Remainder of this lecture and week

- Designing the Internet: a top-down approach
- In the process, discuss a few enduring ideas:
  - Layering
  - The end-to-end principle
  - Fate sharing

How do you solve a problem?

1. Define the problem (and why you're solving it!)
2. Decompose it (into tasks and abstractions)
3. Assign tasks to entities (who does what)

The Internet's problem definition

- Support the transfer of data between endhosts
- ... across multiple networks
  - The Internet
- More on this in our next lecture!

How do you solve a problem?

1. Define the problem (and why you're solving it!)
2. Decompose it (into tasks and abstractions)
3. Assign tasks to entities (who does what)

Modularity

Modularity based on abstraction is the way things are done

– Barbara Liskov, Turing lecture

What is Modularity?

- Decomposing systems into smaller units
  - Providing a “separation of concerns”
- Plays a crucial role in computer science…
- The trick is to find the *right* modularity
  - Our exercise in today’s lecture

Network System Modularity

- The need for modularity still applies
  - **And is even more important!** ***(why?)***
- Normal modularity organizes code
- But network implementations are not just distributed across many lines of code…
  - Also distributed across many machines (hosts, routers)
  - … *and* different players (clients, server, ISPs)

How do we decompose the job of transferring data between end-hosts?

Inspiration…

- **CEO A writes letter to CEO B**
  - Folds letter and hands it to administrative aide
- **Aide:**
  - Puts letter in envelope with CEO B's full name
  - Takes to FedEx
- **FedEx Office**
  - Puts letter in larger envelope
  - Puts name and street address on FedEx envelope
  - Puts package on FedEx delivery truck
- **FedEx delivers to other company**

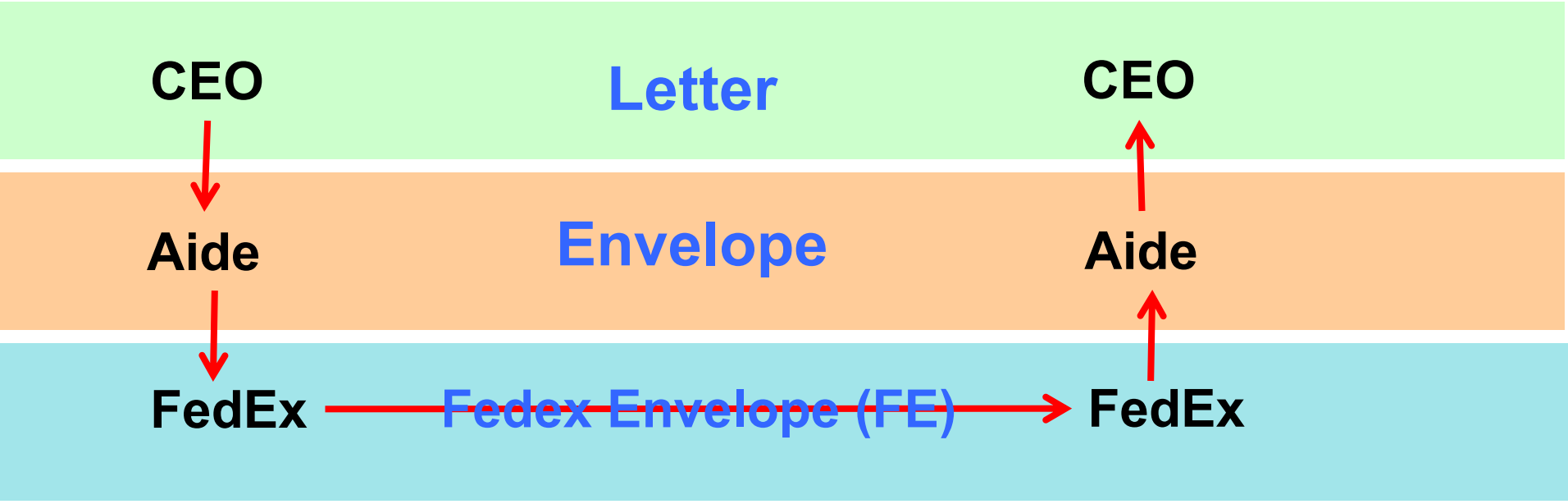
Dear John,

Your days are numbered.

--Pat

The Path of the Letter

- "Peers" understand the same things
- No one else needs to
- Lowest level has most "packaging"



Thought Experiment

- How would ***you*** break the Internet into tasks?
- Do not consider application or control tasks
  - Naming, computing forwarding tables, etc.
- Just focus on what is needed to get packets between processes on different hosts….

Breakdown into Tasks

- Bits across a wire
- Packets across a wire
- Deliver packets across local network
  - Local addresses
- Deliver packets across multiple networks
  - Global addresses
- Deliver data reliably
- Do something with the data

Breakdown into Tasks

- Bits across a wire
- Packets across a wire and local network
  - Local addresses
- Deliver packets across multiple networks
  - Global addresses
- Deliver data reliably
- Do something with the data

In the Internet: organization

Applications

Reliable (or unreliable) data delivery

Best-effort *global* packet delivery

Best-effort local *packet* delivery

Physical transfer of bits

In the Internet: organization

Applications

…built on…

Reliable (or unreliable) data delivery

…built on…

Best-effort *global* packet delivery

…built on…

Best-effort local *packet* delivery

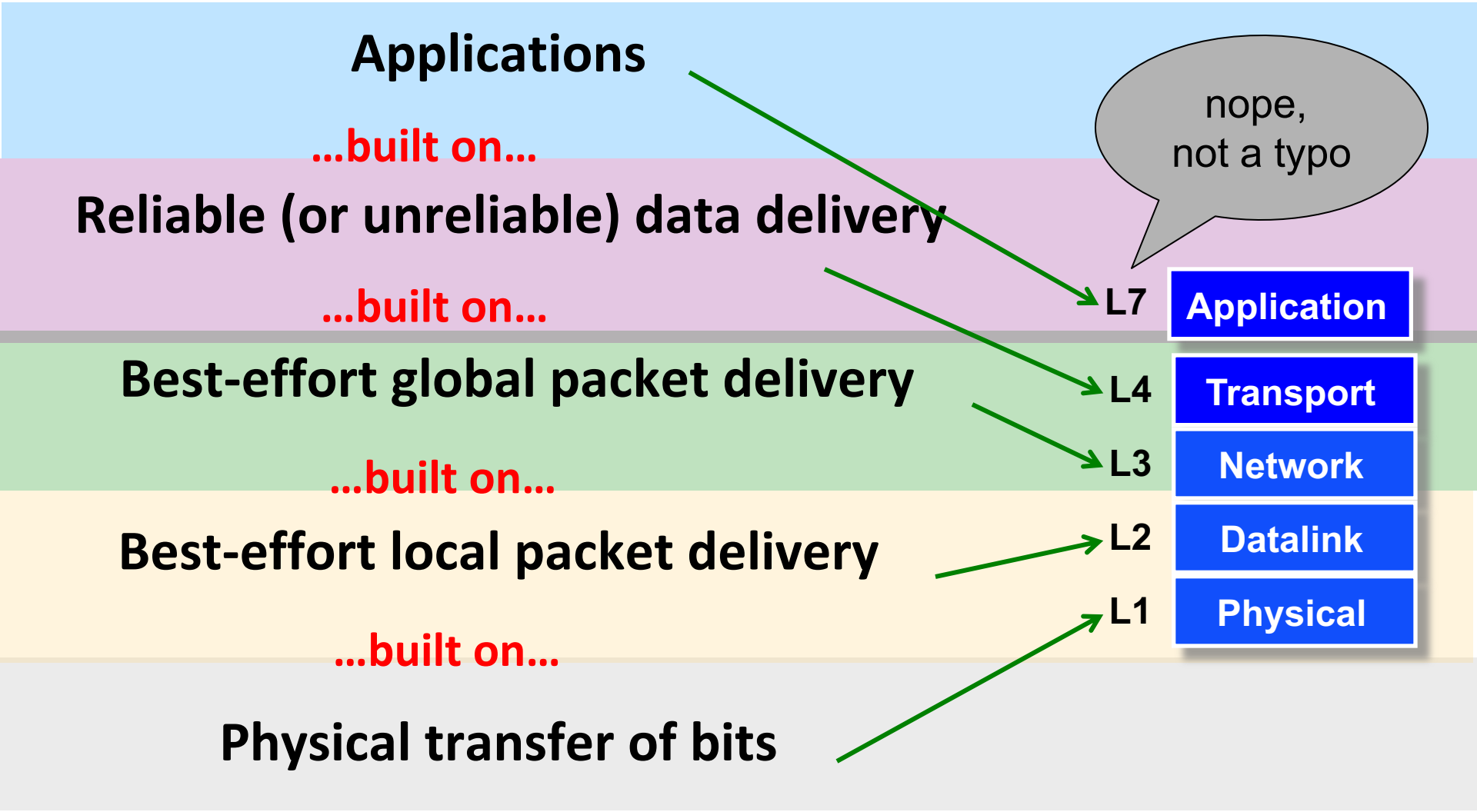
…built on…

Physical transfer of bits

A layered architecture

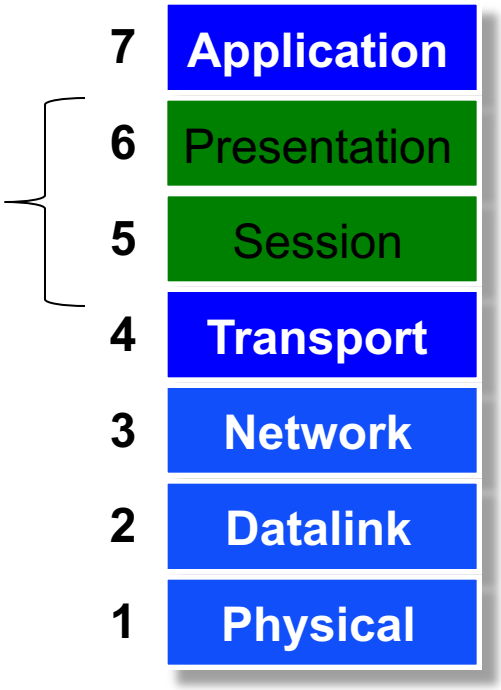
- Layer = a part of a system with well-defined interfaces to other parts
- **One layer interacts only with layer above and layer below**
- Two layers interact only through the interface between them

In the Internet: organization



Ancient history (late 1970s)

The Open Systems Interconnect (OSI) model developed by the International Organization for Standardization (ISO) included two additional layers that are often implemented as part of the application



Questions?

Recall: peers understand the same things

CEO	Letter	CEO
Aide	Envelope	Aide
FedEx	Fedex Envelope (FE)	FedEx

Protocols and Layers

Communication between peer layers on different systems is defined by protocols

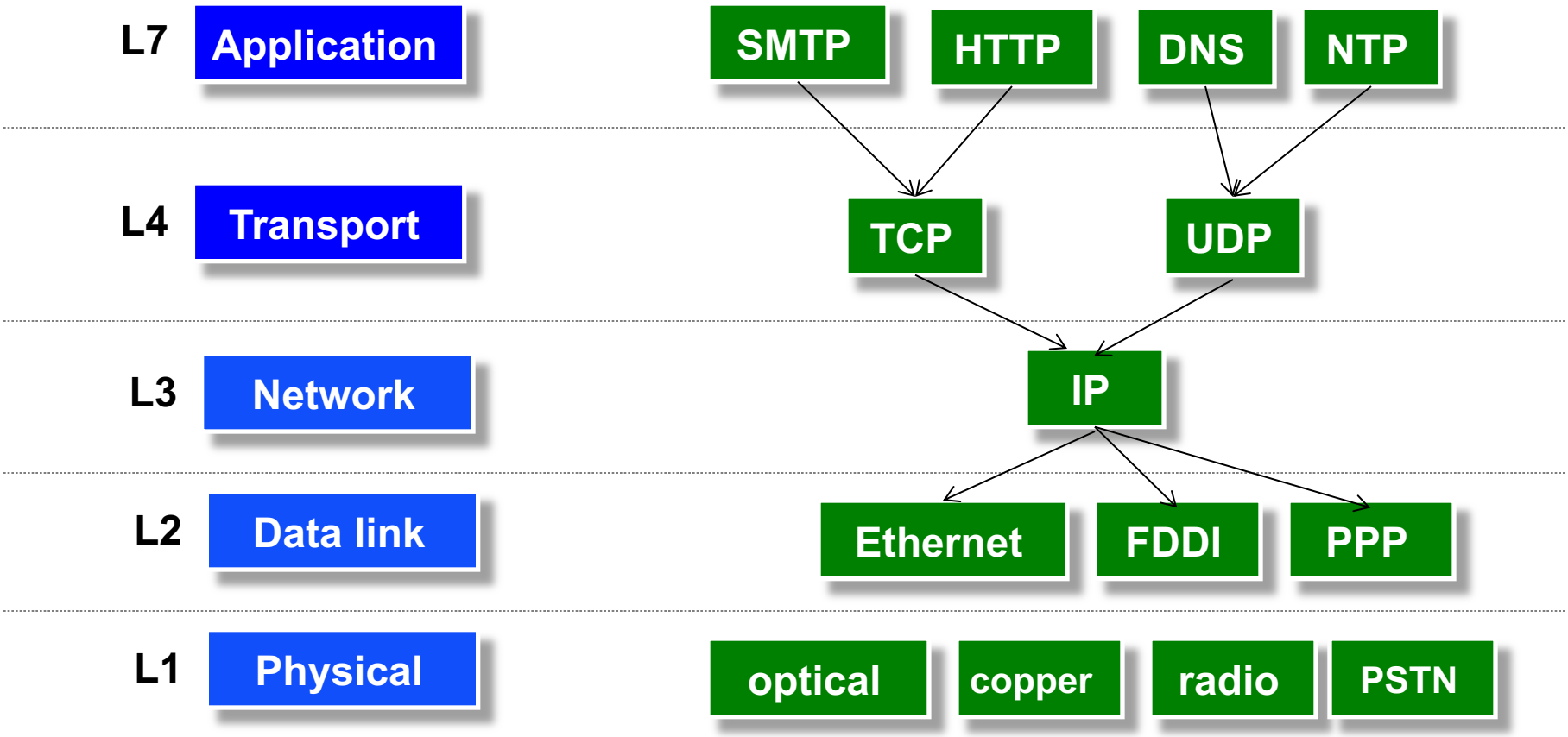
What is a Protocol?

- An agreement between parties on how to communicate
- Defines the syntax of communication
  - Each protocol defines the format of its packet **headers**
    - *e.g. "the first 32 bits carry the destination address"*

What is a Protocol?

- An agreement between parties on how to communicate
- Defines the syntax of communication
- And semantics
  - "first a hullo, then a request…"
  - essentially, a state machine
  - we'll study many protocols later in the semester
- Protocols exist at many levels, hardware and software
  - defined by a variety of standards bodies (IETF, IEEE, ITU)

Protocols at different layers



There is just one network-layer protocol!

Protocols at different layers

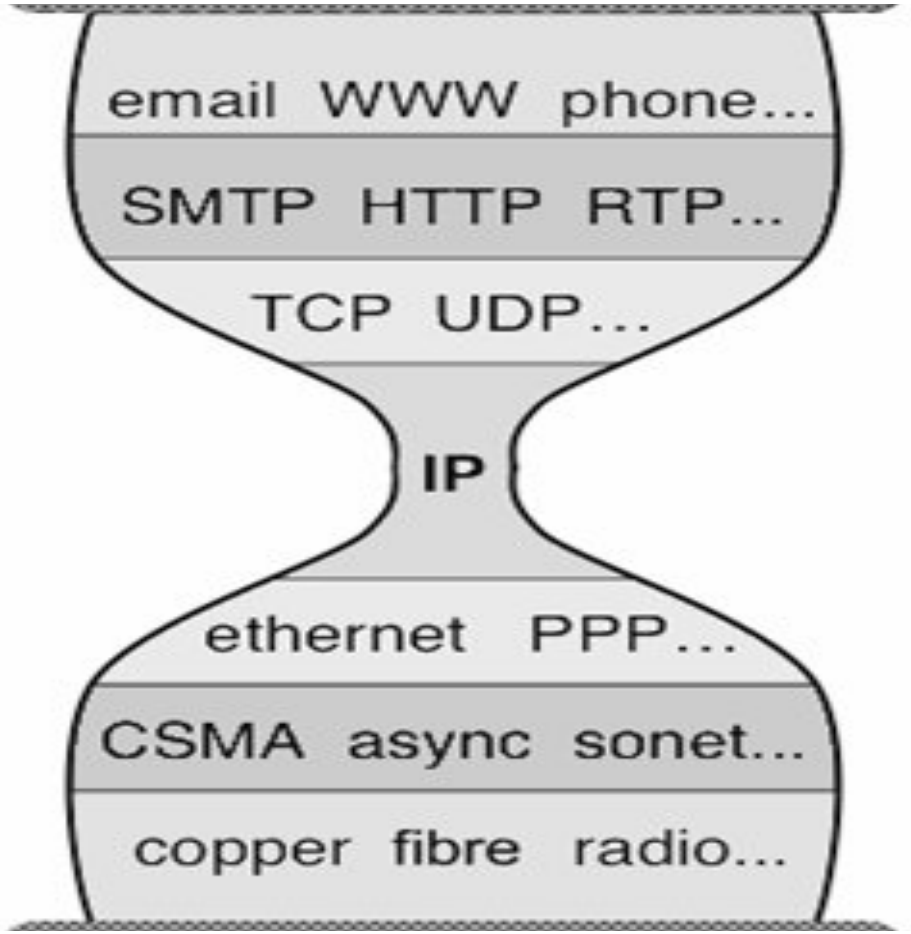
L7 Application

L4 Transport

L3 Network

L2 Data link

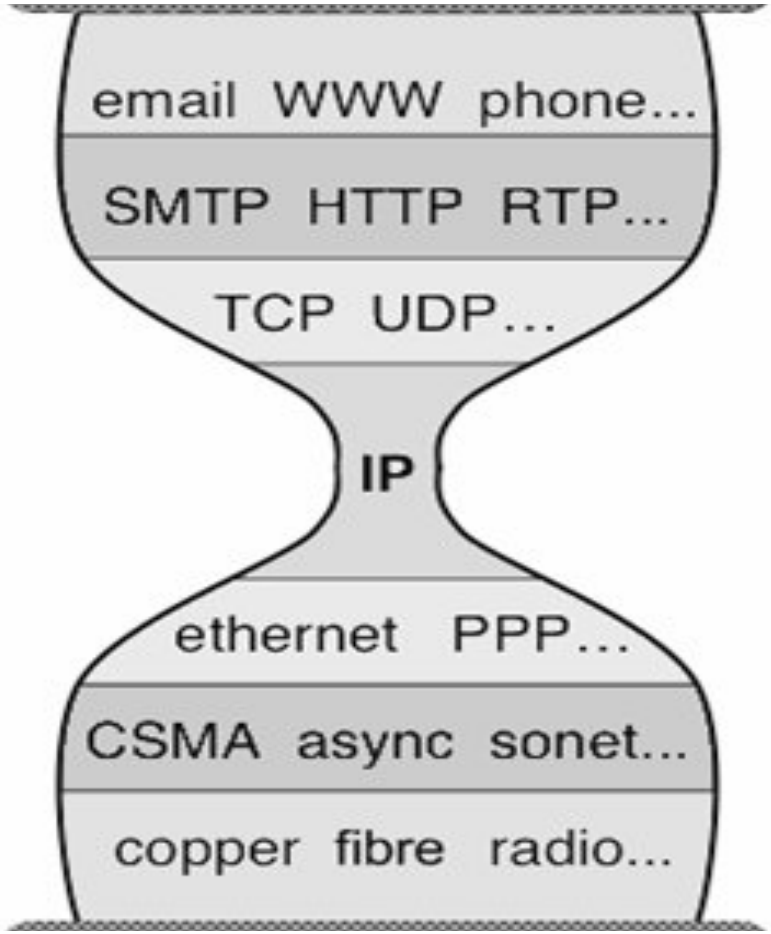
L1 Physical



There is just one network-layer protocol!

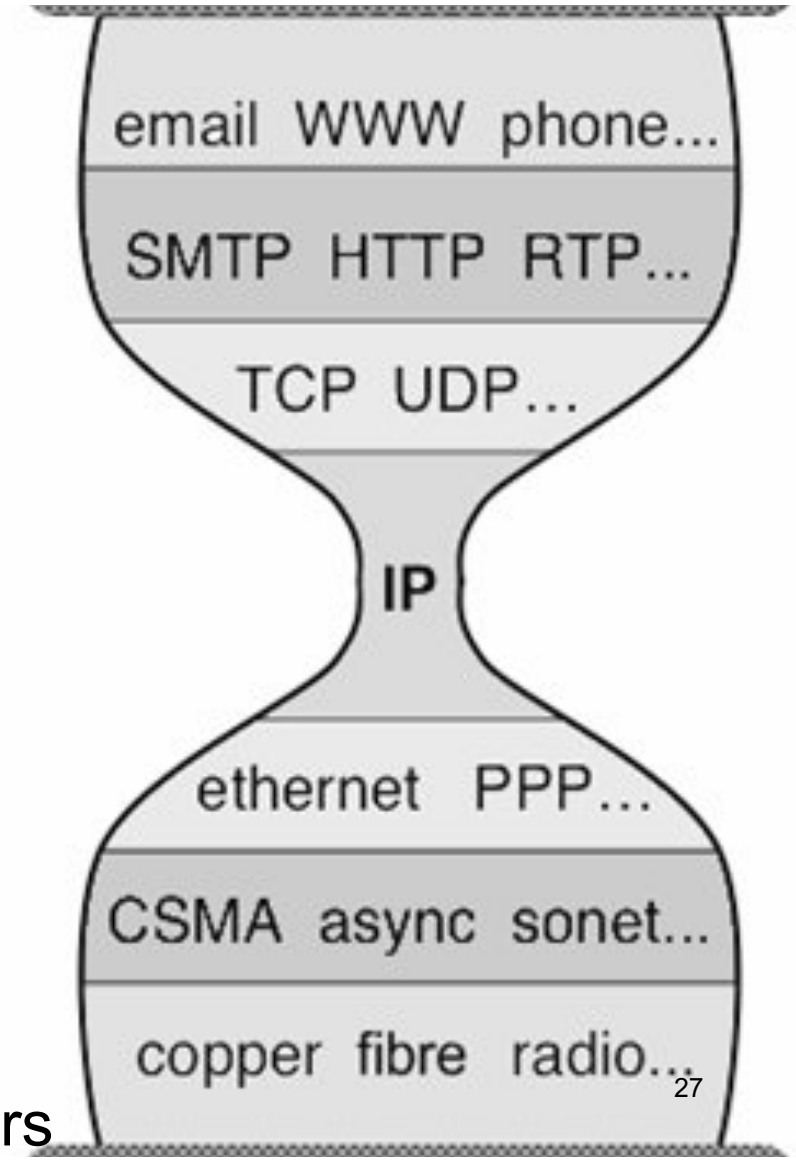
Recap: Three important properties

- Each layer:
  - Depends on layer below
  - Supports layer above
  - Independent of others
- Multiple versions in a layer
  - Interfaces differ somewhat
  - Components pick which lower-level protocol to use
- But only one IP layer
  - Unifying protocol



Why was layering important?

- Innovation at most levels
  - Applications (lots)
  - Transport (some)
  - Network (few)
  - Physical (lots)
- Innovation proceeded largely in parallel
  - **Payoff of modularity!**
- Pursued by very different communities
  - App developers, chip designers



Questions?

How do you solve a problem?

1. Define the problem (and why you're solving it!)
2. Decompose it (into tasks and abstractions)
3. Assign tasks to entities (who does what)

Distributing Layers Across Network

- Layers are simple if only on a single machine
  - Just stack of modules interacting with those above/below
- But we need to implement layers across machines
  - Hosts
  - Routers (switches)
- What gets implemented where?

What gets implemented at the end host?

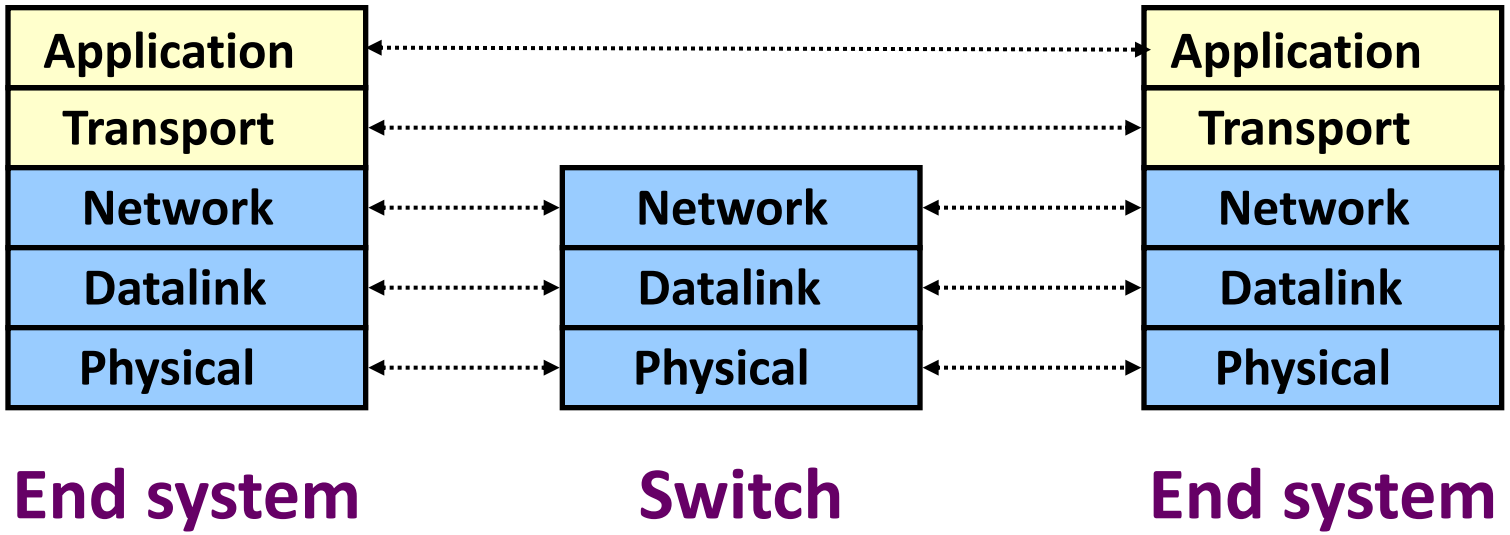
- Bits arrive on wire, must make it up to application
- Therefore, all layers must exist at host!

What gets implemented in the network?

- Bits arrive on wire → physical layer (L1)
- Packets must be delivered across links and local networks → datalink layer (L2)
- Packets must be delivered between networks for global delivery → network layer (L3)
- The network does not support reliable delivery
  - Transport layer (and above) ***not*** supported

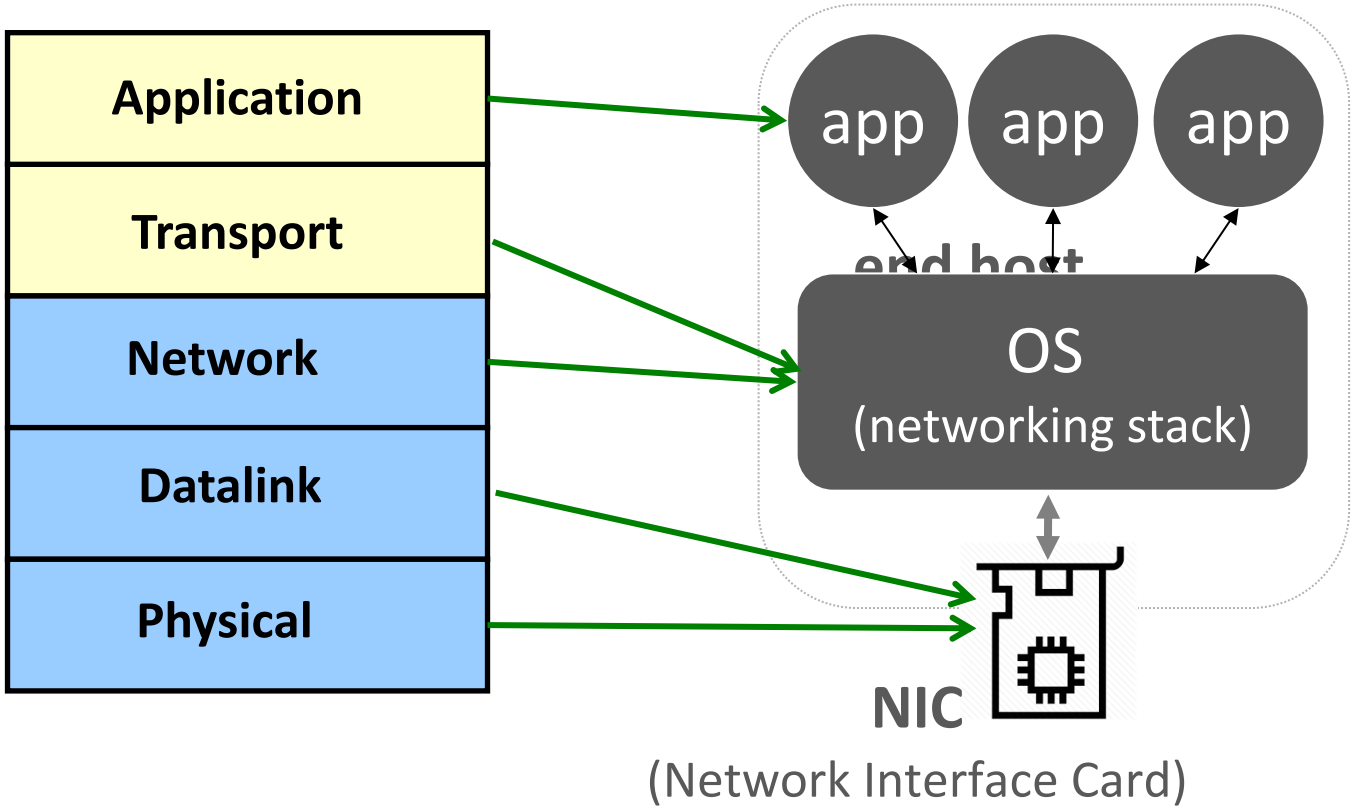
Simple Diagram

- Lower three layers implemented everywhere
- Top two layers implemented only at hosts



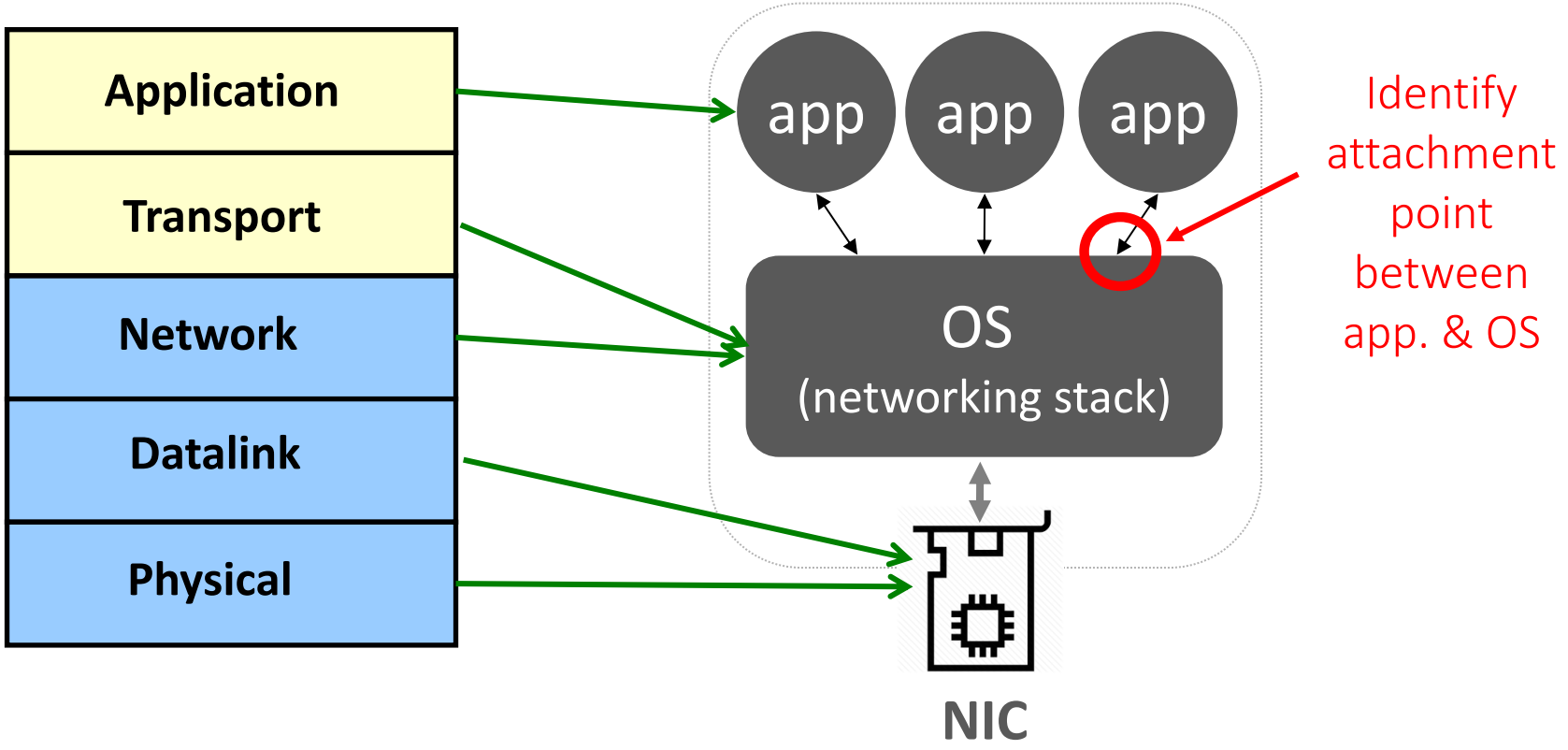
To be continued in lecture 4...

A closer look: end host



Note: addressing within the end host

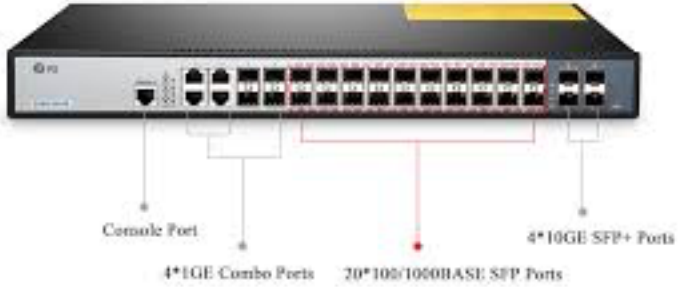
Recall: packet contains the destination host's address



When a packet arrives at the host, how does the OS know which app to send the packet to?

Network “ports”: two types

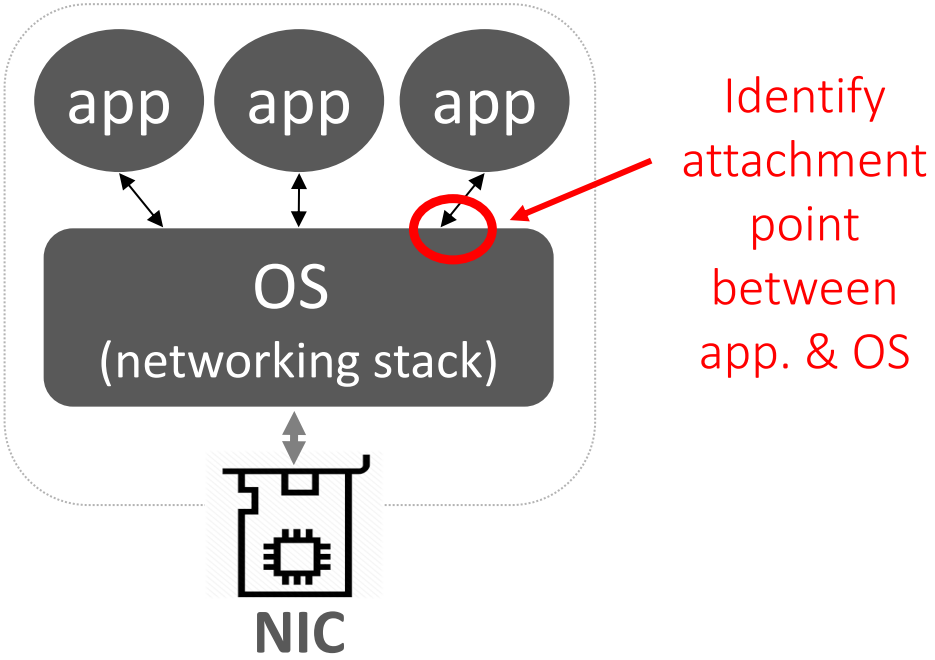
- Switches/routers have **physical ports**:
  - Places where links connect to switches



Network “ports”: two types

- Switches/routers have **physical ports**:
  - Places where links connect to switches

- The OS supports **logical ports**:
  - Place where app connects to OS network stack



Of Sockets and Ports

- **Socket:** an OS mechanism that connects app processes to the networking stack
- When an app wants access to the network, it opens a **socket**, which is associated with a **port**
  - *This is not a physical port, just a logical one*
- The **port number** is used by the OS to direct incoming packets to its associated socket

Details about sockets in next week's section!

Implications for Packet Header

- Packet header must include:
  - Destination host address (used by network to reach host)
  - Destination port (used by end-host OS to reach app)

- When a packet arrives at the destination end-host, it is delivered to the socket (process) associated with the packet's destination port

OS Network Stack Is An Intermediary

- Application has very clear task (w.r.t. network)
  - Thinks about data
- NIC/driver has very clear task
  - Thinks about packets
- Network stack in the intermediary between them
  - Translates between their abstractions

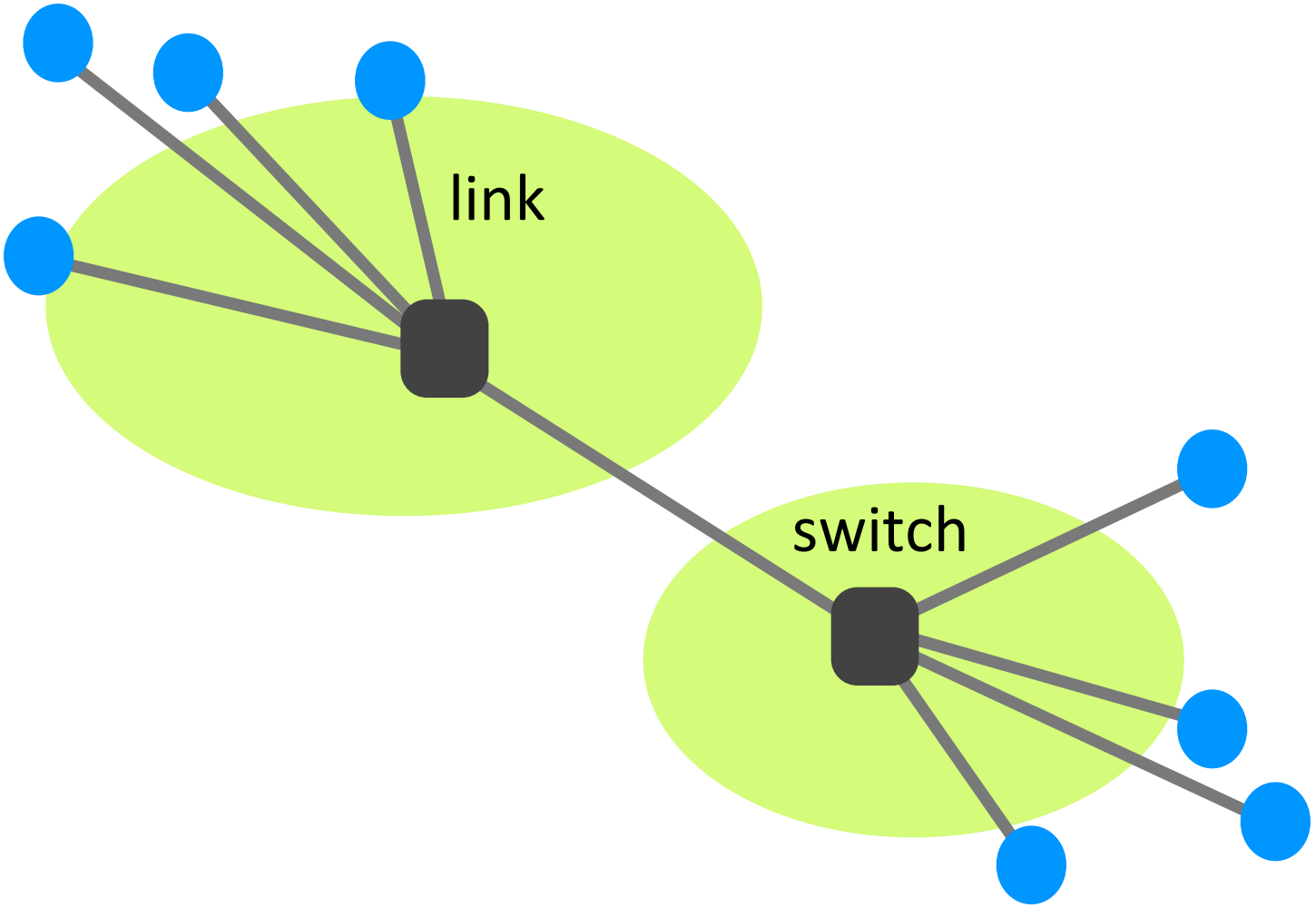
Recap: layers at the end host

- **Application layer (L7)**
  - Part of the app: browser, mail client,…
- **Transport and network layer (L3, L4)**
  - typically part of the OS
- **Datalink and physical layer (L1, L2)**
  - hardware/firmware/drivers

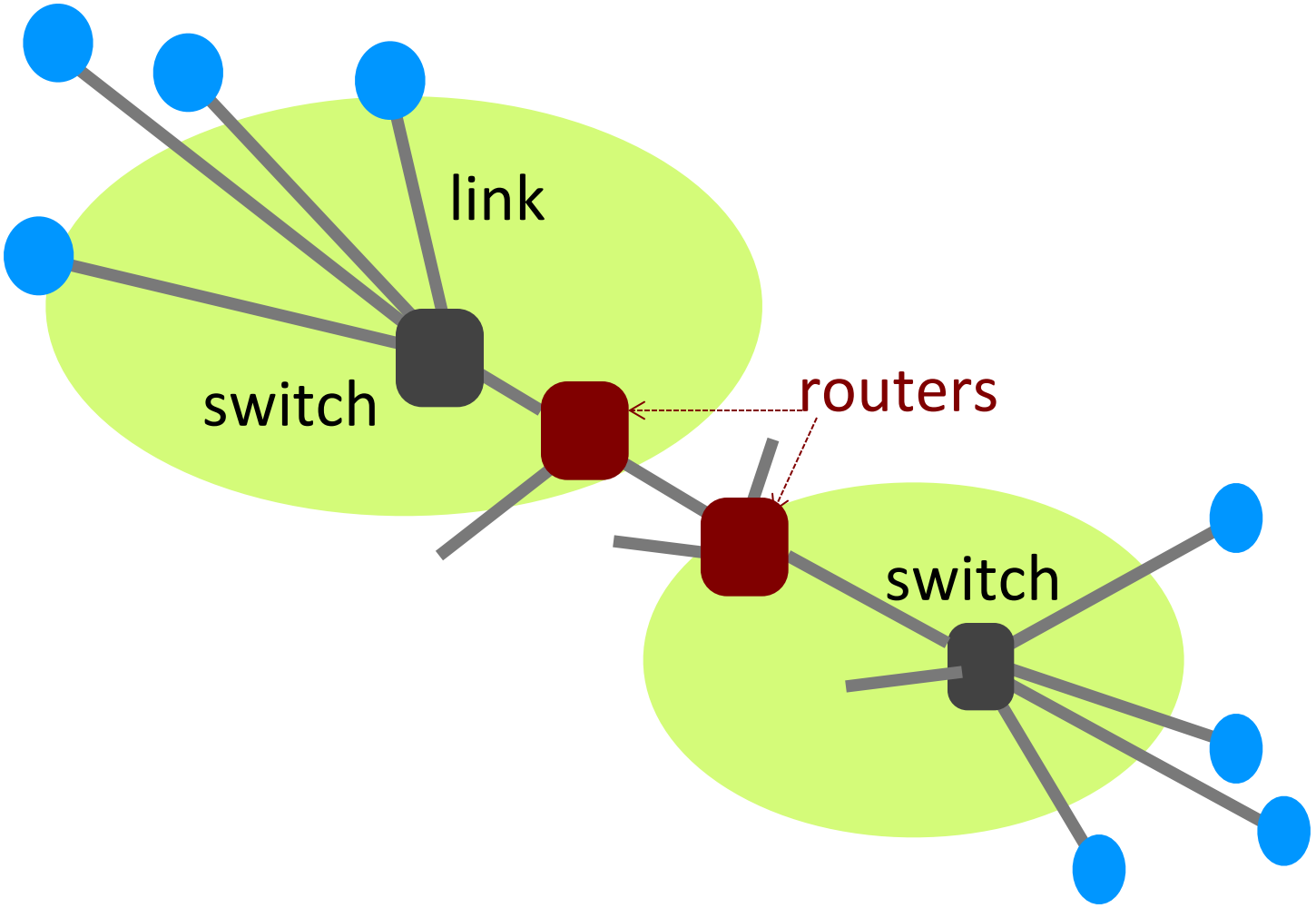
Recall: what gets implemented in the network?

- Bits arrive on wire → physical layer (L1)
- Packets must be delivered across links and local networks → datalink layer (L2)
- Packets must be delivered between networks for global delivery → network layer (L3)

A closer look: network



A closer look: network



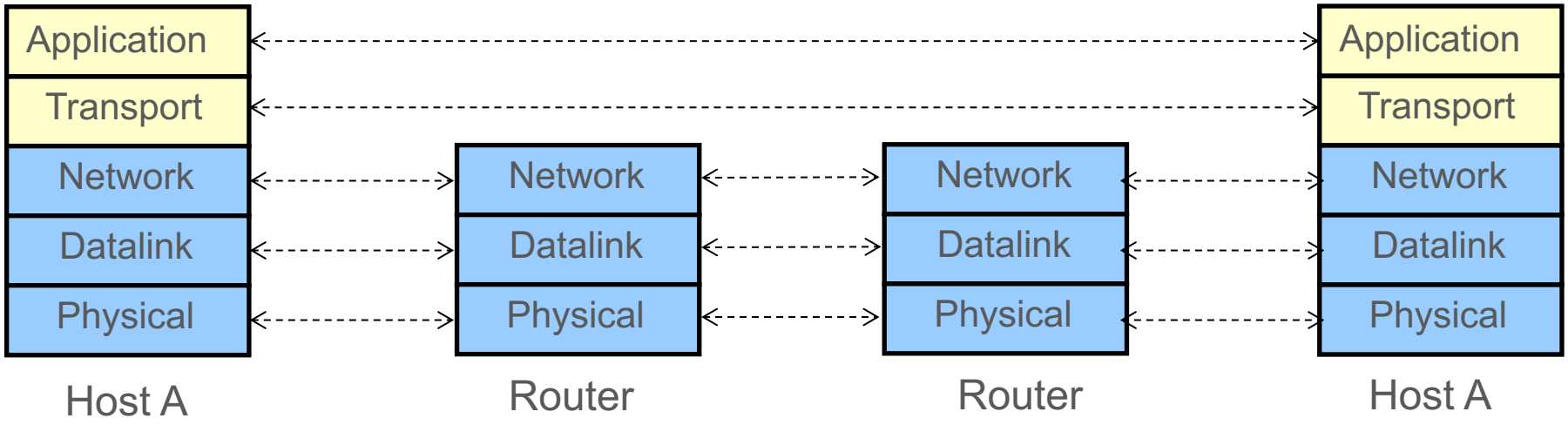
Switches *vs.* Routers

- Switches do what routers do but don't participate in global delivery, just local delivery
  - switches only need to support L1, L2
  - routers support L1-L3

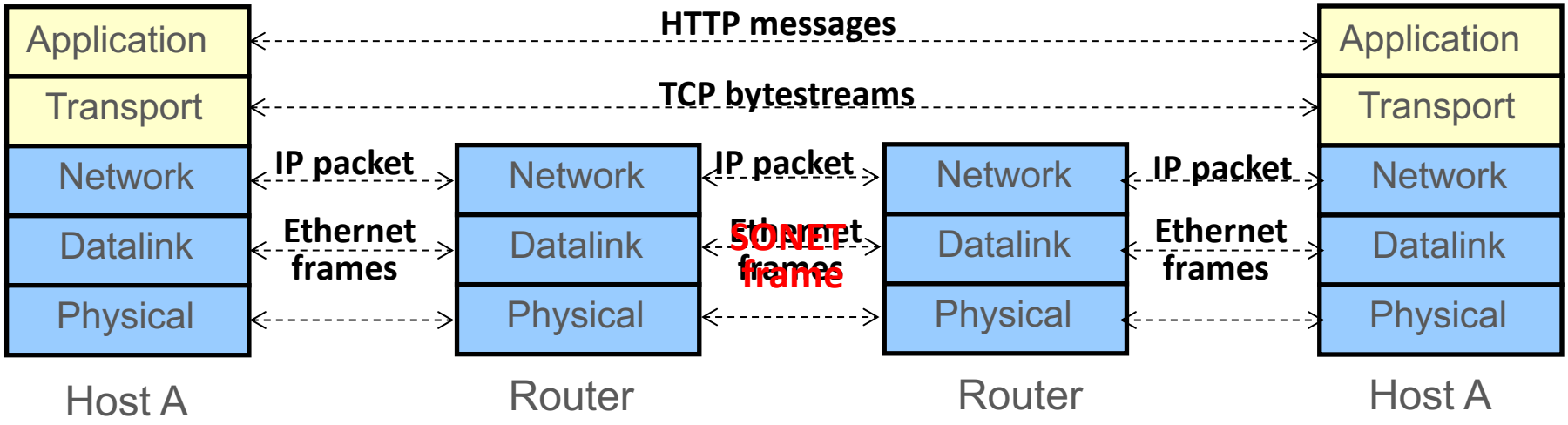
- In general, won't focus on the router/switch distinction
  - when I say switch, I almost always mean router
  - almost all switches support network layer these days

Recap: Logical Communication

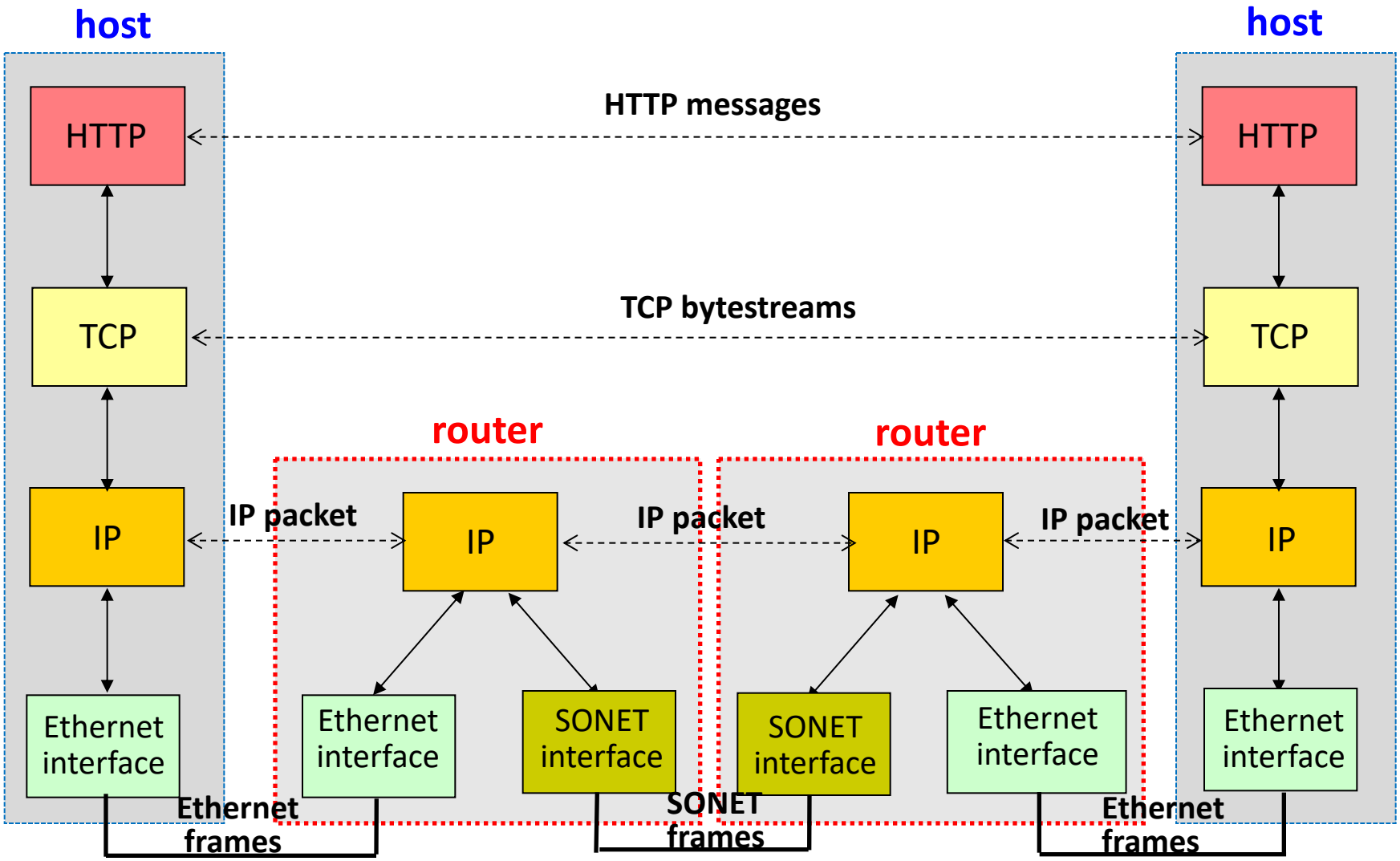
- Layers interact with peer's corresponding layer
- Lower three layers implemented everywhere
- Top two layers implemented only at hosts



Example: simple protocol diagram

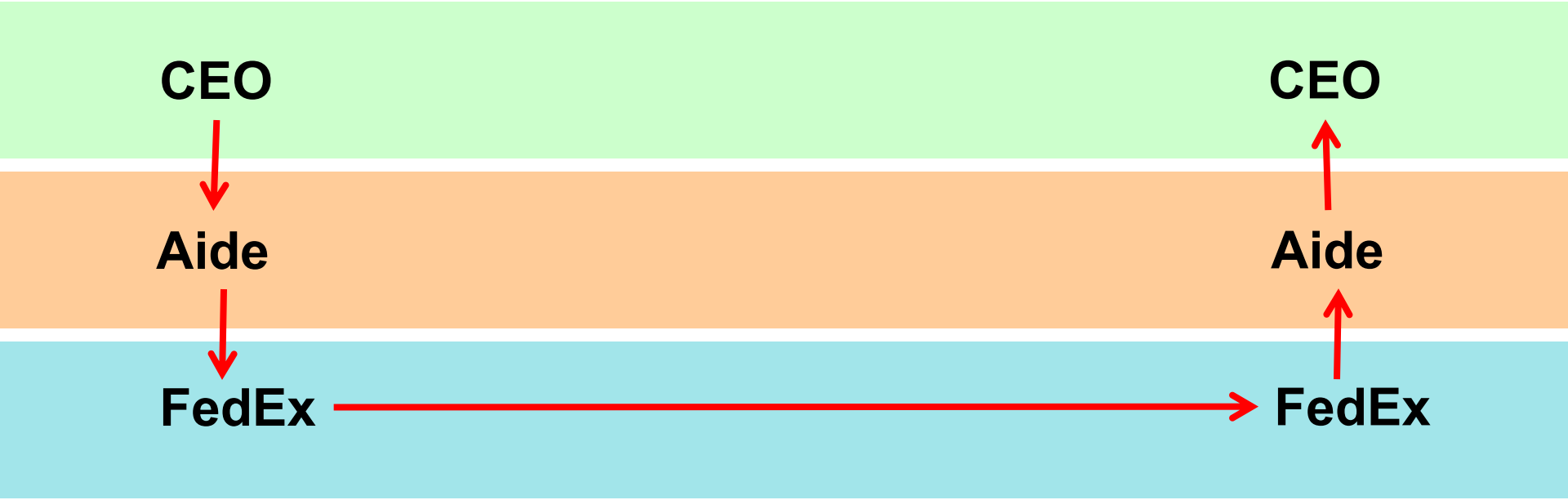


Complicated protocol diagram



Recap: Physical Communication

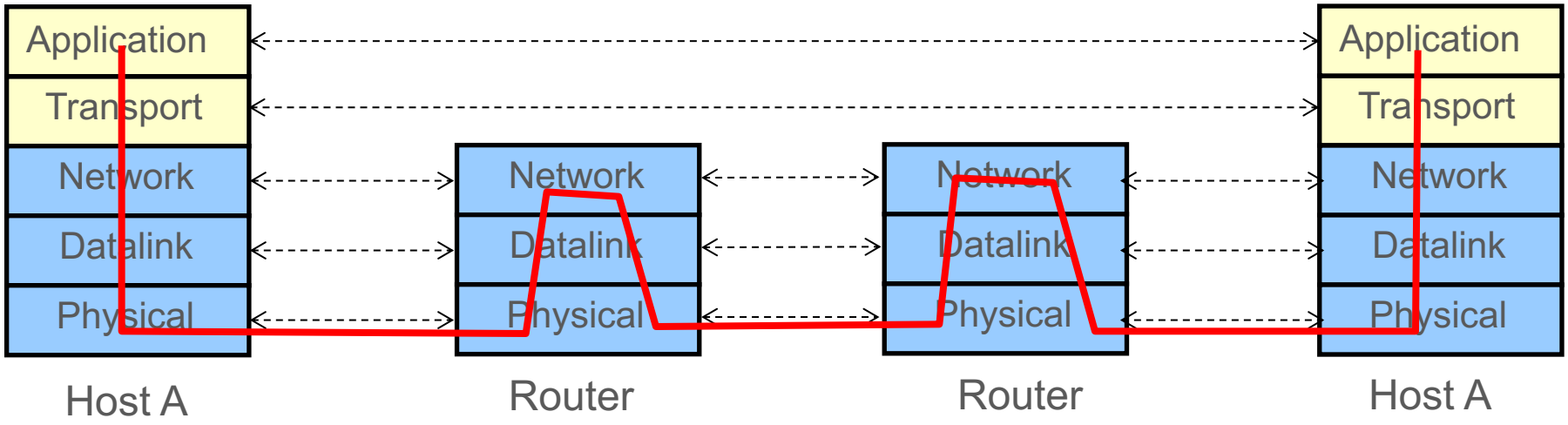
- Communication goes down to physical network
- Then up to relevant layer



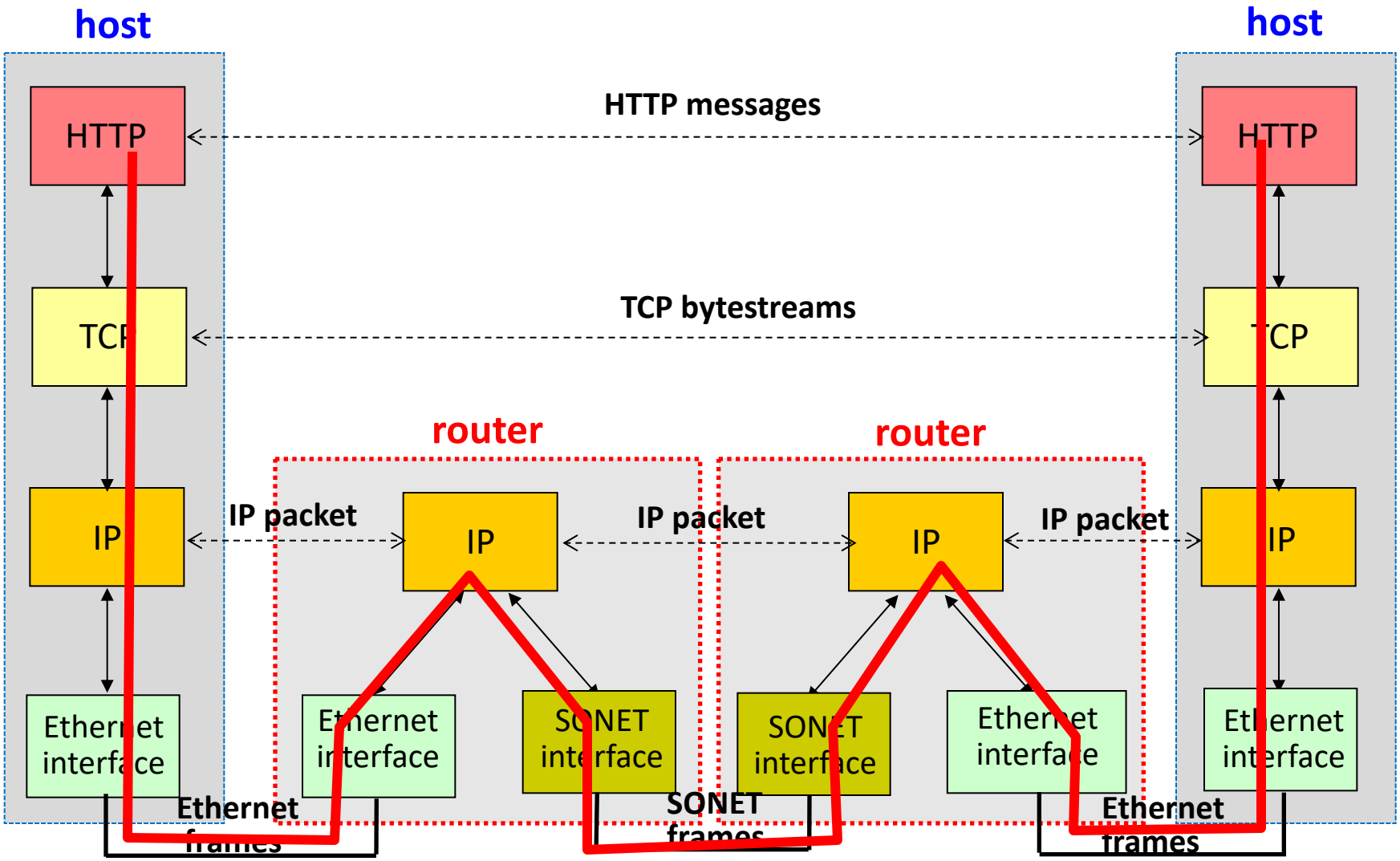
Recall: the path of the letter

Recap: Physical Communication

- Communication goes down to physical network
- Then up to relevant layer

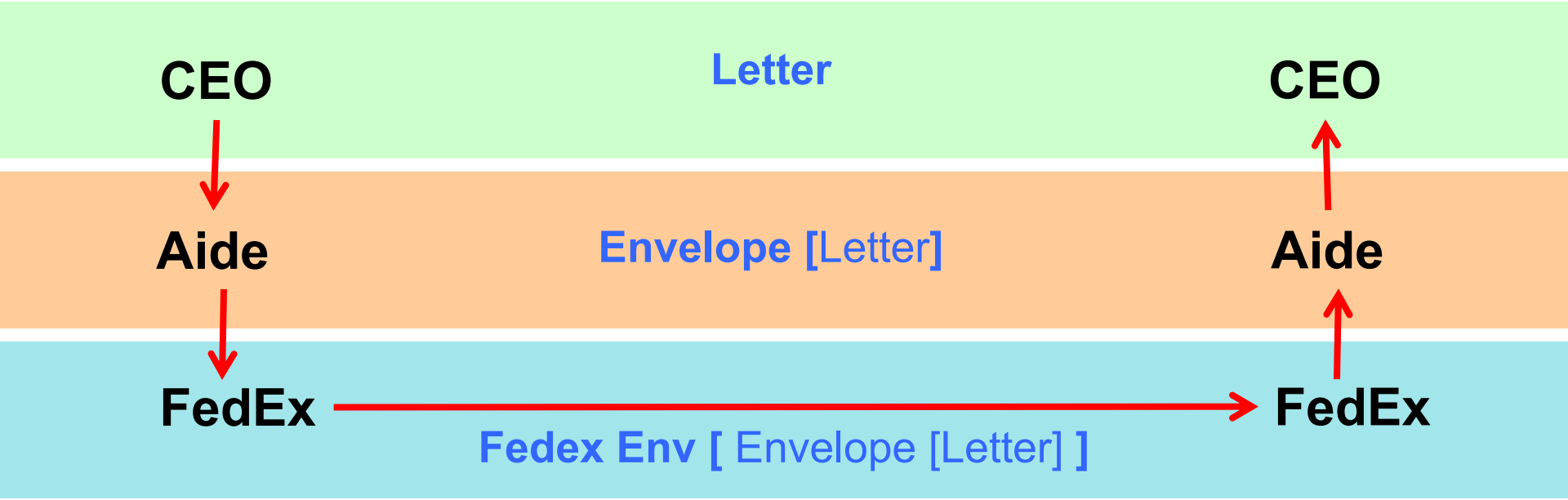


Complicated protocol diagram

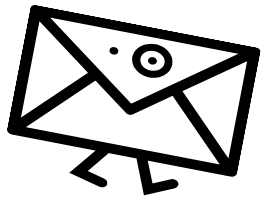


Recap: Physical Communication

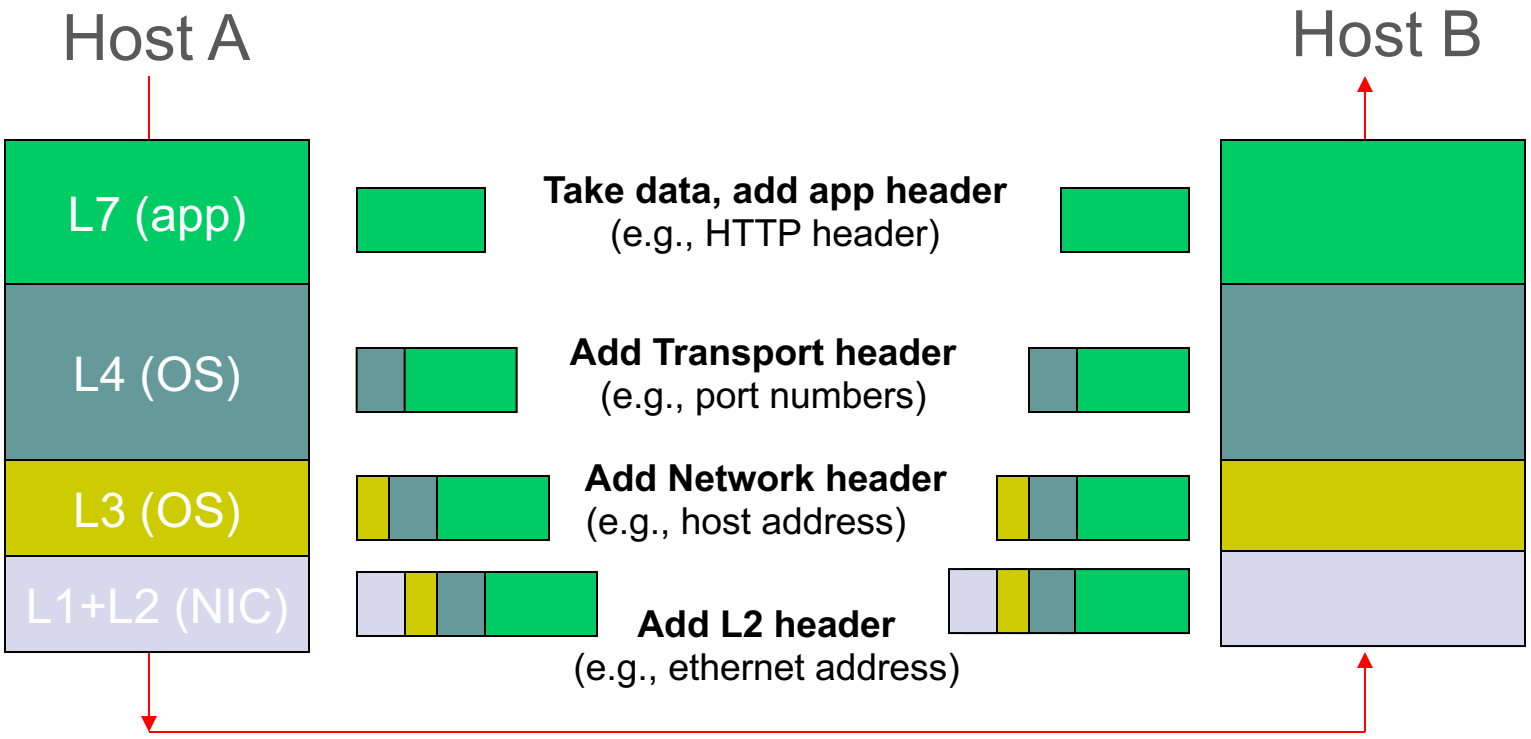
- Communication goes down to physical network
- Then up to relevant layer
- Lowest layer has the most “packaging”



Layer Encapsulation



Packets contain multiple headers!

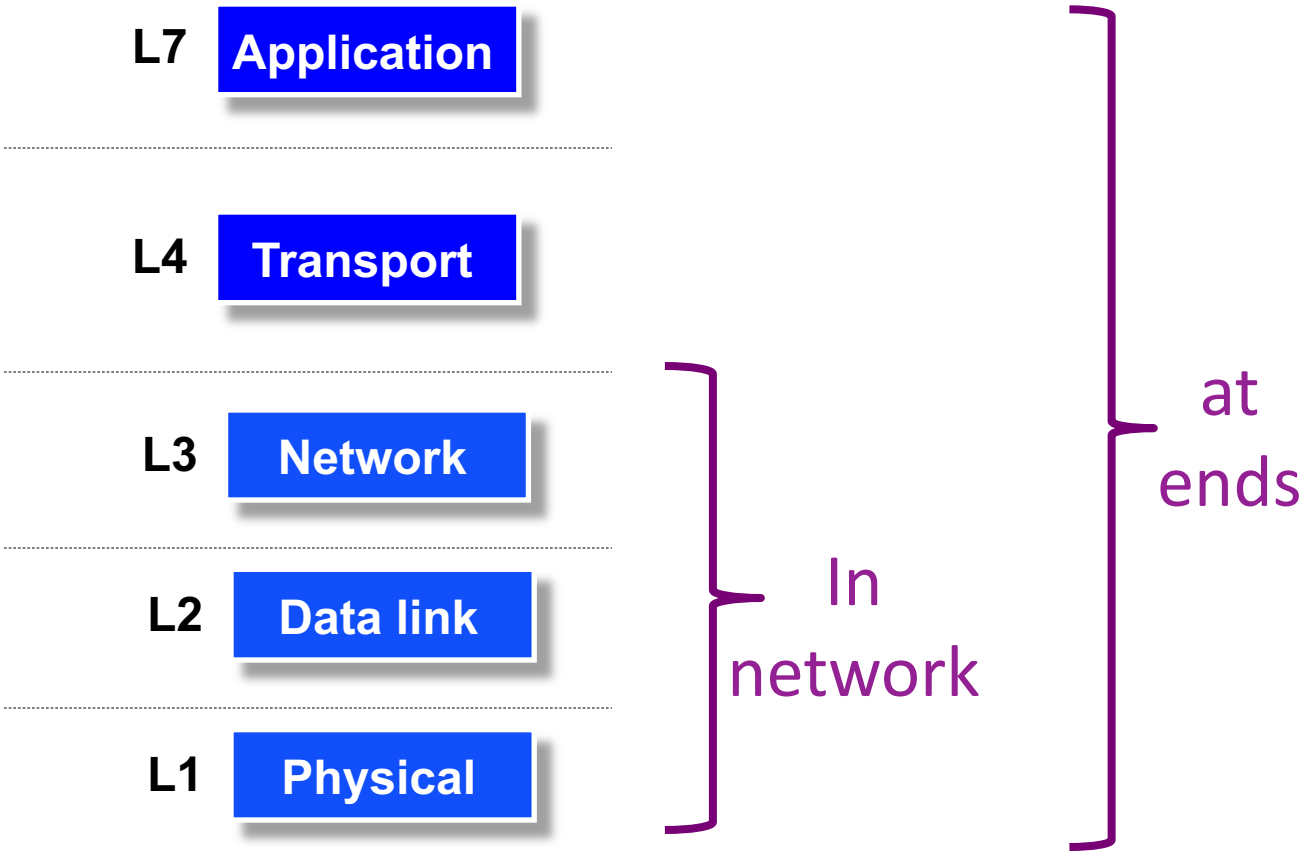


On wire: packet has data + headers from all layers!

Review

- Modularity → layered organization of components
- Peers in a layer “talk” via standardized protocols
- End hosts implement all layers (L1-L7)
- Network implements a subset of layers (L1-L3)
- One universal L3 protocol (IP)

Review



Next: why is *this* assignment of tasks good?