Intro to the Internet (continued), ARP, DHCP, and WPA

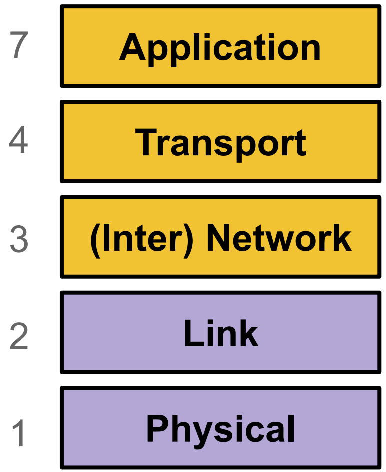
CS 161 Spring 2022 - Lecture 17

Announcements

- Project 2 Design Doc Grades are being released on a rolling basis
  - 84% of submissions have been graded & released. The rest will be out by tonight.
- Sign up for a Project 2 design review with a TA!
  - There are still slots open for tomorrow (Wednesday)
  - We may offer a handful of slots after spring break, but we **strongly encourage** signing up this week to guarantee a review slot with a TA
  - Attending a design review is the **only** way to earn back credit for points you may have missed
  - See Piazza for more details
- Homework 5 has been released and is due on Friday, April 1 at 11:59 PM
- Office hours this week have been cancelled for design reviews
- LOST discussion section this week has been cancelled for design reviews

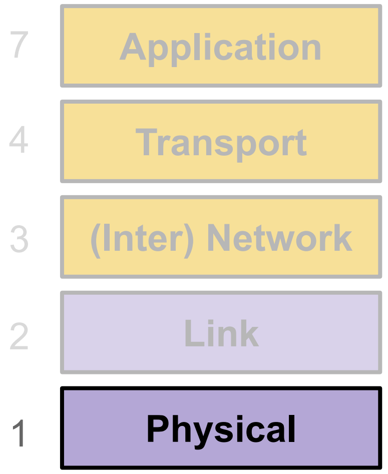
OSI Model

- **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



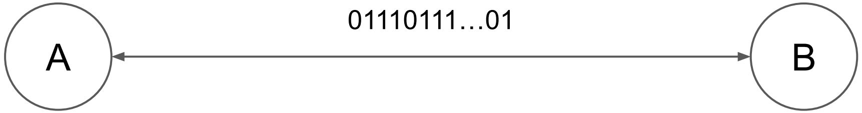
Layer 1: Physical Layer

- **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)

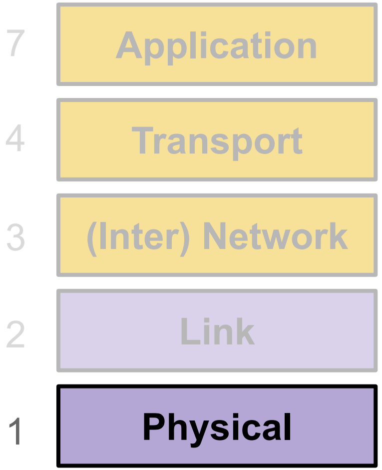


Layer 1: Physical Layer

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

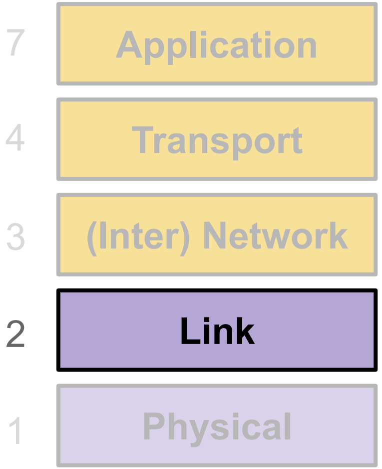


Next: How do we talk to more than one device?



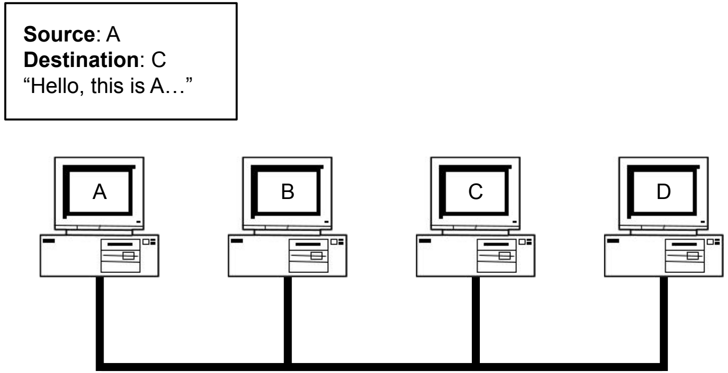
Layer 2: Link Layer

- **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?")

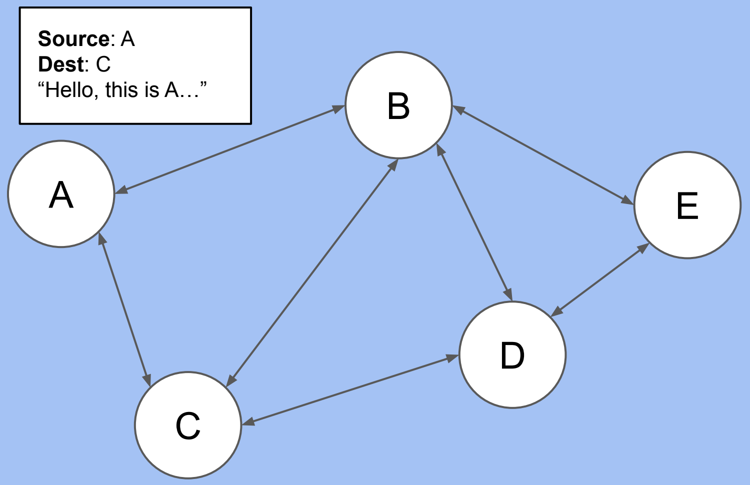


Ethernet and MAC Addresses

- **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

- 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"



Ethernet and MAC Addresses

<table>
<tr><td colspan="2">Source MAC Address (6 bytes)</td></tr>
<tr><td colspan="2">Destination MAC Address (6 bytes)</td></tr>
<tr><td>VLAN Tag (4 bytes)</td><td>Type (2 bytes)</td></tr>
<tr><td colspan="2">Data (variable-length)</td></tr>
</table>

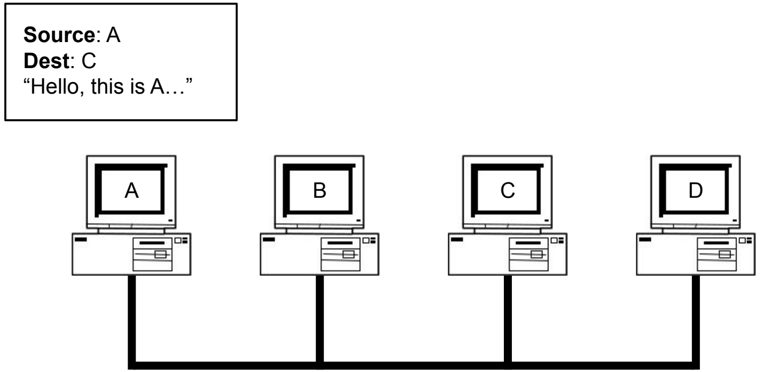
Ethernet header

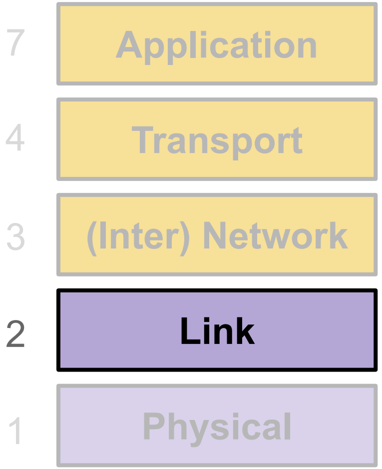
Layer 3: Network Layer

- 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

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?"

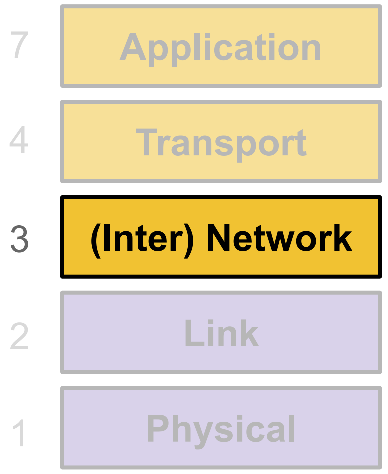




Next: How do we address every device in existence?

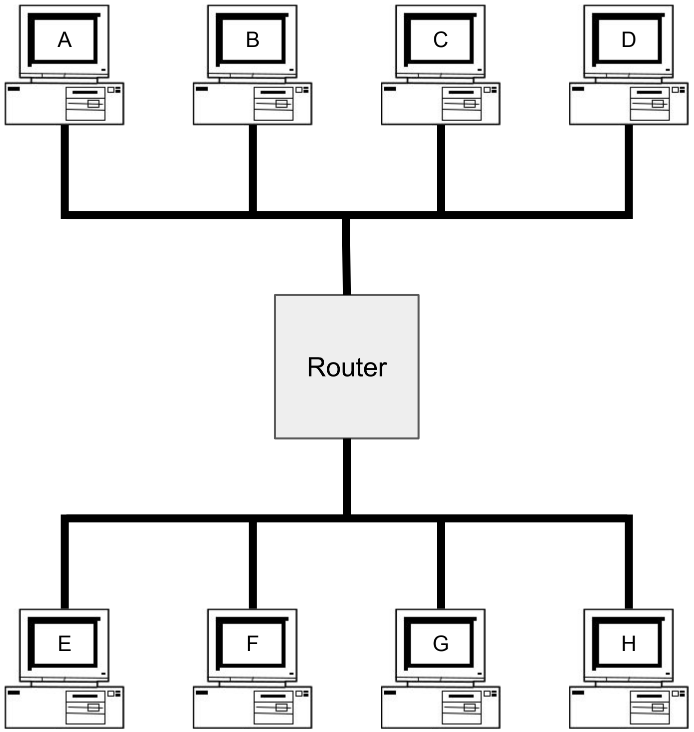
Layer 3: Network Layer

- **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)

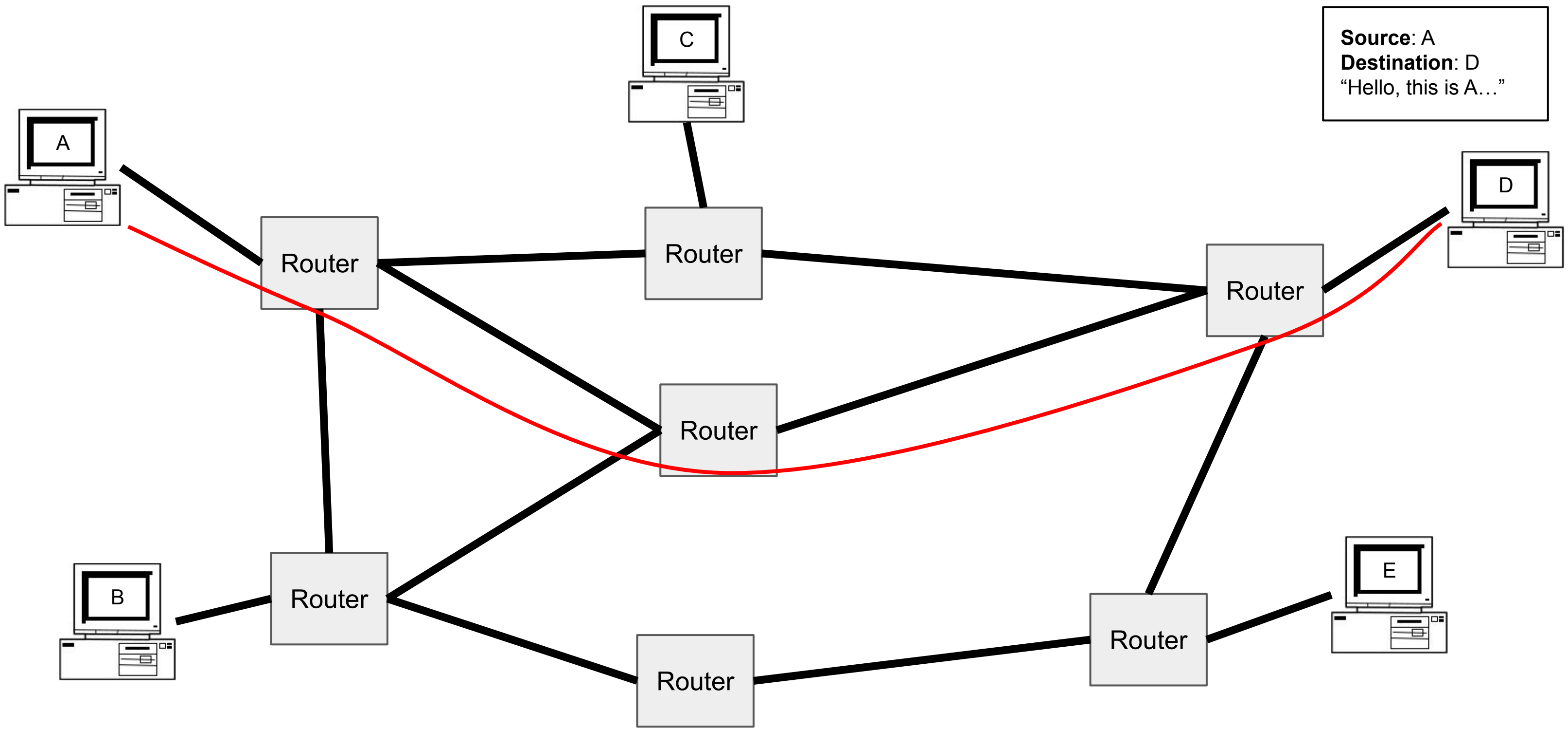


Layer 3: Network Layer

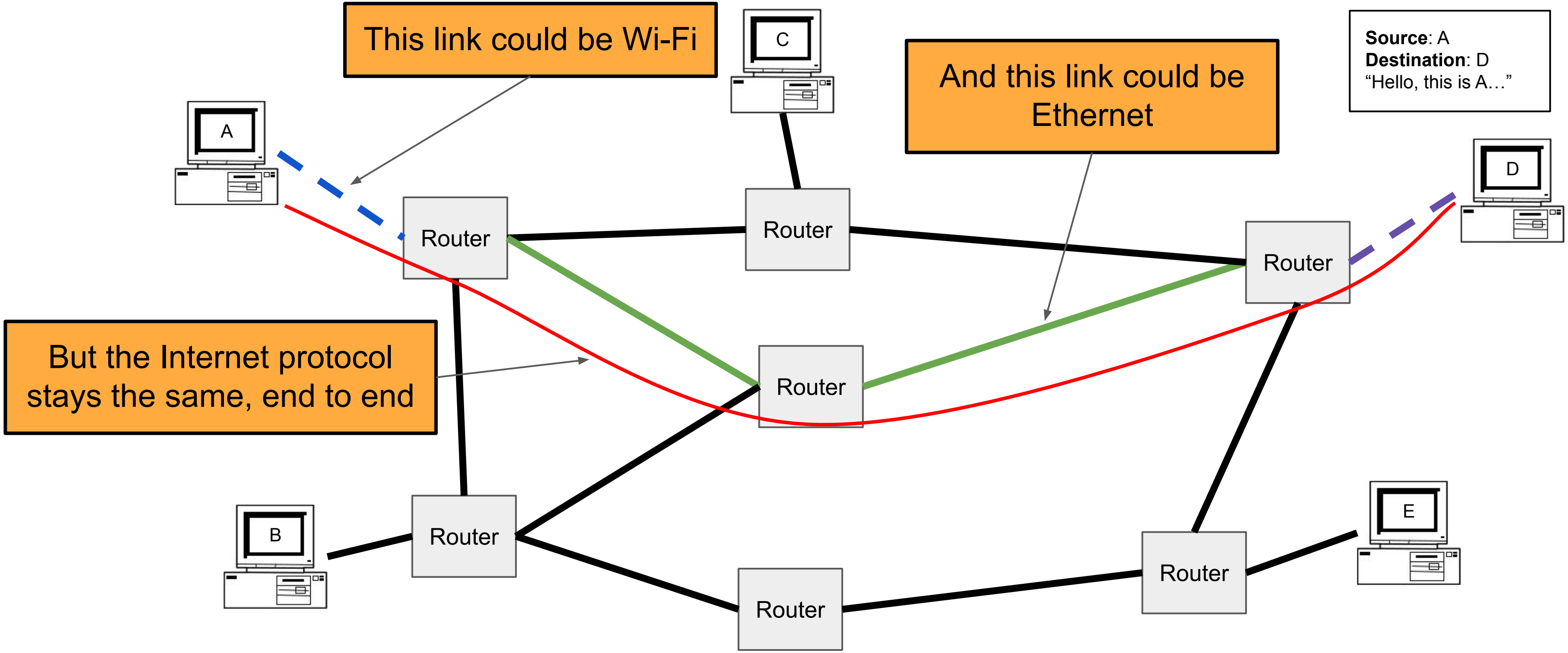
- Recall the ideal layer 2 model: All devices can directly address all other devices
  - This would not scale to the size of the Internet!
- Instead, allow packets to be **routed** across different devices to reach the destination
  - Each hop is allowed to use its own physical and link layers!
- Basic 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



Layer 3: Network Layer



Internet Protocol (IP)

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

IPv4 header

Internet Protocol (IP)

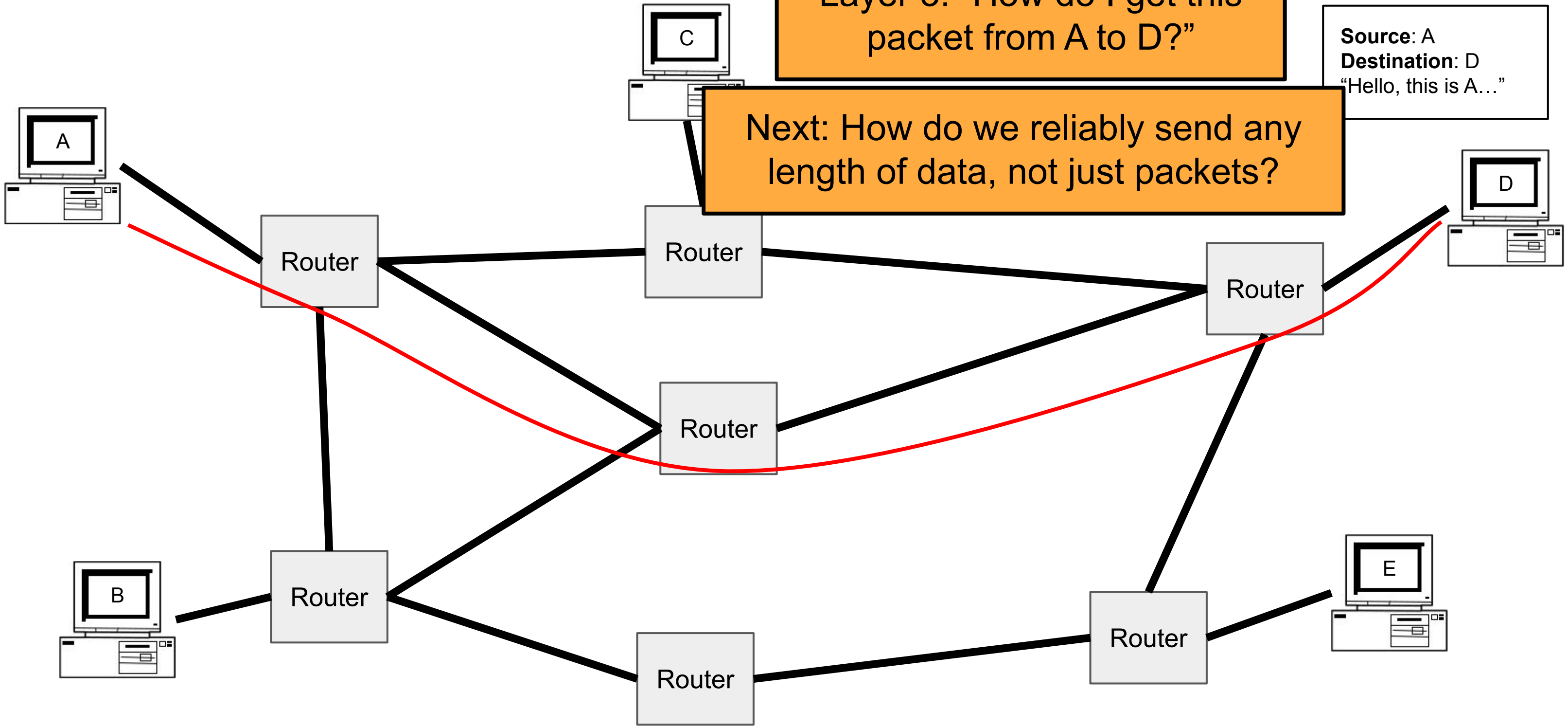
- **Internet Protocol** (**IP**): The universal layer-3 protocol that all devices use to transmit data over the Internet
- **IP address**: An address that identifies a device on the Internet
    - IPv4 is 32 bits, typically written as 4 decimal octets, e.g. **35.163.72.93**
    - IPv6 is 128 bits, typically 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

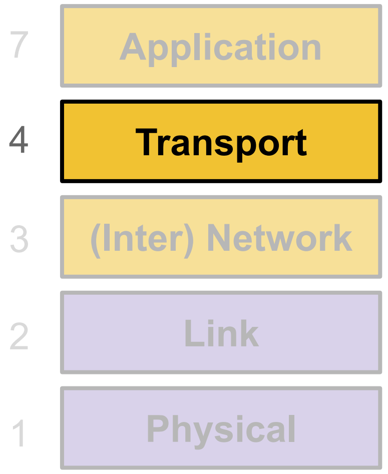
Layer 3: "How do I get this packet from A to D?"



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

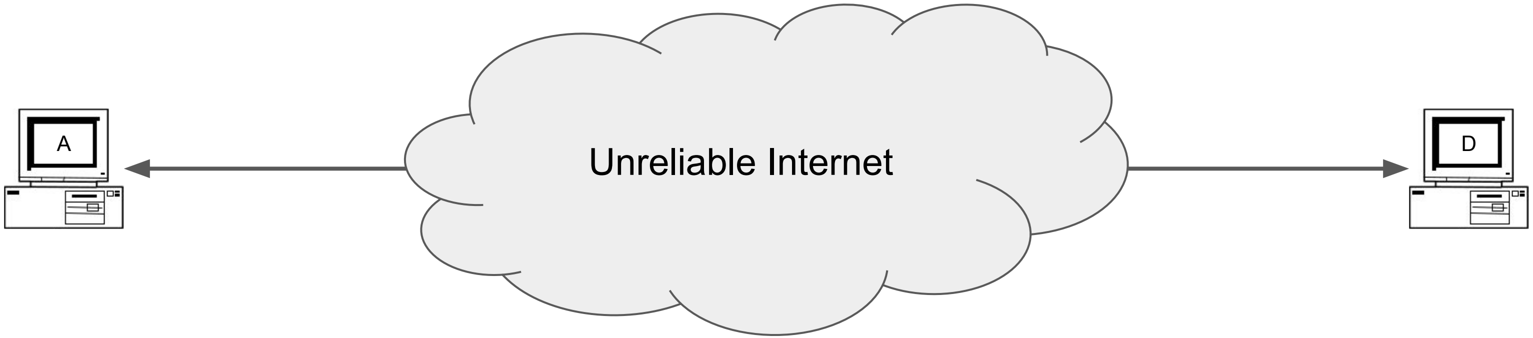
Layer 4: Transport Layer

- **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

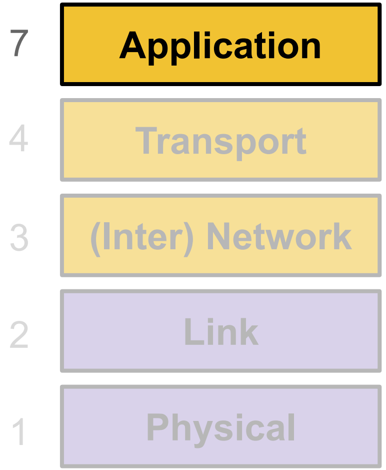
I am now sending an arbitrary length message that will probably be broken into several packets…



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

Layer 7: Application Layer

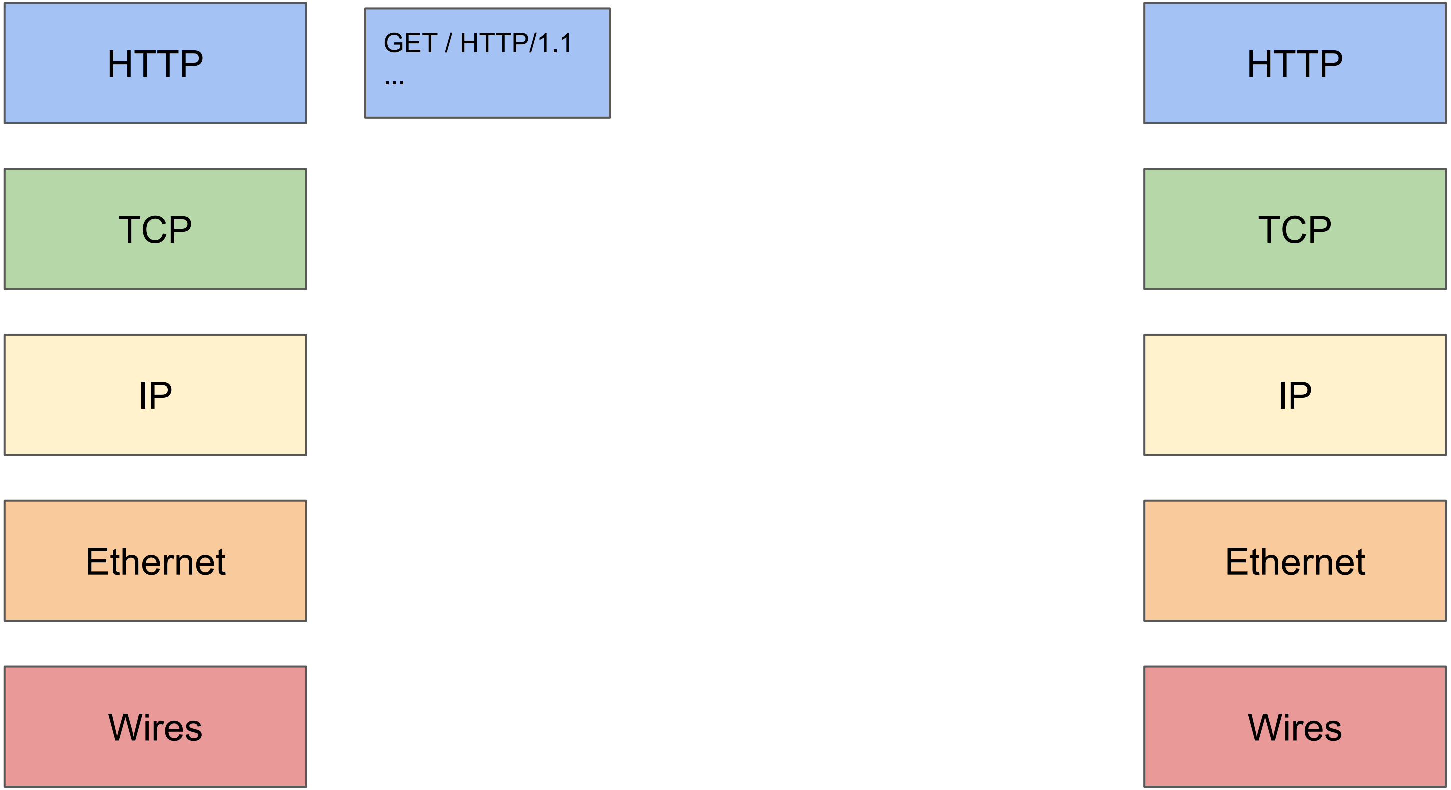
- **Provides**: Applications and services to users!
  - **Relies upon**: Transportation of variable-length data from any point to any other point
- Every online application is Layer 7
  - Web browsing
  - Online video games
  - Messaging services
  - Video calls (Zoom)



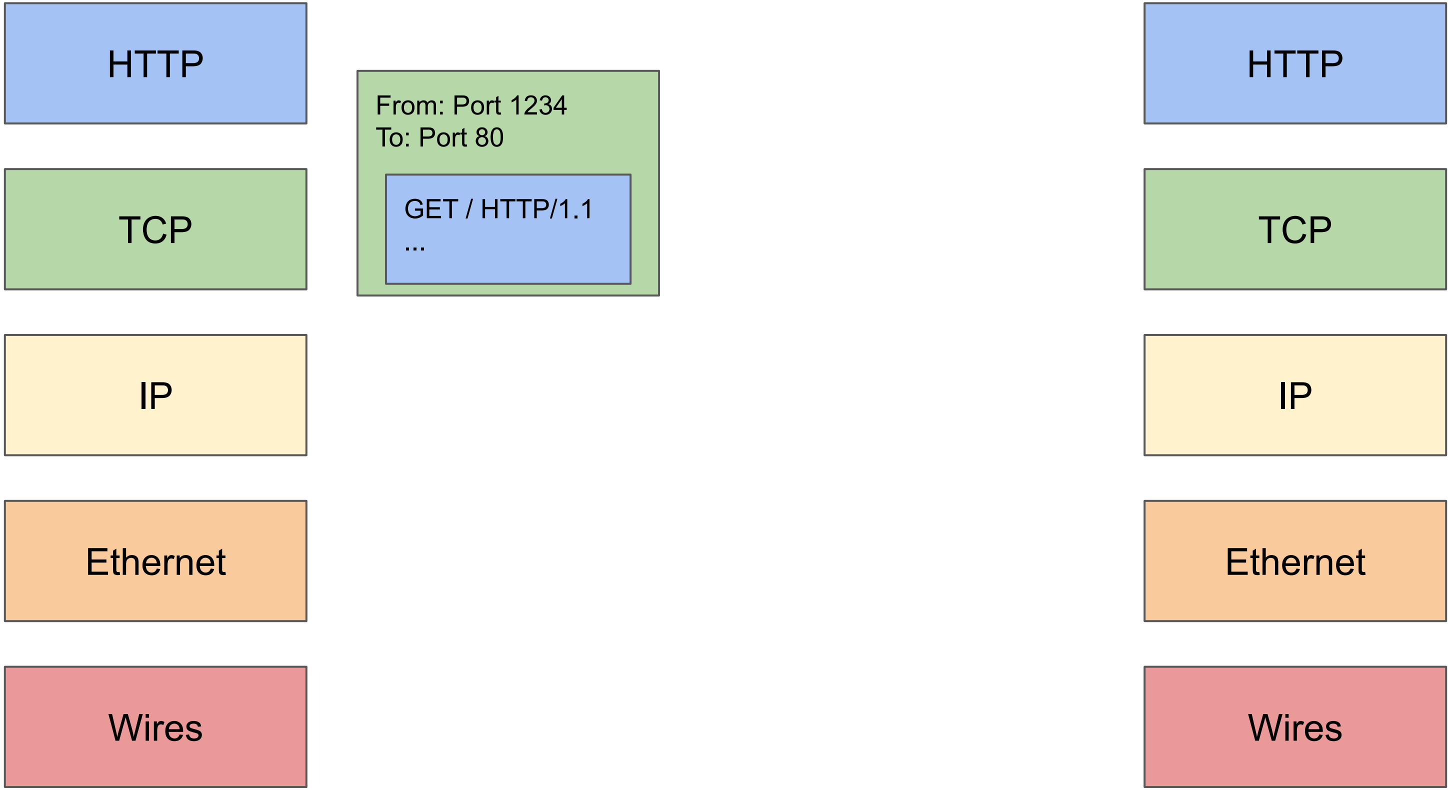
Layers of Abstraction and Headers

- 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

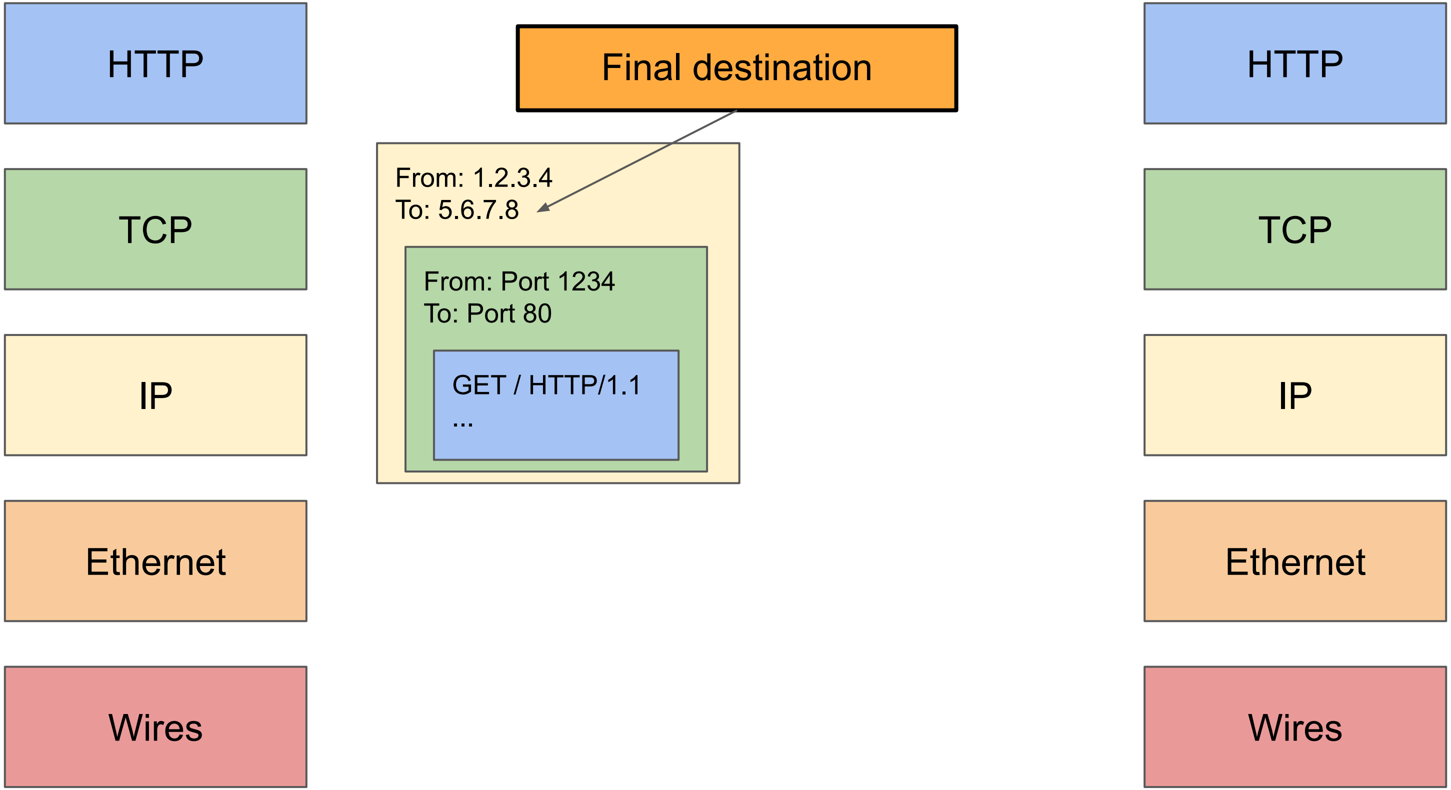
Example: HTTP Request



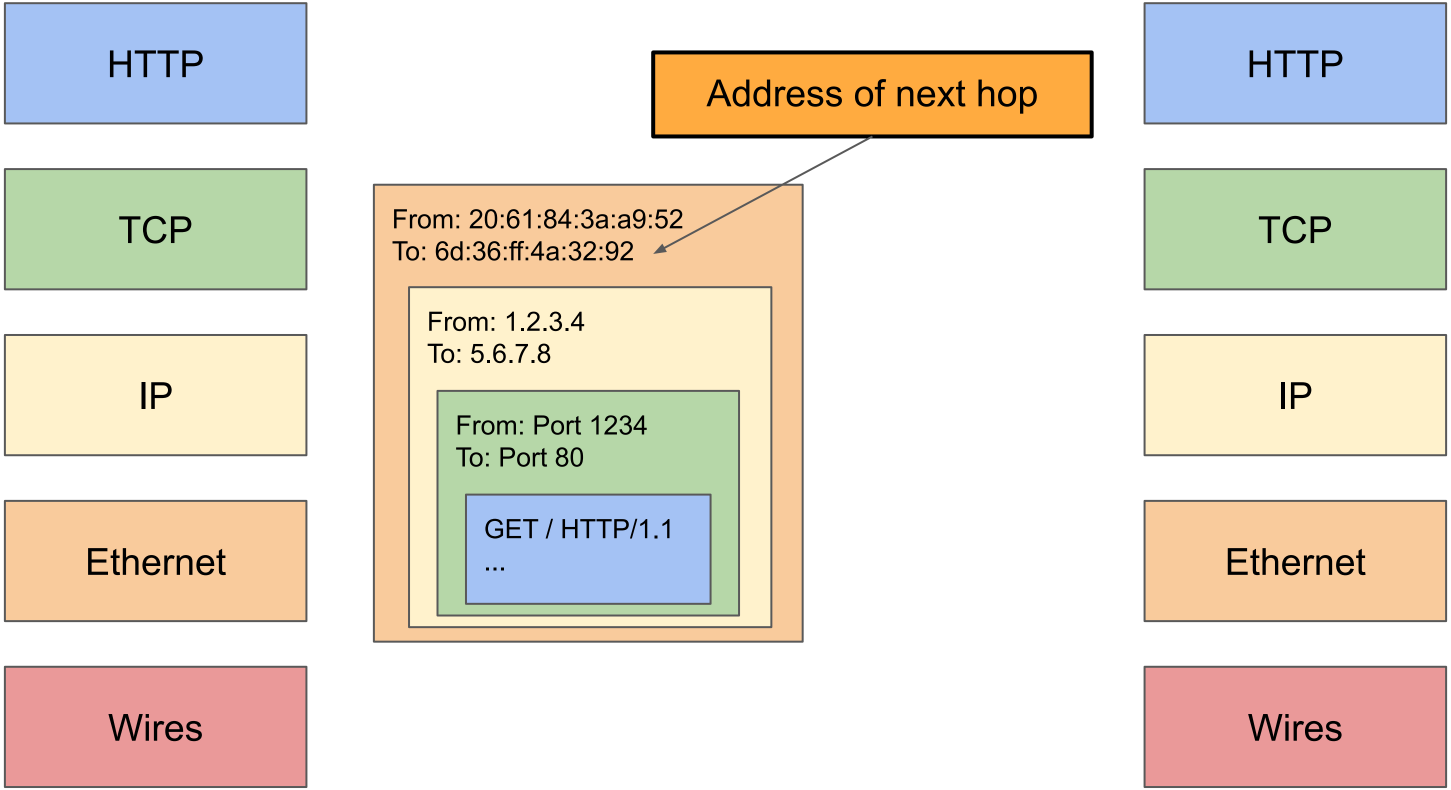
Example: HTTP Request



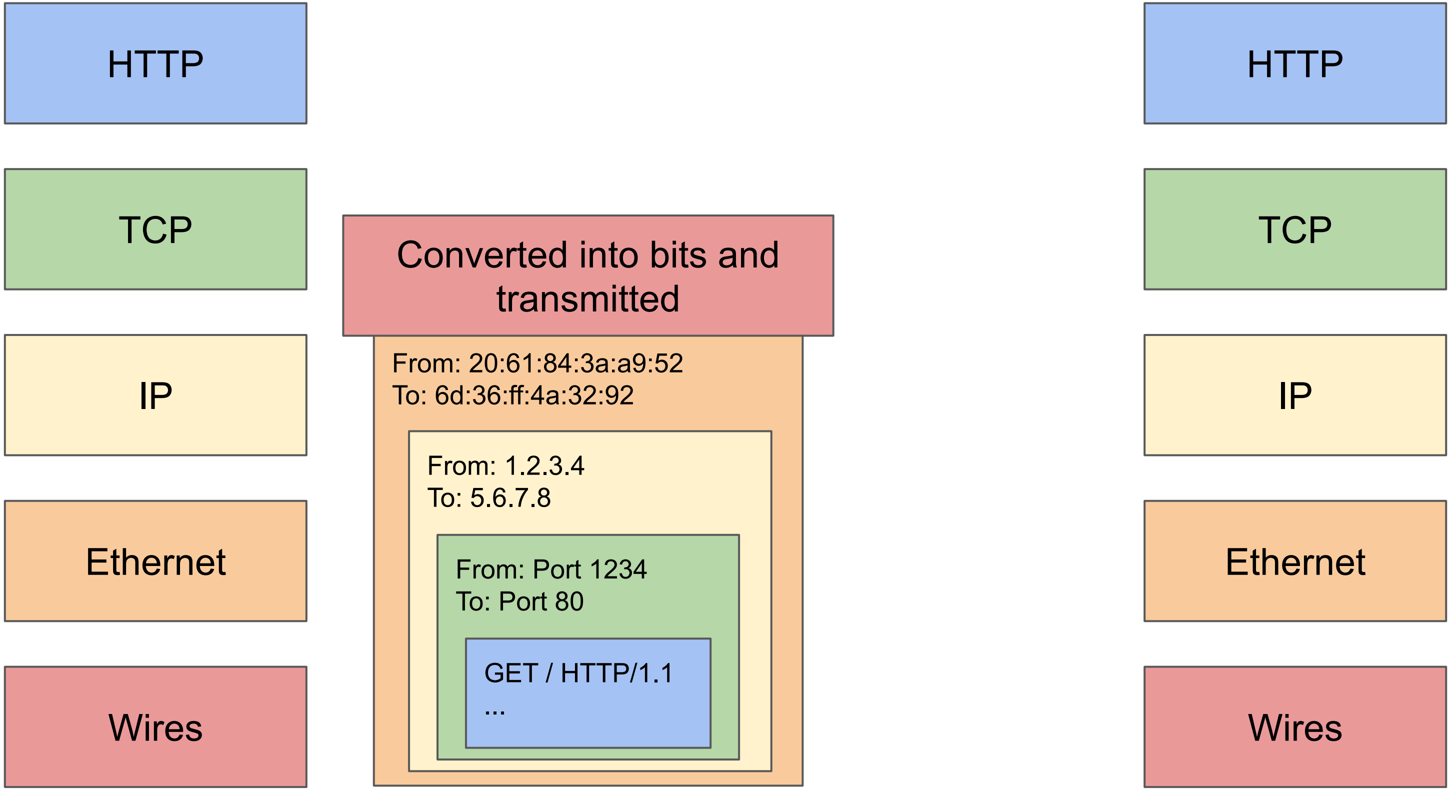
Example: HTTP Request



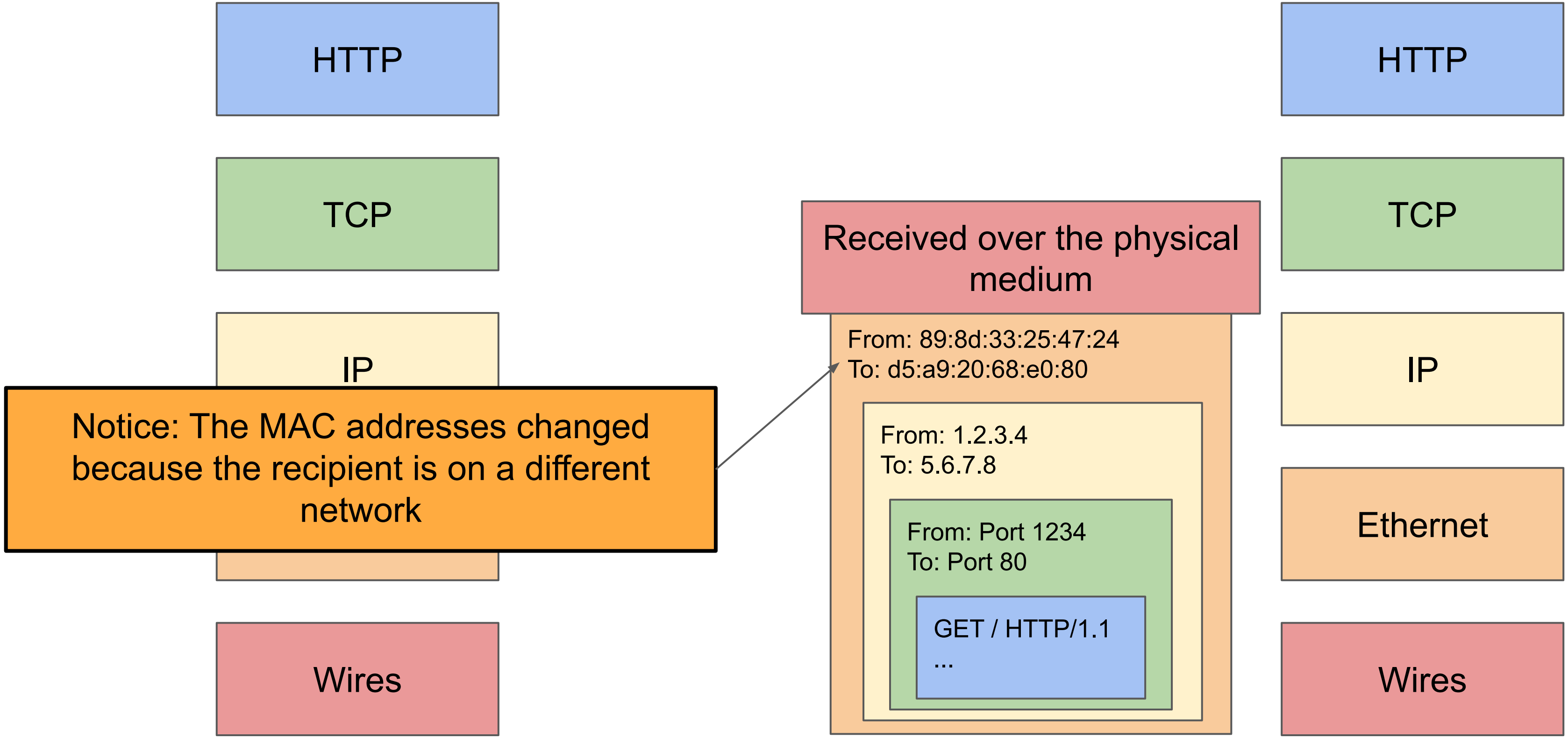
Example: HTTP Request



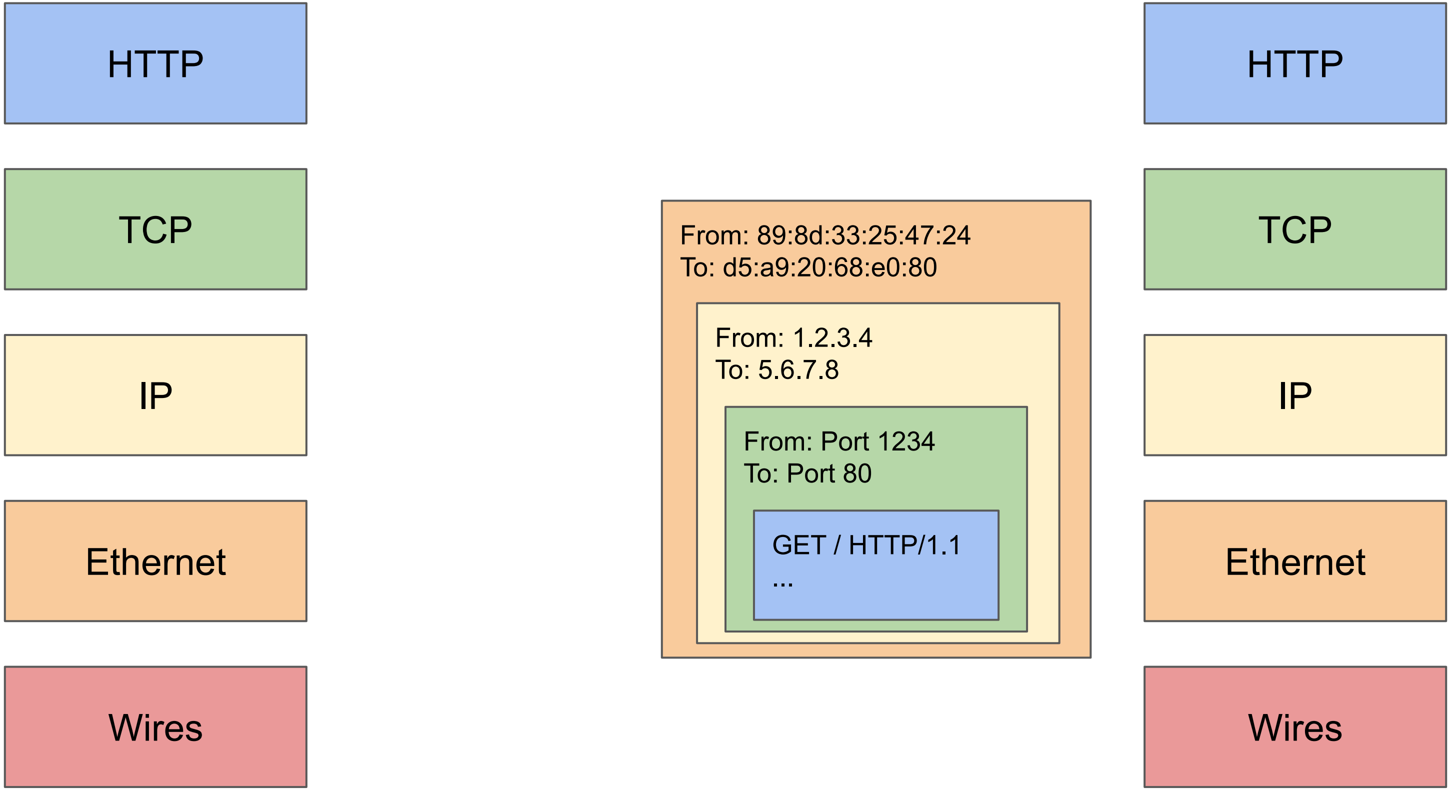
Example: HTTP Request



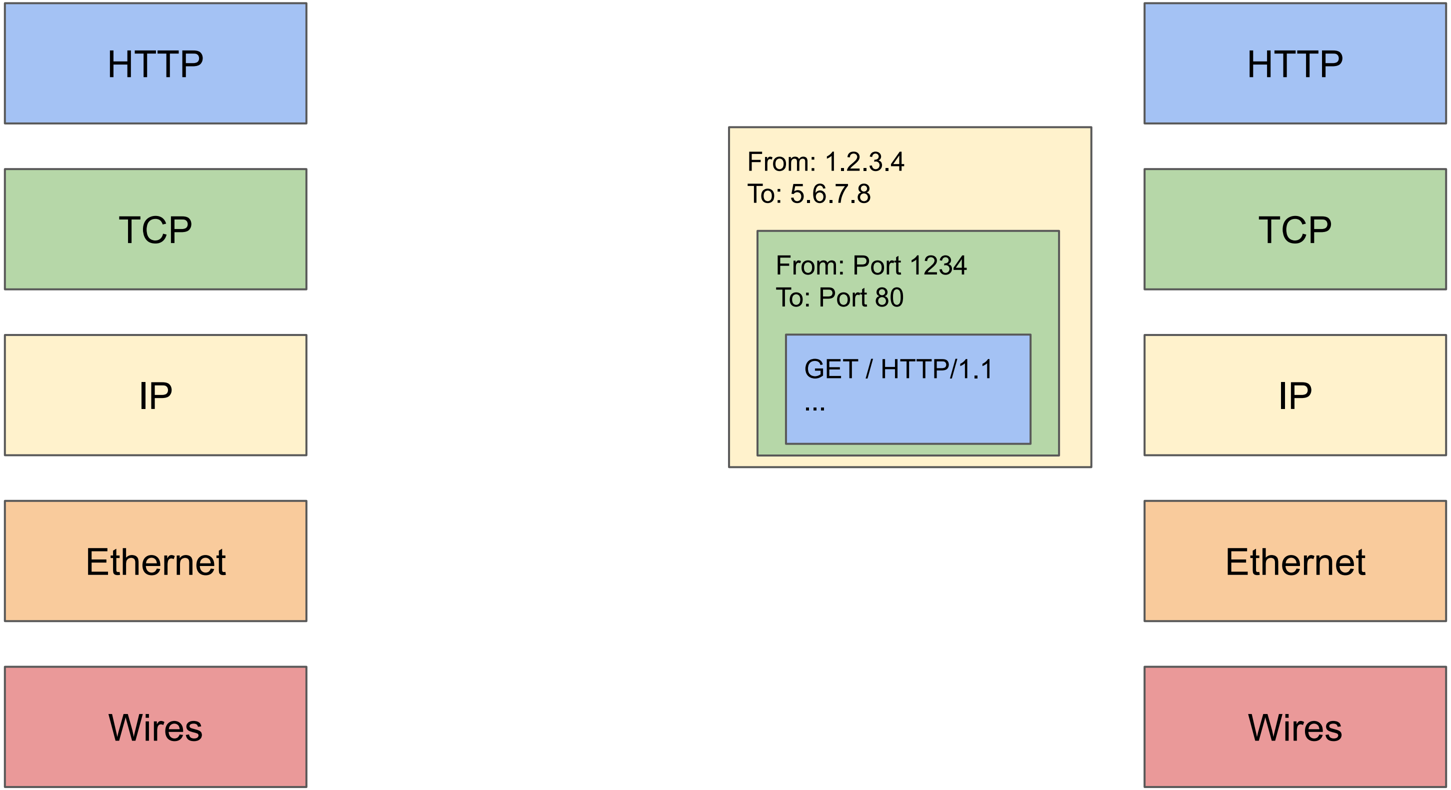
Example: HTTP Request



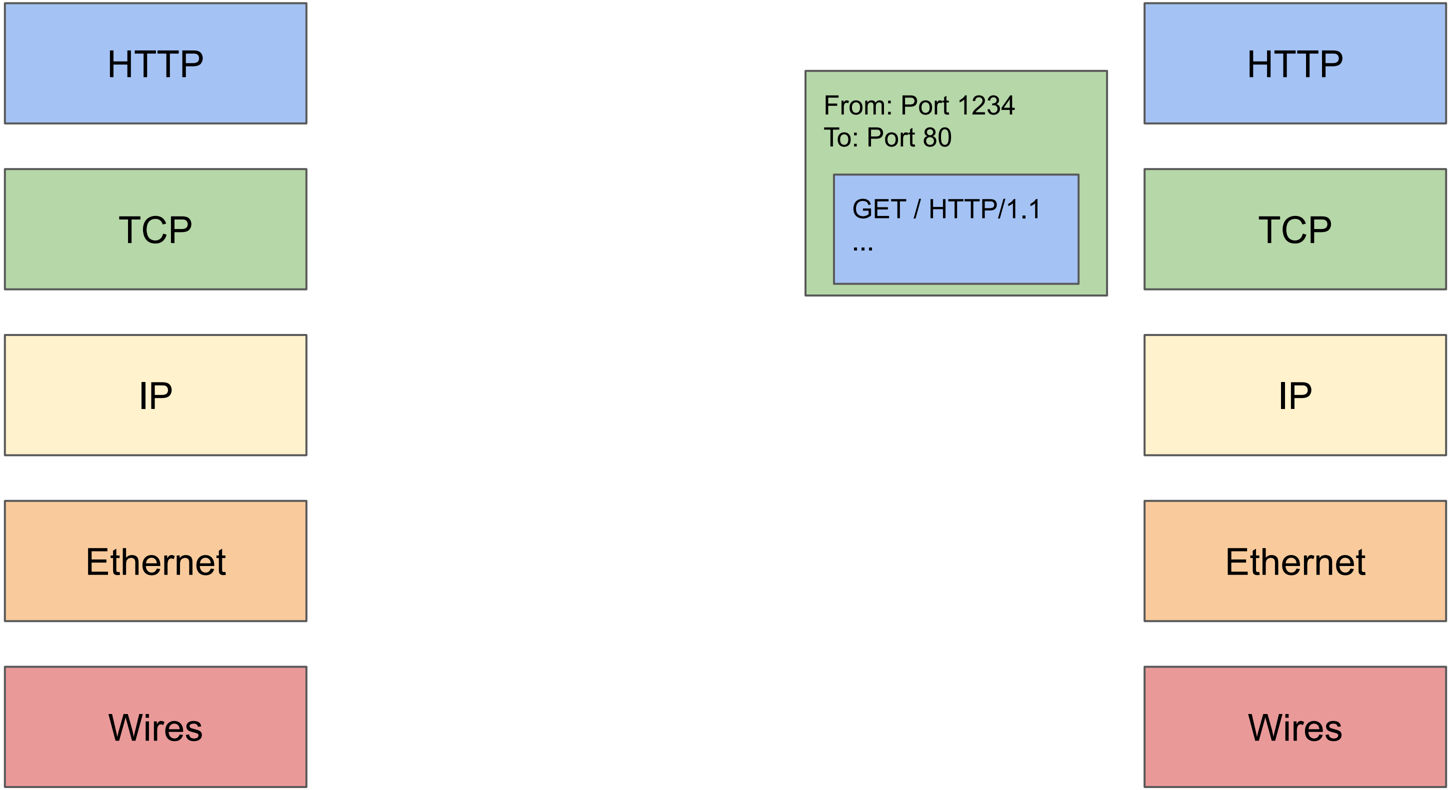
Example: HTTP Request



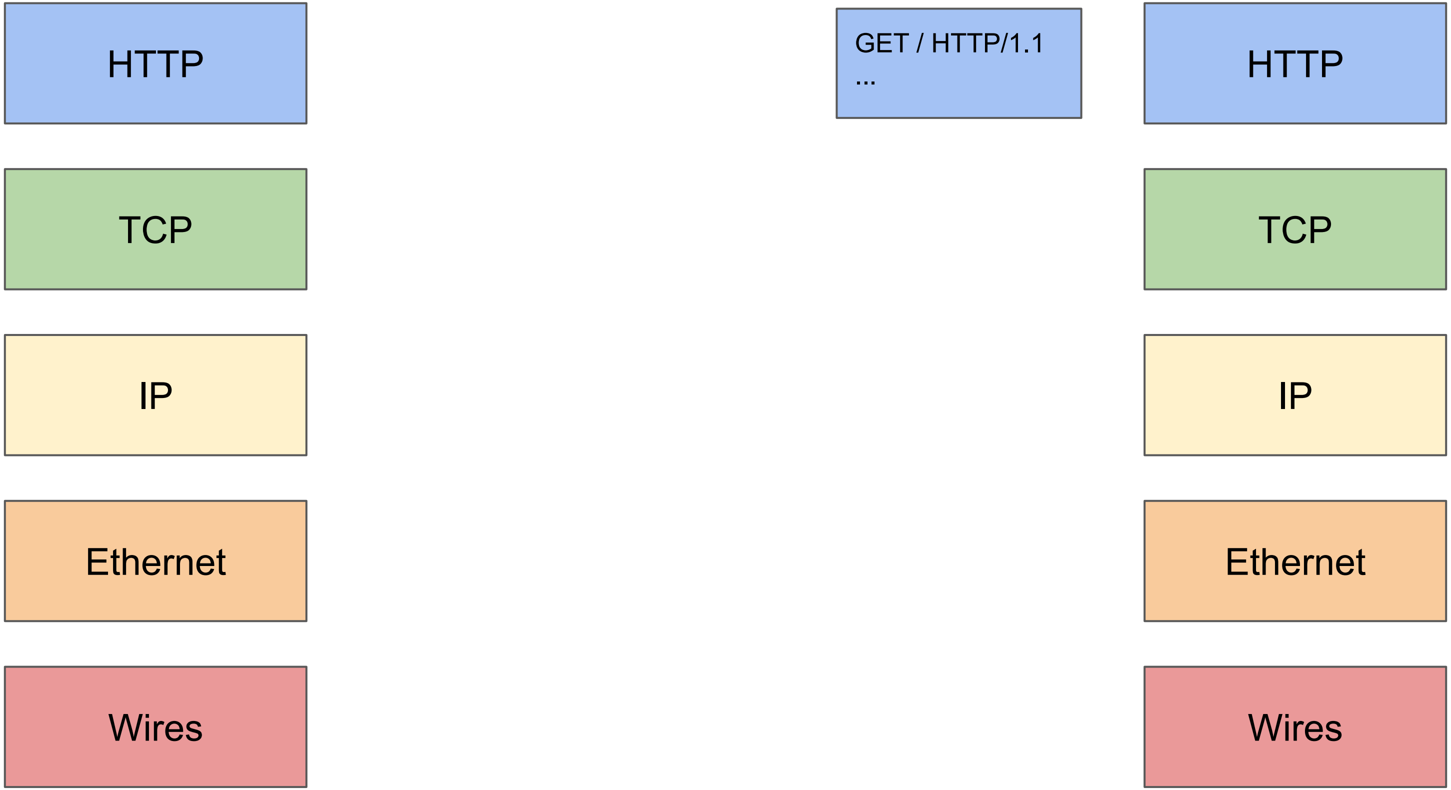
Example: HTTP Request



Example: HTTP Request



Example: HTTP Request



Example: HTTP Request

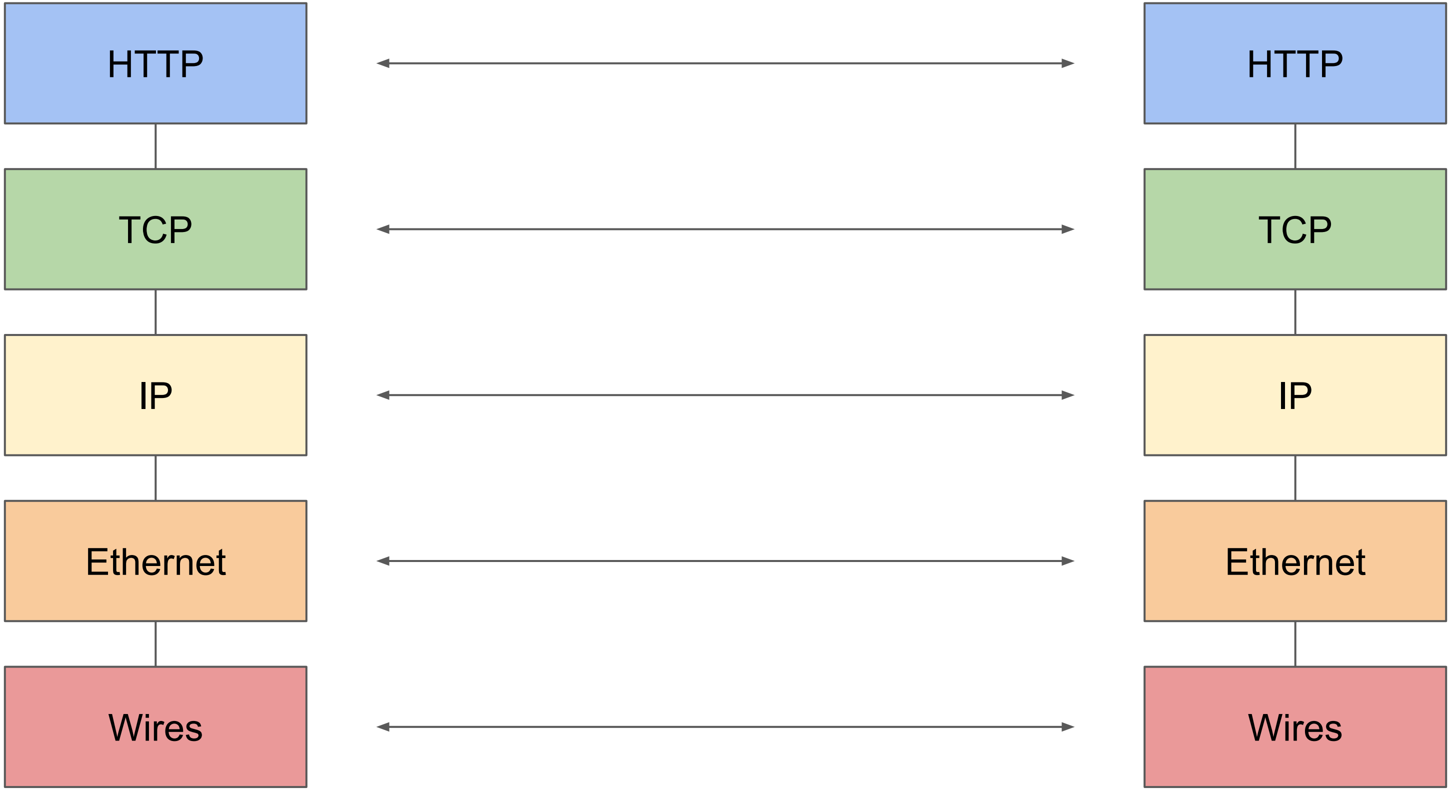
Relies upon: Transport of data

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		

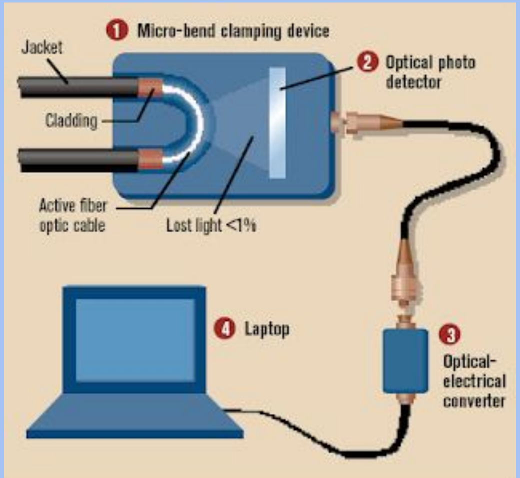
Dynamic Host Configuration Protocol (DHCP)

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



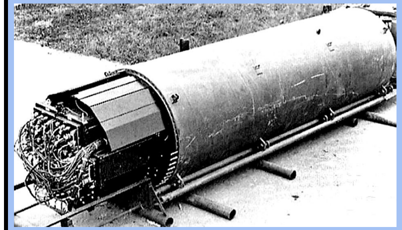
Real-World On-Path Attackers

Military.com

Operation Ivy Bells

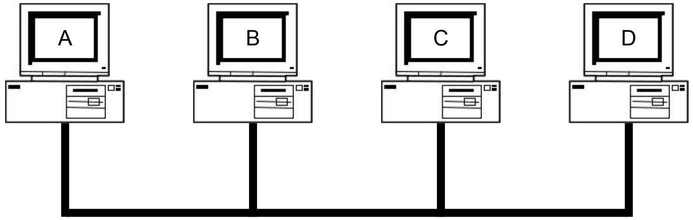
Matthew Carle *February 6, 2017*

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

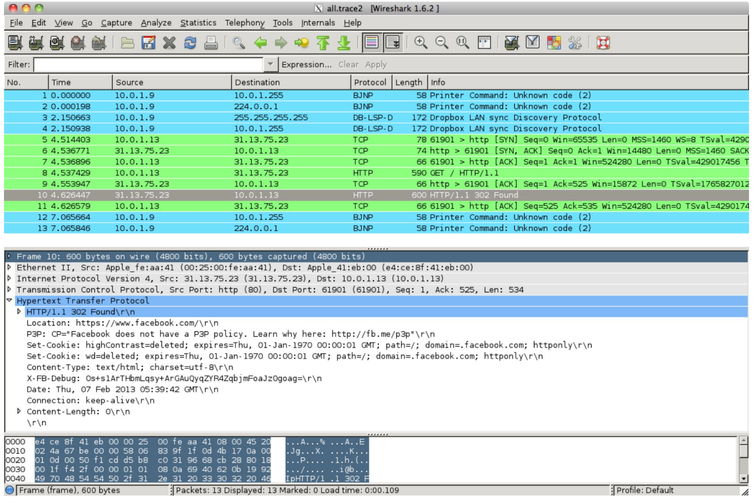
- Layer 2 attack: Read packets sent across the local area network (LAN)
- Recall: A LAN is a network of connected machines
  - Any machine on the LAN can send packets to any other machine on the LAN
- Some LANs use **broadcast technologies**
  - Every packet gets sent to every machine on the LAN
  - 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

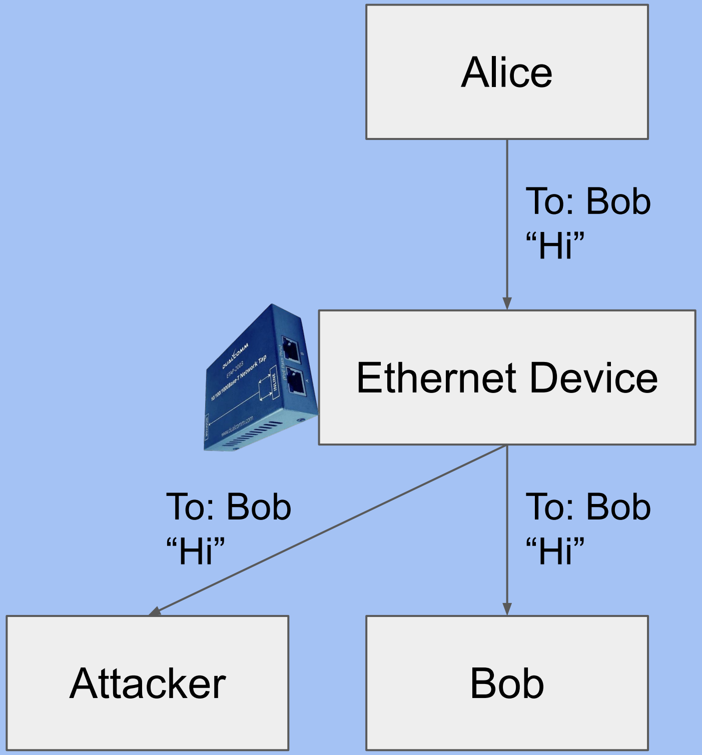
- **`tcpdump`**: A program for reading packets on the local network
  - Uses promiscuous mode to read other machines' packets in broadcast techonlogies
- Wireshark: A graphical user interface (GUI) for analyzing **`tcpdump`** packets

```
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
```



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



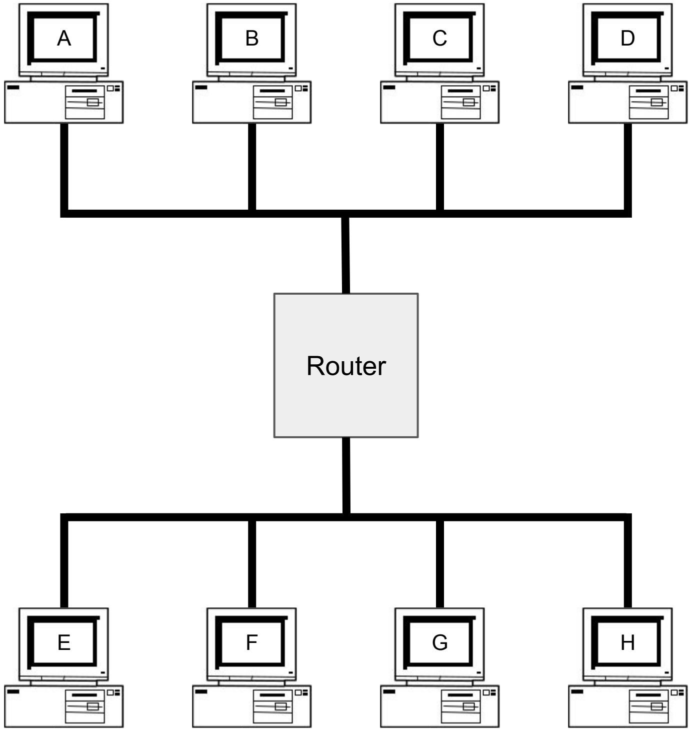
The Law and Sniffing Packets

- 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

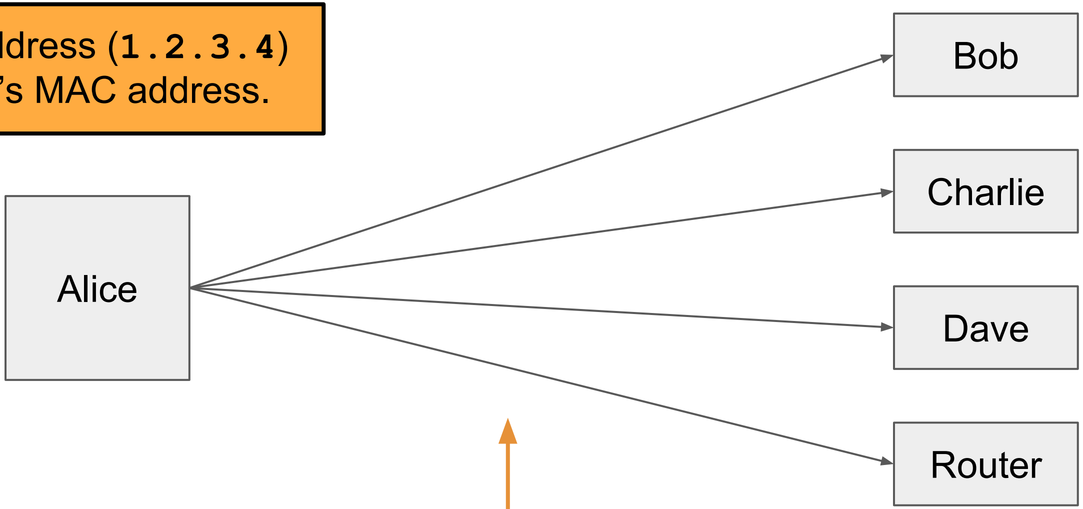
1. Alice checks her cache to see if she already knows the MAC address corresponding to `1.2.3.4`.

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

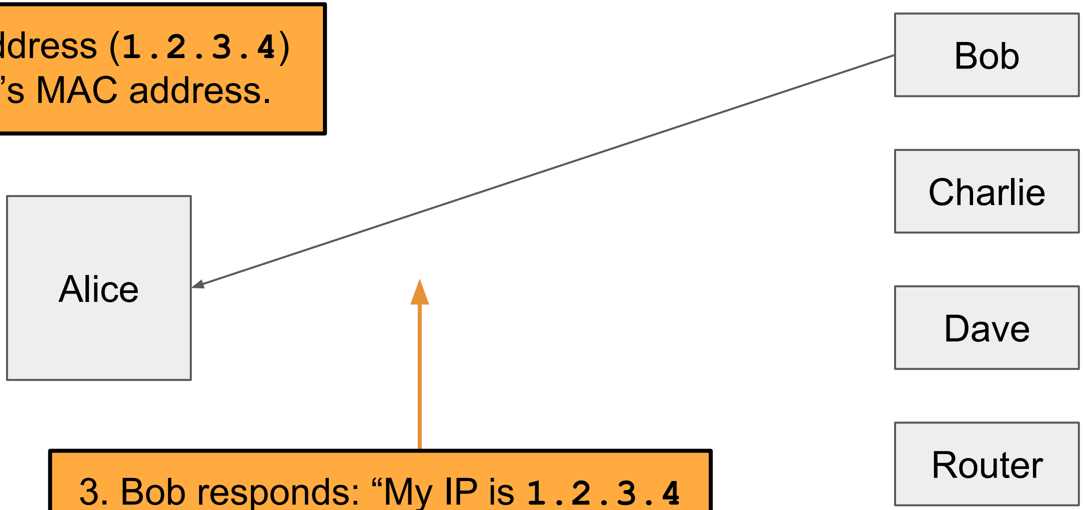


2. Alice asks everyone else on the local network: "What is the MAC address of `1.2.3.4`?"

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`

Alice

Bob

Charlie

Dave

Router

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

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

Alice

Bob

Charlie

Mallory

Router

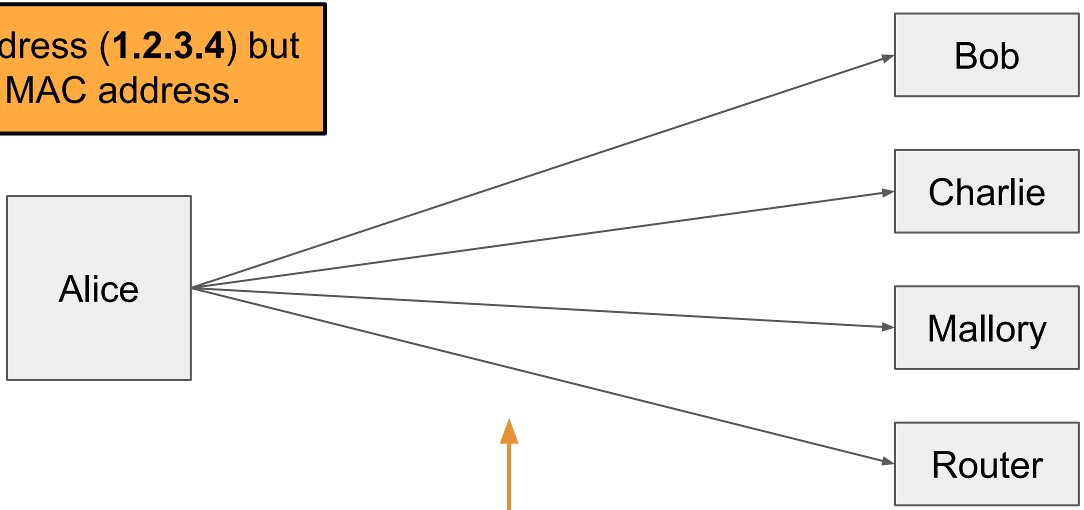
1. Alice checks her cache to see if she already knows the MAC address corresponding to **1.2.3.4**.

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

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

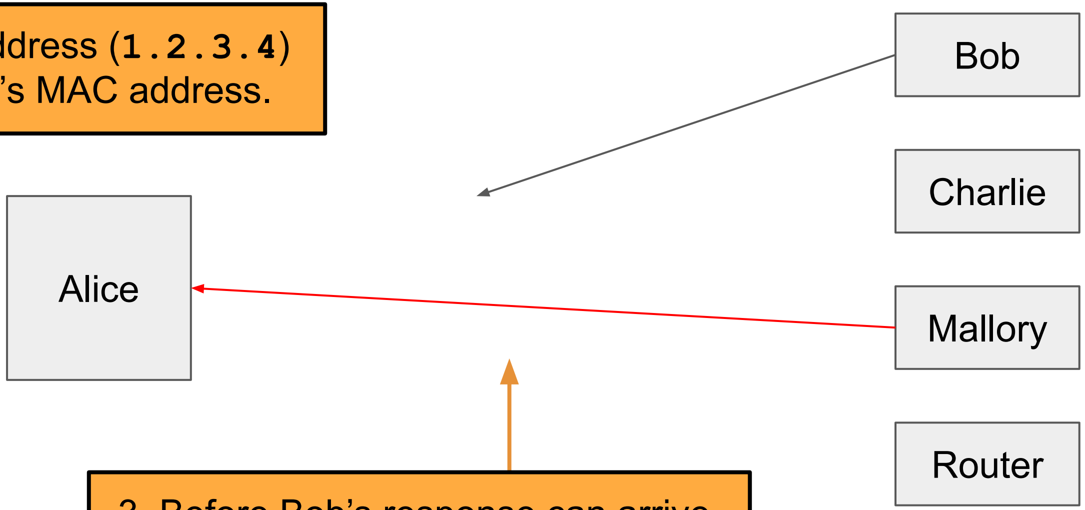


2. Alice asks everyone else on the local network: "What is the MAC address of **1.2.3.4**?"

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**

Alice

Bob

Charlie

Mallory

Router

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

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

DHCP: Initial Network Configuration

- To connect to a network, a user needs:
  - An IP address so that other people can contact the user
  - The IP address of the DNS server (we'll see this soon)
  - The IP address of the router (gateway) so that the user can contact machines outside of the LAN
- The first time a user connects, they don't have this information yet
  - The user also doesn't know who to ask for this information
- **DHCP** gives the user a configuration when they first join the network

Steps of the DHCP Handshake

1. **Client Discover:** The client *broadcasts* a request for a configuration
2. **DHCP Offer**: Any DHCP server can respond with a configuration offer
   - Usually only one DHCP server responds
   - The offer includes an IP address for the client, the DNS server's IP address, and the (gateway) router's IP address
   - The offer also has an expiration time (how long the user can use this configuration)
3. **Client Request**: The client broadcasts which configuration it has chosen
   - If multiple DHCP servers made offers, the ones that were not chosen discard their offer
   - The chosen DHCP server gives the offer to the client
4. **DHCP Acknowledgement**: The chosen server confirms that its configuration has been given to the client

Dynamic Host Configuration Protocol (DHCP)

Alice's configuration	
My IP	???
DNS Server	???
Gateway	???

Alice

Bob

DHCP Server 1

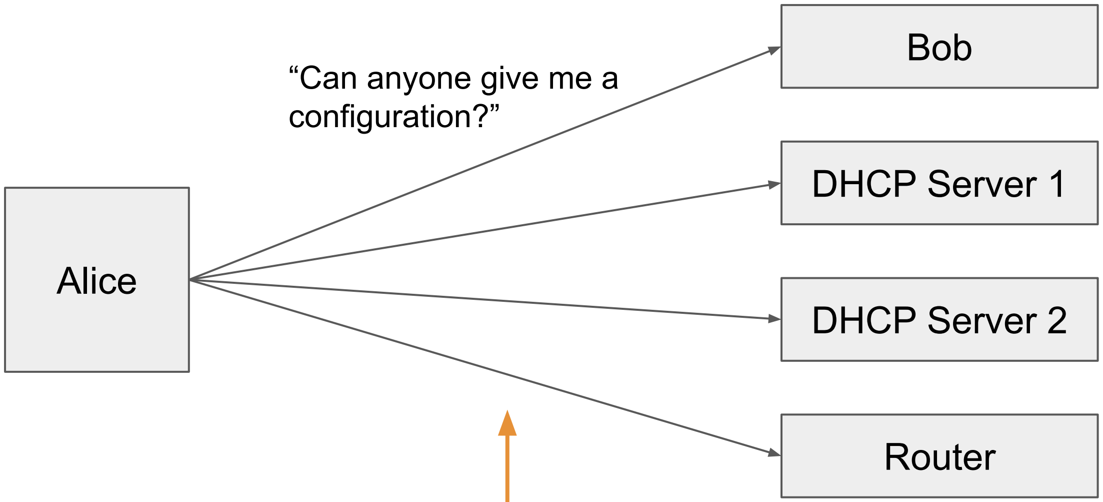
DHCP Server 2

Router

Alice wants to connect to the network, but she's missing a configuration.

Dynamic Host Configuration Protocol (DHCP)

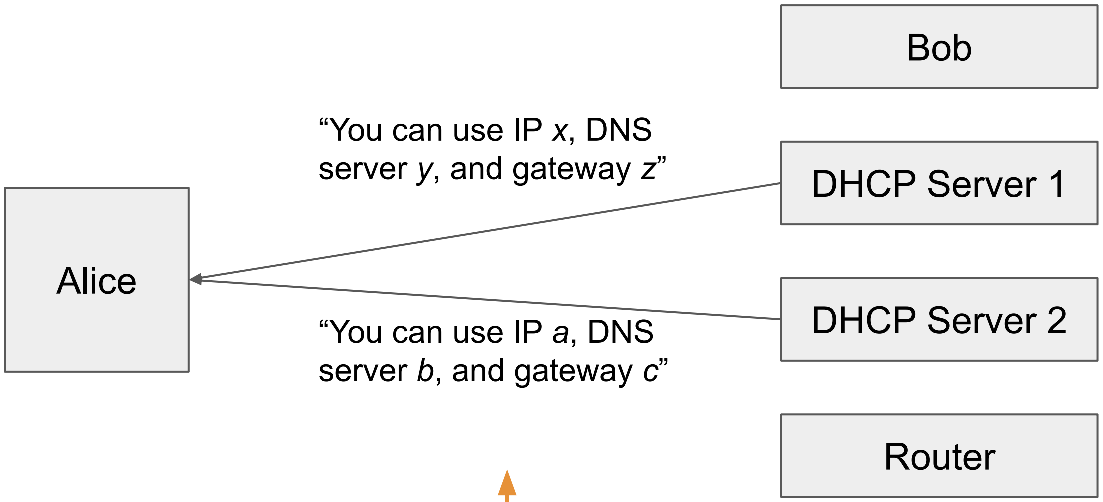
Alice's configuration	
My IP	???
DNS Server	???
Gateway	???



1. **Client Discover**: Alice broadcasts a request for a configuration.

Dynamic Host Configuration Protocol (DHCP)

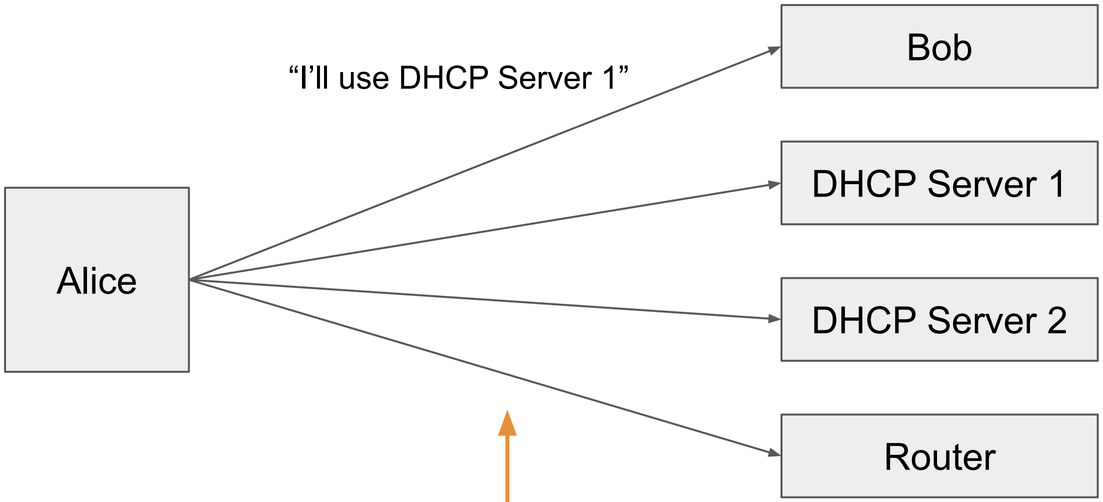
Alice's configuration	
My IP	???
DNS Server	???
Gateway	???



2. **DHCP Offer**: Any DHCP server can reply with an offer for Alice.

Dynamic Host Configuration Protocol (DHCP)

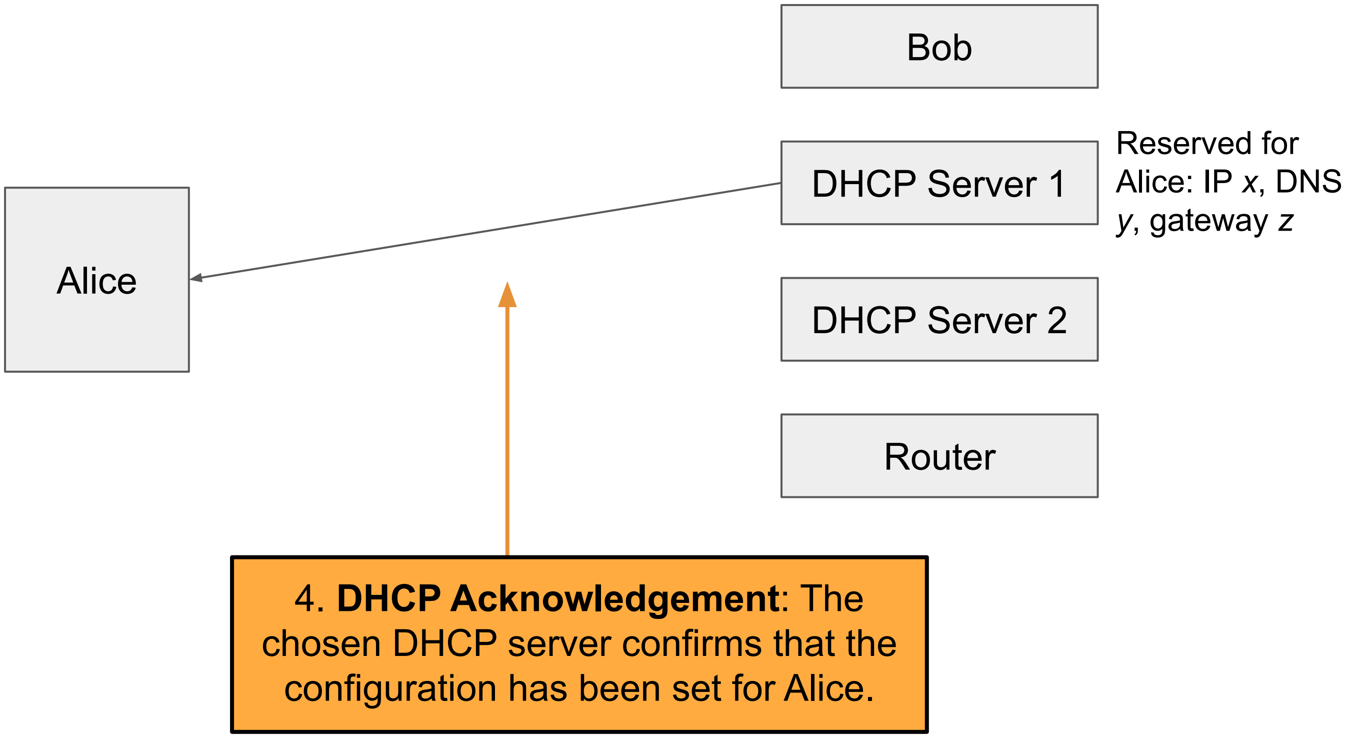
Alice's configuration	
My IP	???
DNS Server	???
Gateway	???



3. **Client Request**: Alice broadcasts which configuration she has chosen.

Dynamic Host Configuration Protocol (DHCP)

Alice's configuration	
My IP	*x*
DNS Server	*y*
Gateway	*z*



DHCP Attacks

- Alice has no way of verifying the DHCP response
  - Spoofing: Any attacker on the network can claim to have a configuration
- Alice usually expects only one DHCP server to respond, so she will accept the first response
  - **Race condition**: As long as the attacker responds faster, Alice will accept the attacker’s response
- DHCP attacks require Mallory to be in the same LAN as Alice
- DHCP attacks let Mallory become a man-in-the-middle (MITM) attacker
  - Mallory claims the gateway router’s address is Mallory’s address
    - When Alice sends a message to the rest of the Internet, she actually sends it to Mallory
    - Mallory can modify the message before sending it to its destination
  - Mallory can also claim the DNS server’s address is Mallory’s address

ARP and DHCP

- The attacks on ARP and DHCP are very similar
  - **Broadcast**: The attacker can see the request because it is shouted to everyone
  - **Spoofing**: The attacker claims to have an answer
  - **Race condition**: The requester accepts the first response. As long as the attacker's response arrives first, it is accepted
- Main vulnerabilities
  - **Broadcast protocols**: Requests are sent to everyone on the LAN, so the attacker can see every request
  - **No trust anchor**: There is no way to verify that responses are legitimate

DHCP Defenses

- DHCP is hard to defend against
  - No root of trust: When we first connect, there's nobody we can trust
- Enterprise-class switches can offer protection
  - Similar to the ARP-spoofing protection
- But for the most part, we rely on defenses provided in higher layers
  - We'll cover this soon!

Summary: ARP and DHCP

- Classes of attackers:
  - Off-path: Can’t see, modify, or drop packets
  - On-path: Can see packets, but can’t modify or drop packets
  - MITM: Can see, modify, and drop packets
- ARP: A protocol to translate local IP addresses to MAC addresses
  - Ask everyone on the network, “Who has the IP 1.2.3.4?”
  - Attack: The attacker can respond instead of the true device with 1.2.3.4, and packets will get routed to the attacker!
  - Defense: Switches
  - Defense: Rely on higher layers
- DHCP: A protocol for a new client to receive a network configuration
  - Ask everyone on the network, “What is the network configuration to use?”
  - Attack: The attacker can respond with a malicious configuration
  - Defense: Rely on higher layers

Next: Wireless Local Networks

- WPA: Communicate securely in a wireless local network
  - 4-way handshake
  - WPA-PSK
  - WPA-Enterprise
  - WPA3/Dragonfly

Wireless Local Networks

Wi-Fi

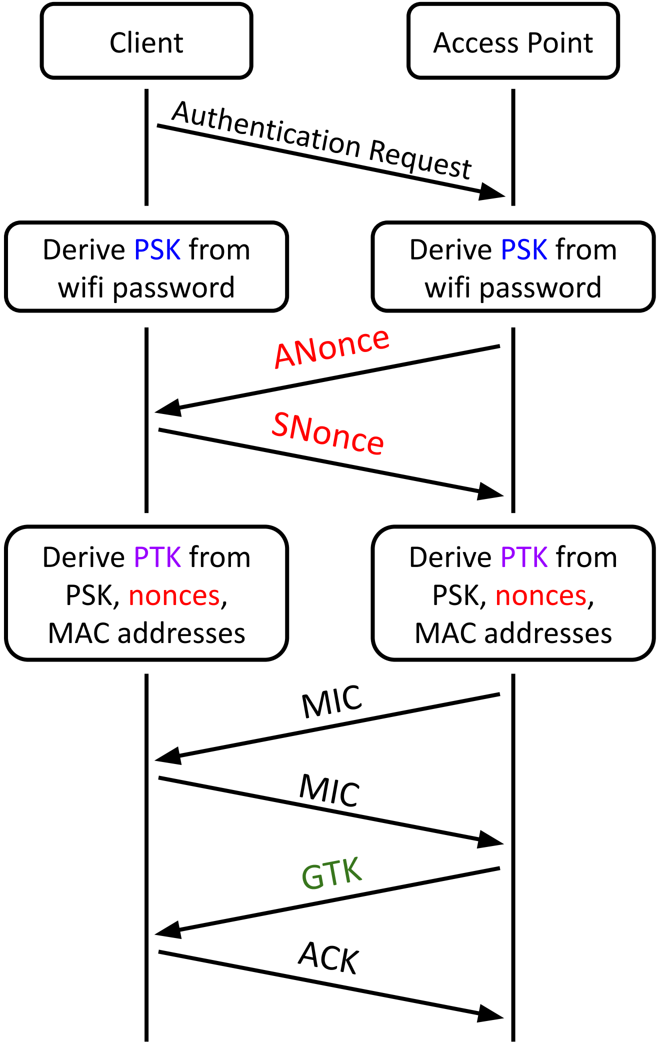
- **Wi-Fi**: A layer 2 protocol that wirelessly connects machines in a LAN
  - Alternative is Ethernet, which uses wires to connect machines in a LAN
- Parts of a Wi-Fi network
  - **Access point**: A machine that will help you connect to the network
  - **SSID** (service set identifier): The name of the Wi-Fi network
  - **Password**: Optionally, a password to secure Wi-Fi communications

WPA2

- **Wi-Fi Protected Access 2** (**WPA2**): A protocol for securing Wi-Fi network communications with cryptography
- Design goals
  - Everyone with the Wi-Fi password can join the network
  - Messages sent over the network are encrypted with keys
  - An attacker who does not know the Wi-Fi network cannot learn the keys
- Most common key exchange: **WPA-Pre Shared Keys** (**WPA-PSK**)

WPA Handshake: Conceptual Breakdown

1. The client sends an authentication request to the access point
2. Both use the password to derive the *PSK* (pre-shared key)
3. Both exchange random nonces
4. Both use the *PSK*, nonces, and MAC addresses to derive the *PTK* (pairwise transport keys)
5. Both exchange MICs (these are MACs from the crypto unit) to ensure no one has tampered with the nonces, and that the PTK was correctly derived
6. The access point encrypts and sends the *GTK* (group temporal key) to the client, used for broadcasts that anyone can decrypt
7. The client acknowledges receiving the GTK



MICs are Cryptographic MACs

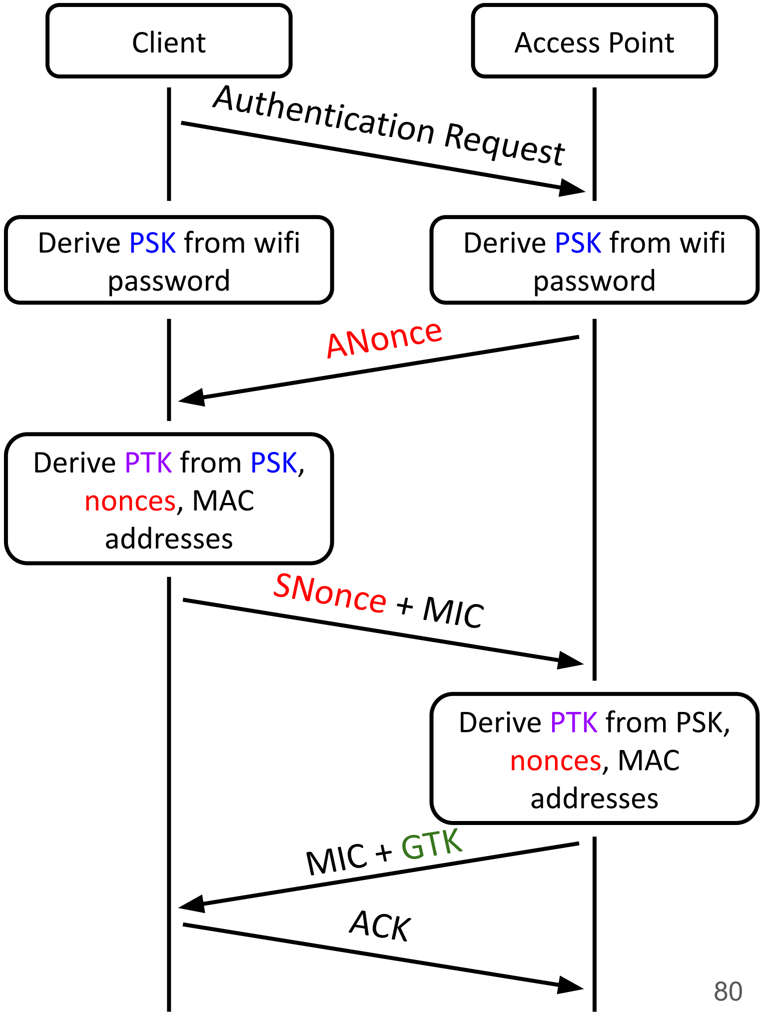
- In cryptography:
  - **MAC == Message Authentication Code**: A cryptographic primitive with a shared secret key that ensures integrity
- In networking:
  - **MAC == Media Access Controller**: A unique identifier for the device on the network
    - In Ethernet, it is 48 bits, like ca:fe:f0:0d:de:ad
  - **MIC == Message Integrity Code**: This is a cryptographic MAC in networking-speak
- The problem of TLAs (Three Letter Acronyms): Namespace collisions

WPA Handshake

- Both sides derive secret keys for communication
  - Wi-Fi password → *PSK*
  - *PSK* + nonces + MAC addresses → *PTK*
  - The *PTK* is used to encrypt and authenticate all future communication
  - Note: The PTK is different for every user, because of the nonces
- The access point encrypts and sends the *GTK* to the client
  - The GTK is used for messages broadcast to the entire network
  - Everyone on the network uses the same GTK
- The optimized version of the handshake decreases the number of messages sent back and forth

WPA Handshake: Optimized 4-Way Handshake

1. The client sends an authentication request to the access point
2. Both use the password to derive the *PSK* (pre-shared key)
3. The AP sends *ANonce* to the client
4. The client generates *SNonce*, uses the *PSK*, nonces, and MAC addresses to derive the *PTK* (pairwise transport keys)
5. The client sends *SNonce* and its MIC to the AP
6. The AP uses the *PSK*, nonces, and MAC addresses to derive the *PTK* (pairwise transport keys)
7. The AP sends its MIC and *GTK* to the client
8. The client acknowledges receiving the GTK



WPA-PSK Attacks

- **Rogue AP**: Pretend to be an AP, and offer your own *ANonce* to the client
    - If you know the password/PSK, you can complete the 4-way handshake with the client and become a MITM!
- **Offline brute-force attack**: People tend to choose bad passwords, and you have enough information to know if you guessed the password correctly
    - Nonces are sent unencrypted, and client and AP MAC addresses are public
    - Eavesdropper guesses a password and derives:
        - Wi-Fi password → *PSK*
        - *PSK* + nonces + MAC addresses → *PTK*
        - Eavesdropper checks that the MIC from the guess matches the MIC that was sent
- **No forward secrecy**: An eavesdropper who records the values of *ANonce* and *SNonce* can derive the key if they later learn the password or *PSK*
    - Compare to Diffie-Hellman: An eavesdropper can't learn the key even if the record g^a and g^b and later compromise Alice's computer

WPA-PSK: Conducting a Brute-Force Attack

- As an eavesdropper (with the handshake):
  - All the information is in the 4-way handshake, if you recorded it
- As a rogue AP:
  - Pretend to be an AP to connect to the wireless network
  - When receiving an authentication request, use a random *ANonce*
  - The client sends the *SNonce* + *MIC* needed to conduct the brute-force attack
- As an eavesdropper (without the handshake):
  - If you didn't record the handshake, you don't have the MIC you need to brute-force…
    - … But you can force the client to disconnect and reconnect!
  - **Disassociatiation attack**: Spoofing a Wi-Fi frame that says "something is wrong; try reconnecting"
    - The frame doesn't contain any integrity since it's meant to be sent the AP has crypto problems
  - When the client reconnects, you can record the new handshake and use it to brute-force

WPA-Enterprise

- Core issue: Every client starts with the same *PSK* to derive the *PTK*
    - Fix: Have each user use their own username and password, instead
        - This is the model that AirBears2 and eduroam use!
- Instead of using a PSK, use a randomly generated key by an authentication server
    - For your client to trust the authentication server, you accept a digital certificate
    - Form a secure channel to the authentication server, which lets you enter your username and password
    - If the username and password are correct, the authentication server sends a one-time key to use instead of a PSK to both the client and the AP (also over a secure channel)
- The rest of the handshake proceeds normally

WPA-Enterprise Attacks

- WPA Enterprise defends against the previous attacks
  - **Rogue AP attack**: The APs must authenticate themselves to the server, which the attacker can't do
  - **Brute-force attack**: The generated PSK replacement is long and random, too long to brute-force
  - **No forward secrecy**: The generated PSK replacement is used once and then discarded, so no information is retained that allows the PTK to be recovered later
- However, it is still vulnerable to higher-layer attacks such as ARP or DHCP spoofing
  - WPA is really a layer 1 protocol, so it can't provide defenses for this!
  - Enterprise-grade APs can provide mitigations similar to enterprise-grade switches

Review: Offline and Online Attacks

- Online attack: The attacker interacts with the service
  - Attacker is limited by how often they can interact with the service
- Offline attack: The attacker performs all the computation themselves
  - Attacker is only limited by how much computation power they have
  - Offline attacks are far more dangerous than online attacks!
- WPA-PSK is vulnerable to **offline** brute-force attacks
  - The attacker can record a handshake and use values to check their guesses by themselves
  - We want to avoid this!

Simultaneous Authentication of Equals

- Goal: Alice and Bob want to create a shared secret
  - Alice and Bob both know a password
  - They can generate a shared secret only if both of them know the password
- If one of them doesn't know the password, they learn nothing about the password during the protocol, unless they correctly guessed the password during the protocol
  - Contrast with WPA2-PSK: During the protocol, an attacker learns enough information for an offline brute-force attack
  - No more offline attacks: An attacker must guess the password during the protocol (online attack)

Dragonfly

- Dragonfly: A protocol for simultaneous authentication of equals
    - Based on Diffie-Hellman (either conventional or elliptic-curve)
- Review: Diffie-Hellman
    - Two public parameters, p and g
    - Alice chooses a and sends $g^a \bmod p$ to Bob
    - Bob chooses b and sends $g^b \bmod p$ to Alice
    - Their shared secret is $(g^a)^b = (g^b)^a = g^{ab} \bmod p$
- Main idea: Use the password and Alice and Bob's identities to generate g
    - Note: Unlike standard Diffie-Hellman, g is not a public parameter anymore
    - Then use g to do a standard Diffie-Hellman key exchange
    - If Alice and Bob know the password, they will derive the same g and obtain the same shared secret
    - If one of them doesn't know the password, they will create a completely different random key!

DH-based Dragonfly

- Slightly more complex version of the previous idea
- Public parameters:
  - A prime p
    - A generator over this prime G
  - A smaller prime q
    - Size of the group defined by G and q is a large prime divisor of $(p-1)/2$
  - A selected generator g is valid if $g < p$ and $g^q \bmod p = 1$
  - Same idea as with DSA (Digital Signature Algorithm), a DH based signature scheme: We can use a smaller specialized group and be sending smaller data elements around
- Identifiers for Alice and Bob
  - EG, MAC addresses, with an ordering function
- Key idea:
  - Select a *random* generator g, called P (or PE = Password Element) based on $H(ID_a \mid\mid ID_b \mid\mid PW)$

Dragonfly: Creating a Shared Secret


```
found = False
counter = 1
n = len(p) + 64
do {
  base = H(max(Alice,Bob) | min(Alice,Bob) | password | counter)
  temp = KDF-n(base, "Dragonfly Hunting And Pecking")
  seed = (temp mod (p - 1)) + 1
  temp = seed ^ ((p-1)/q) mod p
  if (temp > 1)
  then
    if (not found)
      PE = temp
      found = true
    fi
  fi
  counter = counter + 1
} while ((!found) || (counter <= k))
```

"Hunt and peck": Pick an element at (pseudo)random. If it's not mathematically valid, try again until you find a mathematically valid element

Dragonfly: Creating a Shared Secret


```
found = False
counter = 1
n = len(p) + 64
do {
  base = H(max(Alice,Bob) | min(Alice,Bob) | password | counter)
  temp = KDF-n(base, "Dragonfly Hunting And Pecking")
  seed = (temp mod (p - 1)) + 1
  temp = seed ^ ((p-1)/q) mod p
  if (temp > 1)
  then
    if (not found)
      PE = temp
      found = true
    fi
  fi
  counter = counter + 1
} while ((!found) || (counter <= k))
```

Side channel attack: Attacker checks how long the protocol takes and learns information about the secret

Defense: Specify a minimum number of iterations *k* before the protocol ends

We use the first valid element found, but we keep running for *k* iterations (choosing *k* so that the probability of failure is low enough): but still often 40+ times so *k* is substantial

Dragonfly: Creating a Shared Secret


```
found = False
counter = 1
n = len(p) + 64
do {
  base = H(max(Alice,Bob) | min(Alice,Bob) | password | counter)
  temp = KDF-n(base, "Dragonfly Hunting And Pecking")
  seed = (temp mod (p - 1)) + 1
  temp = seed ^ ((p-1)/q) mod p
  if (temp > 1)
  then
    if (not found)
      PE = temp
      found = true
    fi
  fi
  counter = counter + 1
} while ((!found) || (counter <= k))
```

Alice and Bob's identity are part of the input to the shared secret

We can't precompute this the first time because we include Alice and Bob's identity in determining *P*

Eliminating this would eliminate the need for online computation of *P*

But we can cache *P*: an important optimization since this calculation is expensive!

Now to prove that everybody knows the same P... And generate a key

- Alice creates two random values:
  - $1 < r_a < q$ (the random value)
  - $1 < m_a < q$ (the mask value)
- Alice now computes
  - $s_a = (r_a + m_a) \bmod q$
  - $E_a = P^{-mask}$
  - Sends those to Bob, Bob sends his counterparts to Alice
- Now the starting secret…
  - $ss = (P^{s_b} E_b)^{r_a} = (P^{(r_b + m_b - m_b)})^{r_a} = P^{r_a r_b}$
  - Sends those to Bob, Bob sends his counterparts
  - Verify Psb and Sb are valid
  - Computes $H(ss \mid E_a \mid s_a \mid E_b \mid s_b)$ and sends that to Bob
  - verifies Bob's counterpart which uses a different order
- Final:

  $K = H(ss \mid E_a * E_b \mid s_a + s_b)$

Graphically

Alice

Calculate: P

Randoms: $1 < r_a < q$, $1 < m_a < q$

$s_a = (r_a + m_a) \bmod q$

$E_a = P^{-m_a}$

Bob

Calculate: P

Randoms: $1 < r_b < q$, $1 < m_b < q$

$s_a = (r_b + m_b) \bmod q$

$E_b = P^{-m_b}$

s_a, E_a

s_b, E_b

Alice: $ss = (P^{s_b} E_b) r_a$

Bob: $ss = (P^{s_a} E_a) r_b$

$H(ss \mid E_a \mid s_a \mid E_b \mid s_b)$

$H(ss \mid E_b \mid s_b \mid E_a \mid s_a)$

Alice: Verify

Bob: Verify

$$K = H(ss \mid E_a * E_b \mid s_a + s_b)$$

WPA3 and Dragonfly

- WPA3 adds a Dragonfly key exchange before the standard WPA handshake
  - Password from WPA2 is replaced with the shared secret from Dragonfly
  - Recall WPA-PSK: PSK is derived from password
  - WPA3: PSK is the shared secret from Dragonfly
- Performance cost: Extra latency per handshake
  - Additional handshakes before the 4-way handshake to create the shared secret
- Security benefits
  - Offline brute-force attacks are eliminated
  - Provably secure: An incorrect password guess doesn't give the attacker any extra information about the password
  - Eliminates the "adversary with the password" attacks except for a rogue AP with the password

Summary: Wireless Local Networks

- WPA: A protocol to encrypt Wi-Fi connections at layer 1
  - Messages between the client and the AP are encrypted with keys
  - Handshake uses MICs (cryptographic MACs) to verify that both parties have the same PSK and nonces
- WPA-PSK: Use a password to derive a PSK, which is used in a handshake to arrive at a key
  - Attack: Attacker can pretend to be an AP
  - Attack: Brute-force the password after recording a handshake
  - Vulnerability: No forward secrecy
- WPA-Enterprise: Use a third party to provide a one-time “replacement PSK,” used in the same handshake
  - Solves the attacks on WPA-PSK
- WPA3: Use a password to derive *g* in the Diffie-Hellman key exchange, which is then used to derive a one-time “replacement PSK”
  - Solves all attacks except for online password guessing and rogue APs that know the password