Plan of attack

- What is a network made of?
- How is it shared?
- How do we evaluate a network?
- How is communication organized?

Three steps

- Decompose the problem into tasks
- Organize these tasks
- Assign tasks to entities (who does what)

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

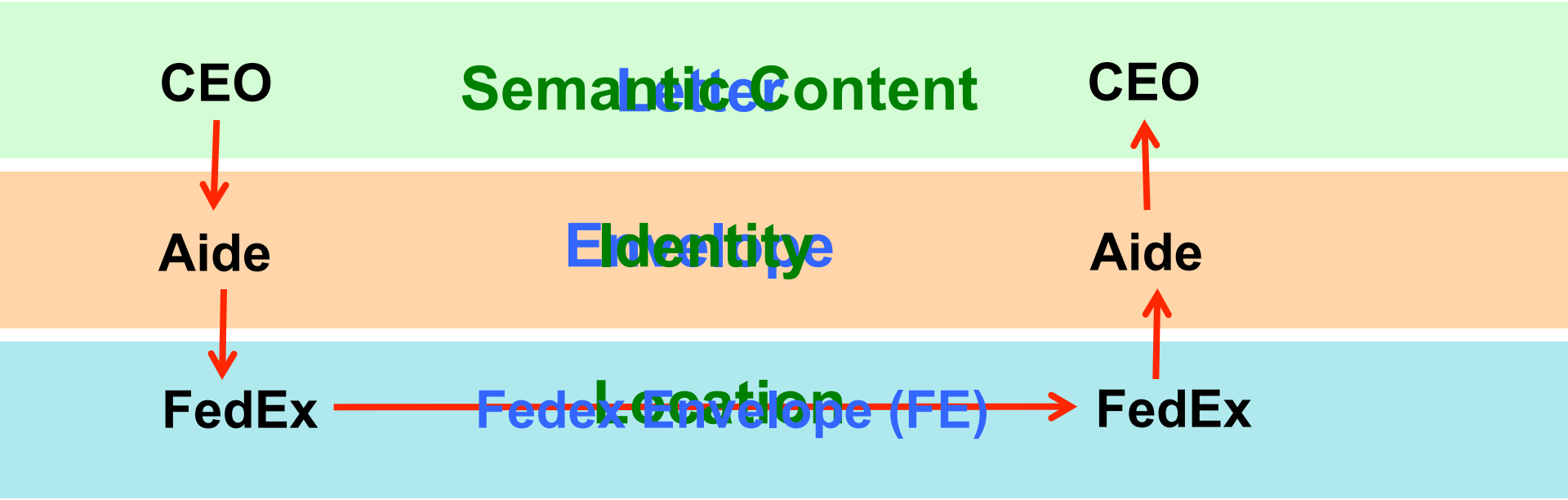
Dear John,

Your days are numbered.

--Pat

The Path of the Letter

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



In the Internet: decomposition

Applications

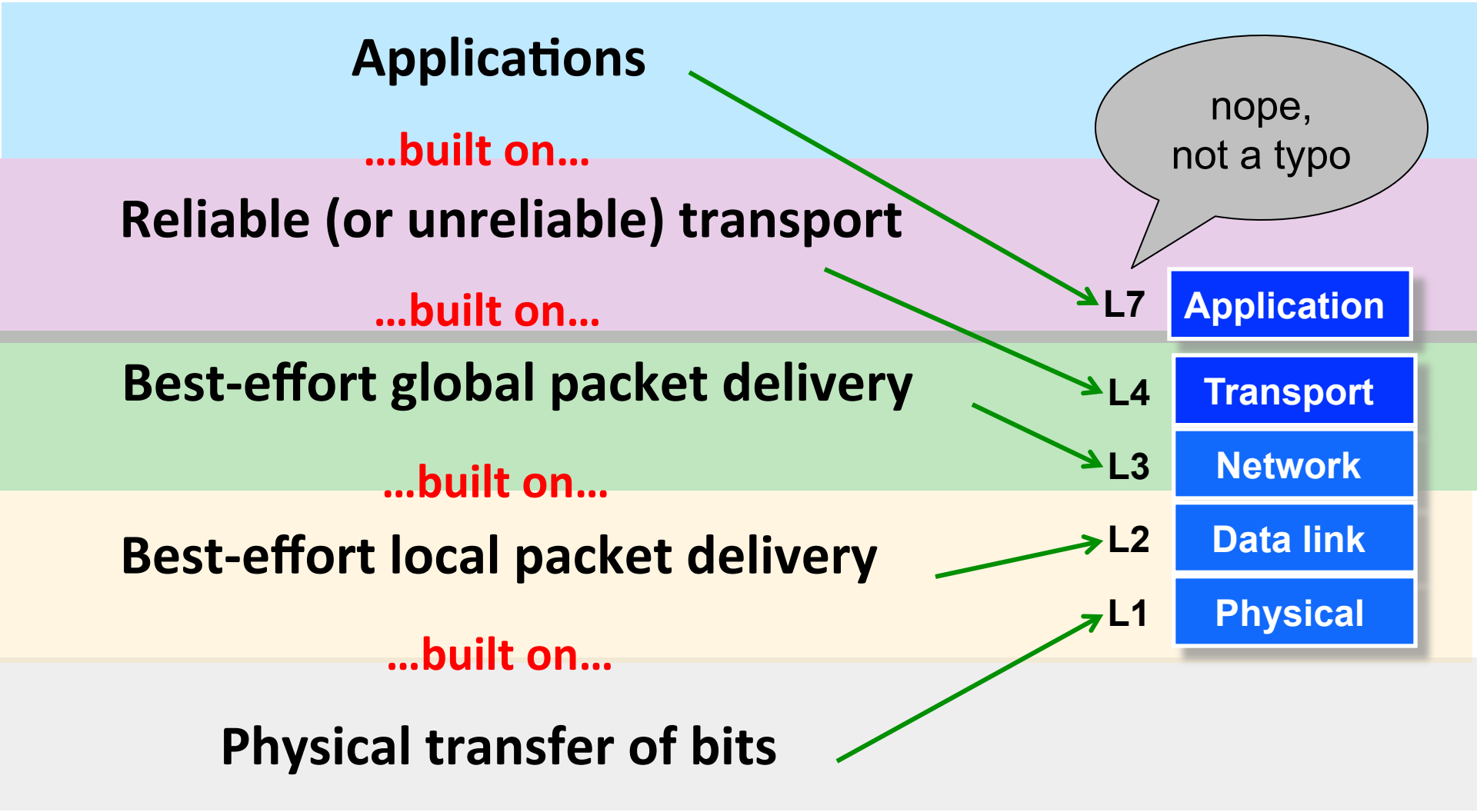
Reliable (or unreliable) transport

Best-effort *global* packet delivery

Best-effort local *packet* delivery

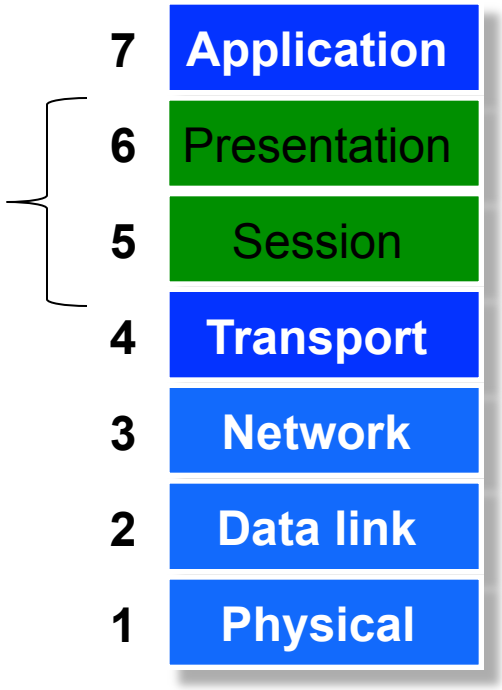
Physical transfer of bits

In the Internet: organization



In the context of the Internet

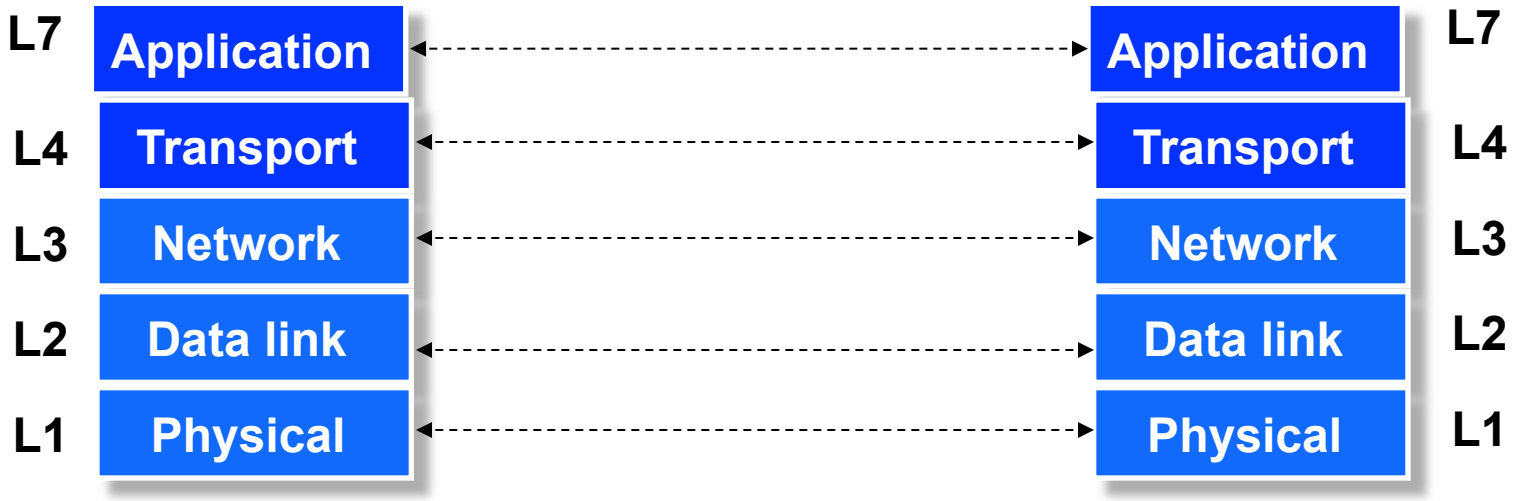
The Open Systems Interconnect (OSI) model developed by the ISO included two additional layers that are often implemented as part of the application



Layers

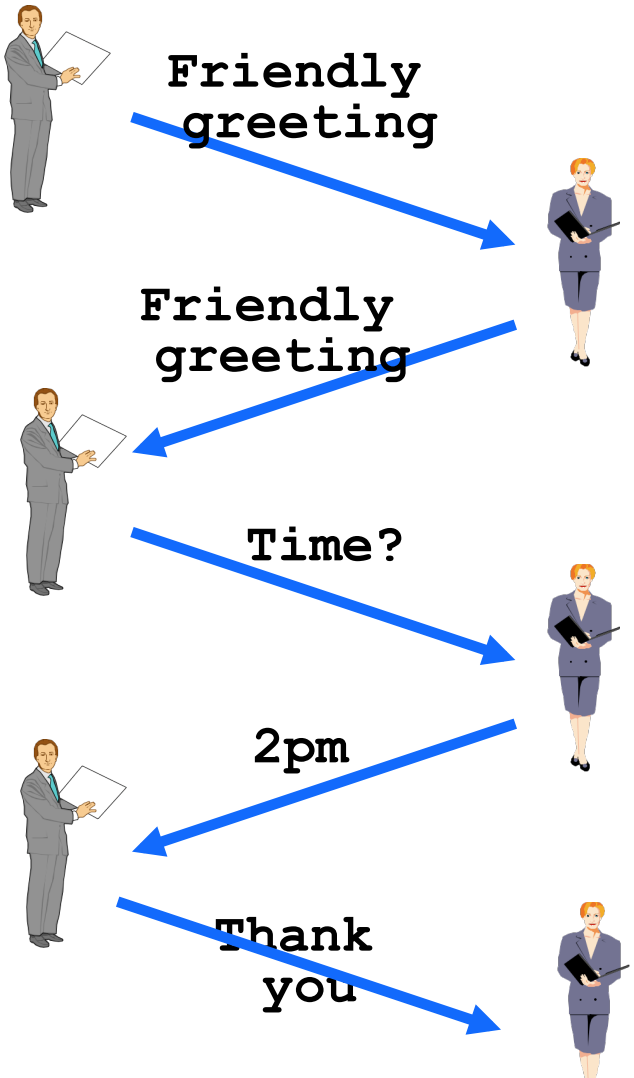
- Layer = a part of a system with well-defined interfaces to other parts
- One layer interacts only with layer above and layer below
- Two layers interact only through the interface between them

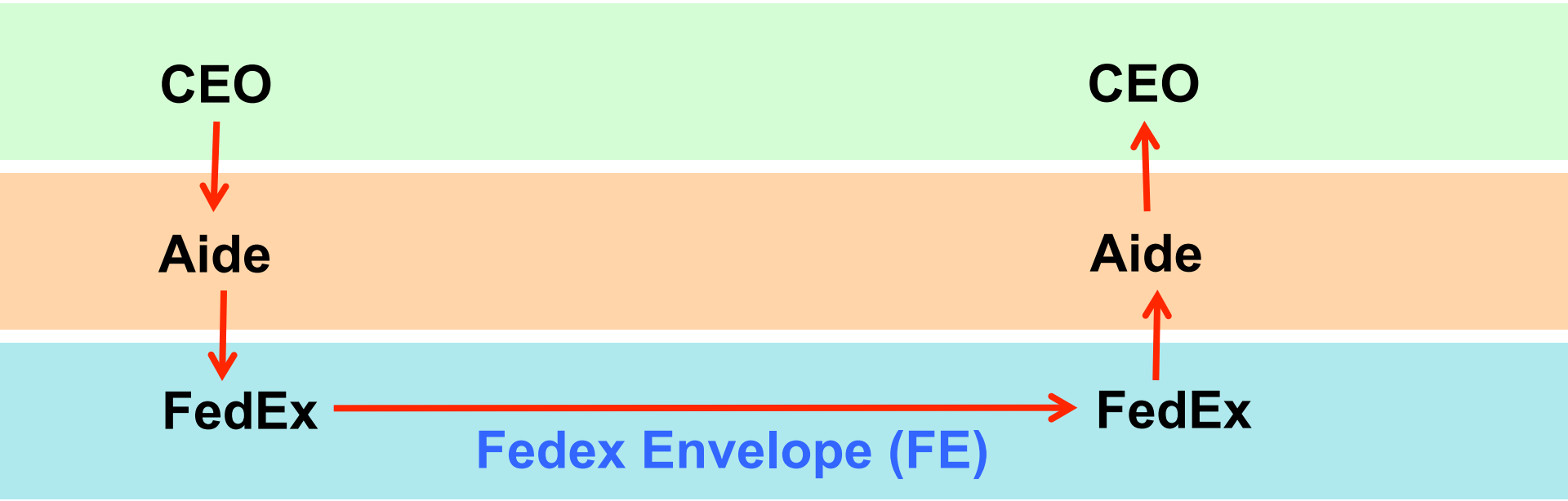
Protocols and Layers



Communication between peer layers on different systems is defined by protocols

What is a Protocol?

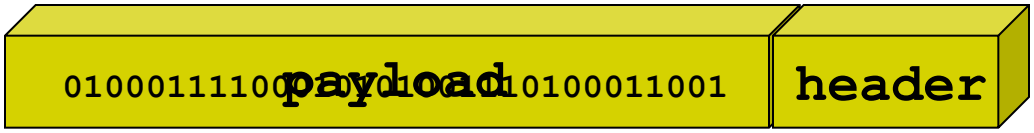



CEO
Aide
FedEx
CEO
Aide
FedEx
Fedex Envelope (FE)

What is a Protocol?

- An agreement between parties on how to communicate

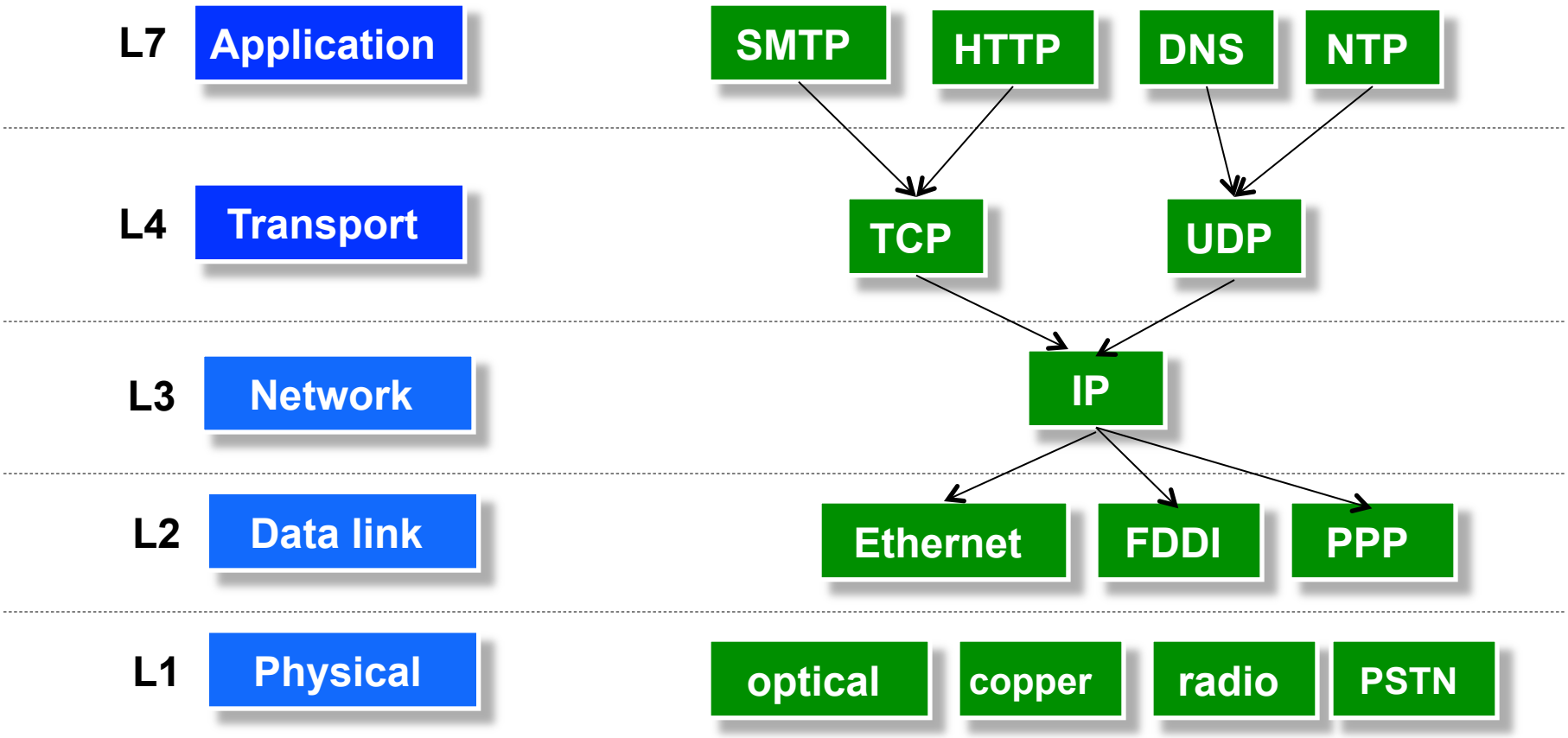
- Defines the syntax of communication
  - header → instructions for how to process the payload
  - Each protocol defines the format of its packet **headers**
    - *e.g. "the first 32 bits carry the destination address"*



What is a Protocol?

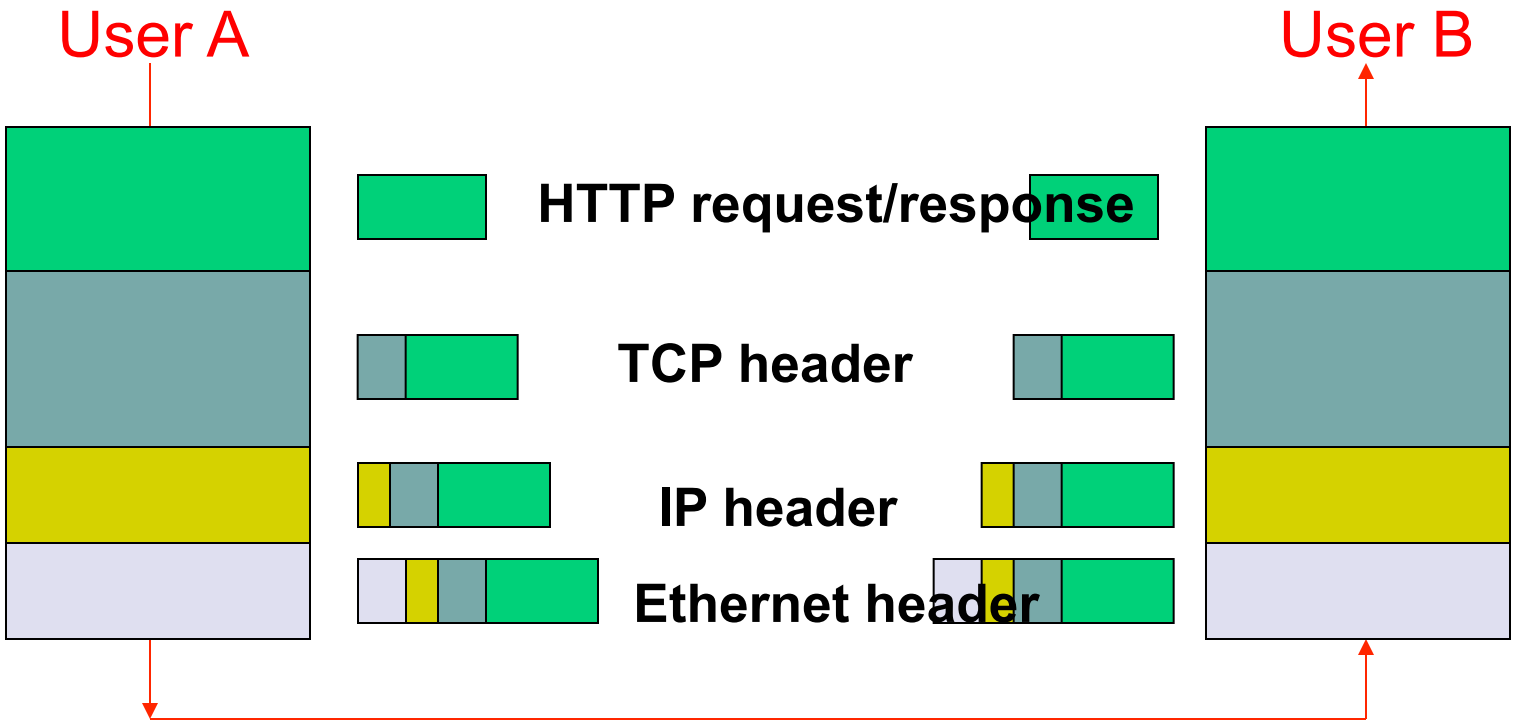
- An agreement between parties on how to communicate
- Defines the syntax of communication
- And semantics
  - "first a hullo, then a request…"
  - we'll study many protocols later in the semester
- Protocols exist at many levels, hardware and software
  - defined by a variety of standards bodies (IETF,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

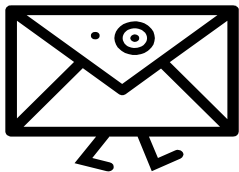
Protocols at different layers



There is just one network-layer protocol!

Layer Encapsulation: Protocol Headers

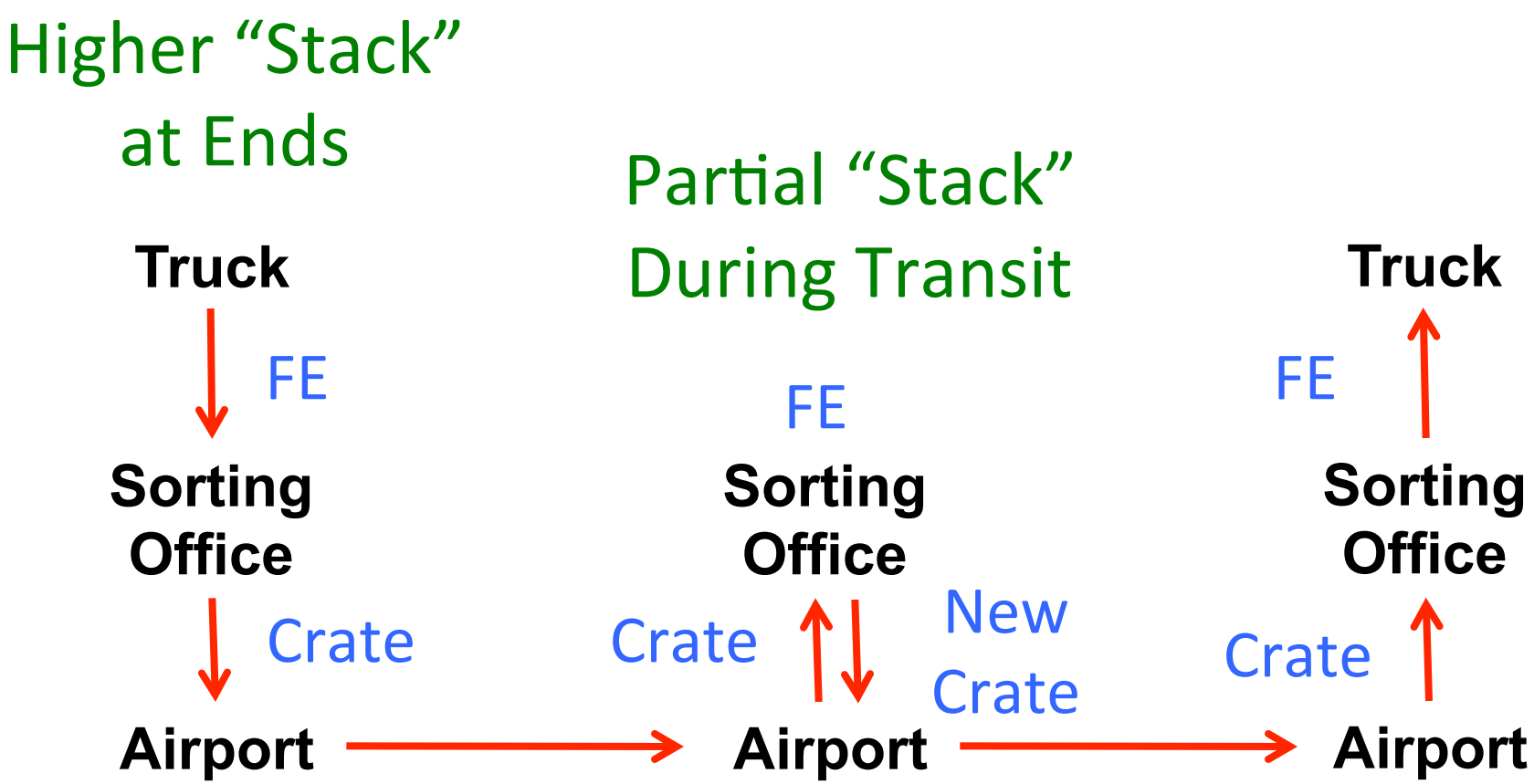




Three steps

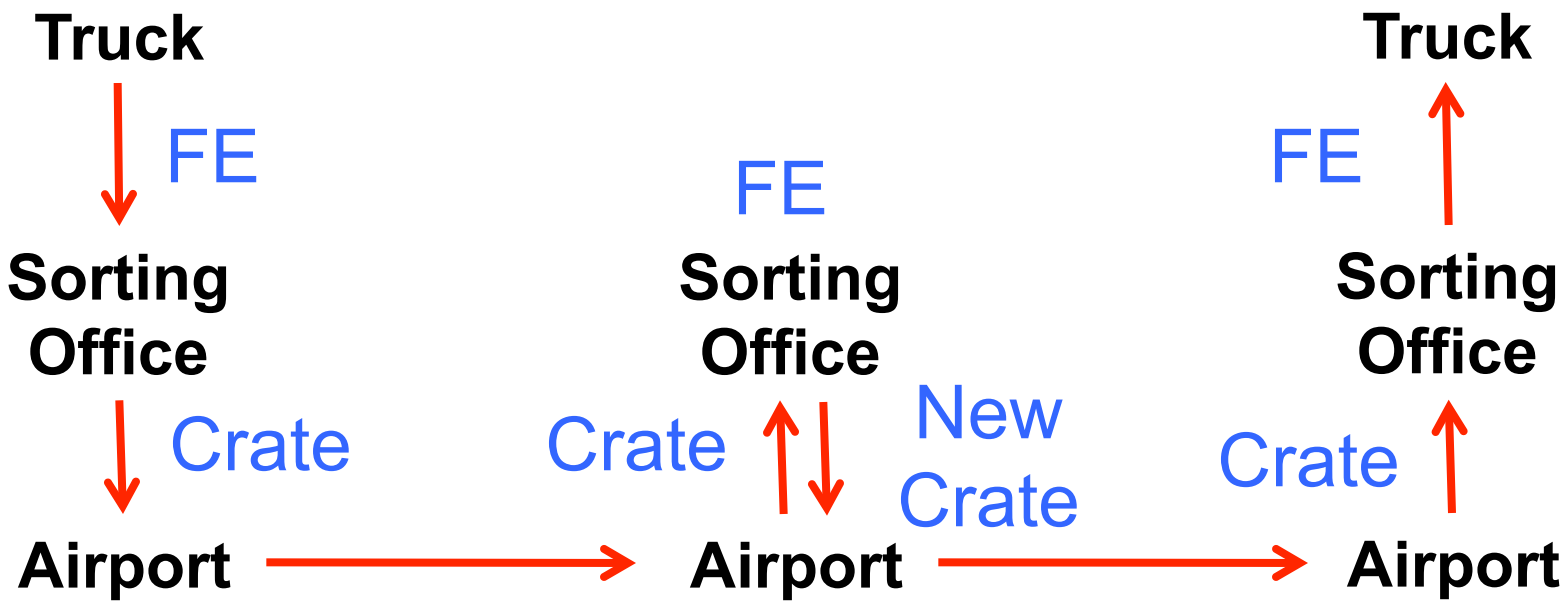
- Decomposition
- Organization
- Assignment

The Path Through FedEx

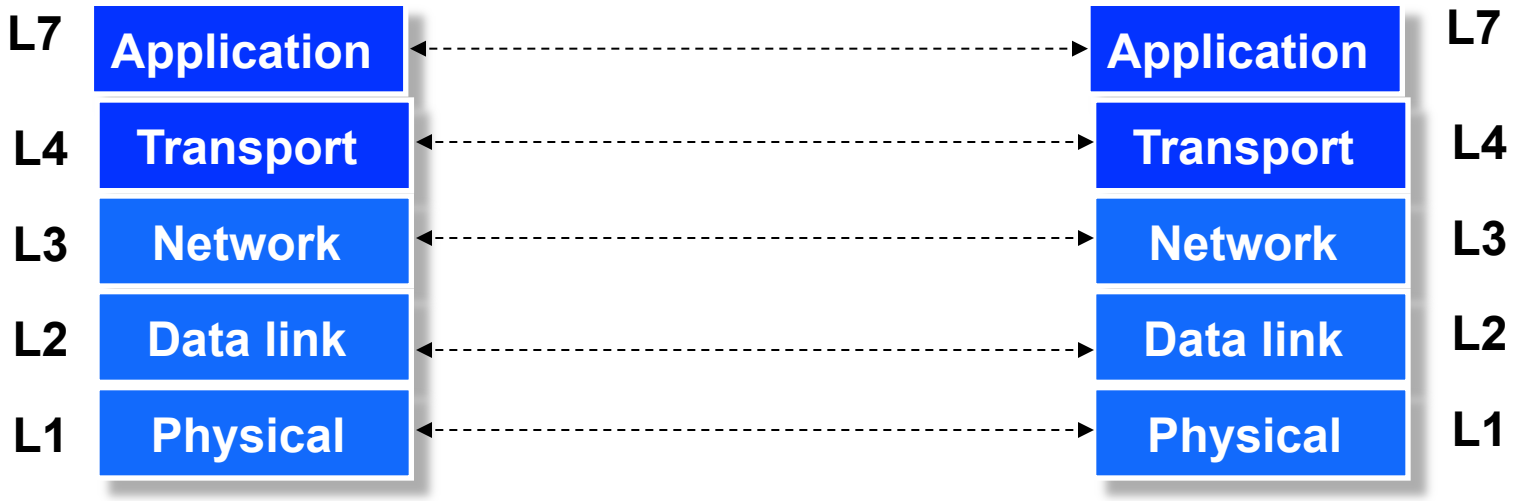


Deepest Packaging (Envelope+FE+Crate)
at the Lowest Level of Transport

The Path Through FedEx



Deepest Packaging (Envelope+FE+Crate) at the Lowest Level of Transport



What gets implemented where?

What gets implemented at the end systems?

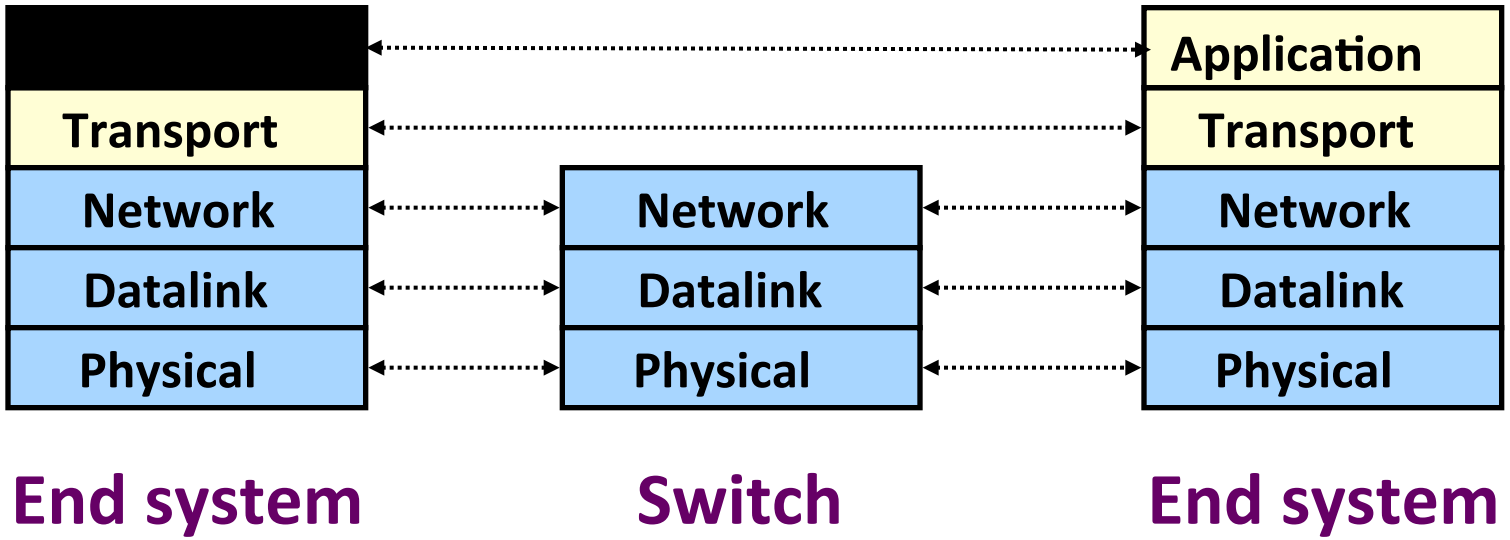
- Bits arrive on wire, must make it up to application
- Therefore, all layers must exist at host!

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

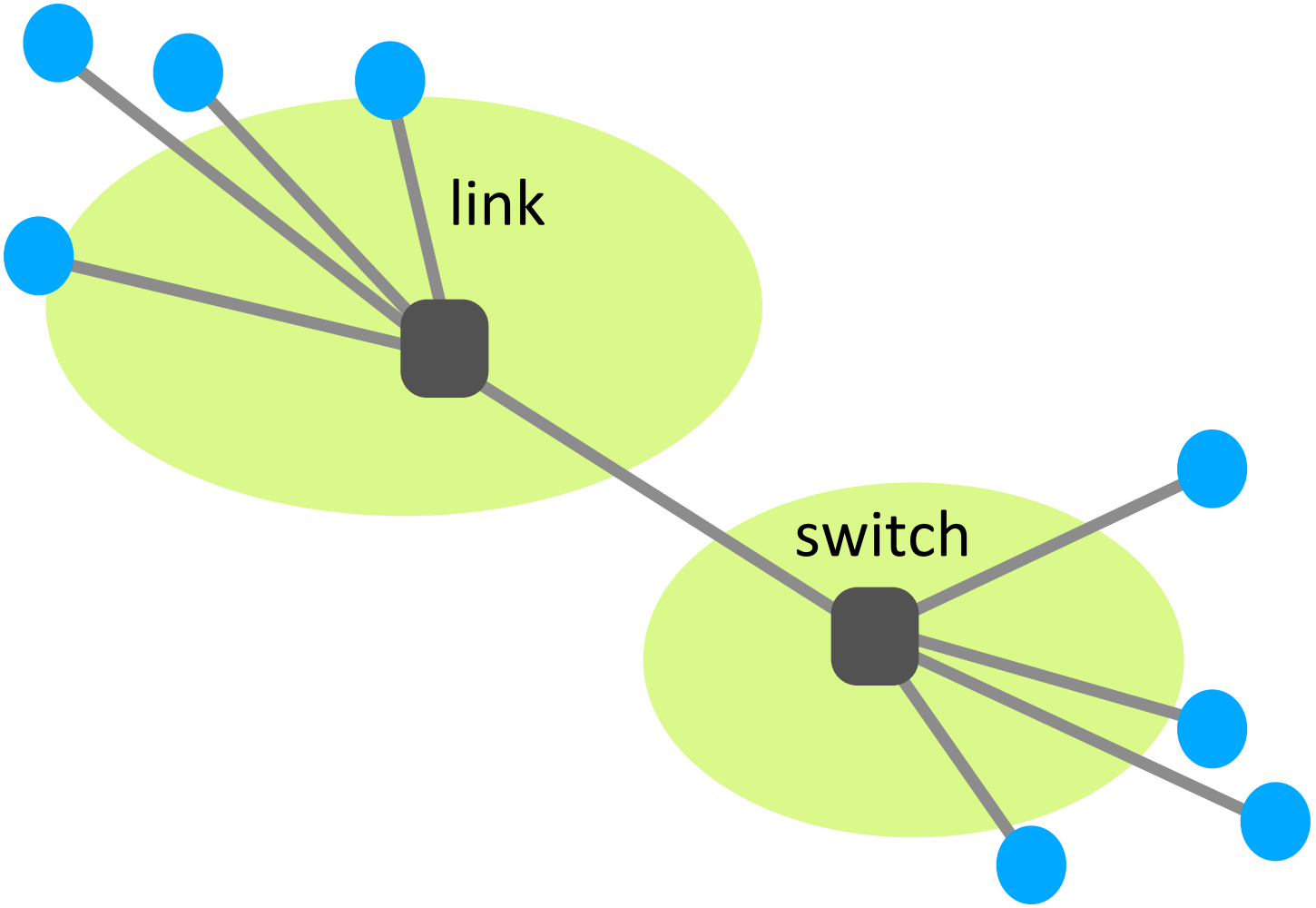
- Lower three layers implemented everywhere
- Top two layers implemented only at hosts



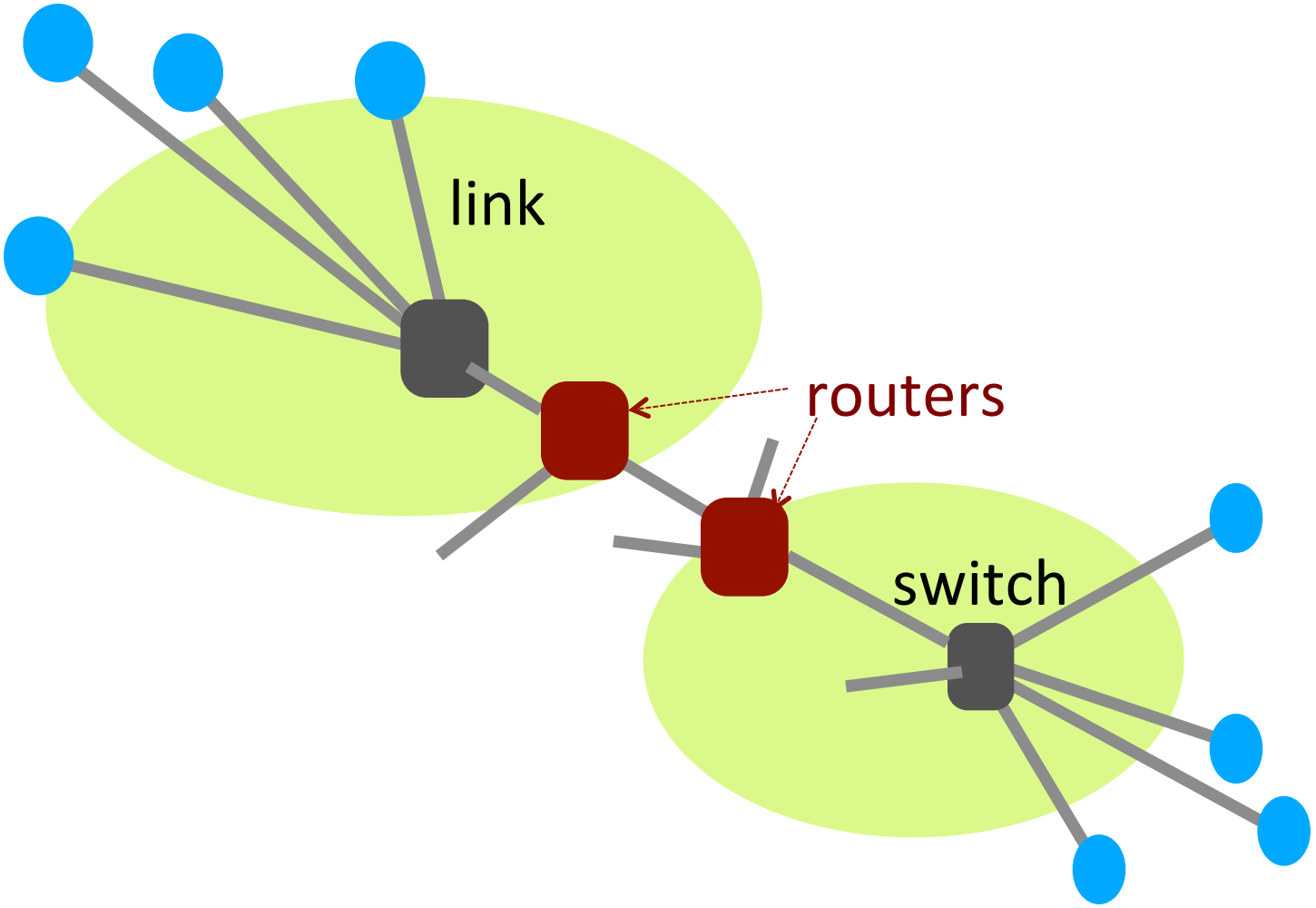
A closer look: end-system

- Application
  - Web server, browser, mail, game
- Transport and network layer
  - typically part of the operating system
- Datalink and physical layer
  - hardware/firmware/drivers

A closer look: network



A closer look: network



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)

- Hence:
  - switches: implement physical and datalink layers (L1, L2)
  - routers: implement physical, datalink, network layers (L1, L2, L3)

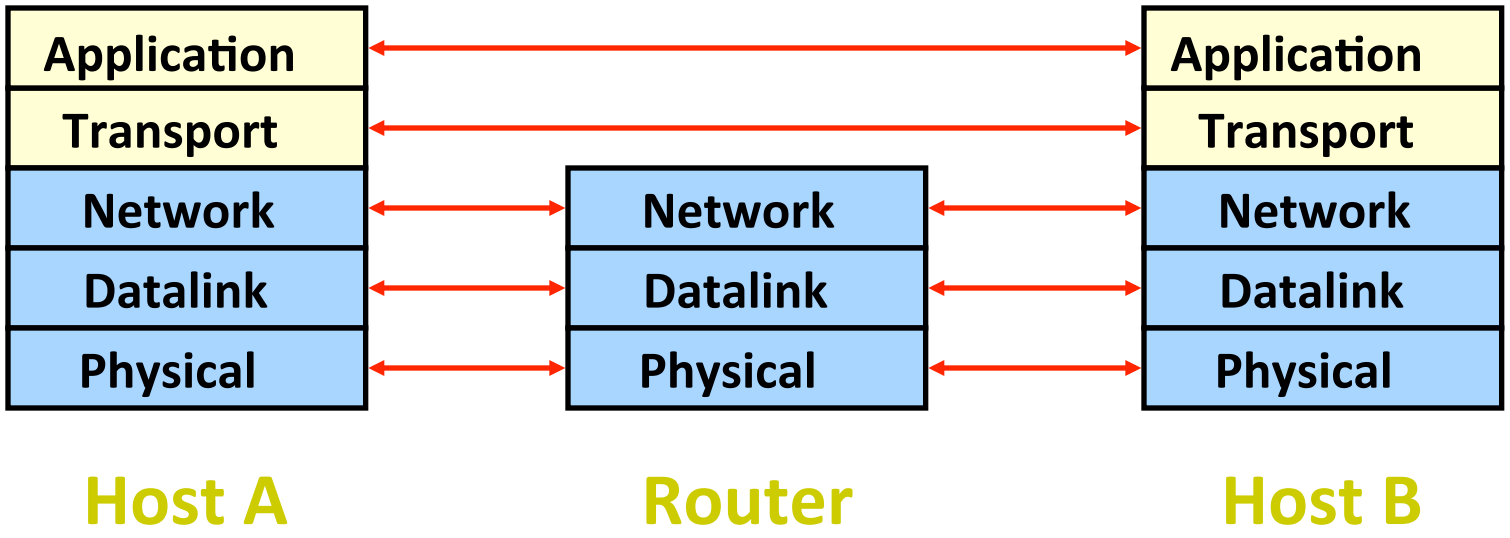
Switches *vs.* Routers

- 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

- Won't focus on the router/switch distinction
  - when I say switch, I almost always mean router
  - almost all boxes support network layer these days

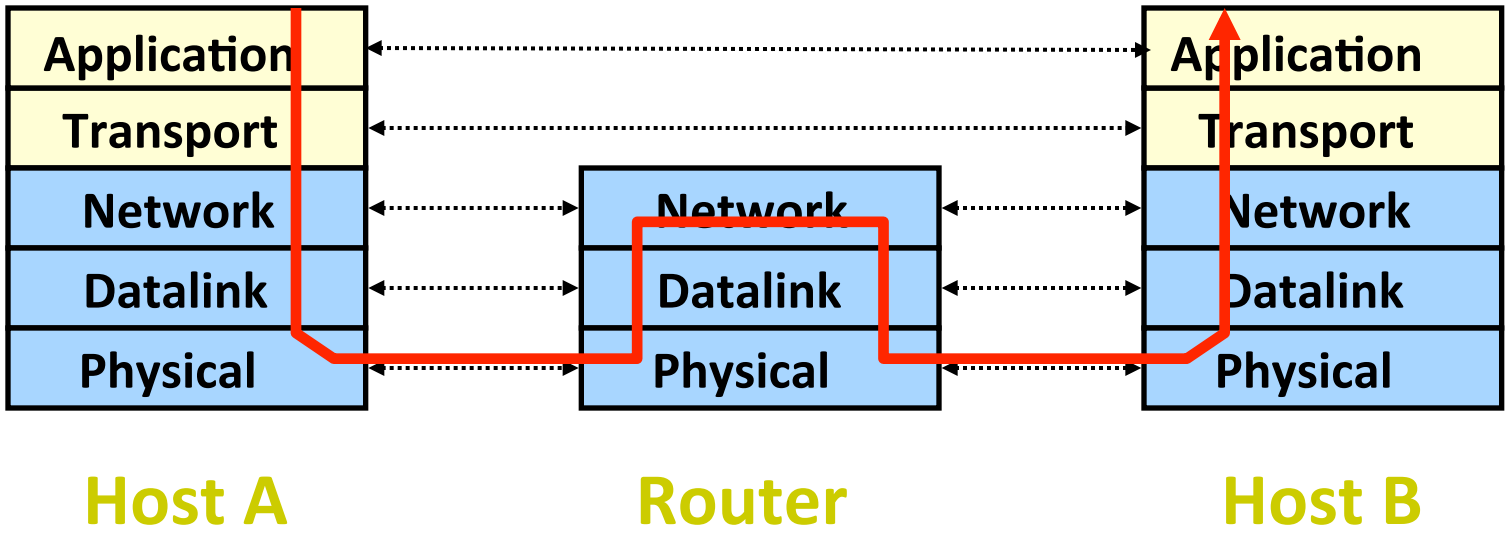
Logical Communication

- Layers interacts with peer's corresponding layer

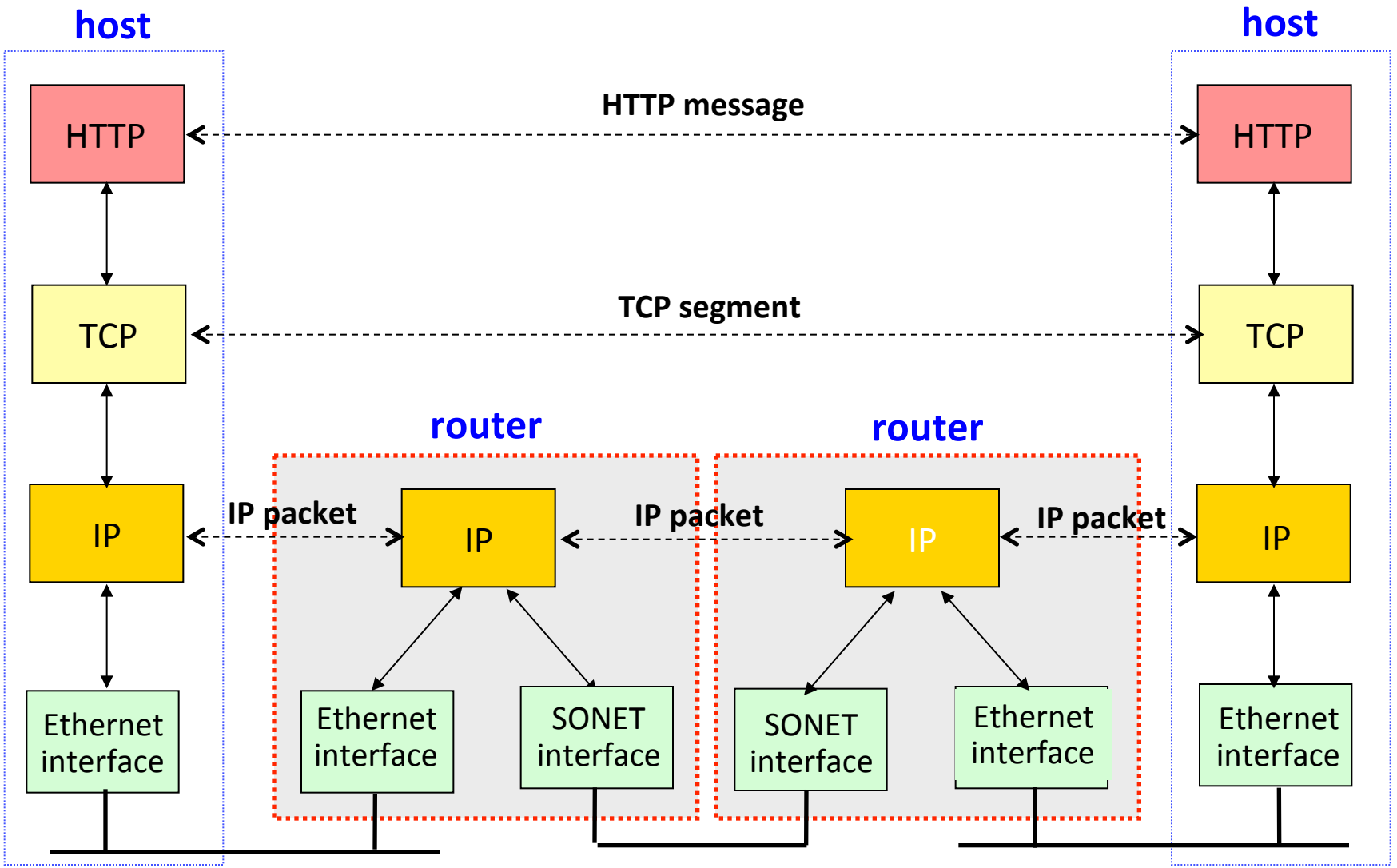


Physical Communication

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



A Protocol-Centric Diagram



Why layers?

- Reduce complexity
- Improve flexibility

Why not?

- sub-optimal performance
- cross-layer information often useful
  - *several "layer violations" in practice*

- *What physical infrastructure is already available?*
- *Reserve or on-demand?*
- *Where's my delay coming from?*
- *To layer or not? What layers? Where?*

Implications of Hourglass

Single network-layer protocol (IP)

- Allows arbitrary networks to interoperate
  - Any network that supports IP can exchange packets
- Decouples applications from low-level networking technologies
  - Applications function on *all* networks
- Supports simultaneous innovations above and below IP
- But changing IP itself is hard (e.g., IPv4 → IPv6)

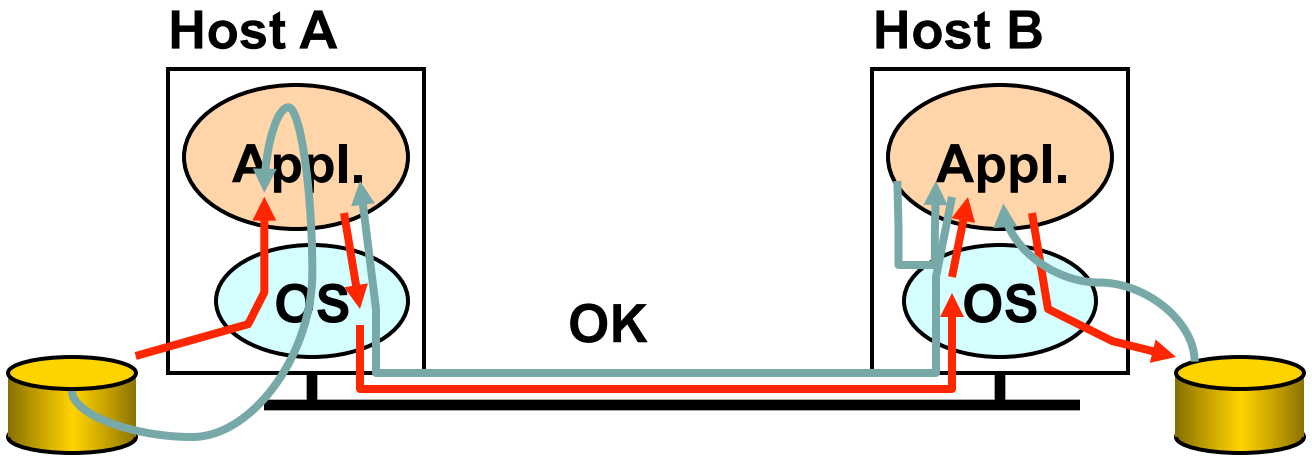
Placing Network Functionality

- Hugely influential paper: "End-to-End Arguments in System Design" by Saltzer, Reed, and Clark ('84)
  - articulated the "End-to-End Principle" (E2E)
- Endless debate over what it means
- Everyone cites it as supporting their position

Basic Observation

- Some application requirements can only be correctly implemented **end-to-end**
  - reliability, security, *etc.*
- Implementing these in the network is hard
  - every step along the way must be fail proof
- Hosts
  - **Can** satisfy the requirement without network's help
  - **Will/must** do so, since they can't rely on the network

Example: Reliable File Transfer



Solution 2: end-to-end **check** and retry

Discussion

- Solution 1 is incomplete
  - What happens if any network element misbehaves?
  - Receiver has to do the check anyway!
- Solution 2 is complete
  - Full functionality can be entirely implemented at application layer with no need for reliability from lower layers
- Is there any need to implement reliability at lower layers?

Summary of End-to-End Principle

- Does increase network complexity
- Probably increases delay and overhead on all applications even if they don't need the functionality (e.g. VoIP)

- However, implementing in the network can improve performance in some cases
  - e.g., consider a very lossy link

Recap

- E2E argument encourages us to keep IP simple