Intro to Networking and ARP

CS 161 Fall 2022 - Lecture 16

Last Time: SQL Injection

- Web servers interact with databases to store data
  - Web servers use SQL to interact with databases
- SQL injection: Untrusted input is used as parsed SQL
  - The attacker can construct their own queries to run on the SQL server!
  - Blind SQL injection: SQLi with little to no feedback from the SQL query
  - Defense: Input sanitization
    - Difficult to implement correctly
  - Defense: Prepared statements
    - Data only ever treated as data; bulletproof!
- Command injection: Untrusted input is used as any parsed language
  - Defense: Keep it simple and use safe API calls

Last Time: CAPTCHAs

- CAPTCHA: A challenge that is easy for a human to solve, but hard for a computer to solve
  - Examples: Reading distorted text, identifying images
  - Original purpose: Distinguishing between humans and bots
  - Modern purpose: Forces the attacker to spend some money to solve the CAPTCHAs
  - Modern purpose: Providing training data for machine learning algorithms
- Issues with CAPTCHAs
  - As computer algorithms get smarter, CAPTCHAs get harder, and not all humans are able to solve them easily
  - Ambiguity: CAPTCHAs might be so hard that the validator doesn't know the solution either!
  - Not all bots are bad

Today: Intro to Networking

- Internet: A global network of computers
- OSI model: A layered model of protocols

What’s the Internet?

What’s the Internet?

- **Network**: A set of connected machines that can communicate with each other
  - Machines on the network agree on a **protocol**, a set of rules for communication
- **Internet**: A global network of computers
  - The web sends data between browsers and servers using the Internet
  - The Internet can be used for more than the web (e.g. SSH)

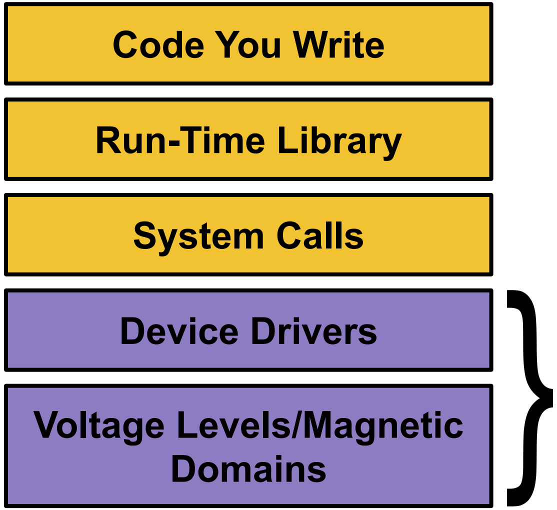
Protocols

- A **protocol** is an agreement on how to communicate that specifies syntax and semantics
  - *Syntax:* How a communication is specified and structured (format, order of messages)
  - *Semantics*: What a communication means (actions taken when sending/receiving messages)
- Example: Protocol for asking a question in lecture?
  1. The student should raise their hand
  2. The student should wait to be called on by the speaker or wait for the speaker to pause
  3. The student should speak the question after being called on or after waiting
  4. If the student has been unrecognized after some time: Vocalize with “Excuse me!”

Layering: The OSI Model

Layering

- Internet design is partitioned into various layers. Each layer…
  - Has a protocol
  - Relies on services provided by the layer below it
  - Provides services to the layer above it
- Analogous to the structure of an application and the "services" that each layer relies on and provides



Fully isolated from user programs

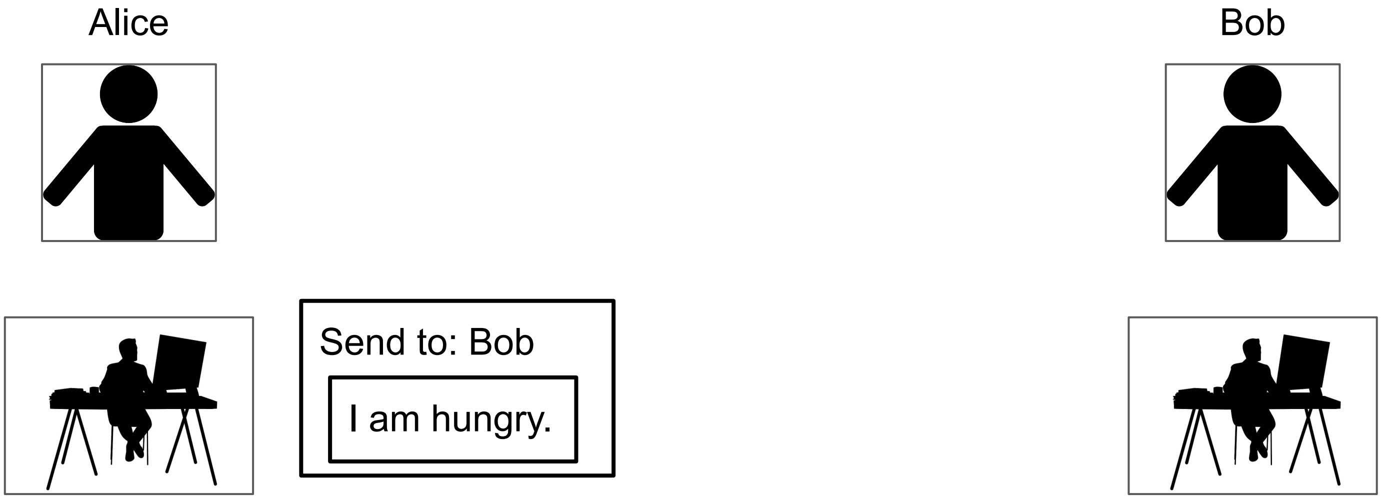
Example: Sending Mail

Alice

I am hungry.

Bob

Example: Sending Mail



Alice

Bob

Send to: Bob

I am hungry.

Example: Sending Mail

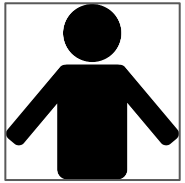
Alice

Mail to: 123 Bob St

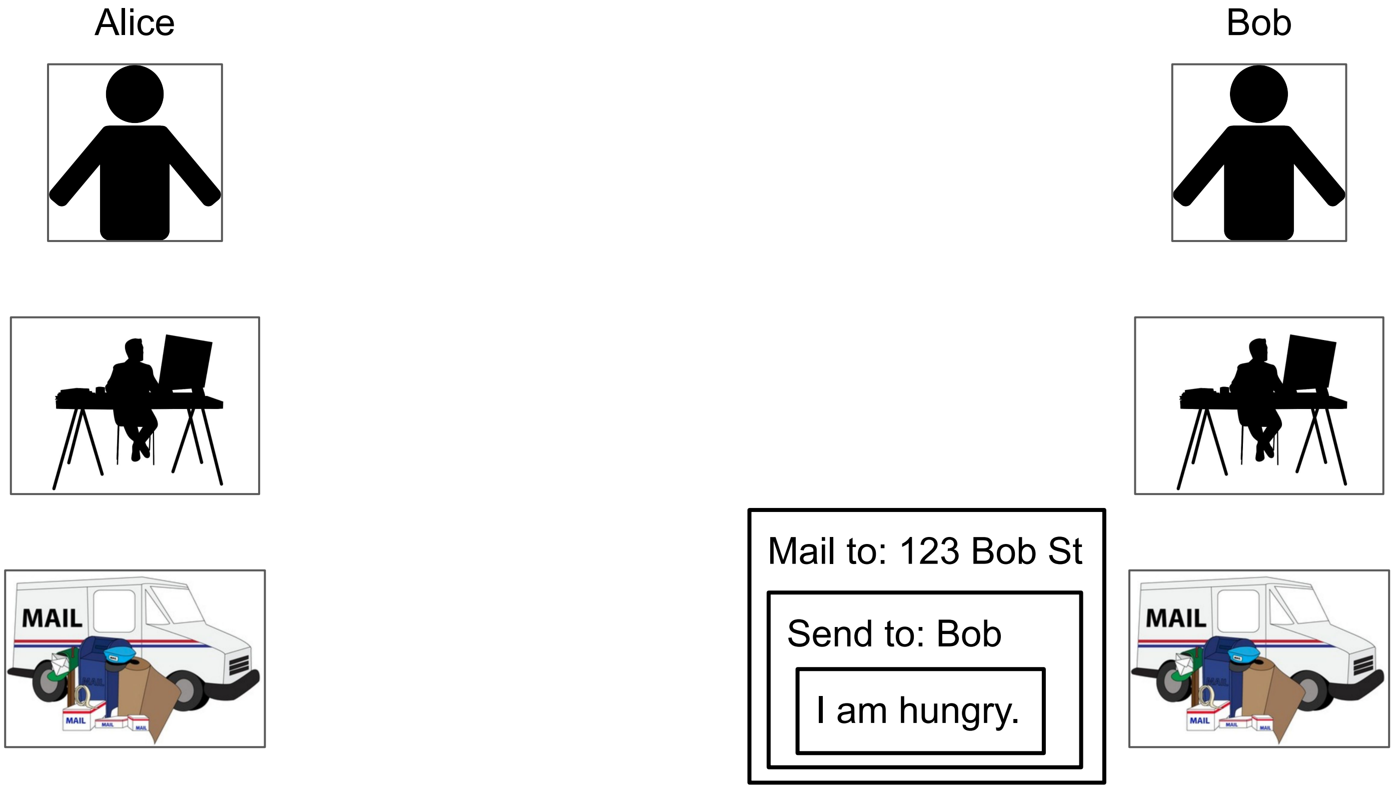
Send to: Bob

I am hungry.

Bob



Example: Sending Mail



Alice

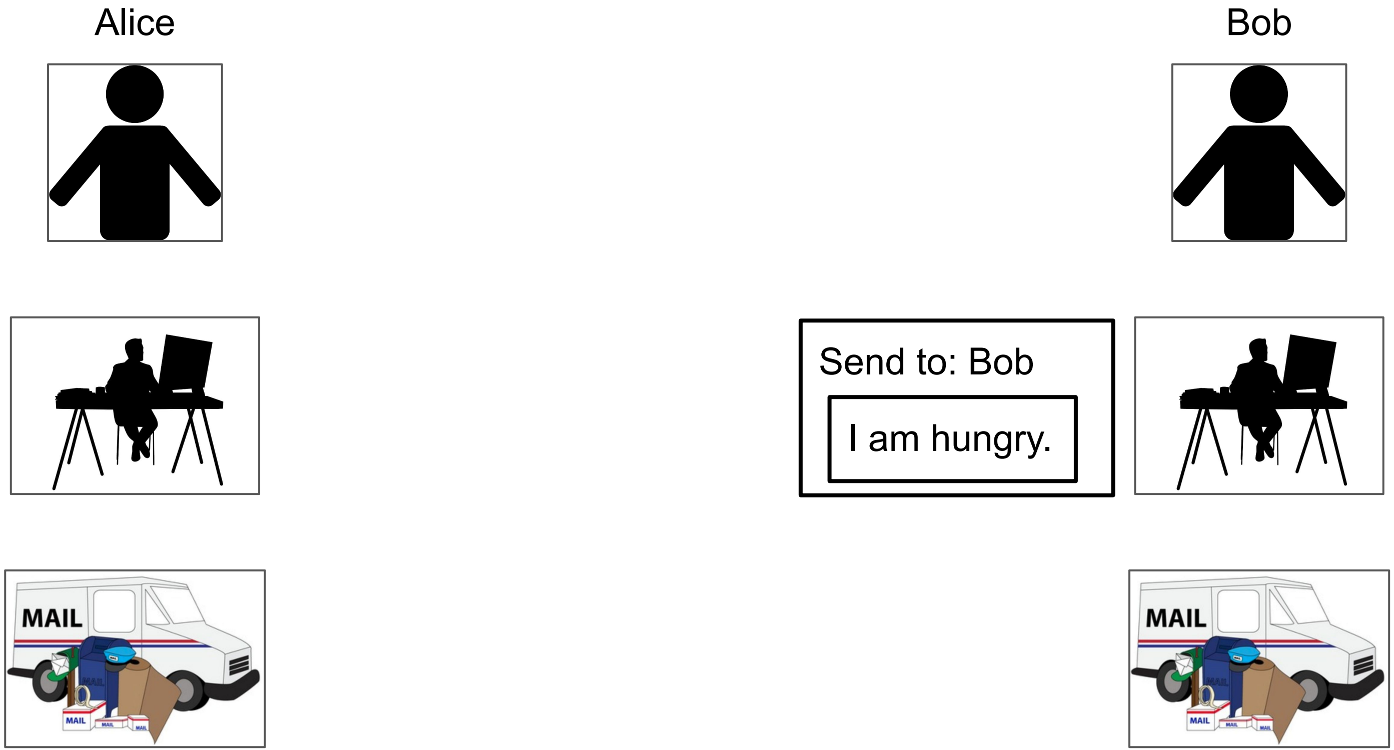
Bob

Mail to: 123 Bob St

Send to: Bob

I am hungry.

Example: Sending Mail



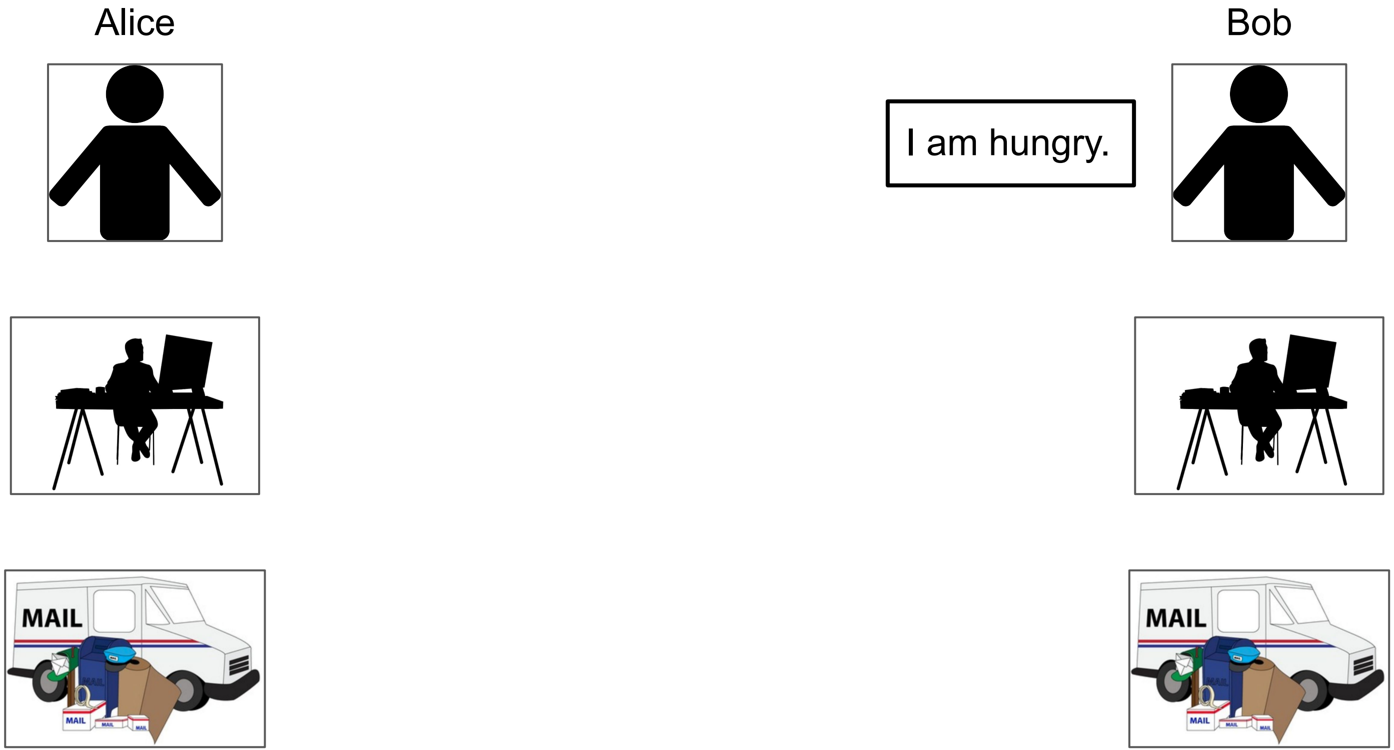
Alice

Bob

Send to: Bob

I am hungry.

Example: Sending Mail

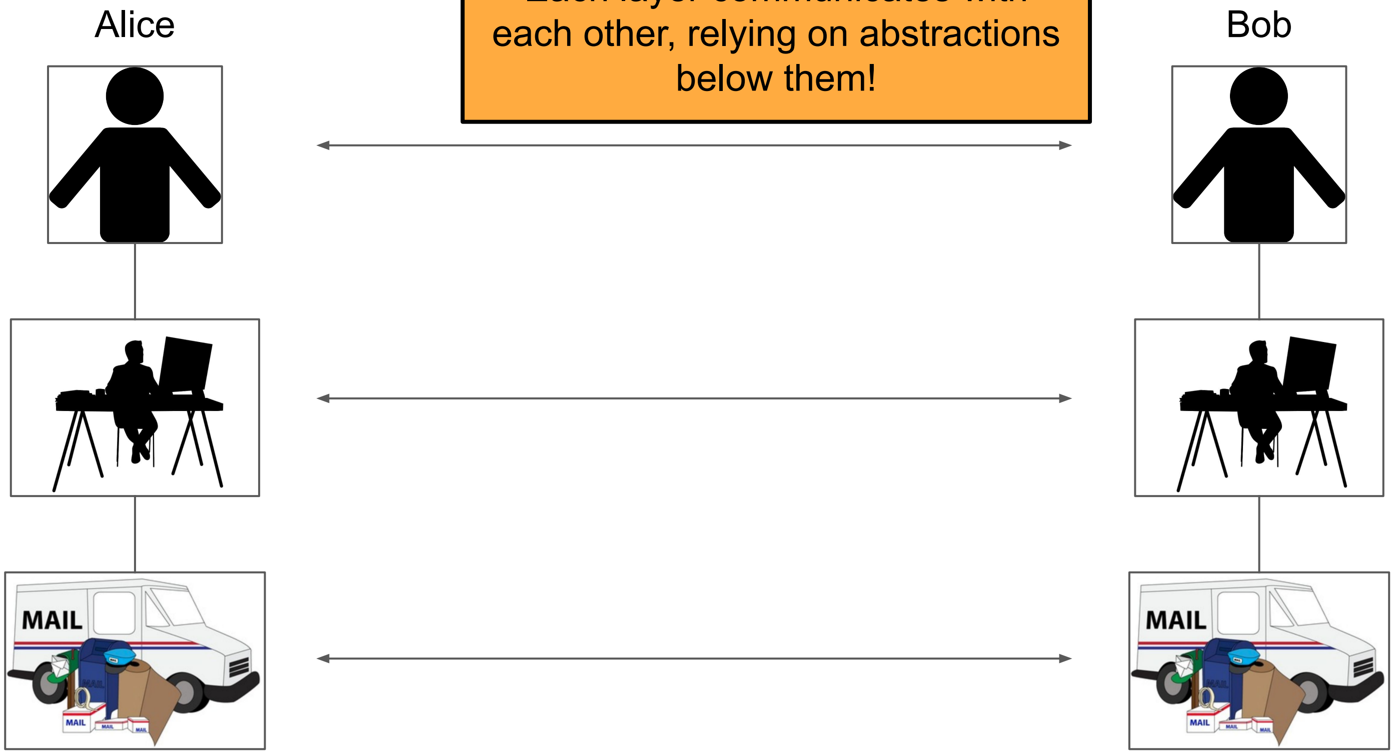


Alice

Bob

I am hungry.

Example: Sending Mail



Each layer communicates with each other, relying on abstractions below them!

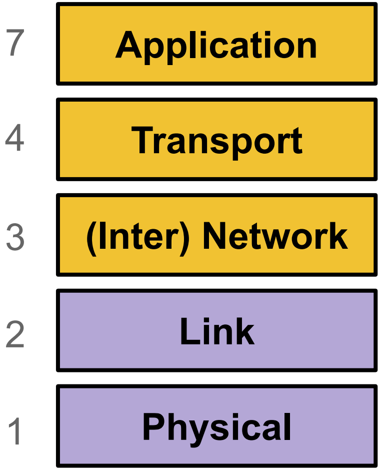
Relies upon: Sending messages to people

Provides: Sending messages to people
Relies upon: Sending messages to addresses

Provides: Sending messages to addresses

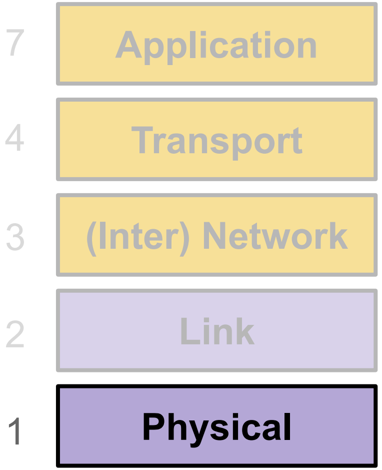
OSI Model

- **OSI model:** Open Systems Interconnection model, a layered model of Internet communication
  - Originally divided into 7 layers
    - But layers 5 and 6 aren’t used in the real world, so we ignore them
    - And we’ll talk about layer 4.5 for encryption later
- Same reliance upon abstraction
  - A layer can be implemented in different ways without affecting other layers
  - A layer’s protocol can be substituted with another protocol without affecting other layers



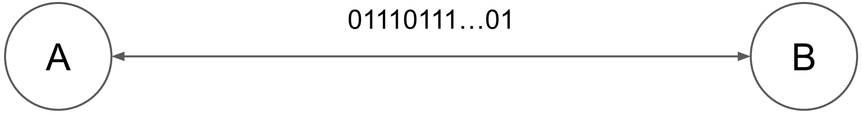
Layer 1: Physical Layer

- **Provides**: Sending bits from one device to another
  - Encodes bits to send them over a physical link
    - Patterns of voltage levels
    - Photon intensities
    - RF modulation
- Examples
  - Wi-Fi radios (IEEE 802.11)
  - Ethernet voltages (IEEE 802.3)

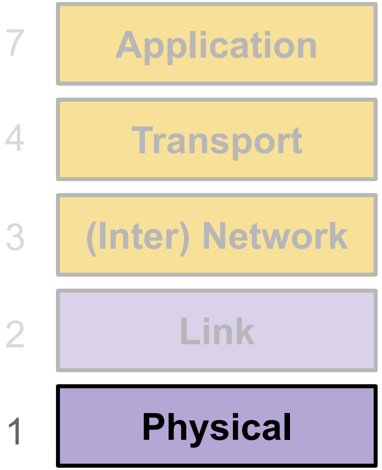


Layer 1: Physical Layer

Physical layer: “How do I transmit this sequence of 0’s and 1’s from A to B?”

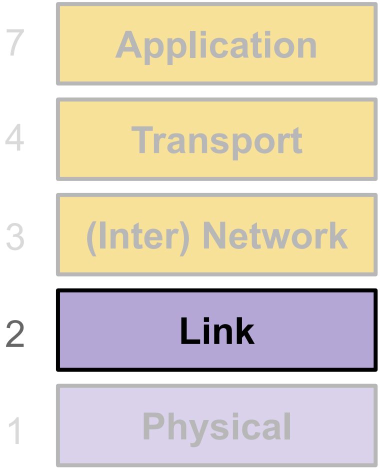


Next: How do we talk to more than one device?



Layer 2: Link Layer

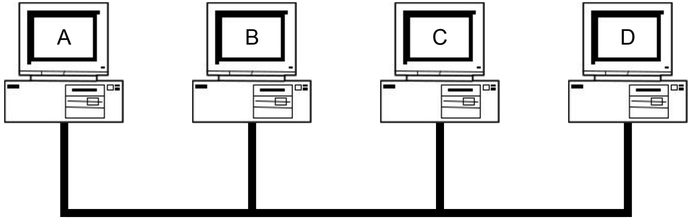
- **Provides**: Sending frames directly from one device to another
  - **Relies upon**: Sending bits from one device to another
  - Encodes messages into groups of bits called "frames"
- Examples
  - Ethernet frames (IEEE 802.3)



Layer 2: Link Layer

- **Local area network** (**LAN**): A set of computers on a shared network that can directly address one another
  - Consists of multiple physical links
- Frames must consist of at least 3 things:
  - Source ("Who is this message coming from?")
  - Destination ("Who is this message going to?")
  - Data ("What does this message say?")

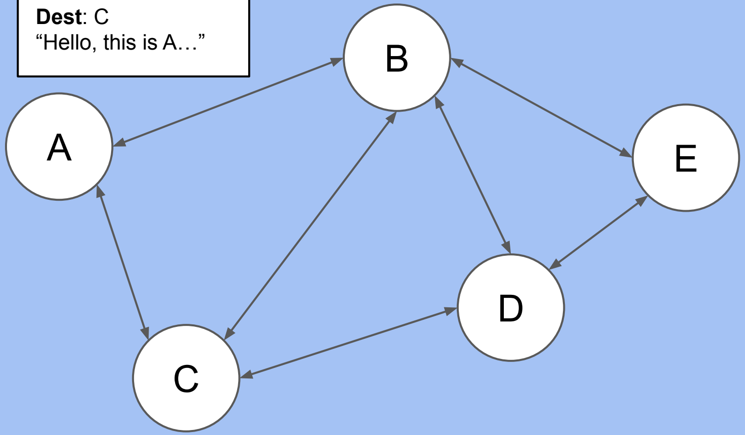
Source: A
Destination: C
"Hello, this is A…"



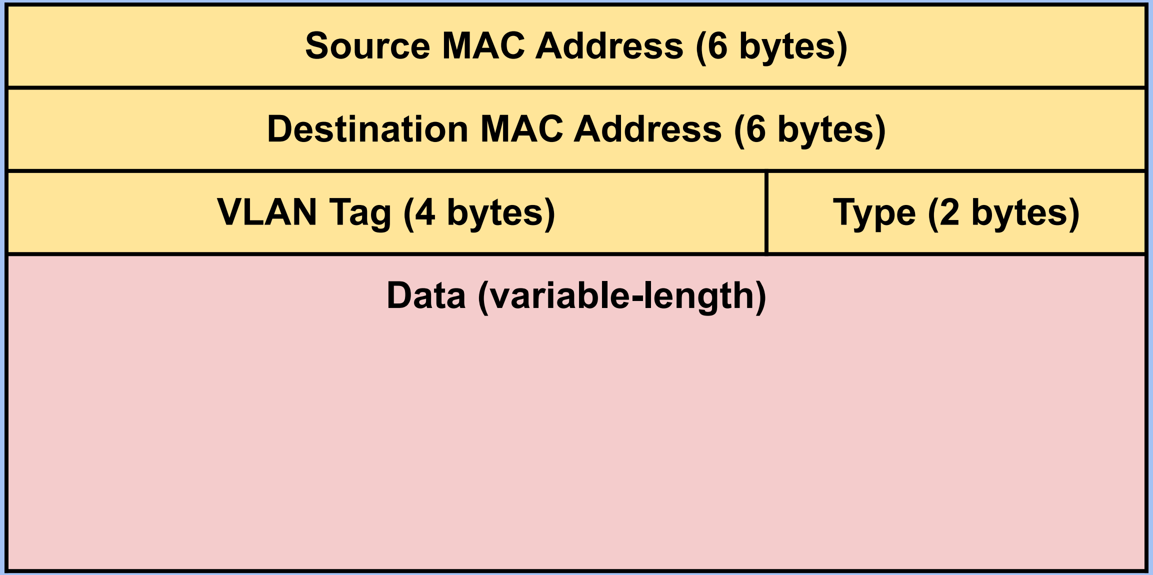
Layer 2: Link Layer

- In reality, computers aren’t all connected to the same wire
  - Instead, local networks are a set of point-to-point links
- However, Layer 2 still allows direct addressing between any two devices
  - Enabled by transmitting a frame across multiple physical links until it reaches its destination
  - Provides an **abstraction** of a “everything is connected to one wire”

Source: A
Dest: C
“Hello, this is A…”



Ethernet and MAC Addresses



Source MAC Address (6 bytes)

Destination MAC Address (6 bytes)

VLAN Tag (4 bytes)

Type (2 bytes)

Data (variable-length)

Ethernet header

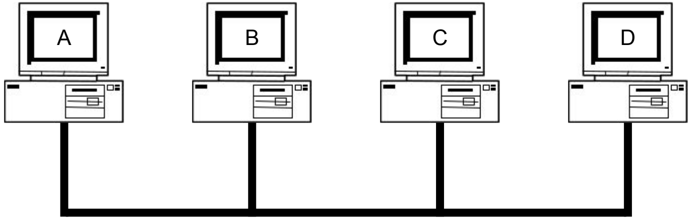
Ethernet and MAC Addresses

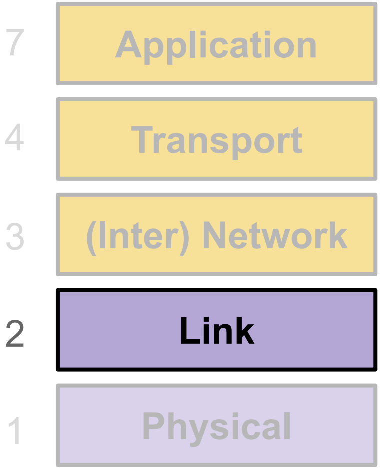
- **Ethernet**: A common layer 2 protocol that most endpoint devices use
- **MAC address**: A 6-byte address that identifies a piece of network equipment (e.g. your phone's Wi-Fi controller)
  - Stands for **Media Access Control**, not message authentication code
  - Typically represented as 6 hex bytes: **13:37:ca:fe:f0:0d**
  - The first 3 bytes are assigned to manufacturers (i.e. who made the equipment)
    - This is useful in identifying a device
  - The last 3 bytes are device-specific

Layer 2: Link Layer

Link layer: “How do I transmit this frame from A to C, making sure that no one else thinks the message is for them?”

Source: A
Dest: C
“Hello, this is A…”

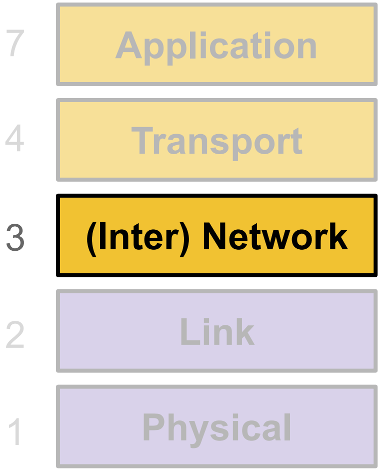




Next: How do we address every device in existence?

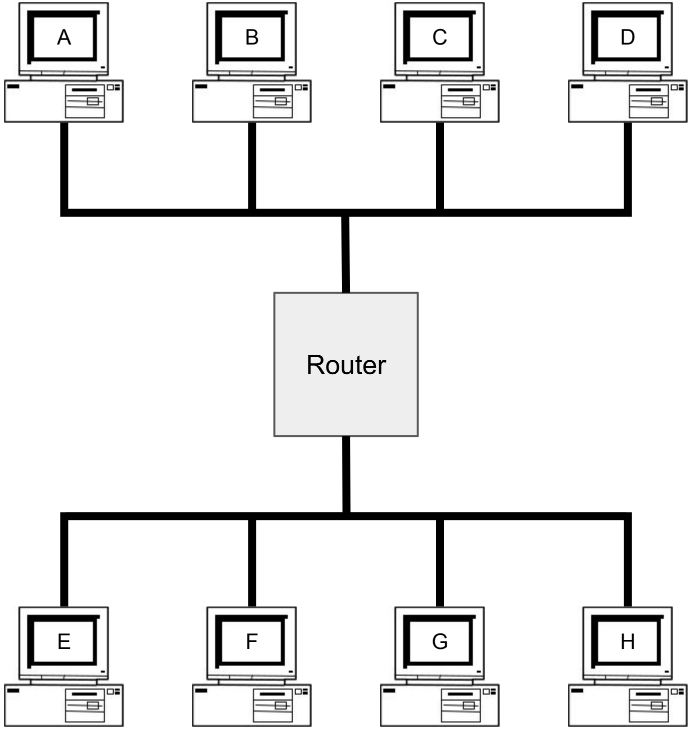
Layer 3: Network Layer

- **Provides**: Sending packets from any device to any other device
  - **Relies upon**: Sending frames directly from one device to another
  - Encodes messages into groups of bits called “packets”
  - Bridges multiple LANs to provide global addressing
- Examples
  - Internet Protocol (IP)

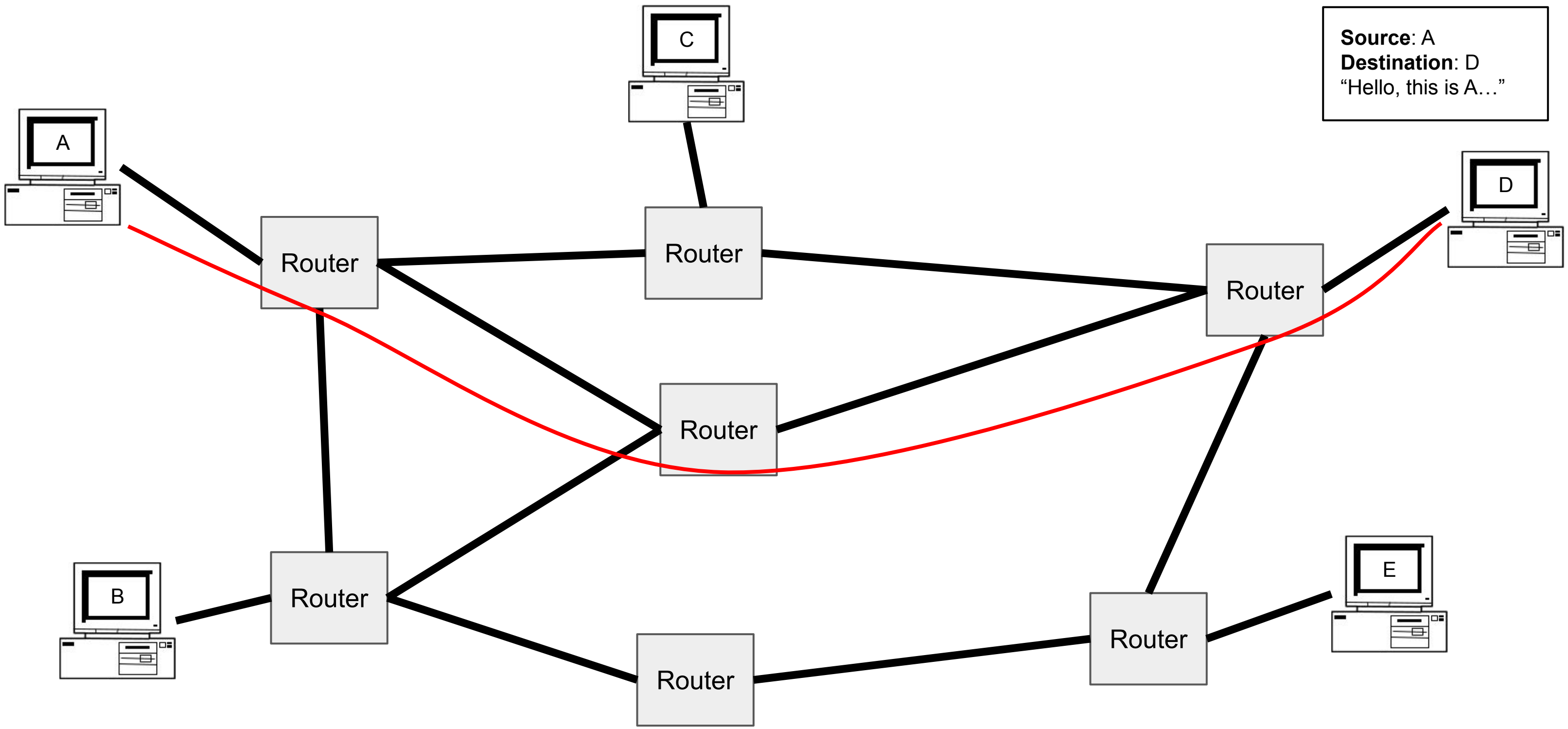


Layer 3: Network Layer

- Recall the ideal layer 2 model: All devices can directly address all other devices
  - This would not scale to the size of the Internet!
- Instead, allow packets to be **routed** across different devices to reach the destination
  - Each hop is allowed to use its own physical and link layers!
- Basic model:
  - Is the destination of the packet directly connected to my LAN?
    - Pass it off to Layer 2
  - Otherwise, **route** the packet closer to the destination

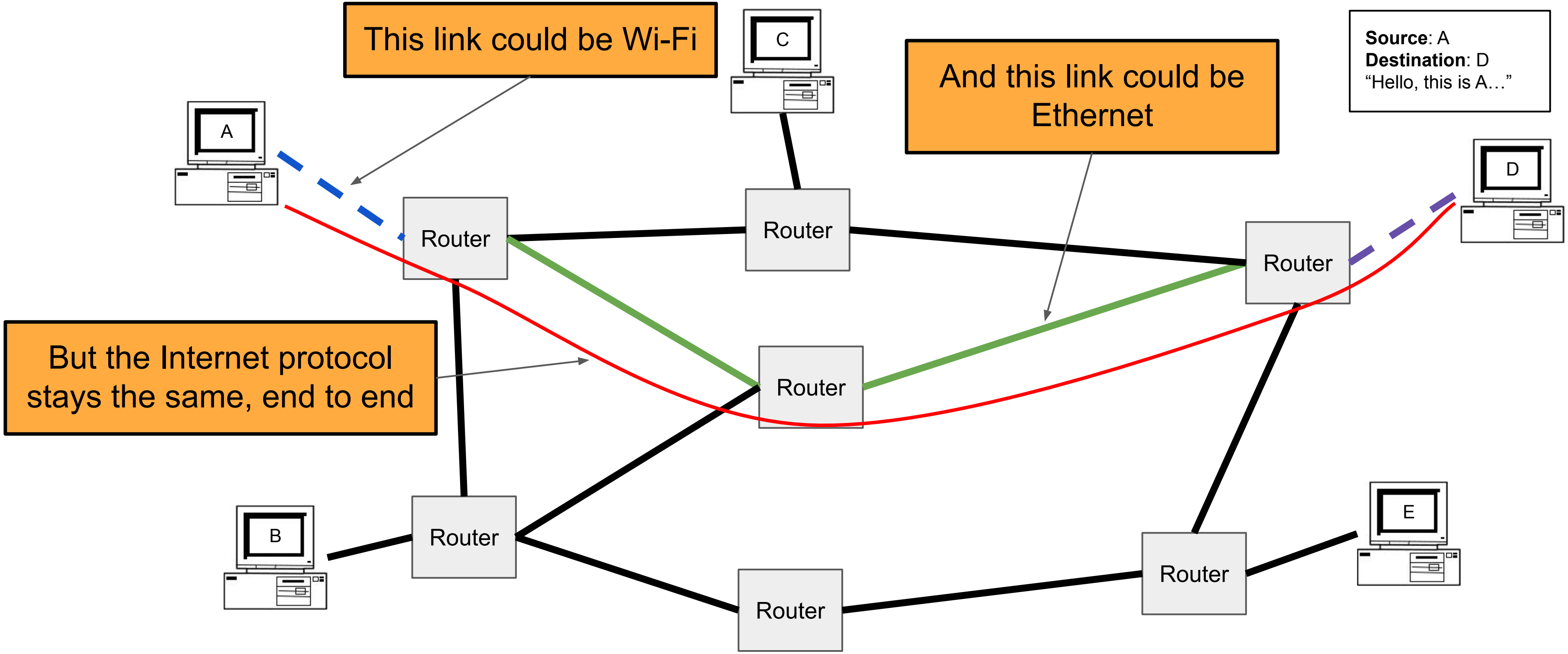


Layer 3: Network Layer



Source: A
Destination: D
“Hello, this is A…”

Layer 3: Network Layer



This link could be Wi-Fi

And this link could be Ethernet

But the Internet protocol stays the same, end to end

Source: A
Destination: D
“Hello, this is A…”

Layer 3: Network Layer

- Packets must consist of at least 3 things:
  - Source (“Who is this message coming from?”)
  - Destination (“Who is this message going to?”)
  - Data (“What does this message say?”)
  - Similar to frames (layer 2)
- Packets may be fragmented into smaller packets
  - Different links might support different maximum packet sizes
  - Up to the recipient to reassemble fragments into the original packet
  - In IPv4, any node may fragment a packet if it is too large to route
  - In IPv6, the sender must fragment the packet themselves
- Each router forwards a given packet to the next hop
  - We will cover how a router knows how to forward—and attacks on it—in the future
- Packets are not guaranteed to take a given route
  - Two packets with the same source and destination may take different routes

Internet Protocol (IP)

<table>
<tr><td colspan="4">Version (4 bits)</td><td colspan="4">Header Length (4 bits)</td><td colspan="6">Type of Service (6 bits)</td><td colspan="2">ECN (2 bits)</td><td colspan="16">Total Length (16 bits)</td></tr>
<tr><td colspan="16">Identification (16 bits)</td><td colspan="3">Flags (3 bits)</td><td colspan="13">Fragment Offset (13 bits)</td></tr>
<tr><td colspan="8">Time to Live (8 bits)</td><td colspan="8">Protocol (8 bits)</td><td colspan="16">Header Checksum (16 bits)</td></tr>
<tr><td colspan="32">Source Address (32 bits)</td></tr>
<tr><td colspan="32">Destination Address (32 bits)</td></tr>
<tr><td colspan="32">Options (variable length)</td></tr>
<tr><td colspan="32">Data (variable length)</td></tr>
</table>

IPv4 header

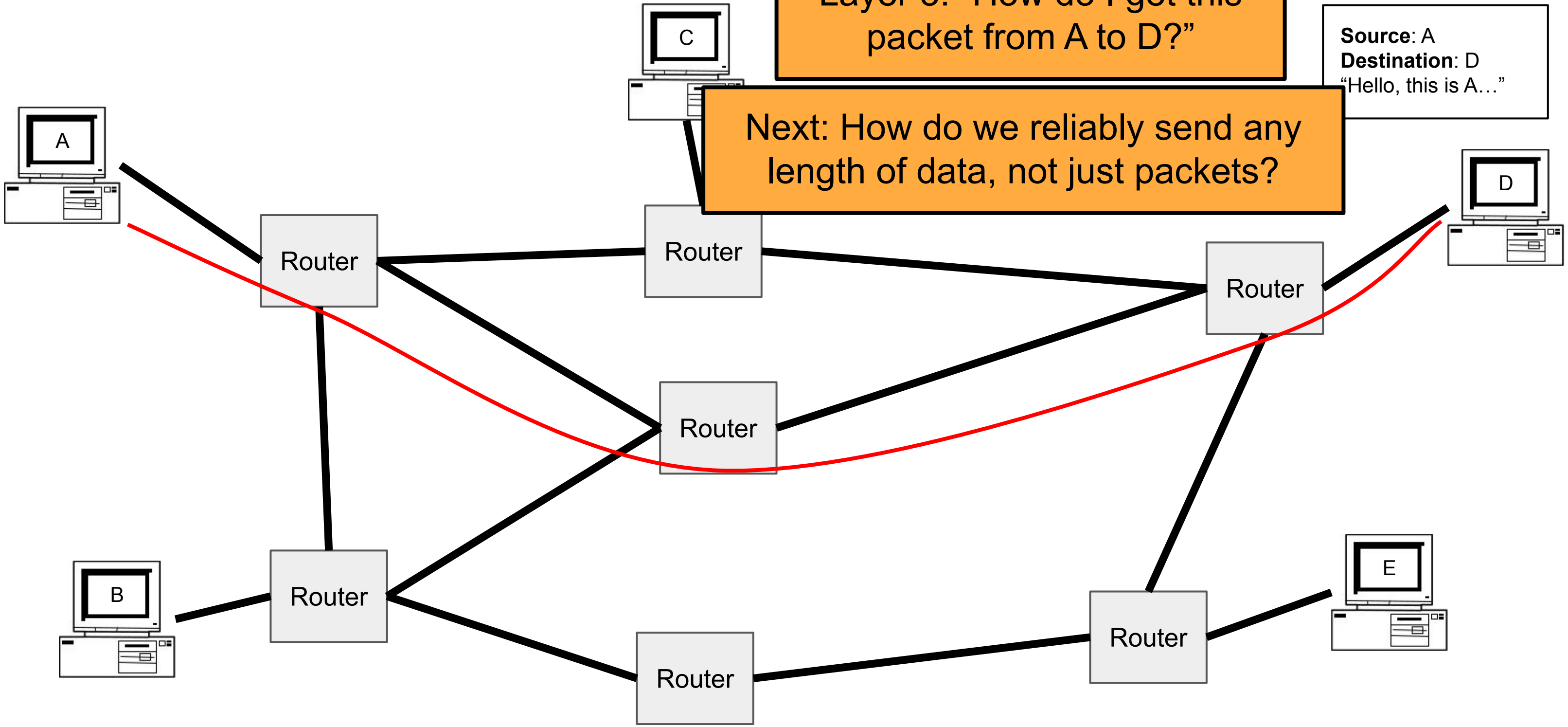
Internet Protocol (IP)

- **Internet Protocol** (**IP**): The universal layer-3 protocol that all devices use to transmit data over the Internet
- **IP address**: An address that identifies a device on the Internet
  - IPv4 is 32 bits, typically written as 4 decimal octets, e.g. **35.163.72.93**
  - IPv6 is 128 bits, typically written as 8 groups of 2 hex bytes: **2607:f140:8801::1:23**
    - If digits or groups are missing, fill with 0's, so **2607:f140:8801:0000:0000:0000:0001:0023**
  - Globally unique from any single perspective
    - For now, you can think of them as just being globally unique
  - IP addresses help nodes make decisions on where to forward the packet

Reliability

- **Reliability** ensures that packets are received correctly or, if random errors occur, not at all
  - This is implemented with a checksum
  - However, there is no cryptographic MAC, so there are no guarantees if an attacker modifies packets
- IP is **unreliable** and only provides a **best effort** delivery service, which means:
  - Packets may be lost (“dropped”)
  - Packets may be corrupted
  - Packets may be delivered out of order
- It is up to higher level protocols to ensure that the connection is reliable

Layer 3: Network Layer



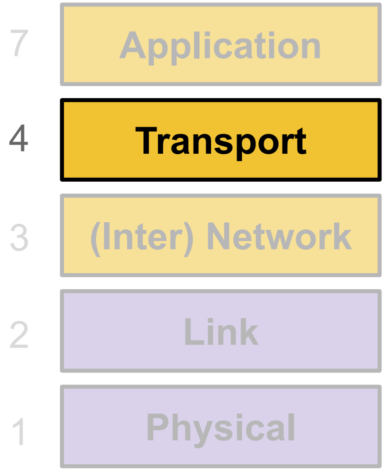
Layer 3: "How do I get this packet from A to D?"

Source: A
Destination: D
"Hello, this is A…"

Next: How do we reliably send any length of data, not just packets?

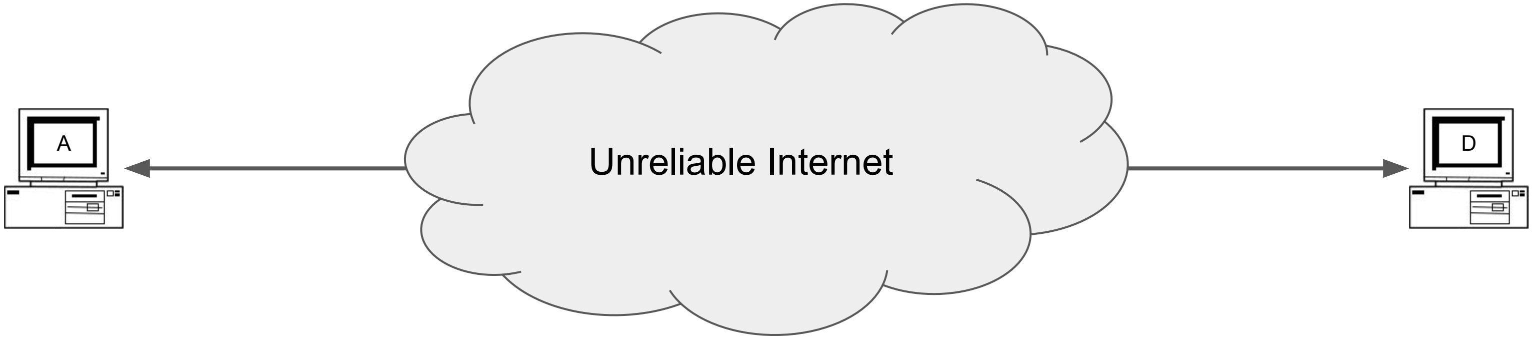
Layer 4: Transport Layer

- **Provides**: Transportation of variable-length data from any point to any other point
  - **Relies upon**: Sending packets from any device to any other device
  - Builds abstractions that are useful to applications on top of layer 3 packets
- Useful abstractions
  - **Reliability**: Transmit data reliably, in order
  - **Ports**: Provide multiple “addresses” per real IP address
- Examples
  - **TCP**: Provides reliability and ports
  - **UDP**: Provides ports, but no reliability
  - We’ll talk a lot about these protocols soon!



Layer 4: Transport Layer

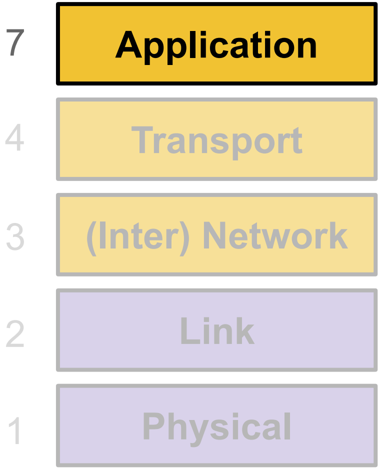
I am now sending an arbitrary length message that will probably be broken into several packets…



Layer 4: "How do I transport this arbitrary data over an unreliable medium?"

Layer 7: Application Layer

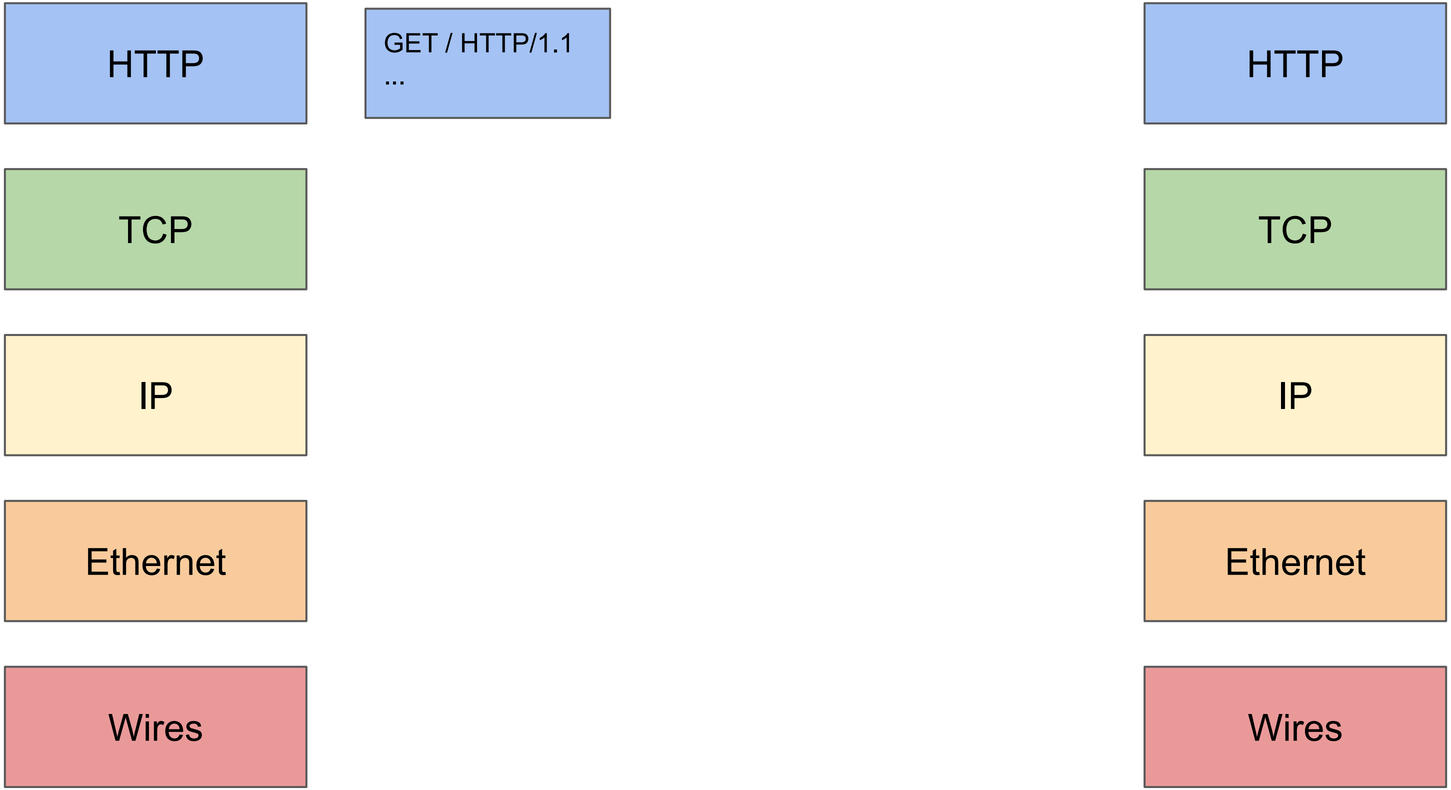
- **Provides**: Applications and services to users!
  - **Relies upon**: Transportation of variable-length data from any point to any other point
- Every online application is Layer 7
  - Web browsing
  - Online video games
  - Messaging services
  - Video calls (Zoom)



Layers of Abstraction and Headers

- As you move to lower layers, you wrap additional headers around the message
- As you move to higher layers, you peel off headers around the message
- When sending a message we go from the highest to the lowest layer
- When receiving a message we go from the lowest to highest layer

Example: HTTP Request



HTTP

GET / HTTP/1.1
...

HTTP

TCP

TCP

IP

IP

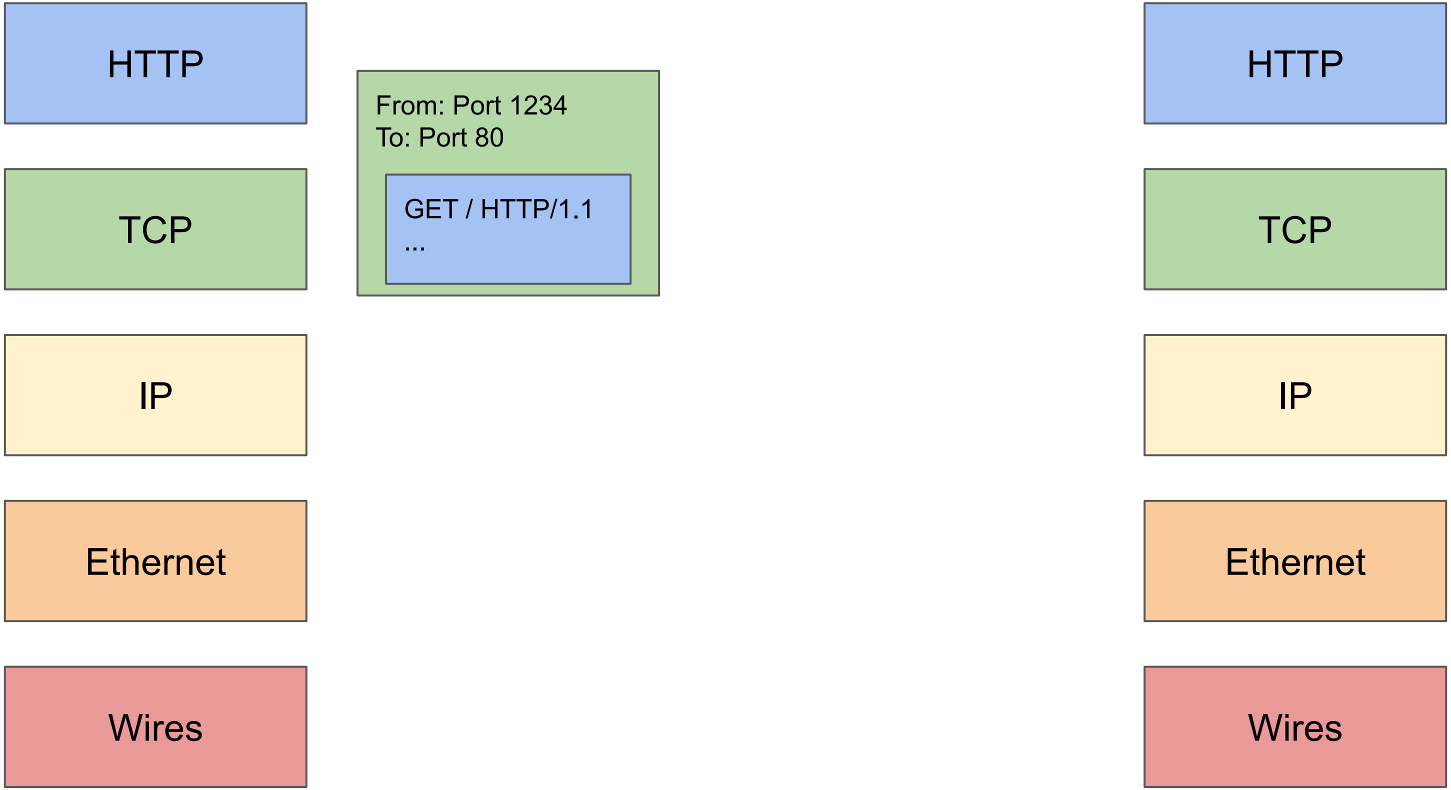
Ethernet

Ethernet

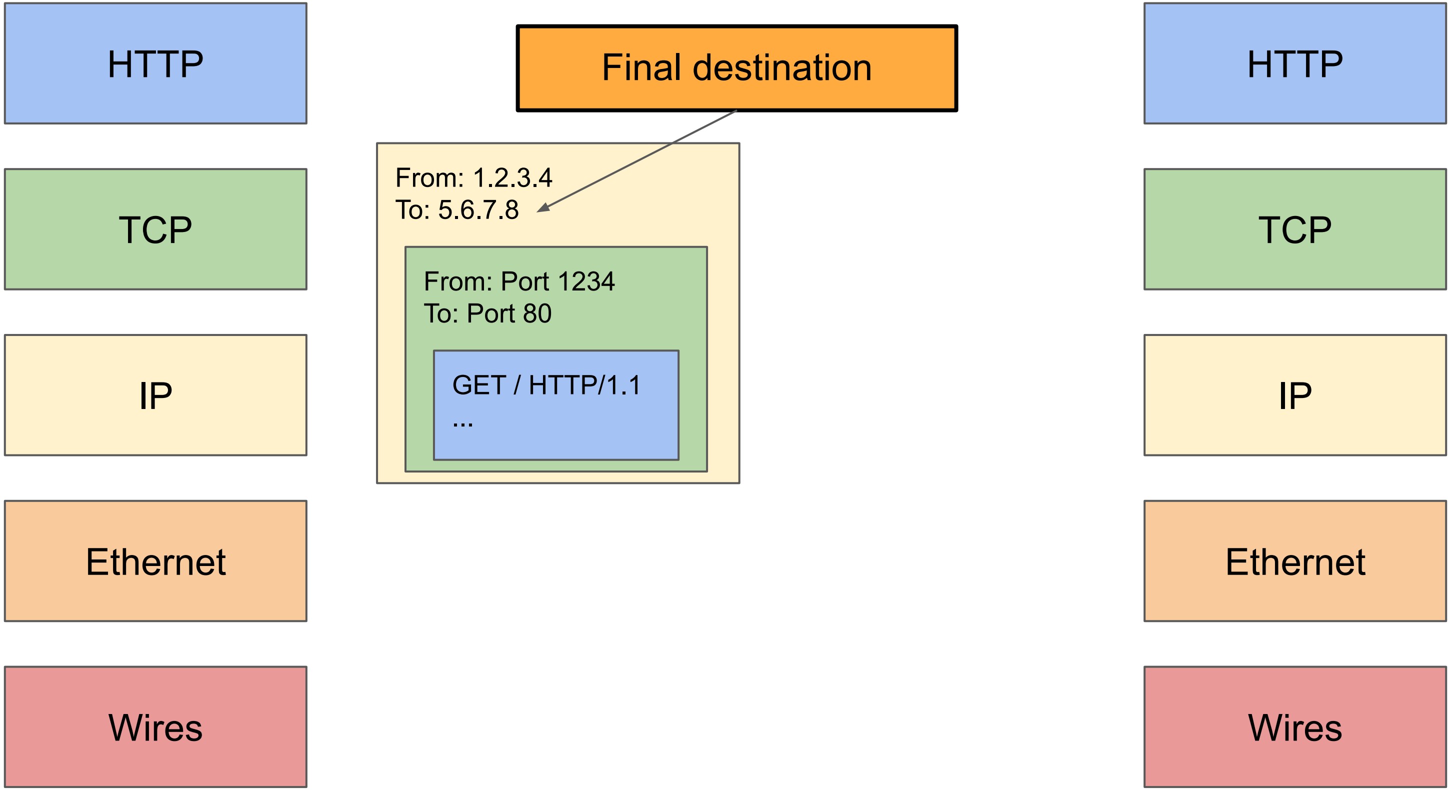
Wires

Wires

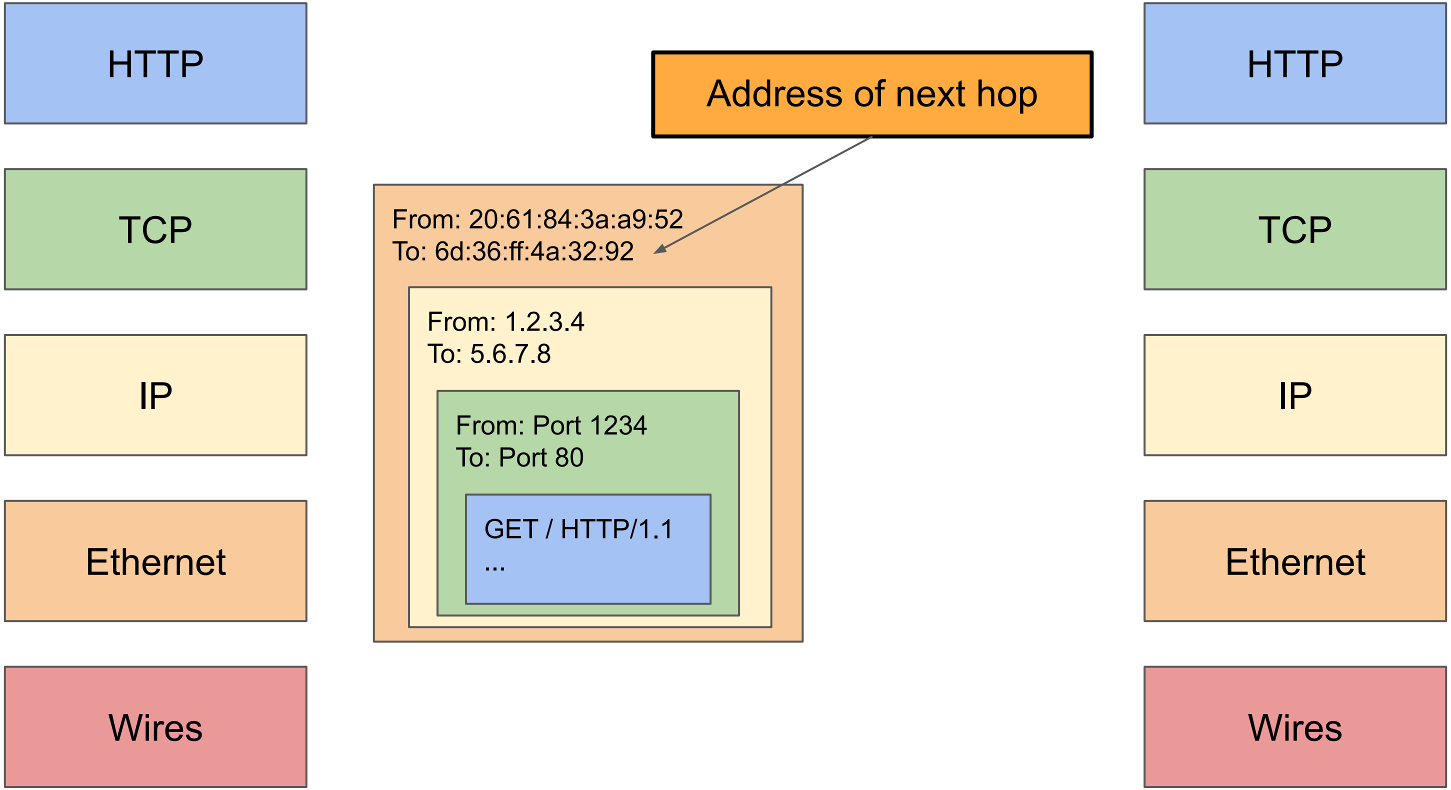
Example: HTTP Request



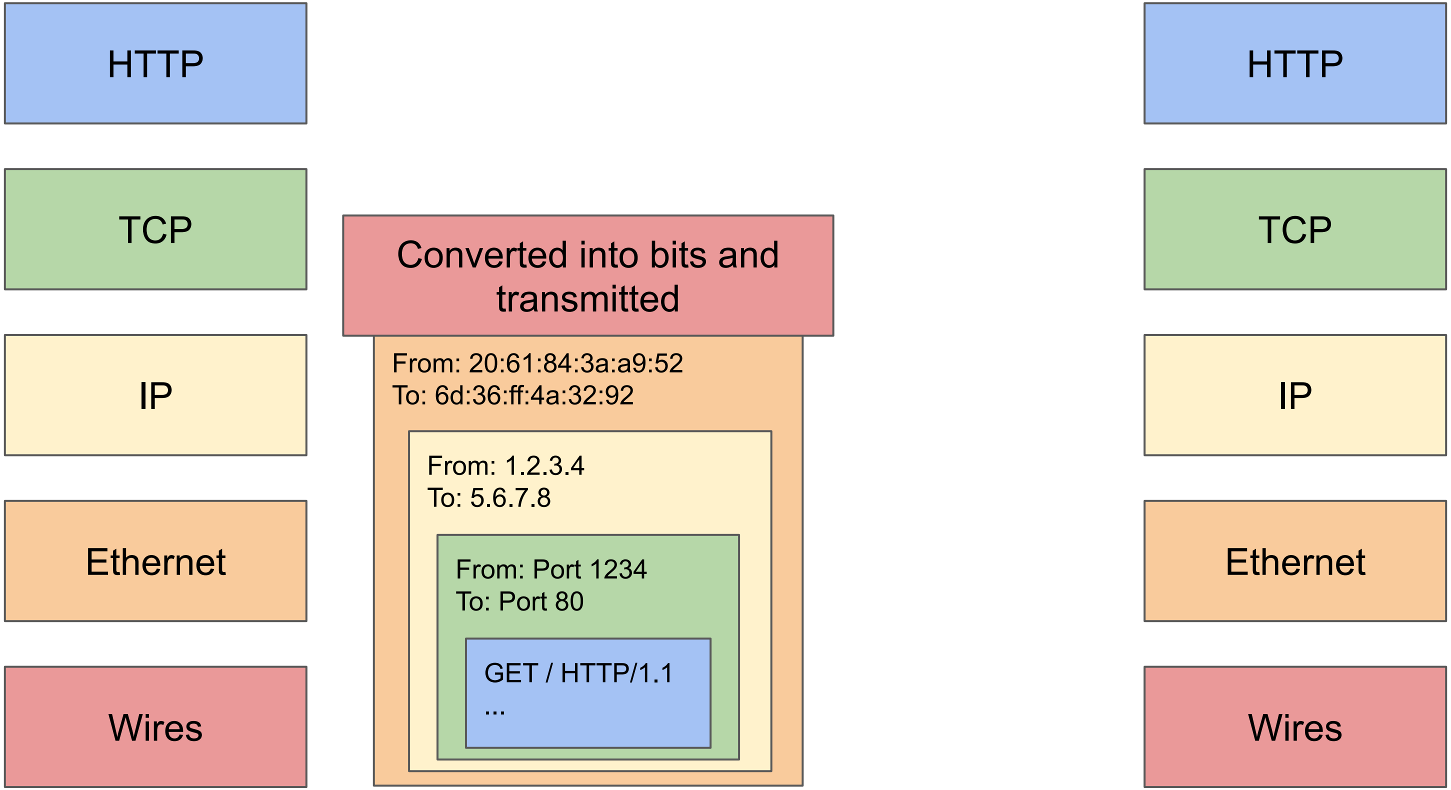
Example: HTTP Request



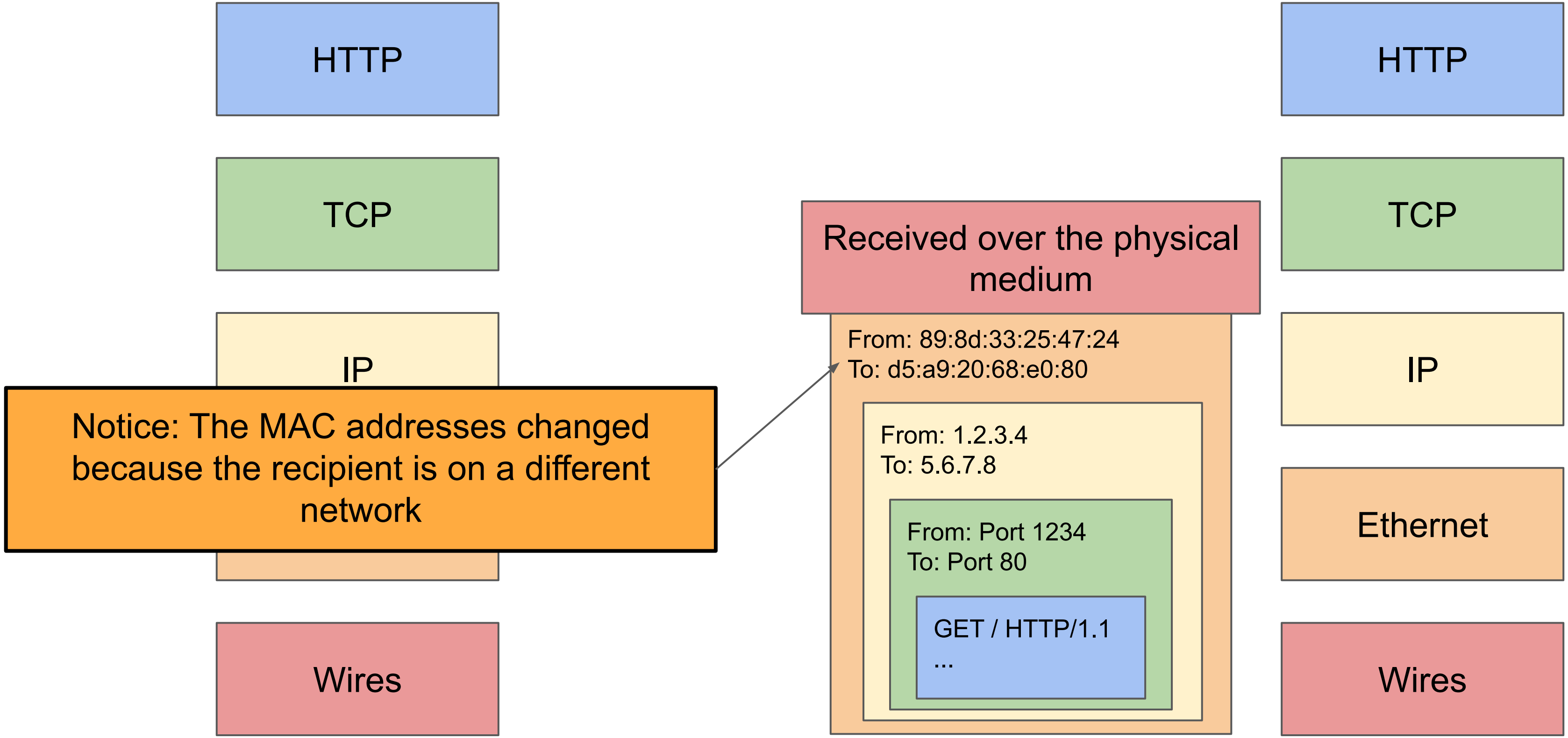
Example: HTTP Request



Example: HTTP Request



Example: HTTP Request



HTTP

TCP

IP

Wires

Received over the physical medium

From: 89:8d:33:25:47:24
To: d5:a9:20:68:e0:80

From: 1.2.3.4
To: 5.6.7.8

From: Port 1234
To: Port 80

GET / HTTP/1.1
...

HTTP

TCP

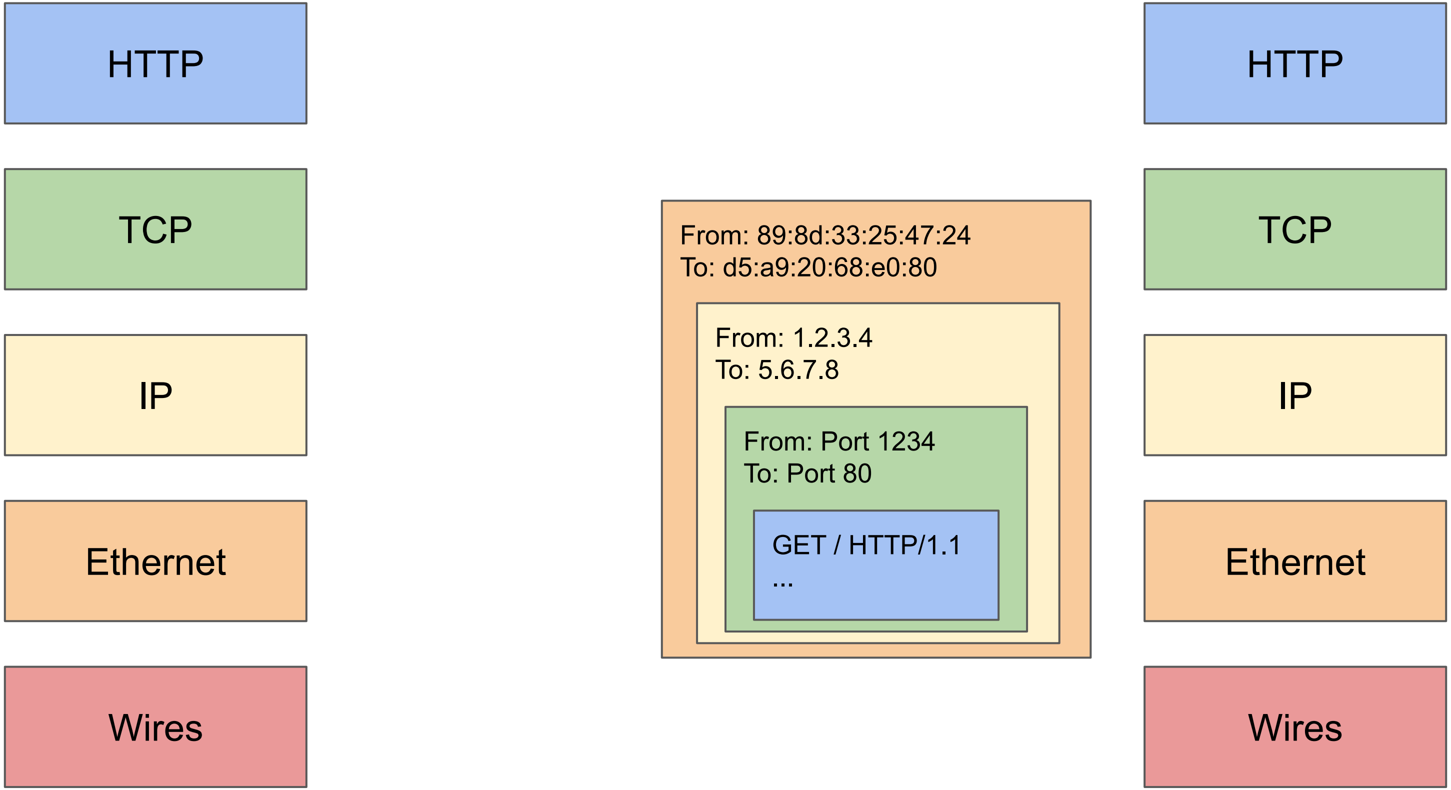
IP

Ethernet

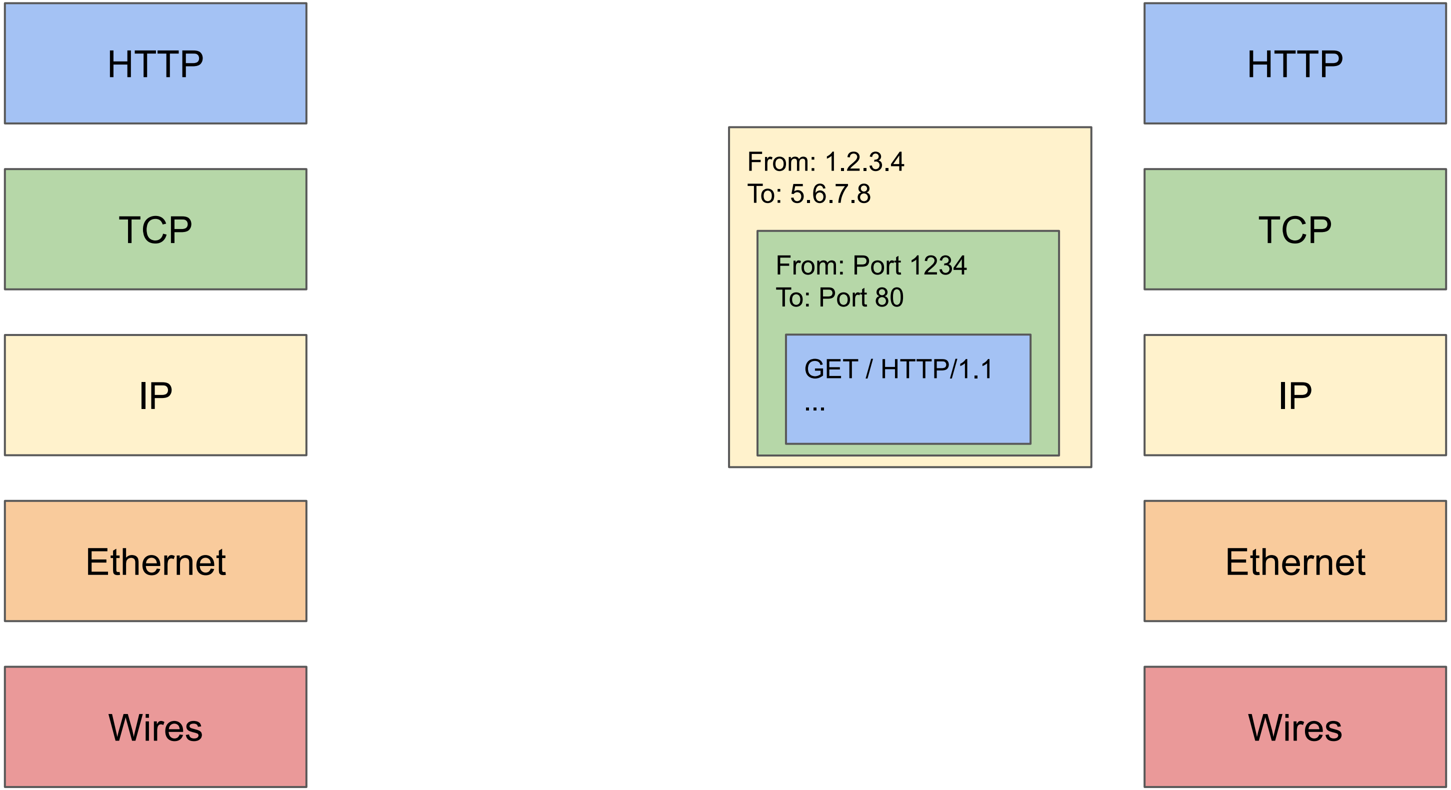
Wires

Notice: The MAC addresses changed because the recipient is on a different network

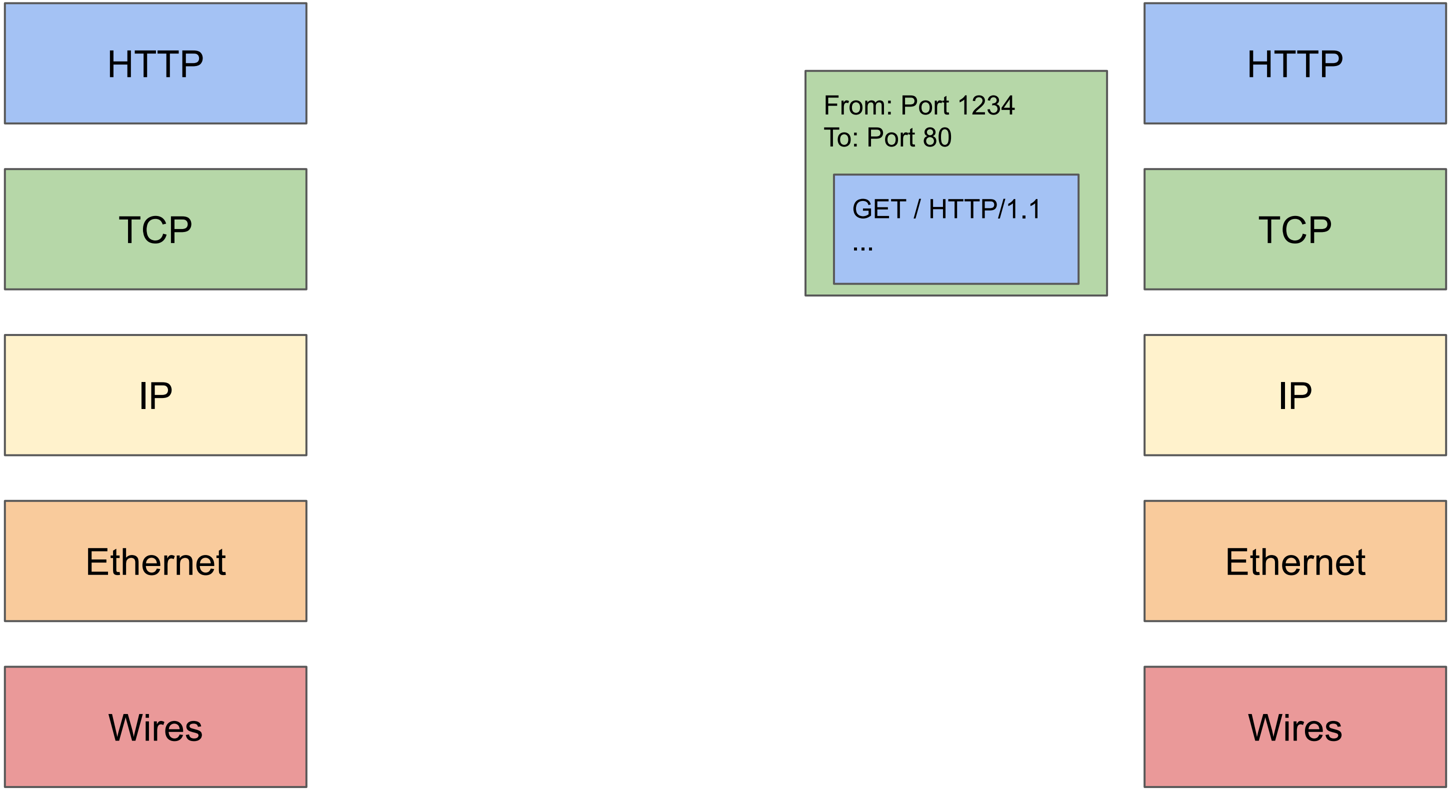
Example: HTTP Request



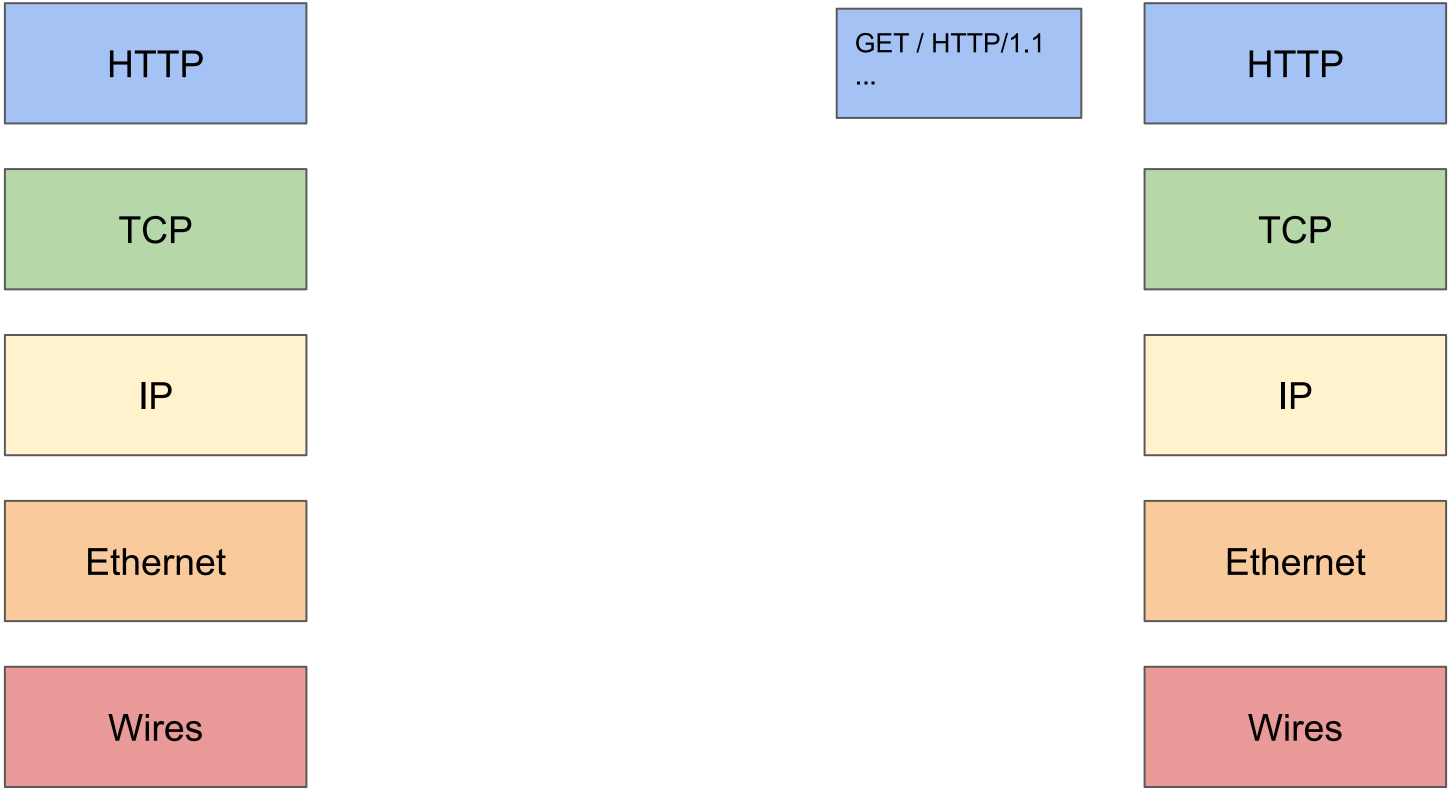
Example: HTTP Request



Example: HTTP Request



Example: HTTP Request



HTTP

TCP

IP

Ethernet

Wires

GET / HTTP/1.1
...

HTTP

TCP

IP

Ethernet

Wires

Example: HTTP Request

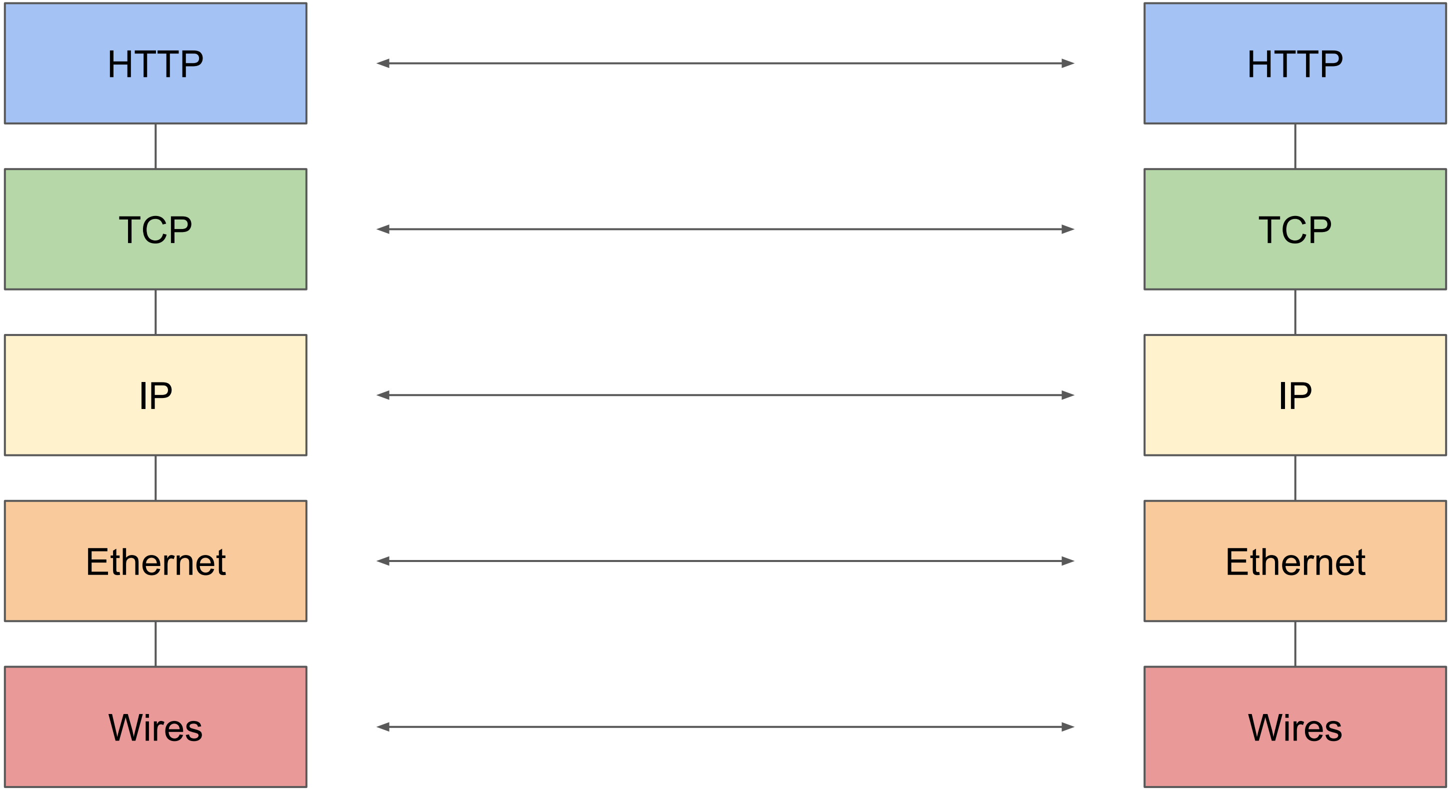
Relies upon: Transport of data

Provides: Transport of data
Relies upon: Global packet delivery

Provides: Global packet delivery
Relies upon: Local frame delivery

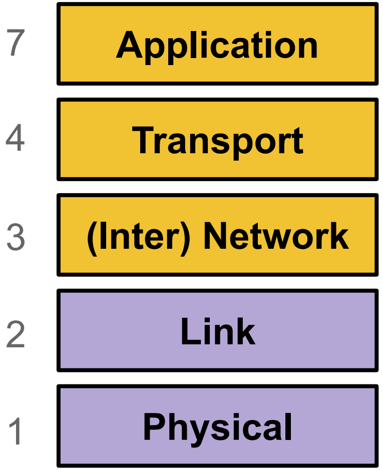
Provides: Local frame delivery
Relies upon: Communication of bits

Provides: Communication of bits



Summary: Intro to Networking

- Internet: A global network of computers
  - Protocols: Agreed-upon systems of communication
- OSI model: A layered model of protocols
  - Layer 1: Communication of bits
  - Layer 2: Local frame delivery
    - Ethernet: The most common Layer 2 protocol
    - MAC addresses: 6-byte addressing system used by Ethernet
  - Layer 3: Global packet delivery
    - IP: The universal Layer 3 protocol
    - IP addresses: 4-byte (or 16-byte) addressing system used by IP
  - Layer 4: Transport of data (more on this next time)
  - Layer 7: Applications and services (the web)



7 Application

4 Transport

3 (Inter) Network

2 Link

1 Physical

Next: Low-Level Network Attacks

- Network Attackers
  - Man-in-the-middle attacker
  - On-path attacker
  - Off-path attacker
- ARP: Translate IP addresses to MAC addresses
- DHCP: Get configurations when first connecting to a network
- WPA: Communicate securely in a wireless local network

Network Attackers

Types of Network Attackers

- Threat model: There are 3 types of attackers we'll consider

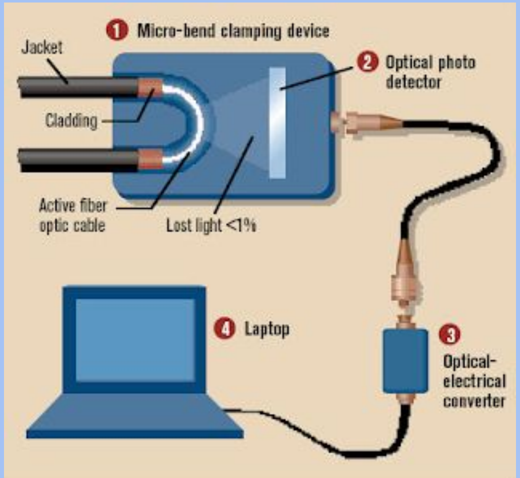
	Can modify or delete packets	Can read packets
Man-in-the-middle/In-path attacker	✓	✓
Man-on-the-side/On-path attacker		✓
Off-path attacker		

Spoofing

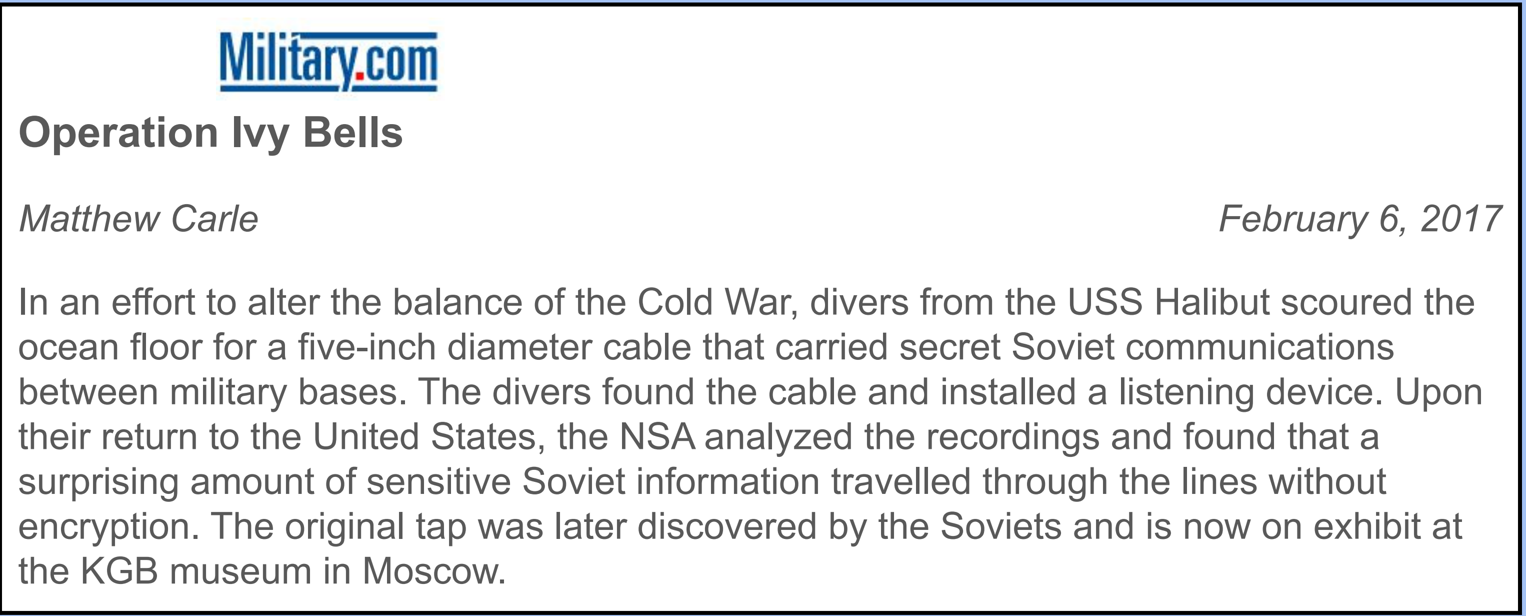
- Anybody can send their own packets through the network
- **Spoofing**: Lying about the identity of the sender
    - Example: Mallory sends a message and says the message is from Alice
    - The attacker can lie about the *source address* in the packet header
- All types of attackers can spoof packets
    - However, some spoofing attacks may be harder if the attacker can’t read or modify packets

Real-World On-Path Attackers

- How might a real-life attacker read packets?
- Layer 1 attack: Use a special device to read bits being transmitted across space



Real-World On-Path Attackers

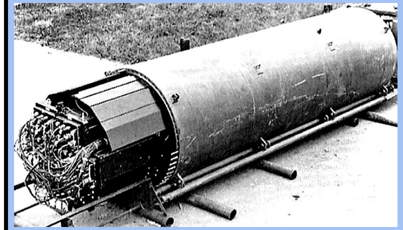


Military.com

Operation Ivy Bells

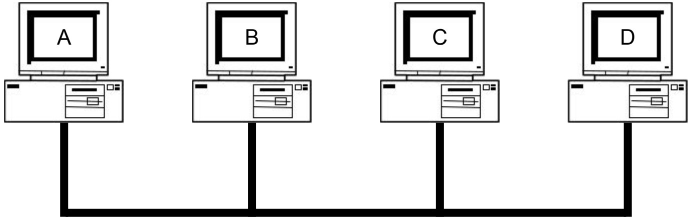
Matthew Carle *February 6, 2017*

In an effort to alter the balance of the Cold War, divers from the USS Halibut scoured the ocean floor for a five-inch diameter cable that carried secret Soviet communications between military bases. The divers found the cable and installed a listening device. Upon their return to the United States, the NSA analyzed the recordings and found that a surprising amount of sensitive Soviet information travelled through the lines without encryption. The original tap was later discovered by the Soviets and is now on exhibit at the KGB museum in Moscow.



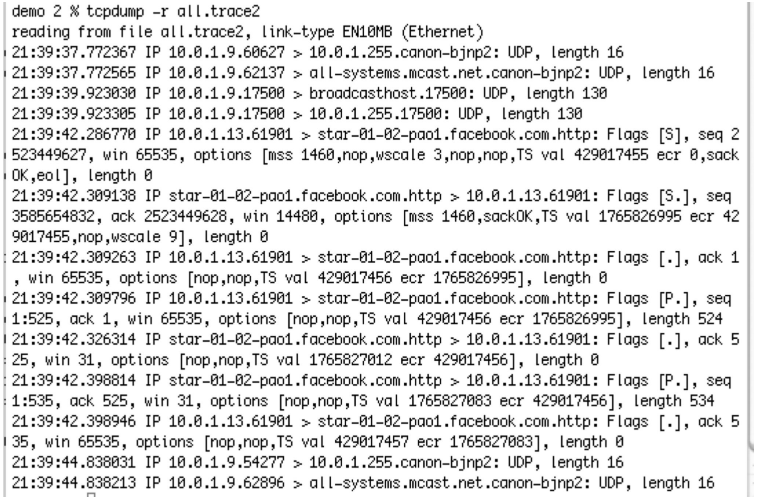
Real-World On-Path Attackers

- Layer 2 attack: Read packets sent across the local area network (LAN)
- Recall: A LAN is a network of connected machines
  - Any machine on the LAN can send packets to any other machine on the LAN
- Some LANs use **broadcast technologies**
  - Every packet gets sent to every machine on the LAN
  - Each machine agrees to ignore packets where the destination is a different machine
- A machine can break the agreement and read packets meant for other machines
  - This is called **promiscuous mode**
  - May require root access on the machine

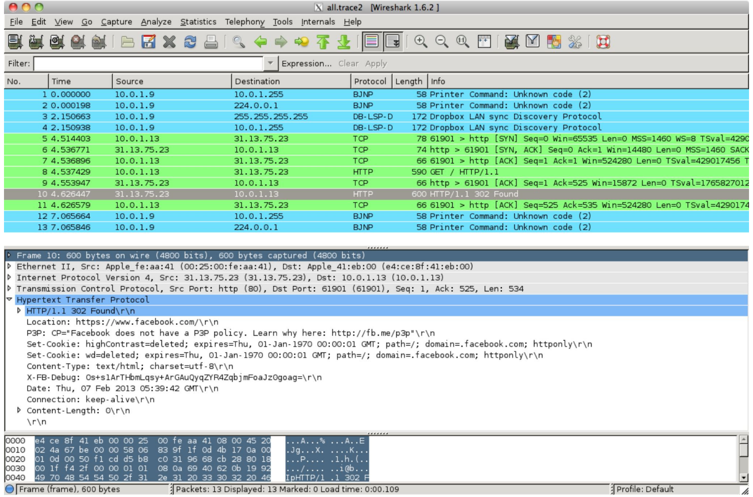


Real-World On-Path Attackers

- `tcpdump`: A program for reading packets on the local network
  - Uses promiscuous mode to read other machines' packets in broadcast techonlogies
- Wireshark: A graphical user interface (GUI) for analyzing `tcpdump` packets



```
demo 2 % tcpdump -r all.trace2
reading from file all.trace2, link-type EN10MB (Ethernet)
21:39:37.772367 IP 10.0.1.9.60627 > 10.0.1.255.canon-bjnp2: UDP, length 16
21:39:37.772565 IP 10.0.1.9.62137 > all-systems.mcast.net.canon-bjnp2: UDP, length 16
21:39:39.923030 IP 10.0.1.9.17500 > broadcasthost.17500: UDP, length 130
21:39:39.923305 IP 10.0.1.9.17500 > 10.0.1.255.17500: UDP, length 130
21:39:42.286770 IP 10.0.1.13.61901 > star-01-02-pao1.facebook.com.http: Flags [S], seq 2
523449627, win 65535, options [mss 1460,nop,wscale 3,nop,nop,TS val 429017455 ecr 0,sack
OK,eol], length 0
21:39:42.309138 IP star-01-02-pao1.facebook.com.http > 10.0.1.13.61901: Flags [S.], seq
3585654832, ack 2523449628, win 14480, options [mss 1460,sackOK,TS val 1765826995 ecr 42
9017455,nop,wscale 9], length 0
21:39:42.309263 IP 10.0.1.13.61901 > star-01-02-pao1.facebook.com.http: Flags [.], ack 1
, win 65535, options [nop,nop,TS val 429017456 ecr 1765826995], length 0
21:39:42.309796 IP 10.0.1.13.61901 > star-01-02-pao1.facebook.com.http: Flags [P.], seq
1:525, ack 1, win 65535, options [nop,nop,TS val 429017456 ecr 1765826995], length 524
21:39:42.326314 IP star-01-02-pao1.facebook.com.http > 10.0.1.13.61901: Flags [.], ack 5
25, win 31, options [nop,nop,TS val 1765827012 ecr 429017456], length 0
21:39:42.398814 IP star-01-02-pao1.facebook.com.http > 10.0.1.13.61901: Flags [P.], seq
1:535, ack 525, win 31, options [nop,nop,TS val 1765827083 ecr 429017456], length 534
21:39:42.398946 IP 10.0.1.13.61901 > star-01-02-pao1.facebook.com.http: Flags [.], ack 5
35, win 65535, options [nop,nop,TS val 429017457 ecr 1765827083], length 0
21:39:44.838031 IP 10.0.1.9.54277 > 10.0.1.255.canon-bjnp2: UDP, length 16
21:39:44.838213 IP 10.0.1.9.62896 > all-systems.mcast.net.canon-bjnp2: UDP, length 16
```



all.trace2 [Wireshark 1.6.2]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply

```
No.  Time      Source        Destination      Protocol  Length Info
   1 0.000000  10.0.1.9      10.0.1.255       BJNP          58 Printer Command: Unknown code (2)
   2 0.000198  10.0.1.9      224.0.0.1        BJNP          58 Printer Command: Unknown code (2)
   3 2.150663  10.0.1.9      255.255.255.255  DB-LSP-D     172 Dropbox LAN sync Discovery Protocol
   4 2.150938  10.0.1.9      10.0.1.255       DB-LSP-D     172 Dropbox LAN sync Discovery Protocol
   5 4.514403  10.0.1.13     31.13.75.23      TCP           78 61901 > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=8 TSval=4290
   6 4.536771  31.13.75.23   10.0.1.13        TCP           74 http > 61901 [SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1460 SACK
   7 4.536896  10.0.1.13     31.13.75.23      TCP           66 61901 > http [ACK] Seq=1 Ack=1 Win=524280 Len=0 TSval=429017456 T
   8 4.537429  10.0.1.13     31.13.75.23      HTTP         590 GET / HTTP/1.1
   9 4.553947  31.13.75.23   10.0.1.13        TCP           66 http > 61901 [ACK] Seq=1 Ack=525 Win=15872 Len=0 TSval=1765827012
  10 4.626447  31.13.75.23   10.0.1.13        HTTP         600 HTTP/1.1 302 Found
  11 4.626579  10.0.1.13     31.13.75.23      TCP           66 61901 > http [ACK] Seq=525 Ack=535 Win=524280 Len=0 TSval=4290174
  12 7.065664  10.0.1.9      10.0.1.255       BJNP          58 Printer Command: Unknown code (2)
  13 7.065846  10.0.1.9      224.0.0.1        BJNP          58 Printer Command: Unknown code (2)
```

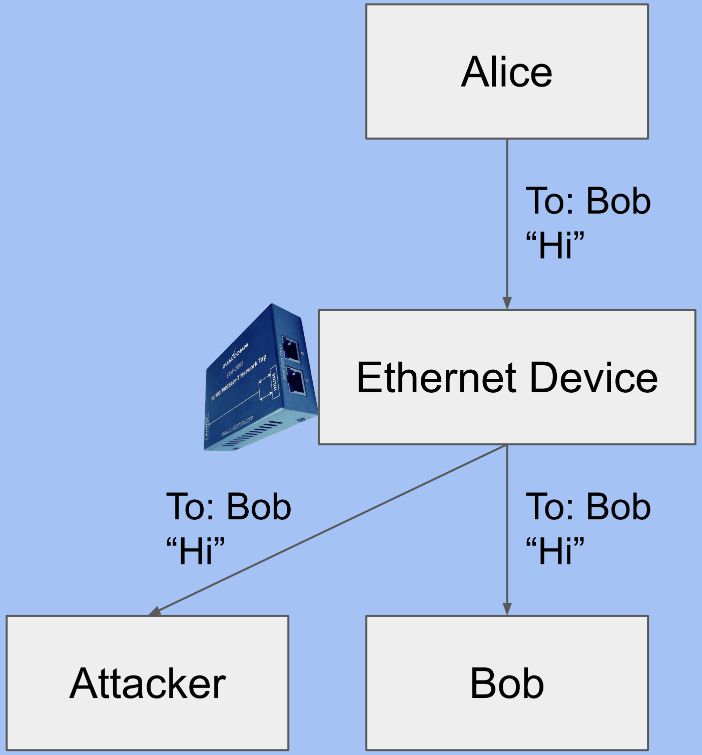
```
Frame 10: 600 bytes on wire (4800 bits), 600 bytes captured (4800 bits)
Ethernet II, Src: Apple_fe:aa:41 (00:25:00:fe:aa:41), Dst: Apple_41:eb:00 (e4:ce:8f:41:eb:00)
Internet Protocol Version 4, Src: 31.13.75.23 (31.13.75.23), Dst: 10.0.1.13 (10.0.1.13)
Transmission Control Protocol, Src Port: http (80), Dst Port: 61901 (61901), Seq: 1, Ack: 525, Len: 534
Hypertext Transfer Protocol
  HTTP/1.1 302 Found\r\n
  Location: https://www.facebook.com/\r\n
  P3P: CP="Facebook does not have a P3P policy. Learn why here: http://fb.me/p3p"\r\n
  Set-Cookie: highContrast=deleted; expires=Thu, 01-Jan-1970 00:00:01 GMT; path=/; domain=.facebook.com; httponly\r\n
  Set-Cookie: wd=deleted; expires=Thu, 01-Jan-1970 00:00:01 GMT; path=/; domain=.facebook.com; httponly\r\n
  Content-Type: text/html; charset=utf-8\r\n
  X-FB-Debug: Os+s1ArTHbmLqsy+ArGAuQyqZYR4ZqbjmFoaJzOgoag=\r\n
  Date: Thu, 07 Feb 2013 05:39:42 GMT\r\n
  Connection: keep-alive\r\n
  Content-Length: 0\r\n
  \r\n
```

```
0000  e4 ce 8f 41 eb 00 00 25  00 fe aa 41 08 00 45 20   ...A...% ...A..E
0010  02 4a 67 be 00 00 58 06  83 9f 1f 0d 4b 17 0a 00   .Jg...X. ....K...
0020  01 0d 00 50 f1 cd d5 b8  c0 31 96 68 cb 28 80 18   ...P.... .1.h.(..
0030  00 1f f4 2f 00 00 01 01  08 0a 69 40 62 0b 19 92   .../.... ..i@b...
0040  49 70 48 54 54 50 2f 31  2e 31 20 33 30 32 20 46   IpHTTP/1 .1 302 F
```

```
Frame (frame), 600 bytes    Packets: 13 Displayed: 13 Marked: 0 Load time: 0:00.109    Profile: Default
```

Real-World On-Path Attackers

- Some layer 2 (Ethernet) devices can be configured to also send a copy of every packet to the attacker
  - Many switches support this through “port mirroring”
  - Or you can use dedicated Ethernet taps
- Example: DualComm ETAP-2003
  - Cost: $200
  - Powered with USB (no extra power supply needed)
  - ETAP-2003R extra fun: Attacker can also send packets



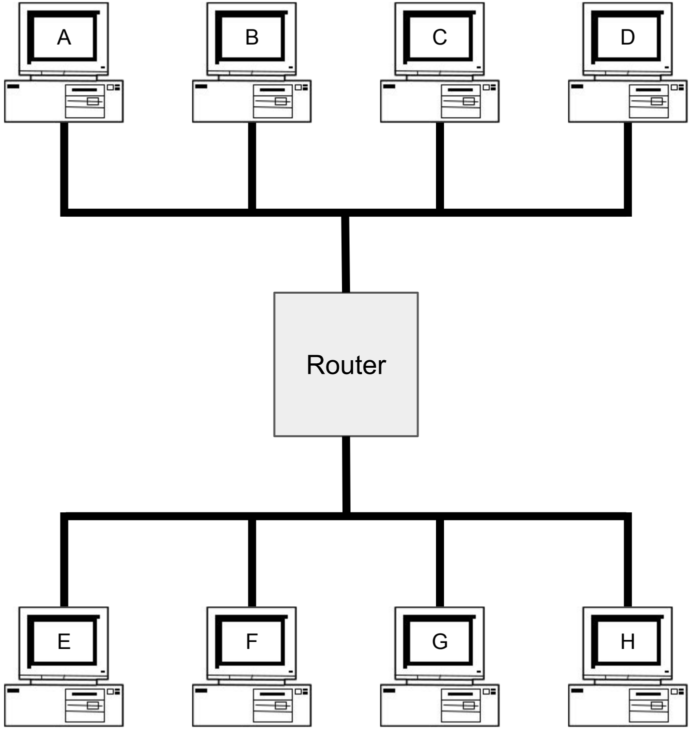
The Law and Sniffing Packets

- You are allowed to sniff packets on your own network
  - After all, it is your computers you are using
  - Network administrators are allowed for network operation
  - *Strongly encourage* you to do so at home and see what you see!
- It is both **grossly immoral** and **highly illegal** to sniff traffic otherwise
  - It is called “wiretapping”
- So **do not do this** at Starbucks or other networks
  - Unless you add a filter to only include packets to/from your computer for debugging purposes

Address Resolution Protocol (ARP)

Review: Layer 2 and Layer 3

- Local area network (LAN): A set of machines connected in a local network
  - The MAC identifies devices on layer 2
- Internet protocol (IP): Many LANs connected together with routers
  - The IP identifies devices on layer 3



Address Resolution Protocol (ARP)

- **ARP**: Translates layer 3 IP addresses to layer 2 MAC addresses
  - Example: Alice wants to send a message to Bob on the local network, but Alice only knows Bob's IP address (**1.2.3.4**). To use layer 2 protocols, she must learn Bob's MAC address.
- Steps of the protocol
  a. Alice checks her cache to see if she already knows Bob's MAC address.
  b. If Bob's MAC address is not in the cache, Alice **broadcasts** to everyone on the LAN: "What is the MAC address of **1.2.3.4**?"
  c. Bob responds by sending a message only to Alice: "My IP is **1.2.3.4** and my MAC address is **ca:fe:f0:0d:be:ef**." Everyone else does nothing.
  d. Alice caches Bob's MAC address.

Address Resolution Protocol (ARP)

Alice knows Bob’s IP address (`1.2.3.4`) but wants to learn Bob’s MAC address.

Alice’s cache	
IP	MAC

Alice
Bob
Charlie
Dave
Router

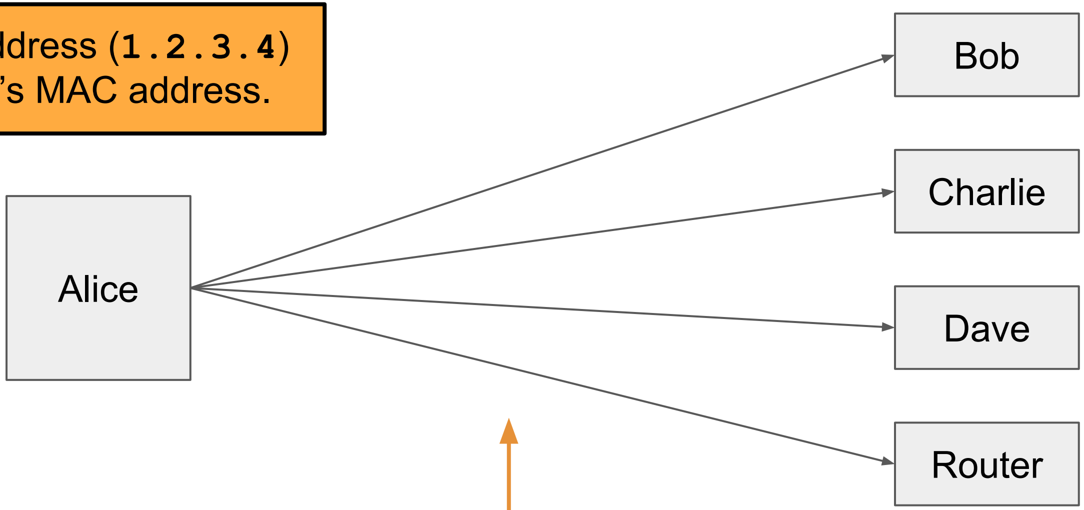
1. Alice checks her cache to see if she already knows the MAC address corresponding to `1.2.3.4`.

Since her cache is empty, she must make a request to find out.

Address Resolution Protocol (ARP)

Alice knows Bob’s IP address (`1.2.3.4`) but wants to learn Bob’s MAC address.

Alice’s cache	
IP	MAC

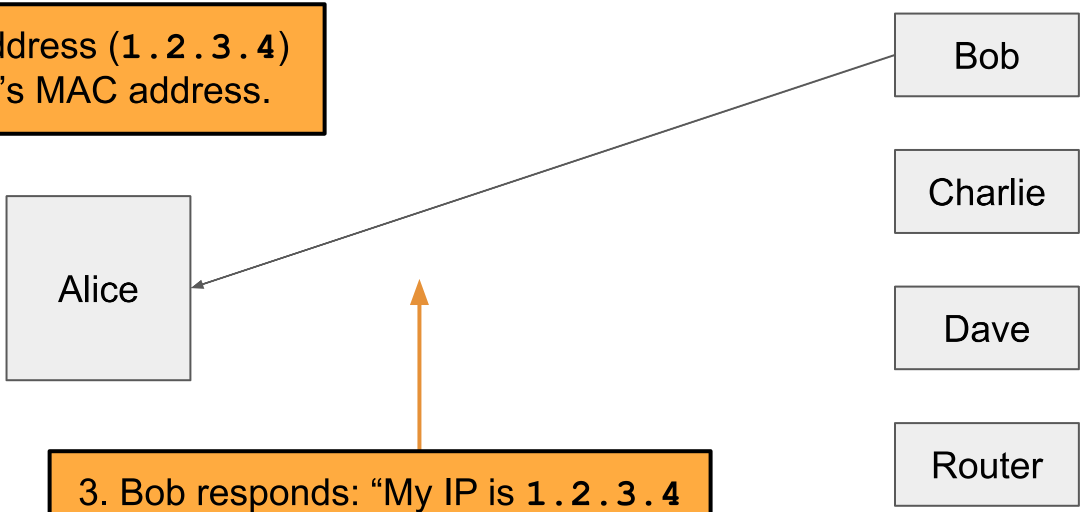


2. Alice asks everyone else on the local network: “What is the MAC address of `1.2.3.4`?”

Address Resolution Protocol (ARP)

Alice knows Bob’s IP address (`1.2.3.4`) but wants to learn Bob’s MAC address.

Alice’s cache	
IP	MAC



3. Bob responds: “My IP is `1.2.3.4` and my MAC address is `ca:fe:f0:0d:be:ef`.”

Everybody else ignores the request.

Address Resolution Protocol (ARP)

Alice knows Bob’s IP address (`1.2.3.4`) but wants to learn Bob’s MAC address.

Alice’s cache	
IP	MAC

`1.2.3.4`	`ca:fe:f0:`
	`0d:be:ef`

4. Alice adds Bob’s MAC address to her cache.

Alice
Bob
Charlie
Dave
Router

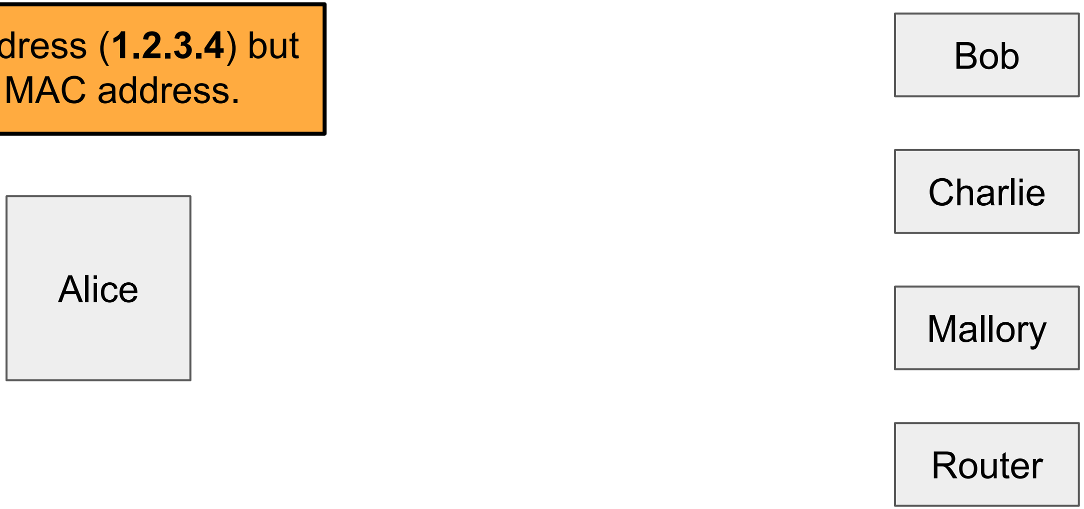
Address Resolution Protocol (ARP)

- If Bob is outside of the LAN, Alice knows this
  - Bob’s IP is not on the same “subnet” as Alice
- But Alice knows the IP address of the “Gateway router”
  - Recall: The router’s job is to make sure that the packet will be forwarded towards Bob (Layer 3)
- So instead Alice generates an ARP request for the gateway router
  - Layer 2 MAC address of the frame is set to the router
  - Layer 3 IP address of the packet remains set as Bob's
  - The router will forward the packet to some other LAN to get it closer to Bob

Attacks on ARP

Alice knows Bob's IP address (**1.2.3.4**) but wants to learn Bob's MAC address.

Alice's cache	
IP	MAC



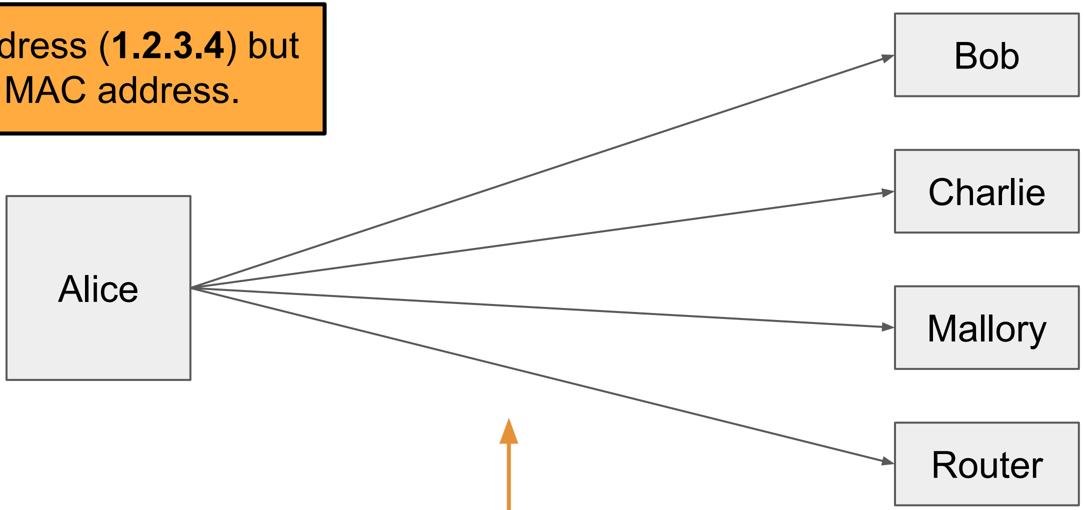
1. Alice checks her cache to see if she already knows the MAC address corresponding to **1.2.3.4**.

Since her cache is empty, she must make a request to find out.

Attacks on ARP

Alice knows Bob’s IP address (**1.2.3.4**) but wants to learn Bob’s MAC address.

Alice’s cache	
IP	MAC

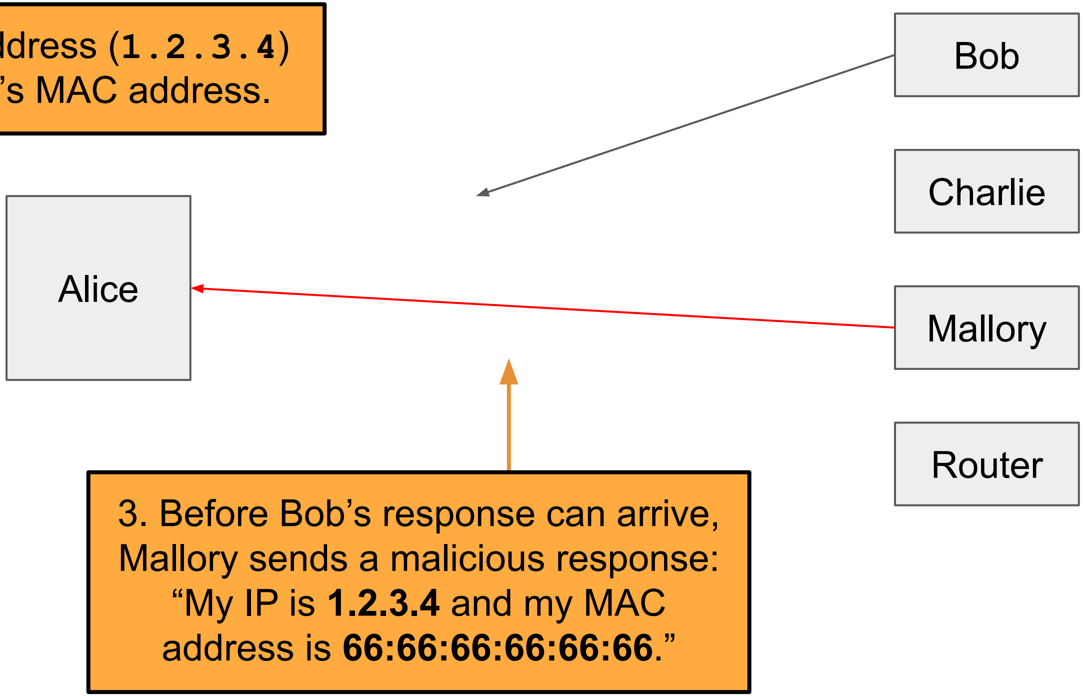


2. Alice asks everyone else on the local network: “What is the MAC address of **1.2.3.4**?”

Attacks on ARP

Alice knows Bob’s IP address (`1.2.3.4`) but wants to learn Bob’s MAC address.

Alice’s cache	
IP	MAC



3. Before Bob’s response can arrive, Mallory sends a malicious response: “My IP is **1.2.3.4** and my MAC address is **66:66:66:66:66:66**.”

Attacks on ARP

Alice knows Bob’s IP address (`1.2.3.4`) but wants to learn Bob’s MAC address.

Alice’s cache

IP	MAC
1.2.3.4	**66:66:66:66:66:66**

4. Alice adds Mallory’s malicious address to her cache.

Alice
Bob
Charlie
Mallory
Router

Attack: ARP Spoofing

- Alice has no way of verifying the ARP response
  - Spoofing: Any attacker on the network can claim to have the requested IP address
- Alice is only expecting one machine to respond, so she will accept the first response
  - **Race condition**: As long as the attacker responds faster, the requester will accept the attacker’s response
- ARP spoofing requires Mallory to be in the same LAN as Alice
- ARP spoofing lets Mallory become a man-in-the-middle (MITM) attacker
  - Alice thinks that Bob’s MAC address is **66:66:66:66:66:66** (Mallory’s MAC address)
  - When Alice sends a message to Bob, she is actually sending the message to Mallory
  - Mallory can modify the message and then send the modified message to Bob

ARP Spoofing: Defenses

- Network switches
    - When Alice wants to send a message to Bob, she sends the message to a switch on the LAN
    - The switch maintains a cache of MAC to port (physical connection) mappings
    - If Bob's MAC address is in the cache, the switch sends the message directly to Bob
    - Otherwise, the switch broadcasts the message to all computers
        - Greatly improves efficiency as now the L1 network is no longer a shared media
- Enterprise-class switches have additional optional features
    - Security: An additional IP/MAC cache that responds first, preventing the attacker from seeing repeated requests
    - Security: Only authorized MAC addresses can connect to specific ports—access control
    - Isolation: Virtual local area networks (VLANs), which splits a single LAN into isolated parts
- Tools like **`arpwatch`** track ARP responses and make sure that there is no suspicious activity