Introduction to Networking

Computer Systems Programming, Spring 2025

Instructor: Travis McGaha

Teaching Assistants:

Andrew Lukashchuk
Angie Cao
Aniket Ghorpade
Ashwin Alaparthi
Austin Lin
Hassan Rizwan
Lobi Zhao
Pearl Liu
Perrie Quek

Poll Everywhere

pollev.com/tqm

- What do you know about networks?

Administrivia

- HW09 – Threads “Grep”
  - Posted☺
  - Due Friday 4/11 at midnight, leaving open till Sunday night tho
  - AG posted soon
  - Some hints gone over in Recitation this week

- Final Project Details Coming soon-ish

Lecture Outline

- **Introduction to Networks**
  - **Layers upon layers upon layers…**

Today's Goals

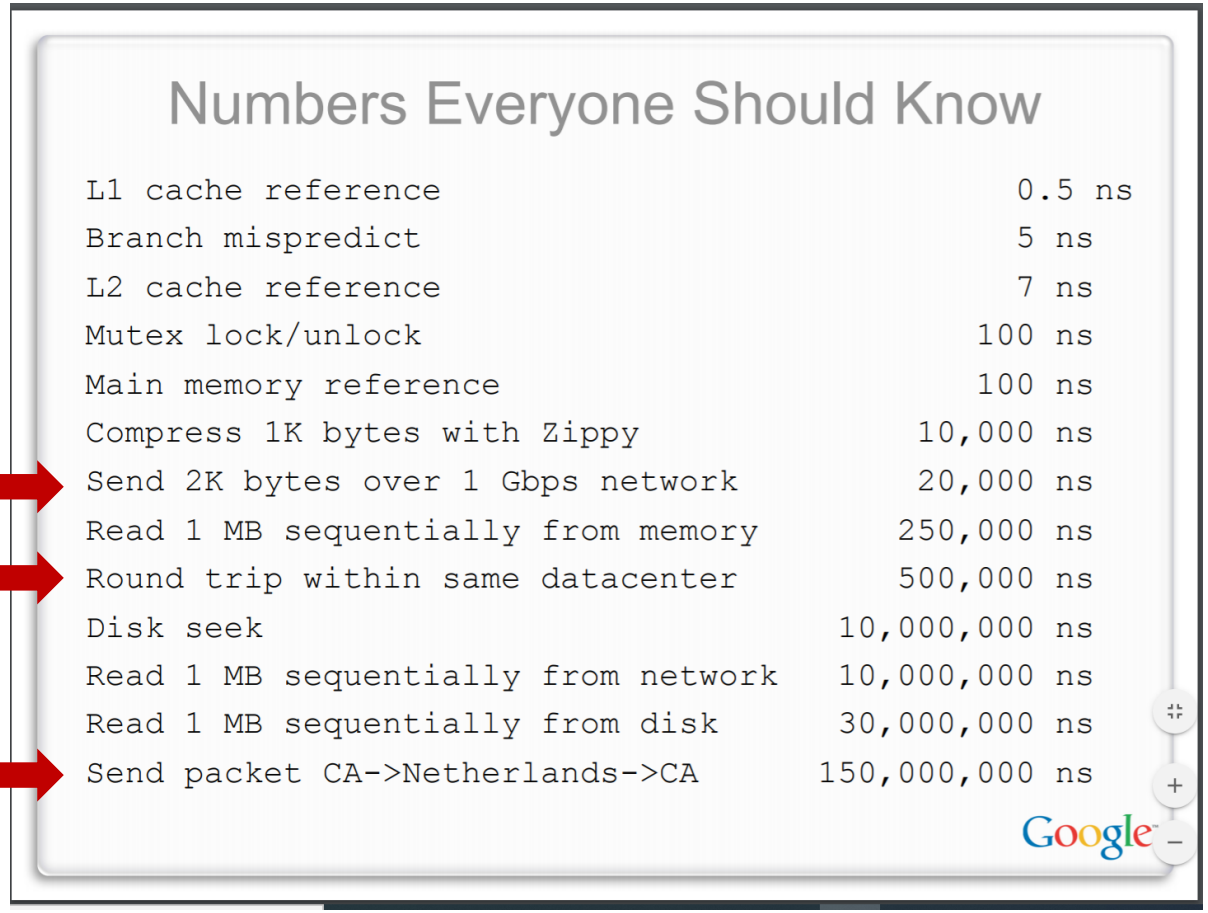
- Networking is a very common programming feature
  - You will likely have to create a program that will read/write over the network at some point in your career
- We want to give you a basic, high-level understanding of how networks work before you use them
  - Lecture will be more "story-like;" we will purposefully skip over most of the details, but hopefully you will learn something new about the Internet today!
  - Take CIS 5530 if you want to know more about the implementations of networks
    CIS 5050 is another option
- Let's also examine "the network" as a *system*
  - Inputs? Outputs? Reliability? Efficiency?

pollev.com/tqm

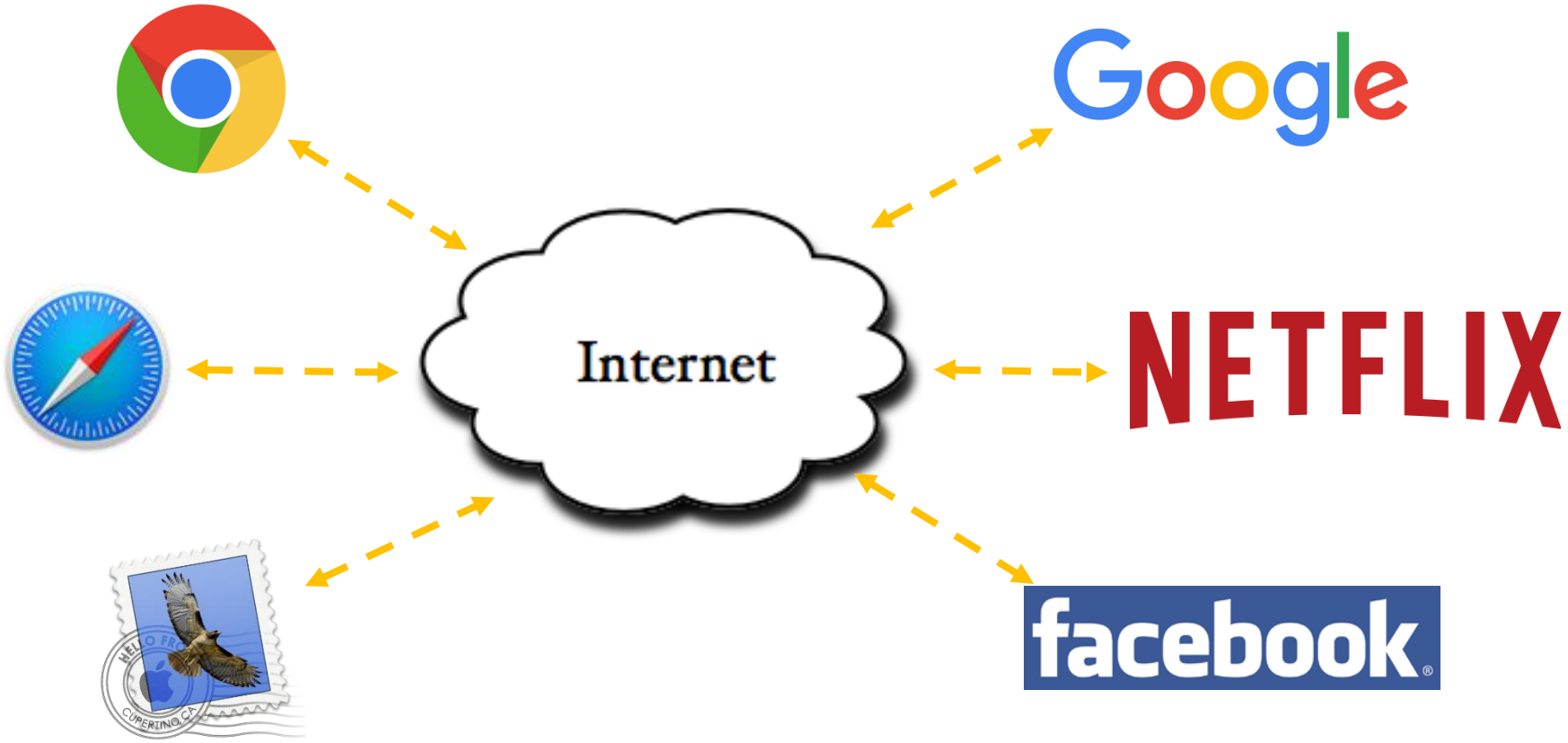
- Which of these are you familiar with? Do you know what they are?
  - HTML
  - HTTP
  - TCP
  - UDP
  - IP Address
  - Port
  - Mac Address

"Network" Latency is Highly Variable

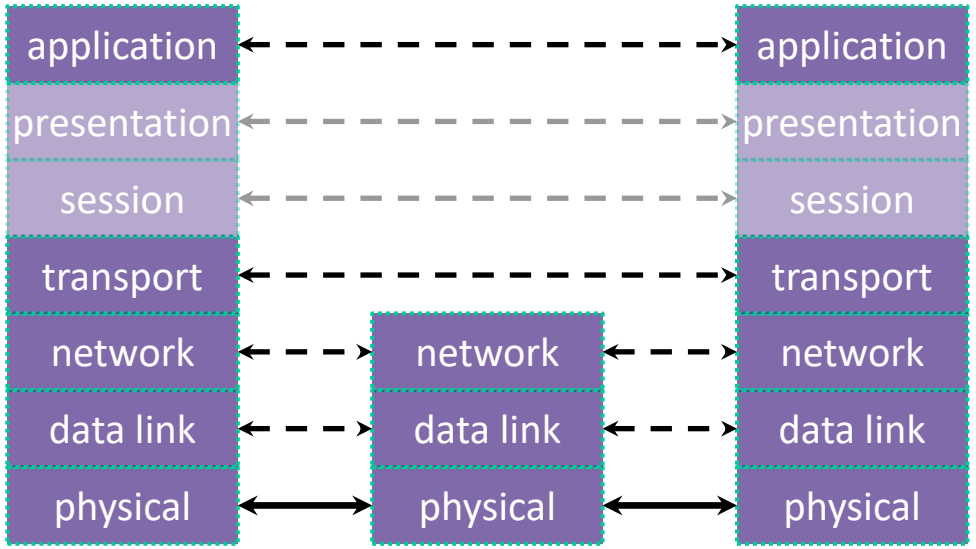
- Jeff Dean's "Numbers Everyone Should Know" (LADIS '09)



Networks From 10,000 ft

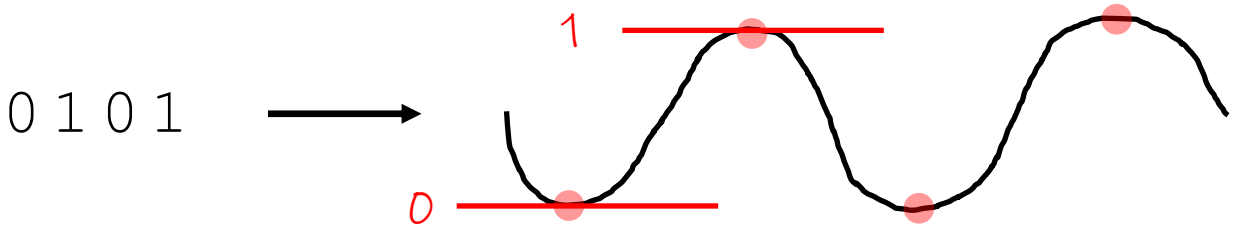


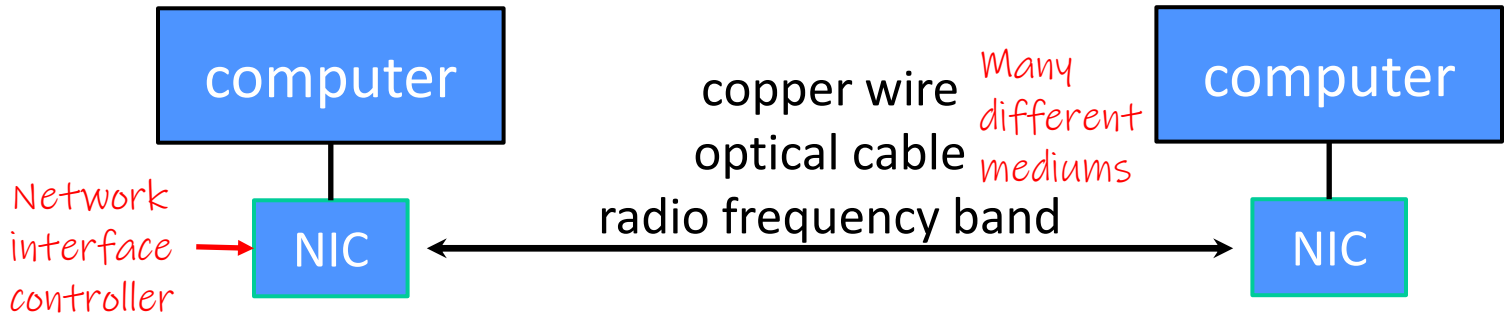
7-Layer OSI Model



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, …





Materials Matter – Latency

- Fiber optic cables are lower-latency and higher-bandwidth than traditional copper wiring
  - Much of the internet's "long haul" data is transmitted on these
  - (signal attenuation is much better too)

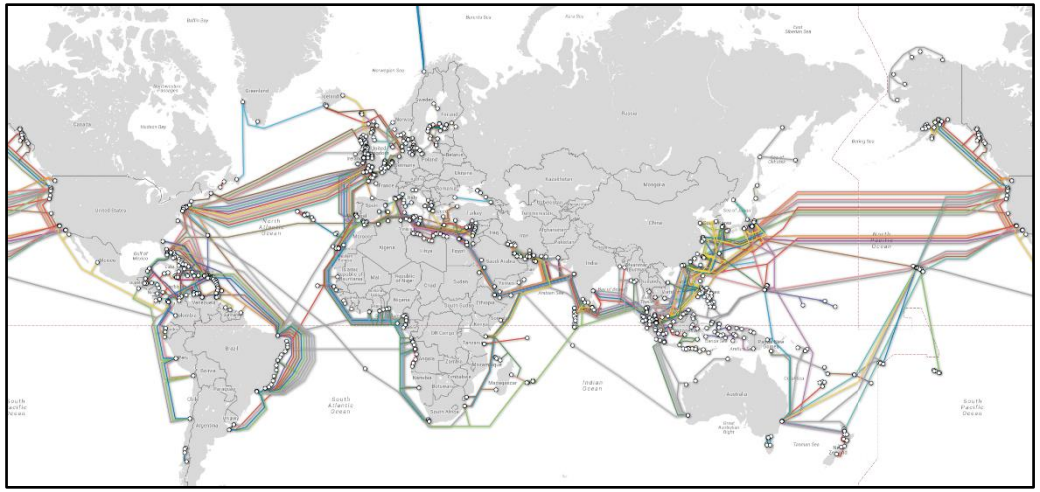
Poll Everywhere

pollev.com/tqm

- If you had to guess a place on the map with higher internet speeds, where would you guess?
  - If you had to guess a place with slower internet speeds?

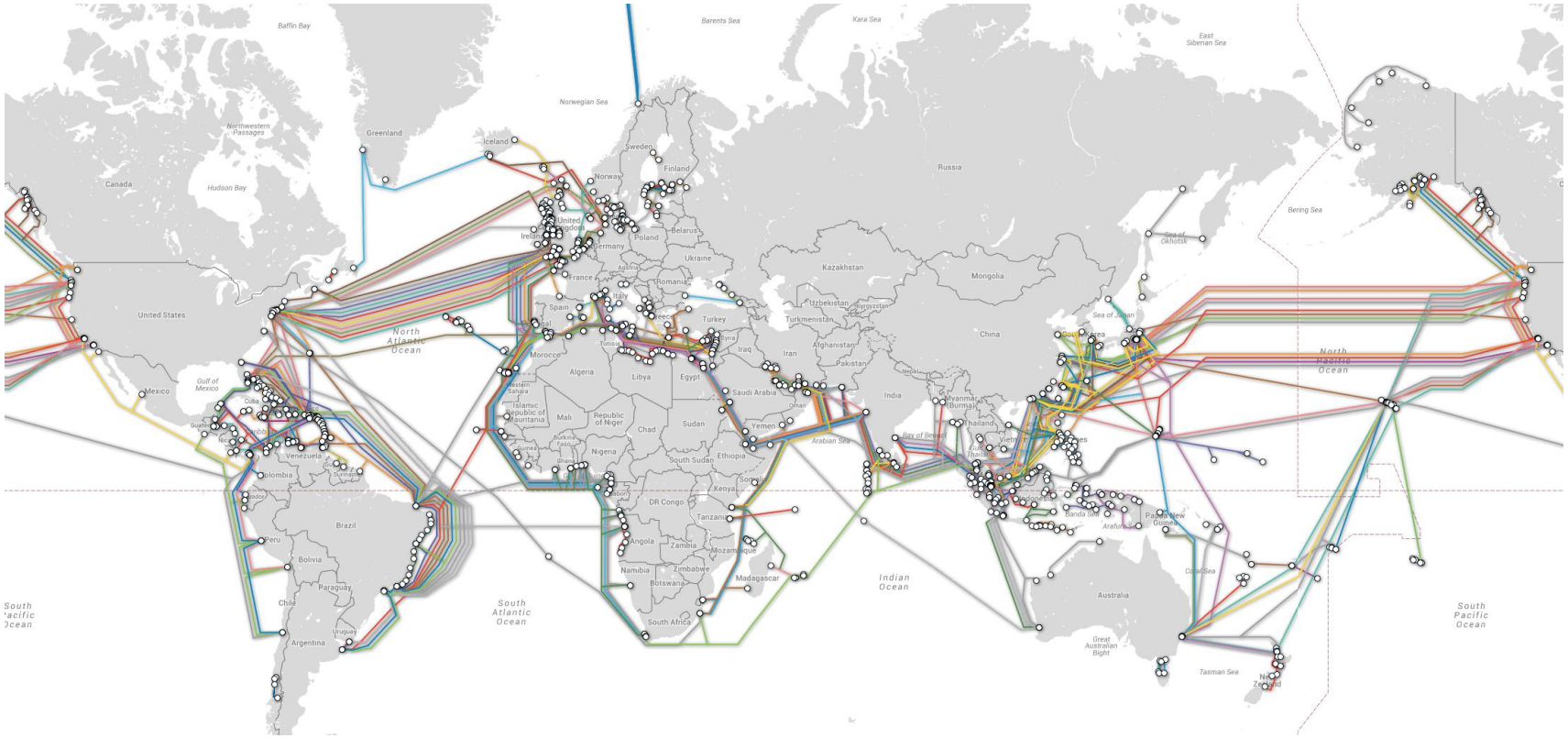
Topology Matters – Latency and Reliability

- Some places are surprisingly well- or poorly-connected to “backbone” infrastructure like fiber optic cables
- Unintuitive topology can create interesting failures
  - *e.g.*, 2006 7.0-magnitude Hengchun Earthquake disrupted communications to Singapore, Philippines, Thailand, China, etc. for a month



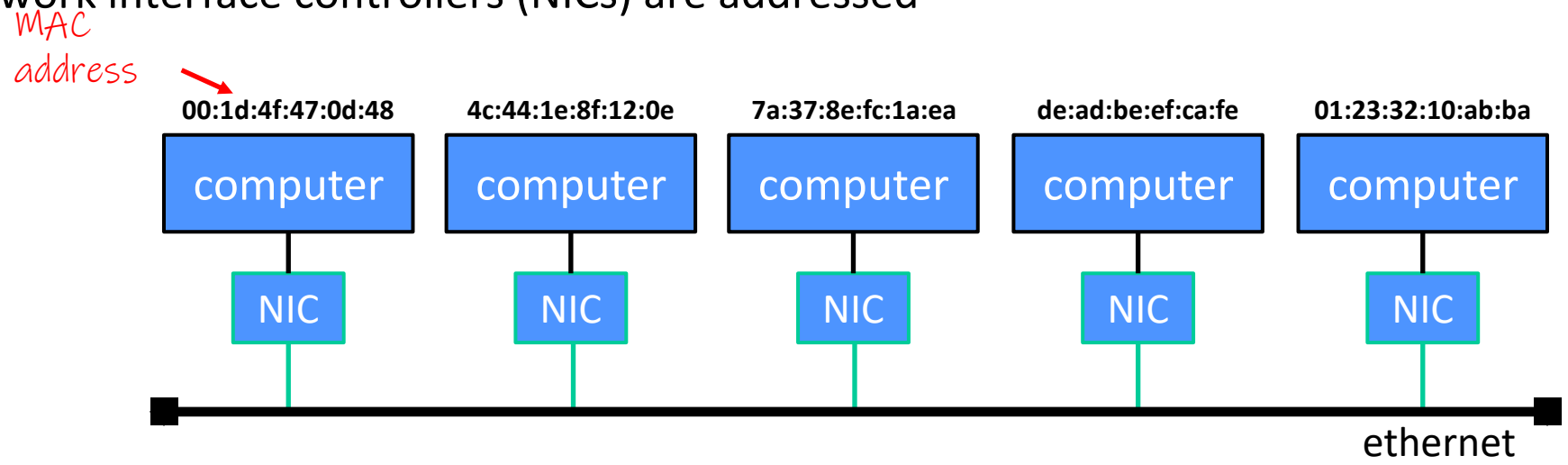
Reliability

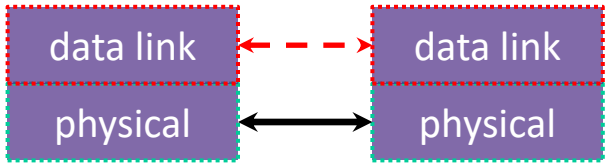
- Packet loss?
- Physical interference?
- Link going down?

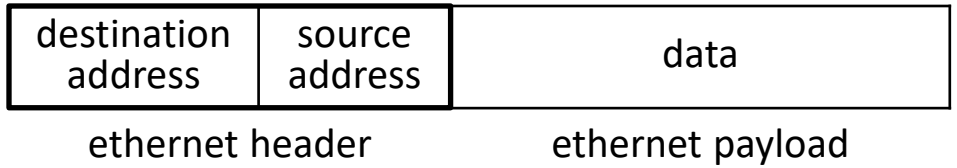


The Data Link Layer

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







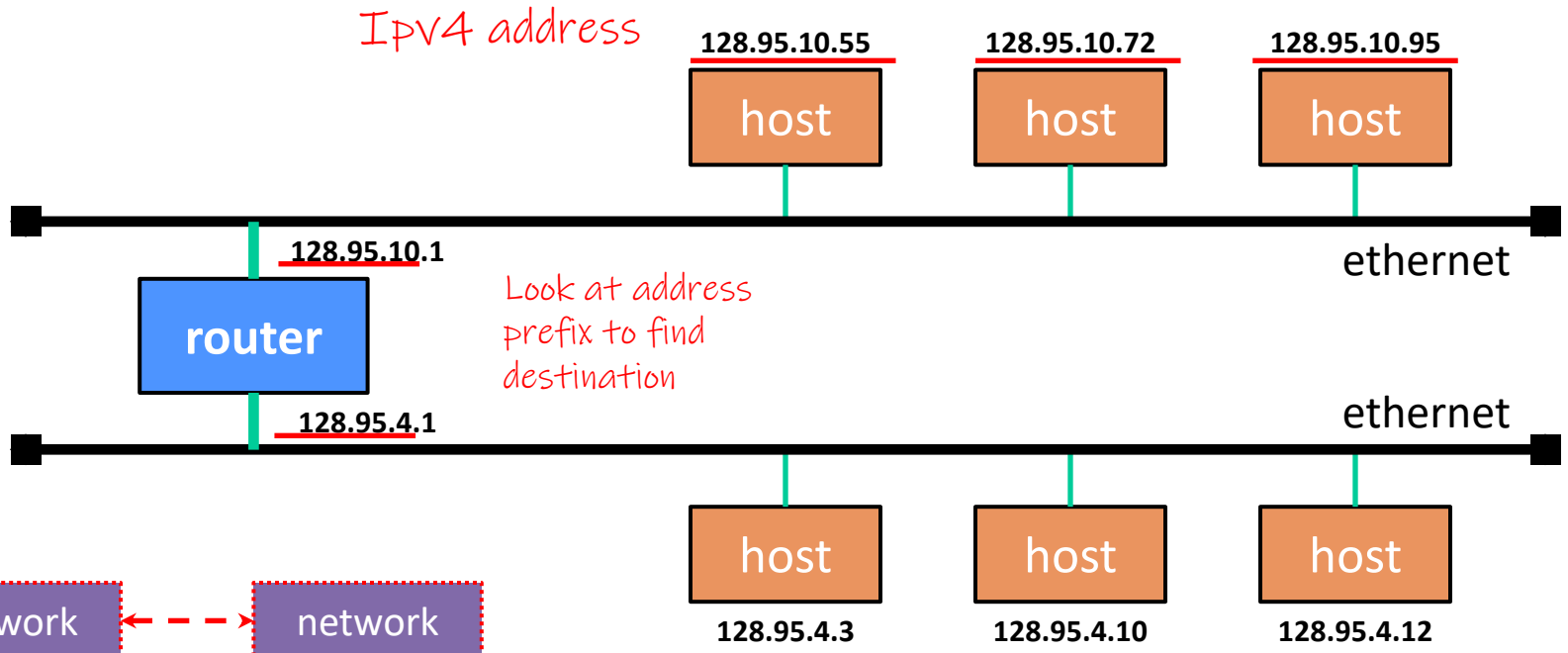
Poll Everywhere

pollev.com/tqm

- Any guesses for how we get computers to share the same physical medium?
  - (Hint: how do we handle people sharing time in a conversation?)

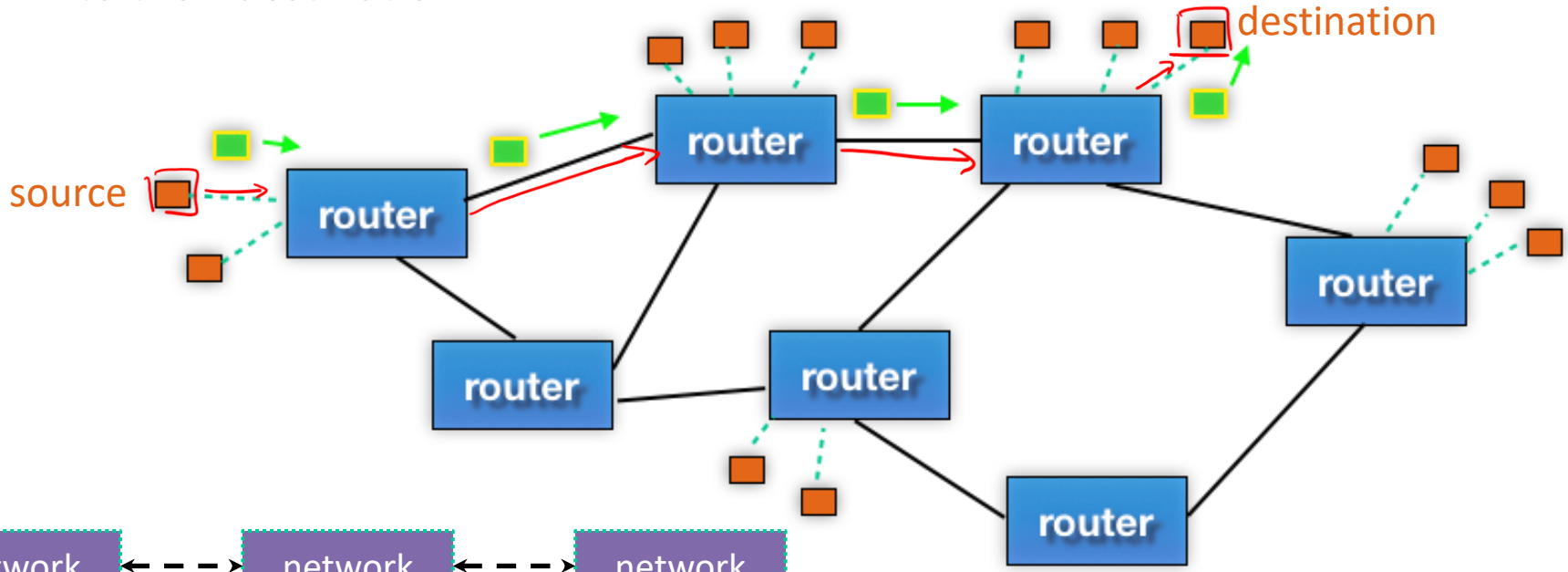
The Network Layer (IP)

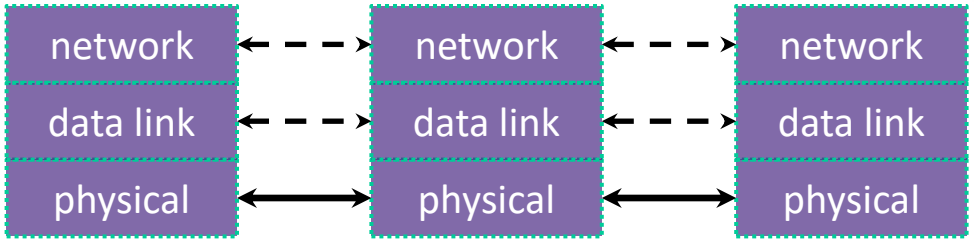
- Internet Protocol (IP) **routes packets across multiple networks**
  - Every computer has a unique IP address*
  - Individual networks are connected by routers that span networks



The Network Layer (IP)

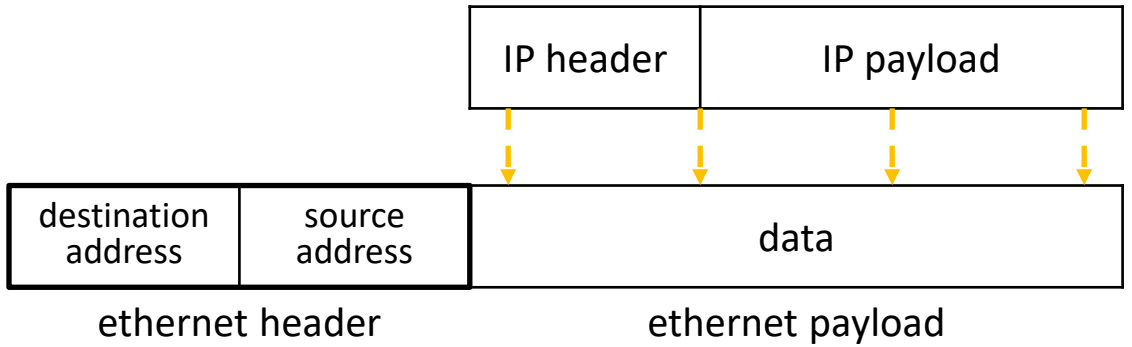
- There are protocols to:
  - Let a host map an IP to MAC address on the same network
  - Let a router learn about other routers to get IP packets one step closer to their destination

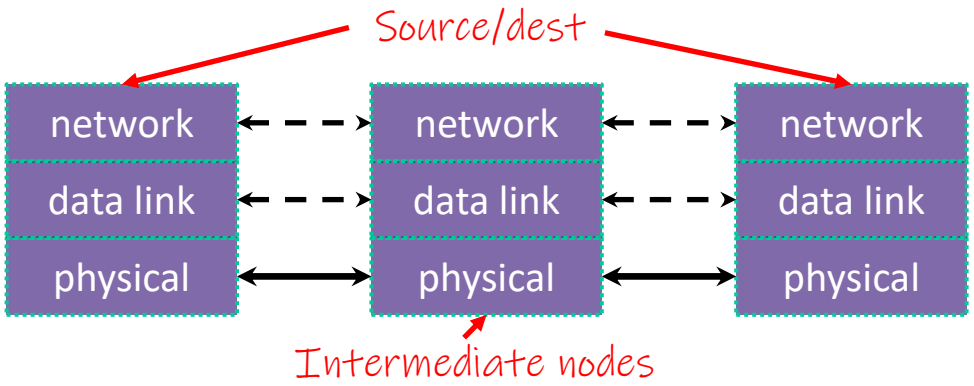




The Network Layer (IP)

- 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





Poll Everywhere

pollev.com/tqm

- If I want to send my friend (in another country) a postcard, how does it get there?
  - What do I need to do?
    - Do I just put my friends name on it and put in a mailbox?
  - How does the post office deliver it?
    - Does one person drive it from here to there?
    - How does the post office know where to send it?

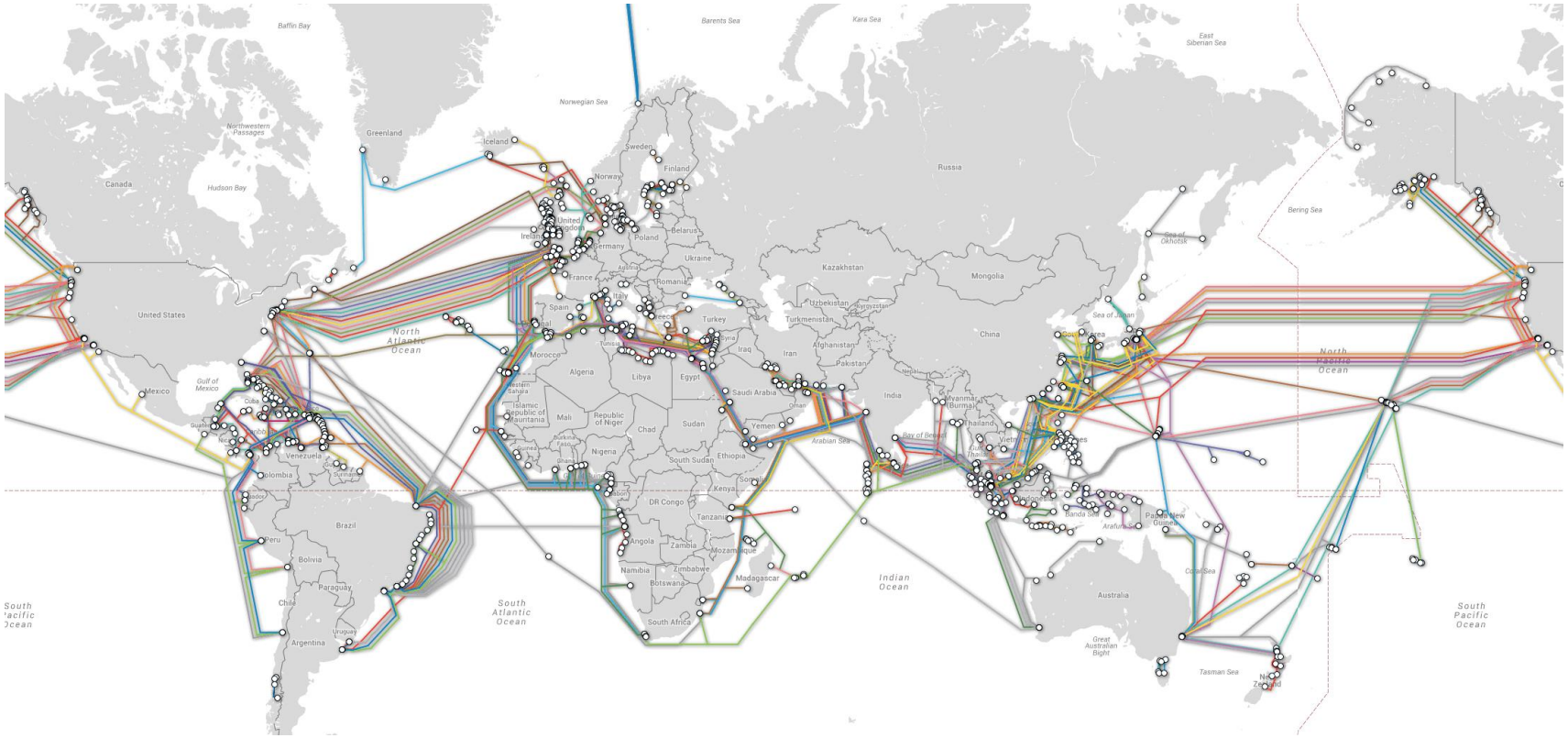
Distance Matters – Latency

- Distances within a single datacenter are smaller than distances across continents
- Even within a datacenter, distances can sometimes matter

123Net Data Center, Wikimedia

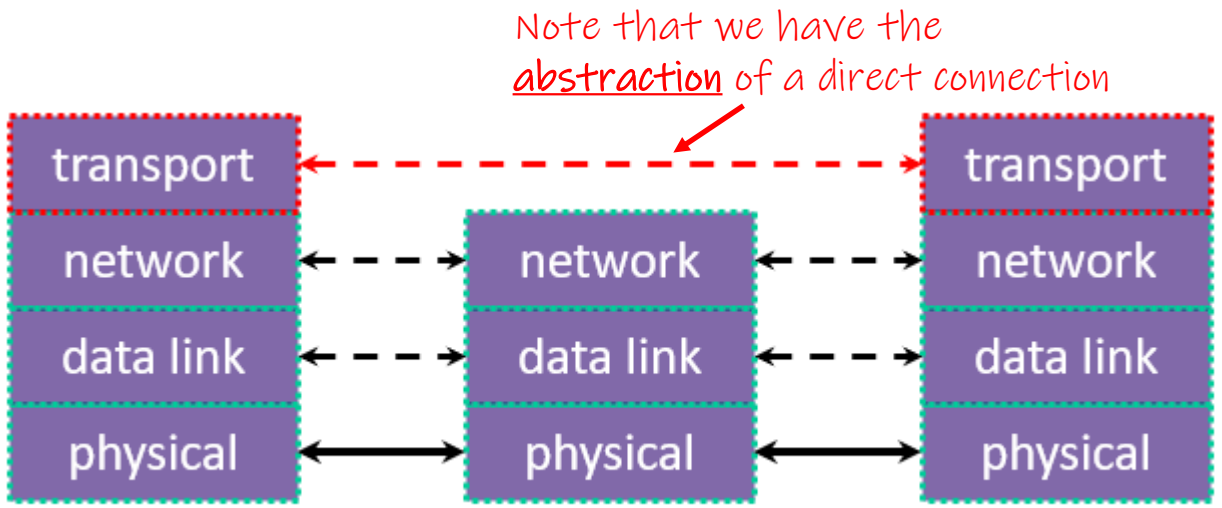
Reliability

- Packet loss?
- Physical interference?
- Link going down?



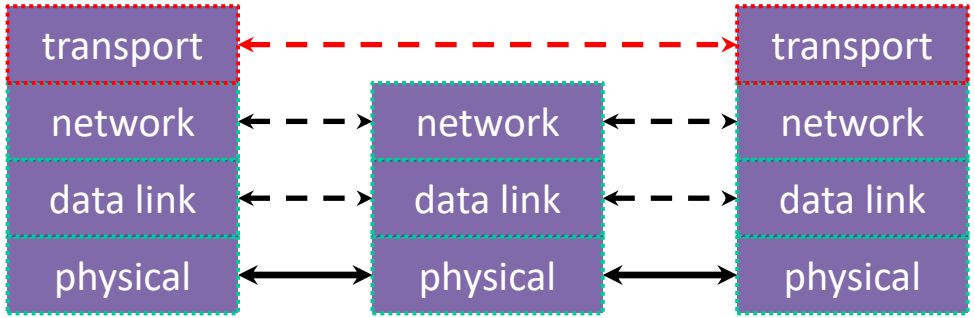
The Transport Layer

- Provides an interface to treat the network as a *data stream*
- Provides different protocols to interface between source and destination:
  - *e.g.*, Transmission Control Protocol (TCP), User Datagram Protocol (UDP)
  - These protocols still work with packets, but manages their order, reliability, multiple applications using the 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. #)

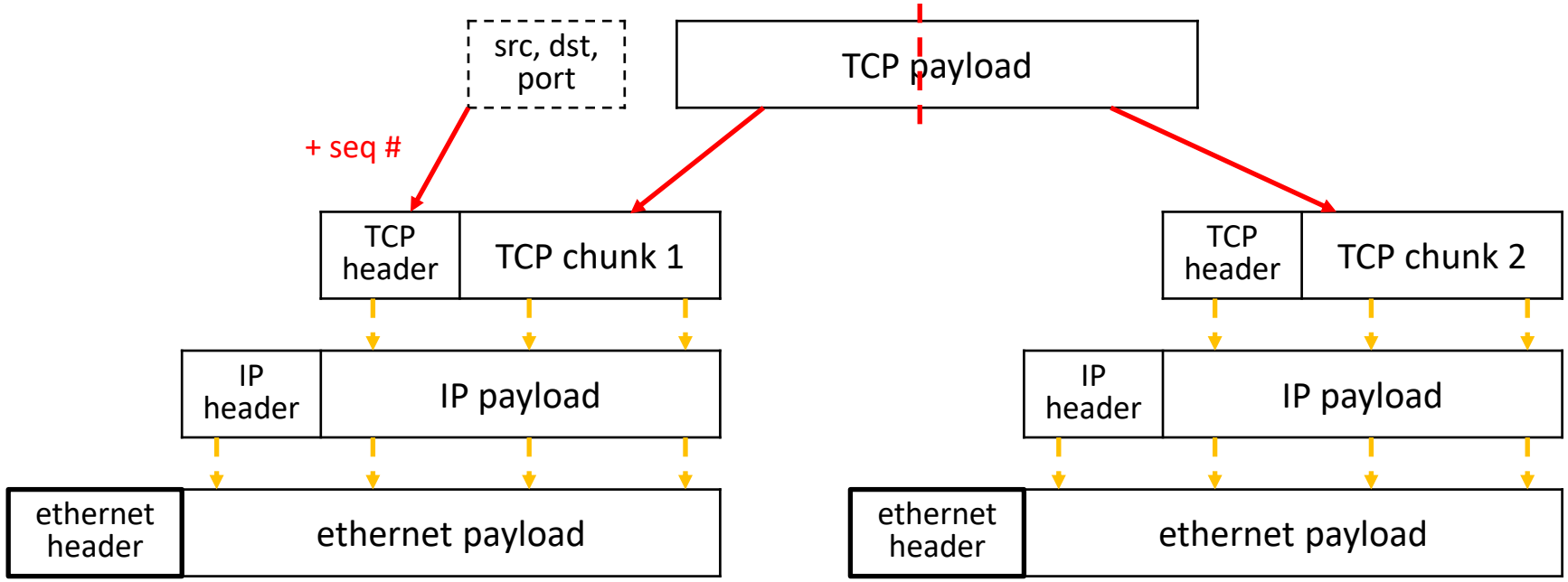


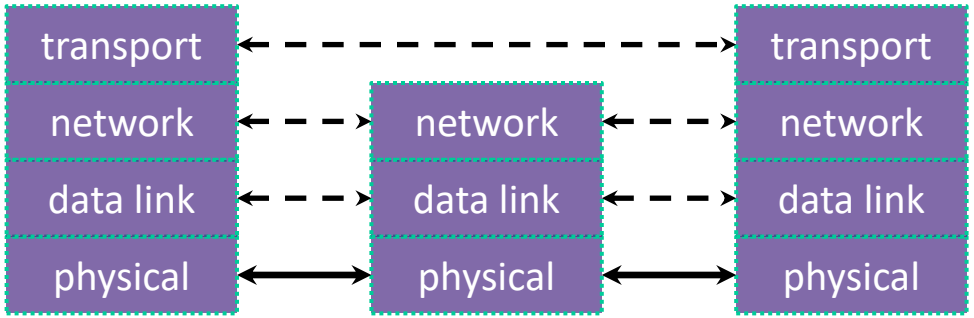
pollev.com/tqm

- Let's say we want to send a book to a friend in another state, but we can only send post cards (small pieces of paper)
  - How do we send the data to our friend?
  - How do we ensure the data gets to our friend?

The Transport Layer (TCP)

- Packet encapsulation – one more nested layer!

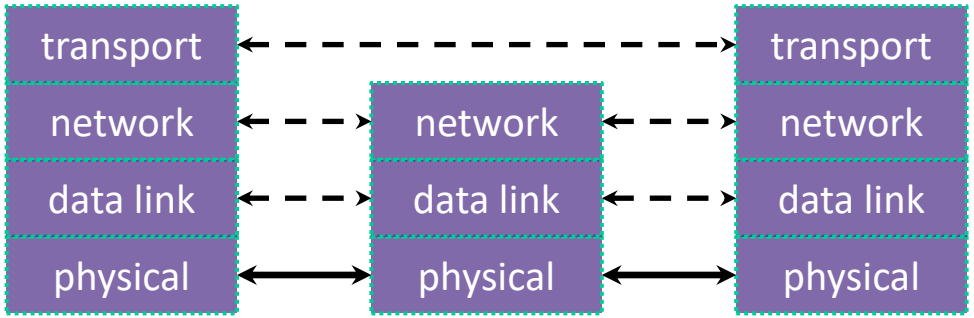




The Transport Layer (TCP)

- Applications use OS services to establish TCP streams:
  - The "Berkeley sockets" API
    - A set of OS system calls (Part of POSIX on linux)
  - 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

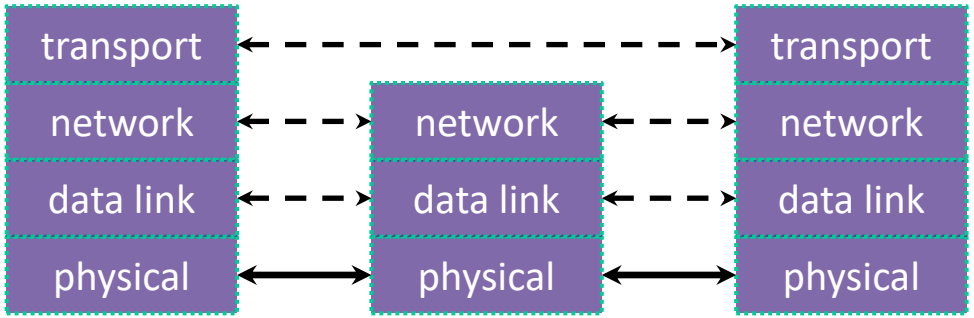
Used same as in File I/O



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

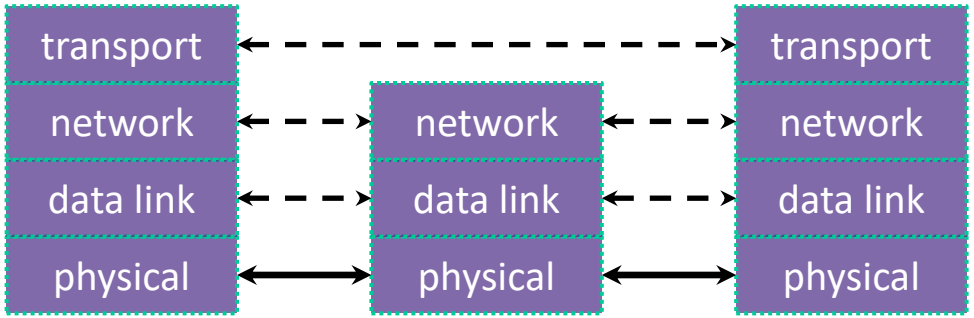
Ok when we want speed. (VOIP or ZOOM)



The Transport Layer

TCP:

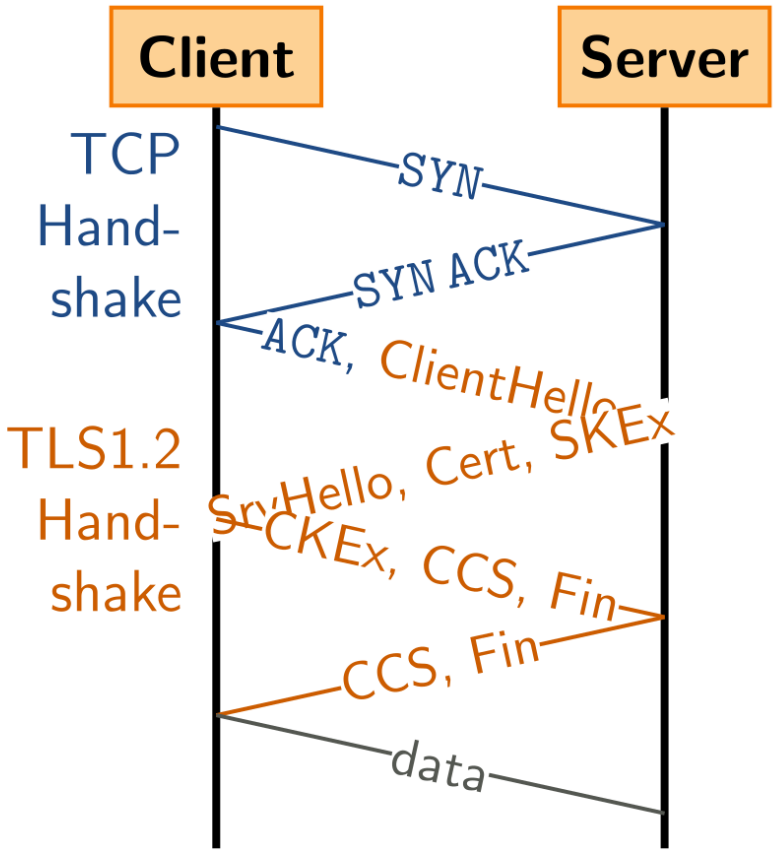
UDP:



In reality, TCP goes back and forth 3 times before to make sure:

"Hey, do you want to share popcorn and are you who I think you are"

TCP Overhead

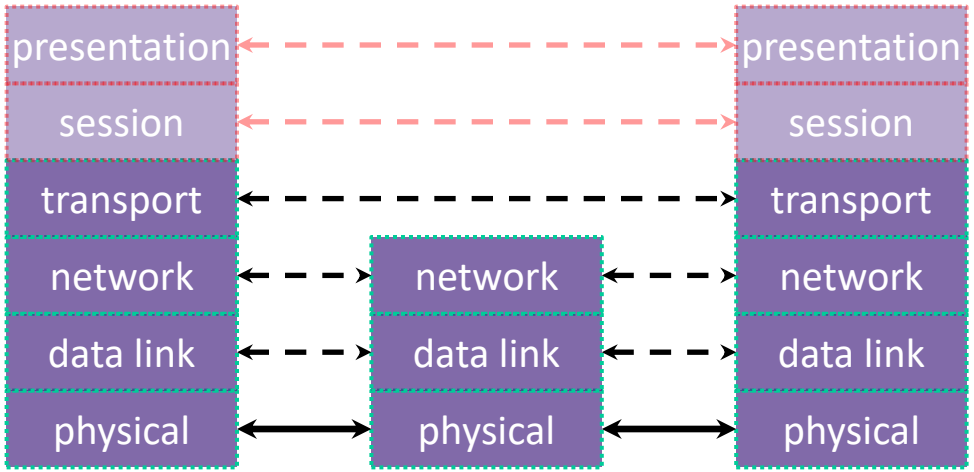


- Setting up a TCP connection typically requires **3 round trips**
  - (which is a relatively long time)
- If a packet in a sequence is dropped, then the "Stream" must wait to recover that packet before it can process other things in the stream.
- Solution: QUIC

By Sedrubal - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=114587250

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

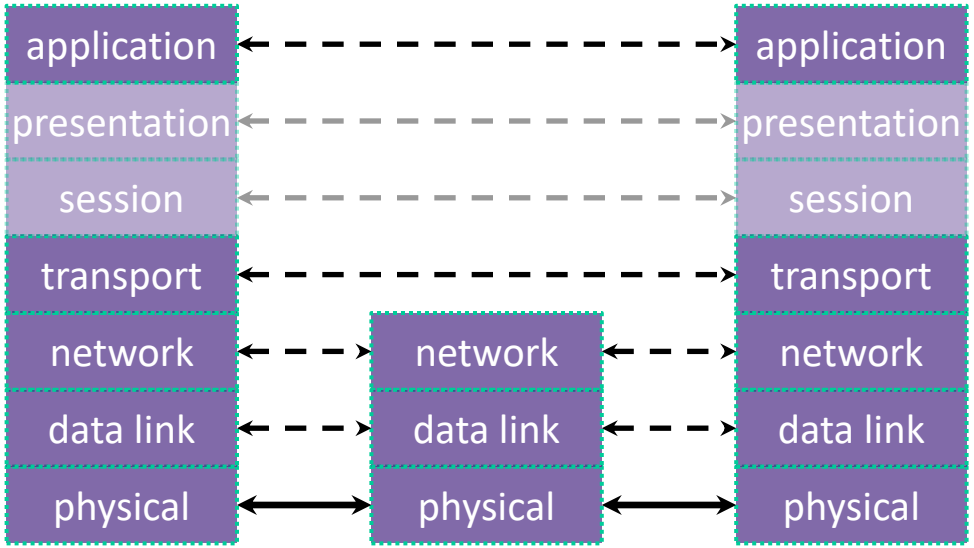


pollev.com/tqm

- Can we guarantee that data gets sent reliably from one computer to another?
  - If so, how?

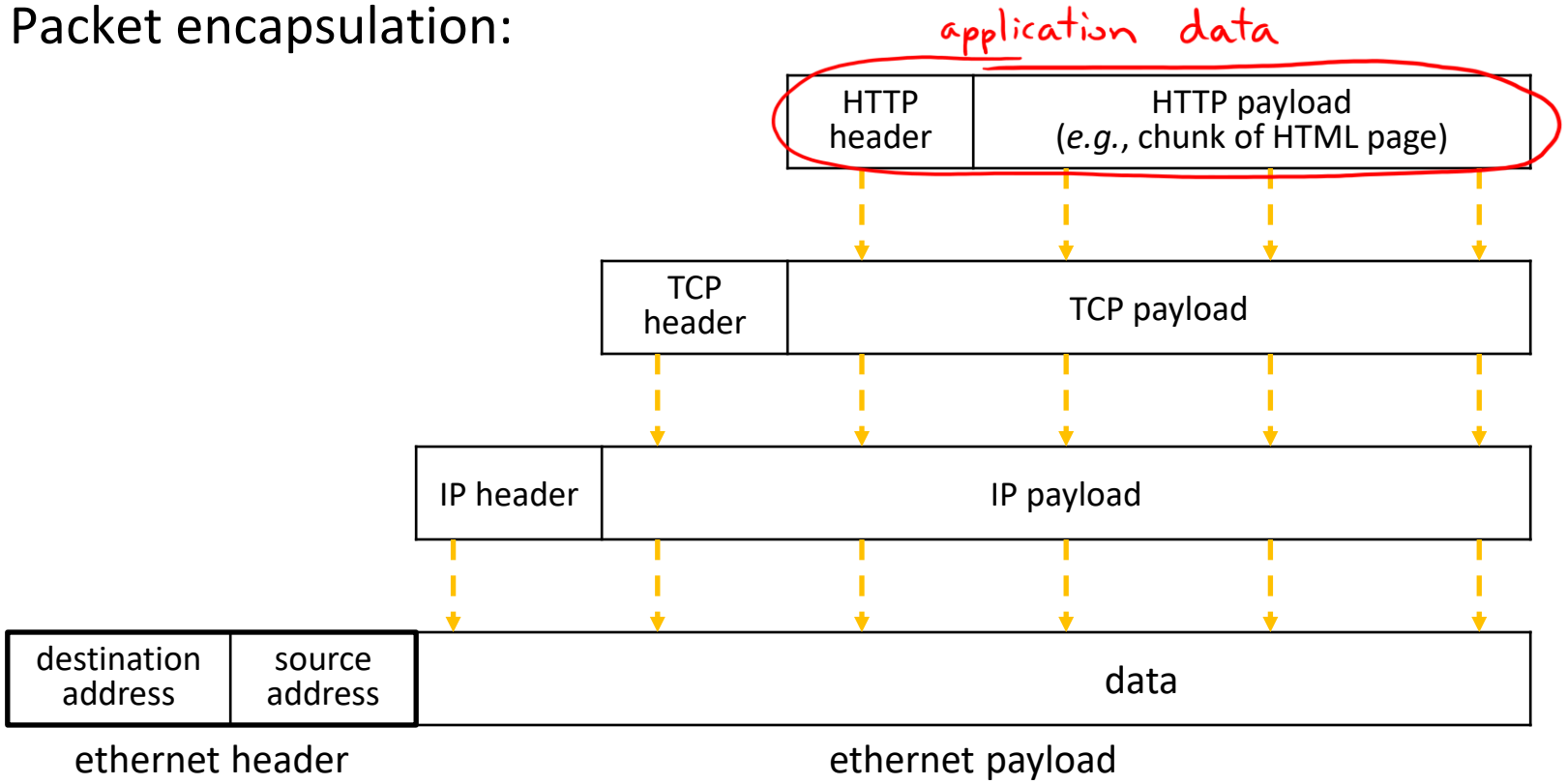
The Application Layer

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



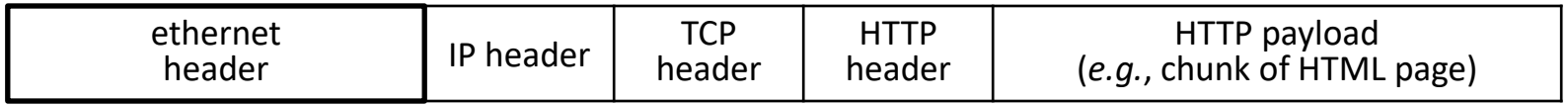
The Application Layer

- 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`

Next Lecture

- Socket Programming!