Network Security

- Today: background in networking, so we can explore network security for next 3 weeks
  - Speed running a month of networking in one lecture, so I'll focus on aspects that are security-relevant
  - Please ask questions when things are unclear!

Protocols

- A protocol is an agreement on how to communicate
- Includes syntax and semantics
  - How a communication is specified & structured
    - Format, order messages are sent and received
  - What a communication means
    - Actions taken when transmitting, receiving, or timer expires
- E.g.: making a comment in lecture?
  1. Raise your hand.
  2. Wait to be called on.
  3. Or: wait for speaker to **pause** and vocalize
  4. If unrecognized (after timeout): vocalize w/ “excuse me”

What is the goal of the Internet?

- Move data from one location to another
- Analogy: I write a message on a piece of paper. How do I send this message to you?
- Solution: Postal system

Building block 1: something that moves

- Mailman, Pony Express, carrier pigeon, etc.

IP over Avian Carriers

From Wikipedia, the free encyclopedia

In computer networking, **IP over Avian Carriers** (**IPoAC**) is a proposal to carry Internet Protocol (IP) traffic by birds such as homing pigeons. IP over Avian Carriers was initially described in RFC 1149, a Request for Comments (RFC) issued by the Internet Engineering Task Force (IETF), written by D. Waitzman, and released on April 1, 1990. It is one of several April Fools' Day Request for Comments.

Waitzman described an improvement of his protocol in RFC 2549, *IP over Avian Carriers with Quality of Service* (1 April 1999). Later, in RFC 6214—released on 1 April 2011, and 13 years after the introduction of IPv6—Brian Carpenter and Robert Hinden published *Adaptation of RFC 1149 for IPv6*.[1]

IPoAC has been successfully implemented, but for only nine packets of data, with a packet loss ratio of 55% (due to operator error),[2] and a response time ranging from 3,000 seconds (≈50 minutes) to over 6,000 seconds (≈1.77 hours). Thus, this technology suffers from poor latency. Nevertheless, for large transfers, avian carriers are capable of high average throughput when carrying flash memory devices, effectively implementing a sneakernet. During the last 20 years, the information density of storage media and thus the bandwidth of an avian carrier has increased 3 times as fast as the bandwidth of the Internet.[3] IPoAC may achieve bandwidth peaks of orders of magnitude more than the Internet when used with multiple avian carriers in rural areas. For example: If 16 homing pigeons are given eight 512 GB SD cards each, and take an hour to reach their destination, the throughput of the transfer would be 145.6 Gbit/s, excluding transfer to and from the SD cards.

Under RFC 1149, a homing pigeon (exemplar in Scheßlitz) can carry Internet Protocol traffic.

Are pigeons faster than the Internet?

Building block 1: something that moves

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

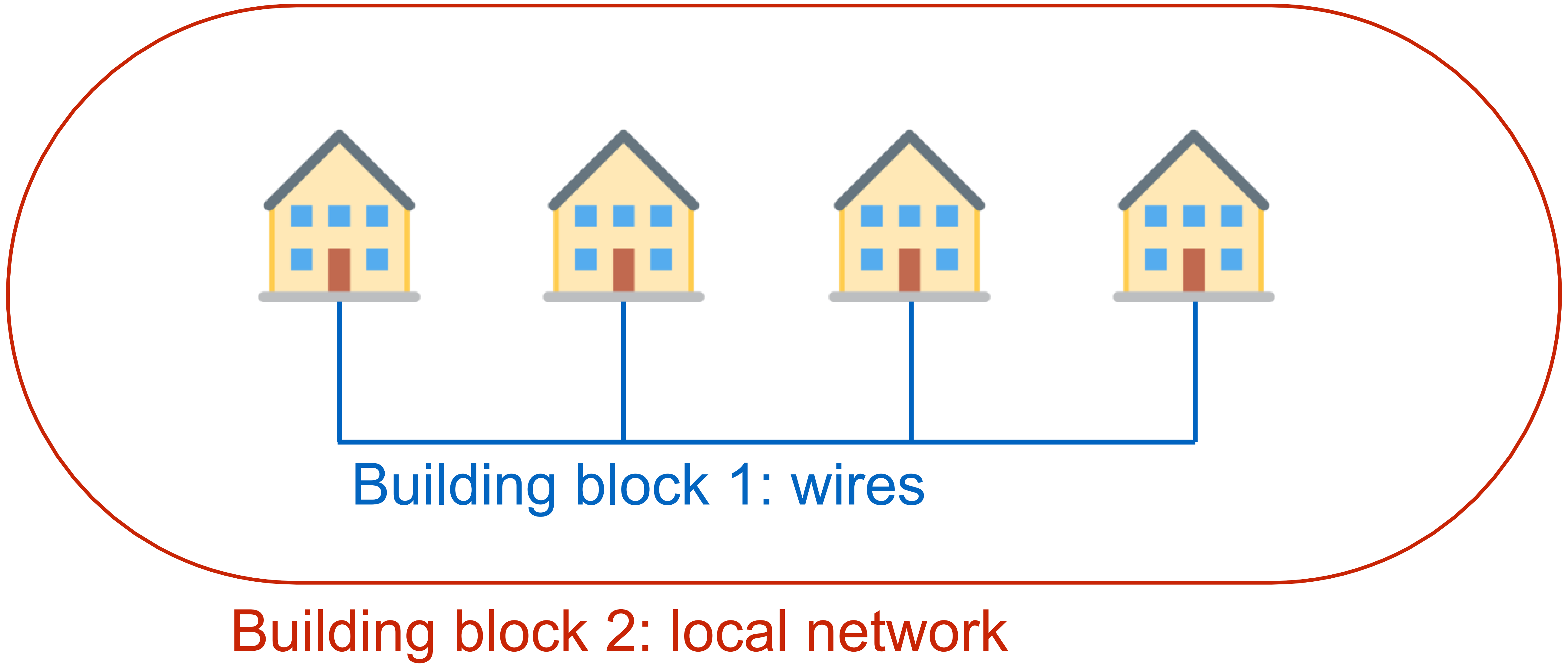
Risks [edit]

Although collisions are unlikely, packets can be lost, particularly to raptors.

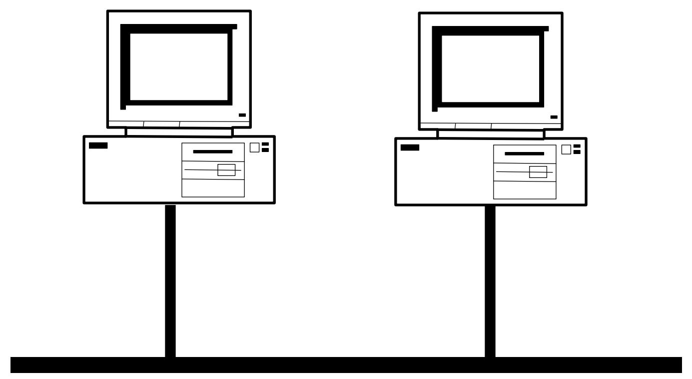
An example of packet loss.

Building block 2: talking to the apartment complex

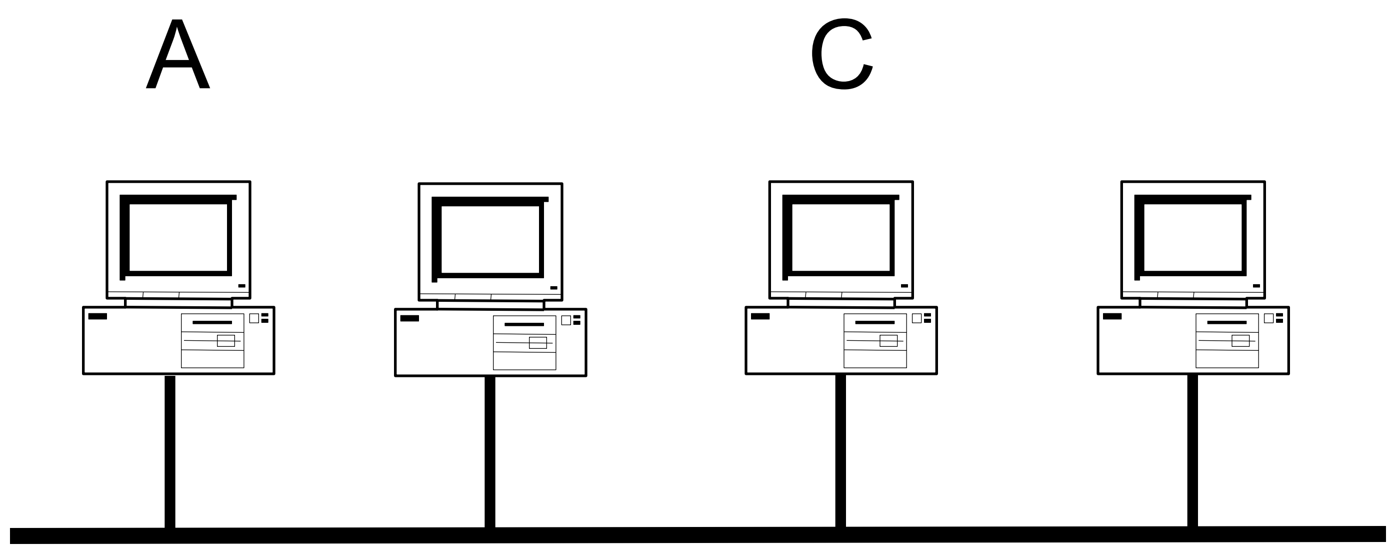
- Using building block 1, we can link up people within a local apartment complex
- Forms a **local area network (LAN)**



Local-Area Networks



point-to-point



shared

How does computer A send a message to computer C?

Local-Area Networks (LAN): Packets

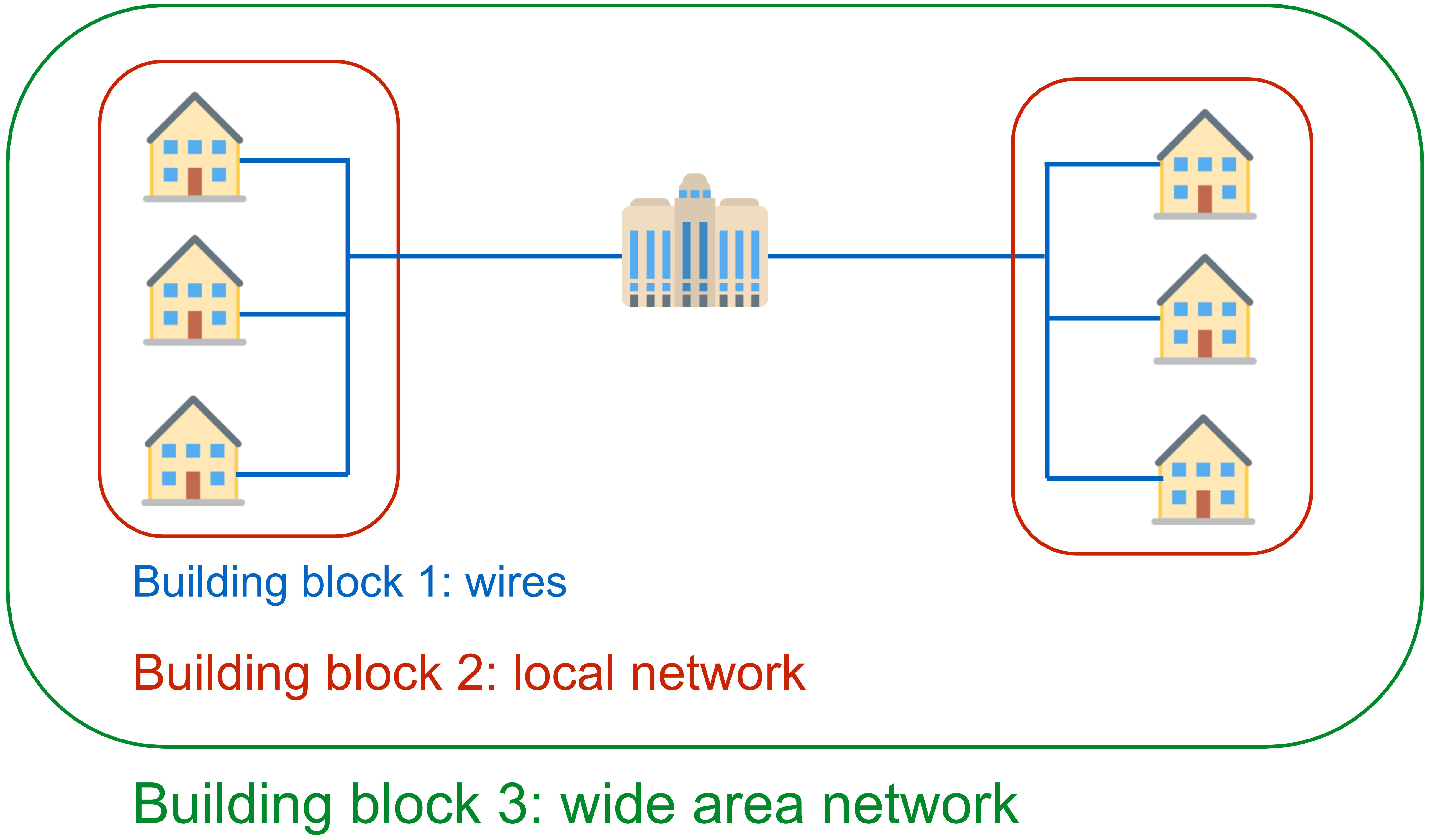

```
Source: A
Destination: C
Message: Hello world!
```

A	C	Hello world!

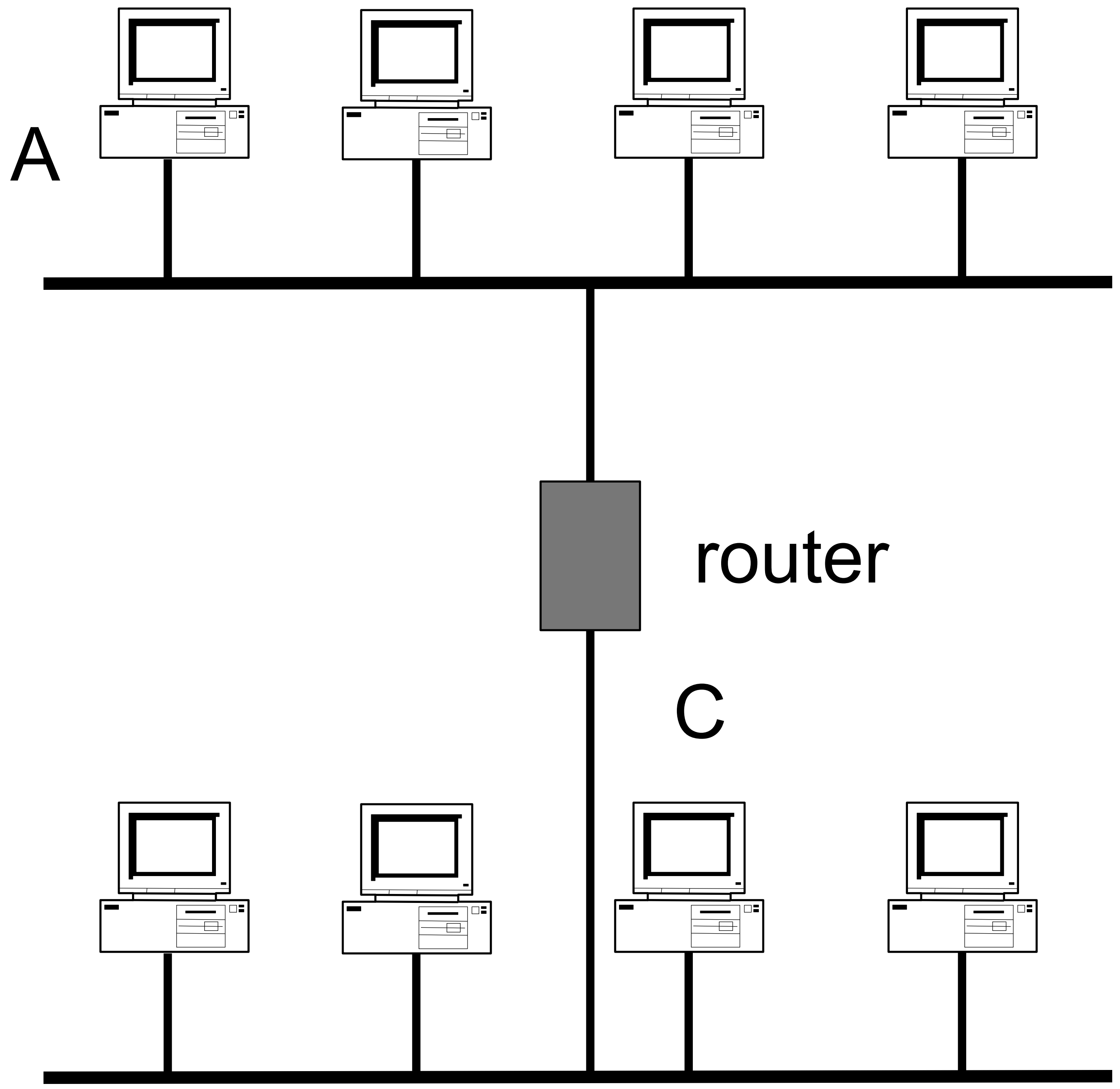
A	C
Hello world!	

Building block 3: Post offices

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

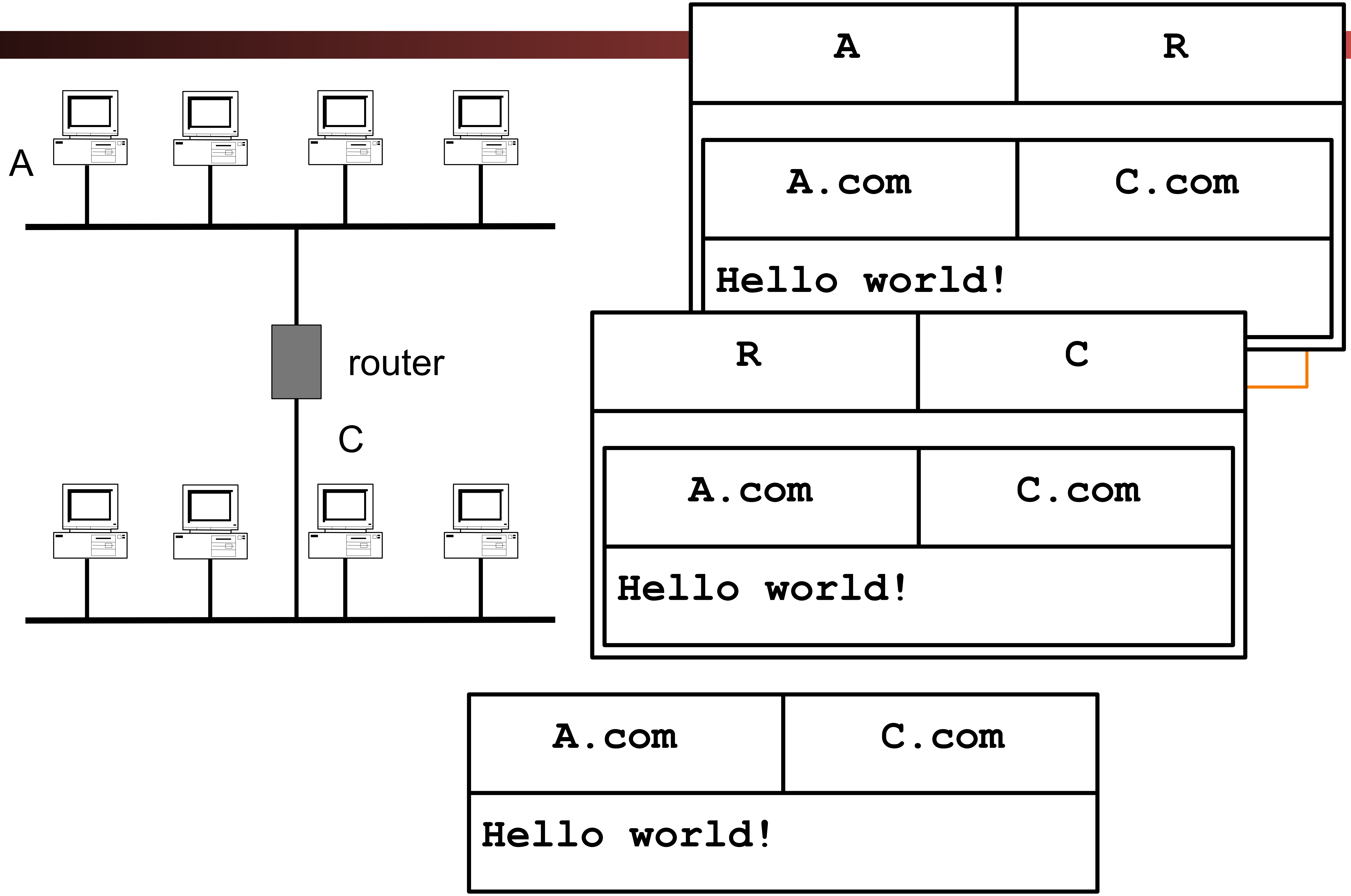


Wide-Area Networks



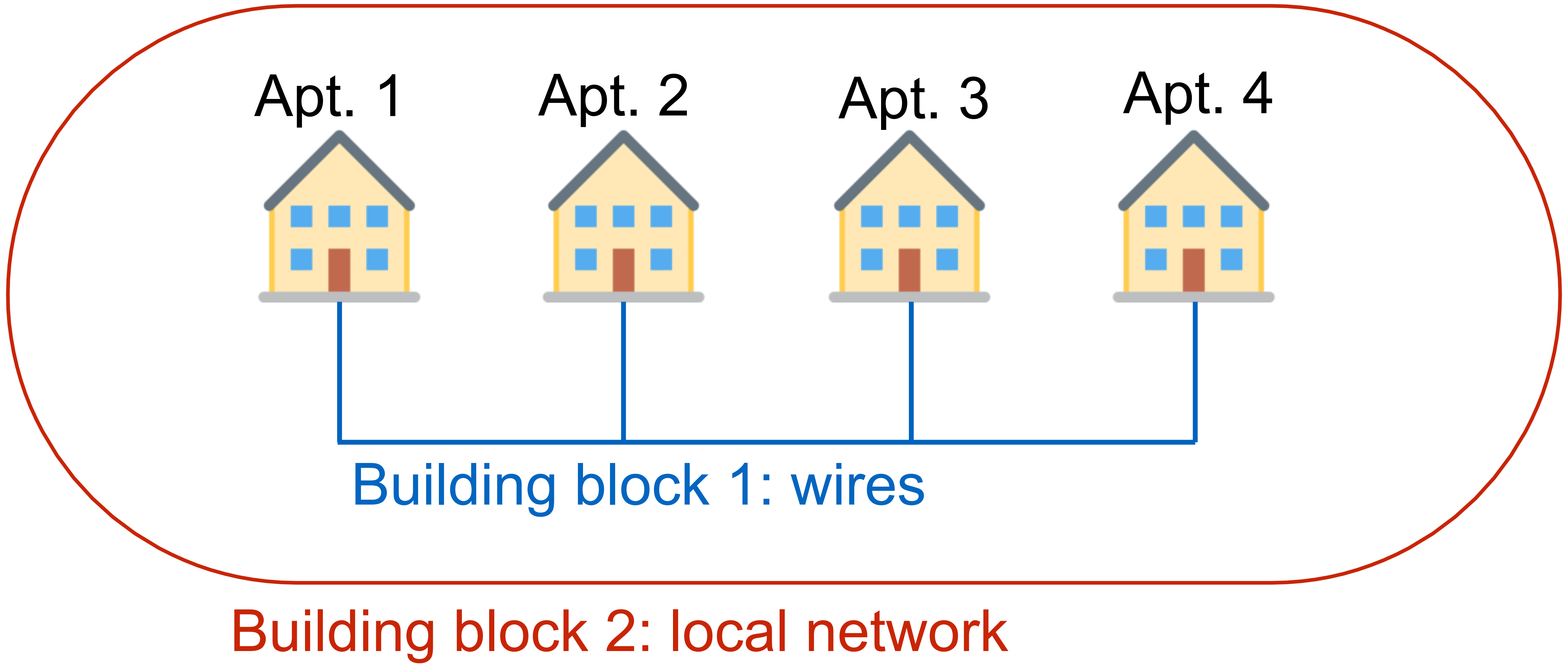
How do we connect two LANs?

Wide-Area Networks



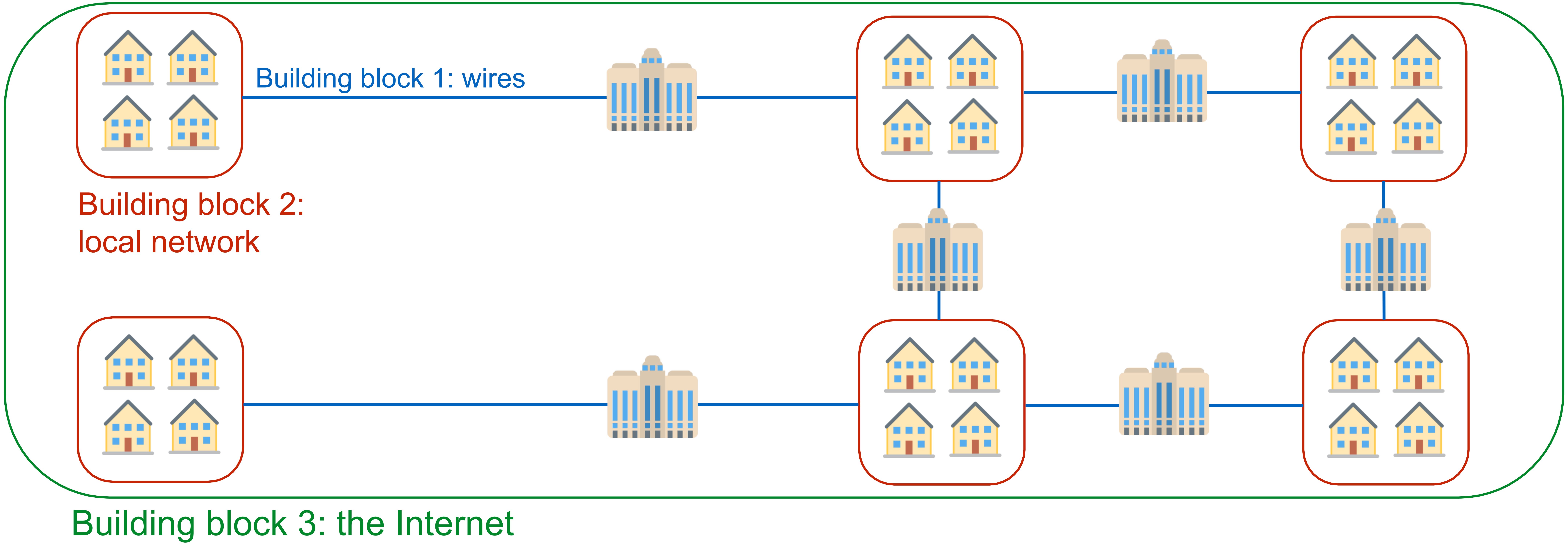
MAC Addresses

- Machines on LANs have unique **MAC Addresses**
- Not to be confused with MAC (message authentication code) from crypto
- Like apartment numbers: useless for global addressing!



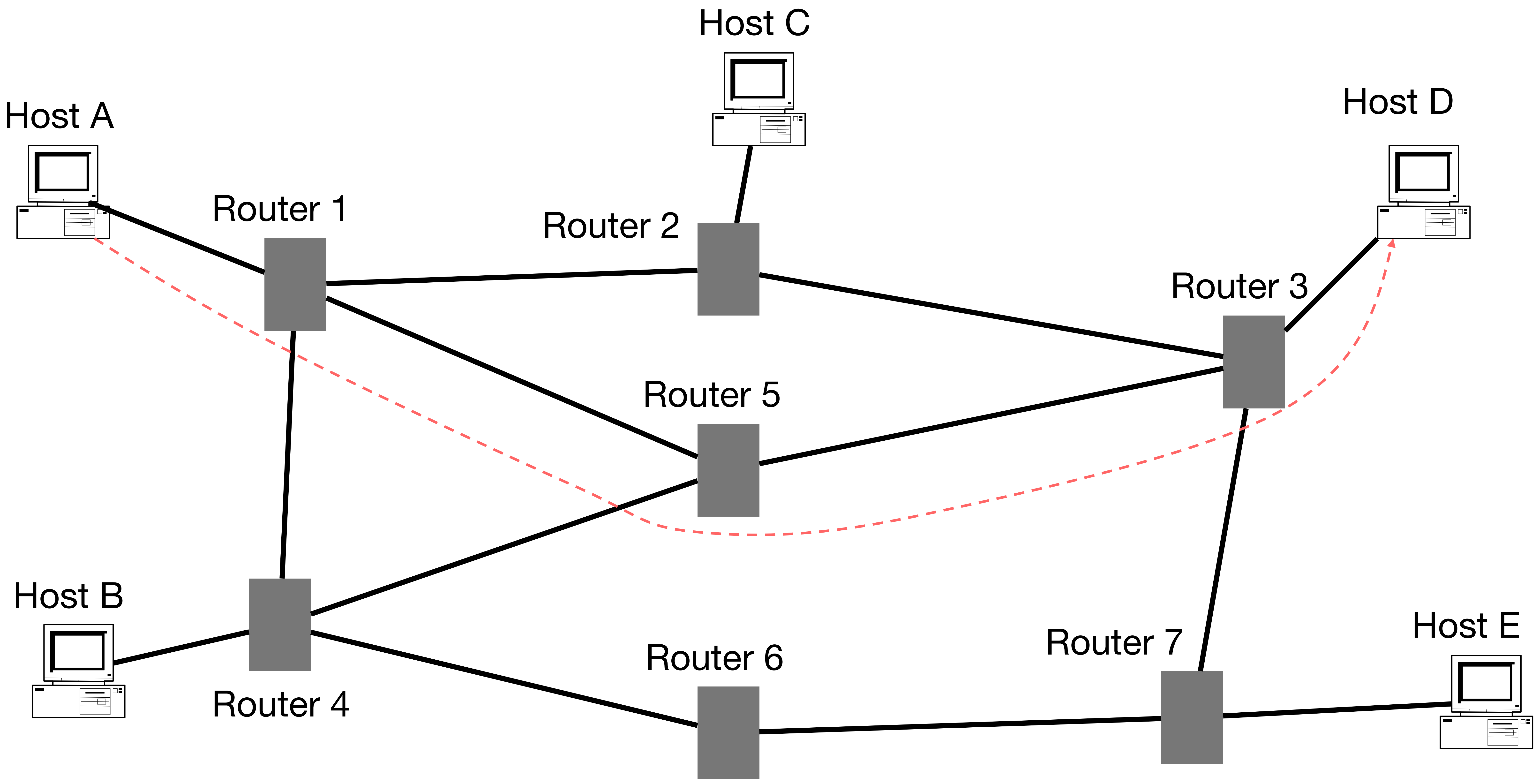
Building block 3: The Internet

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



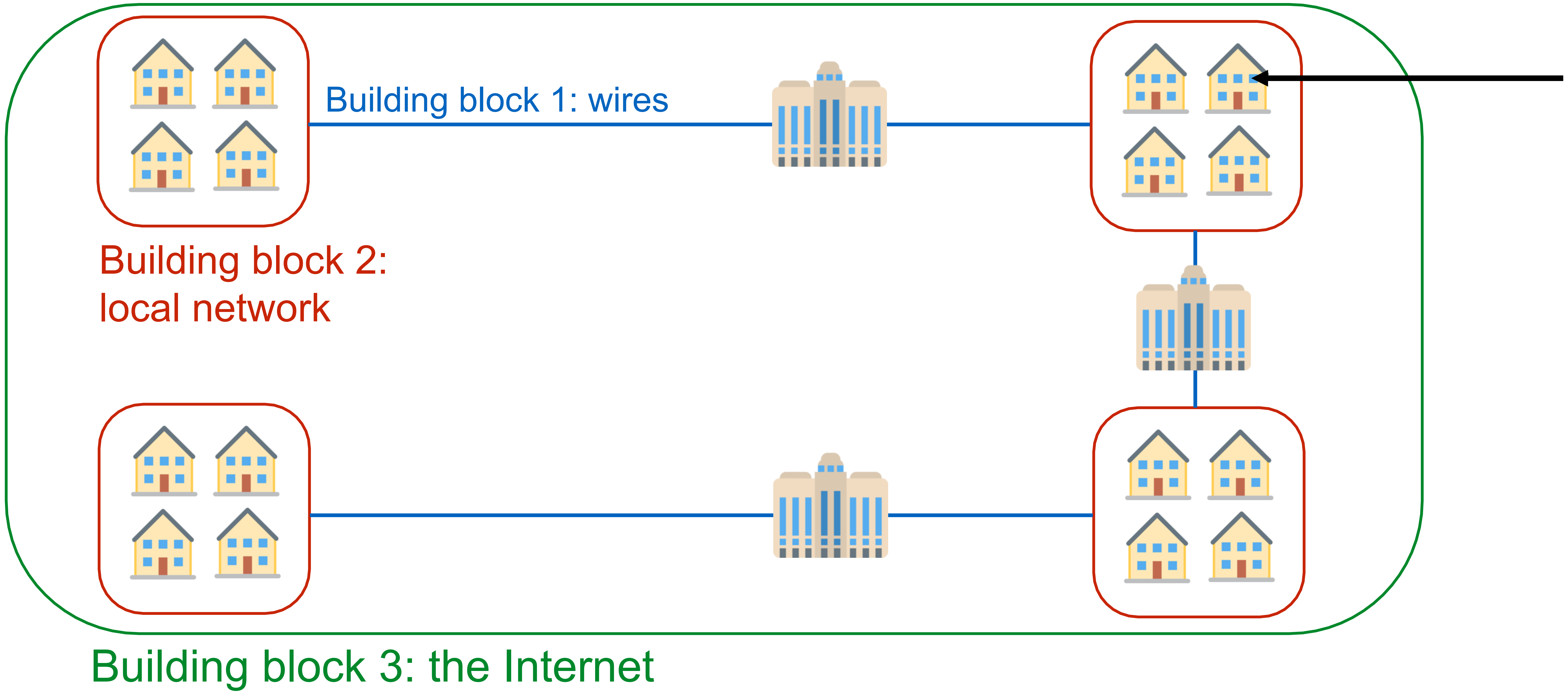
Hop-By-Hop vs. End-to-End Layers

Host A communicates with Host D



IP Addresses

- Global addressing – each IP is unique in the entire world
- Not to be confused with MAC addresses (local addressing)



This apartment has IP address 1:2:3:4. No other apartment in the world has this IP address.

It also has a MAC address, which is only useful for addressing it within the local network (red box).

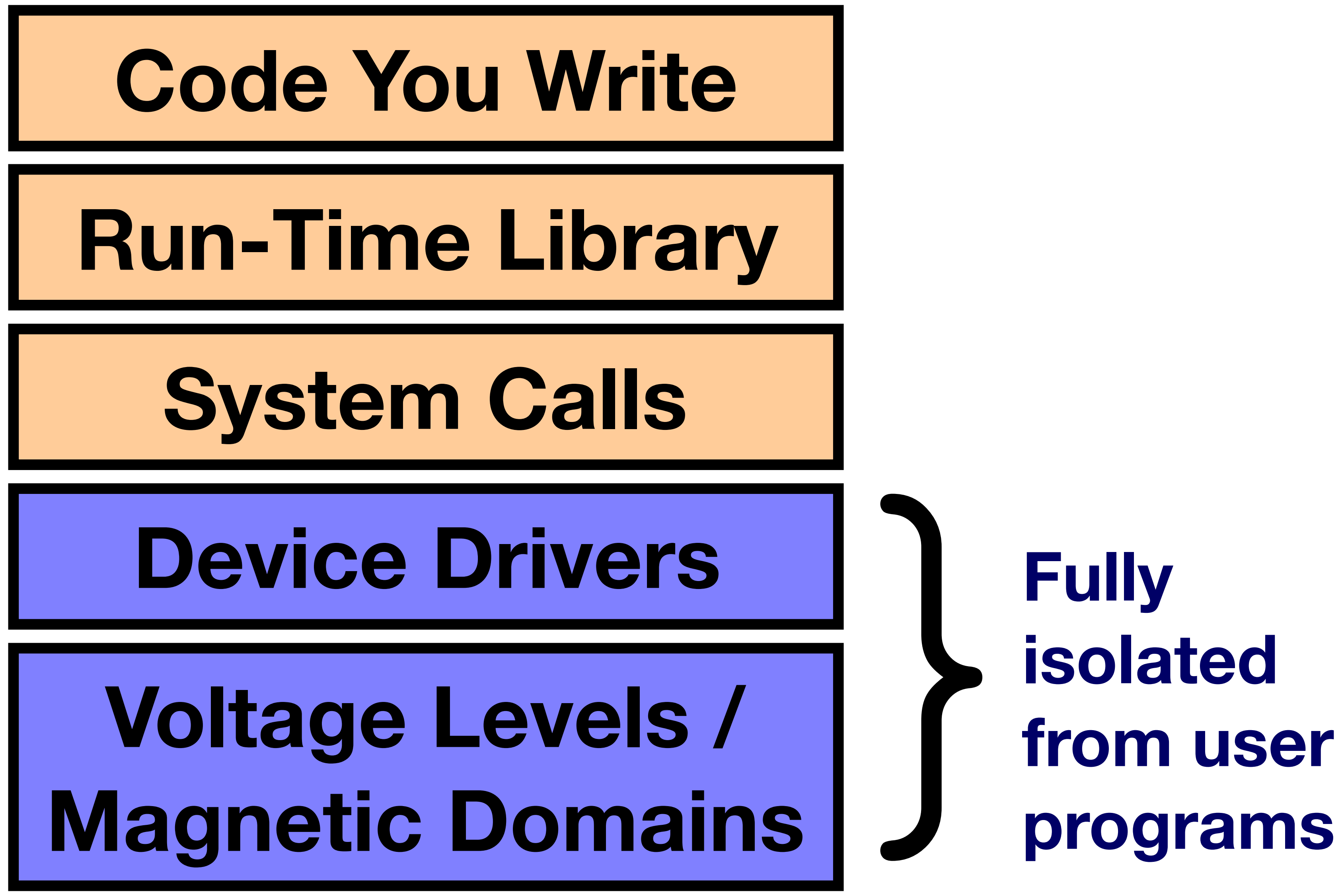
Layers of abstraction

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

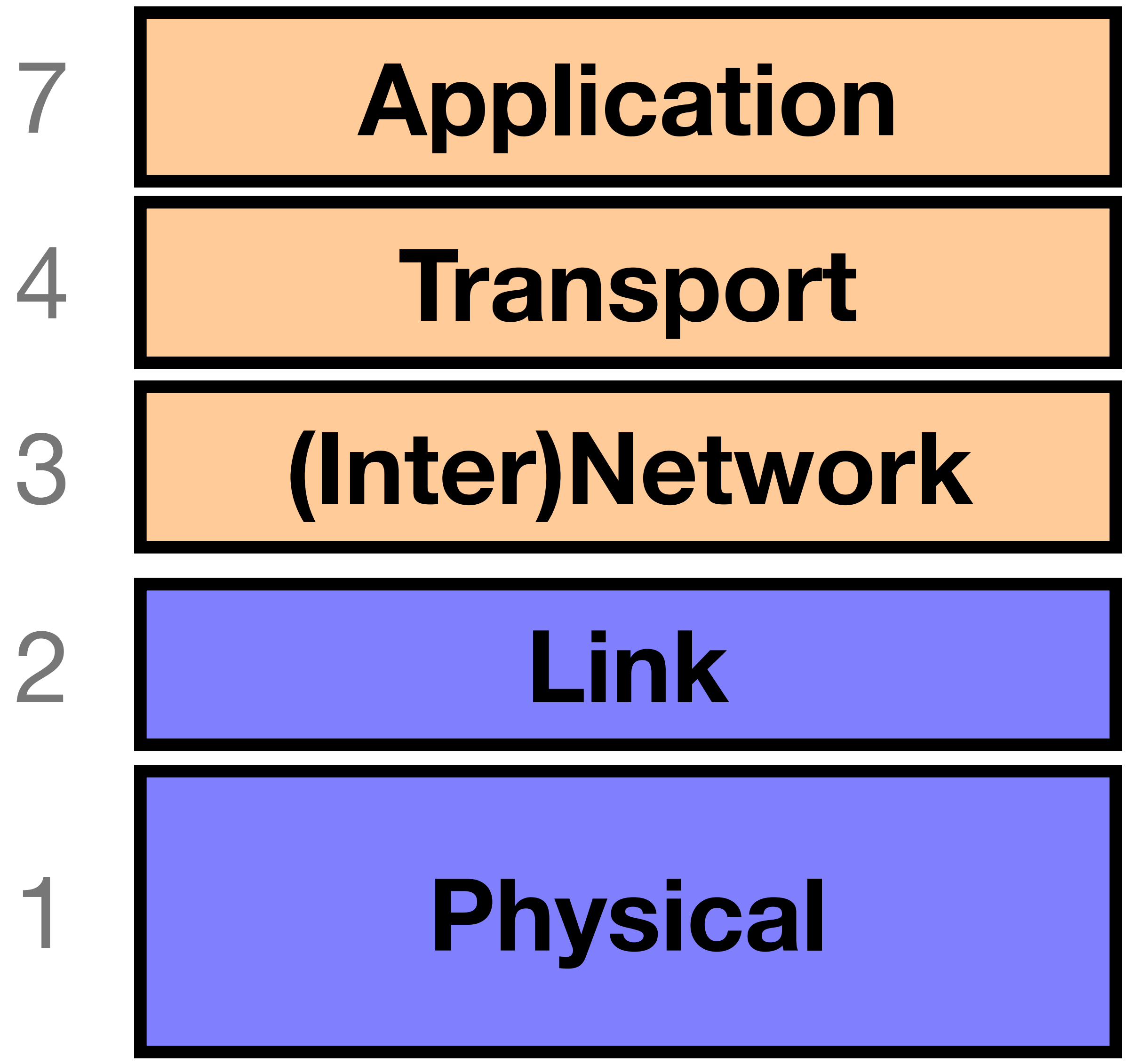
- A change in layer 1 implementation (wireless instead of wires) doesn't affect the other layers
- A change in layer 2 protocols doesn't affect the other layers

Layering

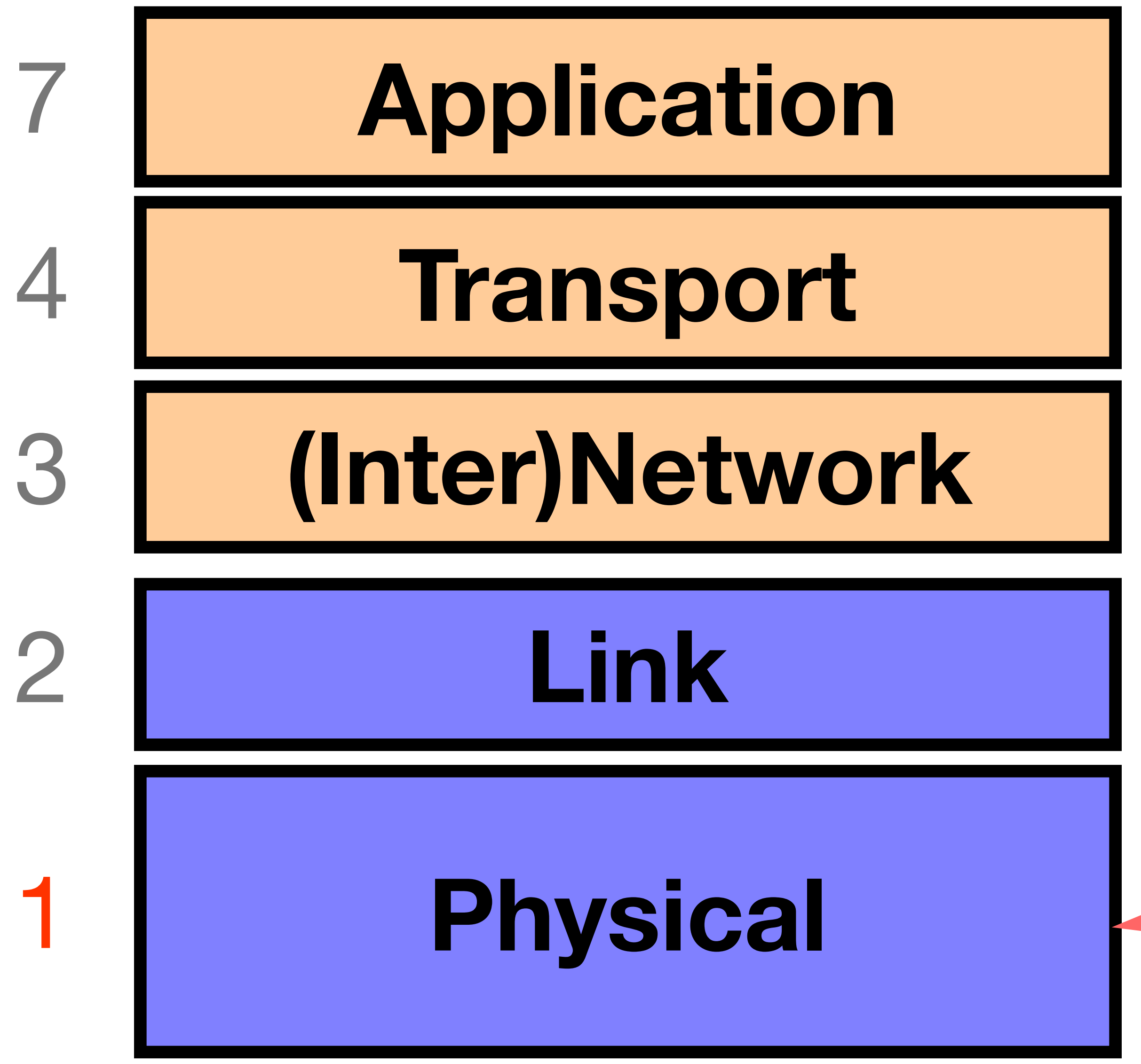
- Internet design is partitioned into layers
  - Each layer relies on services provided by next layer below …
  - … and provides services to layer above it
- Analogy:
  - Consider structure of an application you've written and the "services" each layer relies on / provides



Internet Layering (“Protocol Stack”)



Layer 1: Physical Layer

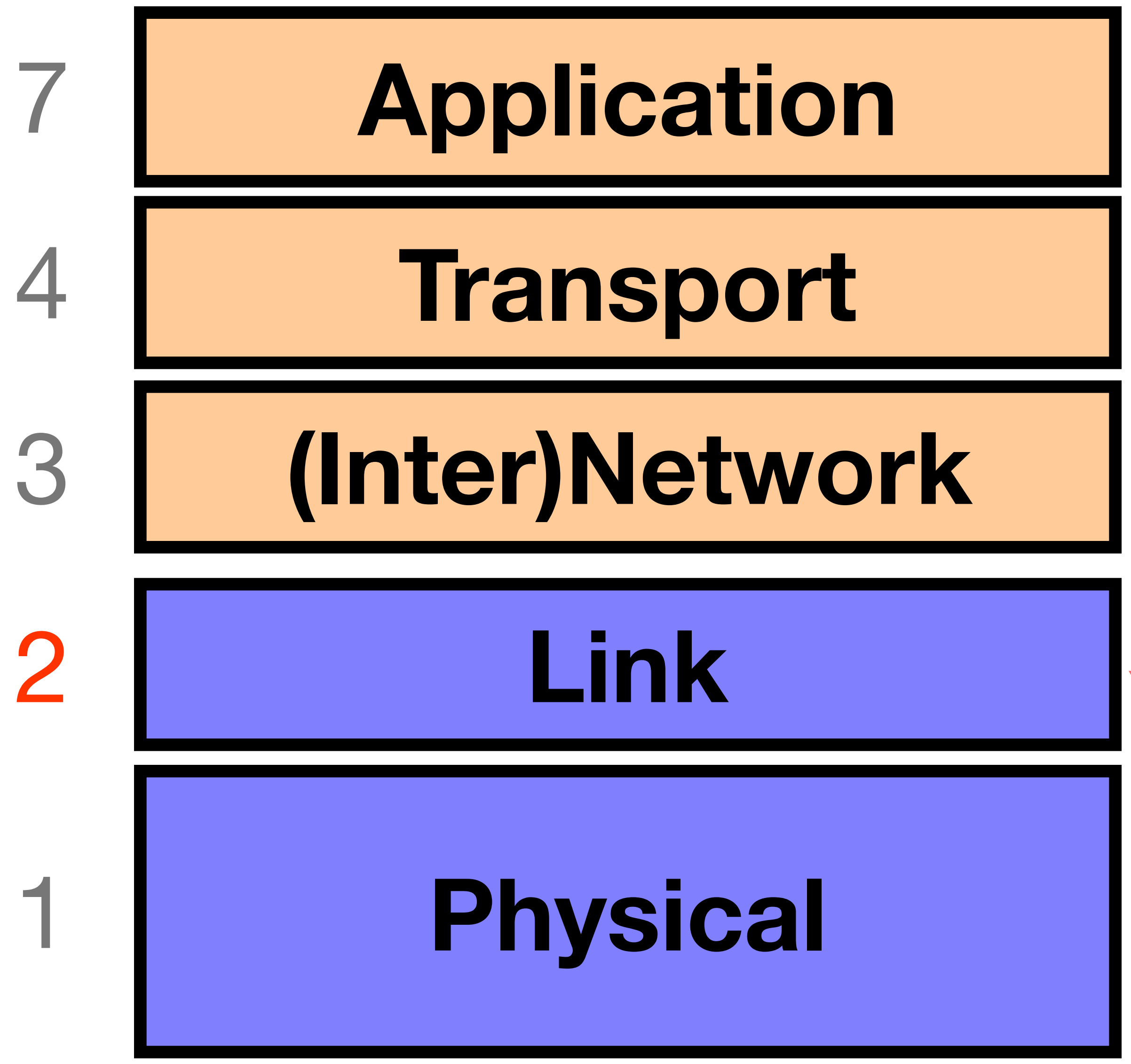


Encoding bits to send them over a single **physical link**

e.g. patterns of

voltage levels /
photon intensities /
RF modulation

Layer 2: Link Layer

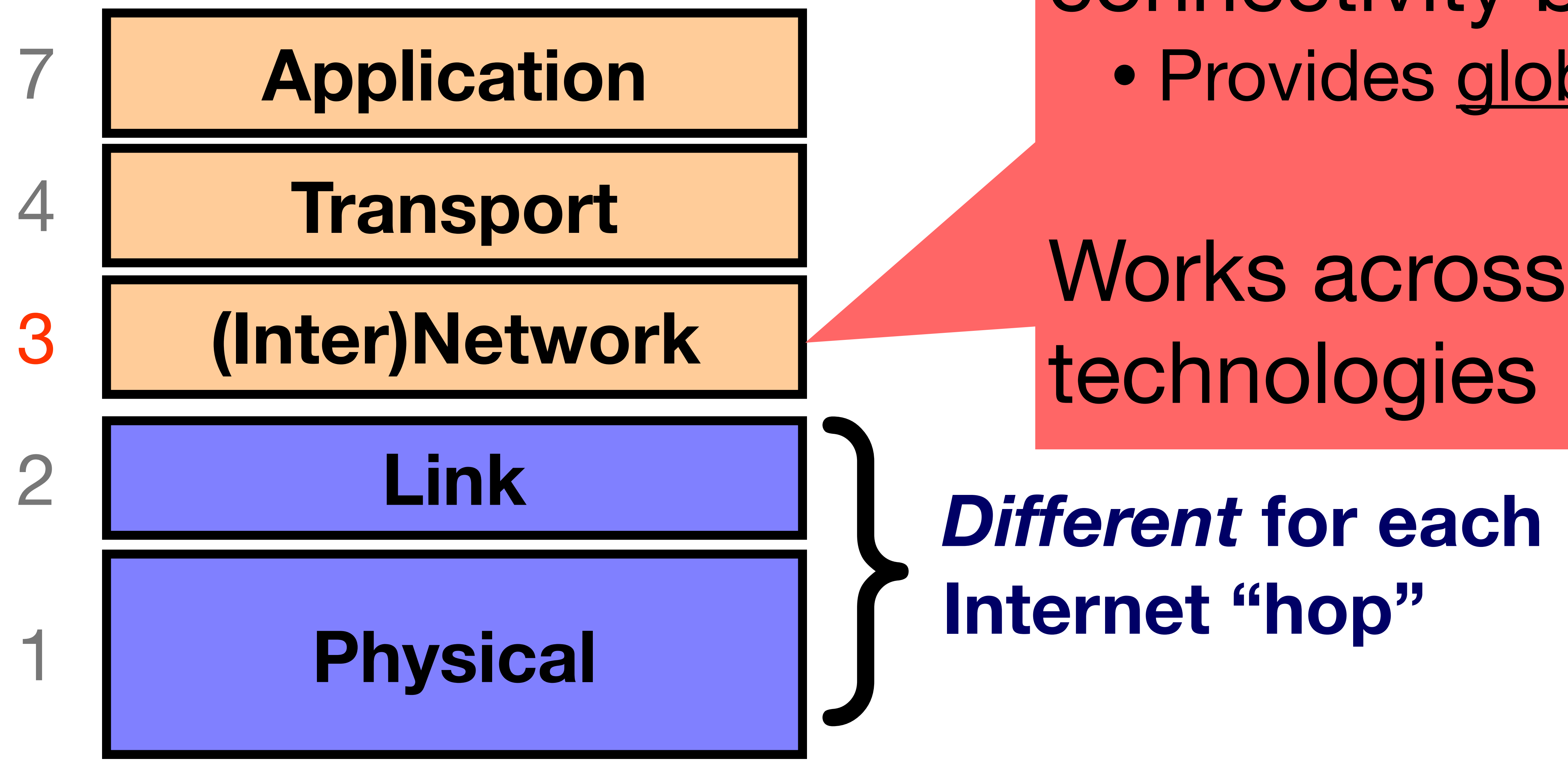


Framing and transmission of a collection of bits into individual messages sent across a single “subnetwork” (one physical technology)

Might involve multiple *physical links* (e.g., modern Ethernet)

Often technology supports broadcast transmission (**every** “node” connected to subnet receives)

Layer 3: (Inter)Network Layer *(IP)*



Bridges multiple "subnets" to provide *end-to-end* internet connectivity between nodes

- Provides global addressing

Works across different link technologies

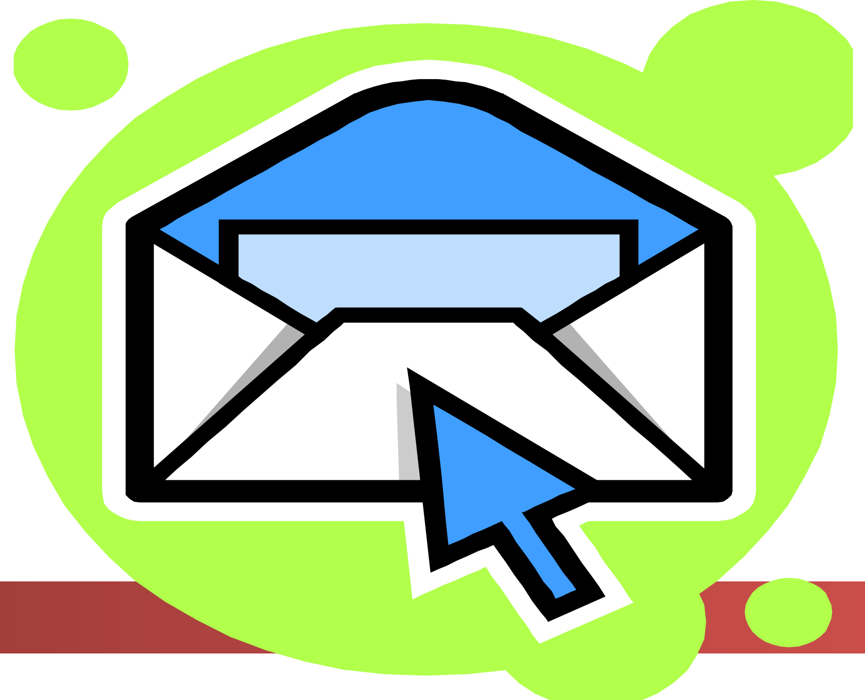
Packets and The Network

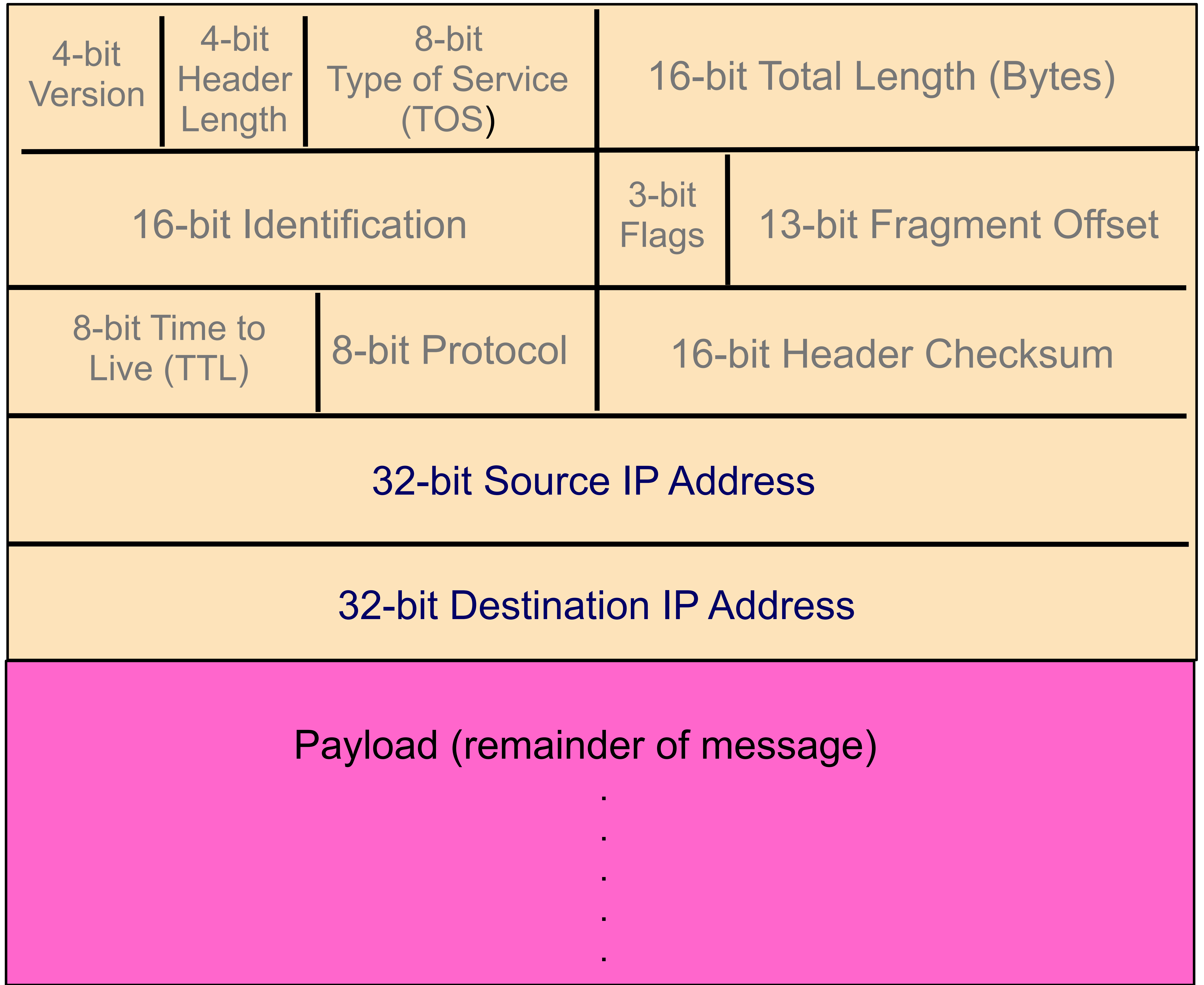
- Modern networks break communications up into packets
  - For our purposes, packets contain a variable amount of data up to a maximum specified by the particular network
- The sending computer breaks up the message and the receiving computer puts it back together
  - So the software doesn't actually see the packets per-se
  - Network itself is ***packet switched***: sending each packet on towards its next destination

Reliability

- Packets are received ***correctly*** or not at all, if ***random*** errors occur
  - Packets have a checksum
  - No guarantees if adversary modifies packets (no cryptographic MACs)
- Packets may be ***unreliable*** and “dropped”
  - It’s up to higher-level protocols to make the connection reliable

Self-Contained IP Packet Format

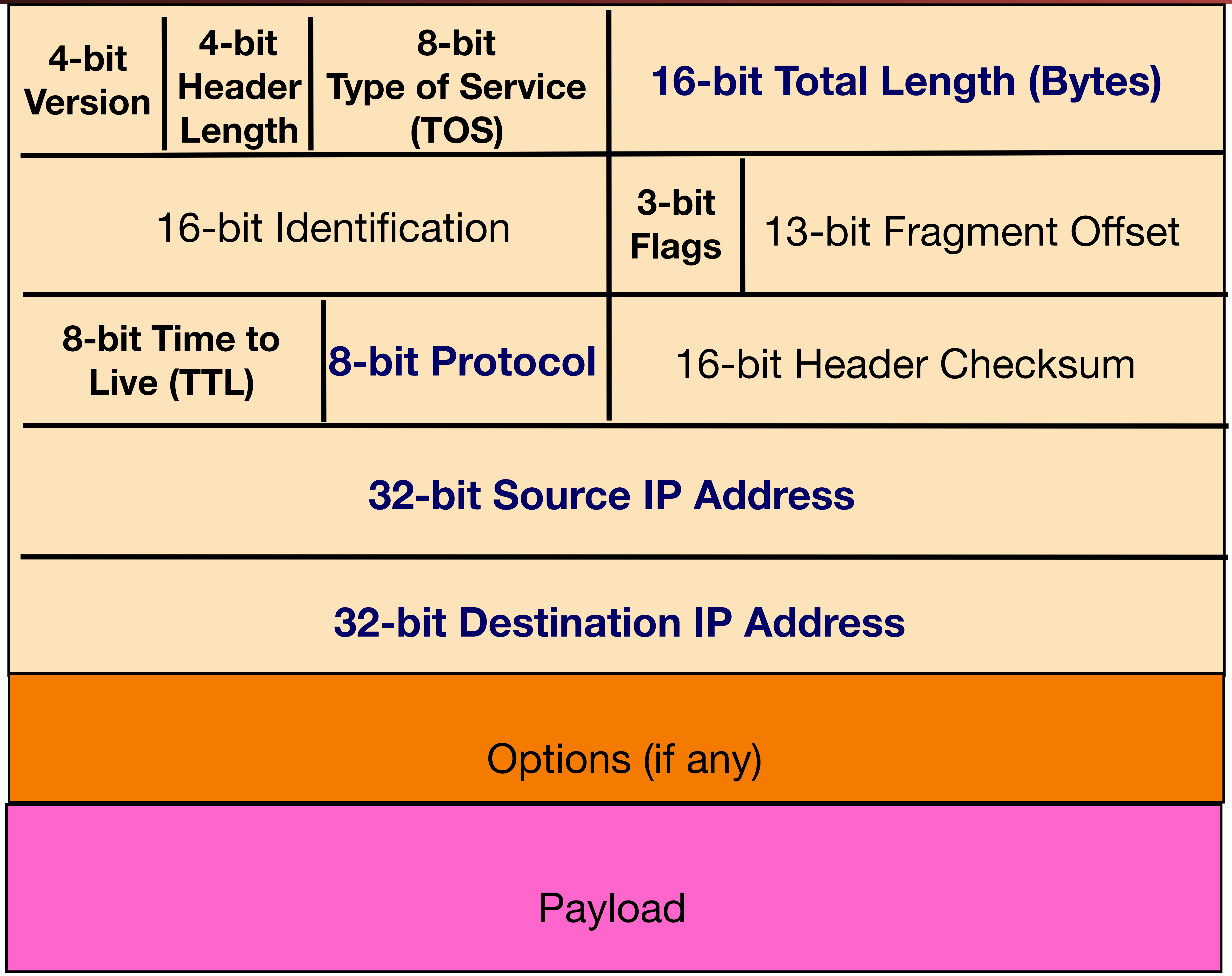




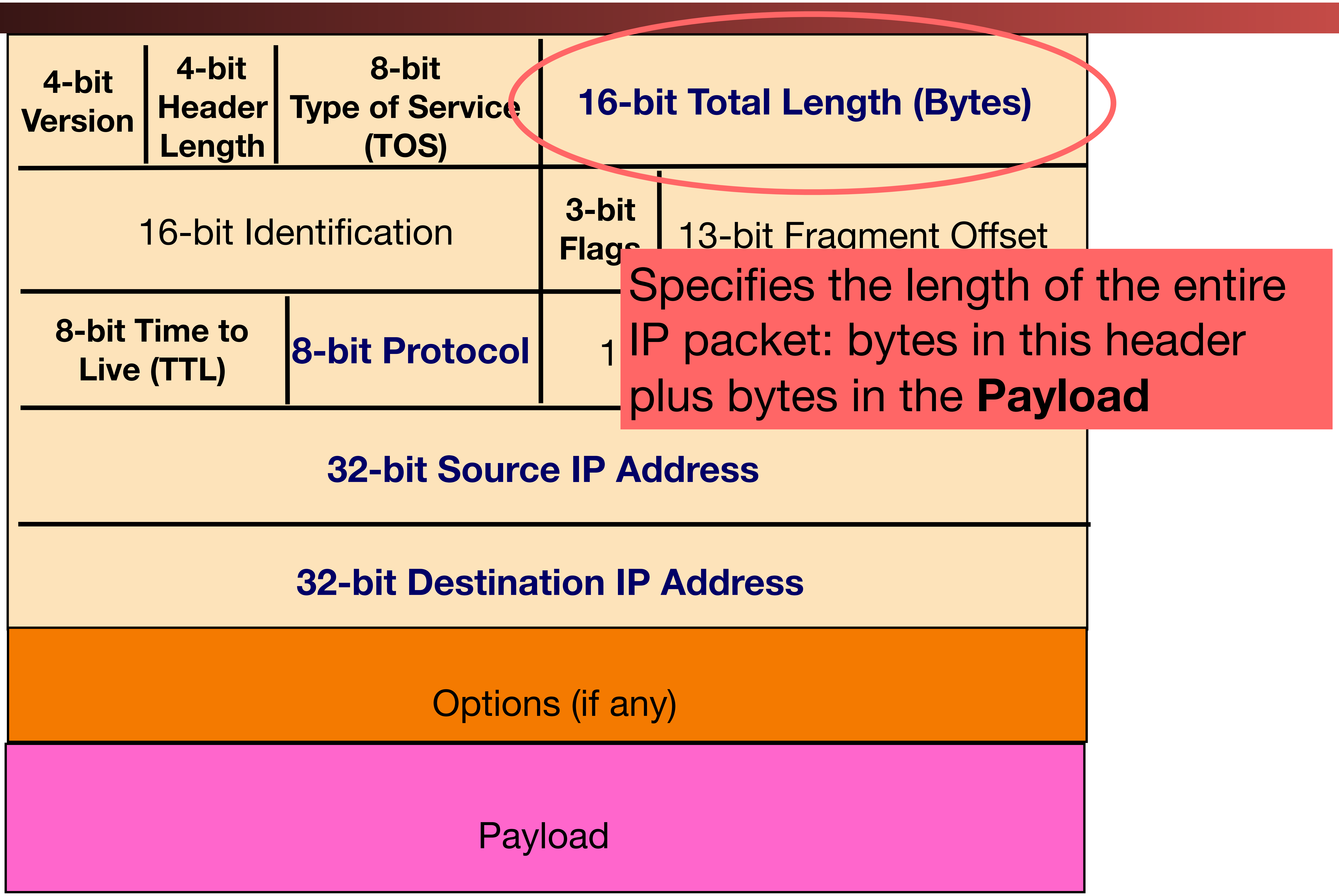
IP = Internet *Protocol*

***Header* is like a letter envelope: contains all info needed for delivery**

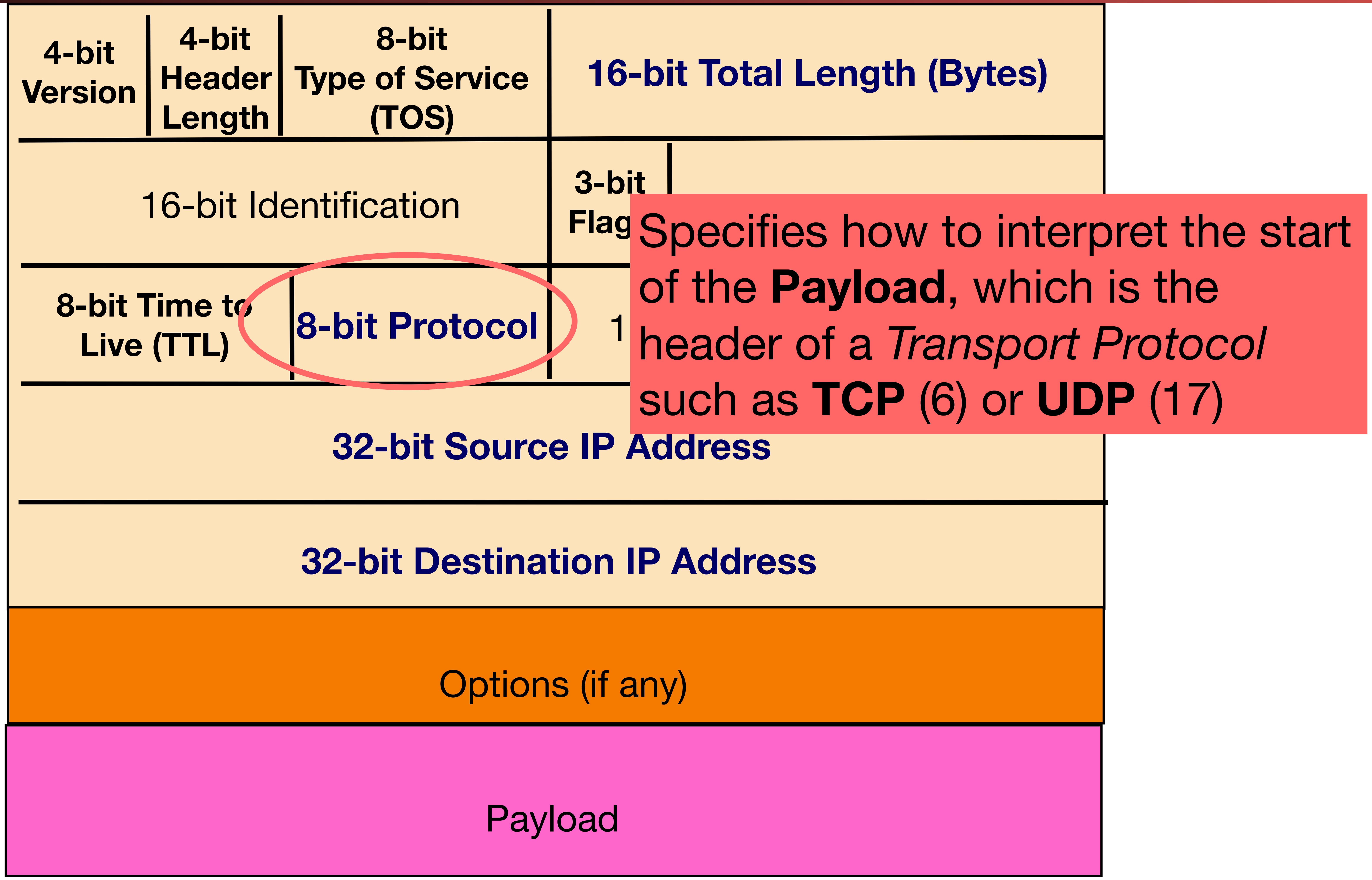
IPv4 Packet Structure (IP version 6 is different)



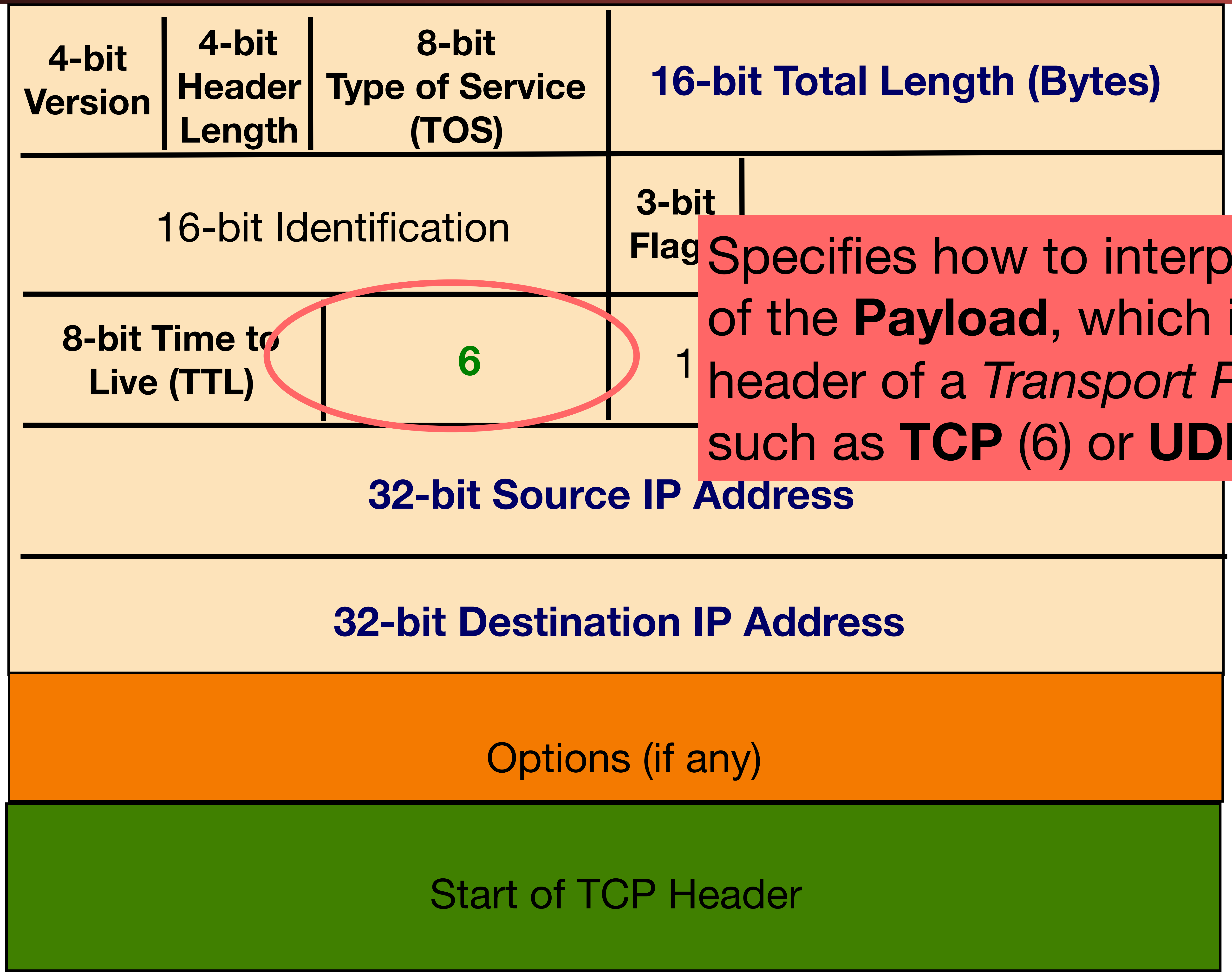
IP Packet Structure



IP Packet Structure

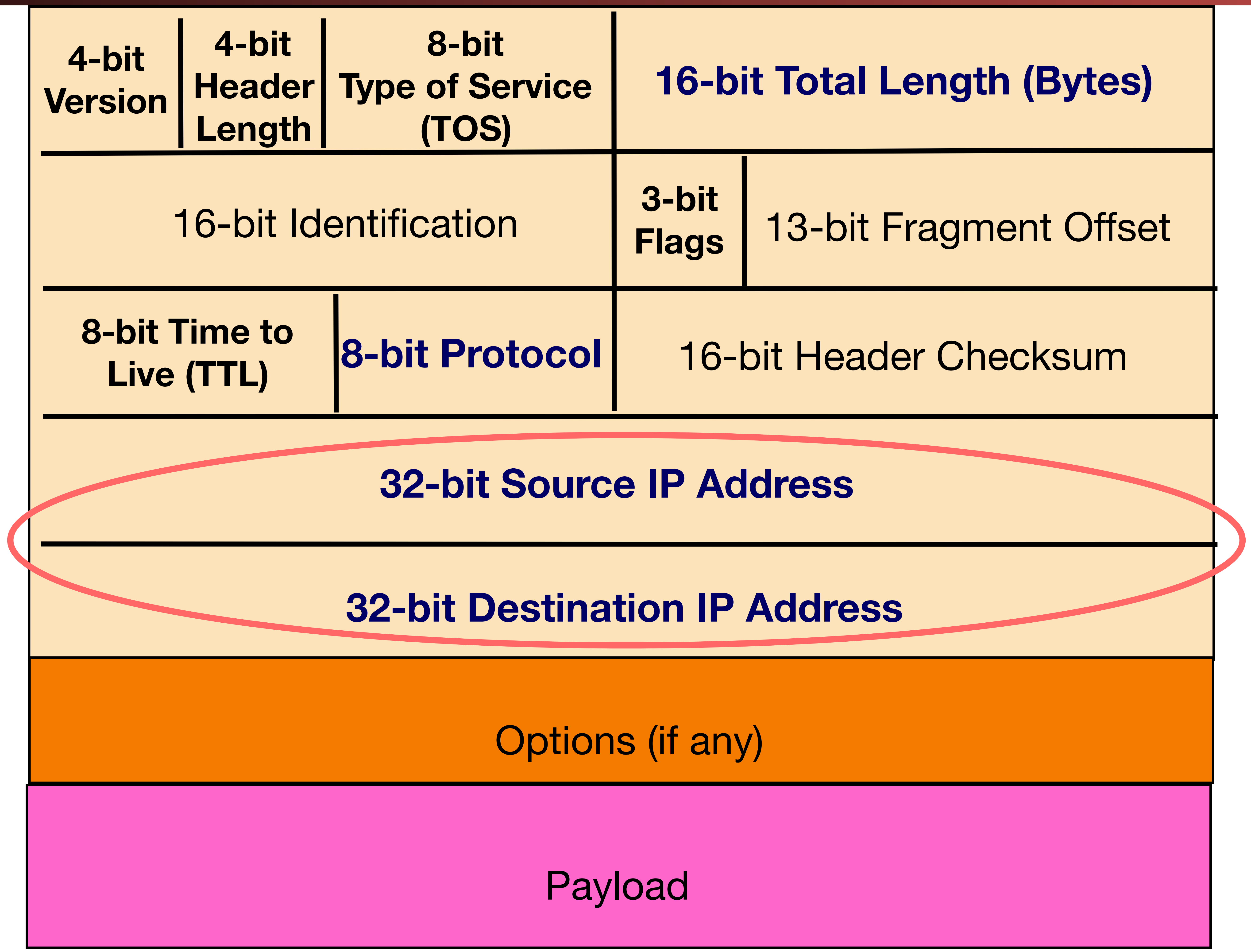


IP Packet Structure



Specifies how to interpret the start of the **Payload**, which is the header of a *Transport Protocol* such as **TCP** (6) or **UDP** (17)

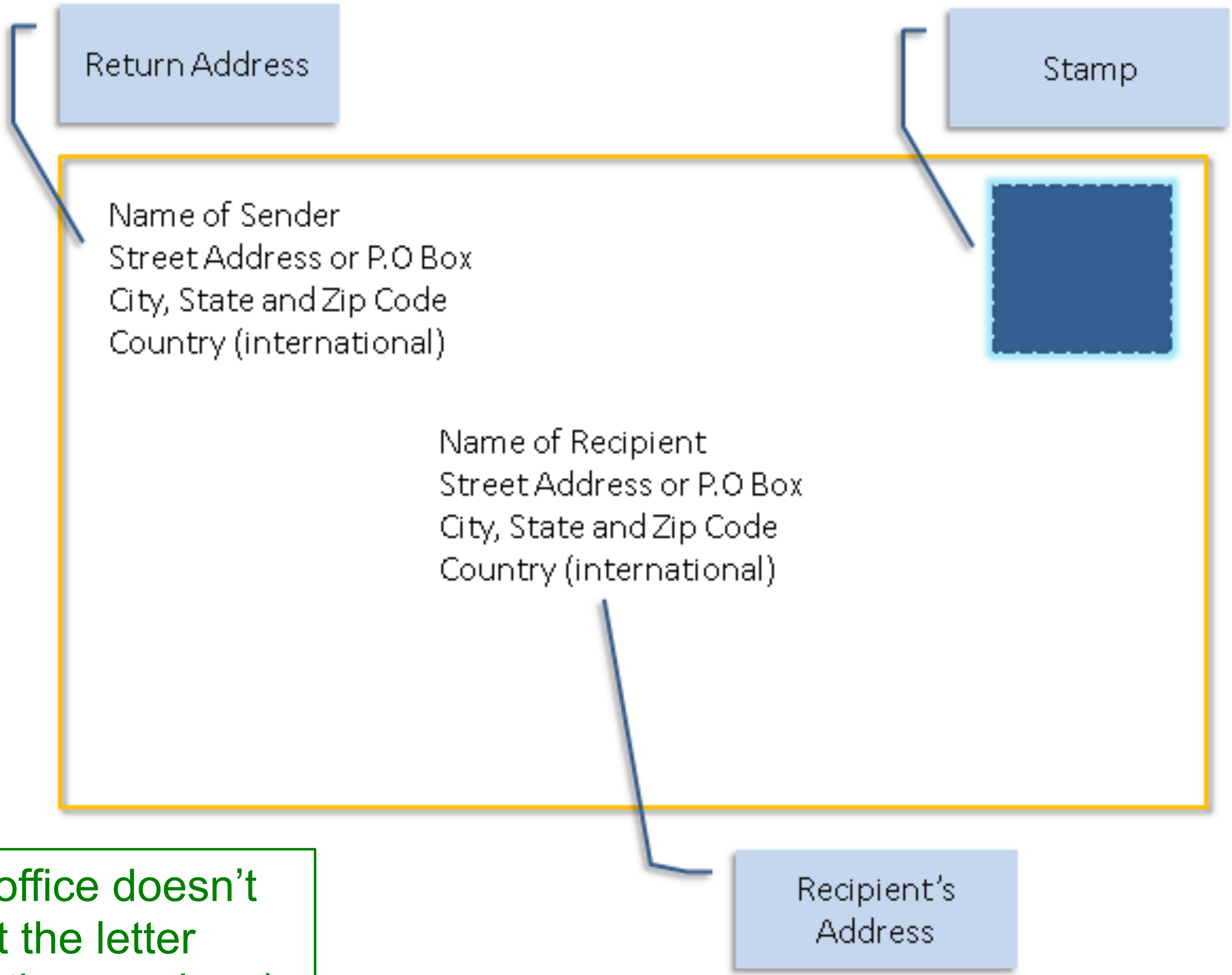
IP Packet Structure



IP Packet Header - IP addresses

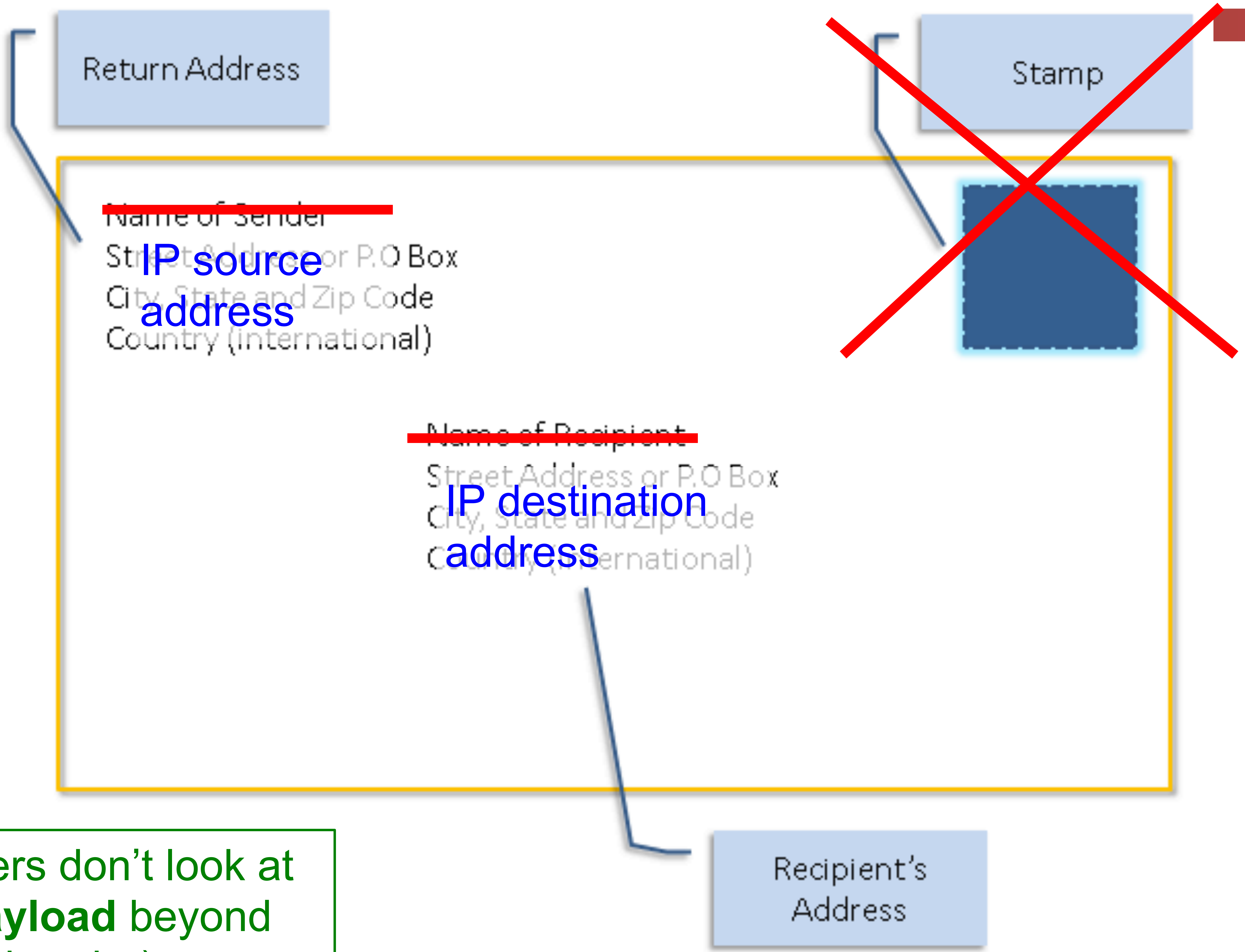
- Source address (32 bits)
  - Unique identifier/locator for the sending host
  - Recipient can decide whether to accept packet
  - Enables recipient to send reply back to source
- Destination address (32 bits)
  - Unique identifier/locator for the receiving host
  - Allows each node to make forwarding decisions

Postal Envelopes:



(Post office doesn't look at the letter inside the envelope)

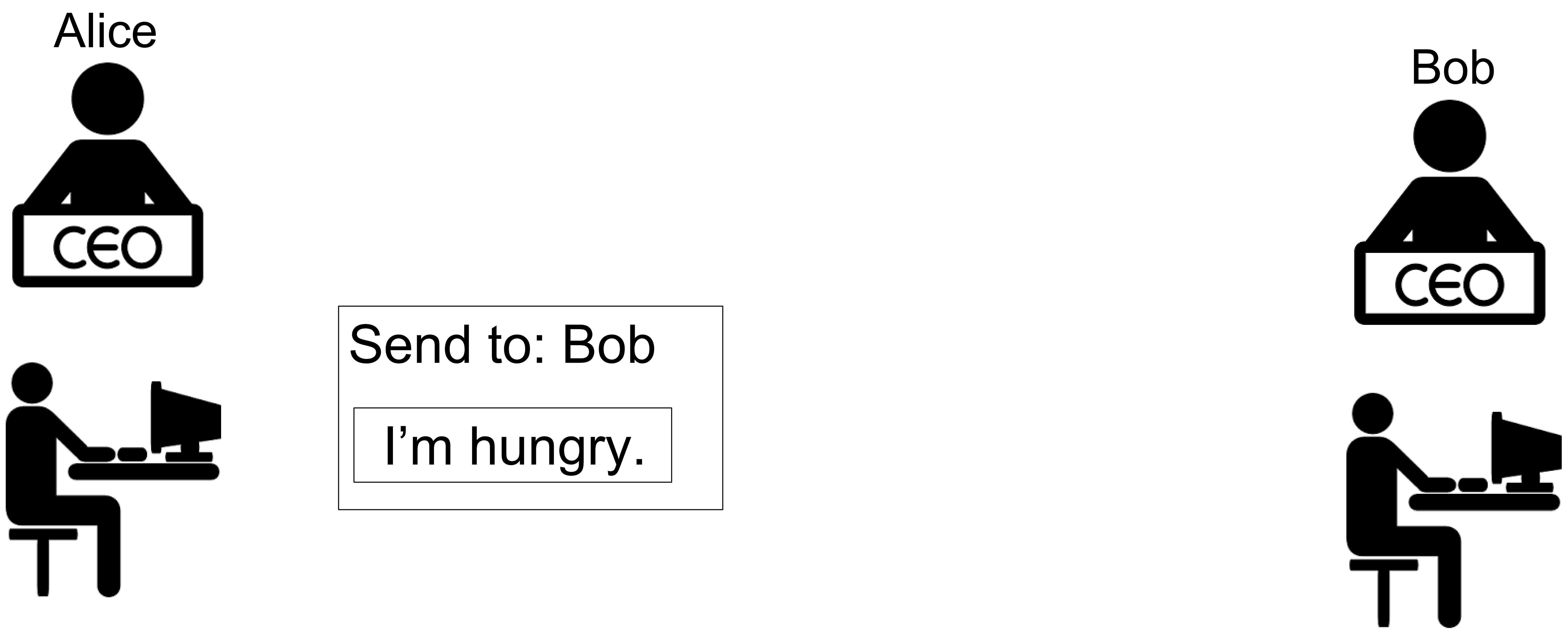
Analogy of IP to Postal Envelopes:



(Routers don't look at the **payload** beyond the IP header)

Layers of abstraction

Layers of abstraction



Layers of abstraction

Alice

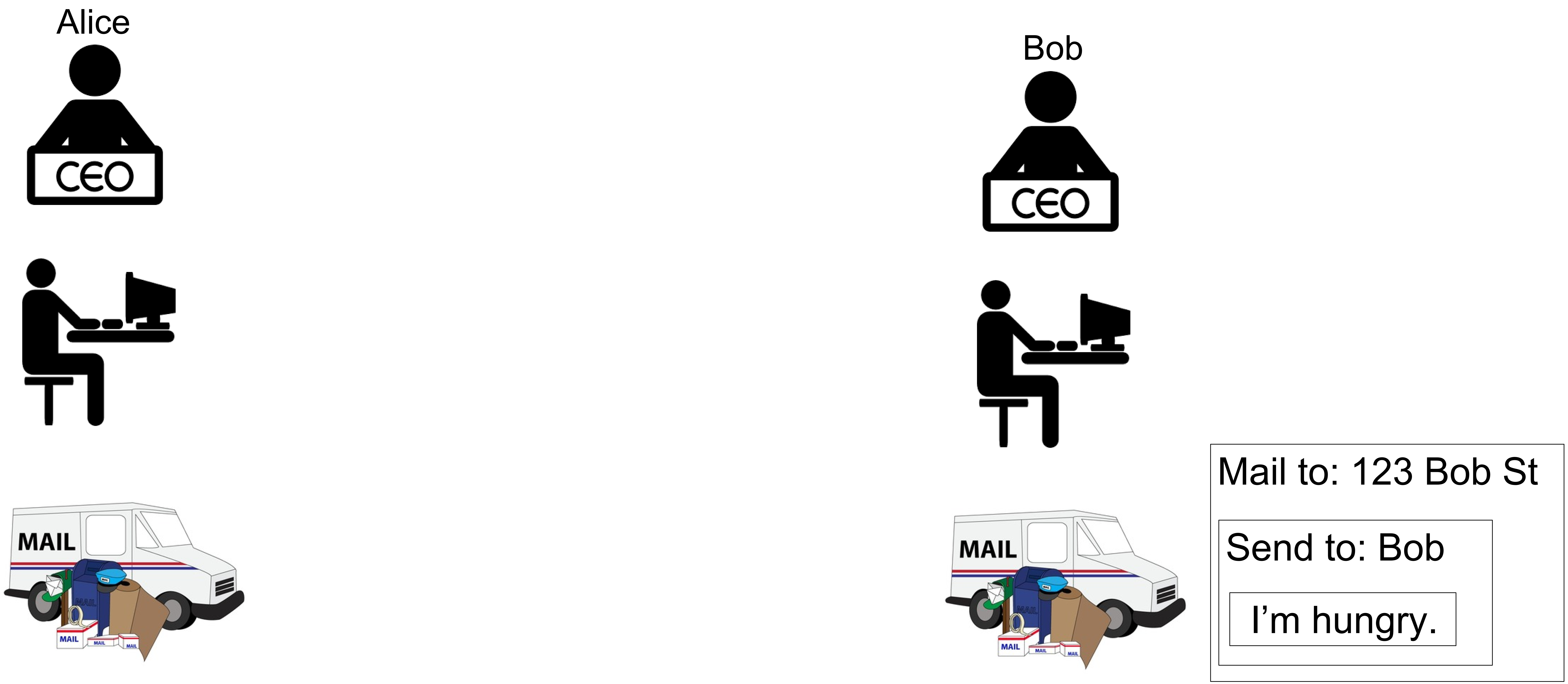
Mail to: 123 Bob St

Send to: Bob

I'm hungry.

Bob

Layers of abstraction



Layers of abstraction

Alice

Bob

Send to: Bob

I'm hungry.

Layers of abstraction

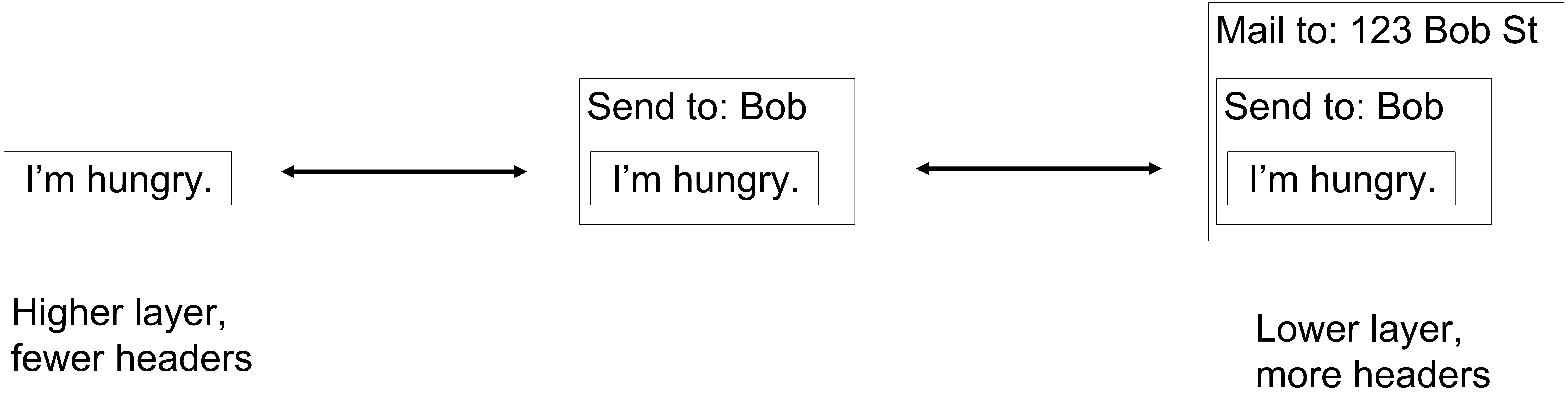
Alice

Bob

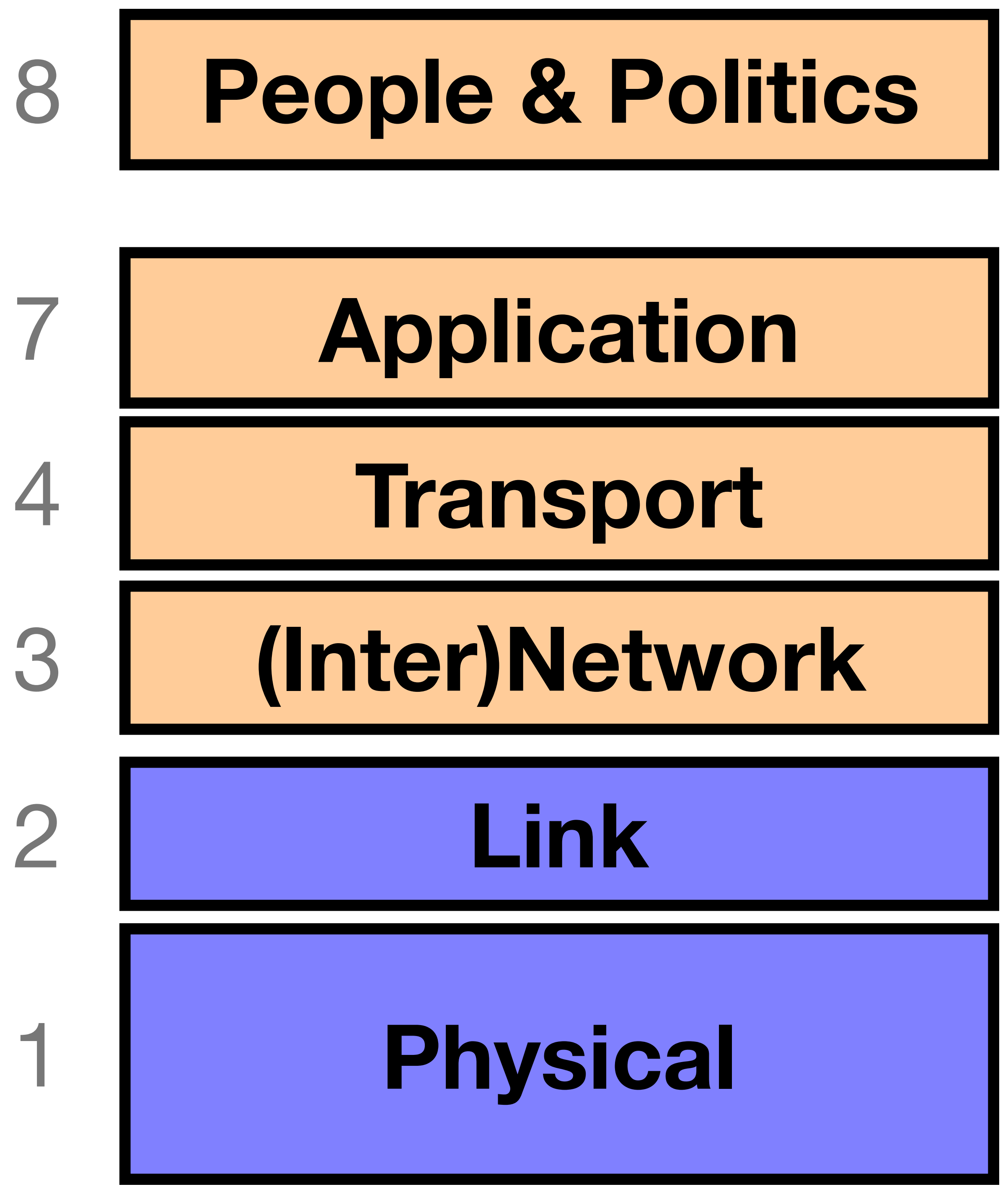
I’m hungry.

Layers of abstraction

- As you move to lower layers, we wrap additional headers around the message
- As you move to higher layers, you peel off headers around the message



Internet Layering (“Protocol Stack”/“**OSI Model**”)



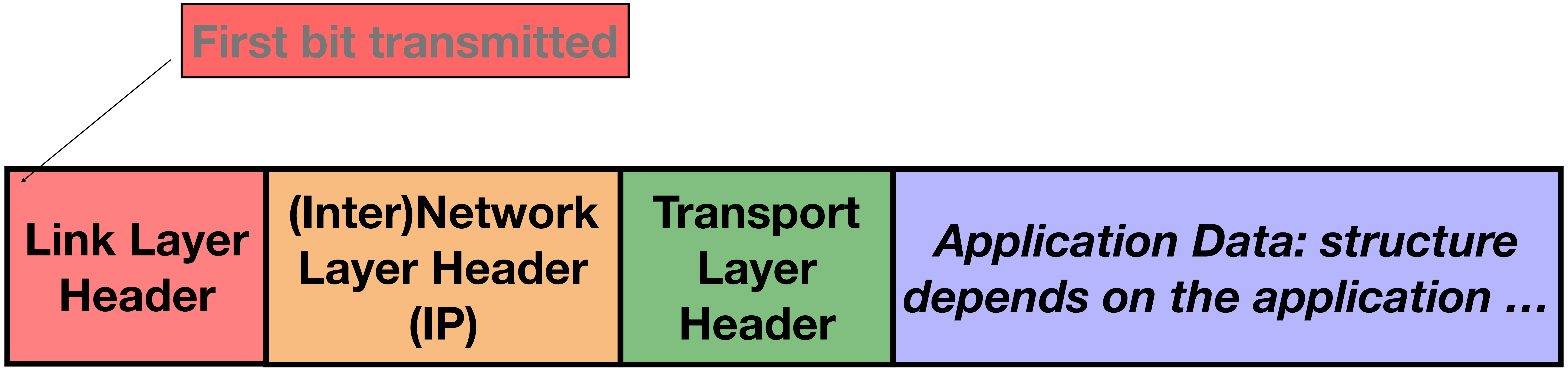
Note on a point of potential confusion: these diagrams are always drawn with lower layers **below** higher layers …

But diagrams showing the layouts of packets are often the *opposite*, with the lower layers at the **top** since their headers <u>precede</u> those for higher layers

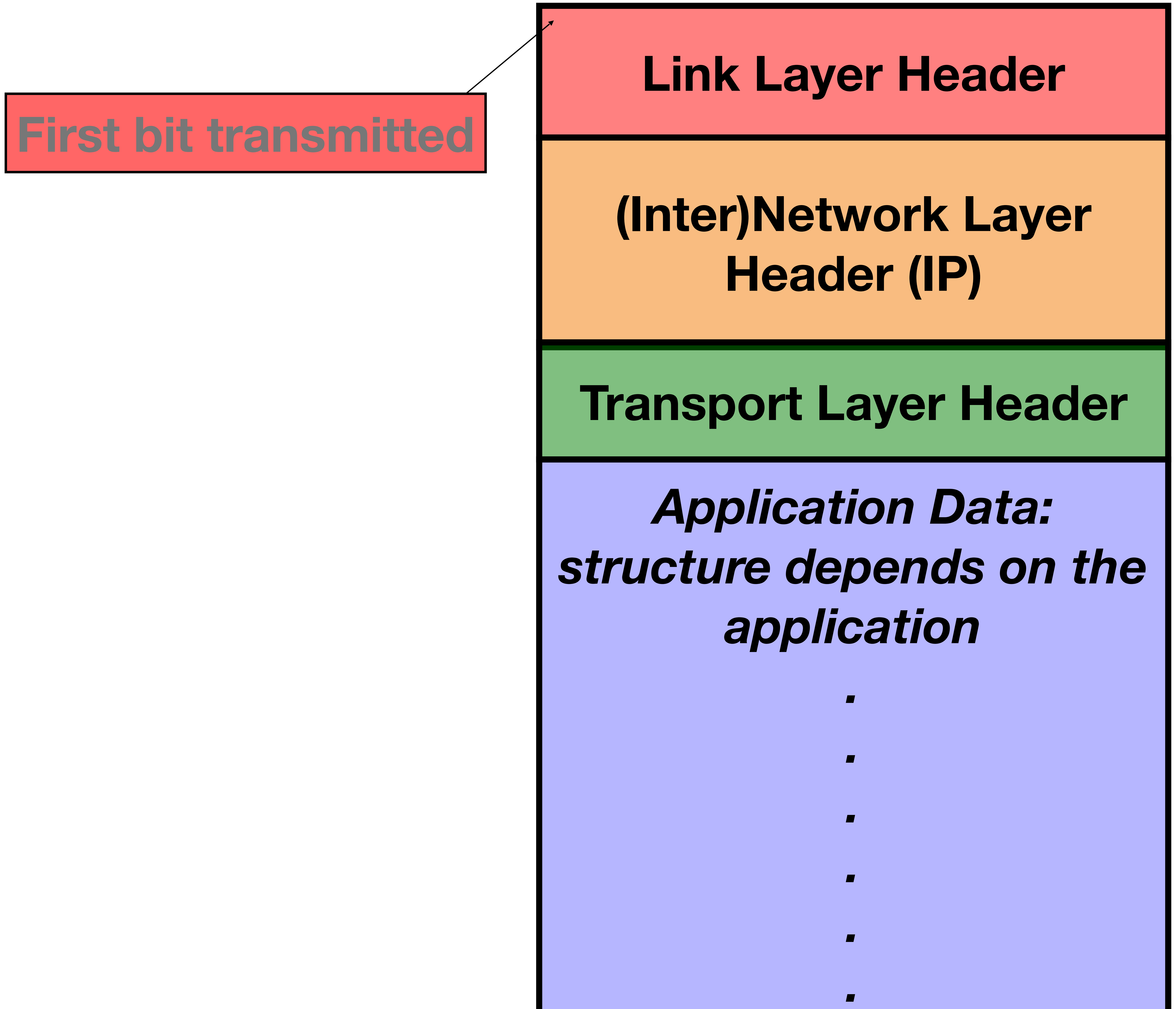
(And nobody remembers what layers 5 and 6 are for (“Session” and “Presentation) for the trivia buffs because they aren’t really used)

(also, layer 8 is a “joke”, but really is important)

Horizontal View of a Single Packet



Vertical View of a Single Packet



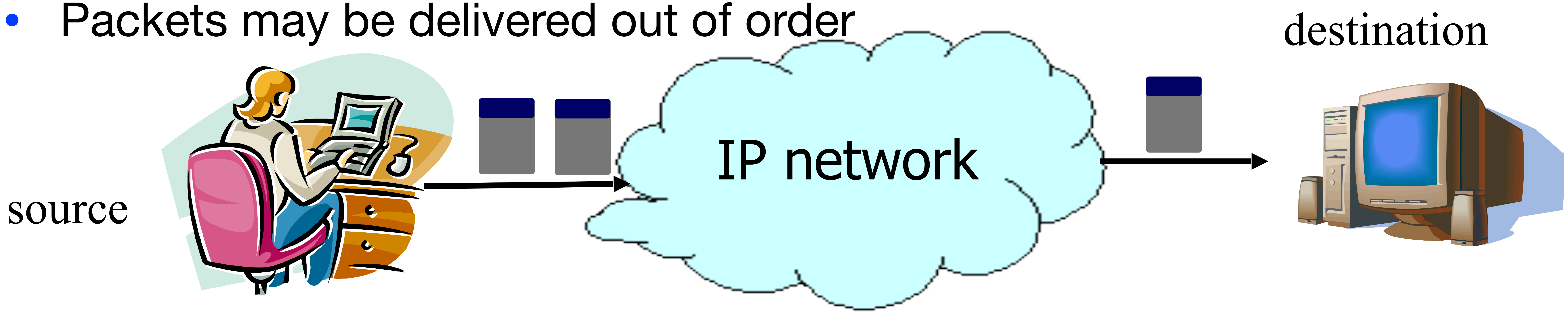
Network is Dumb

- Original Internet design: interior nodes ("routers") have no knowledge* of ongoing connections going through them
- Not how you picture the telephone system works
  - Which internally tracks all of the active voice calls
- Instead: the postal system!
  - Each Internet message ("packet") self-contained
  - Interior routers look at destination address to forward
  - If you want smarts, build it "*end-to-end*", not "hop-by-hop"
  - Buys simplicity & robustness at the cost of shifting complexity into end systems

* Today's Internet is full of hacks that violate this

IP: "*Best Effort* " Packet Delivery

- Routers inspect destination address, locate "next hop" in forwarding table
  - Address = ~unique identifier/locator for the receiving host
- Only provides a "*I'll give it a try*" delivery service:
  - Packets may be lost
  - Packets may be corrupted
  - Packets may be delivered out of order



“Best Effort” is Lame! What to do?

- It’s the job of our Transport (layer 4) protocols to build services our apps need out of IP’s modest layer-3 service

“Best Effort” is Lame! What to do?

- #1 workhorse: TCP (Transmission Control Protocol)
- Service provided by TCP:
  - Connection oriented (explicit set-up / tear-down)
    - End hosts (processes) can have multiple concurrent long-lived communication
  - **Reliable**, in-order, *byte-stream* delivery
    - Robust detection & retransmission of lost data