# Introduction To Networking

# 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)

# The Internet

Global network that provides **best-effort** delivery of **packets** between connected hosts

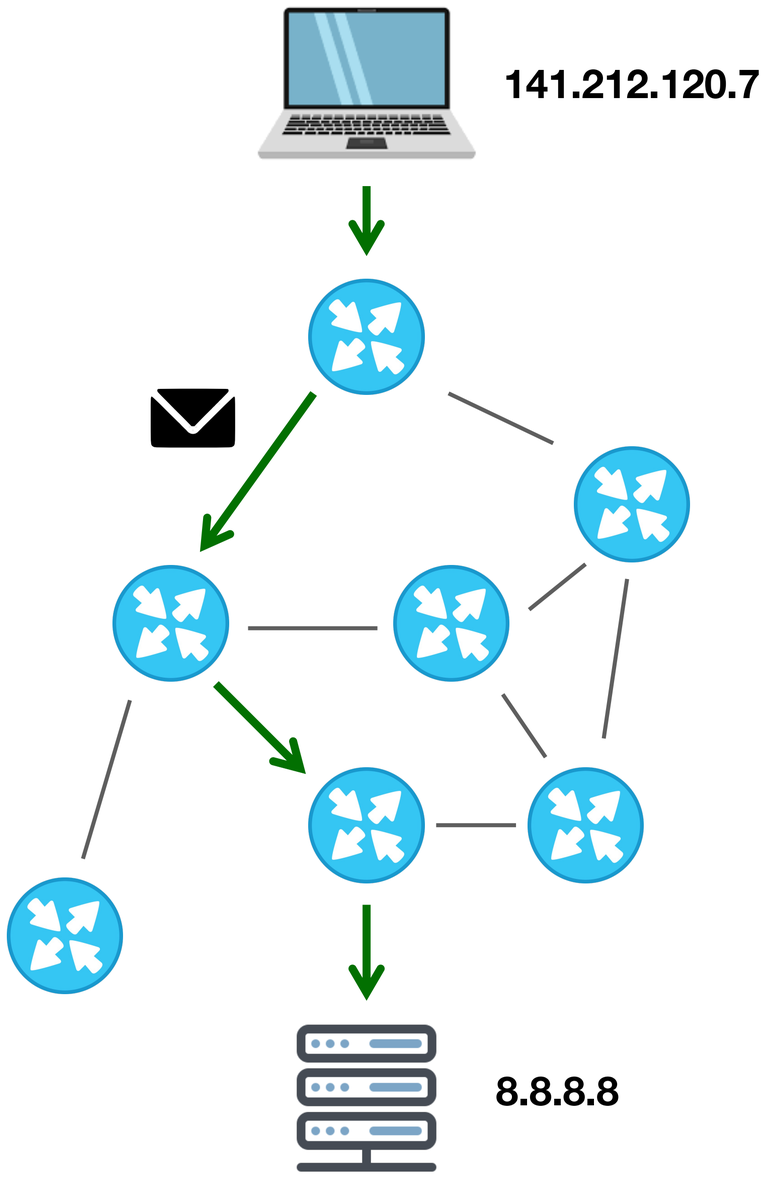
**Packet**: a structured sequence of bytes

**Header:** metadata used by network

**Payload:** user data to be transported

Every host has a unique identifier — IP address

Series of routers receive packets, look at destination address on the header and send it one hop towards the destination IP address

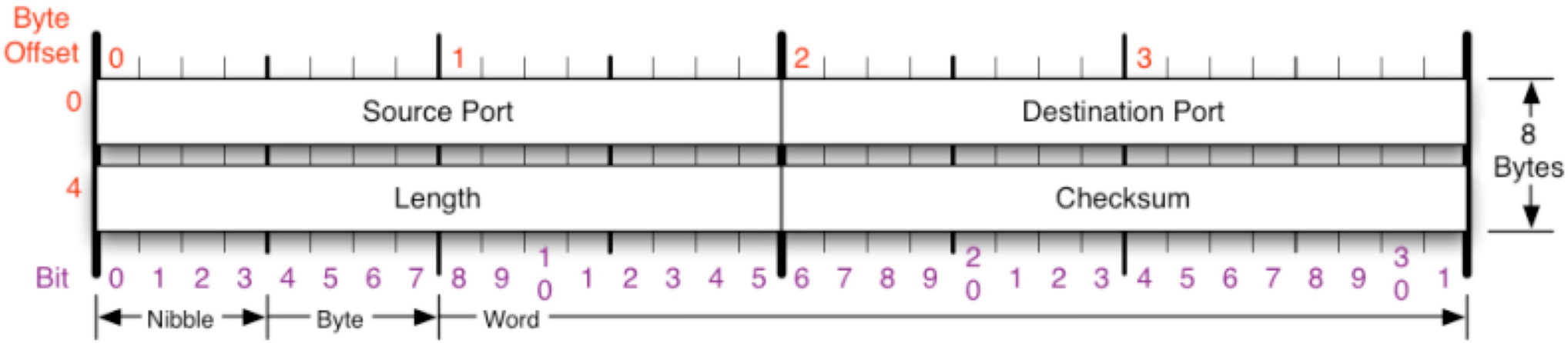


# Network Protocols

We define how hosts communicate in published network protocols

**Syntax:** How communication is structured (e.g., format and order of messages)

**Semantics:** What communication means. Actions taken on transmit or receipt of message, or when a timer expires. What assumptions can be made.




**Example: What bytes contain in each field in a packet header**

# Internet Analogy: Mail

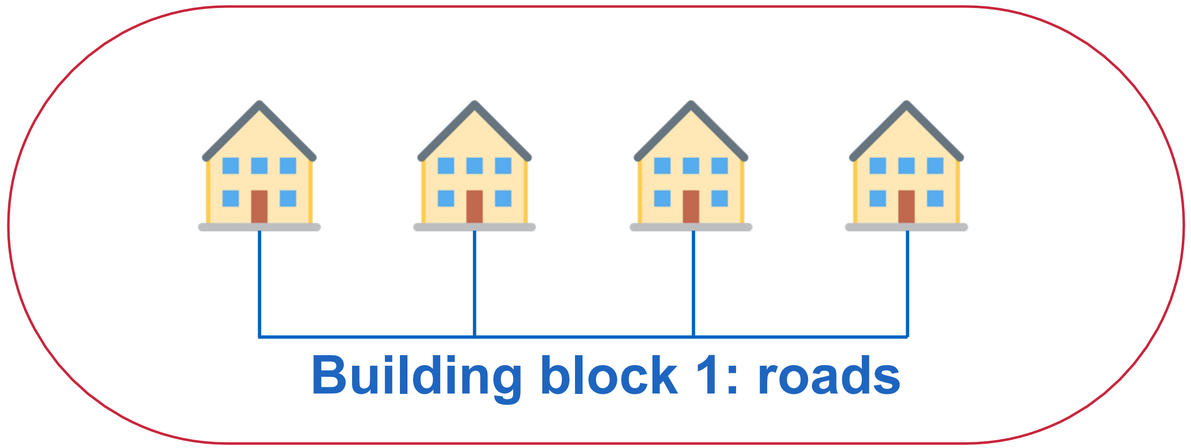
# Building block 1: Something That Moves Data

- The Internet is built on technology that moves bits across space
- Voltages on wires, wireless technology, radio waves, etc.

**Building block 1: roads**

# Building block 2: Apartment Complex

- Using building block 1, we can link people within a local apartment complex
- Forms a local area network (LAN)

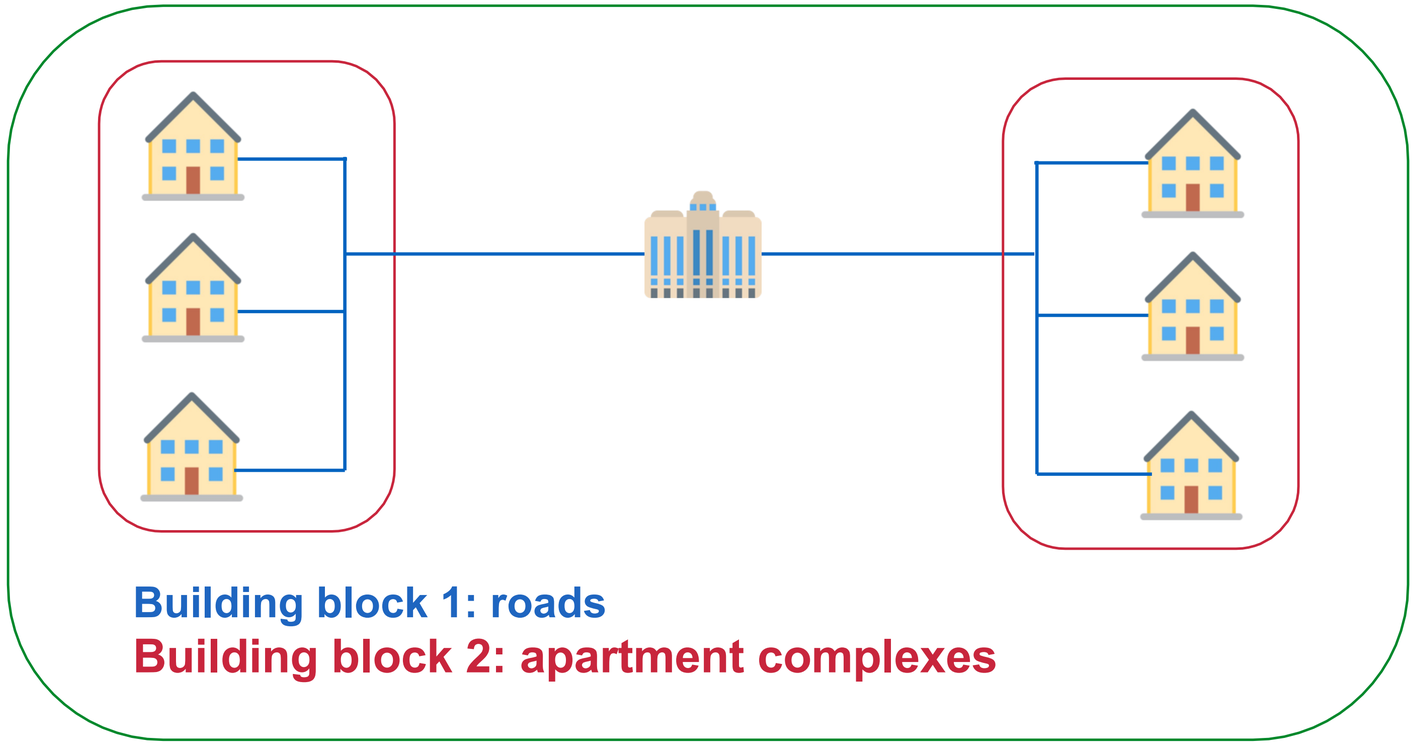



**Building block 2: apartment complex**

# Building block 3: Post offices

- A post office connects two or more apartment complexes
- Forms a wide area network

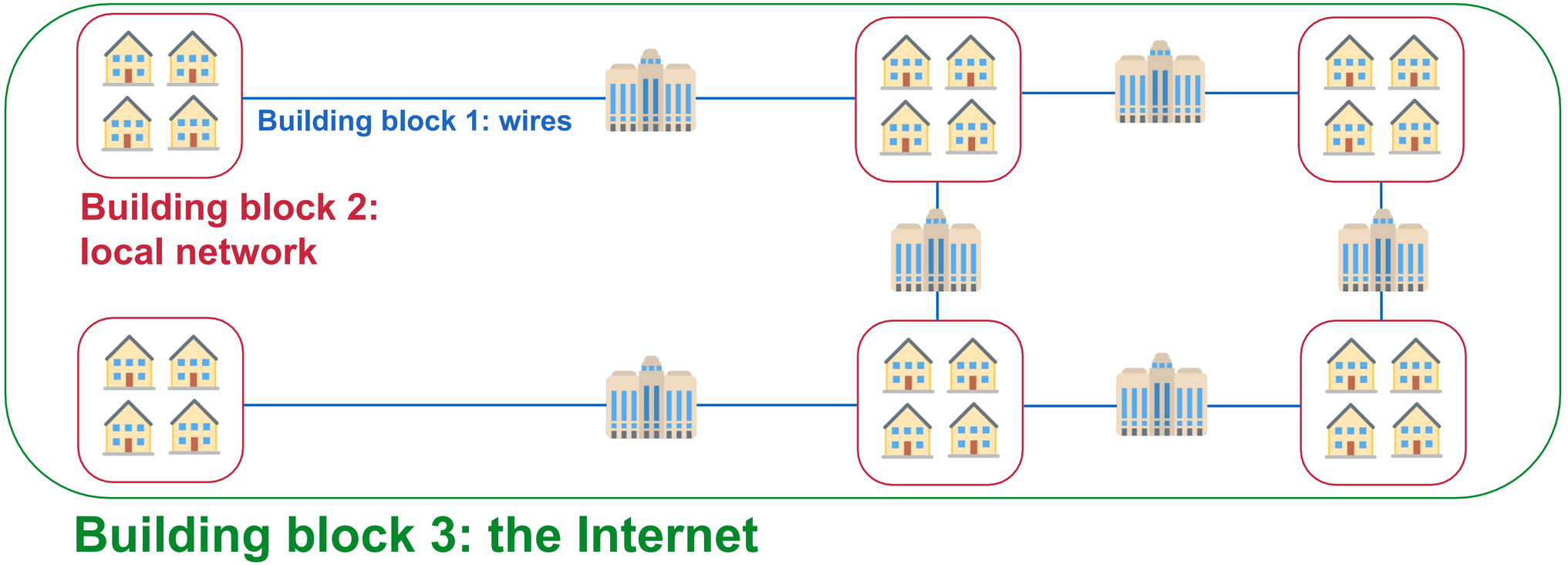




# Building block 3: The Internet

- Connect the entire world using post offices
- Messages may pass through multiple post offices before reaching destination





# Layers of abstraction

Layer 3: Connect many local networks to form a global network

Layer 2: Create links in a local area

Layer 1: Move bits across space

- A change in layer 1 implementation (wireless instead of wires) doesn't affect the other layers

- A change in layer 2 protocols doesn't affect the other layers

# Protocol Layering

Networks use a stack of protocol layers

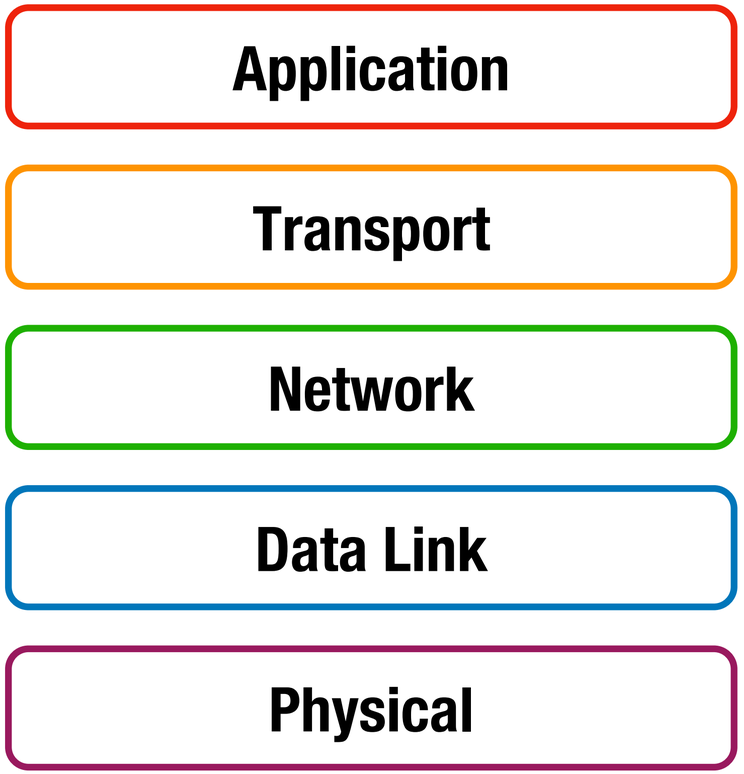
- Each layer has different responsibilities.
- Layers define abstraction boundaries

Lower layers provide services to layers above

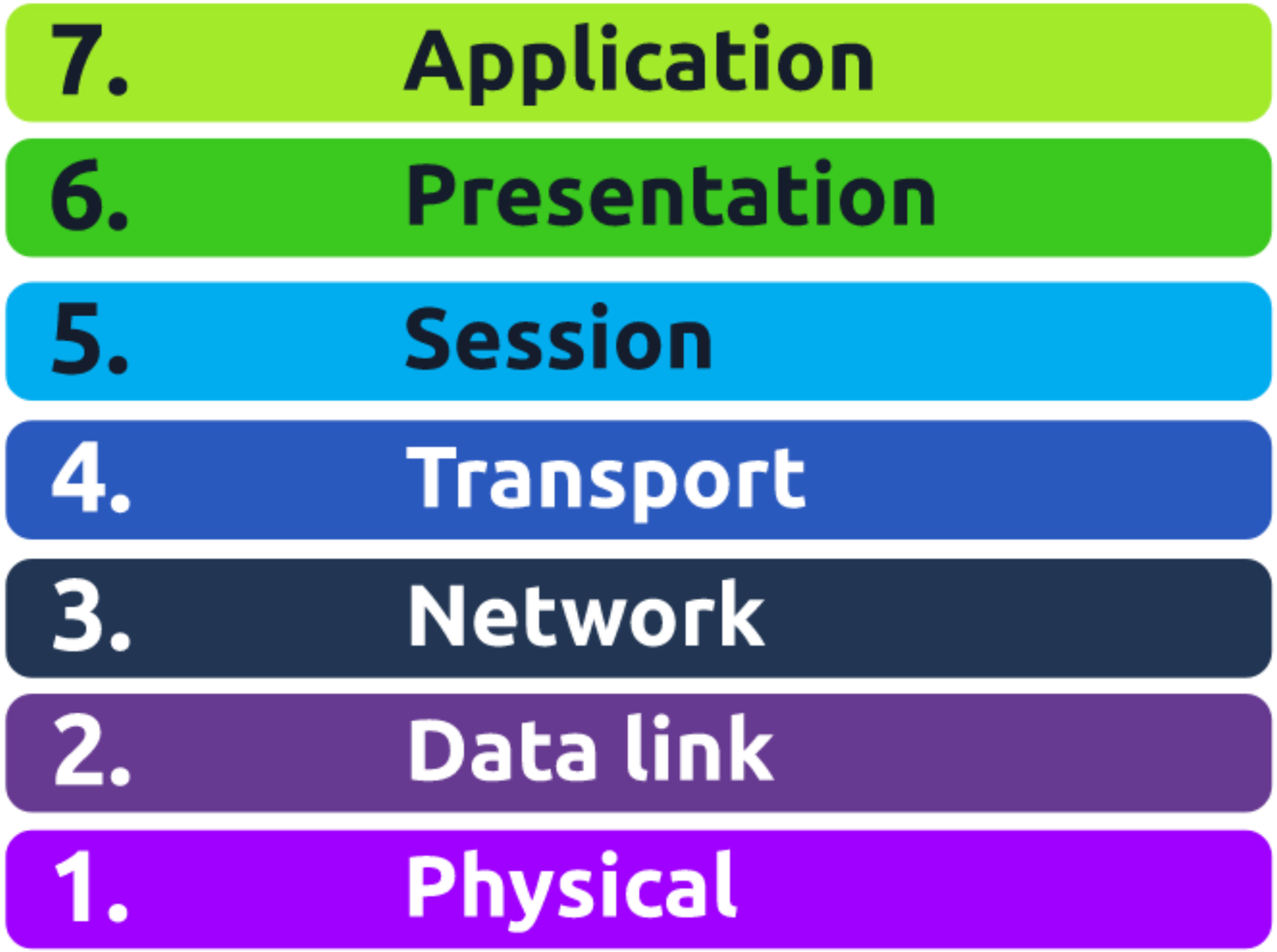
- Don’t care what higher layers do

Higher layers use services of layers below

- Don’t worry about how it works

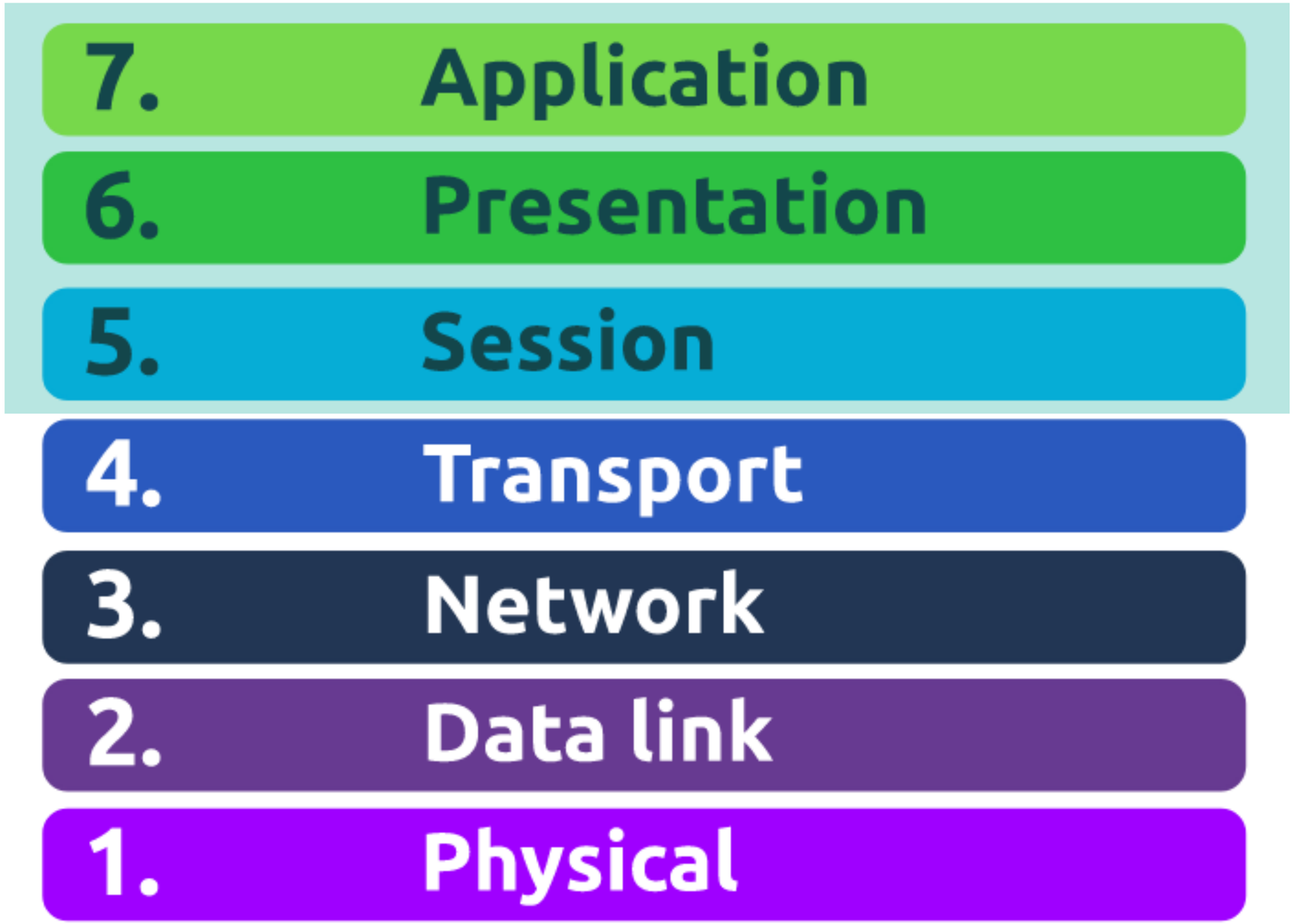


# Open Systems Interconnection (OSI) Model



# Open Systems Interconnection (OSI) Model

Application →



# OSI 5 Layer Model

**Physical**

How do bits get translated into electrical, optical, or radio signals

# OSI 5 Layer Model

| | |
|---|---|
| **Data Link** | How to get packet to the next hop. Transmission of data frames between two nodes connected by a physical link. |
| **Physical** | How do bits get translated into electrical, optical, or radio signals |

# OSI 5 Layer Model

| | |
|---|---|
| **Network** | Packet forwarding. How to get a packet to the final destination when there are many hops along the way. |
| **Data Link** | How to get packet to the next hop. Transmission of data frames between two nodes connected by a physical link. |
| **Physical** | How do bits get translated into electrical, optical, or radio signals |

# OSI 5 Layer Model

| | |
|---|---|
| **Transport** | Allows a client to establish a connection to specific services (e.g., web server on port 80). Provides reliable communication. |
| **Network** | Packet forwarding. How to get a packet to the final destination when there are many hops along the way. |
| **Data Link** | How to get packet to the next hop. Transmission of data frames between two nodes connected by a physical link. |
| **Physical** | How do bits get translated into electrical, optical, or radio signals |

# OSI 5 Layer Model

| | |
|---|---|
| **Application** | Defines how individual applications communicate. For example, **HTTP** defines how browsers send requests to web servers. |
| **Transport** | Allows a client to establish a connection to specific services (e.g., web server on port 80). Provides reliable communication. |
| **Network** | Packet forwarding. How to get a packet to the final destination when there are many hops along the way. |
| **Data Link** | How to get packet to the next hop. Transmission of data frames between two nodes connected by a physical link. |
| **Physical** | How do bits get translated into electrical, optical, or radio signals |

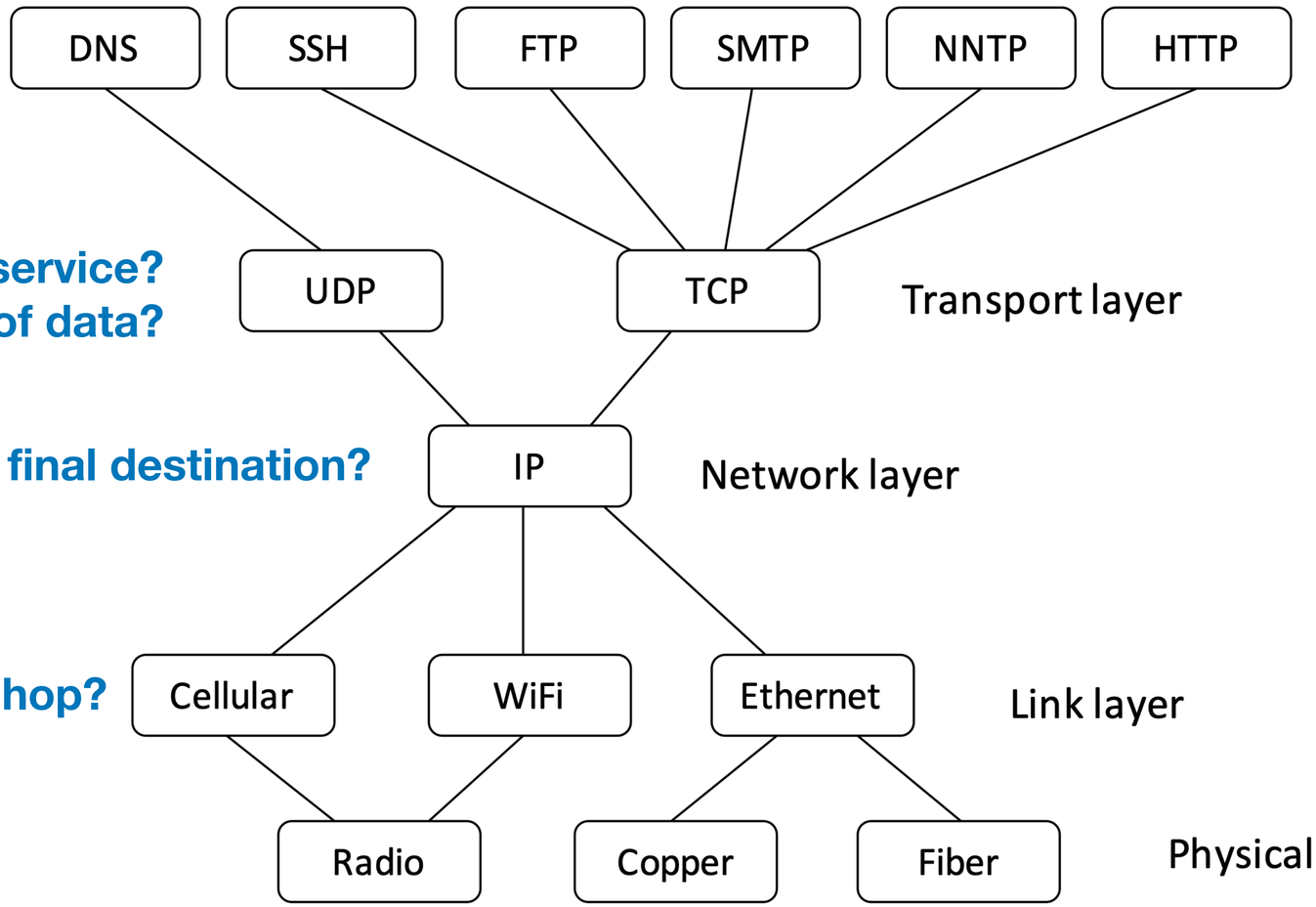
# Protocol Layering

How does Application structure data?

How do I get to the right service?
How do I have a reliable “stream” of data?

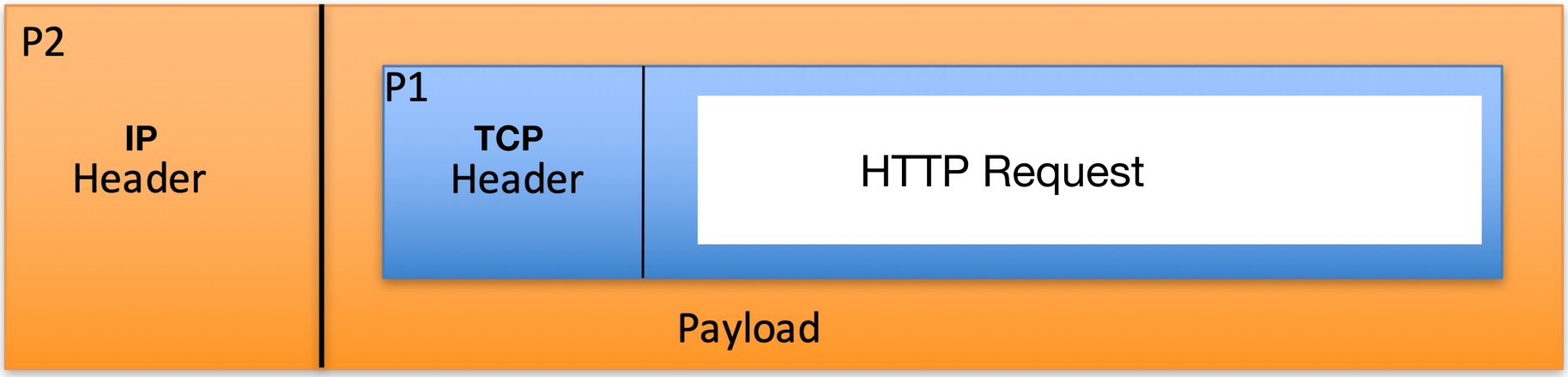
How a does packet final destination?

How do I get to next hop?



# Packet Encapsulation

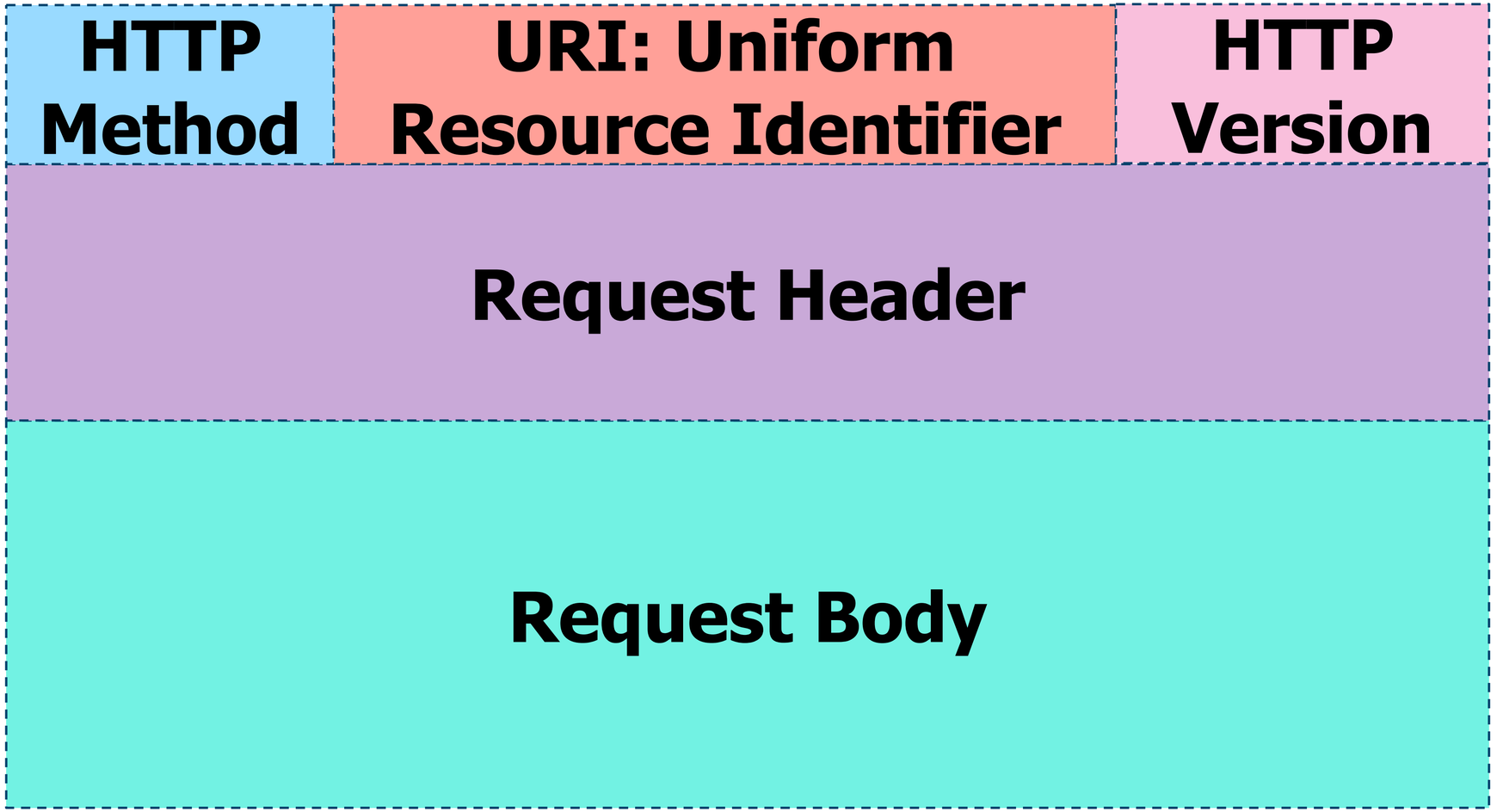
Protocol N1 can use the services of lower layer protocol N2
- A packet P1 of N1 is encapsulated into a packet P2 of N2
- The payload of p2 is p1
- The control information of p2 is derived from that of p1



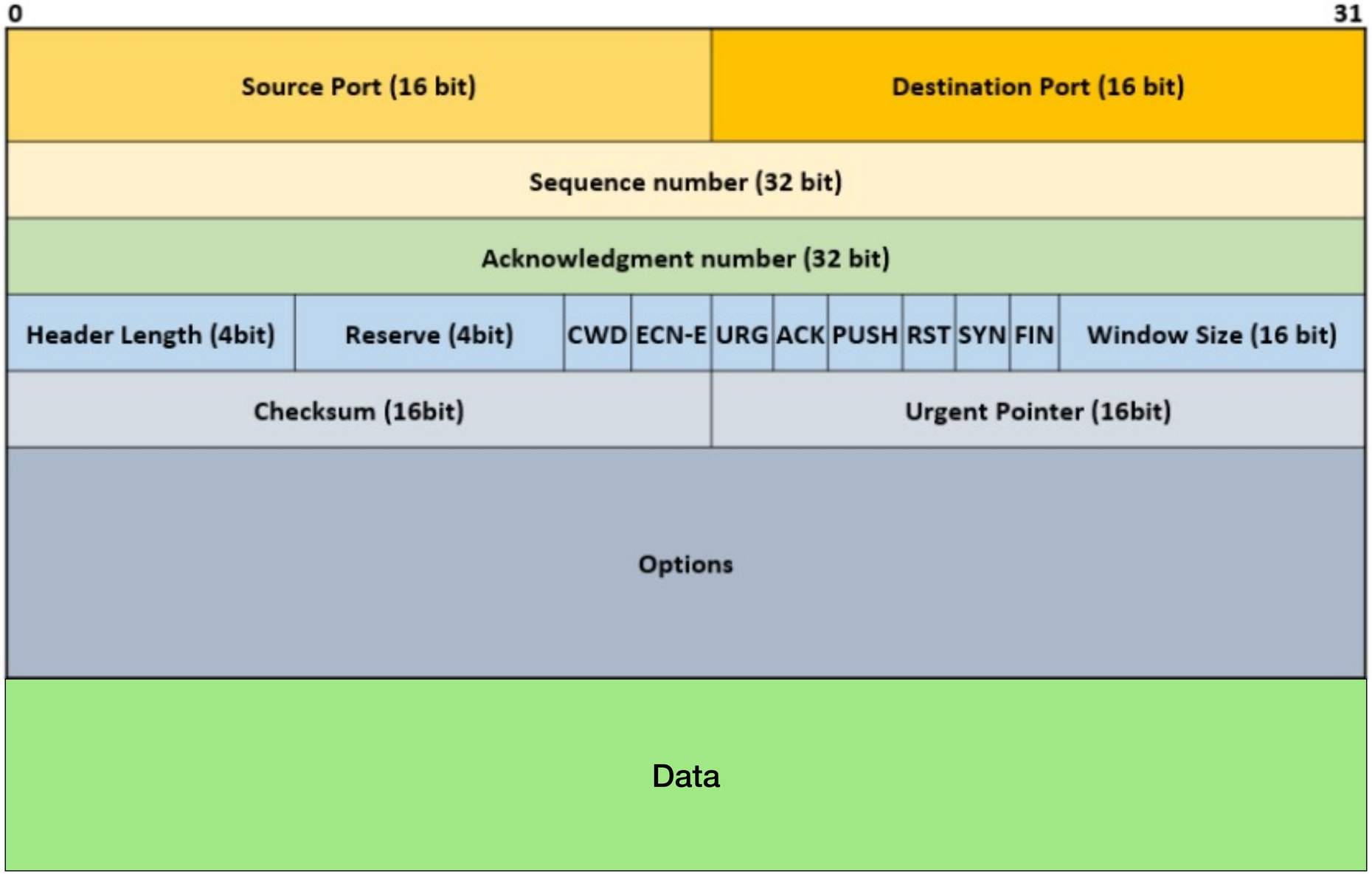
# Layers of Abstraction and Headers

- 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

# Layer 5 Header (HTTP)



# Layer 4 Header (TCP)

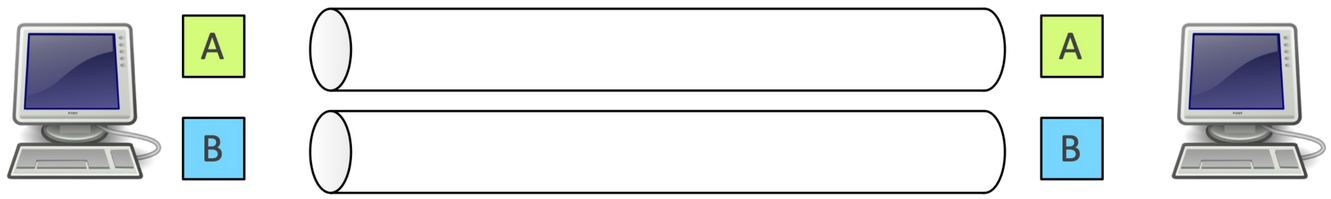


# Ports

Each application on a host is identified by a *port number*

TCP connection established between port *A* on host *X* to port *B* on host *Y*
Ports are 1–65535 (16 bits)

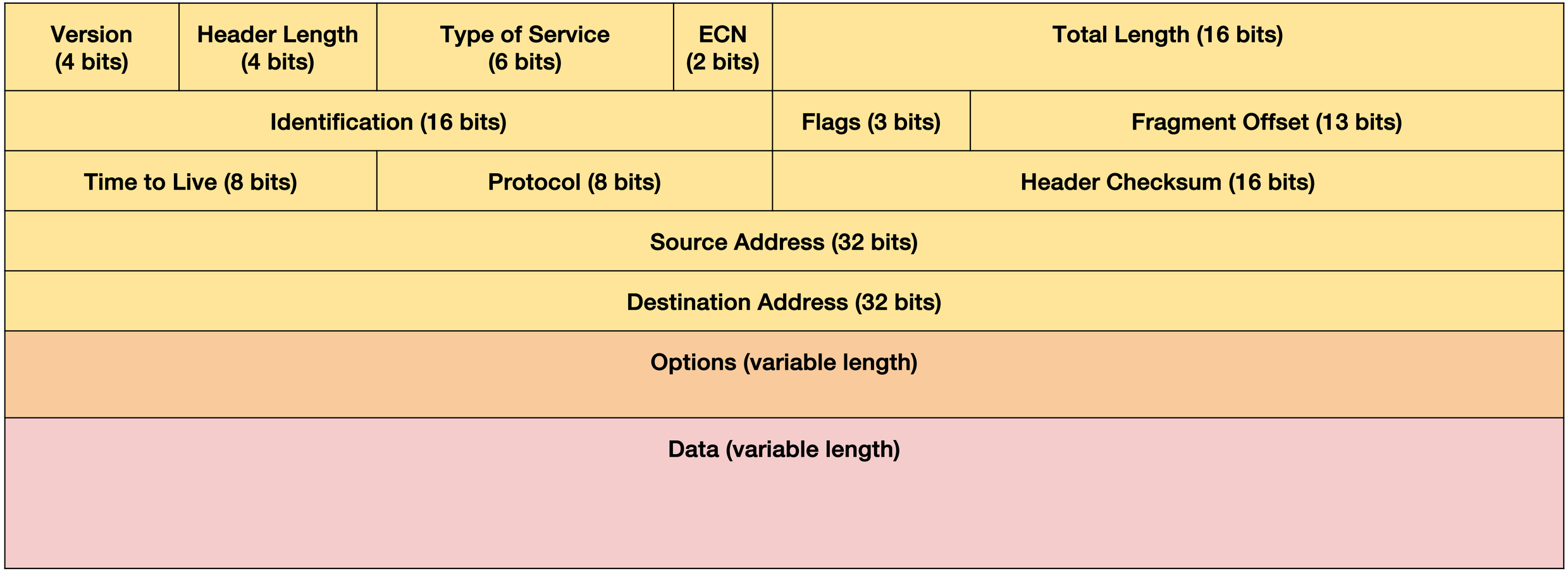
Some destination port numbers used for specific applications by convention



# Common Ports

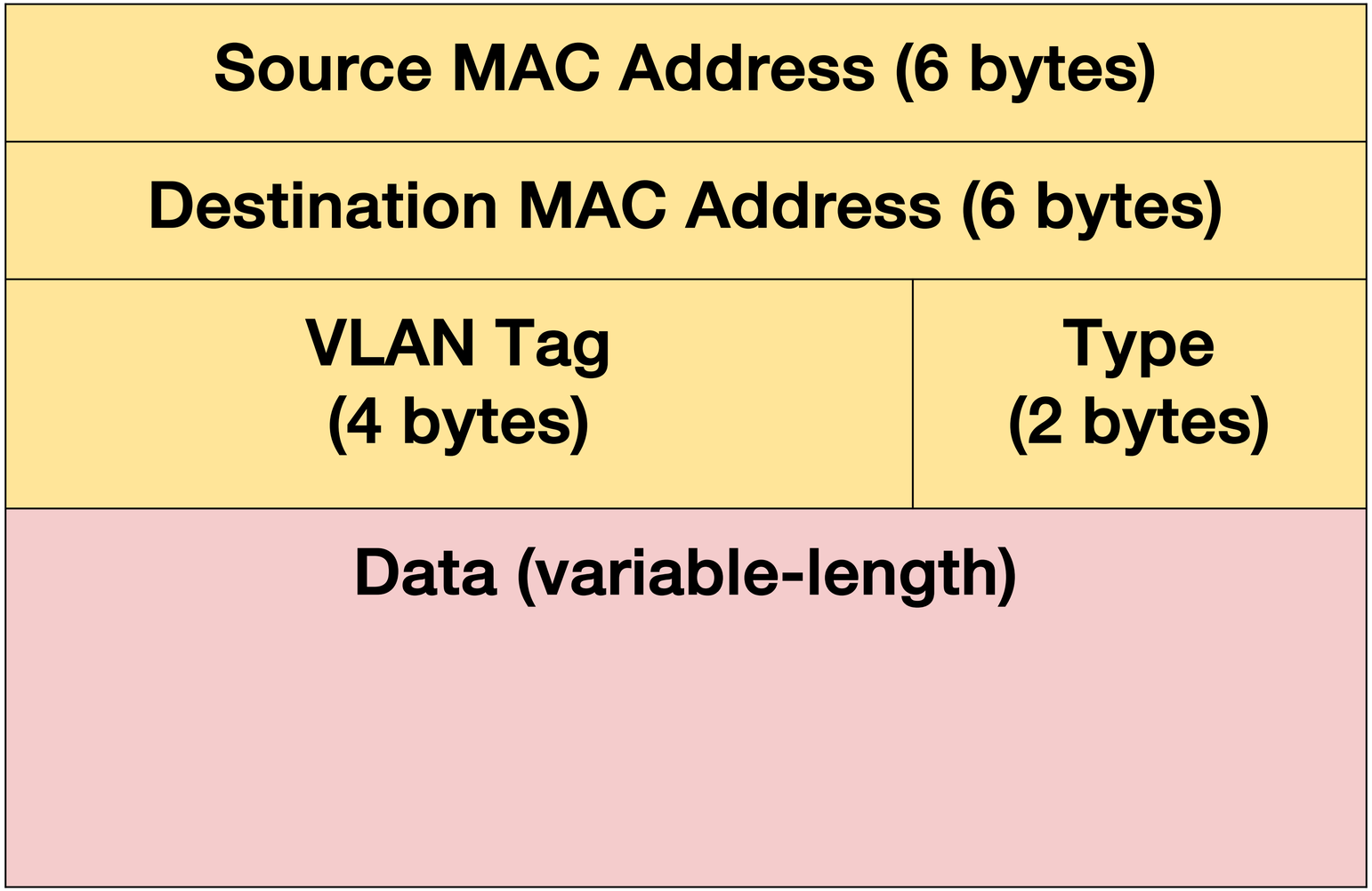
| Port | Application |
|---|---|
| 80 | HTTP (Web) |
| 443 | HTTPS (Web) |
| 25 | SMTP (mail) |
| 67 | DHCP (host config) |
| 22 | SSH (secure shell) |
| 23 | Telnet |

# Layer 3 Header (IP)



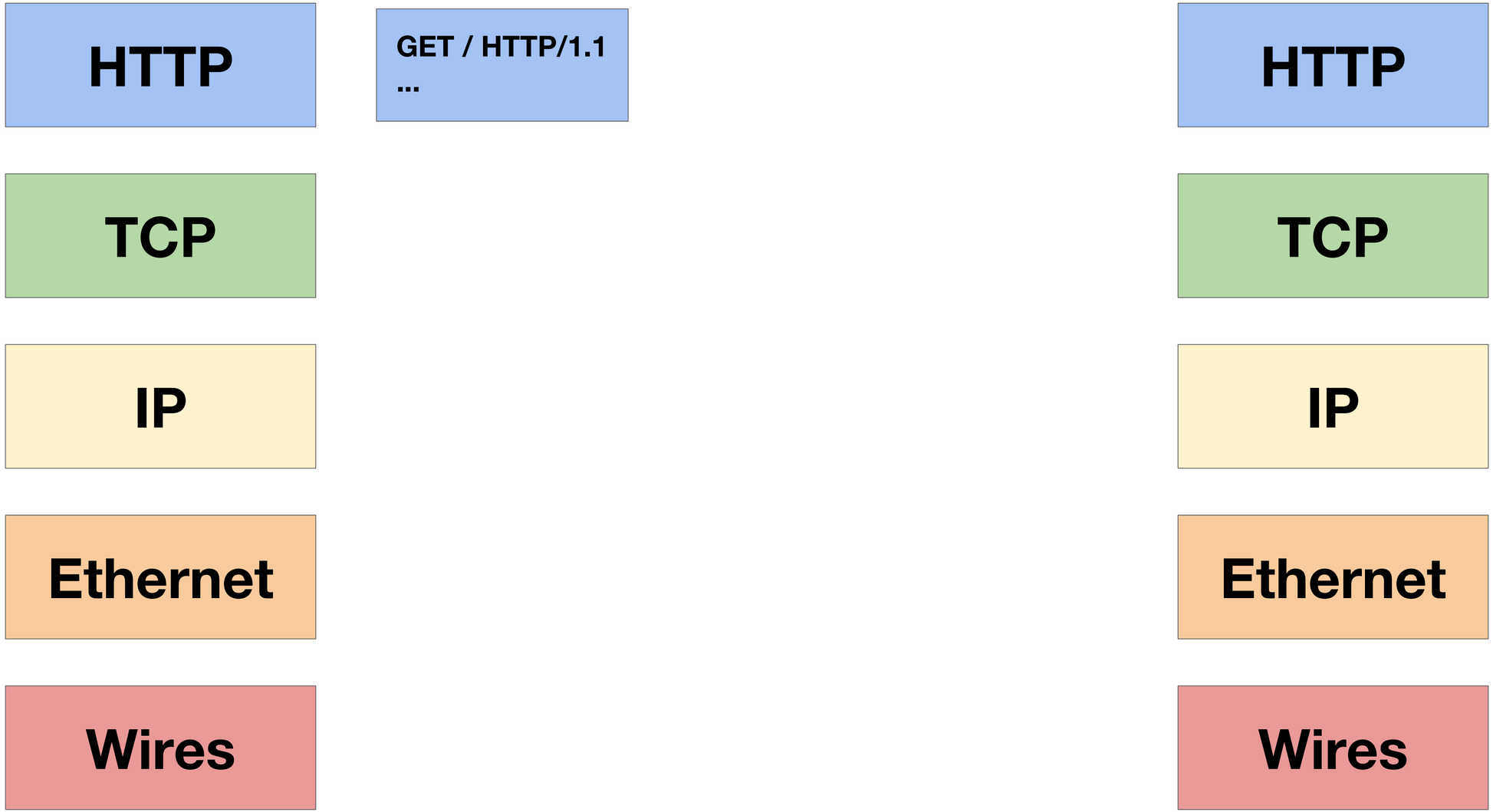

IPv4 header

# Layer 2 Header

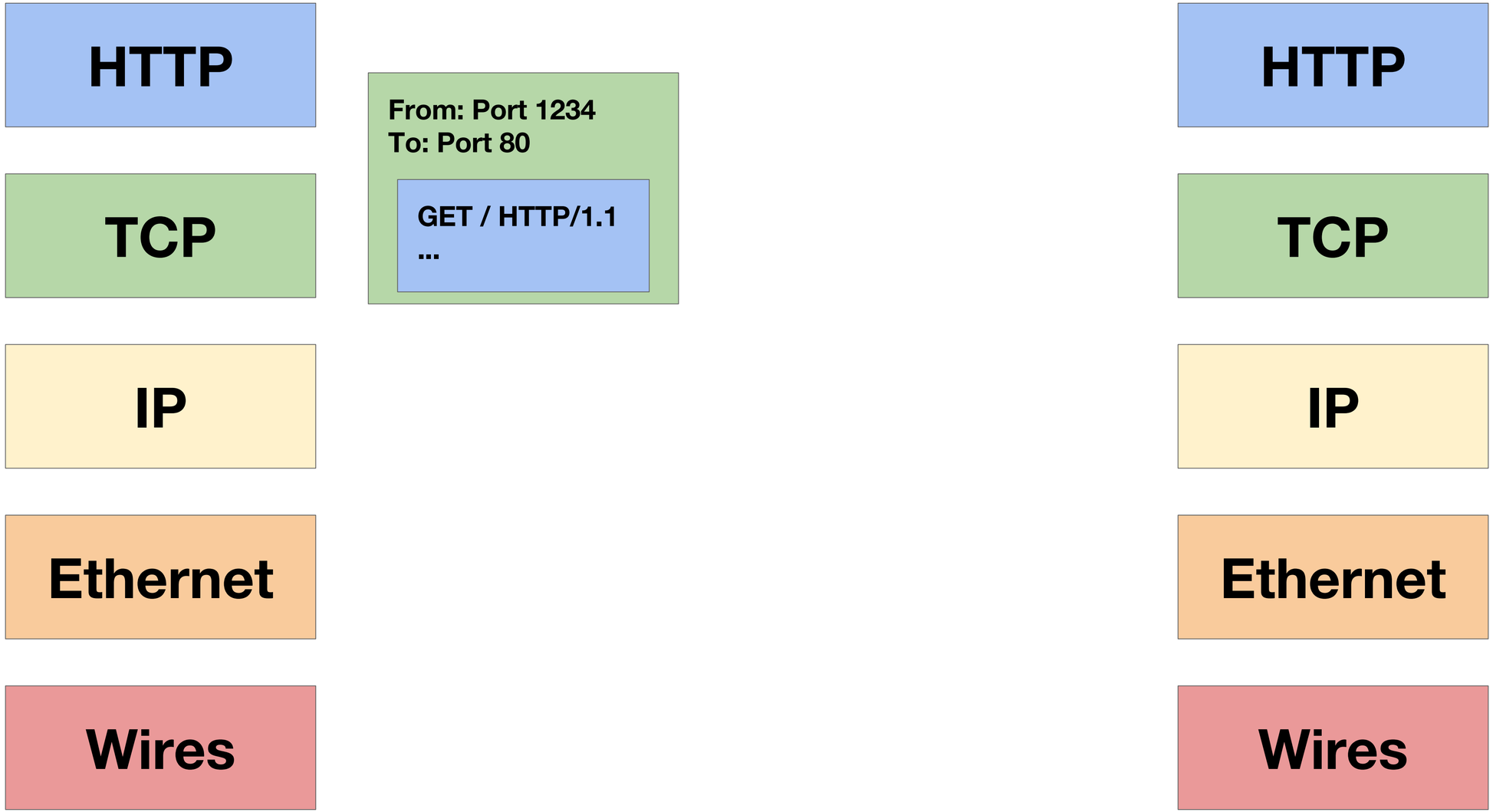



Ethernet and MAC address headers

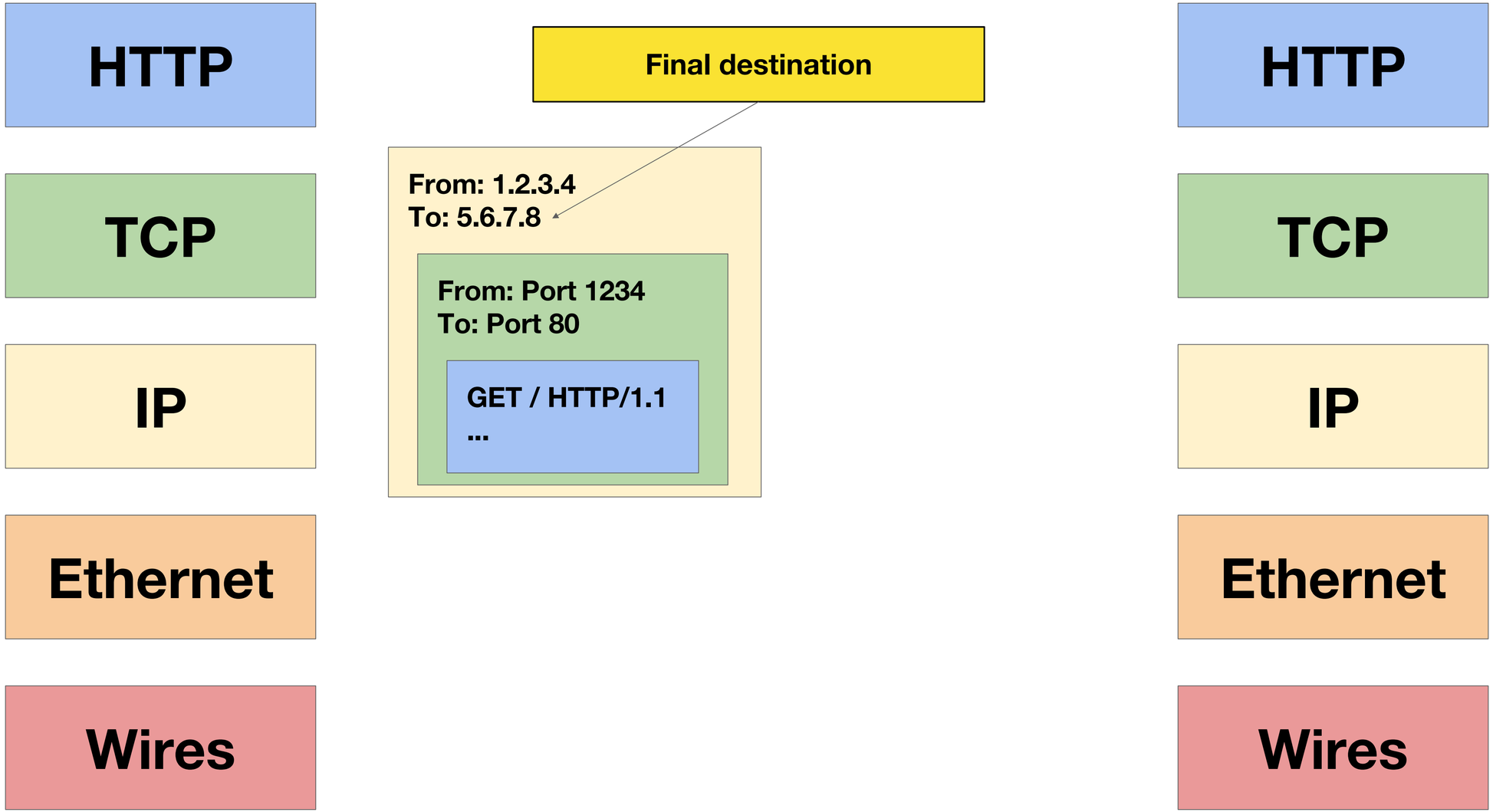
# Example: HTTP Request



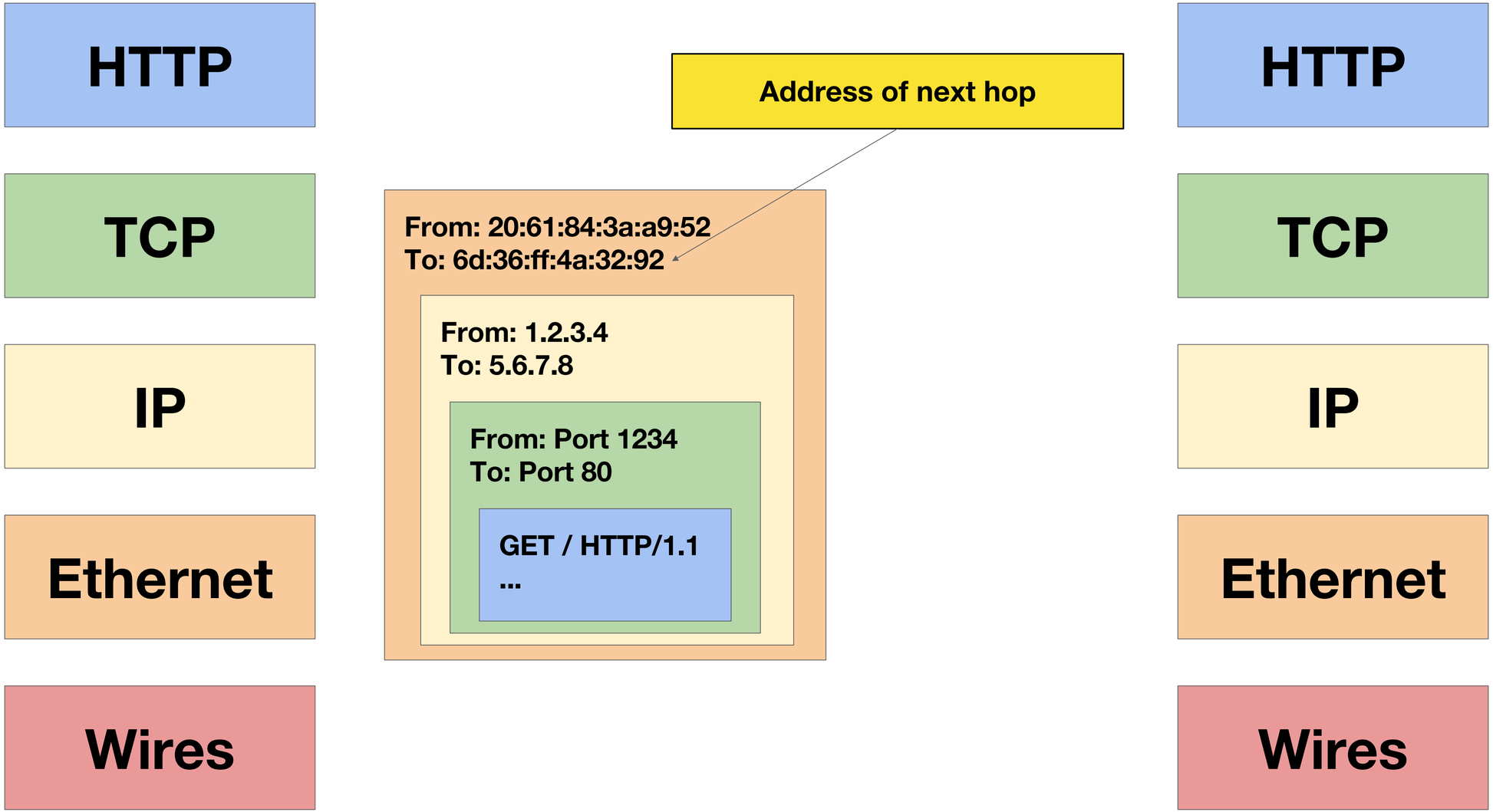
# Example: HTTP Request



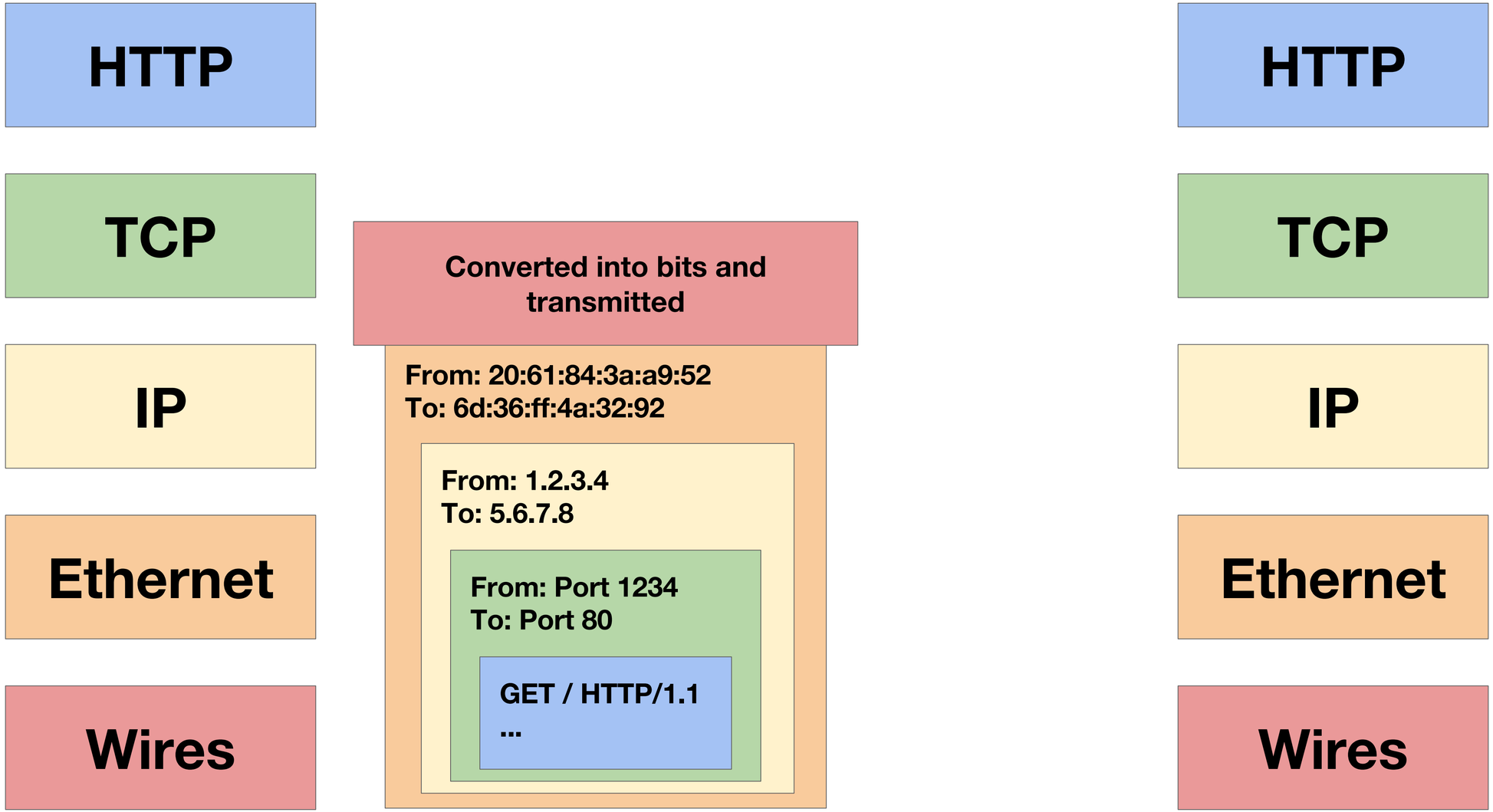
# Example: HTTP Request



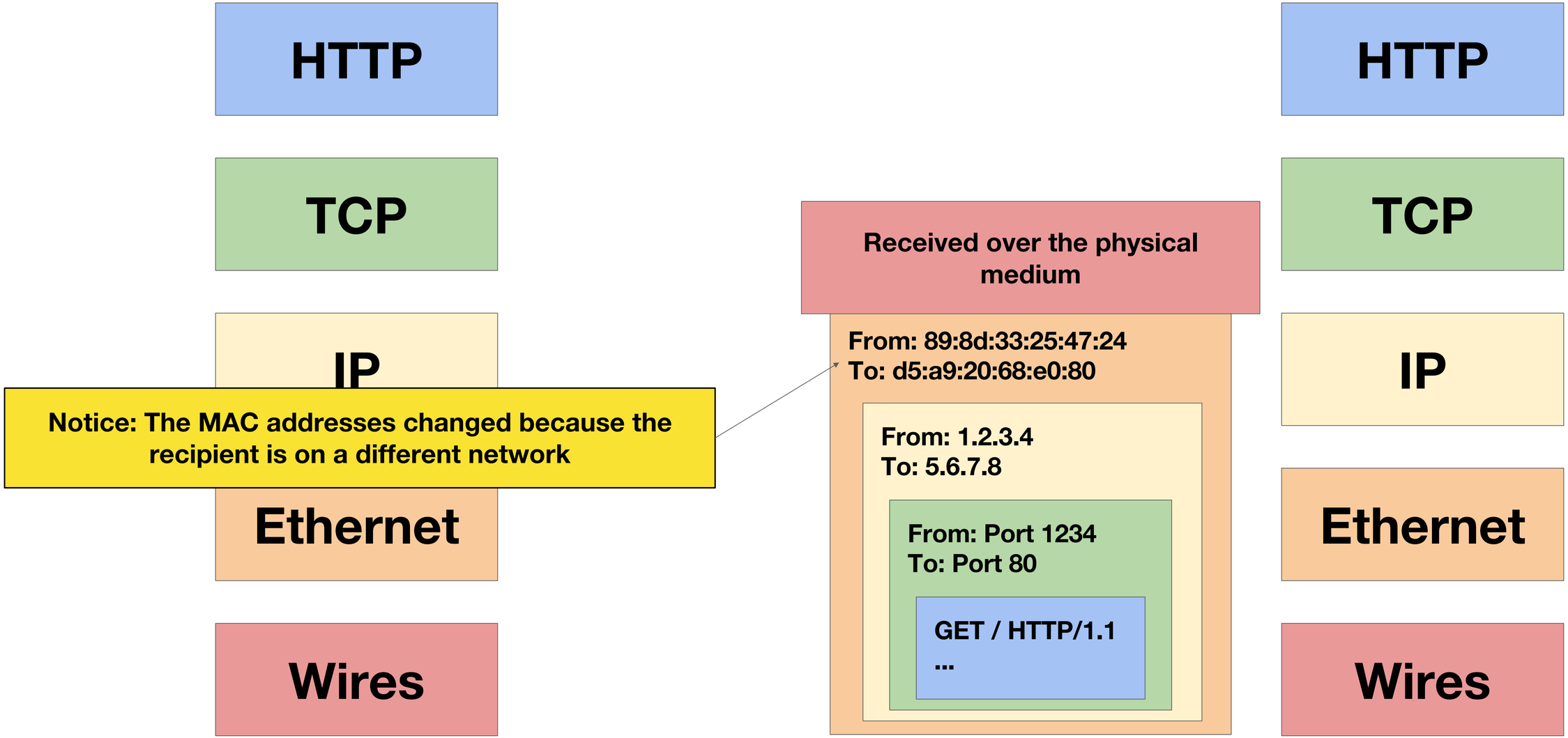
# Example: HTTP Request



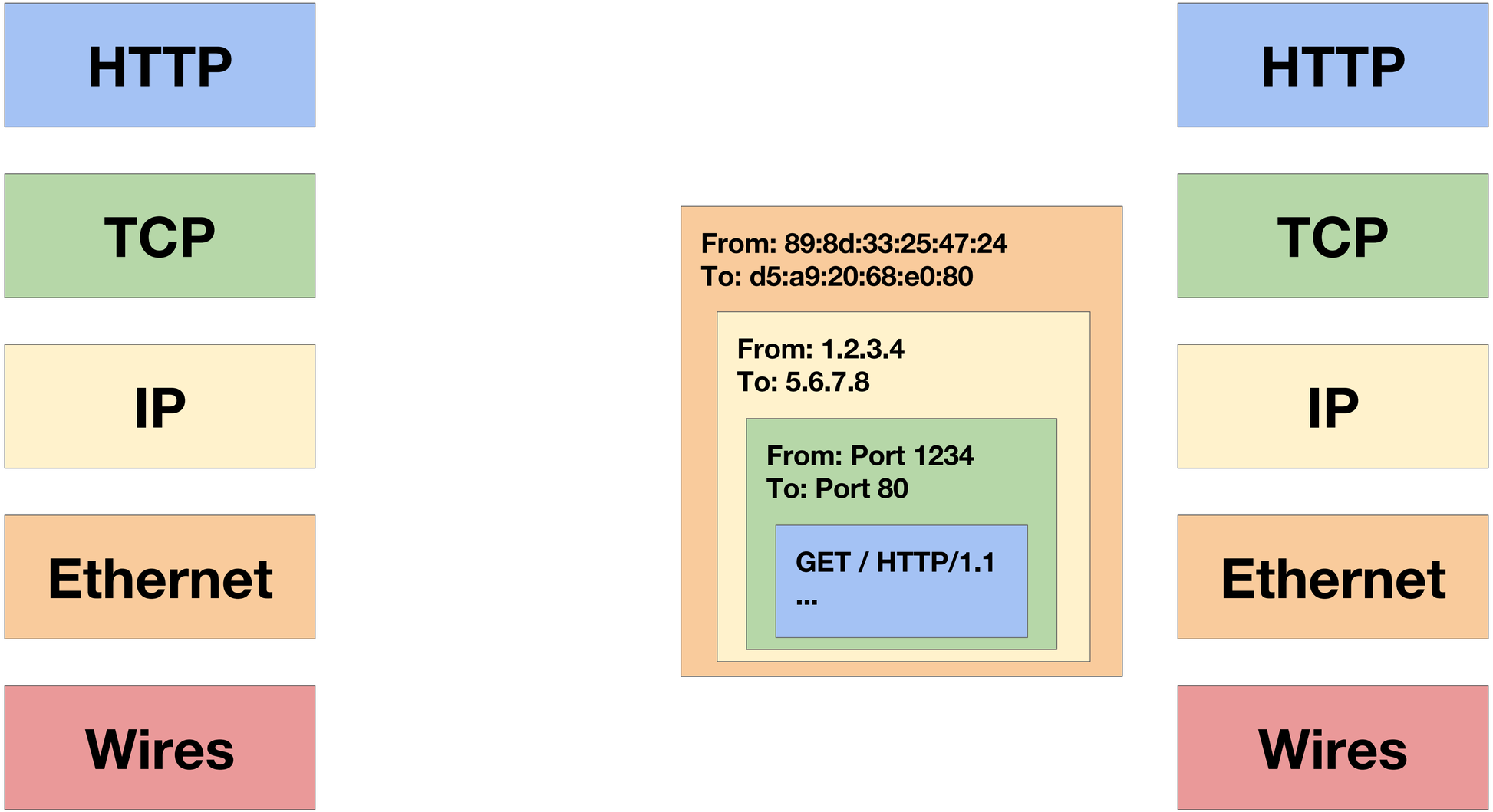
# Example: HTTP Request



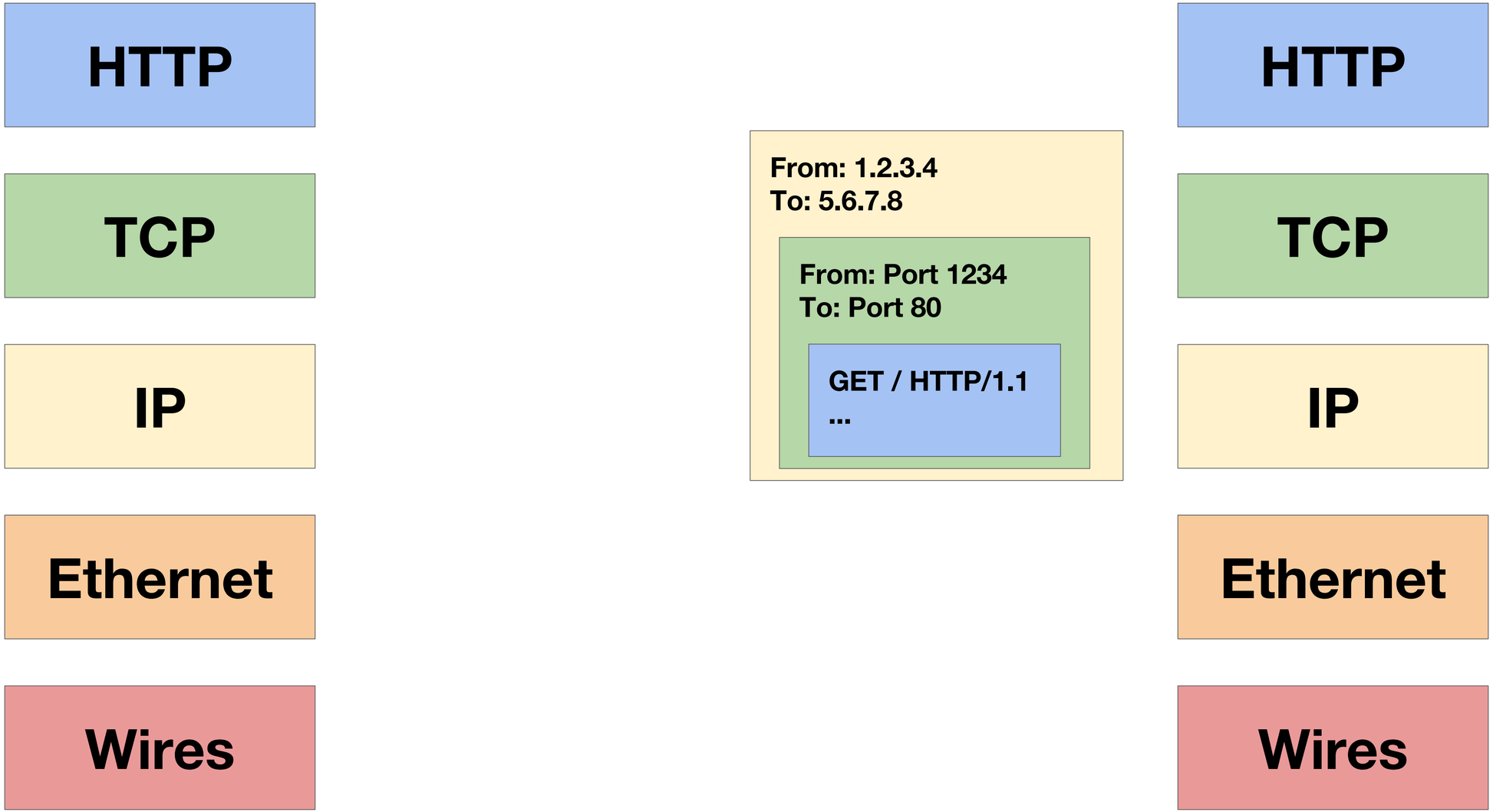
# Example: HTTP Request



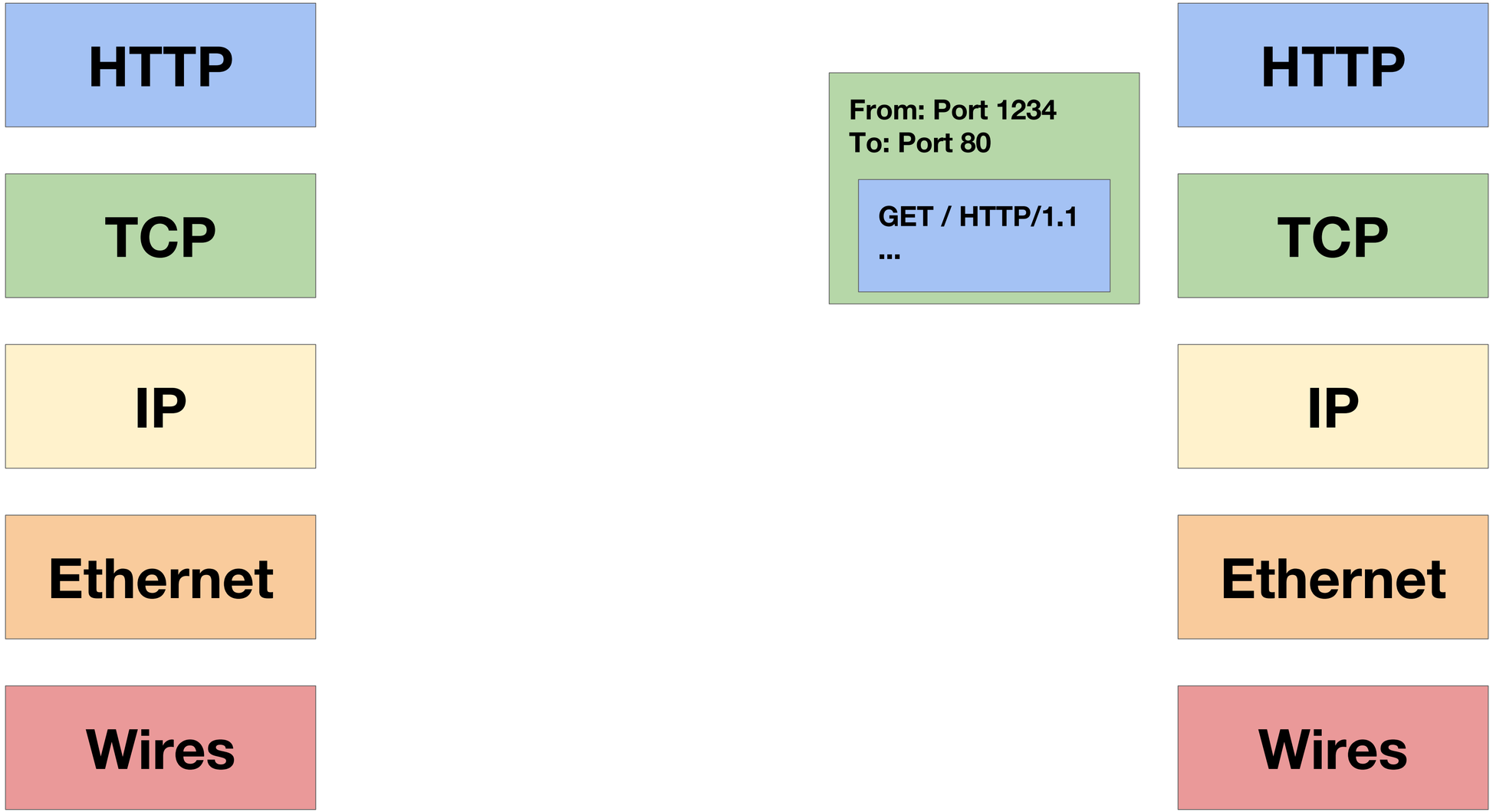
# Example: HTTP Request



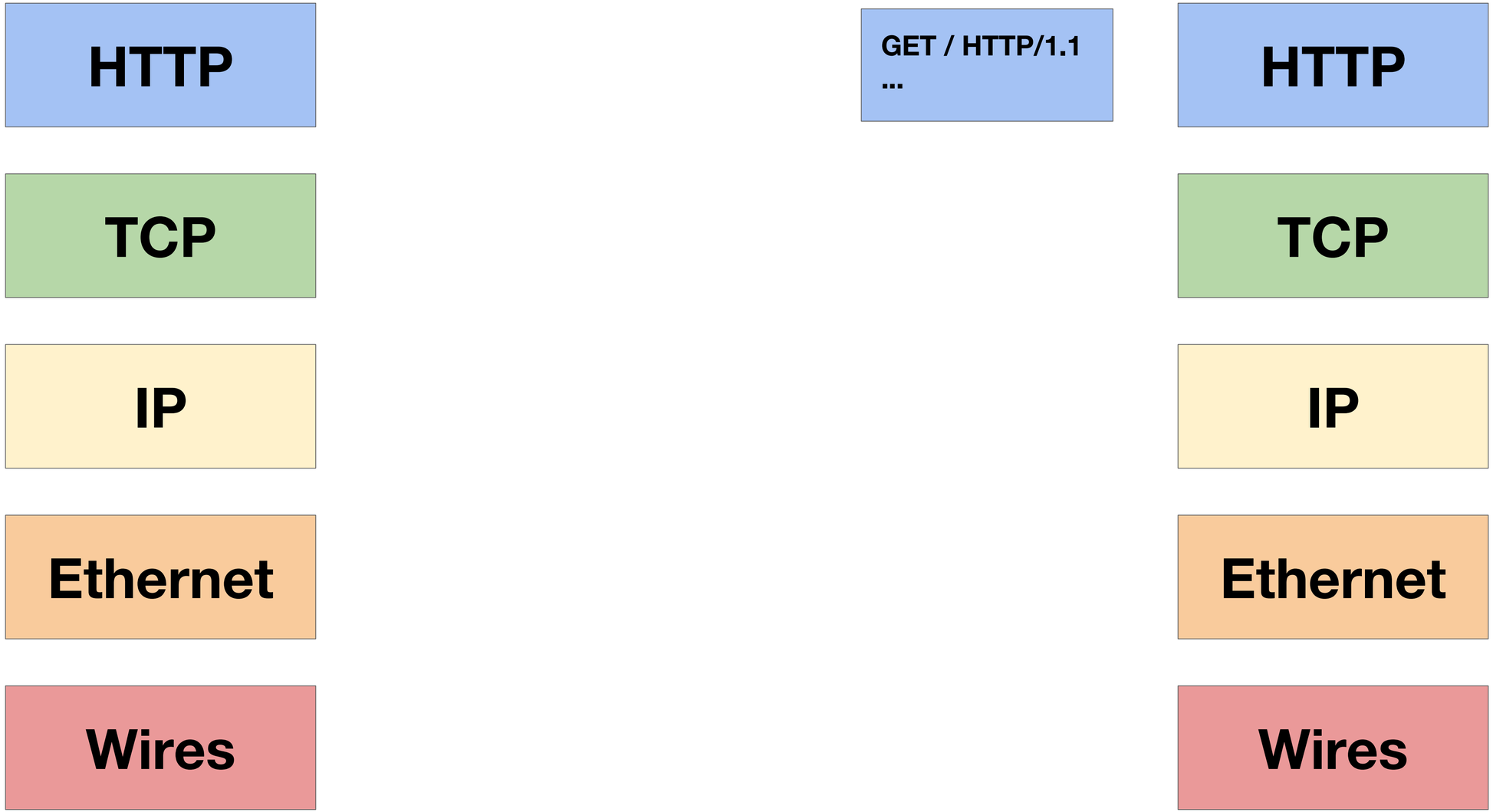
# Example: HTTP Request



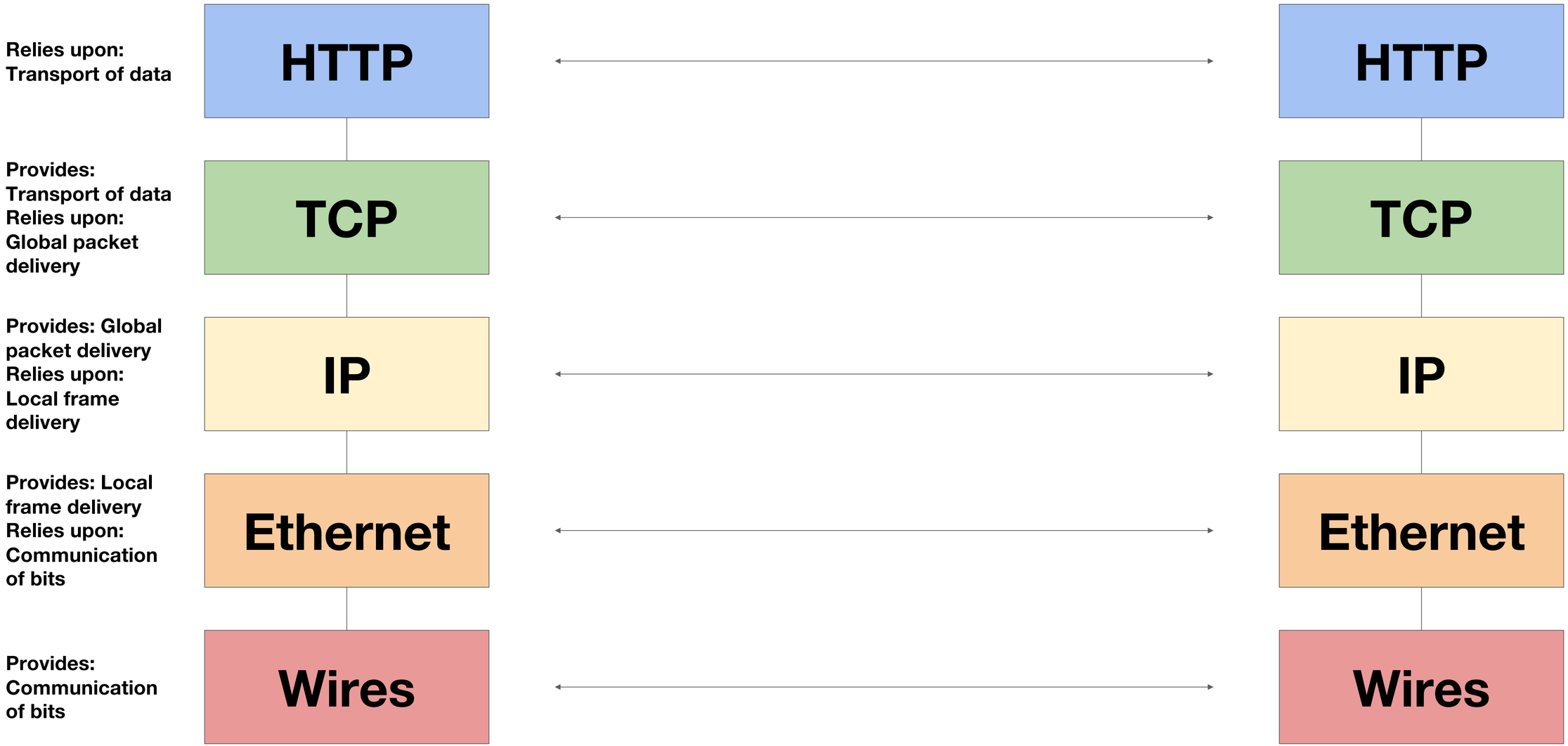
# Example: HTTP Request



# Example: HTTP Request



# Example: HTTP Request



# Internet Protocol (IP)

**Internet Protocol (IP)** defines what packets that cross the Internet need to look like to be processed by routers

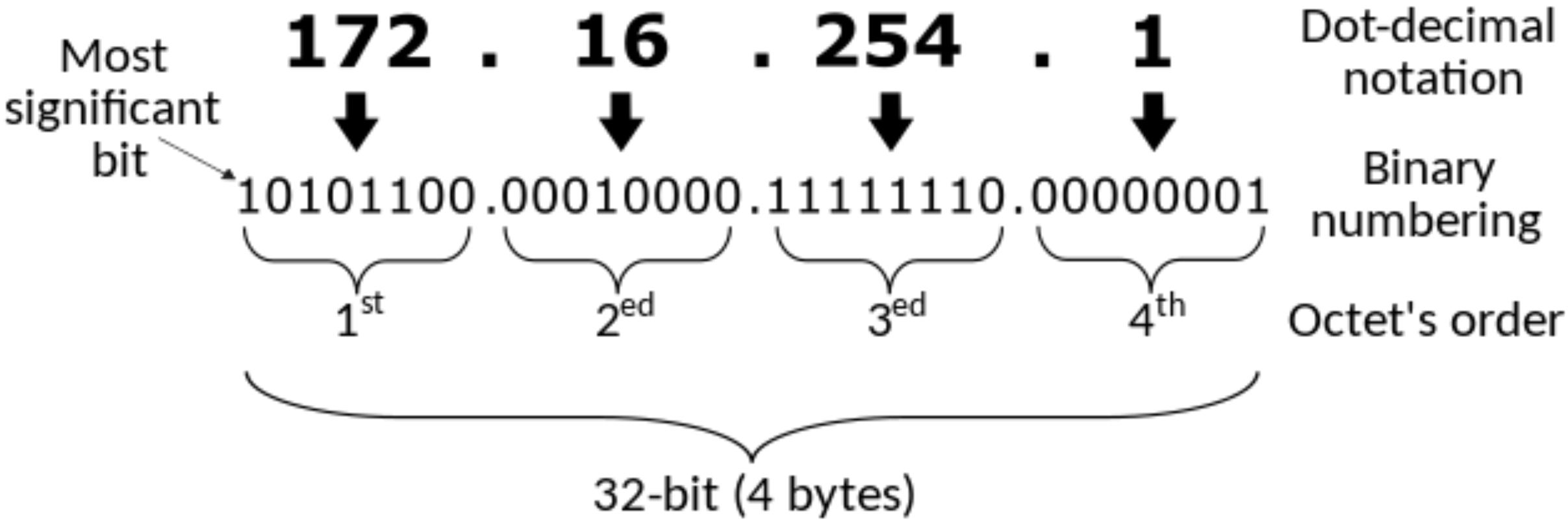
Every host is assigned a unique identifier (“IP Address”)

Every packet has an IP header that indicates its sender and receiver

Routers forward packet along to *try* to get it to the destination host

Rest of the packet should be ignored by the router

# IPv4 Address

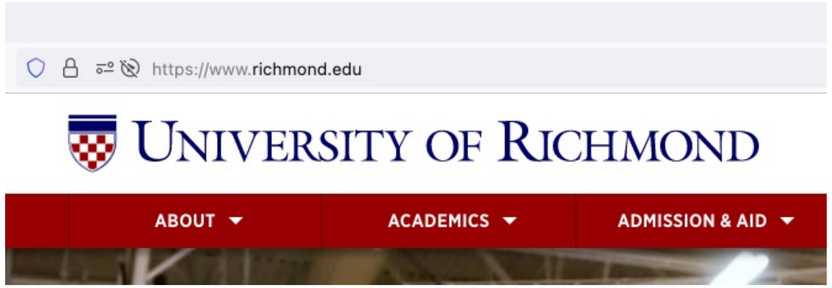


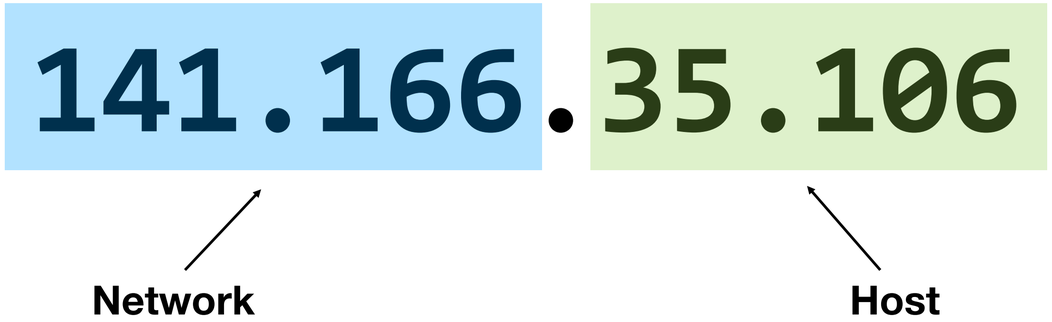
# IPv4 Address

1. The first part of an Internet address identifies the network on which the host resides

2. The second part identifies the particular host on the given network


**richmond.edu**



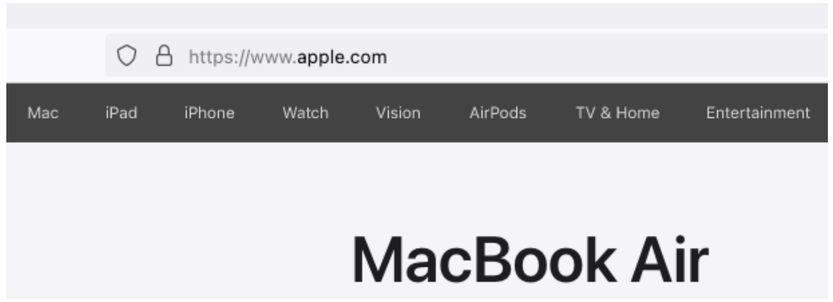



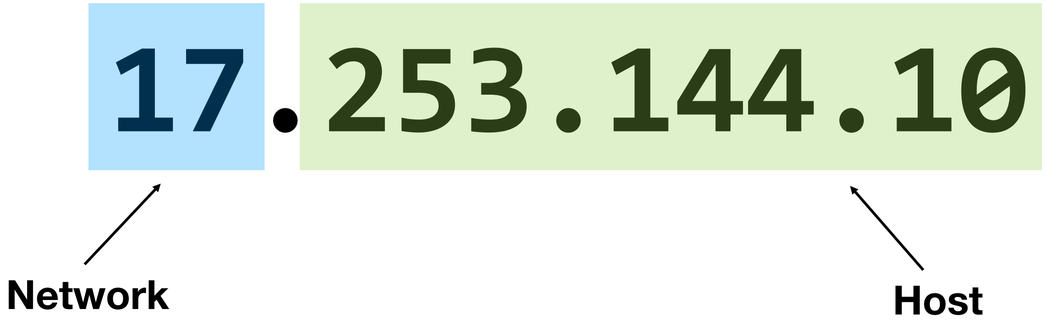
# IPv4 Address

1. The first part of an Internet address identifies the network on which the host resides

2. The second part identifies the particular host on the given network


**apple.com**

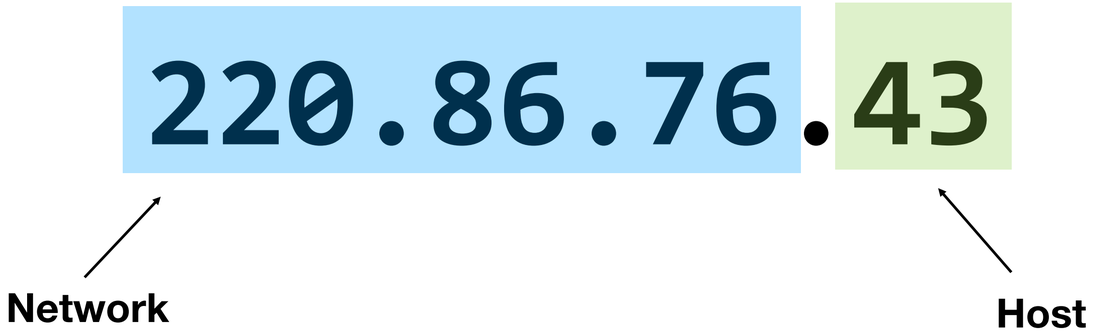


# IPv4 Address

1. The first part of an Internet address identifies the network on which the host resides

2. The second part identifies the particular host on the given network

# IPv4 Address

1. The first part of an Internet address identifies the network on which the host resides

2. The second part identifies the particular host on the given network


## Subnet Mask

**255.255.255.0**

Network number will be this part

**220.86.76.43**

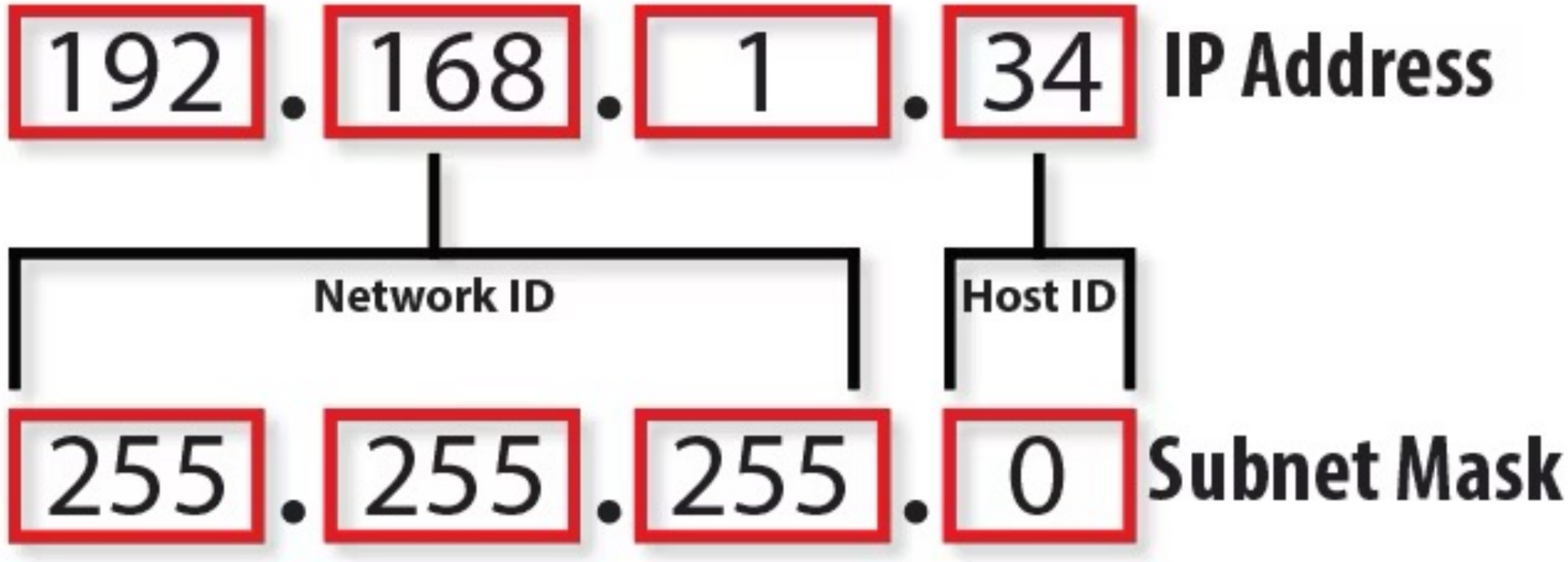
Network

Host

# IPv4 Address

1. The first part of an Internet address identifies the network on which the host resides

2. The second part identifies the particular host on the given network



# IPv4 Classes

| | Public IP Range | Private IP Range | Subnet Mask | # of Networks | # of Hosts per Network |
|---|---|---|---|---|---|
| **Class A** | 1.0.0.0 to 127.0.0.0 | 10.0.0.0 to 10.255.255.255 | 255.0.0.0 | 126 | 16,777,214 |
| **Class B** | 128.0.0.0 to 191.255.0.0 | 172.16.0.0 to 172.31.255.255 | 255.255.0.0 | 16,382 | 65,534 |
| **Class C** | 192.0.0.0 to 223.255.255.0 | 192.168.0.0 to 192.168.255.255 | 255.255.255.0 | 2,097,150 | 254 |

# IPv4 Private Addresses

- Class A Private Range: 10.0.0.0 to 10.255.255.255
- Class B Private Range: 172.16.0.0 to 172.31.255.255
- Class C Private Range: 192.168.0.0 to 192.168.255.255

# MAC Address

- **M**edium **A**ccess **C**ontrol address is a unique identifier assigned to a network interface controller (NIC)

- Assigned by device manufacturers, and are therefore often referred to as the **burned-in address**, or as an **Ethernet hardware address**