

EECS 489

Computer Networks

Winter 2024

Mosharaf Chowdhury

Material with thanks to Aditya Akella, Sugih Jamin, Philip Levis, Sylvia Ratnasamy, Peter Steenkiste, and many other colleagues.

Recap

- Delay refers to the time to send data from src to dst
 - It has **four components**: transmission, propagation, queueing, and processing
- Bandwidth is the characteristics of a link
- Throughput is the rate at which a src and dst can communicate
 - Bounded by the bottleneck link on the path

Agenda

- How is communication organized?

What we want

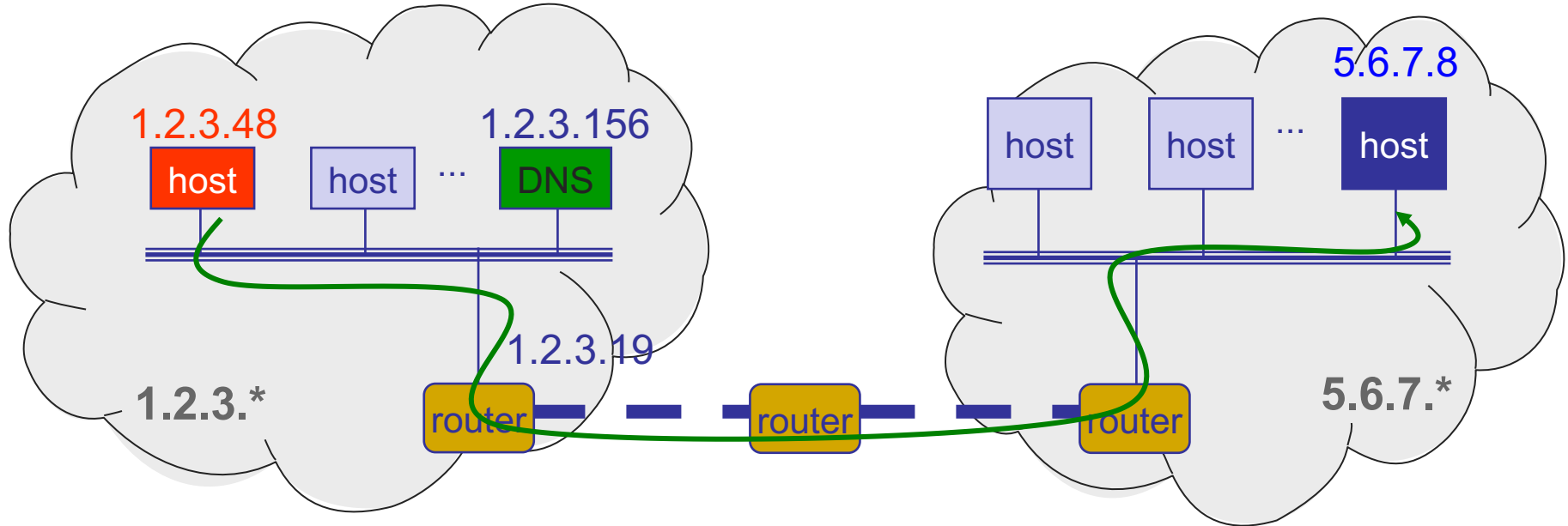
`http://123.xyz`



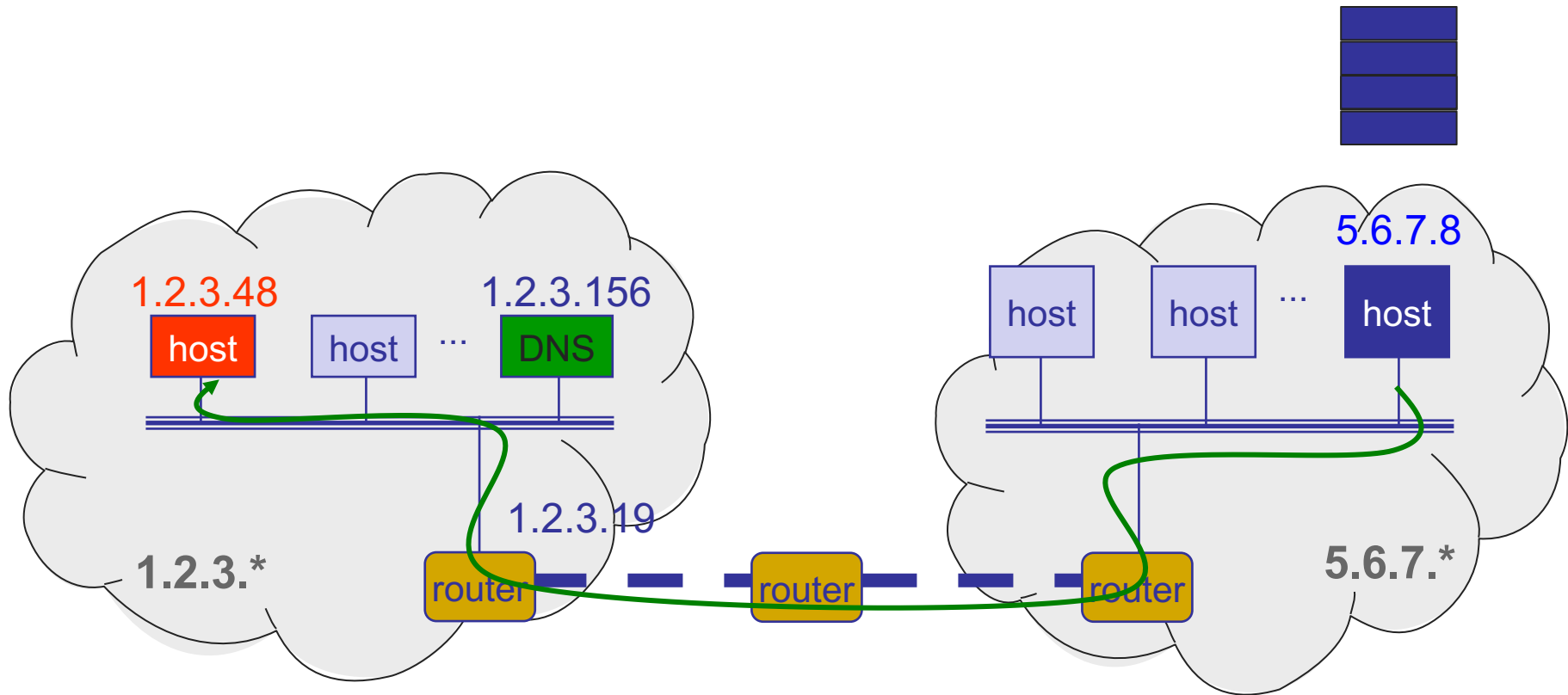
123.xyz server



(Some of) What happens...



(More of) What happens



What we get



123.xyz server



Inspiration...

- CEO A writes letter to CEO B

Dear John,

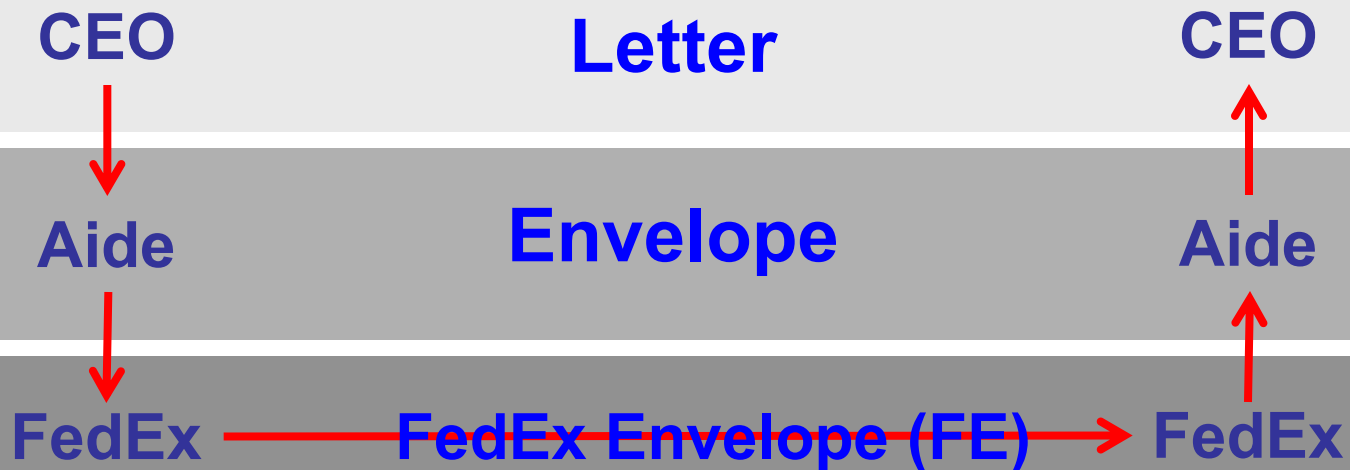
Your days are numbered.

--Pat

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

The path of the letter



The path of the letter

- “Peers” in same layer understand each other
- No one else needs to
- Lowest level has most packaging

CEO

Semantic Content

CEO

Aide

Identity

Aide

FedEx

Location

FedEx

Three steps

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

Back to the Internet: Decomposition

Applications

in built on

Reliable or unreliable transport

in built on

Best-effort **global** packet delivery

in built on

Best-effort **local** packet delivery

in built on

Physical transfer of bits

Communication organization

Applications

in built on

Reliable or unreliable transport

in built on

Best-effort **global** packet delivery

in built on

Best-effort **local** packet delivery

in built on

Physical transfer of bits

L7

Application

L4

Transport

L3

Network

L2

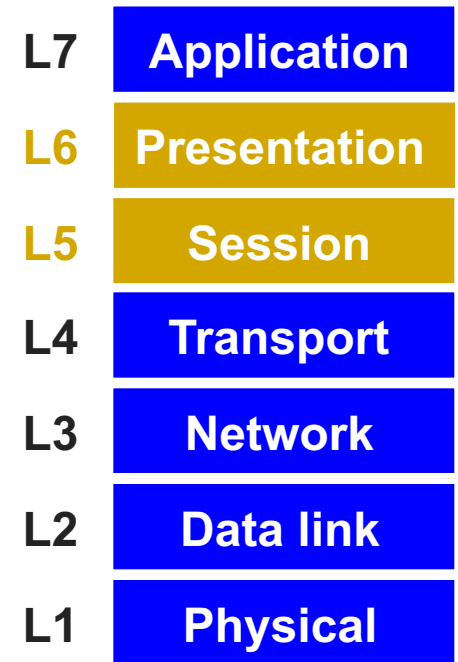
Data link

L1

Physical

OSI layers

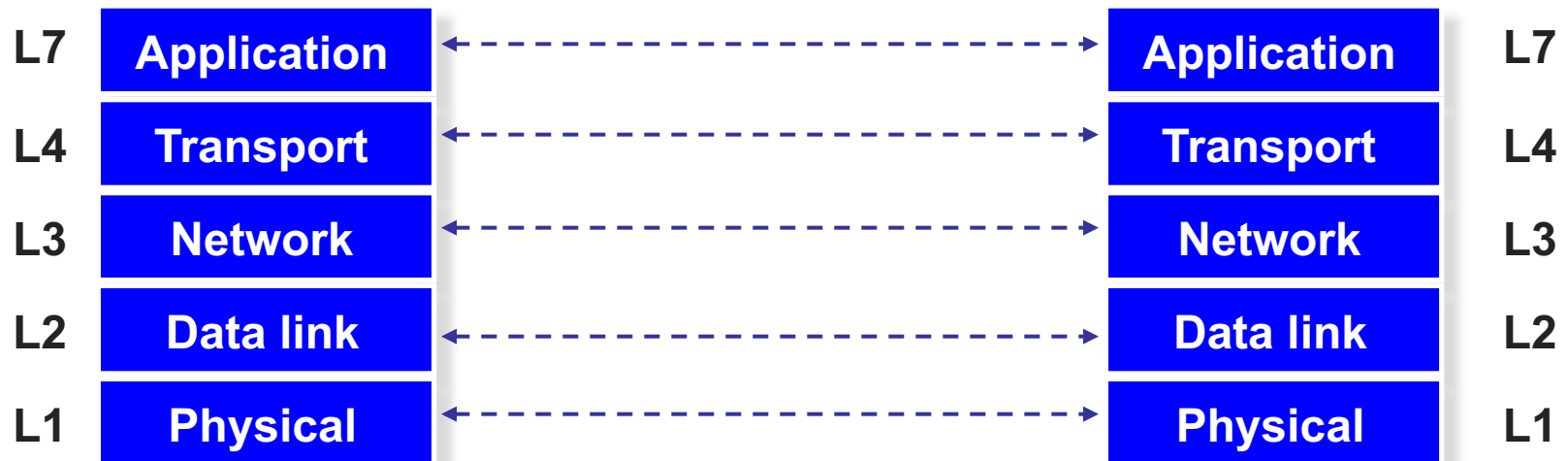
- OSI stands for Open Systems Interconnection model
 - Developed by the ISO
- Session and presentation layers are often implemented as part of the application layer



Layers

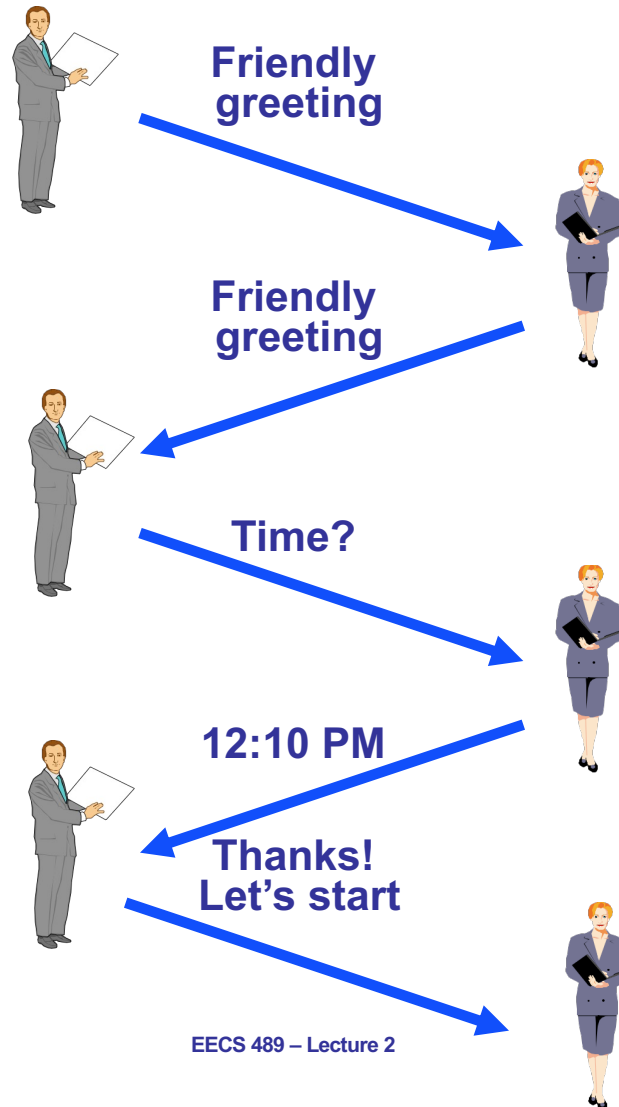
- Layer: a part of a system with well-defined interfaces to other parts
- One layer interacts only with layer above and layer below
- Two layers interact only through the interface between them

Layers and protocols



- Communication between peer layers on different systems is defined by **protocols**

What is a Protocol?



What is a Protocol?

- An agreement between parties (in the same later) on how to communicate
- Defines the **syntax** of communication
 - **Header** → instructions on how to process **payload**
 - Each protocol defines the format of its 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 hello, then a request...”
 - We will study many protocols later in the semester
- Protocols exist at many levels, hardware, and software
 - Defined by standards bodies like IETF, IEEE, ITU

Protocols at different layers

L7 **Application**

SMTP

HTTP

DNS

NTP

L4 **Transport**

TCP

UDP

L3 **Network**

IP

L2 **Data link**

Ethernet

FDDI

PPP

L1 **Physical**

Optical

Copper

Radio

PSTN

ONE network layer protocol

L7 **Application**

SMTP

HTTP

DNS

NTP

L4 **Transport**

TCP

UDP

L3 **Network**

IP

L2 **Data link**

Ethernet

FDDI

PPP

L1 **Physical**

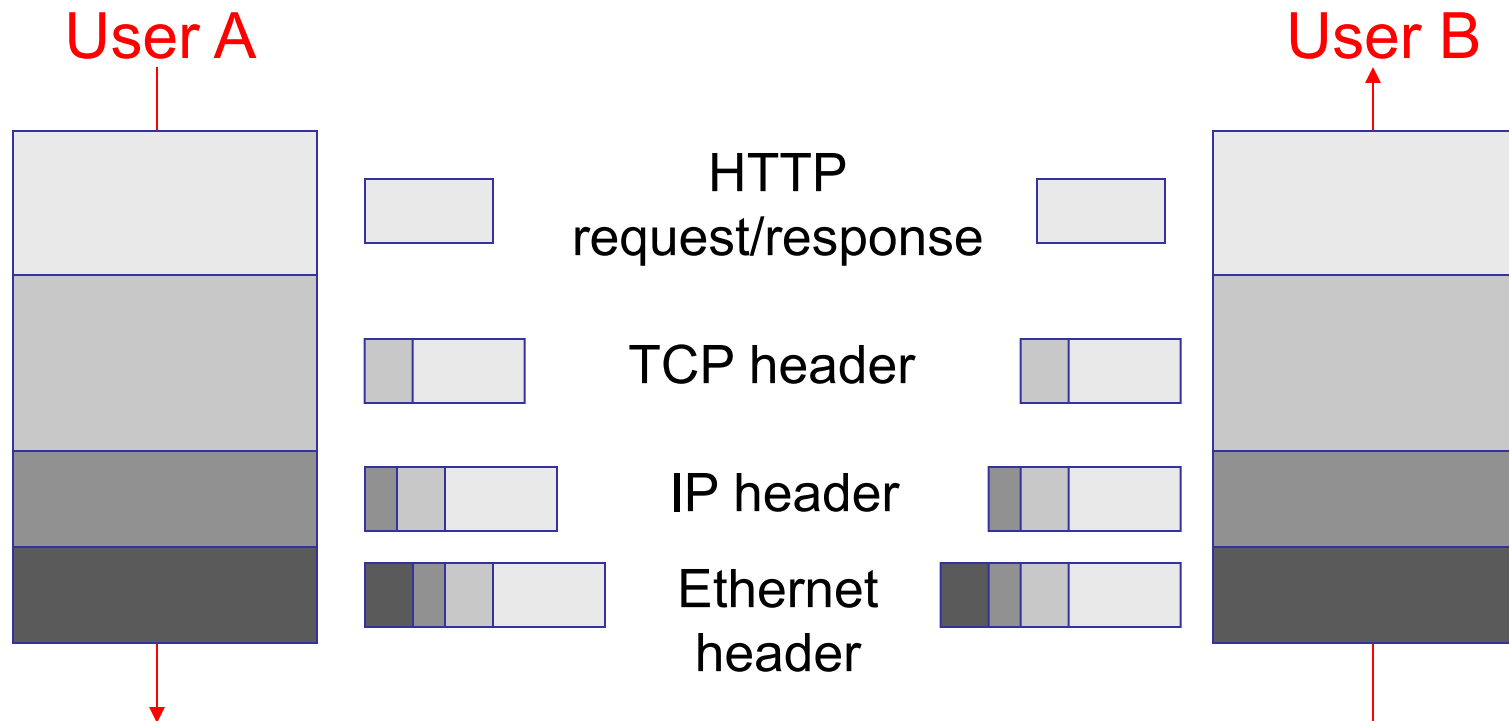
Optical

Copper

Radio

PSTN

Layer encapsulation: Protocol headers



5-MINUTE BREAK!

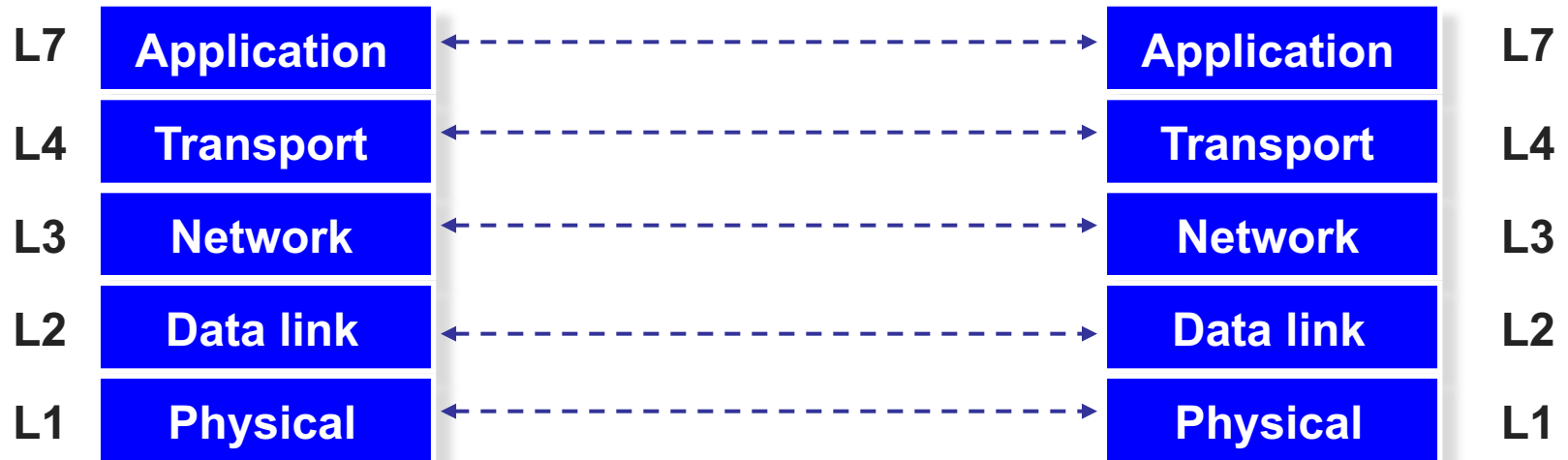
Announcements

- Assignment 1 is out!
 - Due Jan 26, 2024
- Register your github username
 - Link in A1 spec
 - And the spec is located at <https://github.com/mosharaf/eecs489>

Three steps

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

What gets implemented where?



What gets implemented at the end systems?

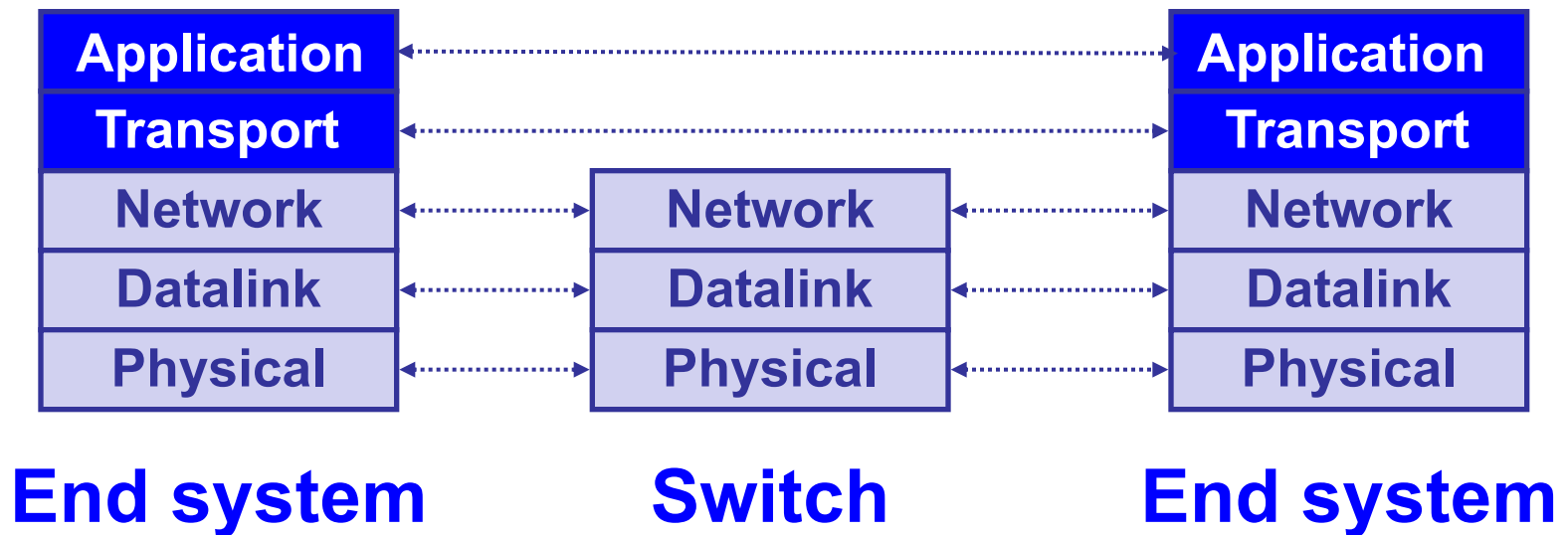
- Bits arrive on wire, must make it up to application
- Therefore, **all layers must exist at host!**

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

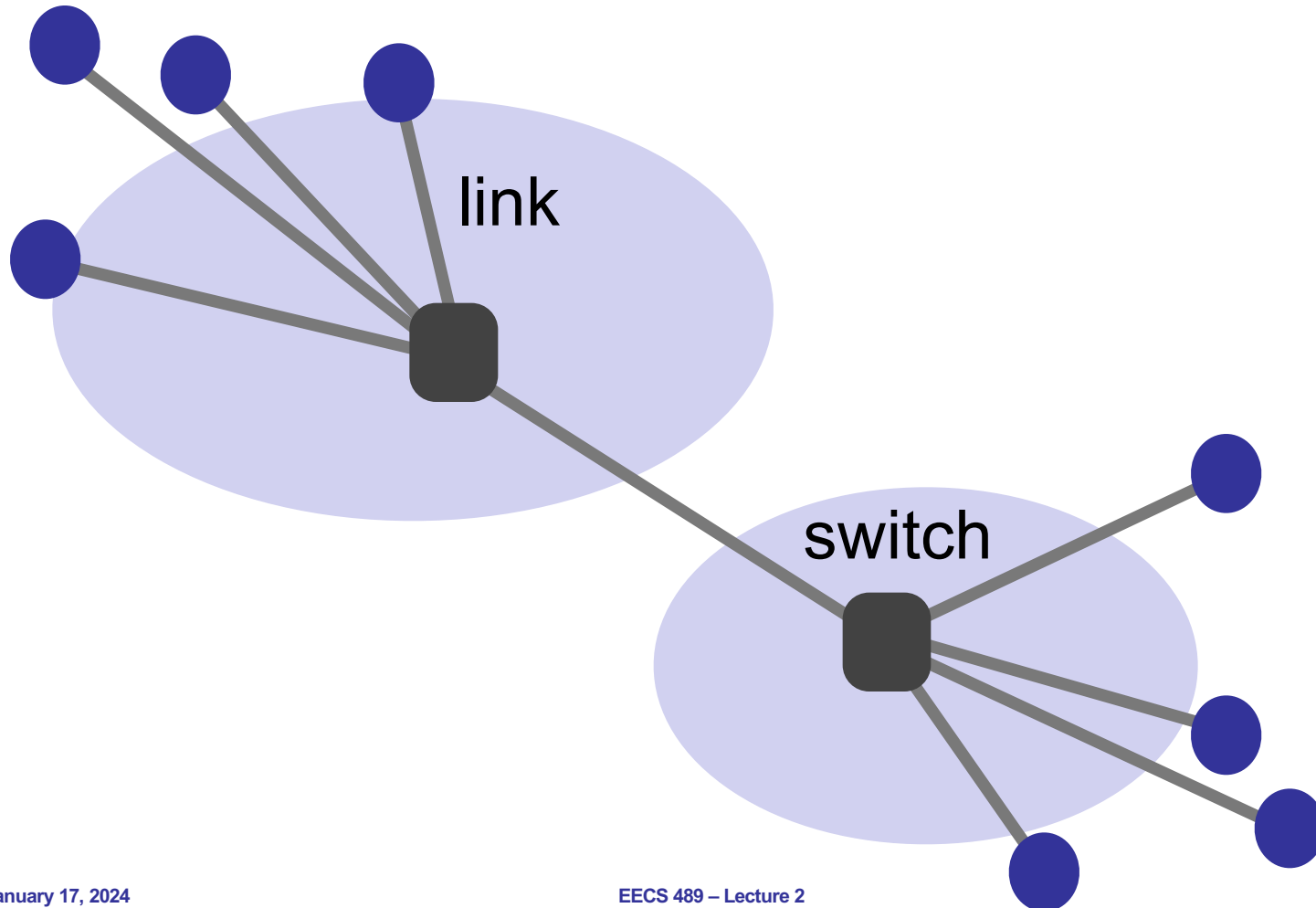
- Application
 - Web server, browser, mail, game
- Transport and network layer
 - typically part of the operating system
- Datalink and physical layer
 - hardware/firmware/drivers

What gets implemented in the network?

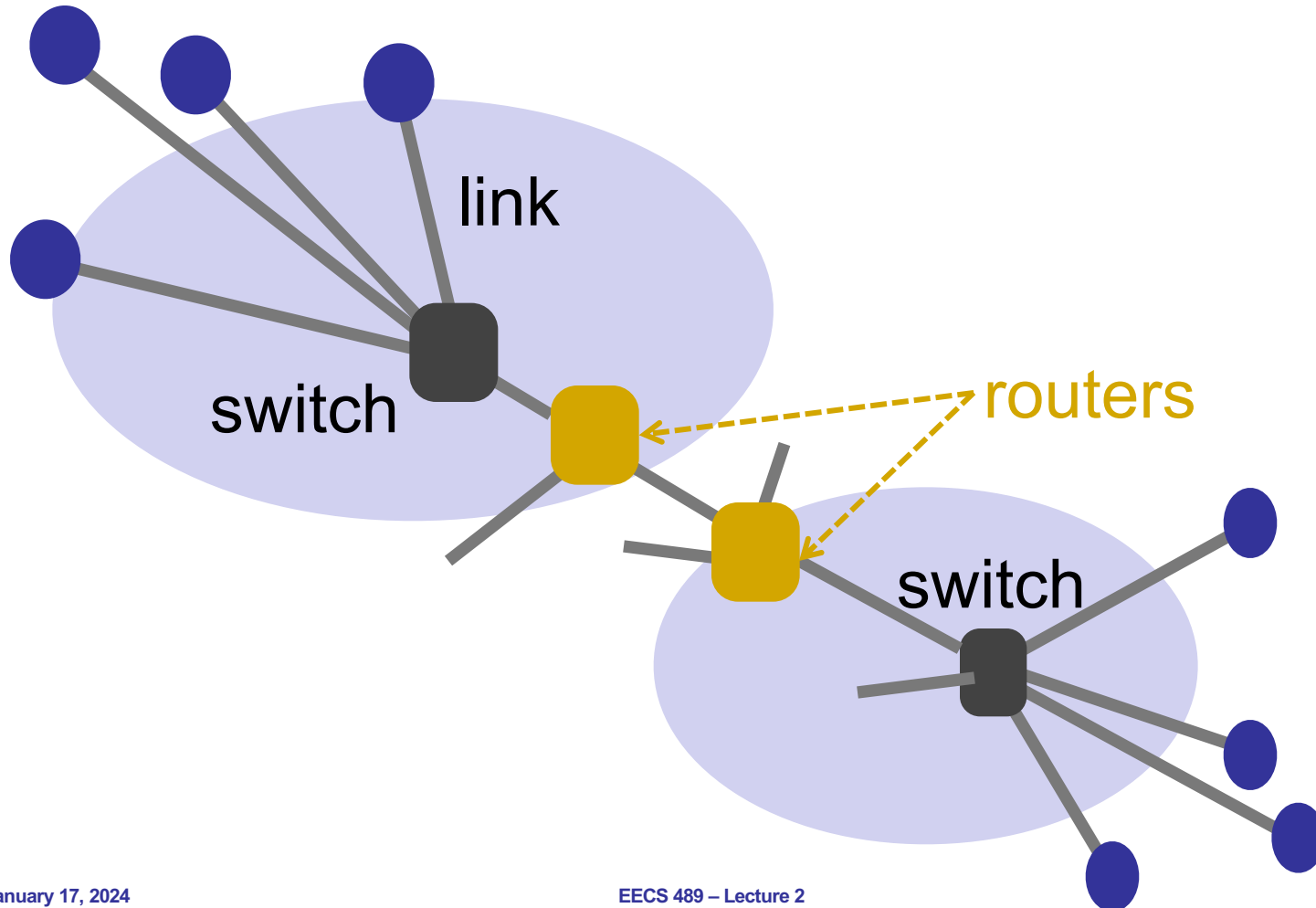
- Bits arrive on wire → physical layer (L1)
- Packets must be delivered across links and local networks → datalink layer (L2)
- Packets must be delivered between networks for global delivery → network layer (L3)

- **Switches** implement only physical and datalink layers (L1, L2)
- **Routers** implement the network layer too (L1, L2, L3)

A closer look at the network



A closer look at the network

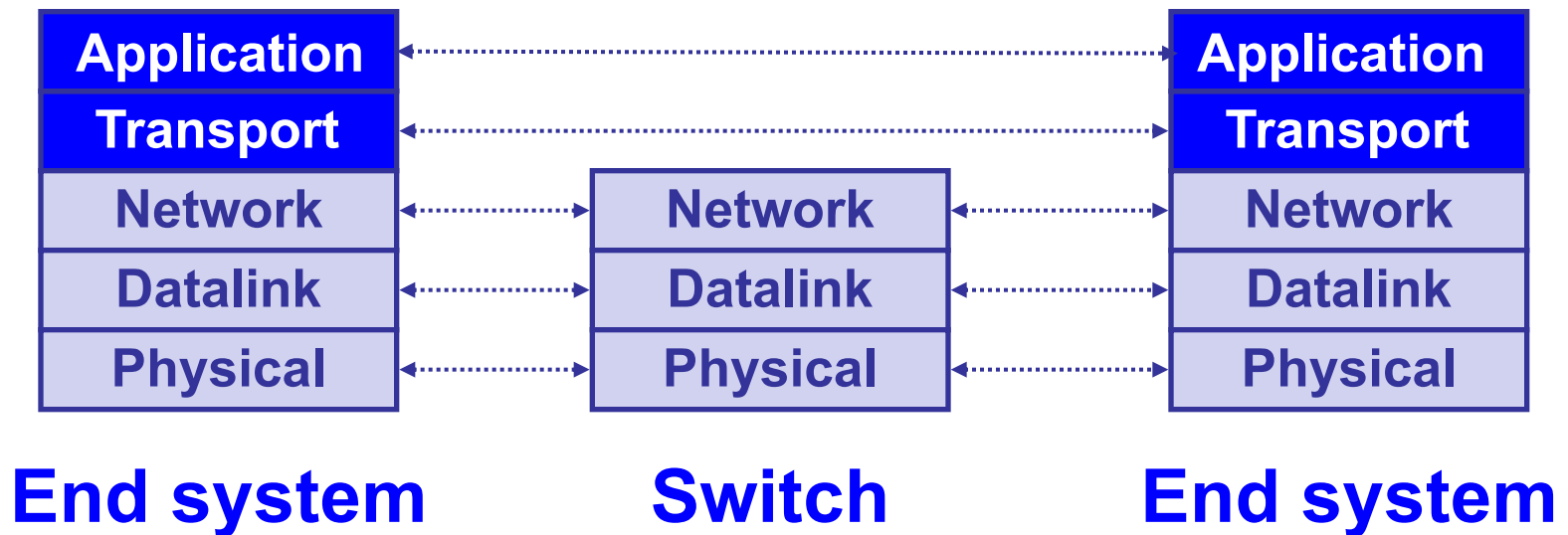


Switches vs. Routers

- 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
- Won't focus on the router/switch distinction
 - Almost all boxes support network layer these days

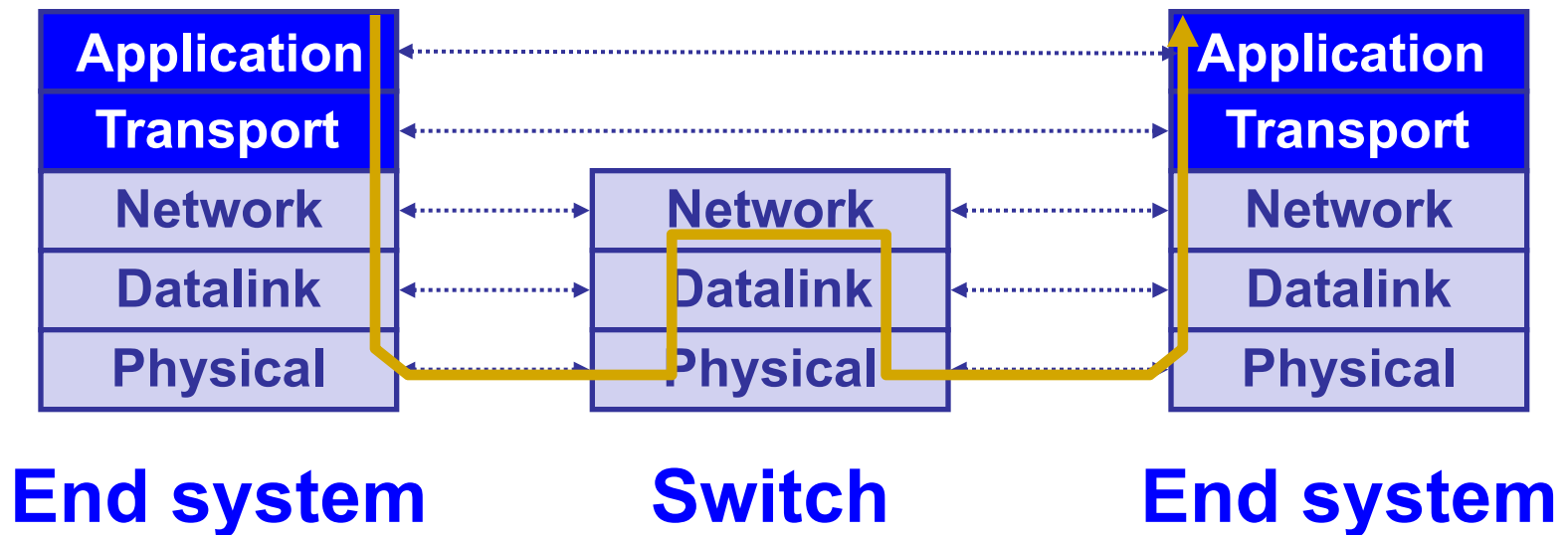
Logical communication

- A layer interact with its peers corresponding layer

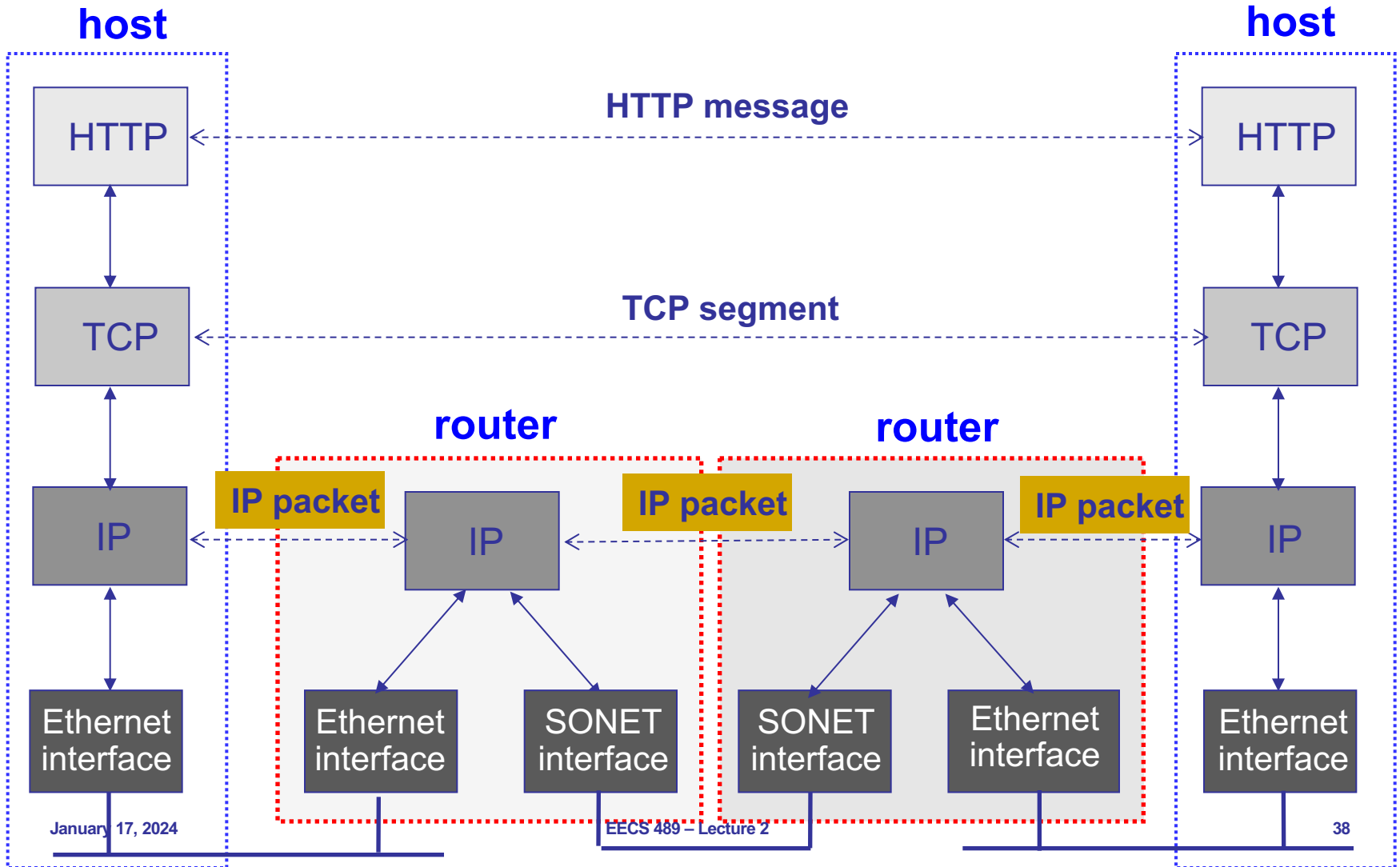


Physical communication

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



A protocol-centric diagram



Pros and cons of layering

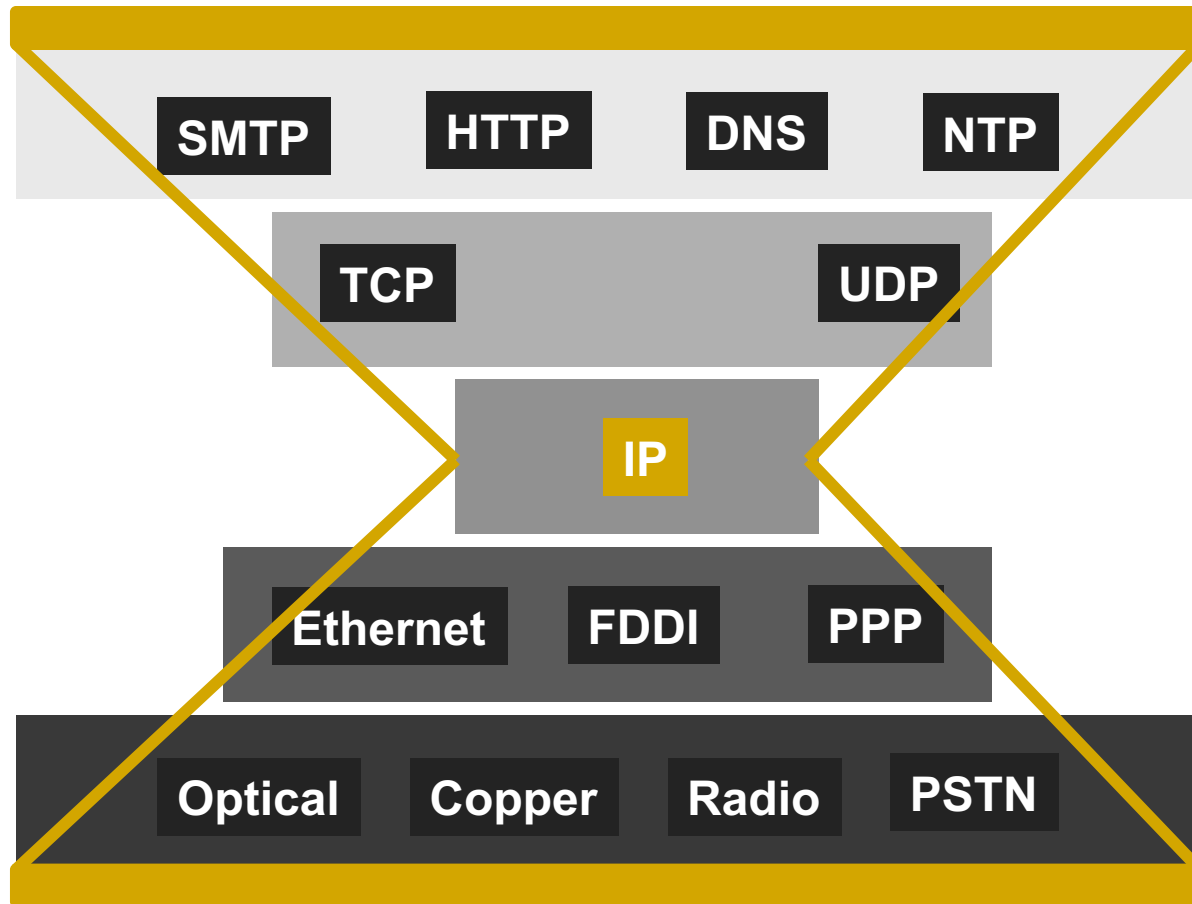
Why layers?

- Reduce complexity
- Improve flexibility

Why not?

- Higher overheads
- Cross-layer information often useful

IP is the narrow waist of the layering hourglass



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

- End-to-end arguments by Saltzer, Reed, and Clark
 - Dumb network and smart end systems
 - Functions that can be *completely* and *correctly* implemented *only* with the knowledge of application end host, should not be pushed into the network
 - Sometimes necessary to break this for performance and policy optimizations
 - **Fate sharing**: fail together or don't fail at all

Summary

- Layering is a good way to organize networks
- Unified Internet layer decouples applications from networks
- E2E argument encourages us to keep IP simple