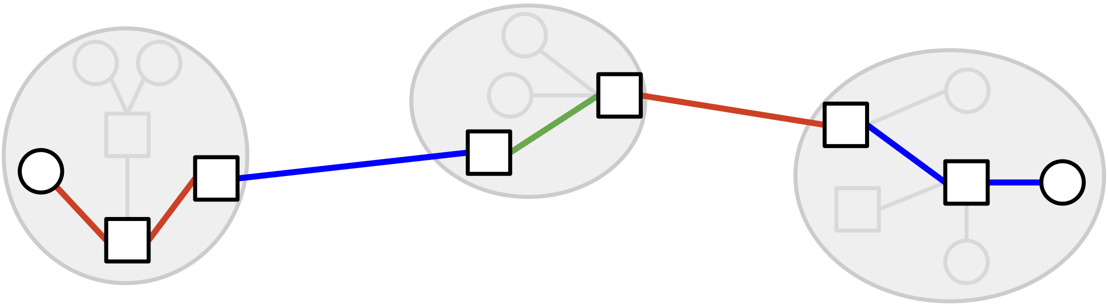


Lecture 1 (Intro 1)

Introduction, Layers of the Internet

CS 168, Spring 2025 @ UC Berkeley

Slides credit: Sylvia Ratnasamy, Rob Shakir, Peyrin Kao

Slides template credit: Josh Hug, Lisa Yan

Course Logistics

Lecture 1, CS 168, Spring 2025

Course Logistics

All course logistics are available on our website: https://sp25.cs168.io

- Policies page: https://sp25.cs168.io/policies/
- FAQs page: https://fa24.cs168.io/sp25-faqs/
- Won't cover logistics live during lecture, so please read those pages!
- We'll just do some quick reminders (copied from the pages above).

Joining Ed, bCourses, and Gradescope:

- Please do not email us if you are a concurrent enrollment student with a pending application; you will be added automatically within 3–4 days of submitting your application.
- If you just enrolled in the class, please don't email us about being added; we will sync the roster and add you within 3–4 days.

Course Logistics

Discussions and office hours start next week (January 27).

- You can attend any discussion section you want. Attendance is not taken.

Exam dates are on the website.

- Only one alternate exam, in-person only, immediately after main exam.

Stress management and accommodations:

- If you are registered with the Disabled Students' Program (DSP), please send us your letter of accommodations through the DSP portal as soon as possible.
- Your well-being is more important than this class. The website has a link to a form to request extensions.

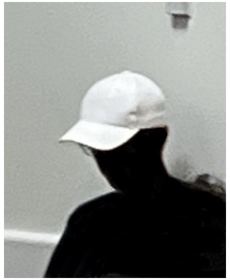
Our Talented Course Staff!

Peyrin Kao
he/him

Sylvia Ratnasamy
she/her

Aditya Tummala
he/him

Anita Ding
she/her



Arjun Damerla
he/him

David Li
he/him

Hongbo Wei
he/him

Ian Dong
he/him

Jaewon Lee
he/him

Ziming Mao
he/him

What is the Internet?

Lecture 1, CS 168, Spring 2025

What is the Internet?

The Internet transfers data between computers.

- Laptops, phones, tablets, car navigators, pacemakers, etc.

We'll focus on the *infrastructure* that ties these devices together.

- Less focus on the applications that rely on the Internet (e.g. Google, Facebook).

Our running analogy: Postal system.

- Focus on the infrastructure for sending mail.
- Less focus on what's inside the letters.

Why Study the Internet?

The Internet has and is transforming everything!

- The way we do business. *(Retail, advertising, cloud computing.)*
- The way we have relationships. *(Twitter, chat.)*
- The way we learn. *(Wikipedia, ChatGPT, AR/VR.)*
- The way we govern. *(E-voting, censorship, cyber-warfare.)*
- The way we cure diseases. *(Digital health, remote surgery.)*

Why is the Internet Interesting?

Networking is different from many traditional computer science fields.

- Theorists: "What's your formal model of the Internet?"
- Hardware engineers: "You don't have performance benchmarks?"
- My parents: "Doesn't the Internet already work?"

Internet Design Challenges – Federation

Challenge: The Internet is **federated**.

- No single operator. Over 100,000 different network operators!
  - UC Berkeley, AT&T, China Telecom, etc.
- Operators most *cooperate* to form a global network.
- Must consider business incentives.
  - Rivals might not want to share private information.
- Complicates innovation.
  - Operators have to run the same software to talk to each other.
  - If you have a brand-new feature, but nobody else has it, it's useless.

Internet Design Challenges – Scale, Evolution, and Diversity

Challenge: The Internet is **scalable**.

- Billions of users, accessing trillions of web pages.

Challenge: The Internet is **constantly evolving**.

- Demand is constantly increasing!

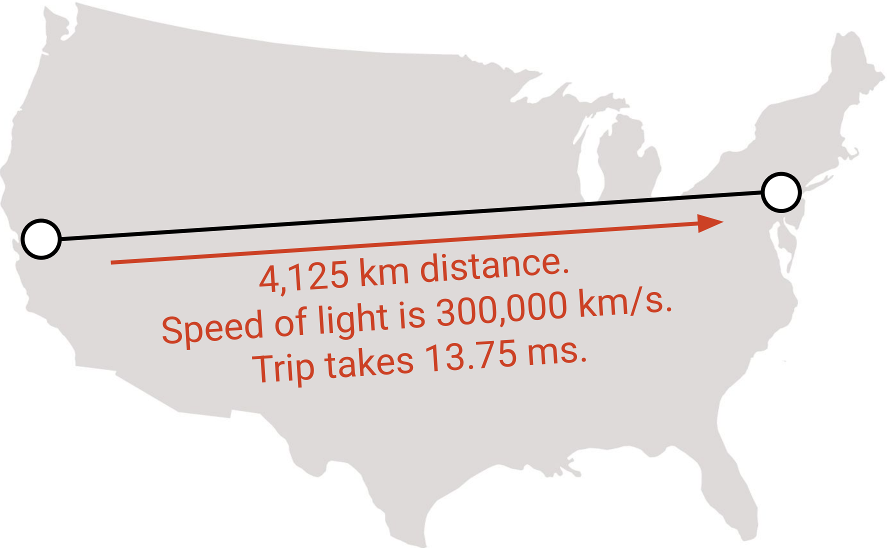
Challenge: The Internet is **diverse**.

- Some users download more data than others.
- Some devices are higher-capacity than others.

Challenge: The Internet is **asynchronous**.

- We're constrained by the speed of light.
- Any data we receive is already dated.

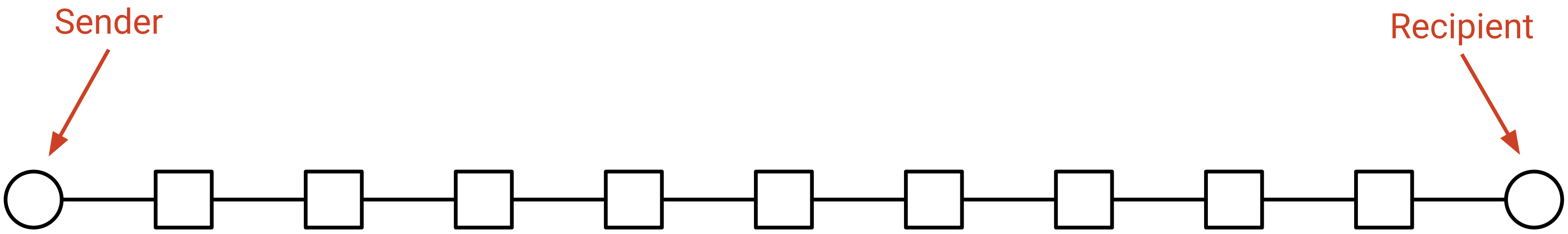
By the time the message is sent,
our 3 GHz CPU has executed
42,000,000 more cycles!



Internet Design Challenges – Fault Tolerance

Challenge: The Internet must **handle failures at scale**.

- Sending a message requires many components (wires, network devices, software).
- Asynchrony: Might take a long time to hear the bad news.
- The Internet was the first system that had to handle failure at scale!



If we had 50 components, each working 99% of the time,
there's a 39.5% chance that at least one of them fails!

Challenges of the Internet – Recap

The Internet:

- Is a federated system.
- Operates at enormous scale.
- Is constantly evolving.
- Has a tremendous range and diversity of users and devices.
- Operates asynchronously.
- Must handle failures at scale.

Why Is the Internet Interesting?

Designing the Internet required new ways of thinking.

- The design of the Internet influenced how we design modern systems!

We have no theoretical model or performance benchmark.

- The Internet is not "optimal" according to any metric.
- But it balances lots of different goals very well.
- Need to think about practical trade-offs.

Writing code that works is not enough.

- Code must respect companies' business incentives. *(Federation.)*
- Code must run at enormous scale. *(Scale, fault-tolerance.)*

Protocols

The Internet is all about designing **protocols**.

- Protocol: A specification on how to communicate.
  - Syntax: Format of messages. What do the 1s and 0s mean?
  - Semantics: What actions should I take in response to certain messages?
- Example: Protocol for asking a question in lecture?
  - Raise your hand.
  - Wait for speaker to call on you.
  - Ask your question after speaker calls on you.
  - If speaker doesn't see you after some time, say "Excuse me!"
- Designing a good protocol is harder than it first seems!
  - The IETF *(Internet Engineering Task Force)* standardizes and publishes protocols in RFC *(Request For Comments)* documents.

Layers of the Internet: Building Layers 1–3

Lecture 1, CS 168, Spring 2025

Layer 1 – Moving Bits Across Space

We need some **physical** technology to move data across space.

- Postal analogy: 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.

Are pigeons faster than the Internet?

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

Layer 1 – Moving Bits Across Space

We need some **physical** technology to move bits across space.

- Voltages on electrical wire.
- Light signals on optical fiber.
- Wireless radio waves.

Won't go into detail in this class.

Risks [edit]

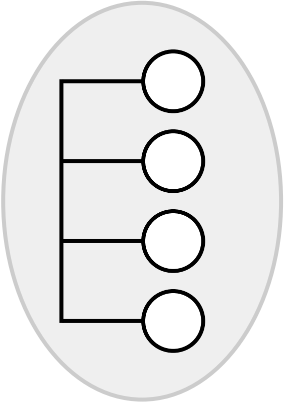
Although collisions are unlikely, packets can be lost, particularly to raptors.

Layer 2 – Local Networks

Postal analogy: Use our physical technology to connect everybody in the local town.

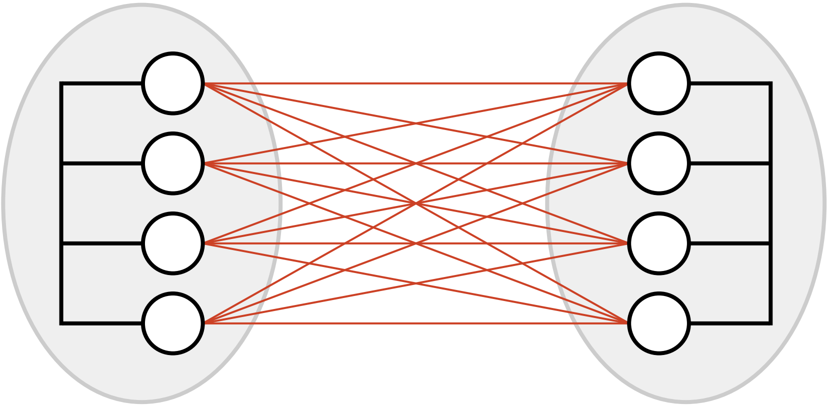
Forming a local network:

- Use physical technology to create a **link** between machines.
- Use links to connect all machines in a local area.
- Machines can exchange **packets**: A group of bits representing a message.



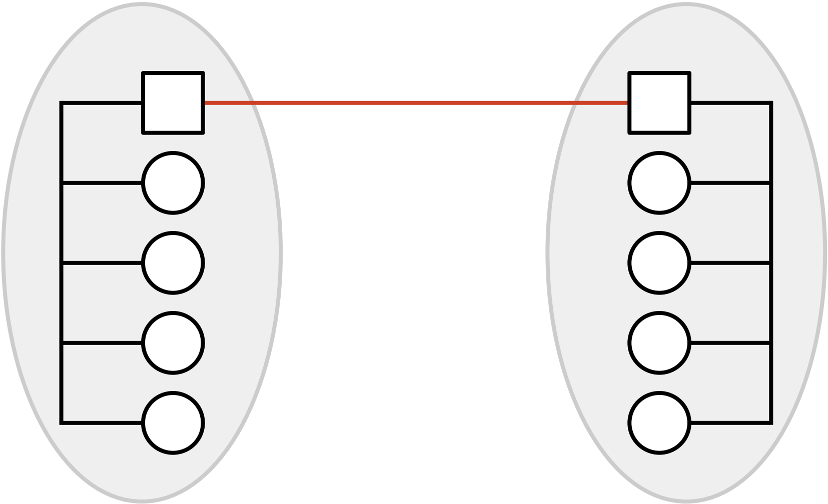
Postal analogy: How do we connect houses from different towns?

- Adding new links between every pair of houses is inefficient.



Postal analogy: How do we connect houses from different towns?

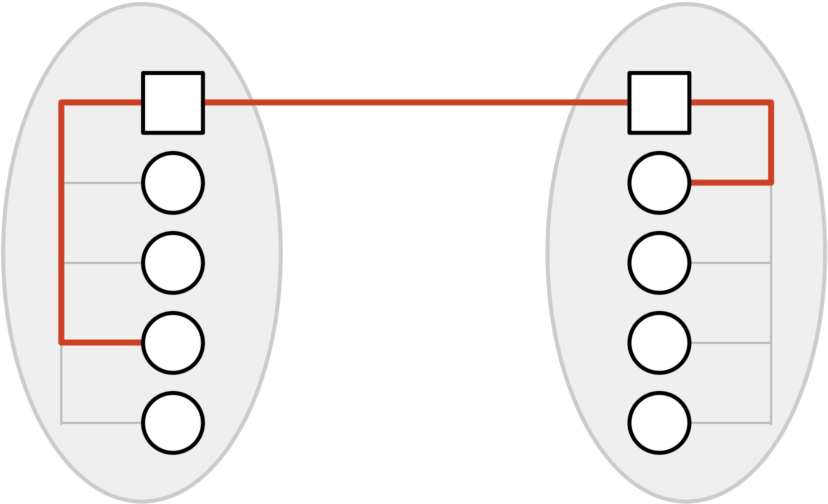
- Solution: Introduce a *post office* in each town.
- Just connect the two post offices.



Layer 3 – Connecting Local Networks

To send a letter to the other town:

- You send the packet to...
- Your local post office, which sends the packet to...
- The other town's post office, which sends the packet to...
- The final destination.



Layer 3 Characteristics

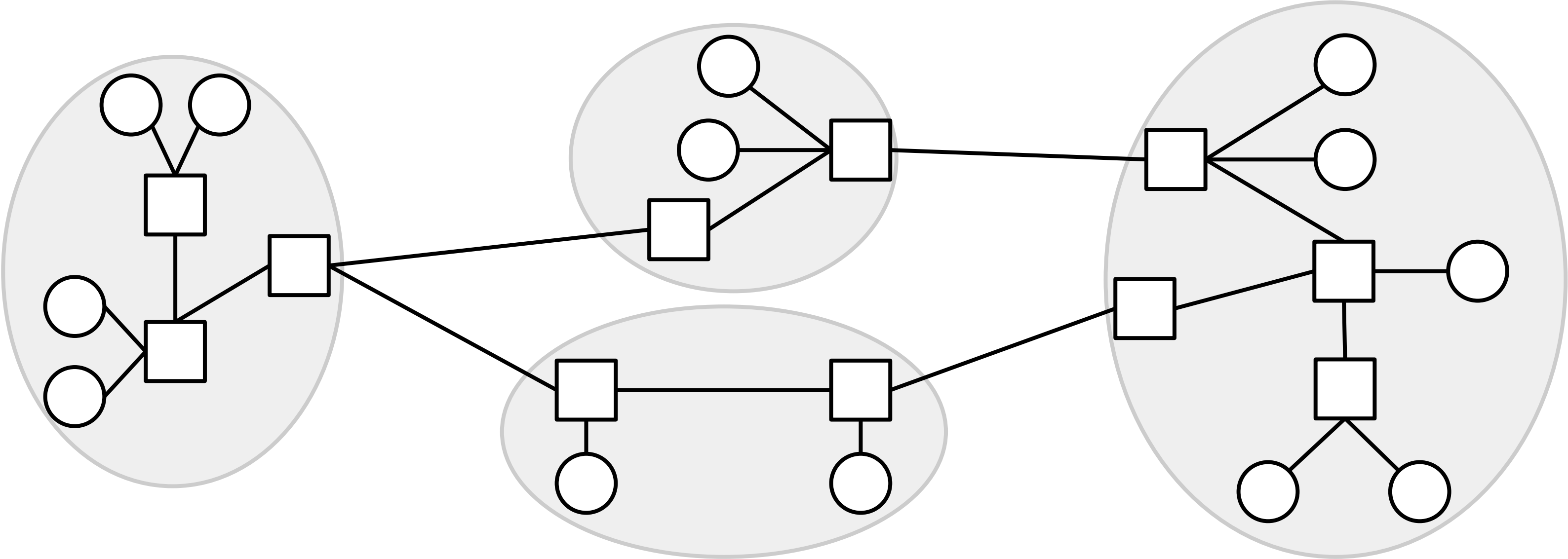
Lecture 1, CS 168, Spring 2025

Network of Networks

With enough post offices, we can connect all the towns in the world!

The Internet is a **network of networks**.

- Each operator runs its own local network.
- The local networks connect to each other to form the Internet.



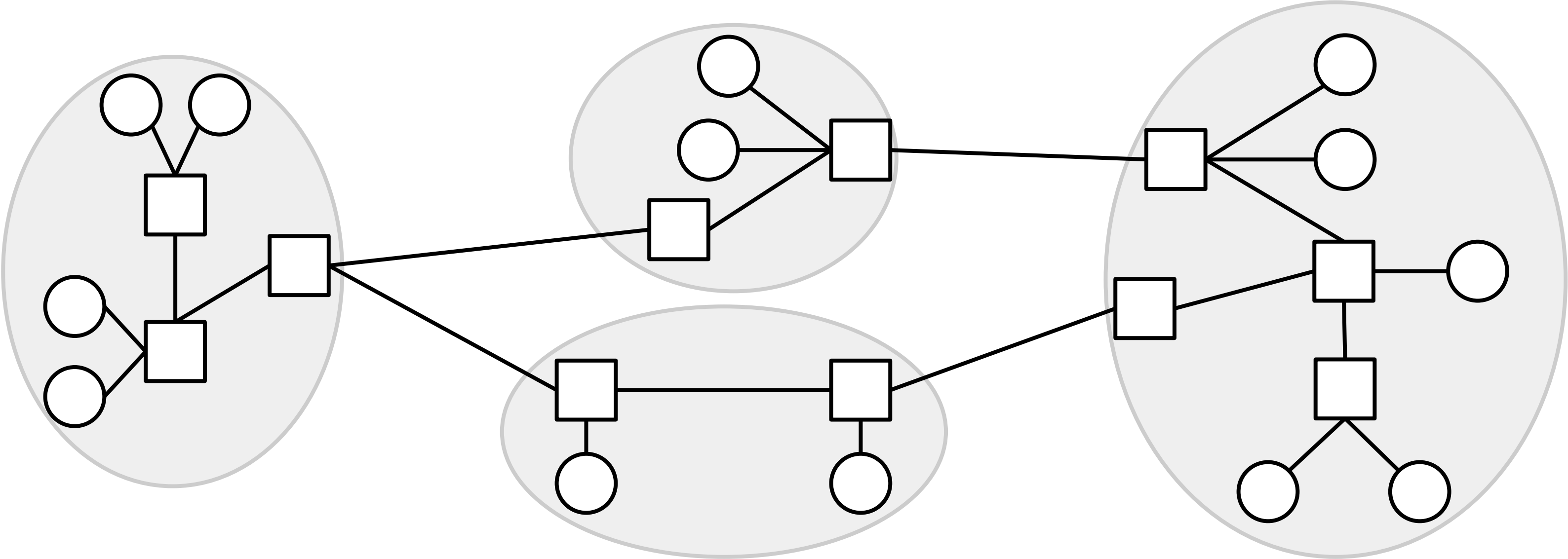
Hosts vs. Switches

End hosts are the machines communicating over the Internet.

- Analogy: Houses.
- Examples: Your laptop, your phone, Google's server.

Switches (aka **routers**) receive packets and forward them toward their destination.

- Analogy: Post offices.



Layers of Abstraction

Modularity: In our design, we decomposed the system into layers of abstraction.

- Each layer relies on services from the layer below.
- Each layer provides services to the layer above.

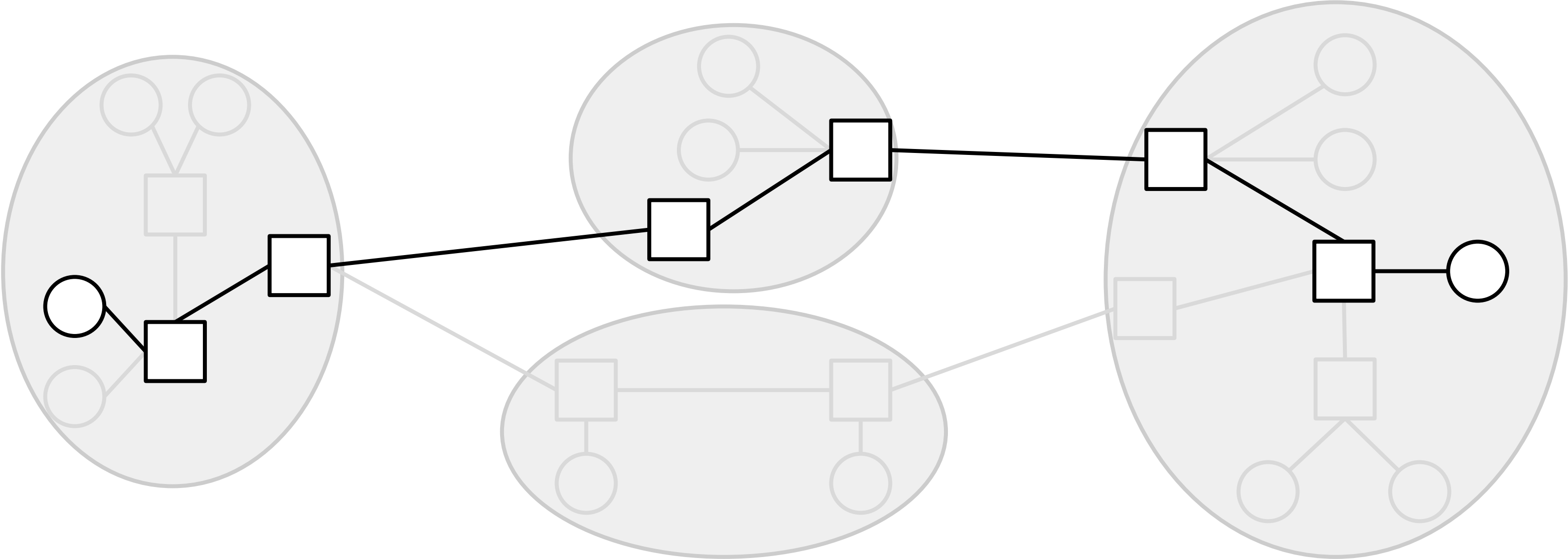
Abstraction is very powerful.

- Layer 3 designer doesn't have to think about voltages on the wire.
- A change in Layer 2 protocols doesn't affect the other layers.

Global Delivery at Layer 3

A packet can take multiple **hops** to reach its destination.

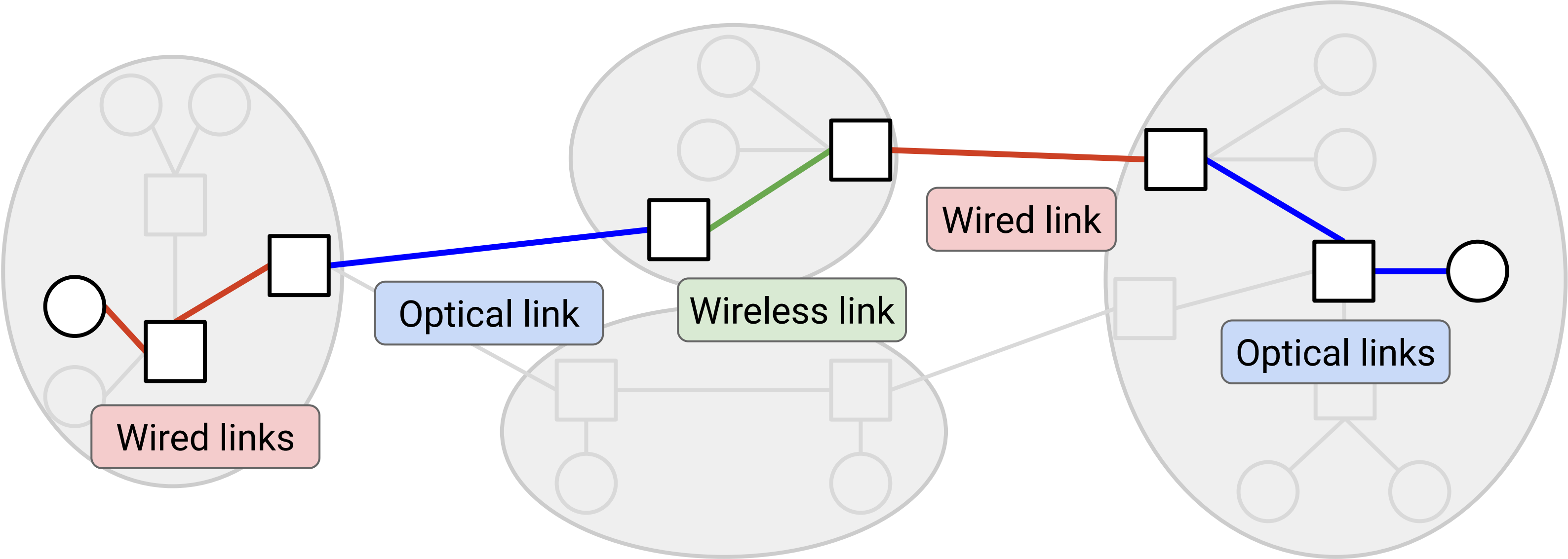
- Each router needs to **forward** the packet closer to its destination.



Global Delivery at Layer 3

A packet can travel across multiple networks to reach its destination.

- Each local network along the way could use a different Layer 2 protocol.



Layer 3 is Best-Effort

Layer 3 offers a **best-effort** service model.

- Packets are limited in size.
- Packets could get lost, reordered, corrupted, etc.
- The network will try its best to deliver your packet, but no guarantee.
- The network won't tell you if the delivery failed.

We need to build more layers if we want to guarantee packet delivery.

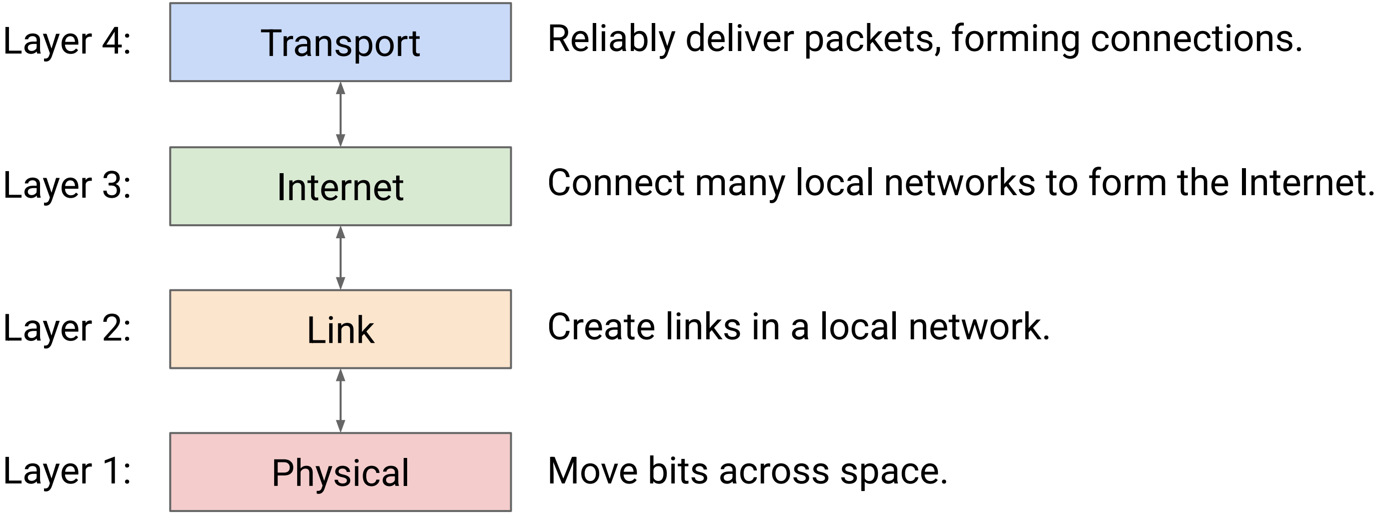
Layers of the Internet: Building Layers 4–7

Lecture 1, CS 168, Spring 2025

Layer 4 – Reliability

Transport layer builds on top of Layer 3 (global packet delivery).

- Adds extra mechanisms (e.g. re-sending lost packets) for reliable packet delivery.
- Splits up large data into packets to send them. Reassembles received packets.
- Instead of individual packets, can think about **flows** (aka **connections**): A stream of packets exchanged between two endpoints.

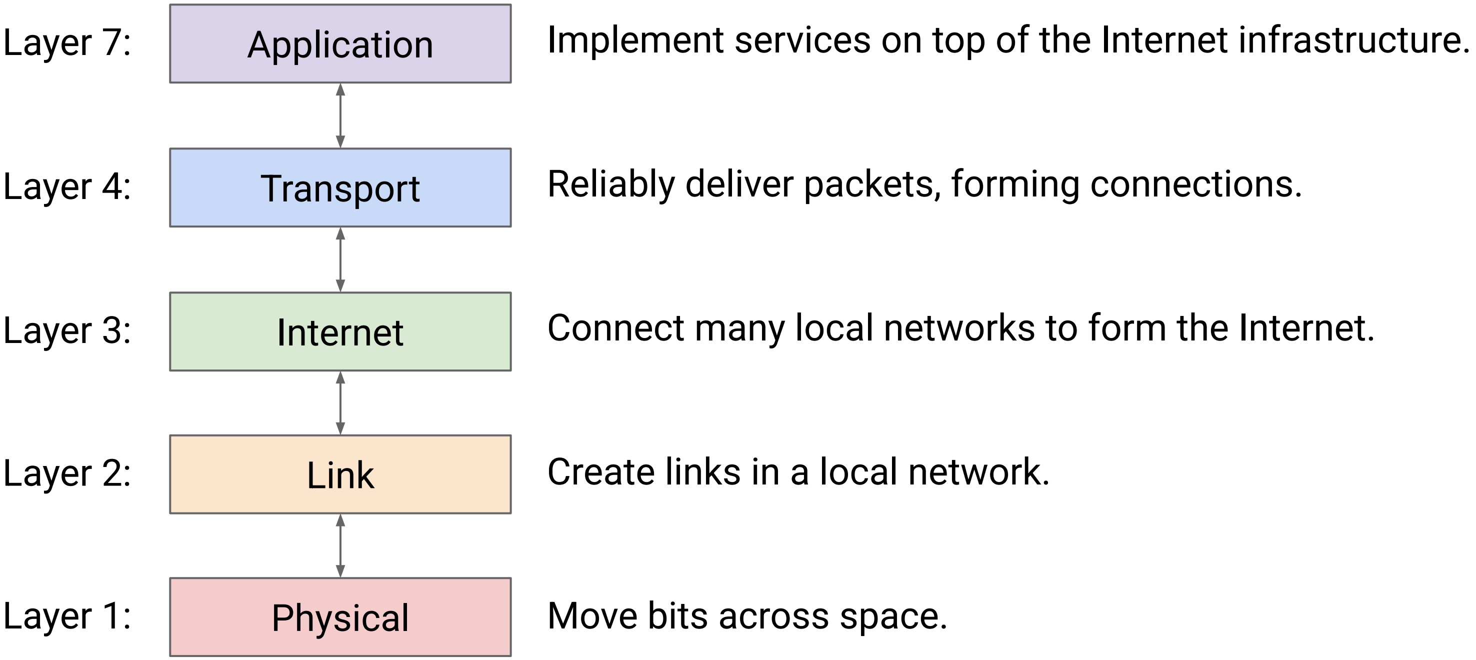


Layer 7 – Application

Application layer builds services (e.g. websites, video streaming) on top of Layer 4.

- This design lets us build different services, all on the same infrastructure.

Note: Layers 5 and 6 are now obsolete.



Header Fields

Lecture 1, CS 168, Spring 2025

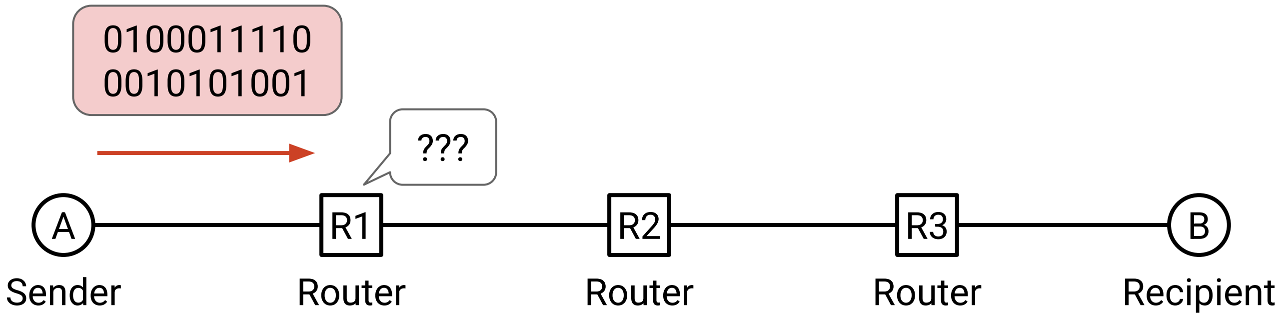
Why Do We Need Headers?

Suppose A wants to send an image to B.

- A forms a packet with the bits of the image. *(May need to split image into multiple packets.)*
- A sends the packet to the next router.
- The router has no idea what these bits are for!

The packet needs some extra **metadata**, to tell us what to do with the packet.

- Analogy: Letter needs to be put in an envelope.
  Envelope describes what to do with the letter.



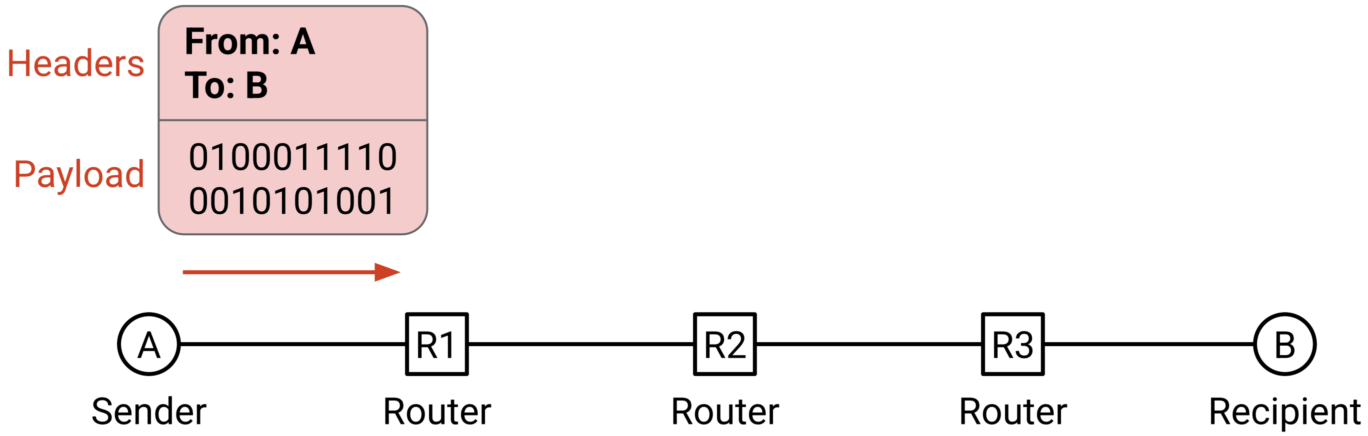
Common Header Fields

The packet **header** contains metadata describing how the data should be sent.

Some common fields in a header:

- Destination address: Required to deliver the packet.
- Source address: Useful if the recipient wants to send replies back.

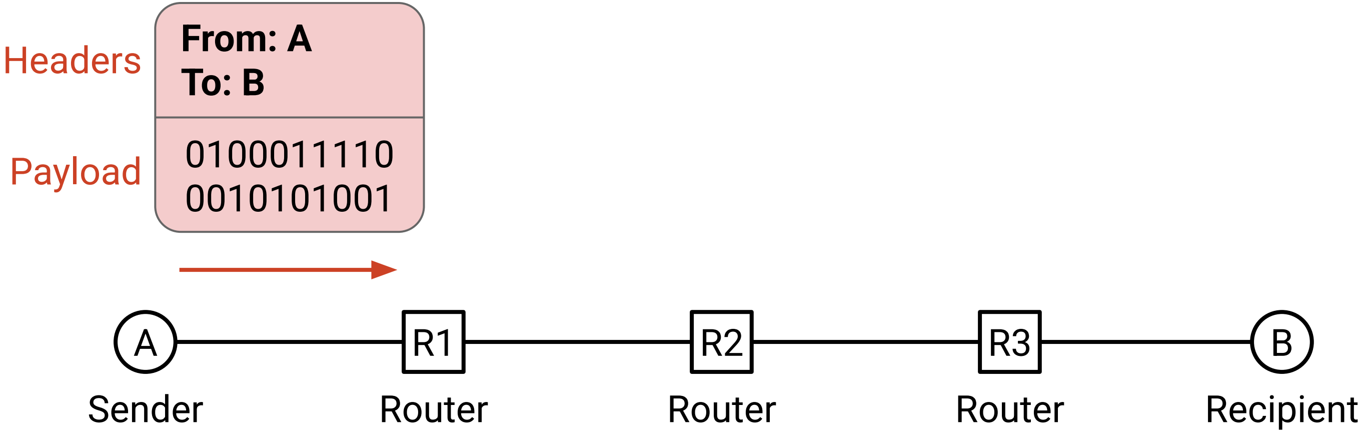
The actual data in the packet is called the **payload**.



Headers are Standardized

Everybody needs to agree on the format of the header.

- "First 8 bits are the source, next 8 bits are the destination..."
- If we use a different format, others won't understand the header.



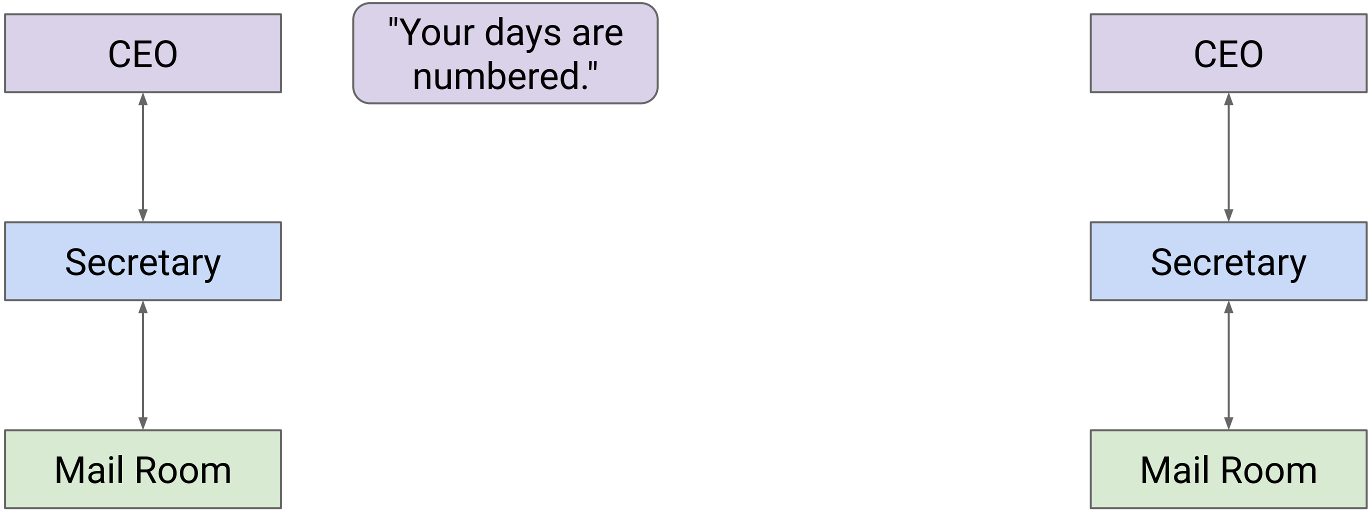
Multiple Headers (Endpoints Only)

Lecture 1, CS 168, Spring 2025

Postal Analogy

CEO Alice wants to send a message to CEO Bob.

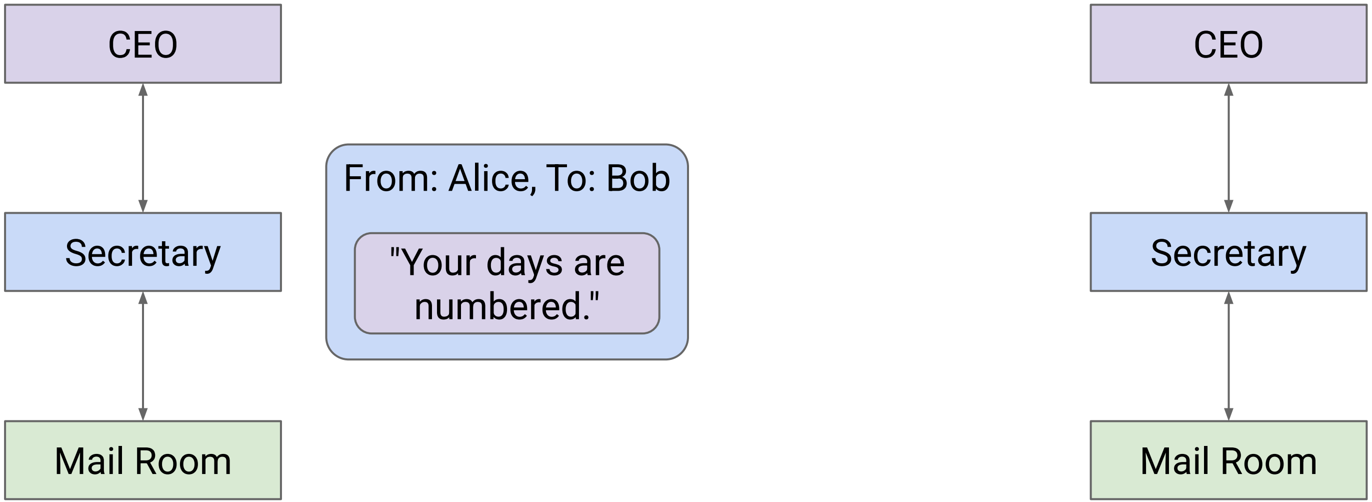
Alice writes a letter.



Postal Analogy

Alice passes the letter down to her secretary.

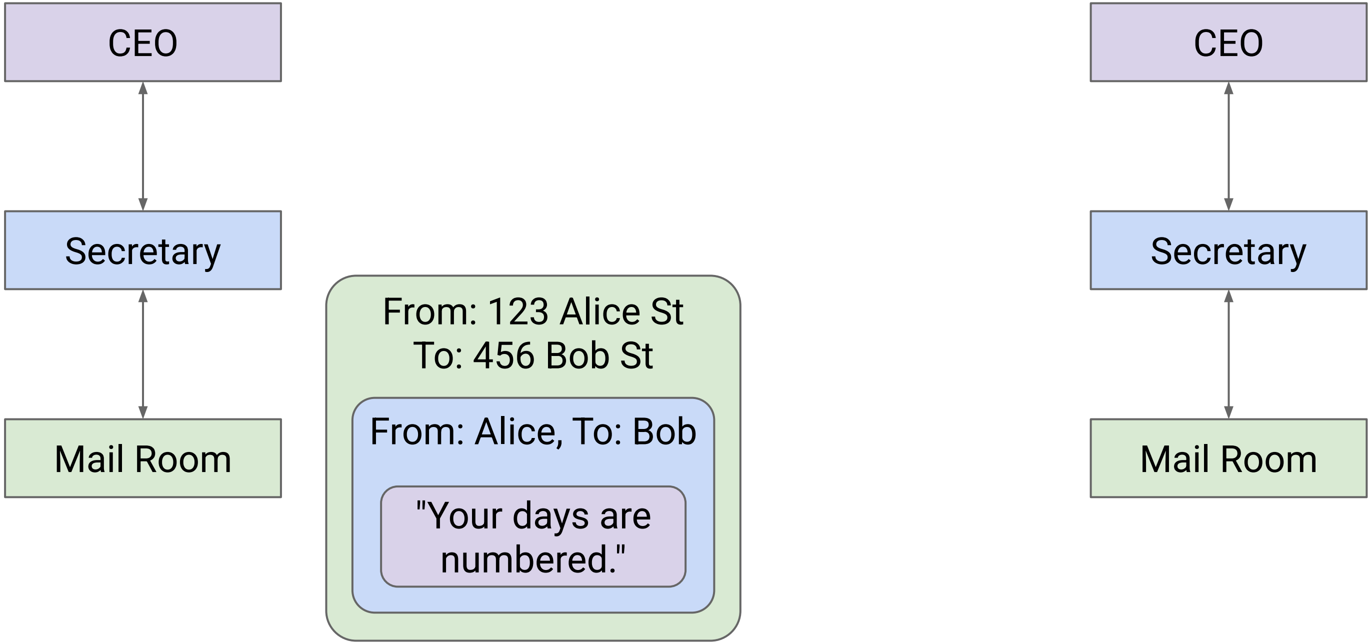
Her secretary puts the letter in an envelope.



Postal Analogy

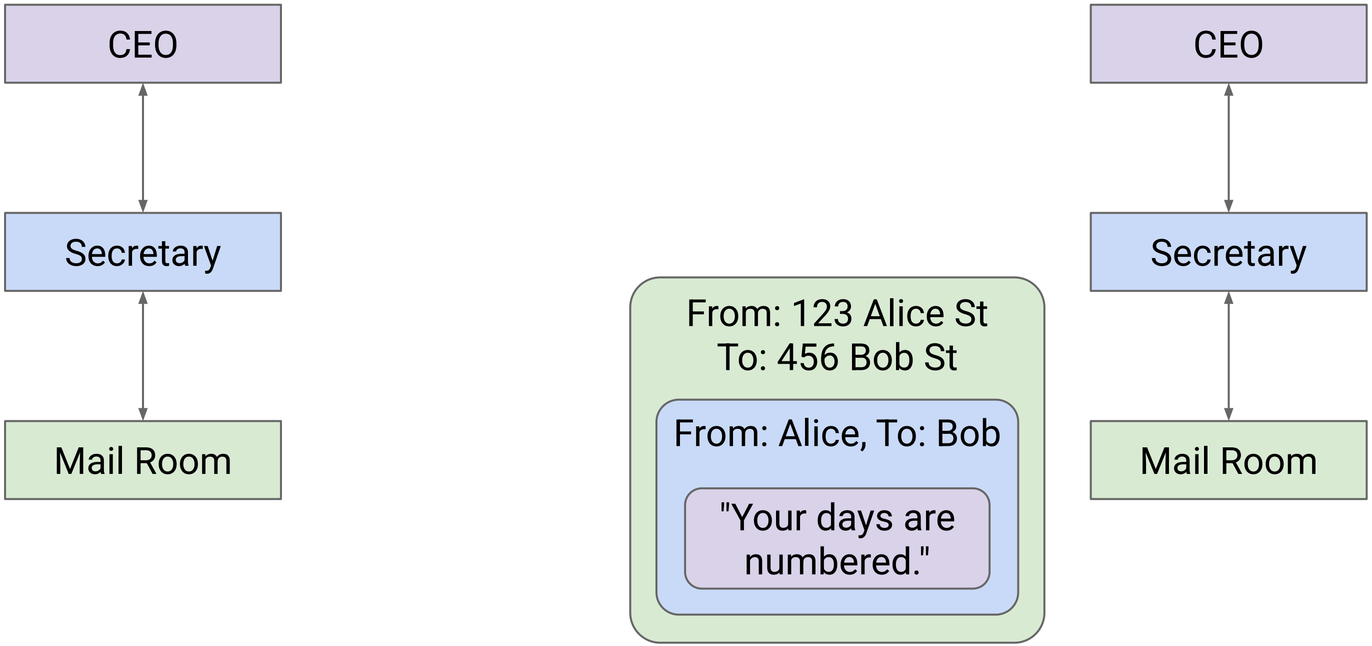
Her secretary passes the letter down to the mailman.

The mailman puts the envelope in a box.



Postal Analogy

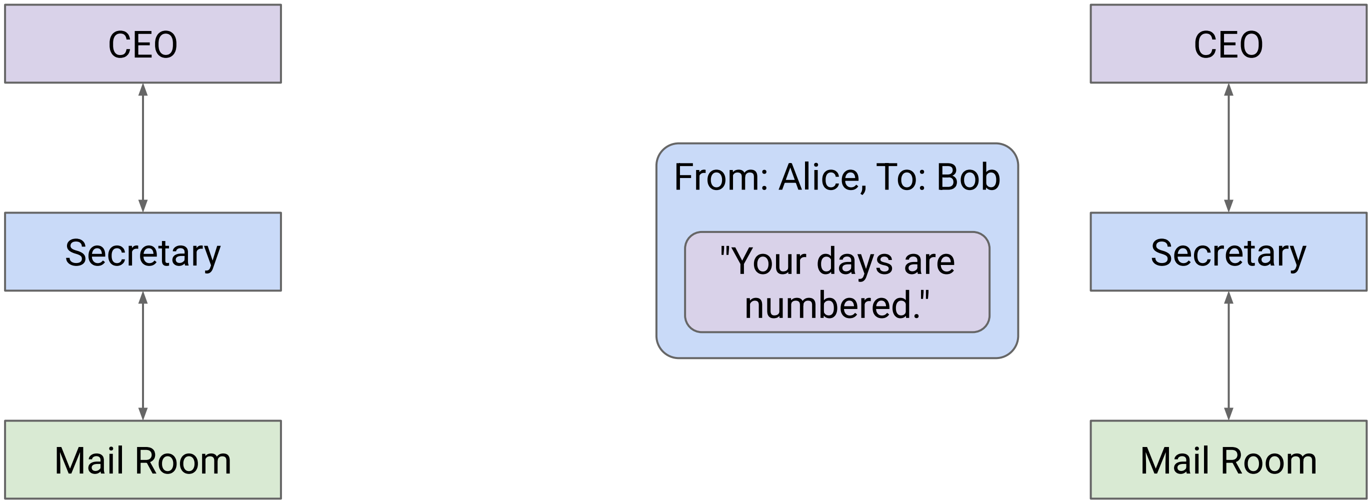
The packet travels through the postal system, to Bob's building.



Postal Analogy

The mailman *unwraps* the box, revealing the envelope inside.

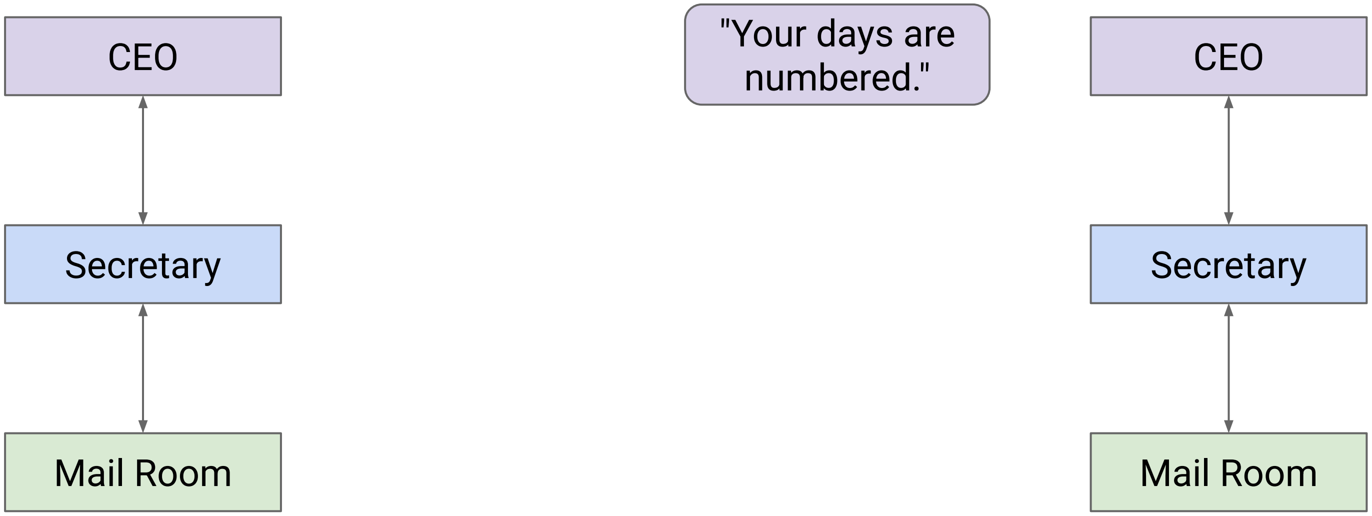
The mailman passes the envelope up to the secretary.



Postal Analogy

The secretary *unwraps* the envelope, revealing the letter inside.

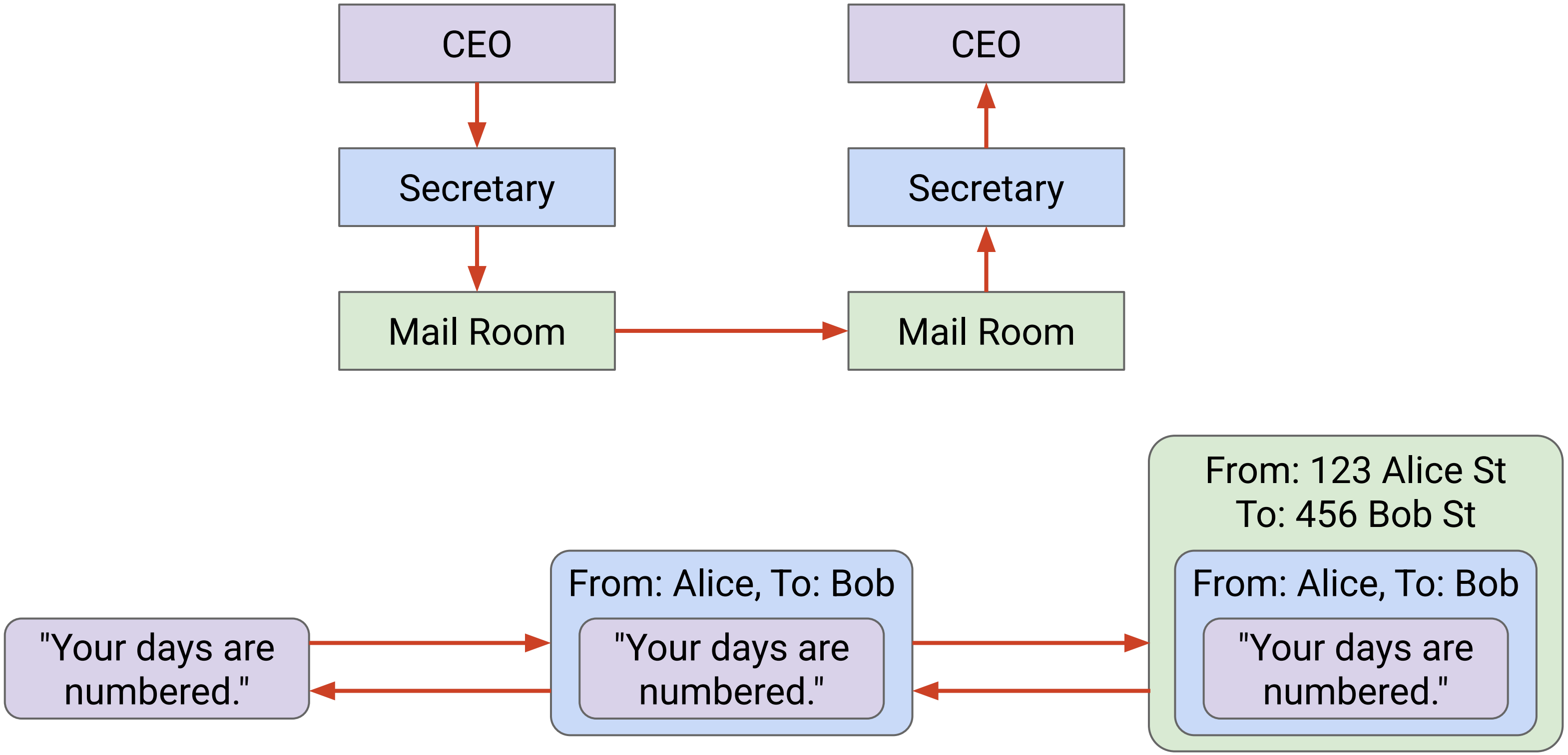
The secretary passes the letter up to Bob.



Postal Analogy

As we move to lower layers, we wrap additional headers around the packet.

As we move to higher layers, we peel off headers, revealing the inner headers.



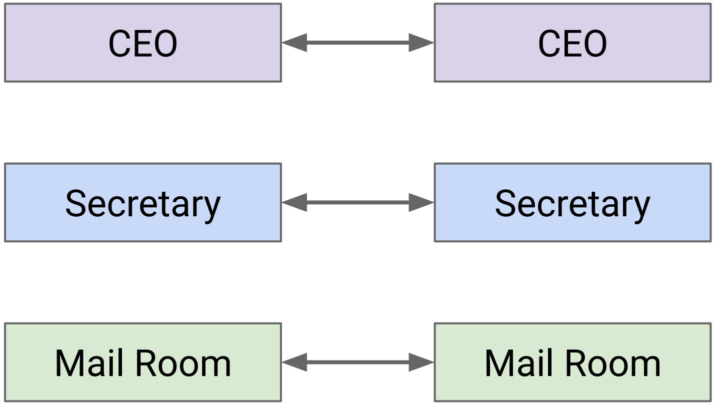
Postal Analogy

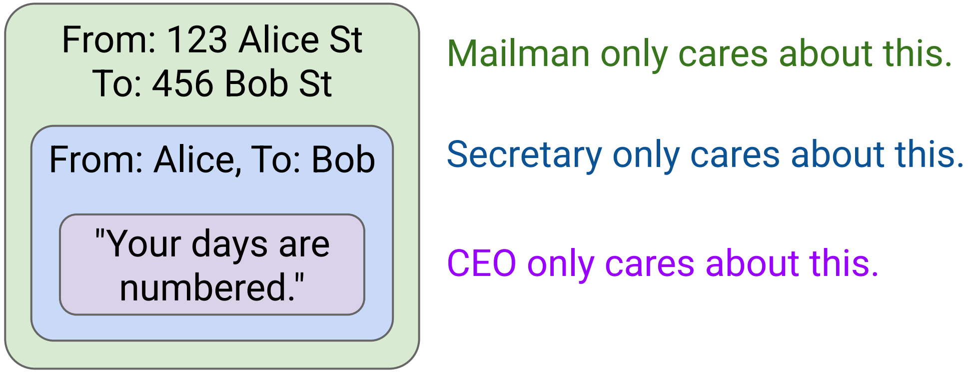
Each person only cares about the headers at their layer.

- Mailman reads the green header, ignores all the payload inside.

Each person communicates with its peers at the same layer.

- Alice's secretary writes the blue header, for Bob's secretary to read.
- A protocol at a specific layer only makes sense to people at that layer.

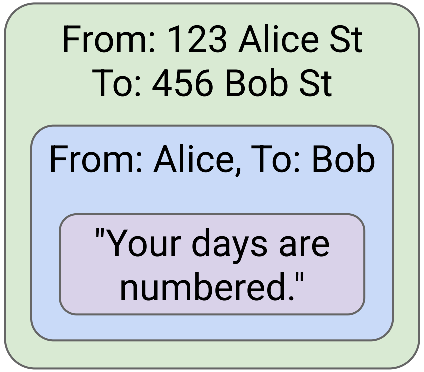




Addressing at Different Layers

Notice: Different layers use different addressing schemes.

- Inside a building: "413 Soda Hall."
- In the postal system: "2551 Hearst Ave, Berkeley, CA."

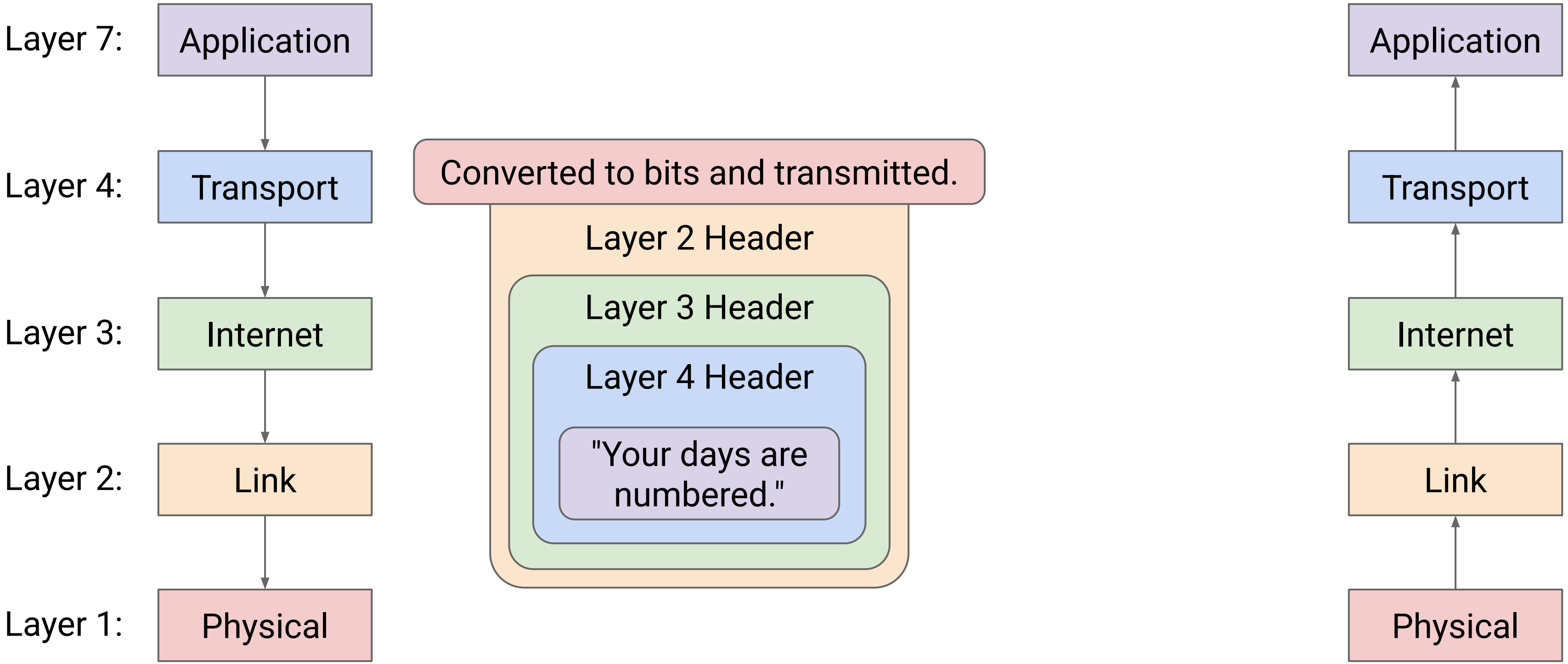


These addresses make sense to the mailman.

These names make sense to the secretary.

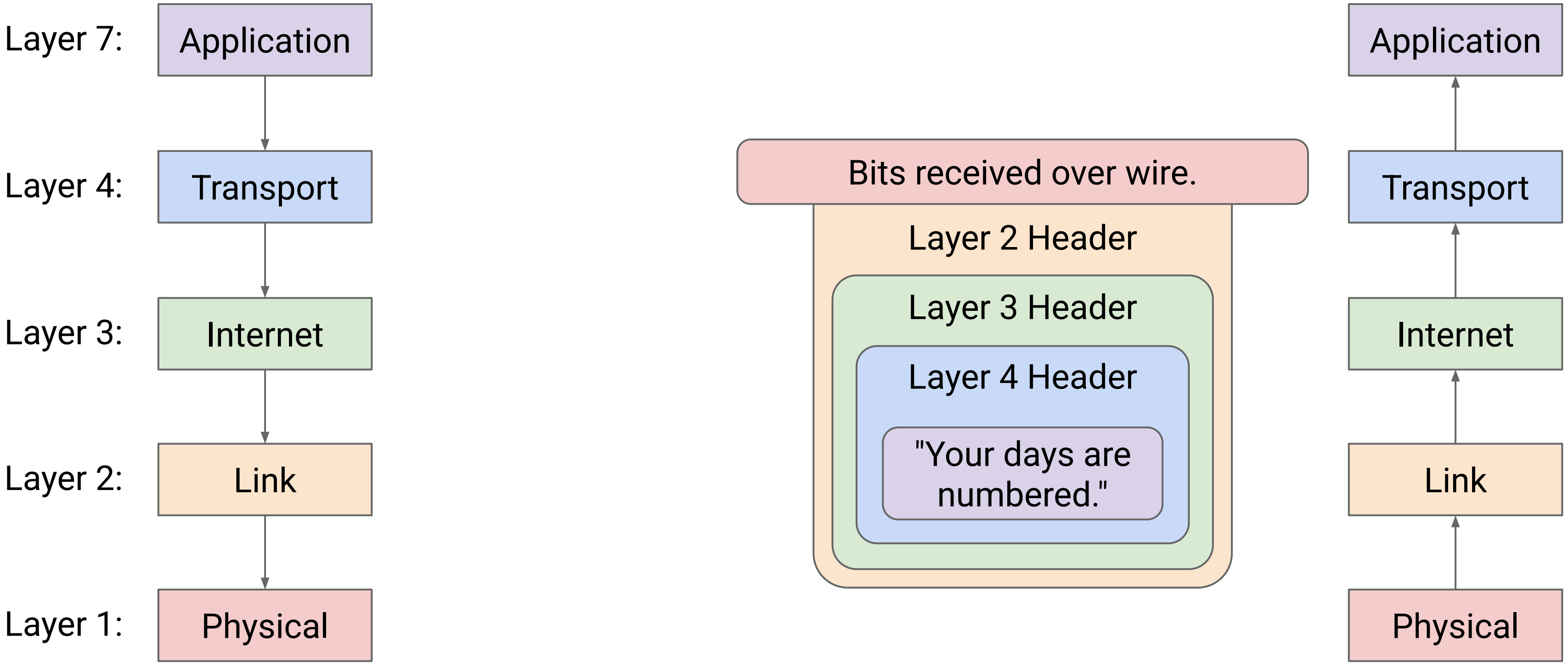
Multiple Headers

As we move to lower layers, we wrap additional headers around the packet.



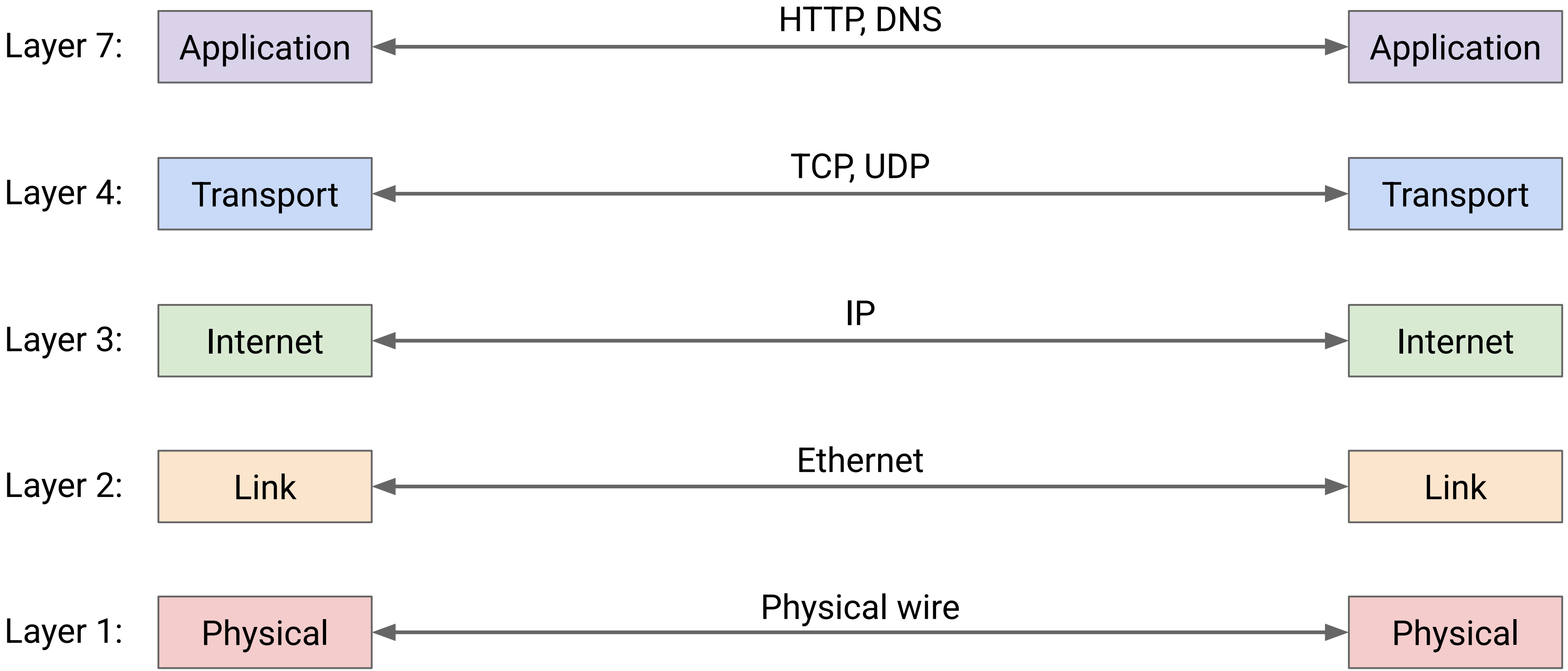
Multiple Headers

As we move to higher layers, we peel off headers, revealing the inner headers.



Multiple Headers

Peers at the same layer communicate with each other using the header at that layer.

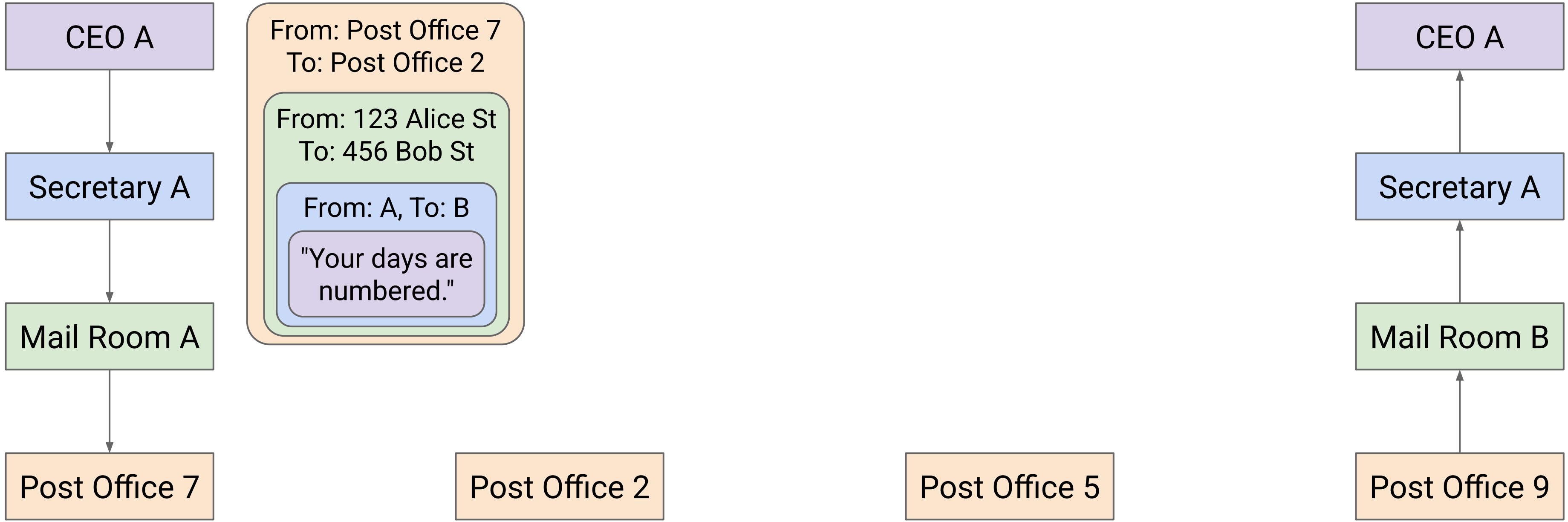


Multiple Headers (With Routers)

Lecture 1, CS 168, Spring 2025

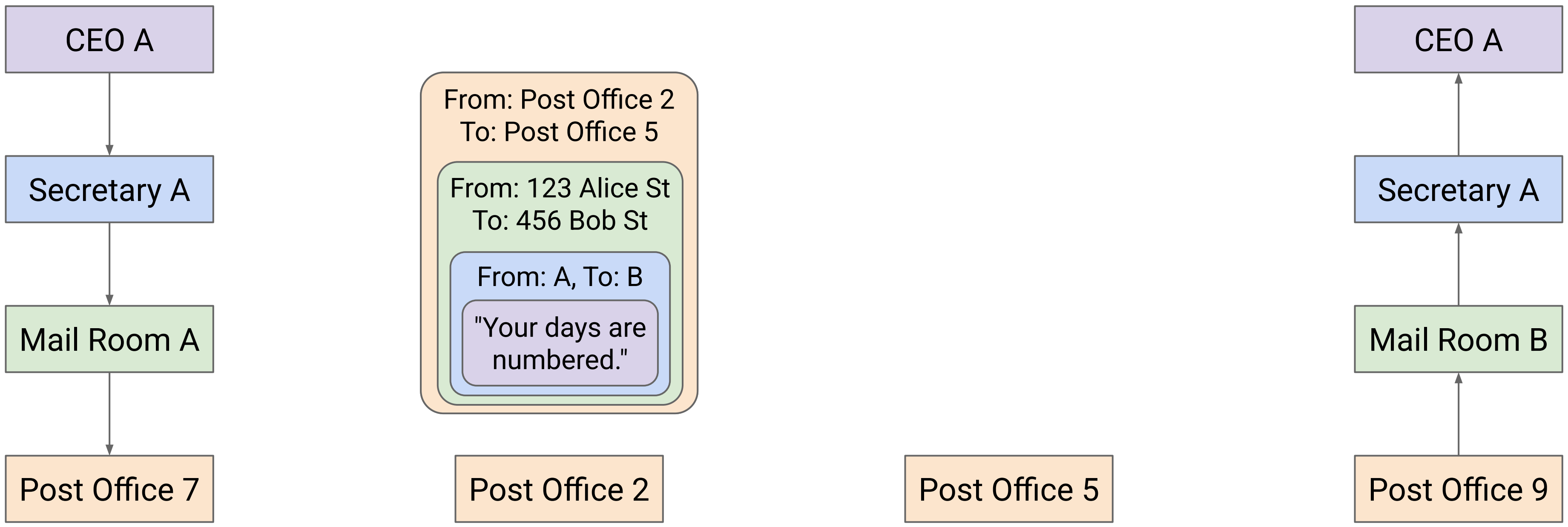
Postal Analogy

The letter might hop across multiple post offices.
Each post office unwraps the box, revealing the envelope inside.
Then, it puts the letter in a new box, destined for the next post office.



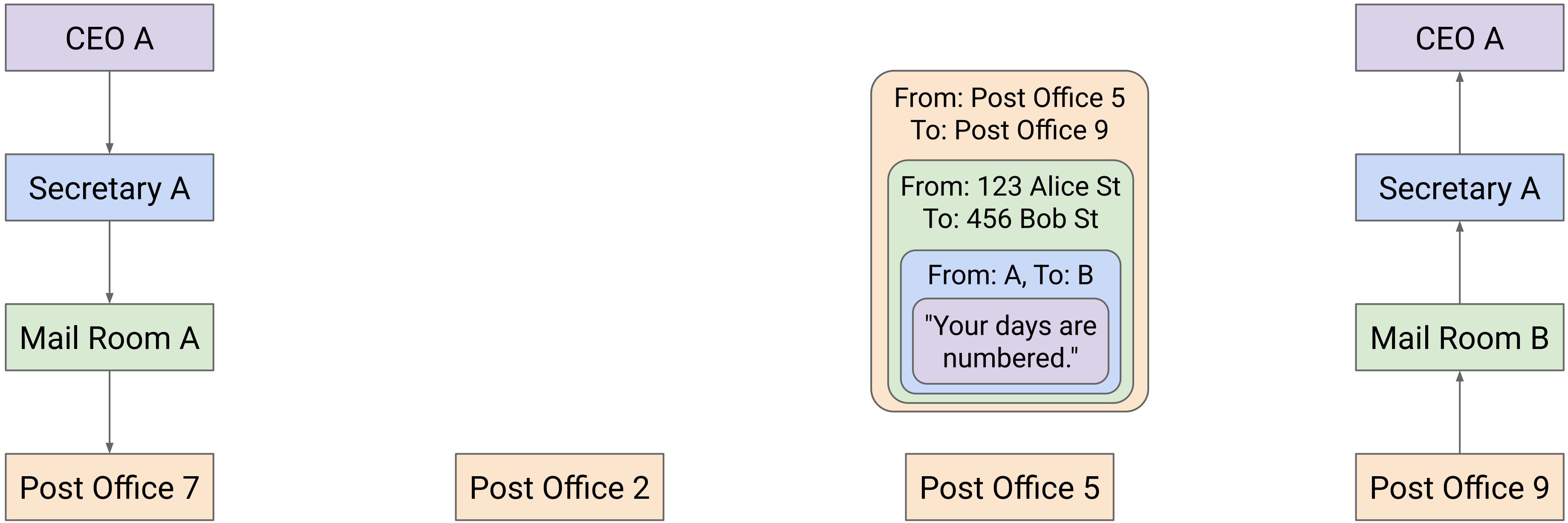
Postal Analogy

The letter might hop across multiple post offices.
Each post office unwraps the box, revealing the envelope inside.
Then, it puts the letter in a new box, destined for the next post office.



Postal Analogy

The letter might hop across multiple post offices.
Each post office unwraps the box, revealing the envelope inside.
Then, it puts the letter in a new box, destined for the next post office.

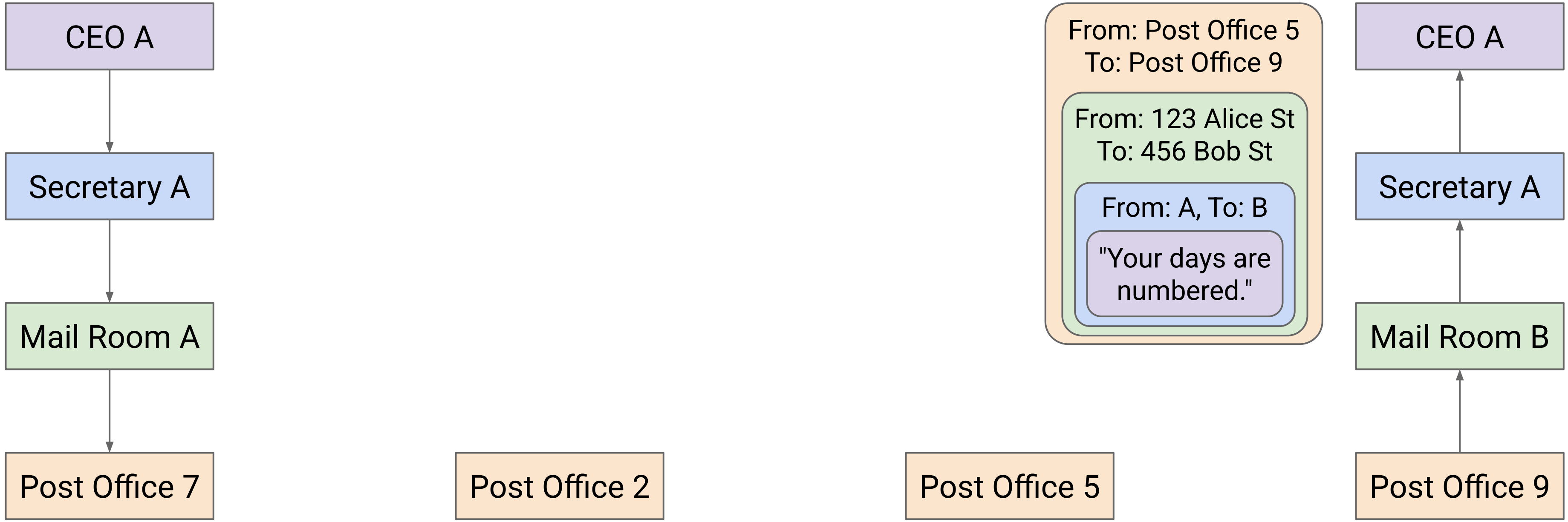


Postal Analogy

The letter might hop across multiple post offices.

Each post office unwraps the box, revealing the envelope inside.

Then, it puts the letter in a new box, destined for the next post office.

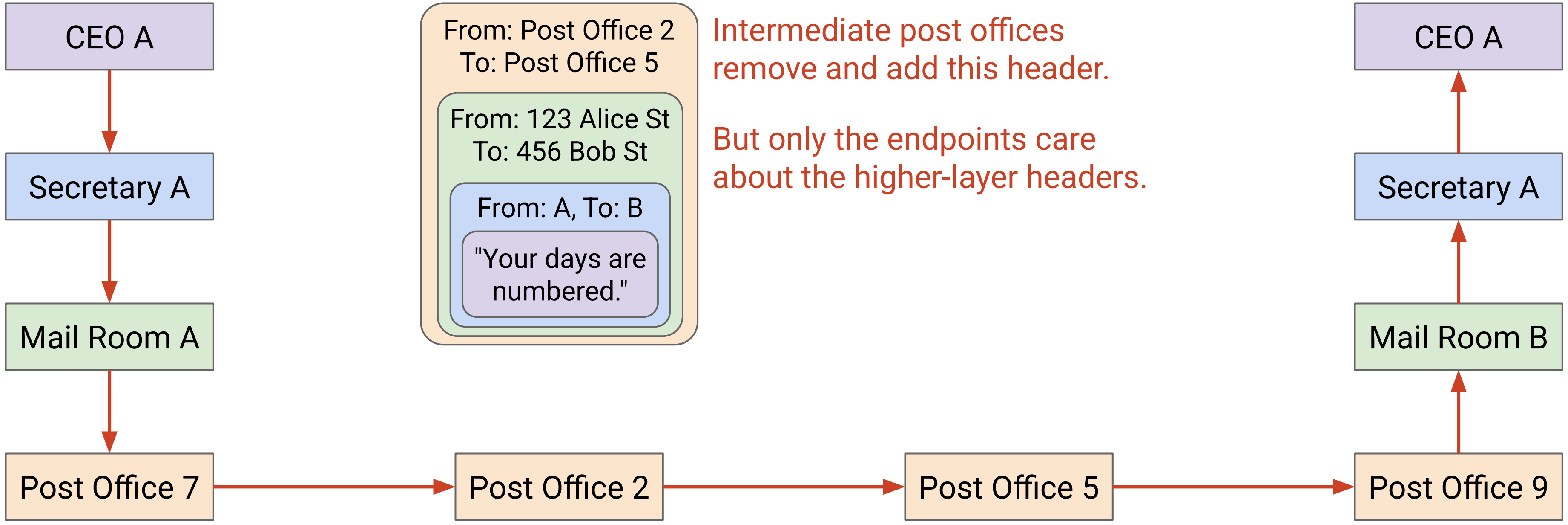


Postal Analogy

The letter might hop across multiple post offices.

Each post office unwraps the box, revealing the envelope inside.

Then, it puts the letter in a new box, destined for the next post office.



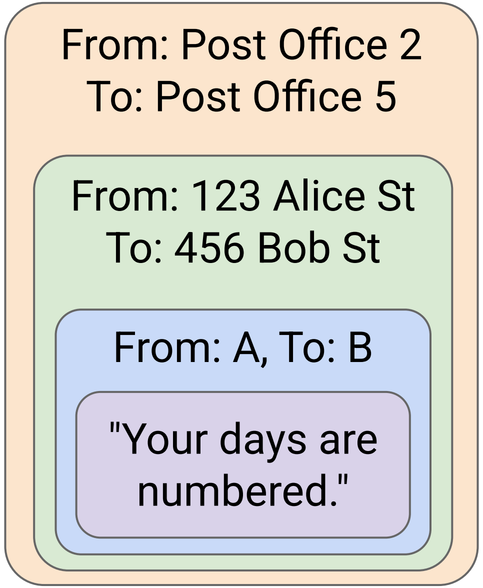
Intermediate post offices remove and add this header.

But only the endpoints care about the higher-layer headers.

Postal Analogy

Recall: Different layers use different addressing schemes.

- Each addressing scheme only makes sense to the protocol at that layer.



Layer 2 header: Destination is the next intermediate post office.

Layer 3 header: Destination is always the actual endpoint.

Layer 4 header: A and B identify specific people in the endpoint (inside the building).

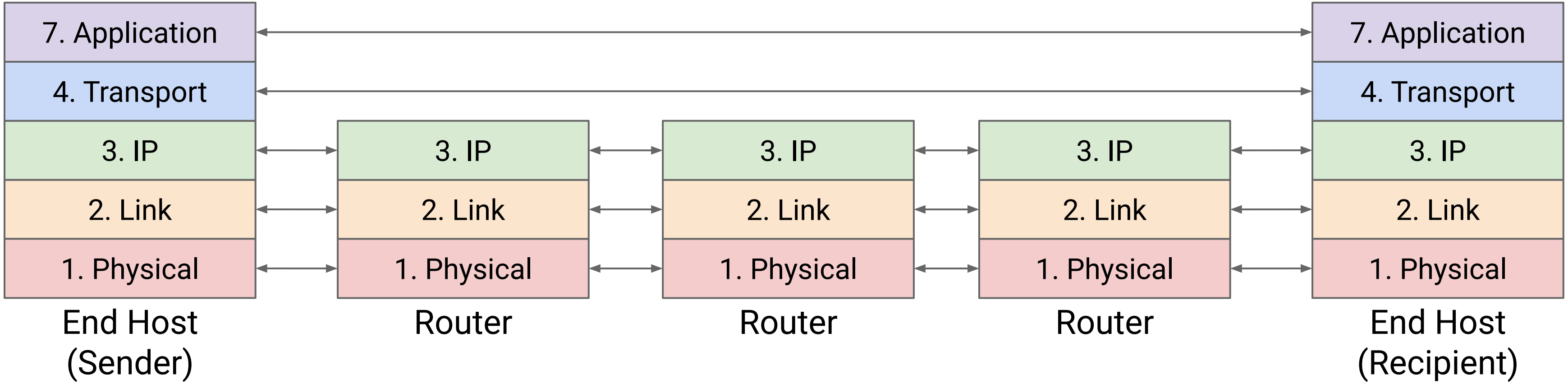
Implementing Layers at Routers and End Hosts

End hosts implement all the layers.

- Must take message, and wrap headers all the way down to bits on the wire.

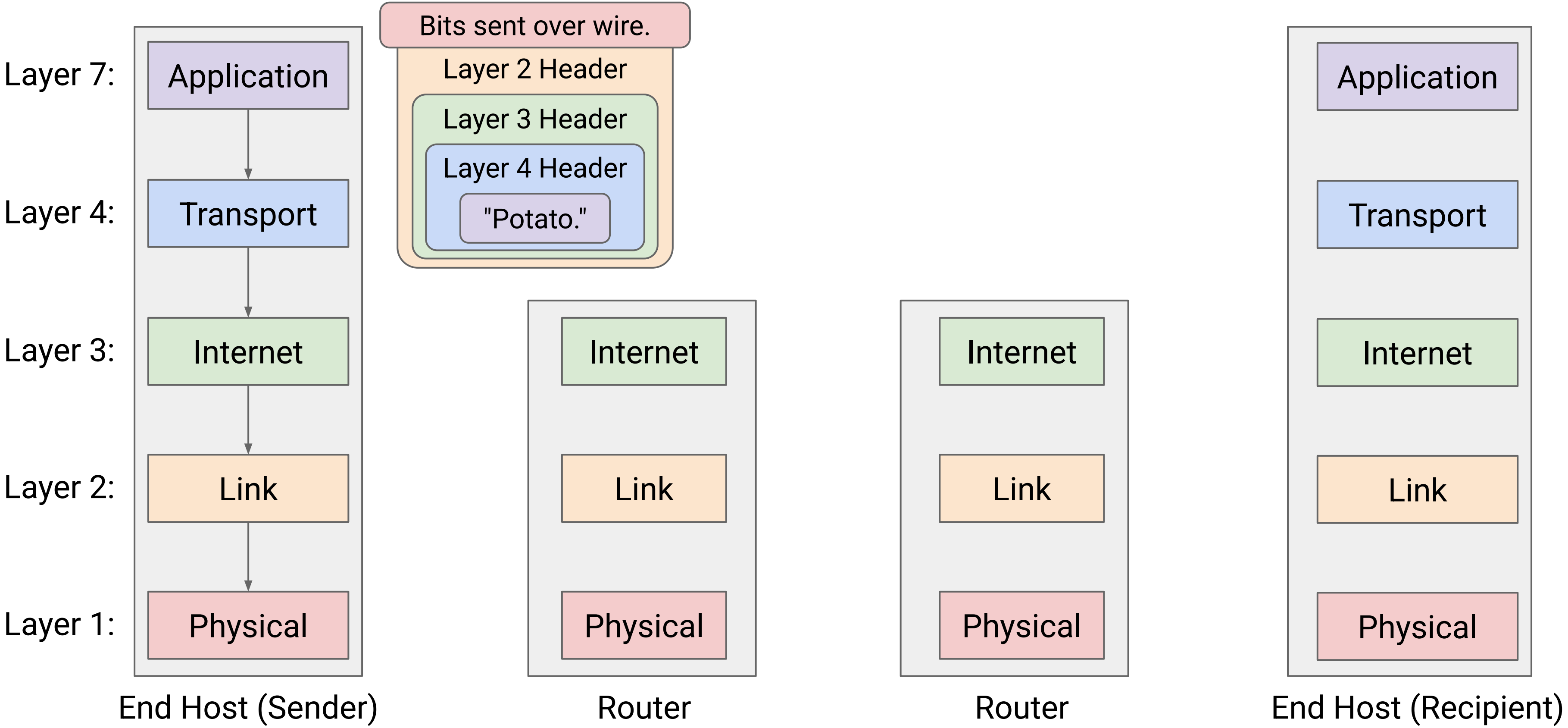
Routers only implement Layers 1–3.

- Must parse the packet (1, 2) and forward to the next router for global delivery (3).
- Routers don't support reliable delivery (4).
- Routers don't care about the application data (7).



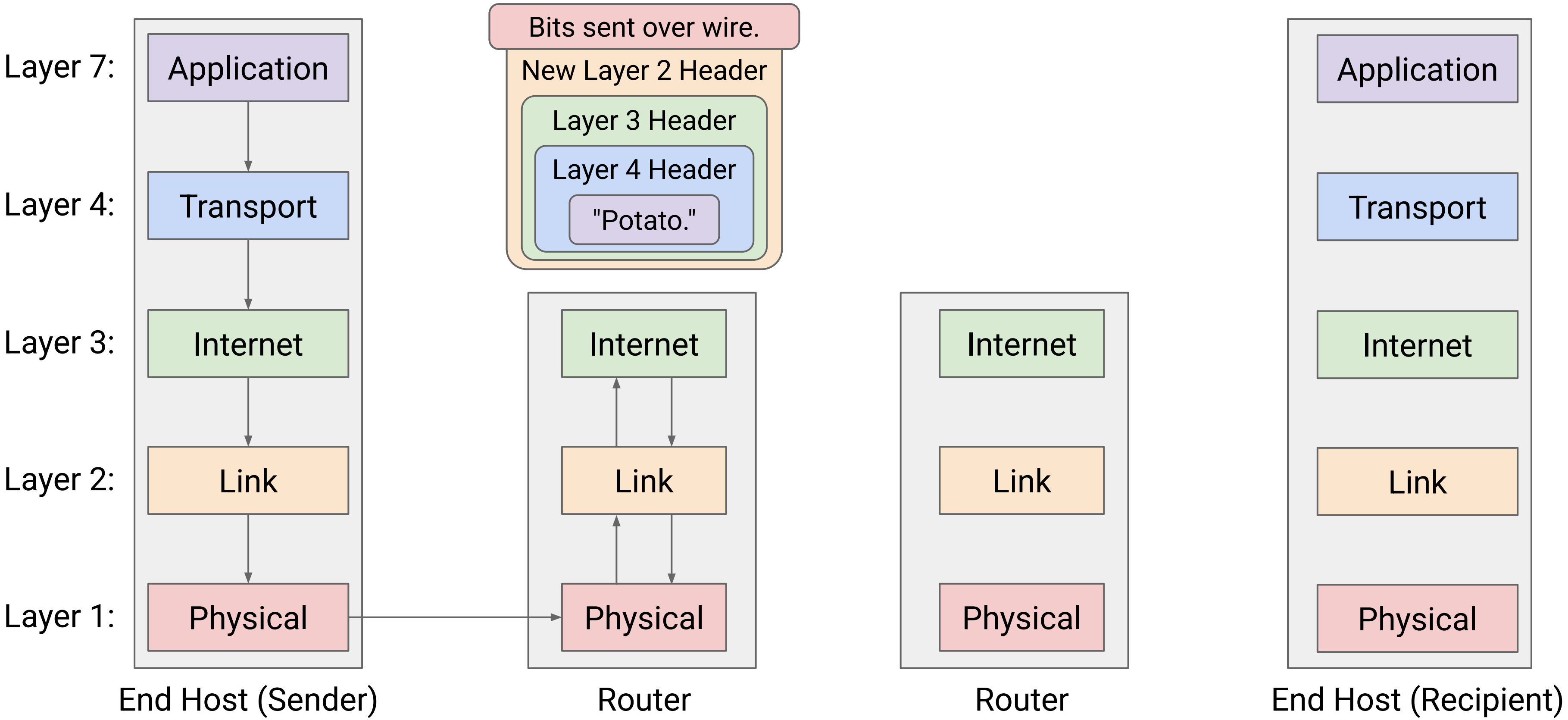
Multiple Headers with Routers

Routers unwrap the Layer 2 header, and add a new Layer 2 header for the next hop.



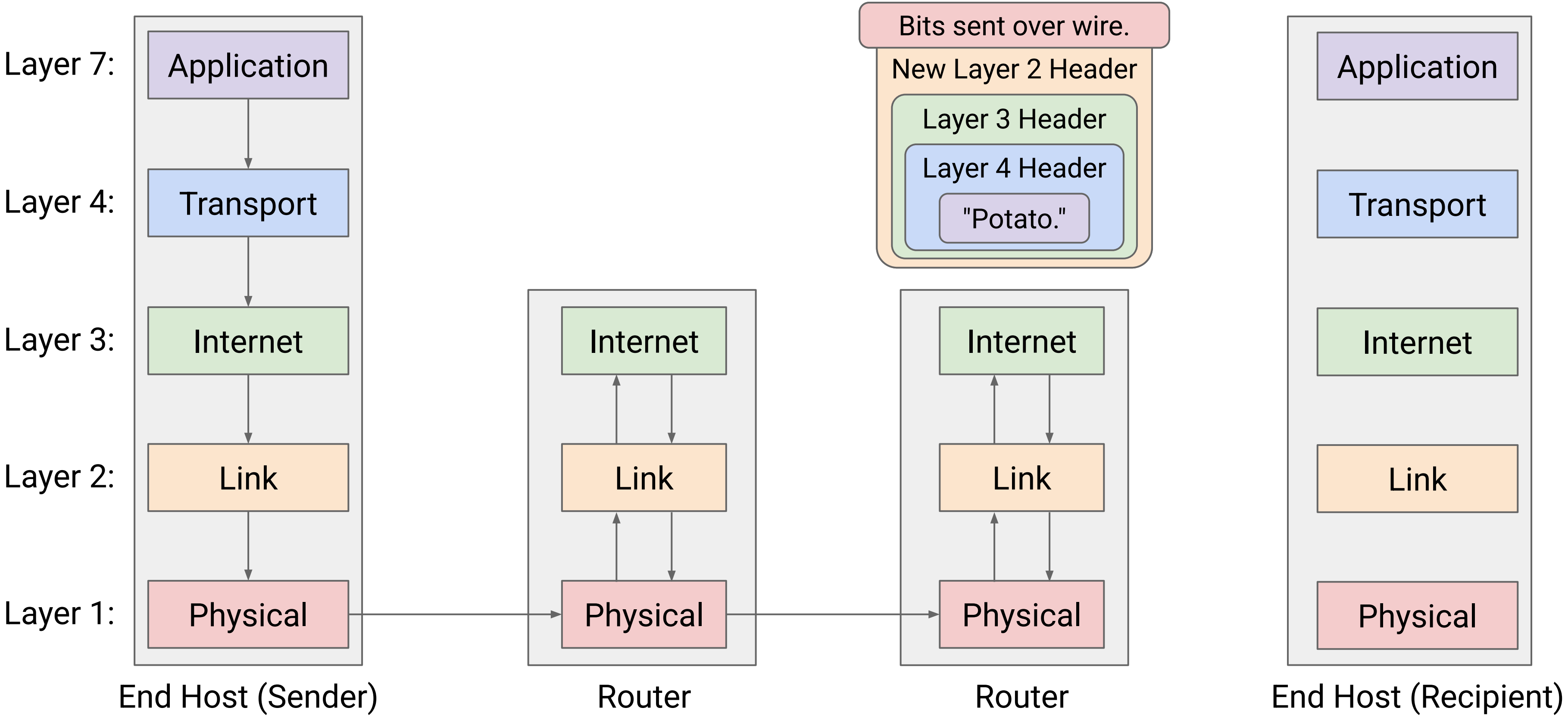
Multiple Headers with Routers

Routers unwrap the Layer 2 header, and add a new Layer 2 header for the next hop.



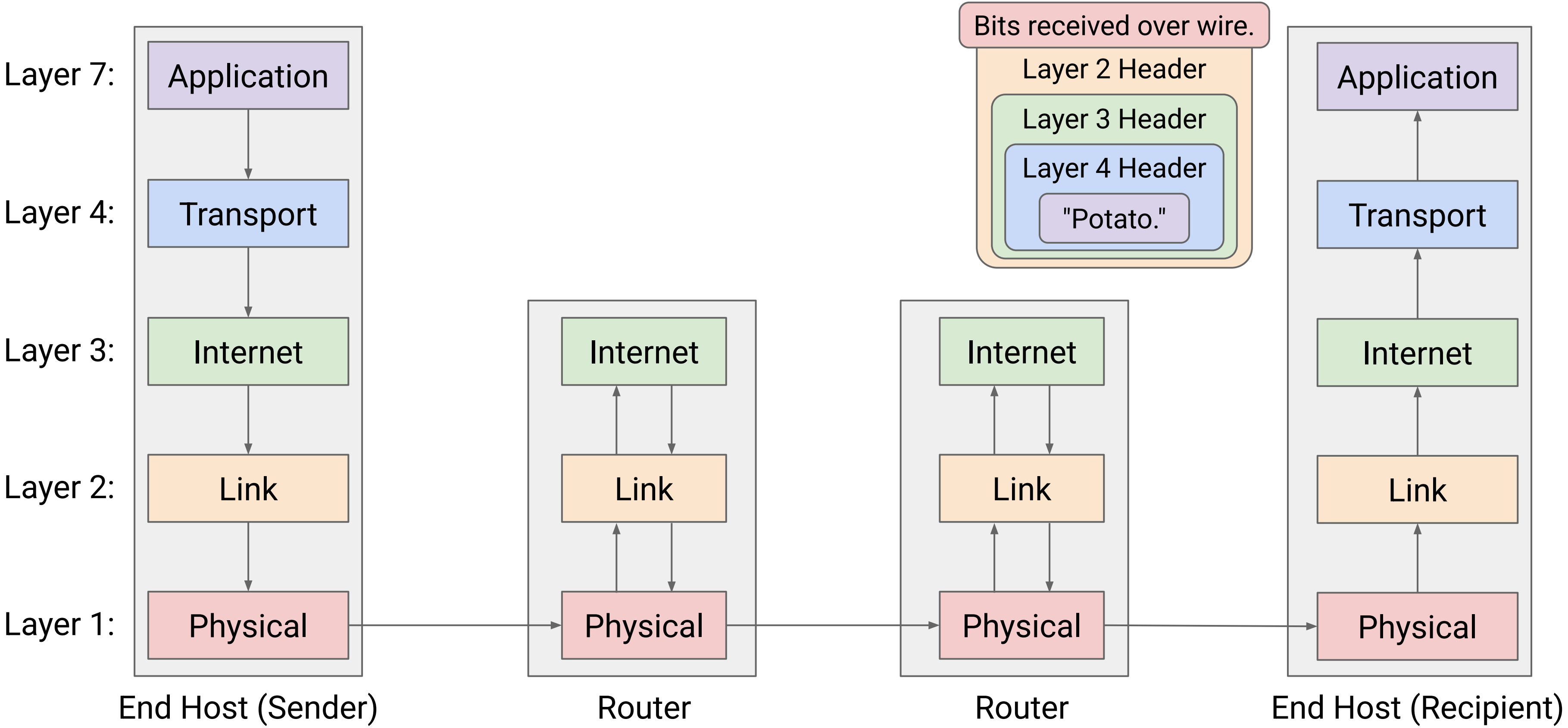
Multiple Headers with Routers

Routers unwrap the Layer 2 header, and add a new Layer 2 header for the next hop.



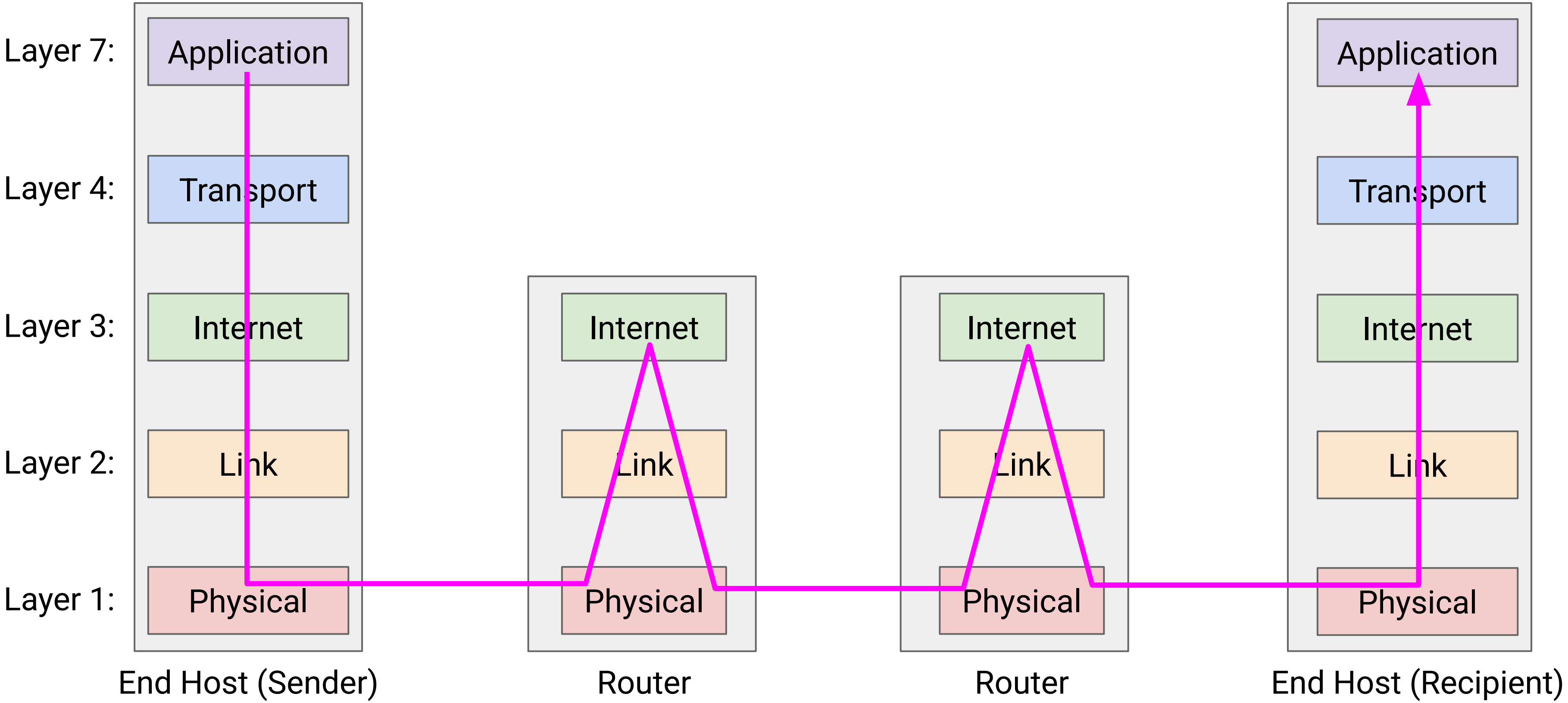
Multiple Headers with Routers

Routers unwrap the Layer 2 header, and add a new Layer 2 header for the next hop.



Multiple Headers with Routers

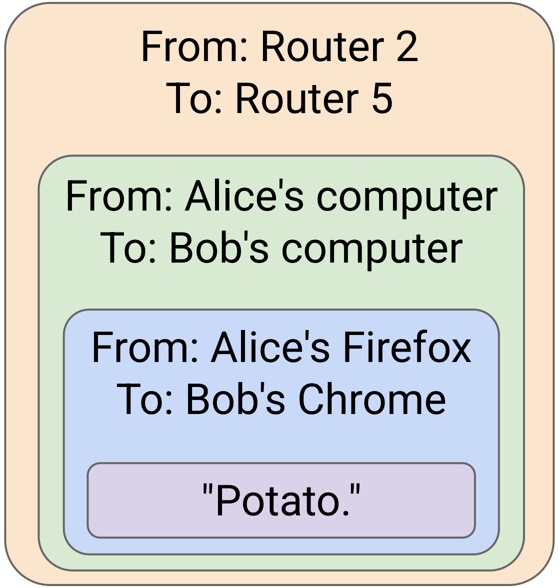
Routers unwrap the Layer 2 header, and add a new Layer 2 header for the next hop.



Postal Analogy

Recall: Different layers use different addressing schemes.

- Each addressing scheme only makes sense to the protocol at that layer.



Layer 2 header: Destination is the next intermediate router.

Layer 3 header: Destination is always the end host.

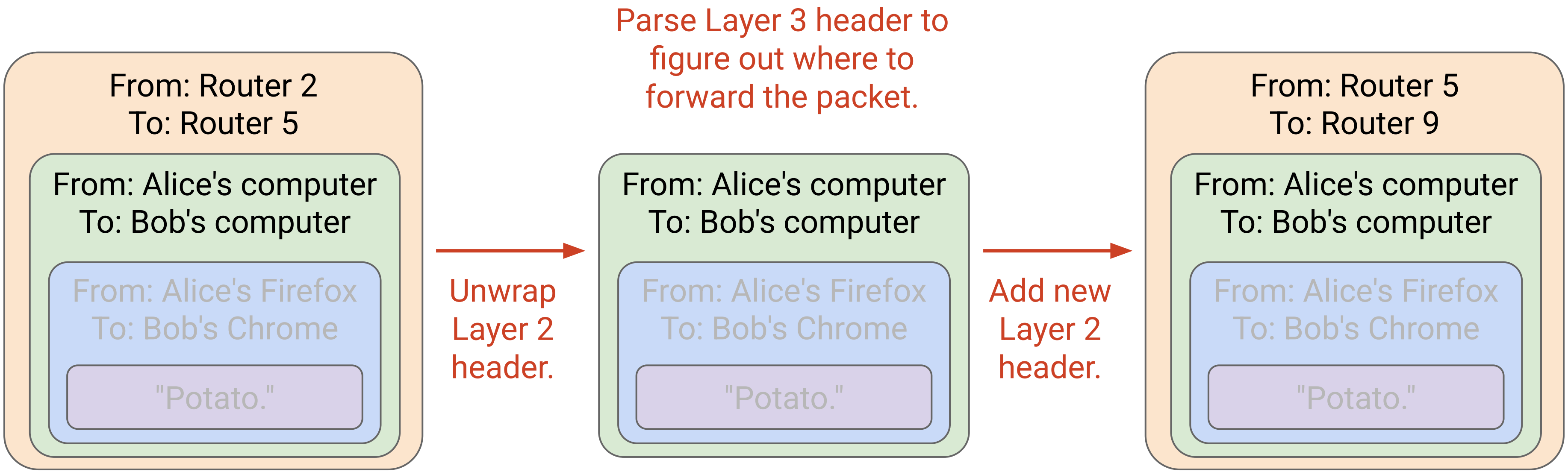
Layer 4 header: Identifies specific application on the end host.

Postal Analogy

Routers don't care about Layer 4 and Layer 7.

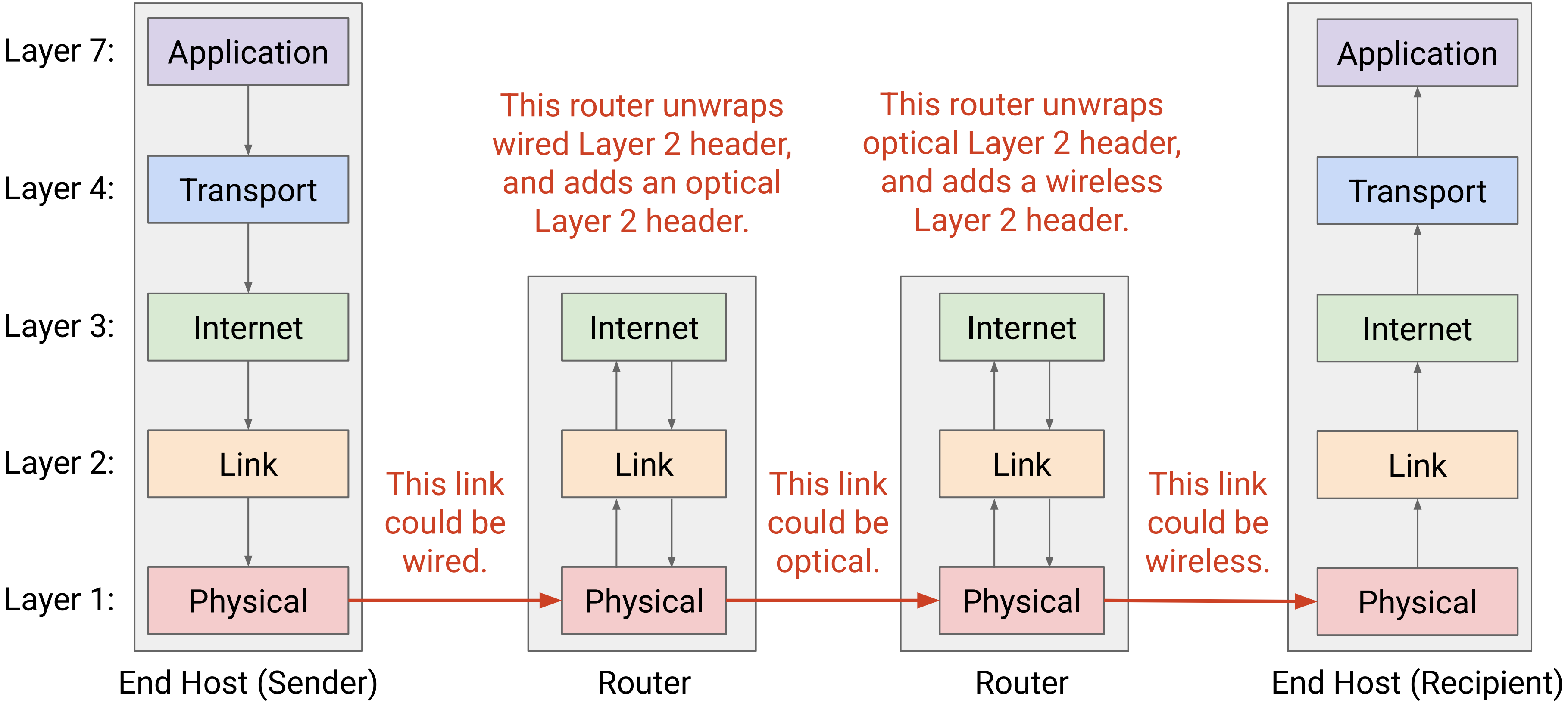
Router parses Layers 1–3 to determine where to forward the packet.

Router unwraps Layer 2 header, and adds a new Layer 2 header for the next hop.



Multiple Headers with Routers

Each hop could use a different Layer 2 protocol.



Summary: Layers of the Internet

- The Internet is built with layers of abstraction.
- Headers are added as the packet moves down the stack, and unwrapped as the packet moves up the stack.
- Hosts parse headers for Layers 1–7.
  Routers parse headers for Layers 1–3.

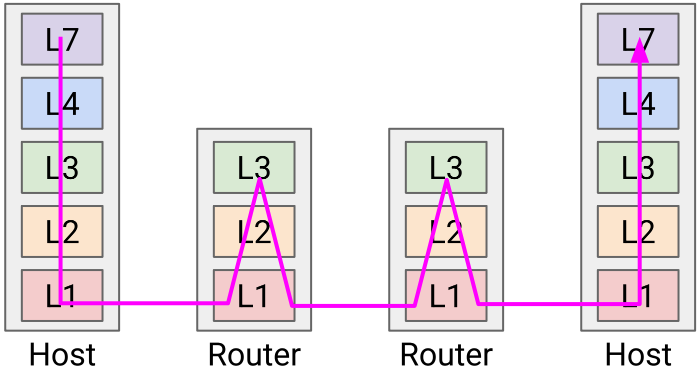
Layer 7: Application

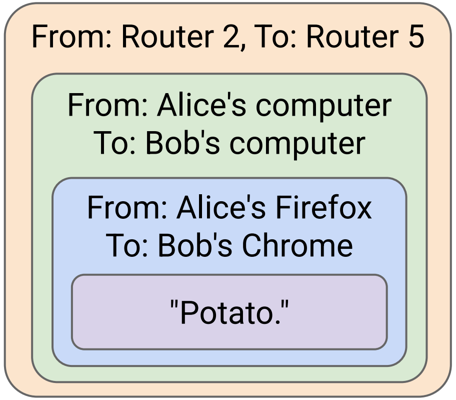
Layer 4: Transport

Layer 3: Internet

Layer 2: Link

Layer 1: Physical





L2 header: Destination is the next intermediate router.

L3 header: Destination is always the end host.

L4 header (port): Identifies specific application on the end host.