11)

Demo: DHCP capture with Wireshark



Layer 4: DHCP Discover

Transport uses UDP (covered soon)

Source port: (0x44 = 68)

Destination port: (0x43 = 67)

Demo: DHCP capture with Wireshark



File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

dhcp

No.	Time	Source	Destination	Protocol	Length	Info
1	2026-08-25 15:1…	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0xb1280342
7	2026-08-25 15:1…	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0xb1280342
10	2026-08-25 15:1…	192.168.159.254	192.168.159.133	DHCP	342	DHCP Offer - Transaction ID 0xb1280342
12	2026-08-25 15:1…	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0xb1280342
13	2026-08-25 15:1…	192.168.159.254	192.168.159.133	DHCP	342	DHCP ACK - Transaction ID 0xb1280342

```
▸ Frame 1: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits) on interface ens160, id 0
▸ Ethernet II, Src: VMware_40:05:21 (00:0c:29:40:05:21), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▸ Internet Protocol Version 4, Src: 0.0.0.0, Dst: 255.255.255.255
▸ User Datagram Protocol, Src Port: 68, Dst Port: 67
▸ Dynamic Host Configuration Protocol (Discover)
```

```
0000  ff ff ff ff ff ff 00 0c  29 40 05 21 08 00 45 10   ........ )@.!..E.
0010  01 48 00 00 00 00 80 11  39 96 00 00 00 00 ff ff   .H...... 9.......
0020  ff ff 00 44 00 43 01 34  72 29 01 01 06 00 b1 28   ...D.C.4 r).....(
0030  03 42 00 00 00 00 00 00  00 00 00 00 00 00 00 00   .B...... ........
0040  00 00 00 00 00 00 00 0c  29 40 05 21 00 00 00 00   ........ )@.!....
0050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0080  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0090  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00d0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00e0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0100  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0110  00 00 00 00 00 00 63 82  53 63 35 01 01 0c 04 73   ......c. Sc5....s
0120  65 65 64 37 0d 01 1c 02  03 0f 06 77 0c 2c 2f 1a   eed7.... ...w.,/.
0130  79 2a ff 00 00 00 00 00  00 00 00 00 00 00 00 00   y*...... ........
0140  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
```

Layer 7: DHCP Discover

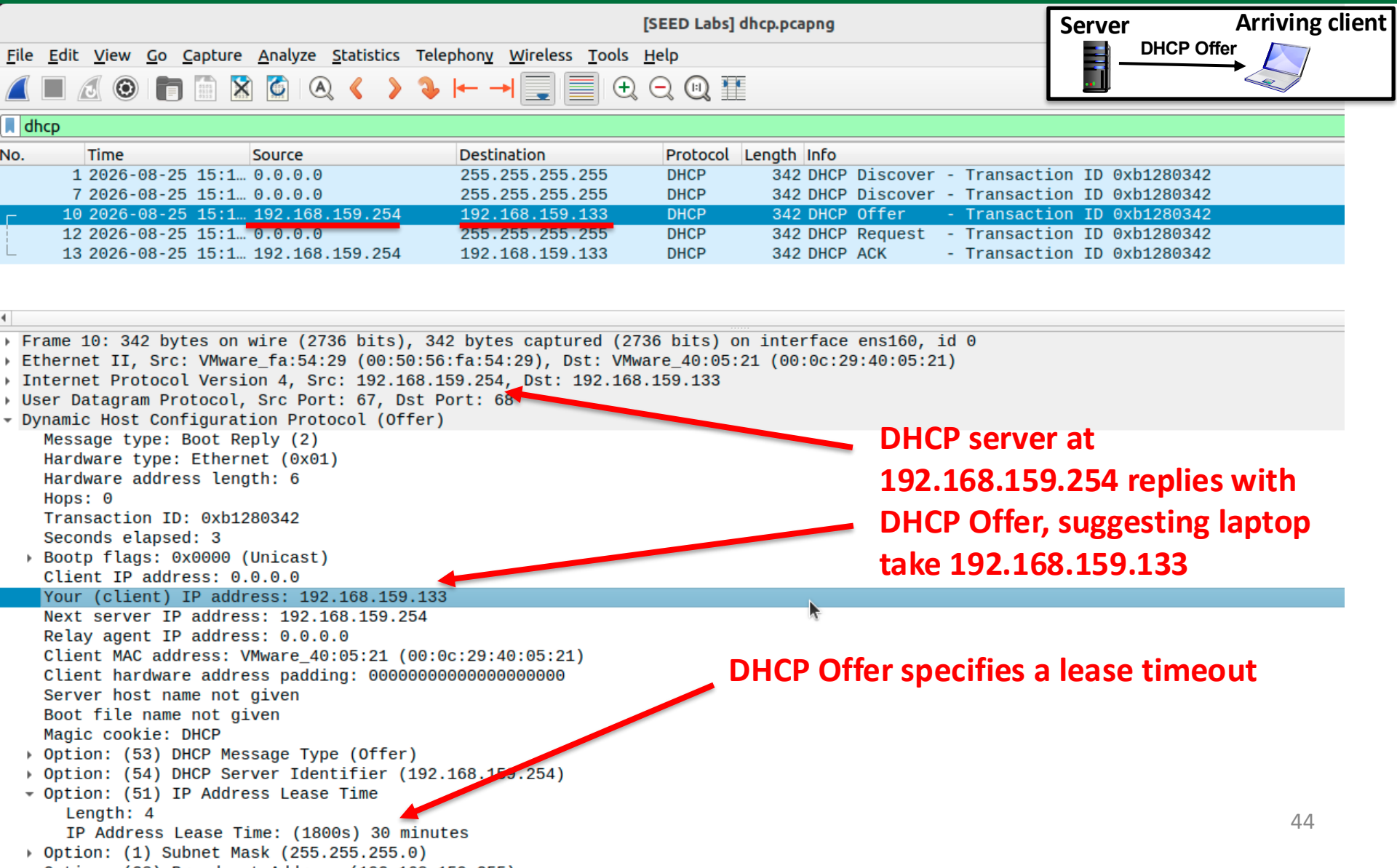
01 = DHCP Discover message

02 = Offer

03 = Request

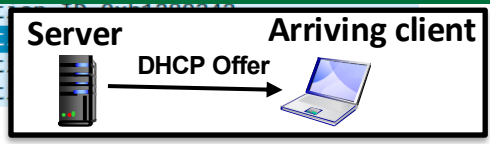
05 = ACK

Demo: DHCP capture with Wireshark



Demo: DHCP capture with Wireshark

```
     7 2026-08-25 15:1… 0.0.0.0          255.255.255.255    DHCP    342 DHCP Discover - Transact
    10 2026-08-25 15:1… 192.168.159.254  192.168.159.133    DHCP    342 DHCP Offer    - Transact
    12 2026-08-25 15:1… 0.0.0.0          255.255.255.255    DHCP    342 DHCP Request  - Transact
    13 2026-08-25 15:1… 192.168.159.254  192.168.159.133    DHCP    342 DHCP ACK      - Transact
```




```
▸ Frame 10: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits) on interface ens160, id 0
▸ Ethernet II, Src: VMware_fa:54:29 (00:50:56:fa:54:29), Dst: VMware_40:05:21 (00:0c:29:40:05:21)
▸ Internet Protocol Version 4, Src: 192.168.159.254, Dst: 192.168.159.133
▸ User Datagram Protocol, Src Port: 67, Dst Port: 68
▾ Dynamic Host Configuration Protocol (Offer)
    Message type: Boot Reply (2)
    Hardware type: Ethernet (0x01)
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0xb1280342
    Seconds elapsed: 3
  ▸ Bootp flags: 0x0000 (Unicast)
    Client IP address: 0.0.0.0
    Your (client) IP address: 192.168.159.133
    Next server IP address: 192.168.159.254
    Relay agent IP address: 0.0.0.0
    Client MAC address: VMware_40:05:21 (00:0c:29:40:05:21)
    Client hardware address padding: 00000000000000000000
    Server host name not given
    Boot file name not given
    Magic cookie: DHCP
  ▸ Option: (53) DHCP Message Type (Offer)
  ▸ Option: (54) DHCP Server Identifier (192.168.159.254)
  ▾ Option: (51) IP Address Lease Time
      Length: 4
      IP Address Lease Time: (1800s) 30 minutes
  ▸ Option: (1) Subnet Mask (255.255.255.0)
  ▸ Option: (28) Broadcast Address (192.168.159.255)
  ▾ Option: (3) Router
      Length: 4
      Router: 192.168.159.2
```

```
0030  03 42 00 03 00 00 00 00  00 00 c0 a8 9f 85 c0 a8   ·B······ ······
0040  9f fe 00 00 00 00 00 0c  29 40 05 21 00 00 00 00   ········ )@·!····
0050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
```

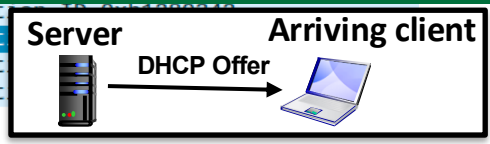
DHCP server at 192.168.159.254 replies with DHCP Offer, suggesting laptop take 192.168.159.133

DHCP Offer specifies a lease timeout

Also gives the IP address of the local area network's router

Demo: DHCP capture with Wireshark

```
    7 2026-08-25 15:1… 0.0.0.0          255.255.255.255    DHCP   342 DHCP Discover - Transact
   10 2026-08-25 15:1… 192.168.159.254  192.168.159.133    DHCP   342 DHCP Offer    - Transact
   12 2026-08-25 15:1… 0.0.0.0          255.255.255.255    DHCP   342 DHCP Request  - Transact
   13 2026-08-25 15:1… 192.168.159.254  192.168.159.133    DHCP   342 DHCP ACK      - Transact
```




```
Frame 10: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits) on interface ens160, id 0
Ethernet II, Src: VMware_fa:54:29 (00:50:56:fa:54:29), Dst: VMware_40:05:21 (00:0c:29:40:05:21)
Internet Protocol Version 4, Src: 192.168.159.254, Dst: 192.168.159.133
User Datagram Protocol, Src Port: 67, Dst Port: 68
Dynamic Host Configuration Protocol (Offer)
    Message type: Boot Reply (2)
    Hardware type: Ethernet (0x01)
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0xb1280342
    Seconds elapsed: 3
  ▸ Bootp flags: 0x0000 (
    Client IP address: 0.
    Your (client) IP addr
    Next server IP addres
    Relay agent IP addres
    Client MAC address: V
    Client hardware addre
    Server host name not
    Boot file name not gi
    Magic cookie: DHCP
  ▸ Option: (53) DHCP Mes
  ▸ Option: (54) DHCP Server Identifier (192.168.159.254)
  ▾ Option: (51) IP Address Lease Time
      Length: 4
      IP Address Lease Time: (1800s) 30 minutes
  ▸ Option: (1) Subnet Mask (255.255.255.0)
  ▸ Option: (28) Broadcast Address (192.168.159.255)
  ▾ Option: (3) Router
      Length: 4
      Router: 192.168.159.2
```

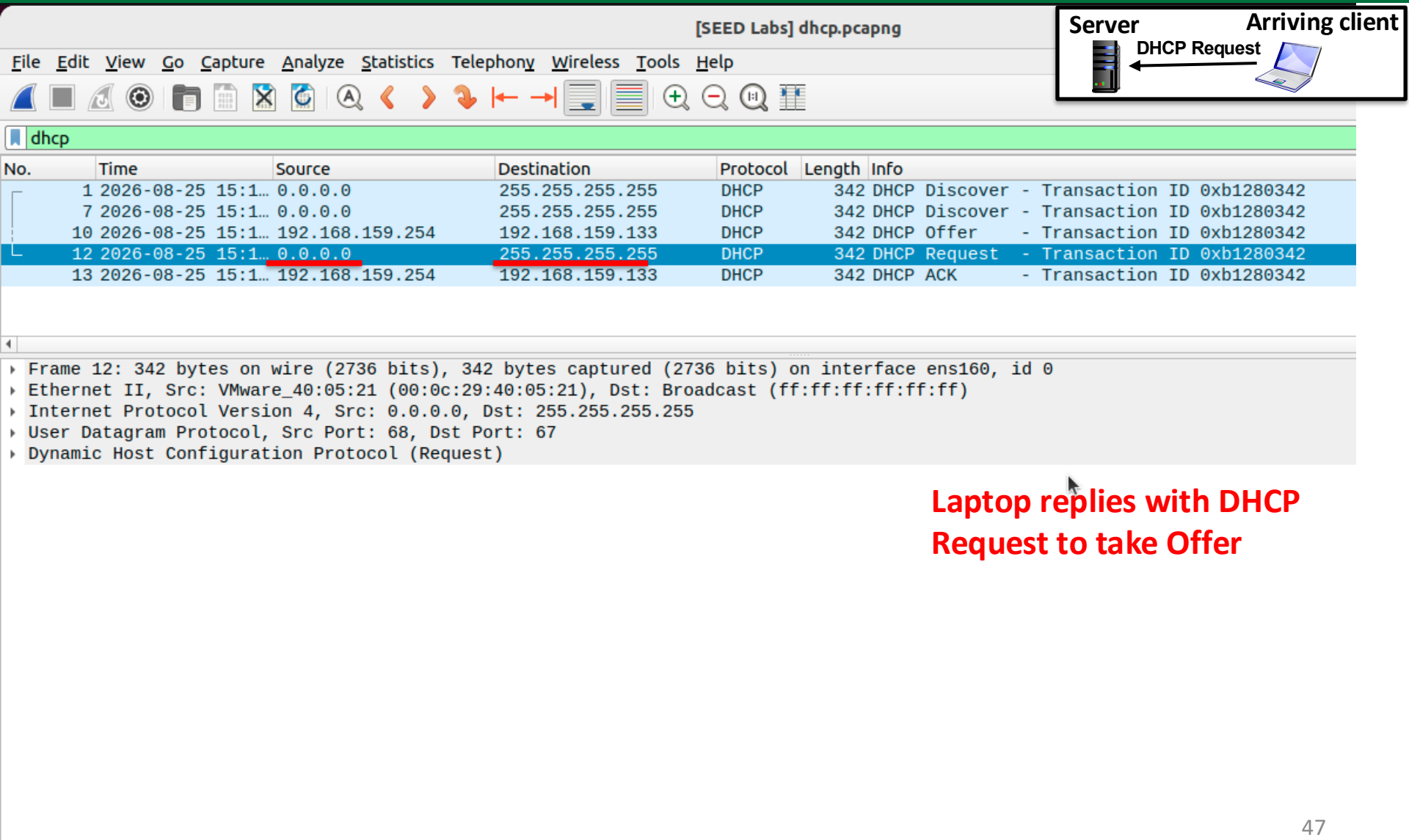
Next hop router IP address shown as default (but router's MAC is unknown)

```
$ ip route
default via 192.168.159.2 dev ens160 proto dhcp src 192.168.159.133 metric 100
172.17.0.0/16 dev docker0 proto kernel scope link src 172.17.0.1 linkdown
192.168.159.0/24 dev ens160 proto kernel scope link src 192.168.159.133 metric 100
192.168.159.2 dev ens160 proto dhcp scope link src 192.168.159.133 metric 100
```

Also gives the IP address of the local area network's router

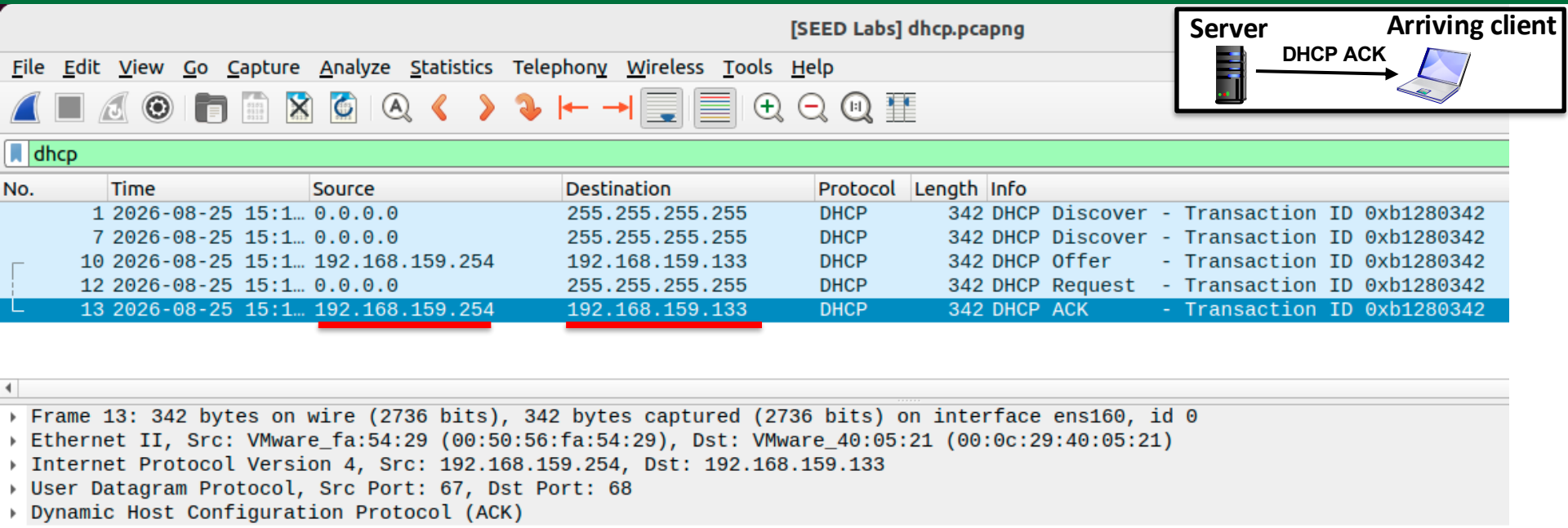
```
0030  03 42 00 03 00 00 00 00  00 00 c0 a8 9f 85 c0 a8   ·B······ ······
0040  9f fe 00 00 00 00 00 0c  29 40 05 21 00 00 00 00   ········ )@·!····
0050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
```

Demo: DHCP capture with Wireshark



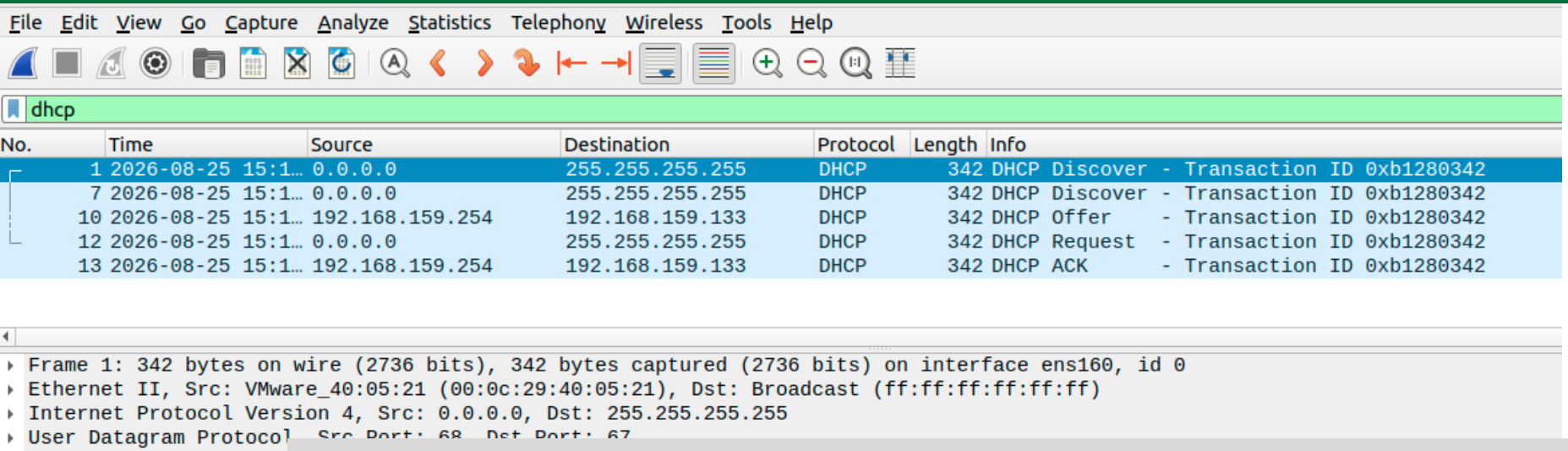
Laptop replies with DHCP Request to take Offer

Demo: DHCP capture with Wireshark



Server replies with ACK that laptop took offered IP address of 192.168.159.133

Demo: DHCP capture with Wireshark




```
$ ifconfig
ens160: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.159.133  netmask 255.255.255.0  broadcast 192.168.159.255
        inet6 fe80::20c:29ff:fe40:521  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:40:05:21  txqueuelen 1000  (Ethernet)
        RX packets 1079736  bytes 1560109088 (1.5 GB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 118827  bytes 20899447 (20.8 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 44  memory 0x3fe00000-3fe20000
```

ifconfig confirms host's IP address

Also know router's IP address and lease time out

Agenda

1. Network layer overview
2. DHCP
3. Ping, traceroute, and ARP
4. Exercises

ICMP: Internet Control Message Protocol

Ping

- ping <IP address>
- Used by hosts and routers to communicate network-level information
  - Error reporting: unreachable host, network, port, protocol
- Ping uses ICMP echo request/reply
  - ICMP messages carried in IP (Layer 3) datagrams
  - Reply returns:
    - Type
    - Code
    - First 8 bytes of IP payload

ping 8.8.8.8 (Google's DNS server)
Why that IP address?
Easy to remember!

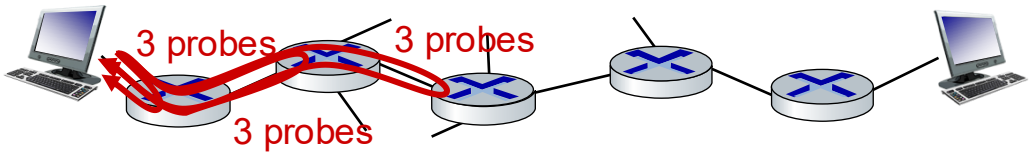
Shows roundtrip time to destination and back

Type	Code	description
0	0	echo reply (ping)
3	0	dest. network unreachable
3	1	dest host unreachable
3	2	dest protocol unreachable
3	3	dest port unreachable
3	6	dest network unknown
3	7	dest host unknown
4	0	source quench (congestion control - not used)
8	0	echo request (ping)
9	0	route advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header

You will write your own ping program in Lab 1

Adapted from Kurose and Ross: Computer Networking: A Top-Down Approach

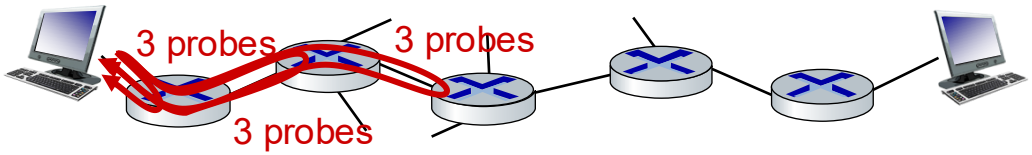
Traceroute and ICMP



- Ping shows if a host is reachable, but does not give the route to the host
- Can you determine the route packets take to the destination?
- Three useful pieces of information:
  1. Each ICMP ping request has a time to live (TTL) = max number of hops
  2. Routers decrement TTL when passing to next router
  3. Routers are *supposed* to return “TTL expired” (type 11, code 0) with router’s name if TTL goes to 0

Adapted from Kurose and Ross: Computer Networking: A Top-Down Approach

Traceroute and ICMP



You will write your own traceroute program in Lab 1

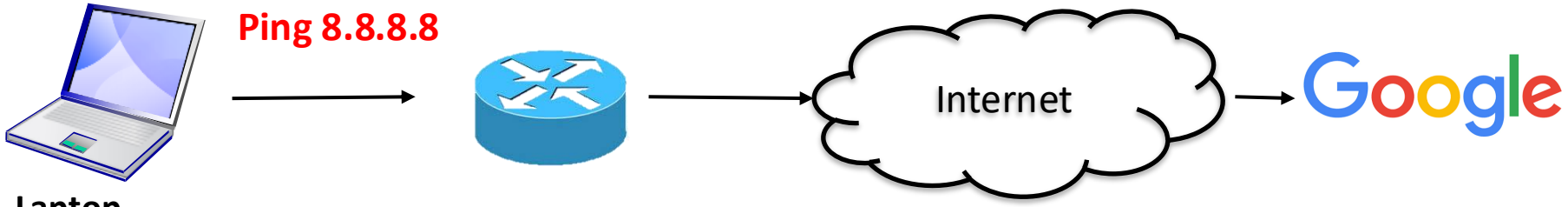
- Source sends sets of UDP segments to destination
  - 1st set has TTL =1, 2nd set has TTL=2, etc.
- Datagram in *n*th set arrives to nth router:
  - Router discards datagram and sends source ICMP message (type 11, code 0 which is TTL expired)
  - ICMP message possibly includes name of router & IP address
- When ICMP message arrives at source: record RTT

Stopping criteria:

- UDP segment eventually arrives at destination host
- Destination returns ICMP "port unreachable" message (type 3, code 3)
- Source stops sending
- Reach limit (Dartmouth blocks)

Adapted from Kurose and Ross: Computer Networking: A Top-Down Approach

Client has IP address thanks to DHCP, now wants to reach the Internet



Laptop

- Layer 2:
  - Knows its own MAC address (burned into NIC)
  - Does not know Router's MAC address
- Layer 3:
  - Got Router's IP address from DHCP
  - Ping Google (ICMP) at 8.8.8.8

Ping is a good way to test network connectivity

- **Send ICMP Echo Request and listen for ICMP Echo Reply**

Laptop can route ping ICMP message to the Internet via the Router

- **Router connects Local Area Network (LAN) to other networks (Internet here)**

Problem: laptop knows Router's IP address, but not its MAC

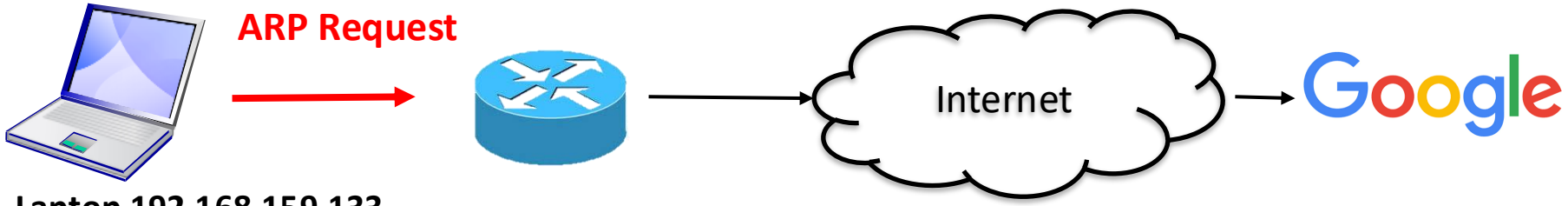
- **Laptop does not know what MAC to use for Layer 2**
- **We need to provide a destination MAC at Layer 2**

Solution: Address Resolution Protocol (ARP)

- **Laptop sends Who Has <IP address> message**
- **Router replies with its MAC**

(Soon we will see how to use Domain Name Service (DNS) so we can remember google.com, not hard to remember IP addresses like 8.8.8.8)

Client has IP address thanks to DHCP, now wants to reach the Internet

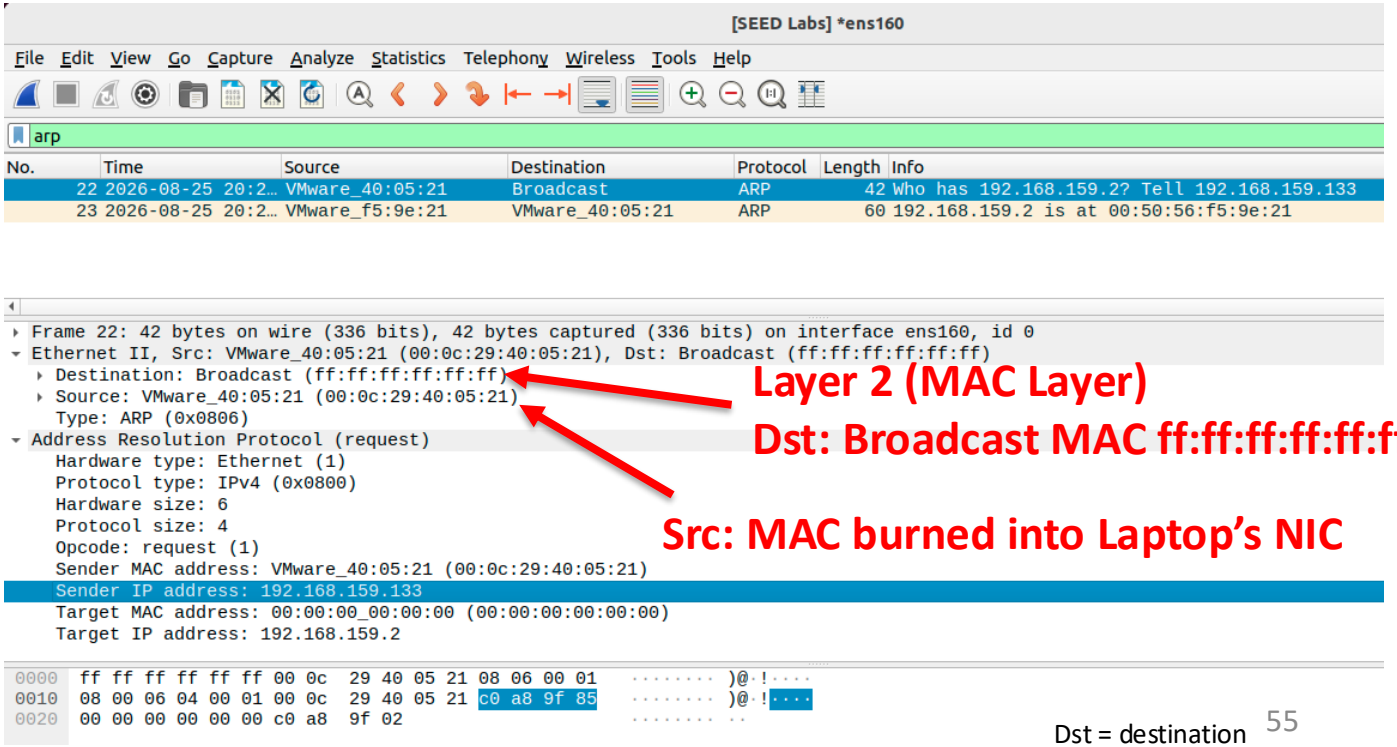


Laptop 192.168.159.133

- Layer 2:
  - Knows its own MAC address (burned into NIC)
  - Does not know Router's MAC address
- Layer 3:
  - Got Router's IP address from DHCP
  - Ping Google (ICMP) at 8.8.8.8

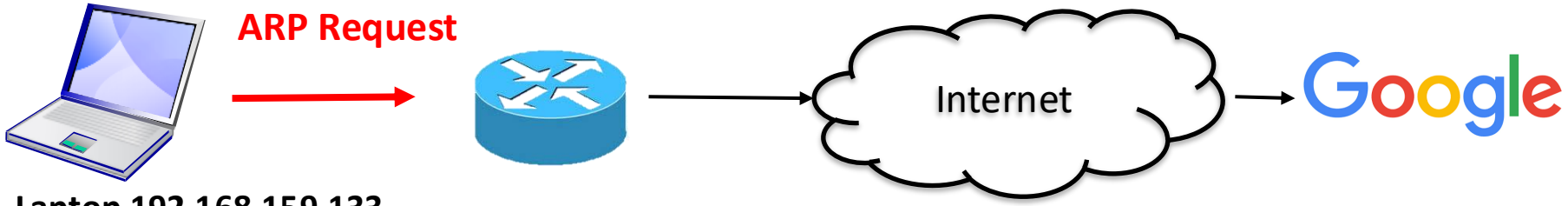
Laptop broadcasts an ARP Who Has message

- **Who Has 192.168.159.2? Tell 192.168.159.133**



Dst = destination
Src = source

Client has IP address thanks to DHCP, now wants to reach the Internet

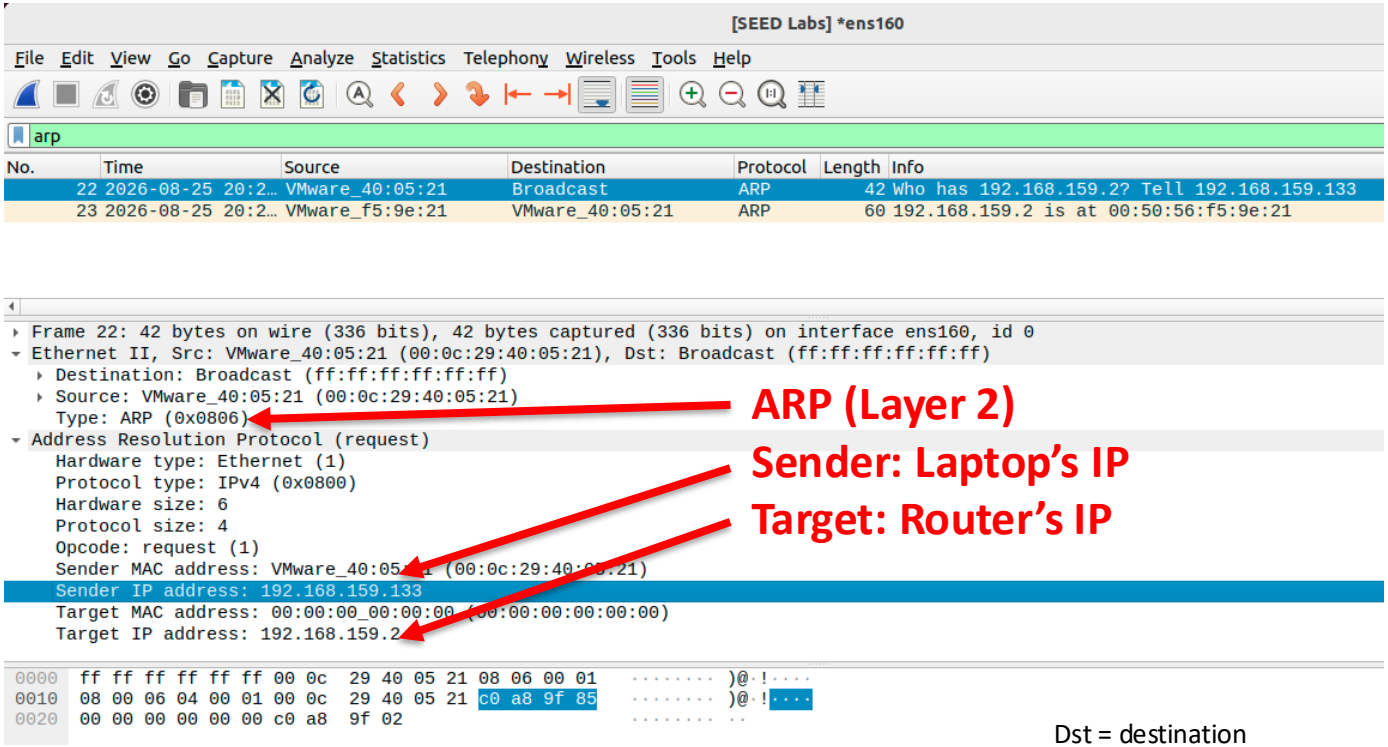


Laptop 192.168.159.133

- Layer 2:
  - Knows its own MAC address (burned into NIC)
  - Does not know Router's MAC address
- Layer 3:
  - Got Router's IP address from DHCP
  - Ping Google (ICMP) at 8.8.8.8

Laptop broadcasts an ARP Who Has message

- **Who Has 192.168.159.2? Tell 192.168.159.133**

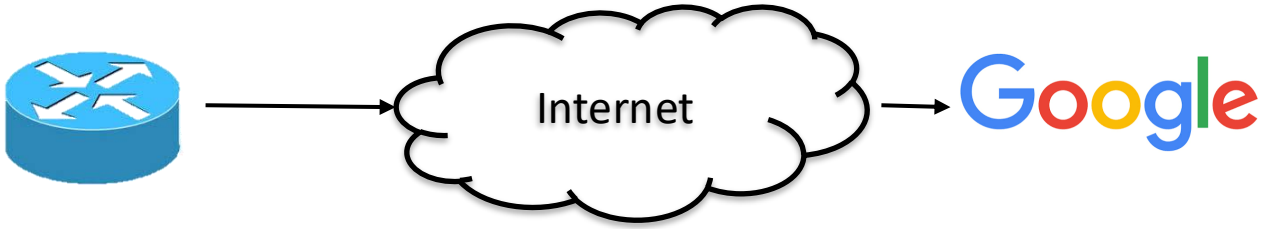


Dst = destination
Src = source

Client has IP address thanks to DHCP, now wants to reach the Internet



ARP Reply

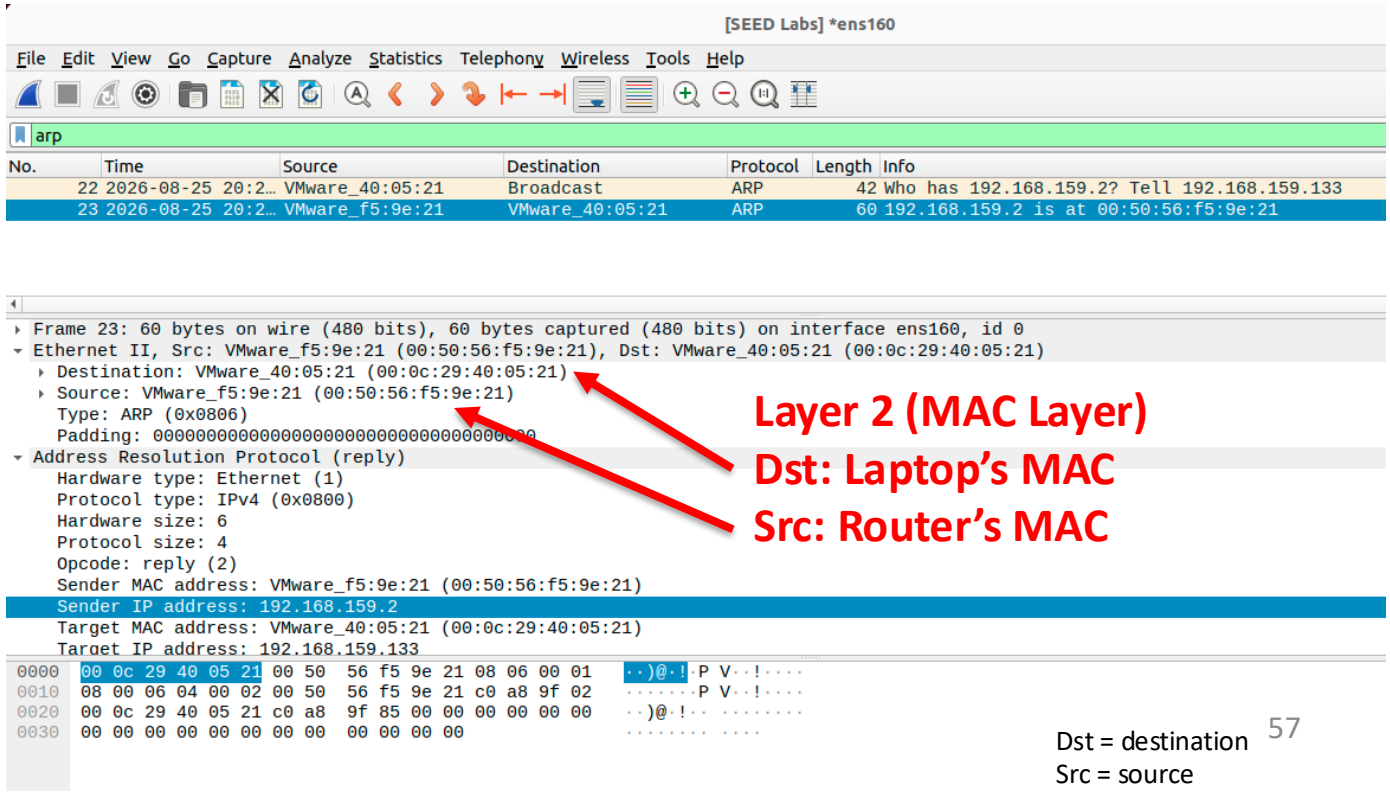


Laptop 192.168.159.133

- Layer 2:
  - Knows its own MAC address (burned into NIC)
  - Does not know Router's MAC address
- Layer 3:
  - Got Router's IP address from DHCP
  - Ping Google (ICMP) at 8.8.8.8

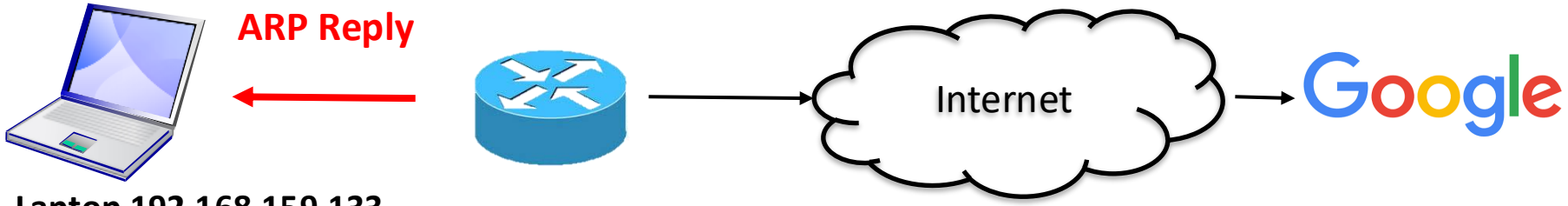
Router sees its IP address in Who Has and replies with its MAC address

- **192.168.159.2 is at 00:50:56:f5:9e:21**



Dst = destination
Src = source

Client has IP address thanks to DHCP, now wants to reach the Internet

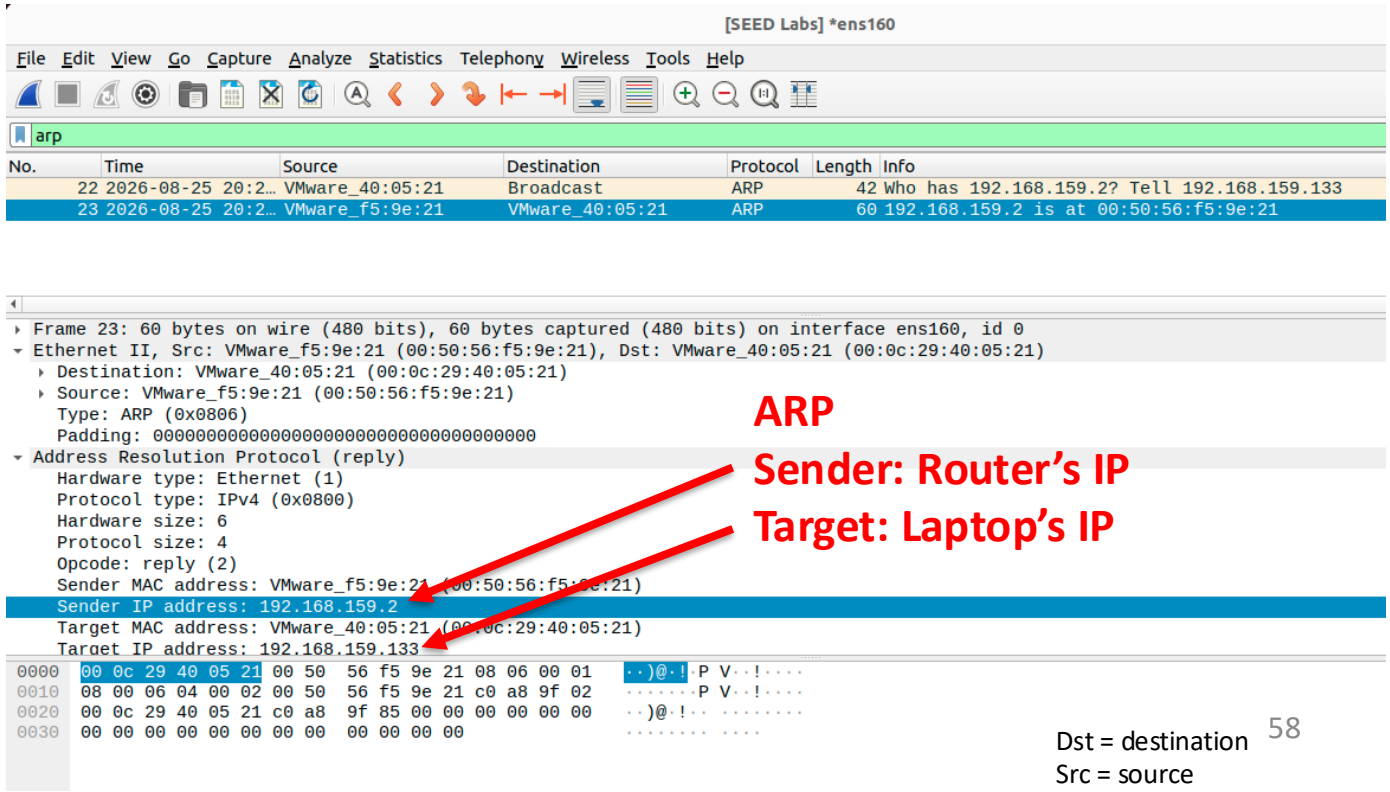


Laptop 192.168.159.133

- Layer 2:
  - Knows its own MAC address (burned into NIC)
  - Does not know Router's MAC address
- Layer 3:
  - Got Router's IP address from DHCP
  - Ping Google (ICMP) at 8.8.8.8

Router sees its IP address in Who Has and replies with its MAC address

- **192.168.159.2 is at 00:50:56:f5:9e:21**

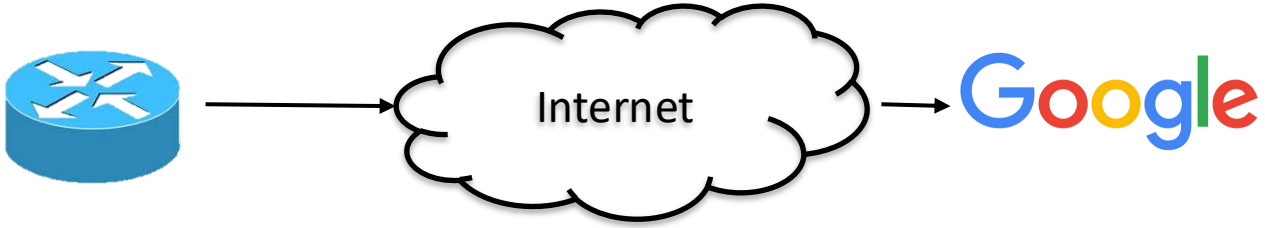


Dst = destination
Src = source

Client has IP address thanks to DHCP, now wants to reach the Internet



ARP Reply

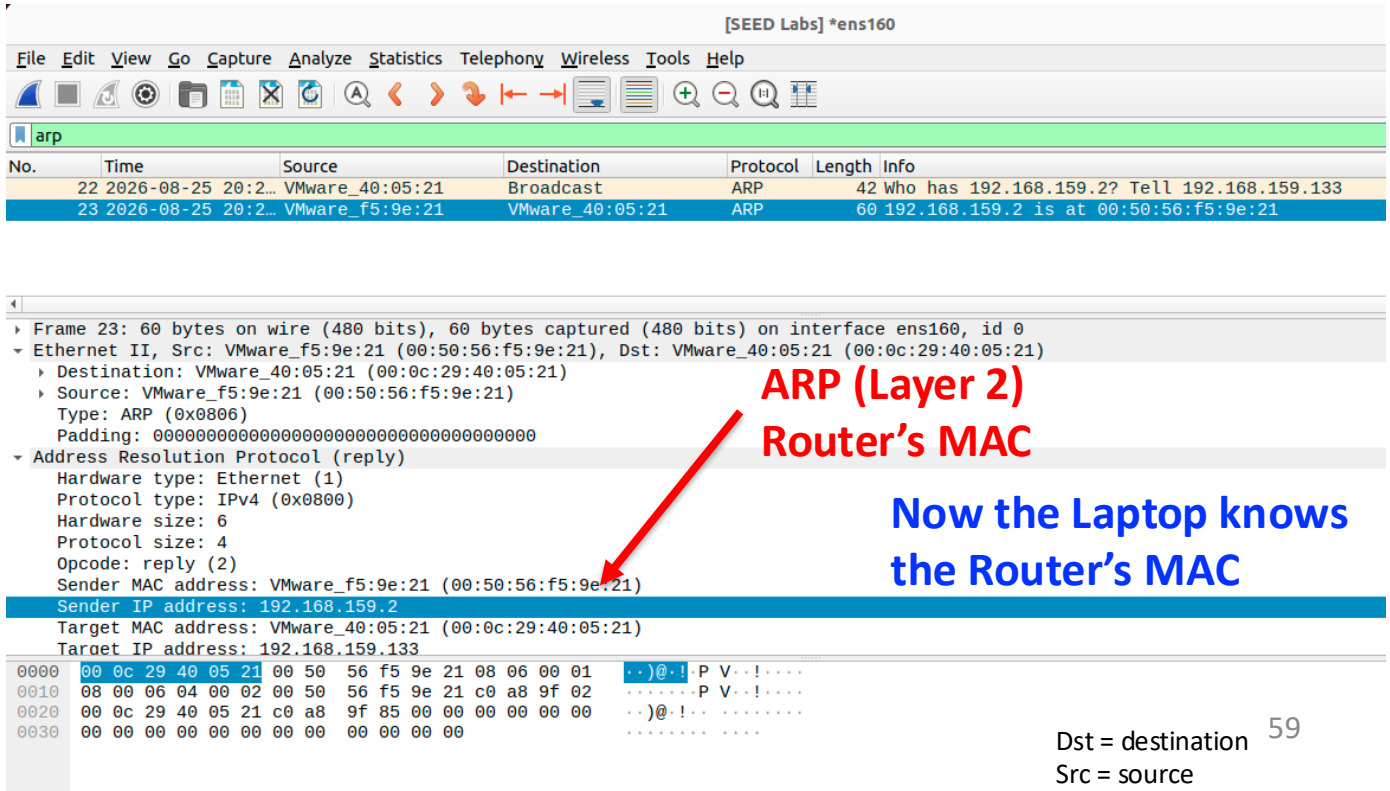


Laptop 192.168.159.133

- Layer 2:
  - Knows its own MAC address (burned into NIC)
  - Does not know Router's MAC address
- Layer 3:
  - Got Router's IP address from DHCP
  - Ping Google (ICMP) at 8.8.8.8

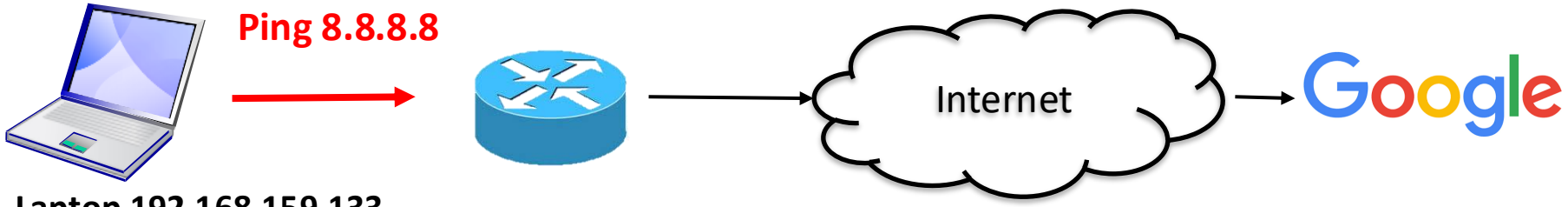
Router sees its IP address in Who Has and replies with its MAC address

- **192.168.159.2 is at 00:50:56:f5:9e:21**



Dst = destination
Src = source

Client has IP address thanks to DHCP, now wants to reach the Internet



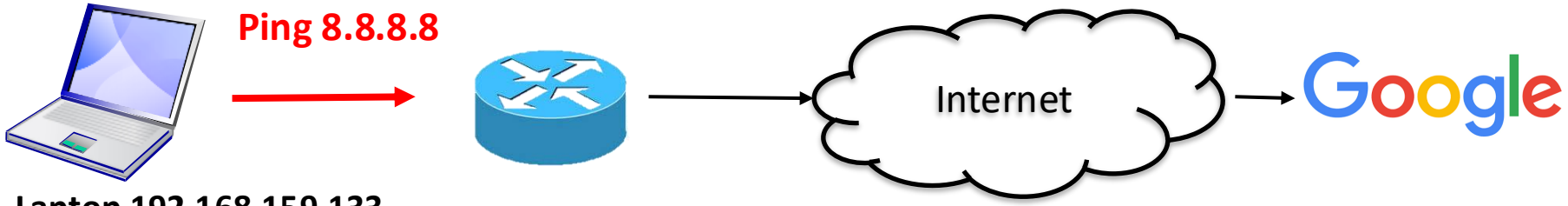
Laptop 192.168.159.133

- Layer 2:
  - Knows its own MAC address (burned into NIC)
  - **Knows Router's MAC address**

- Layer 3:
  - Got Router's IP address from DHCP
  - Ping Google (ICMP) at 8.8.8.8

Dst = destination
Src = source

Client has IP address thanks to DHCP, now wants to reach the Internet



Laptop 192.168.159.133

- Layer 2:
  - Knows its own MAC address (burned into NIC)
  - Knows Router's MAC address
- Layer 3:
  - Got Router's IP address from DHCP
  - Ping Google (ICMP) at 8.8.8.8

Now Laptop can address an ICMP ping to the Router with the Google destination

Remember a Router joins networks

- **In this case, it joins the Laptop's Local Area Network to the Internet**
- **Google sits on another Local Area Network that is connected to the Internet**

Dst = destination
Src = source

Client has IP address thanks to DHCP, now wants to reach the Internet



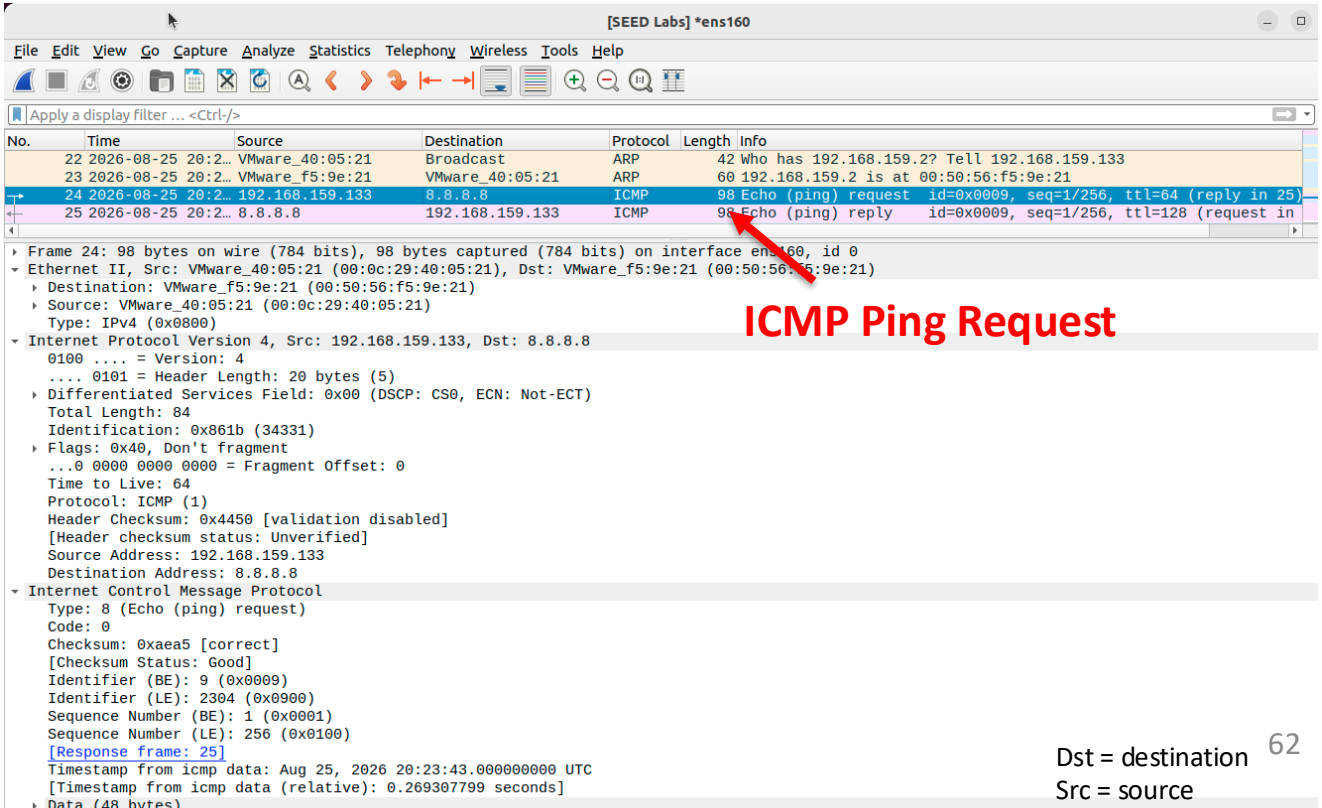
Ping 8.8.8.8



Laptop 192.168.159.133

- Layer 2:
  - Knows its own MAC address (burned into NIC)
  - Knows Router's MAC address
- Layer 3:
  - Got Router's IP address from DHCP
  - Ping Google (ICMP) at 8.8.8.8

Now Laptop can address an ICMP ping to the Router with the Google destination



ICMP Ping Request

Dst = destination
Src = source

Client has IP address thanks to DHCP, now wants to reach the Internet



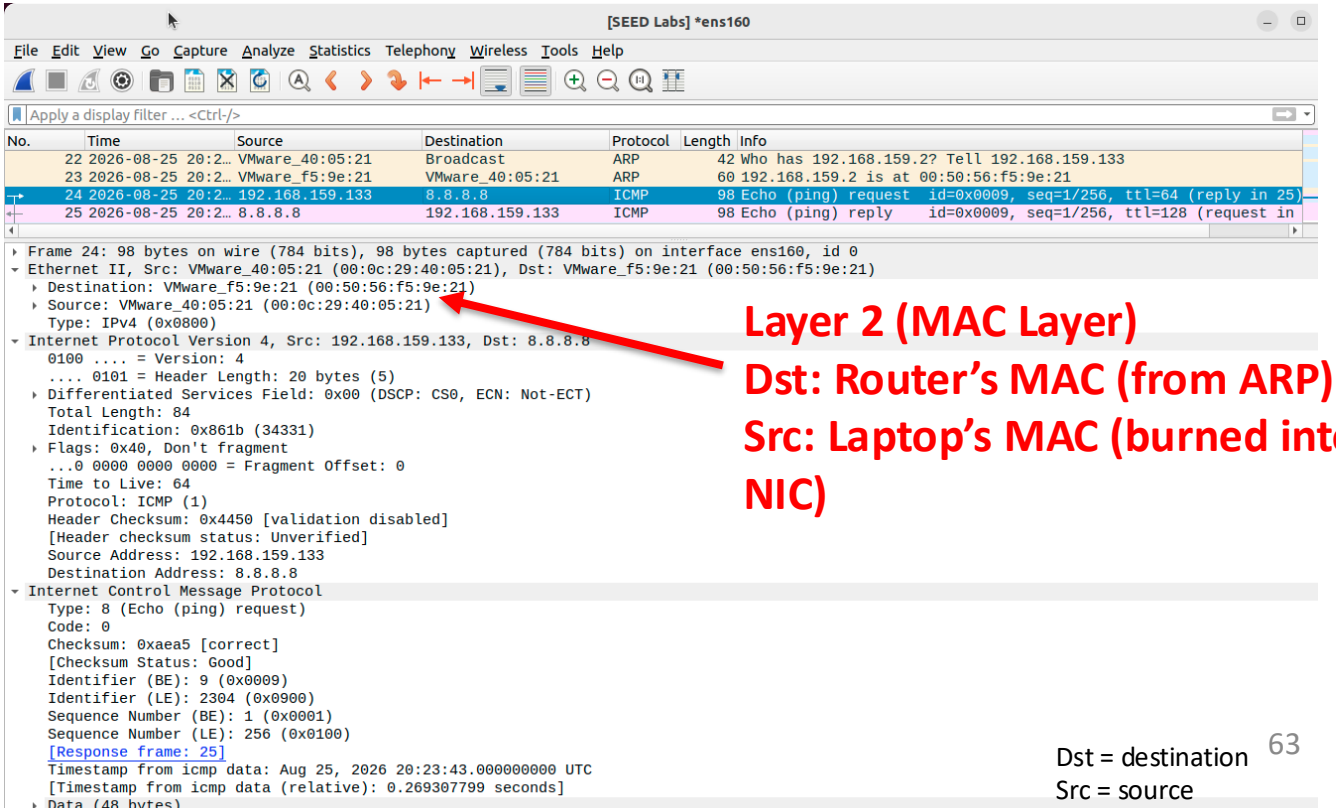
Ping 8.8.8.8



Laptop 192.168.159.133

- Layer 2:
  - Knows its own MAC address (burned into NIC)
  - Knows Router's MAC address
- Layer 3:
  - Got Router's IP address from DHCP
  - Ping Google (ICMP) at 8.8.8.8

Now Laptop can address an ICMP ping to the Router with the Google destination



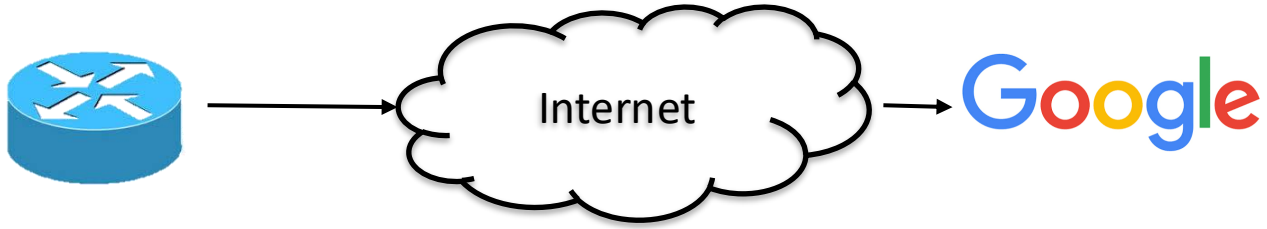
Layer 2 (MAC Layer)
Dst: Router's MAC (from ARP)
Src: Laptop's MAC (burned into NIC)

Dst = destination
Src = source

Client has IP address thanks to DHCP, now wants to reach the Internet



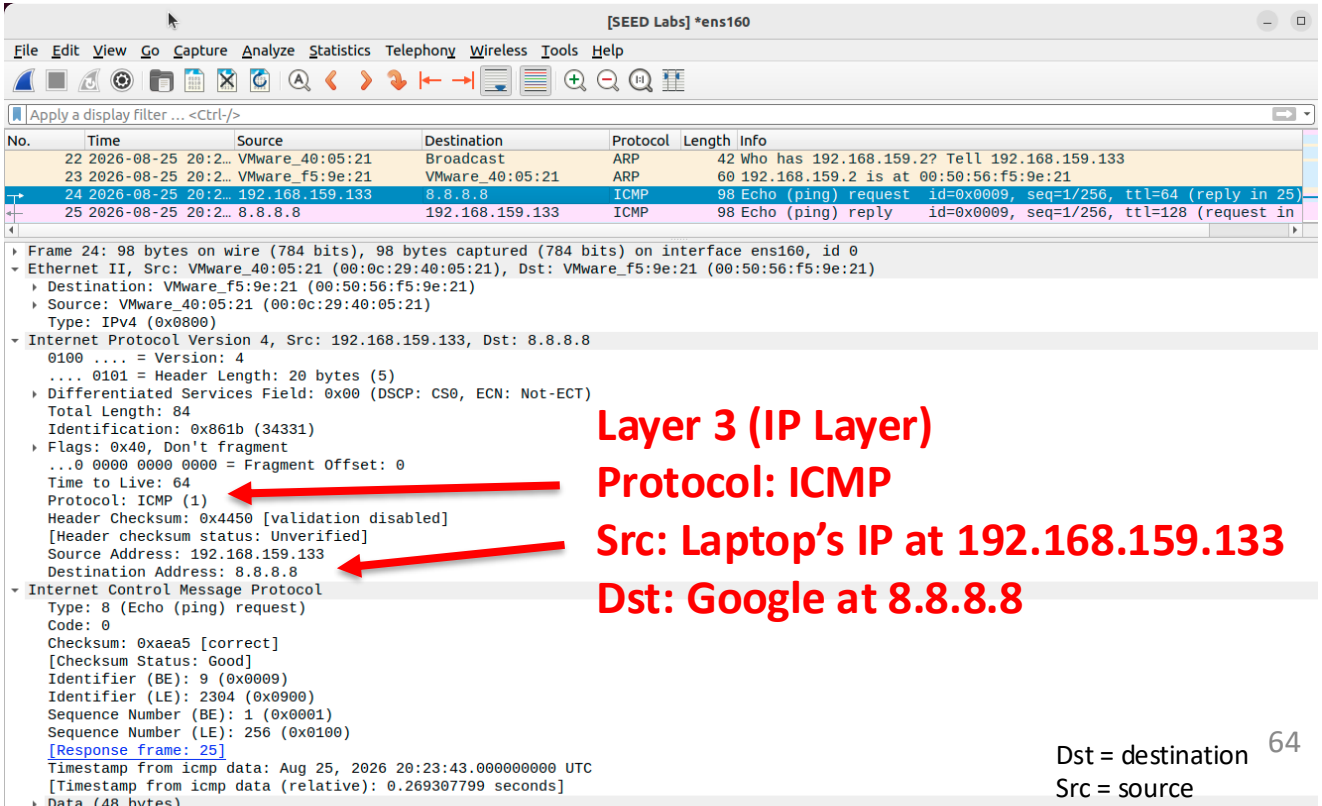
Ping 8.8.8.8



Laptop 192.168.159.133

- Layer 2:
  - Knows its own MAC address (burned into NIC)
  - Knows Router's MAC address
- Layer 3:
  - Got Router's IP address from DHCP
  - Ping Google (ICMP) at 8.8.8.8

Now Laptop can address an ICMP ping to the Router with the Google destination



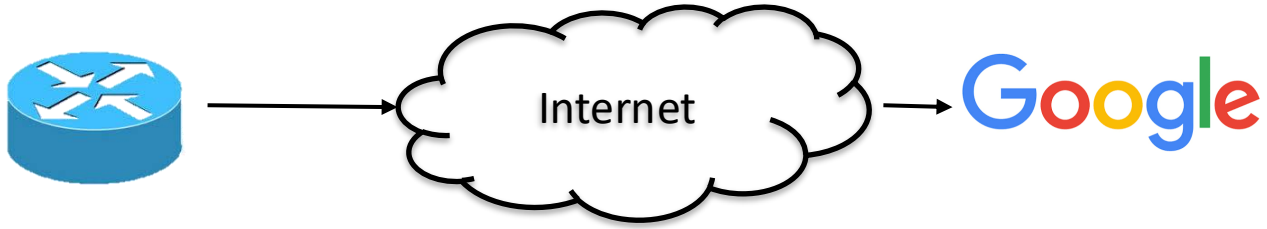
Layer 3 (IP Layer)
Protocol: ICMP
Src: Laptop's IP at 192.168.159.133
Dst: Google at 8.8.8.8

Dst = destination
Src = source

Client has IP address thanks to DHCP, now wants to reach the Internet



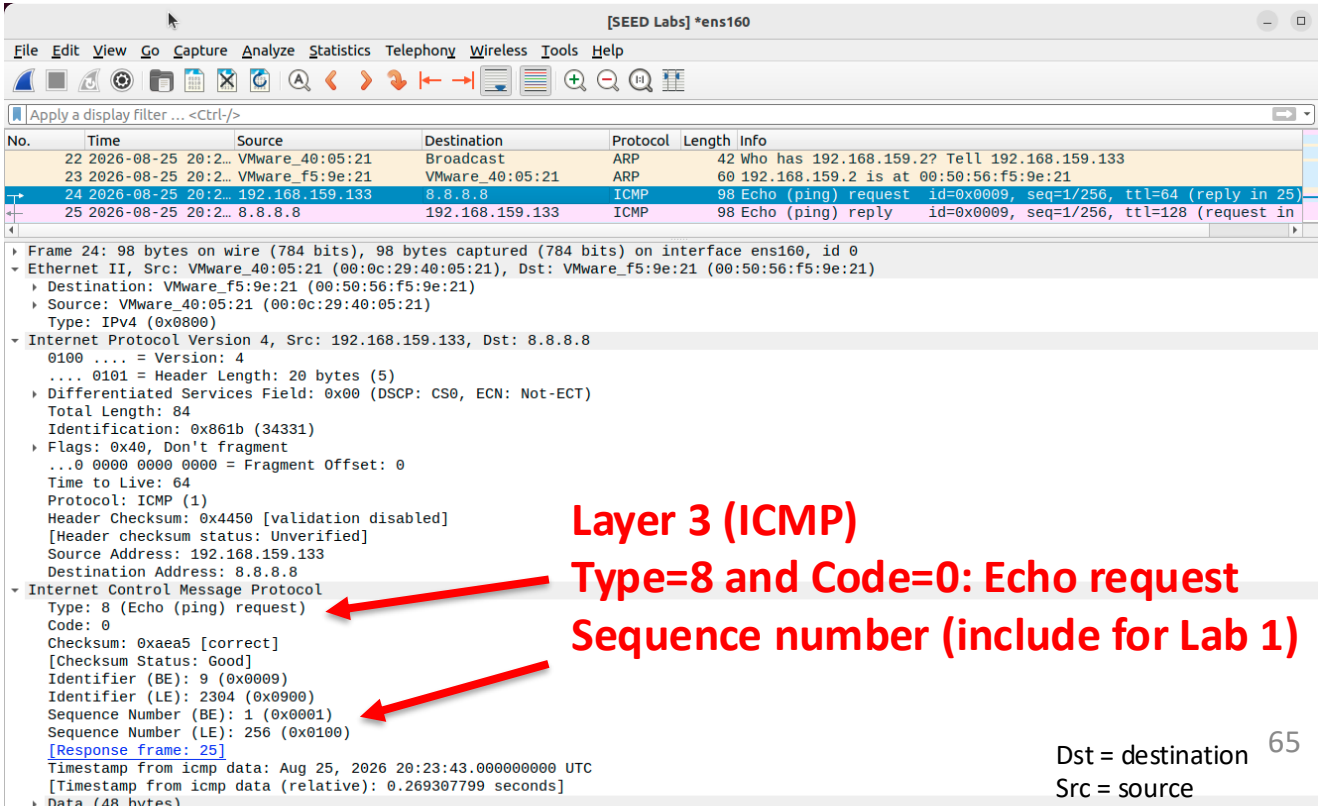
Ping 8.8.8.8



Laptop 192.168.159.133

- Layer 2:
  - Knows its own MAC address (burned into NIC)
  - Knows Router's MAC address
- Layer 3:
  - Got Router's IP address from DHCP
  - Ping Google (ICMP) at 8.8.8.8

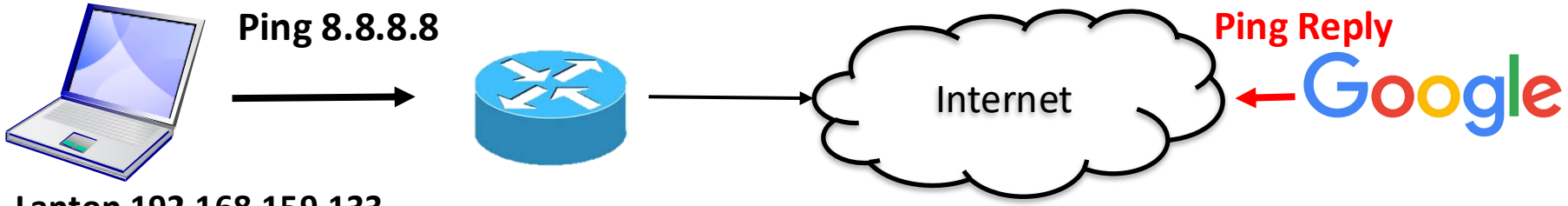
Now Laptop can address an ICMP ping to the Router with the Google destination



Layer 3 (ICMP)
Type=8 and Code=0: Echo request
Sequence number (include for Lab 1)

Dst = destination
Src = source

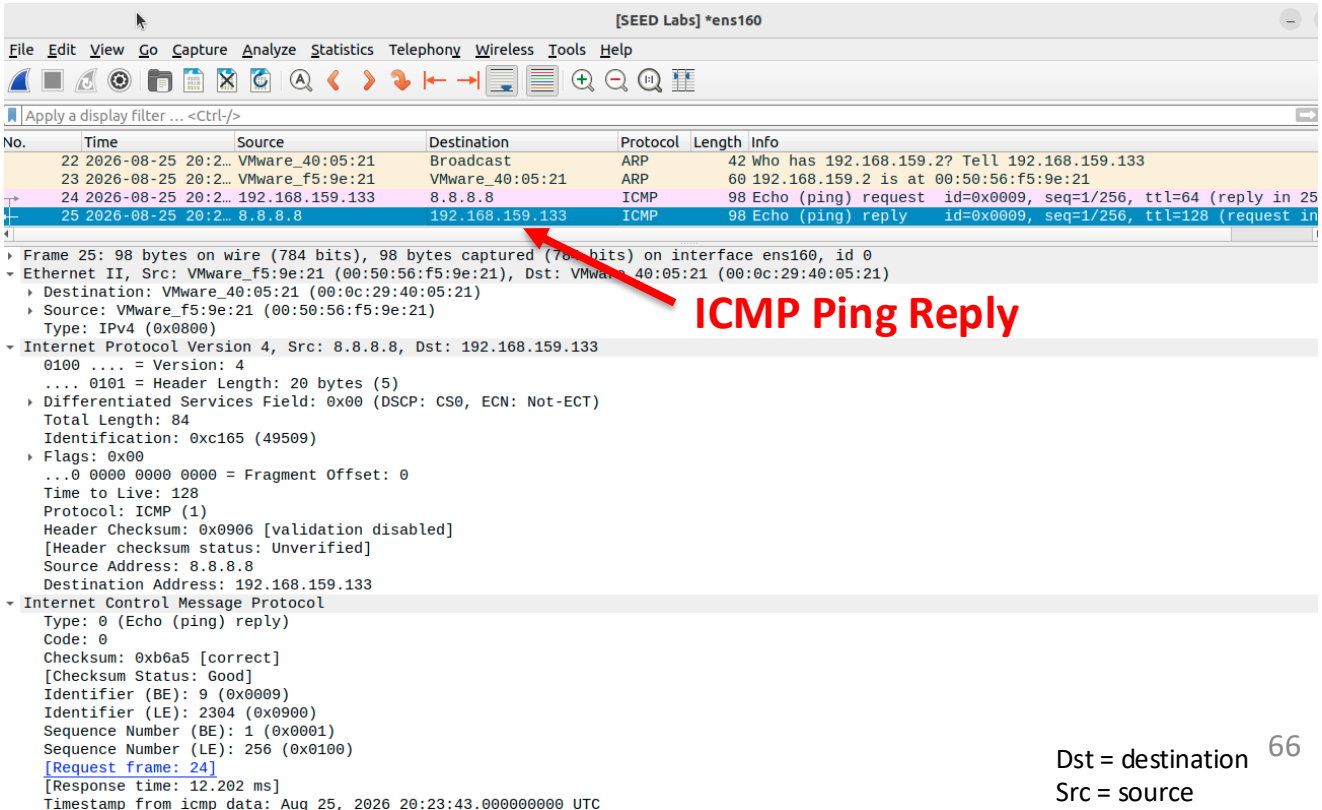
Client has IP address thanks to DHCP, now wants to reach the Internet



Laptop 192.168.159.133

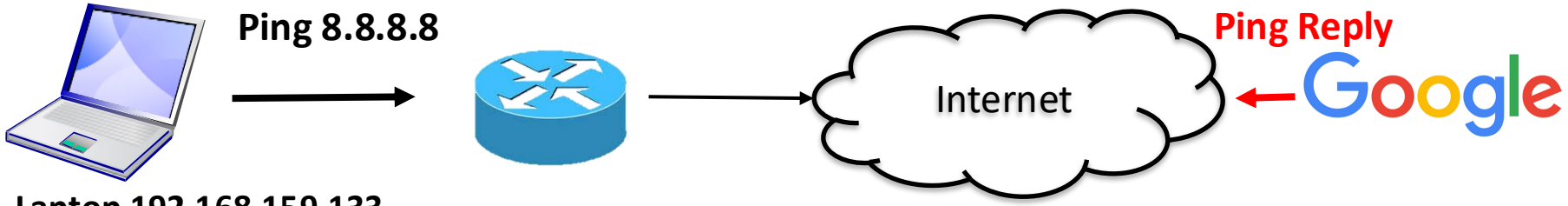
- Layer 2:
  - Knows its own MAC address (burned into NIC)
  - Knows Router's MAC address
- Layer 3:
  - Got Router's IP address from DHCP
  - Ping Google (ICMP) at 8.8.8.8

Google replies to the Echo Request with an Echo Reply



Dst = destination
Src = source

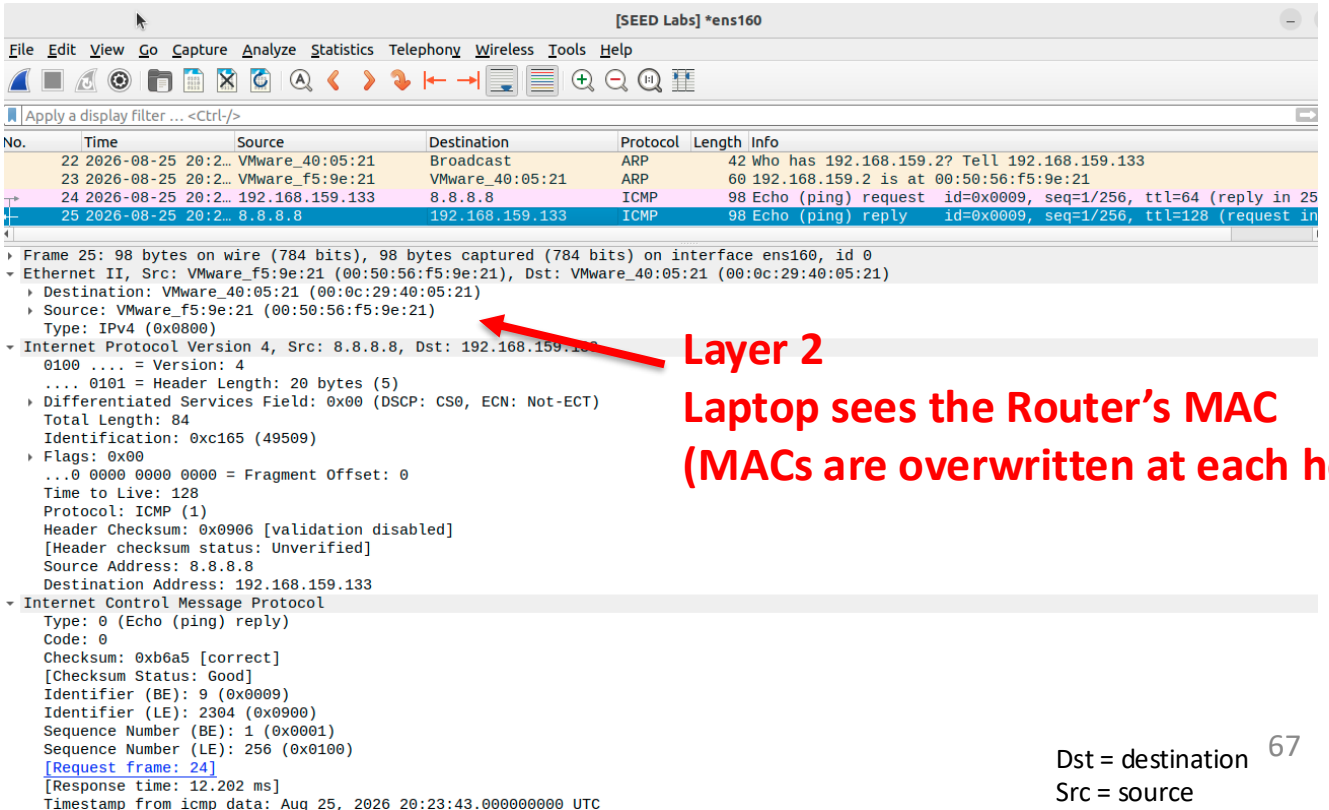
Client has IP address thanks to DHCP, now wants to reach the Internet



Laptop 192.168.159.133

- Layer 2:
  - Knows its own MAC address (burned into NIC)
  - Knows Router's MAC address
- Layer 3:
  - Got Router's IP address from DHCP
  - Ping Google (ICMP) at 8.8.8.8

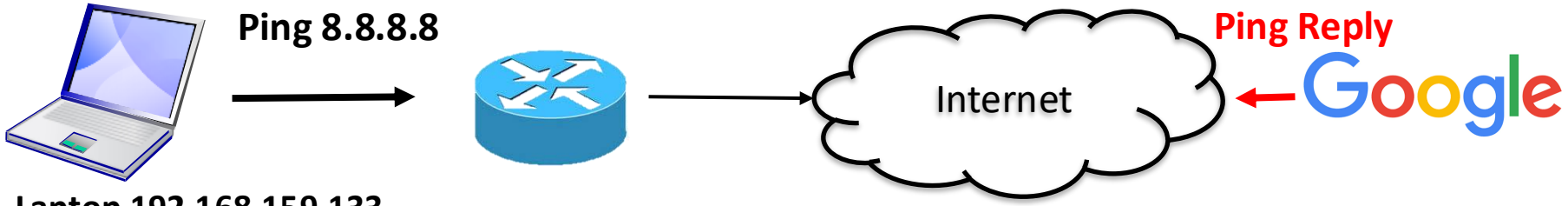
Google replies to the Echo Request with an Echo Reply



Layer 2
Laptop sees the Router's MAC
(MACs are overwritten at each hop)

Dst = destination
Src = source

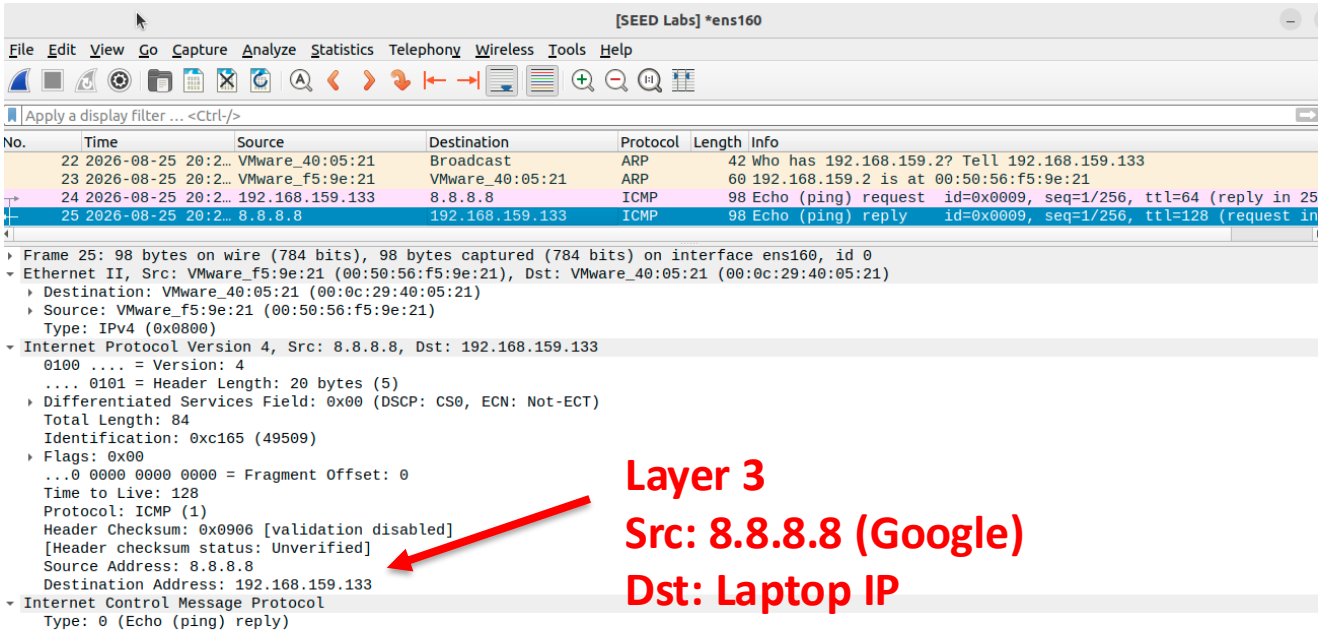
Client has IP address thanks to DHCP, now wants to reach the Internet



Laptop 192.168.159.133

- Layer 2:
  - Knows its own MAC address (burned into NIC)
  - Knows Router's MAC address
- Layer 3:
  - Got Router's IP address from DHCP
  - Ping Google (ICMP) at 8.8.8.8

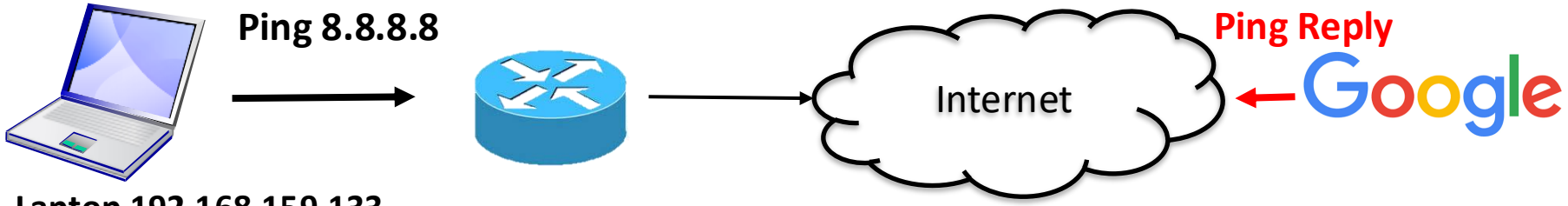
Google replies to the Echo Request with an Echo Reply



Layer 3
Src: 8.8.8.8 (Google)
Dst: Laptop IP

Dst = destination
Src = source

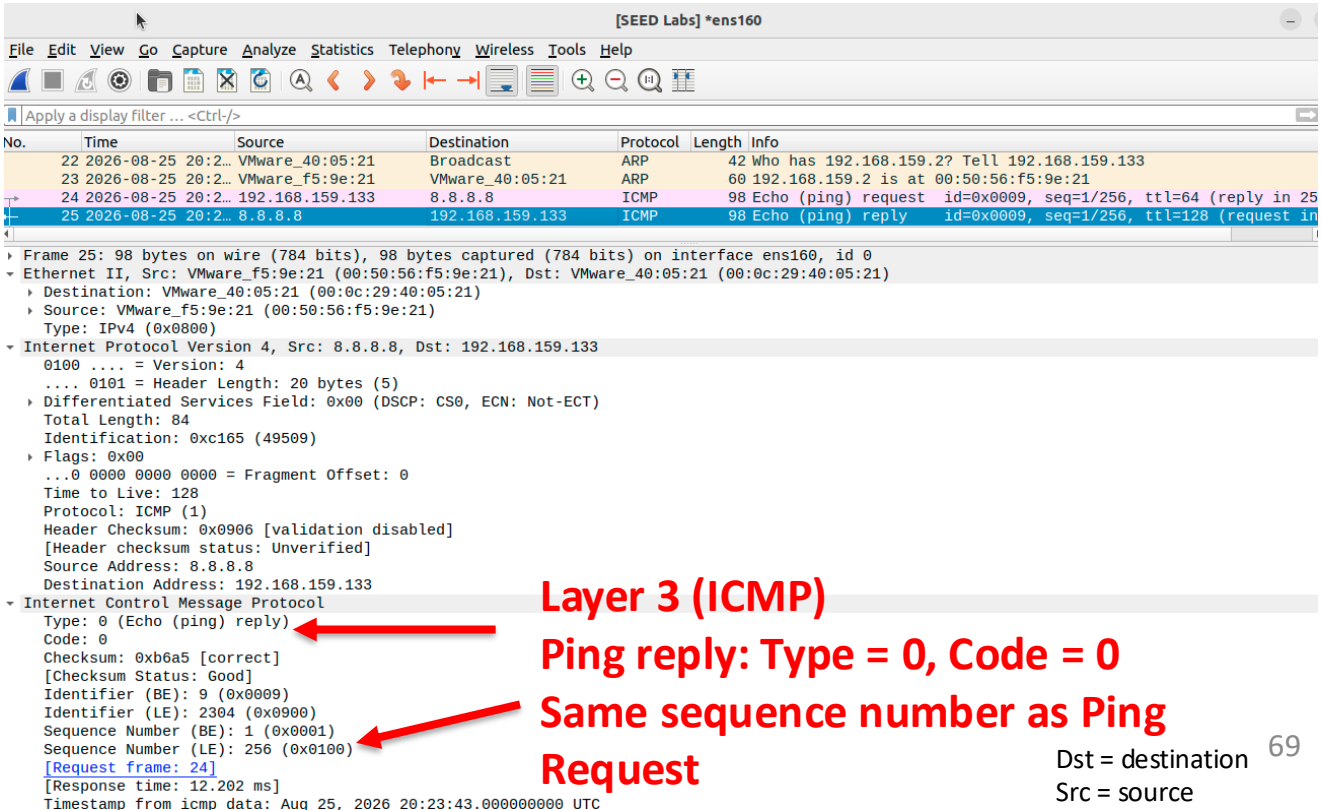
Client has IP address thanks to DHCP, now wants to reach the Internet



Laptop 192.168.159.133

- Layer 2:
  - Knows its own MAC address (burned into NIC)
  - Knows Router's MAC address
- Layer 3:
  - Got Router's IP address from DHCP
  - Ping Google (ICMP) at 8.8.8.8

Google replies to the Echo Request with an Echo Reply



Layer 3 (ICMP)
Ping reply: Type = 0, Code = 0
Same sequence number as Ping Request

Dst = destination
Src = source

Agenda

1. Network layer overview
2. DHCP
3. Ping, traceroute, and ARP
4. Exercises

Exercises

How does your computer get a MAC address? What is your MAC address?
How does your computer get an IP address? What is your IP address?
Start Wireshark capturing in your VM (blue fin at top)

- Ping 8.8.8.8 (let it run for a few pings)
- Stop Wireshark (red block at top)
- Set Wireshark filter ip.addr == 8.8.8.8 (Google)
- See ICMP requests from your computer and ICMP reply from Google
- Look at Layer 2 and Layer 3 for each line

Start Wireshark capturing again

- Start browser and go to http://vibrantcloud.org/ (might have to refresh browser)
- Stop Wireshark capture
- Set Wireshark filter to http
- Find the request your browser made for the web page
- See the web server's response
- Look at the Layer 2 to 7
- Find the web page's HTML text

Turn off Wi-Fi, Start Wireshark, turn on Wi-Fi, filter on DHCP to see request