

Networking Introduction

CSE 333

Instructor: Hannah C. Tang

Teaching Assistants:

Deeksha Vatwani Hannah Jiang

Jen Xu

Leanna Nguyen Nam Nguyen

Sayuj Shahi

Tanay Vakharia Wei Wu

Yiqing Wang

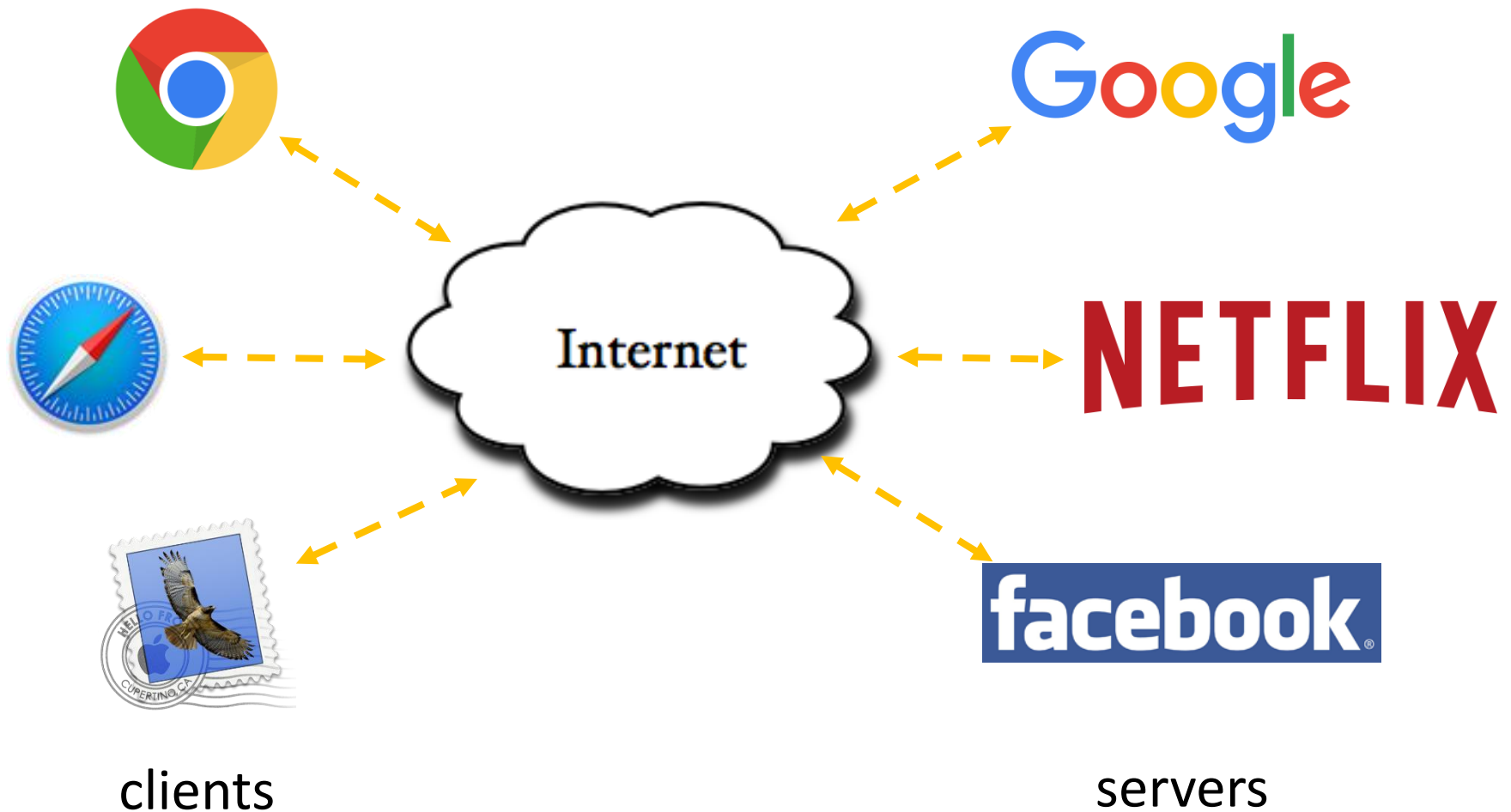
Zohar Le

Lecture Outline

- ❖ Introduction to Networks
 - Layers upon layers upon layers...



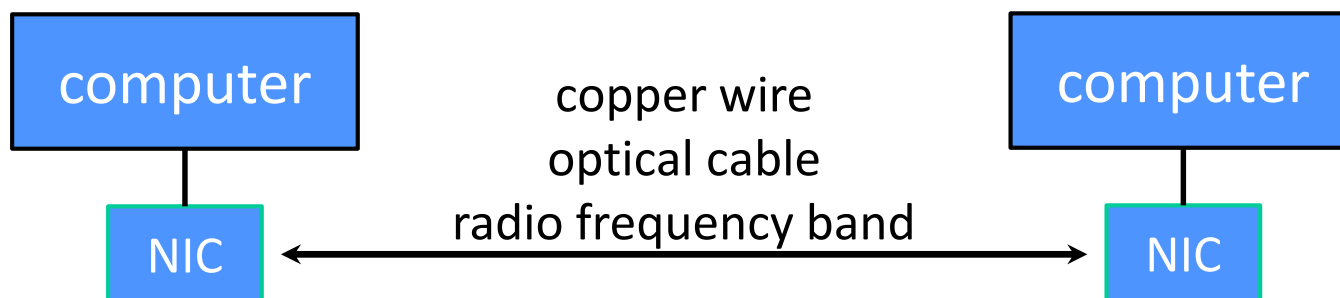
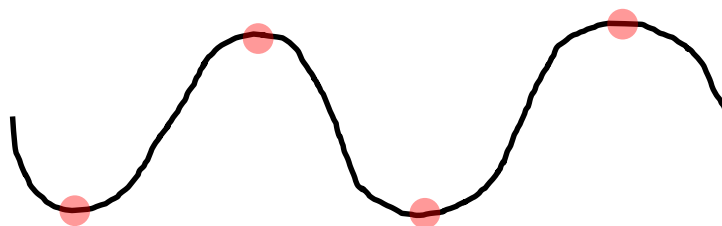
Networks From 10,000 ft



The Physical Layer

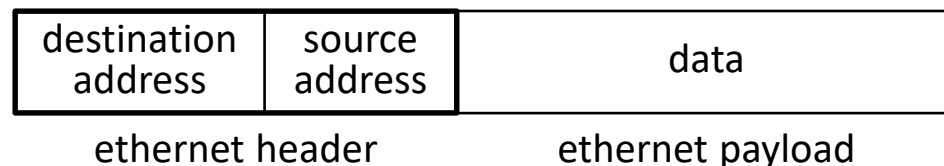
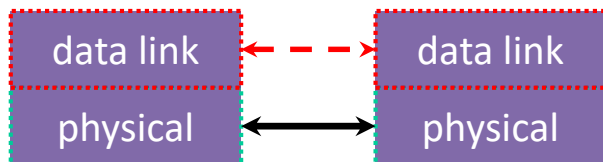
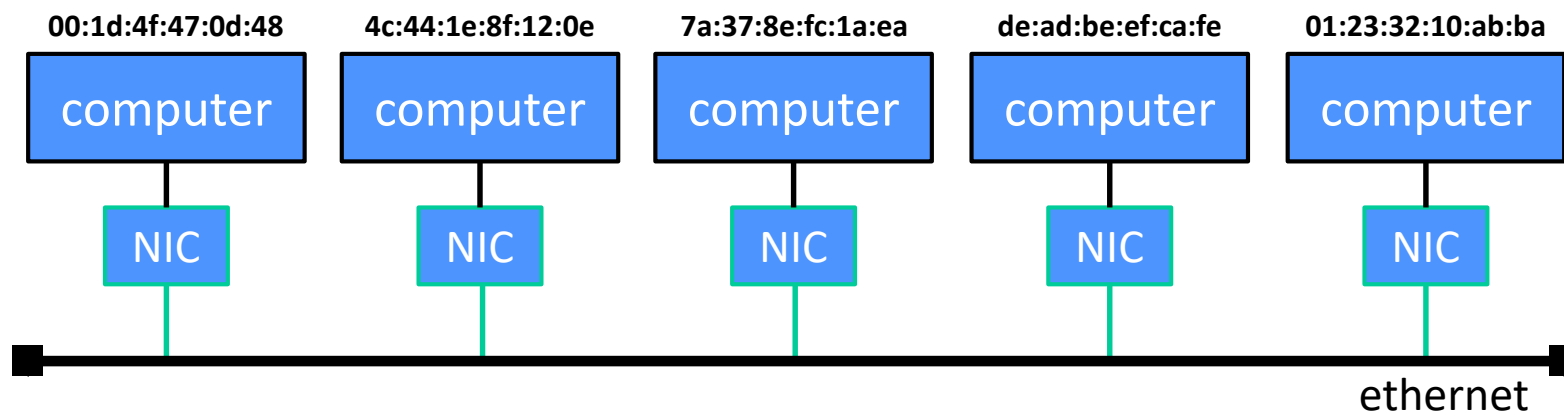
- ❖ Individual bits are modulated onto a wire or transmitted over radio
 - Physical layer specifies how bits are encoded at a signal level
 - Many choices, e.g., encode “1” as +1v, “0” as -0v; or “0”=+1v, “1”=-1v, ...

0 1 0 1



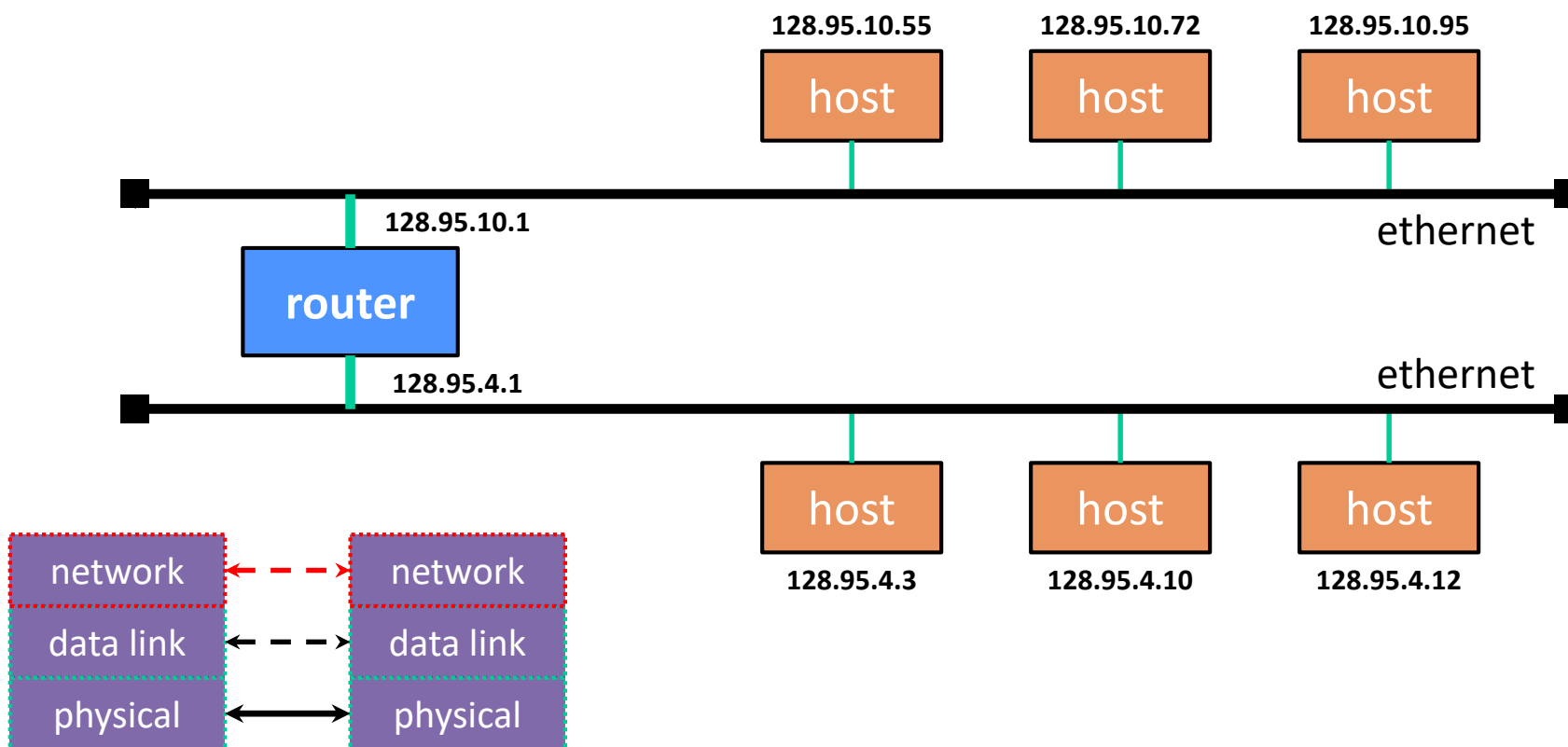
The Data Link Layer

- ❖ Multiple computers on a LAN contend for the network medium
 - Media access control (MAC) specifies how computers cooperate
 - Link layer also specifies how bits are “packetized” and network interface controllers (NICs) are addressed



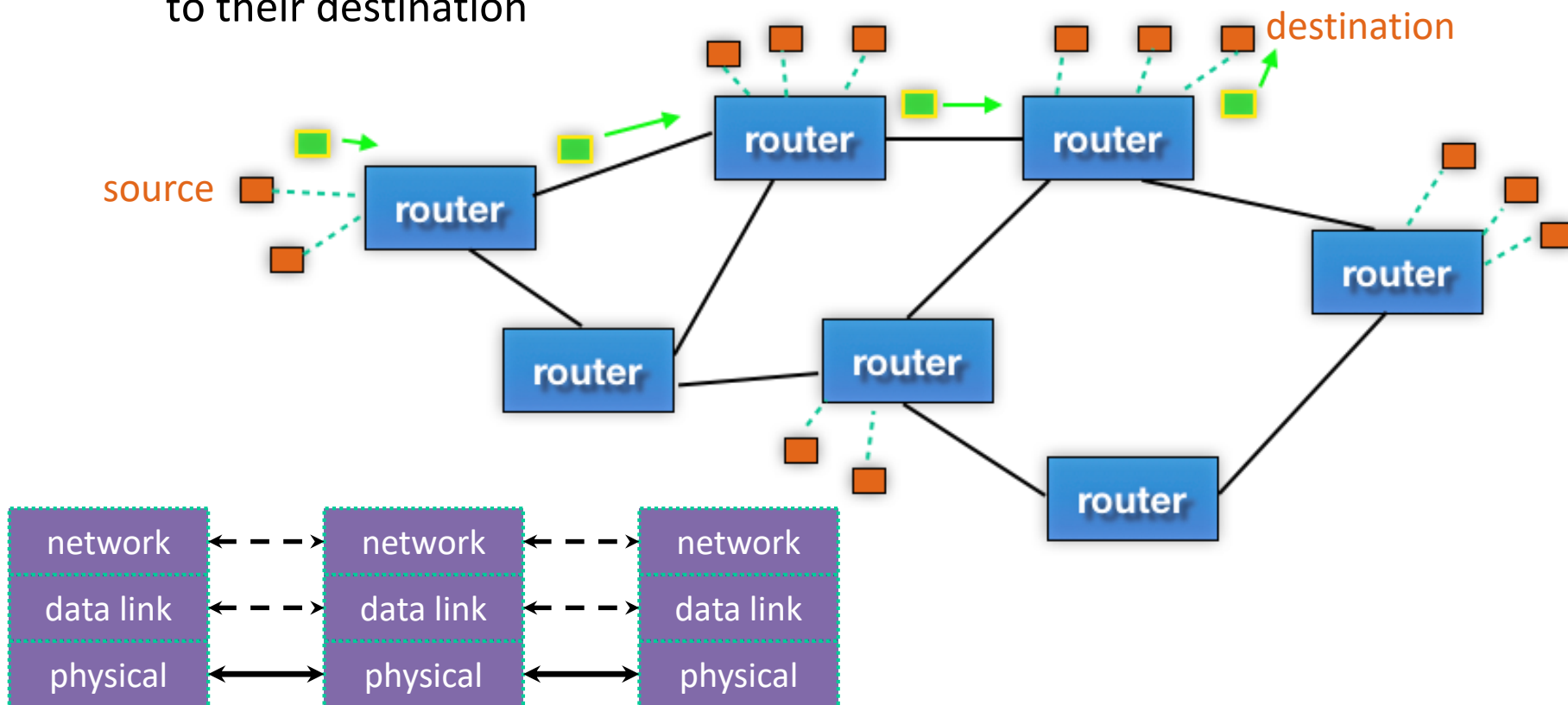
The Network Layer (IP)

- ❖ Internet Protocol (IP) routes packets across multiple networks
 - Every computer has a unique IP address
 - Individual networks are connected by routers that span networks



The Network Layer (IP)

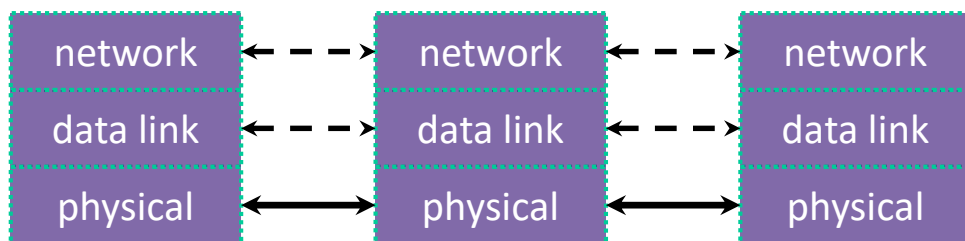
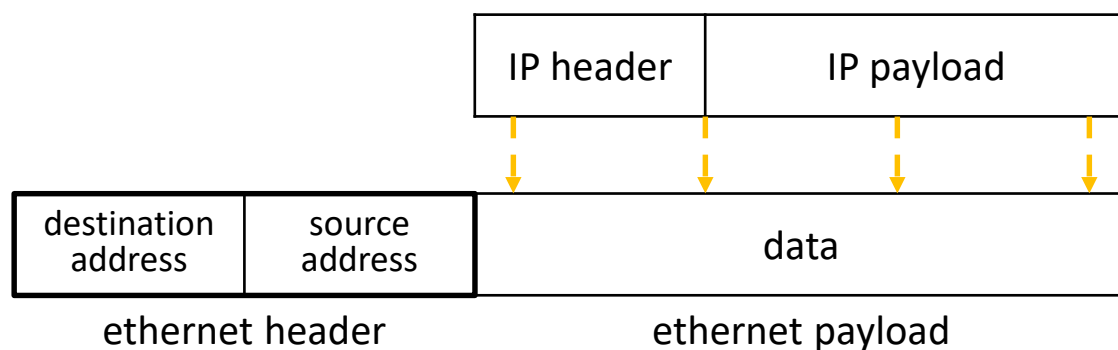
- ❖ There are protocols to:
 - Let a host map an IP to MAC address on the same network
 - Let a router learn about other routers to get IP packets one step closer to their destination



The Network Layer (IP)

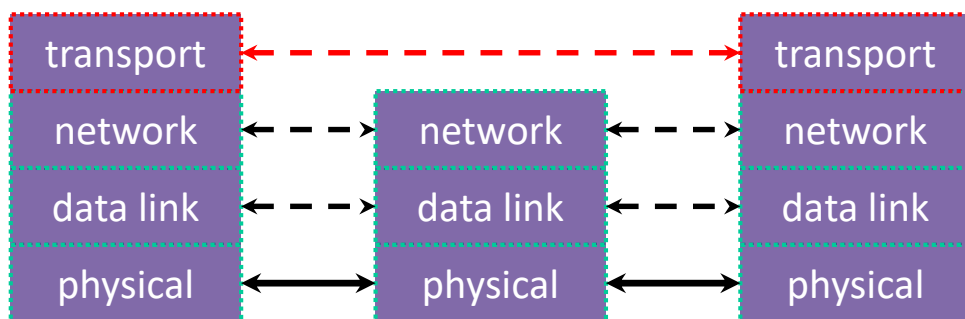
❖ Packet encapsulation:

- An IP packet is encapsulated as the payload of an Ethernet frame
- As IP packets traverse networks, routers pull out the IP packet from an Ethernet frame and plunk it into a new one on the next network



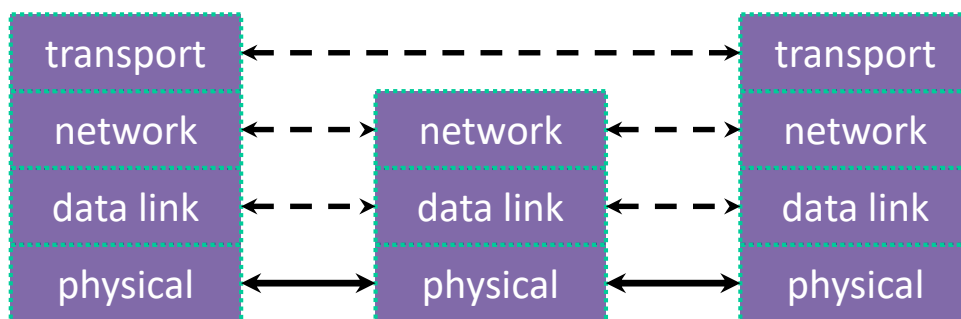
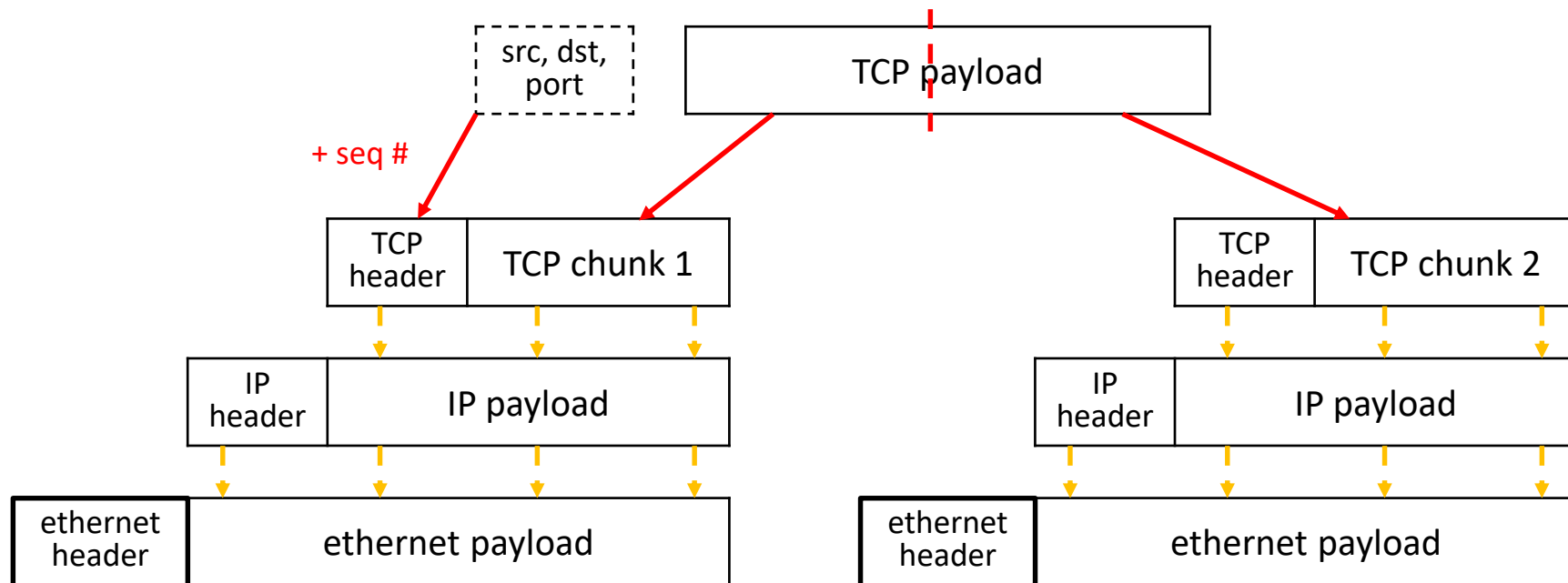
The Transport Layer (TCP)

- ❖ Transmission Control Protocol (TCP):
 - Provides applications with reliable, ordered, congestion-controlled byte streams
 - Sends stream data as multiple IP packets (differentiated by sequence numbers) and retransmits them as necessary
 - When receiving, puts packets back in order and detects missing packets
 - A single host (IP address) can have up to $2^{16} = 65,535$ “ports”
 - Kind of like an apartment number at a postal address (your applications are the residents who get mail sent to an apt. #)



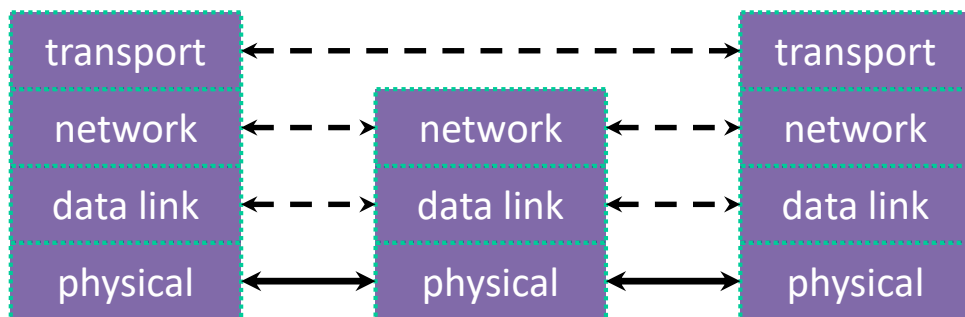
The Transport Layer (TCP)

- ❖ Packet encapsulation – one more nested layer!



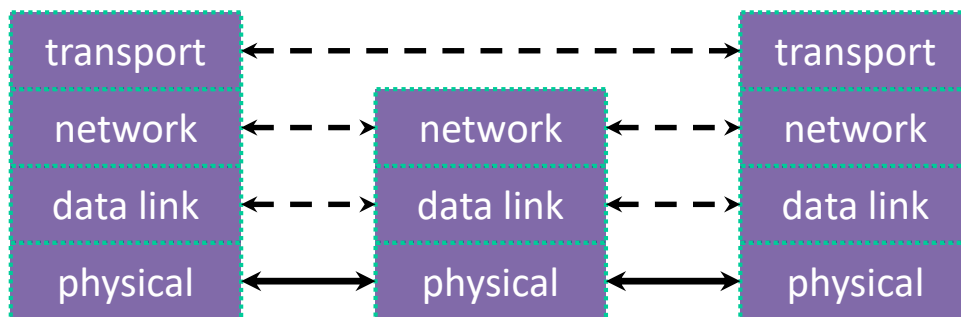
The Transport Layer (TCP)

- ❖ Applications use OS services to establish TCP streams:
 - The “Berkeley sockets” API
 - A set of OS system calls
 - Clients **connect** () to a server IP address + application port number
 - Servers **listen** () for and **accept** () client connections
 - Clients and servers **read** () and **write** () data to each other



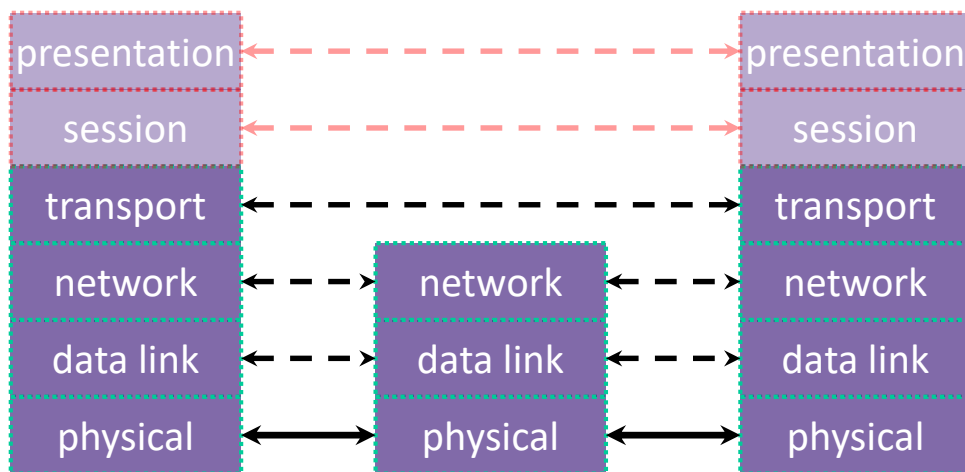
The Transport Layer (UDP)

- ❖ User Datagram Protocol (UDP):
 - Provides applications with *unreliable* packet delivery
 - UDP is a really thin, simple layer on top of IP
 - Datagrams still are fragmented into multiple IP packets



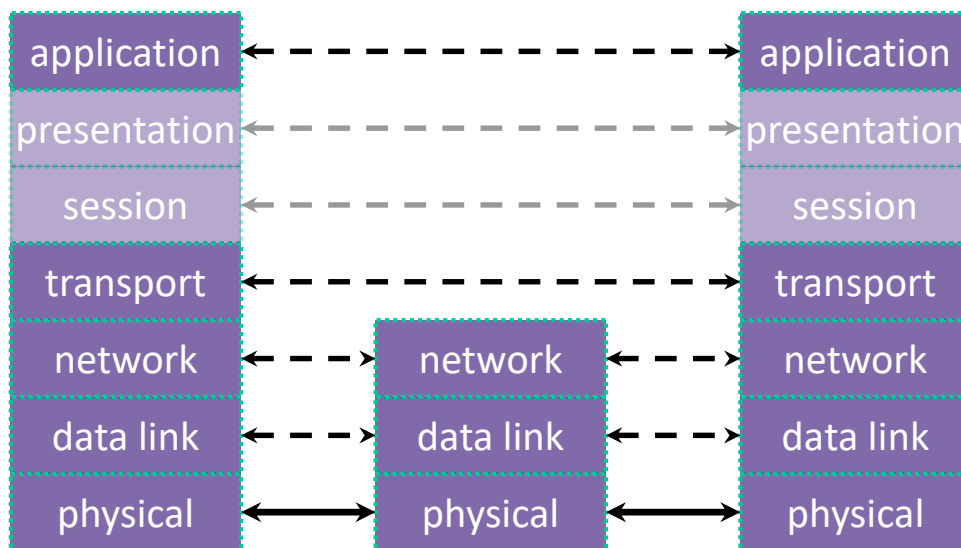
The (Mostly Missing) Layers 5 & 6

- ❖ Layer 5: Session Layer
 - Supposedly handles establishing and terminating application sessions
 - Remote Procedure Call (RPC) kind of fits in here
- ❖ Layer 6: Presentation Layer
 - Supposedly maps application-specific data units into a more network-neutral representation
 - Encryption (SSL) kind of fits in here



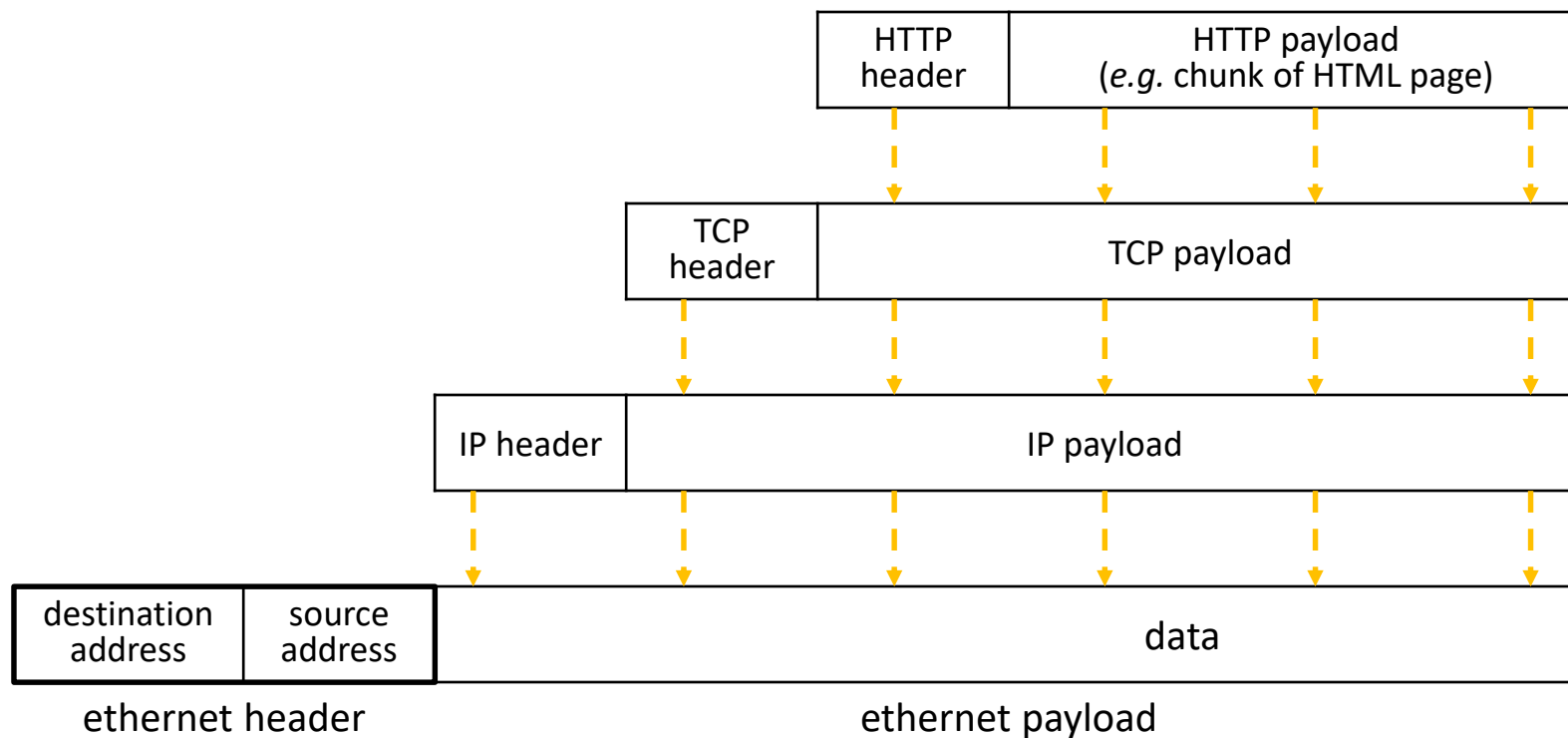
The Application Layer

- ❖ Application protocols
 - The format and meaning of messages between application entities
 - Example: HTTP is an application-level protocol that dictates how web browsers and web servers communicate
 - HTTP is implemented *on top of* TCP streams



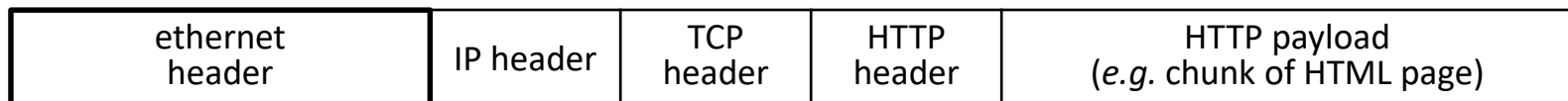
The Application Layer

❖ Packet encapsulation:



The Application Layer

- ❖ Packet encapsulation:



The Application Layer

- ❖ Popular application-level protocols:
 - **DNS:** translates a domain name (*e.g.* www.google.com) into one or more IP addresses (*e.g.* 74.125.197.106)
 - Domain Name System
 - An hierarchy of DNS servers cooperate to do this
 - **HTTP:** web protocols
 - Hypertext Transfer Protocol
 - **SMTP, IMAP, POP:** mail delivery and access protocols
 - Secure Mail Transfer Protocol, Internet Message Access Protocol, Post Office Protocol
 - **SSH:** secure remote login protocol
 - Secure Shell
 - **bittorrent:** peer-to-peer, swarming file sharing protocol

netcat demo (if time)

- ❖ netcat (`nc`) is “a computer networking utility for reading from and writing to network connections using TCP or UDP”
 - <https://en.wikipedia.org/wiki/Netcat>
 - Listen on port: `nc -l <port>`
 - Connect: `nc <IPaddr> <port>`
 - Local host: `127.0.0.1`

The Future of Networking?

