Exam 1 Review

CS 326E Elements of Networking

Mikyung Han

mhan@cs.utexas.edu

Time & Place

- Feb 27 Thursday 6-9 PM @ GDC 2.216 (lecture hall)
- Bring your *cheatsheet*, pen, and eraser
- No calculator allowed
- No electronic devices allowed
- Scratch paper provided

Cheatsheet

- You can bring 1 double-sided 8.5x11 paper
- Type or handwritten

Exam Format

- Multiple choice, T/F, Select All, Short answers
- Fill in the blanks
- Calculations (no need to bring calculator)
- Explain why ABC
- Design your own protocol that does XYZ
- Write code (socket programming)
  - TCP socket programming
  - Syntax will be given

Practice Exam Questions

- EX0 and EX1 are your friends
- In-class activities
- DNS Dig questions
- Discussions during lecture
- Socket programming EX and Project

How to study for Exam 1

- Review EX0, EX1, and In-class activities
- Review Wireshark Lab and Hands-on DNS Dig
- Review Socket programming EX and Project 1
- Go over slides
- Review lecture recordings
- Review textbook

Topics

Intro and Network Performance

- **Layers and protocol**
  - What and why layer
  - What is protocol

- **4 types of delays**
  - What are they?

- **Packet switching vs circuit switching**
  - In-class exercise

| Example Protocols	Layers	Responsible for
FTP, HTTP, SMTP	**Application**	application specific needs
TCP, UDP	**Transport**	process to process data transfer
IP	**Network**	host to host data transfer across different network
Ethernet, WiFi	**Link**	data transfer between physically adjacent nodes
802.3 PHY	**Physical**	bit-by-bit or symbol-by-symbol delivery

Protocols and layers are two building blocks in network communication

Protocols provides ways
for peers to communicate horizontally
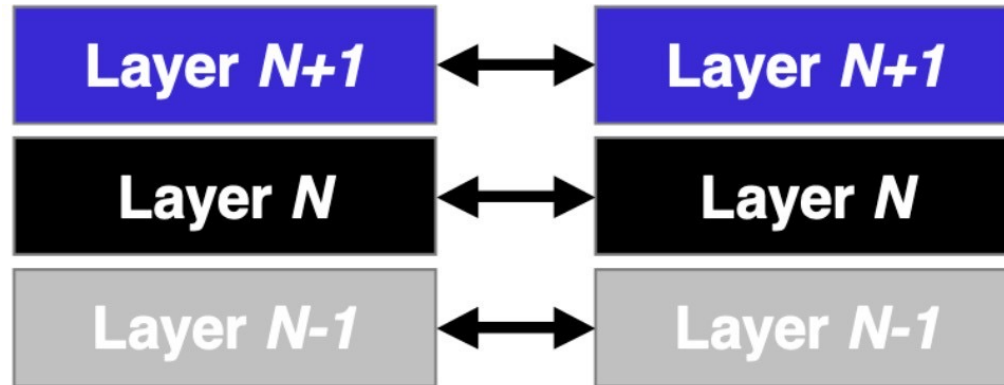
Protocol in layer N only interacts with peers in the same layer N

Packet delay: four sources
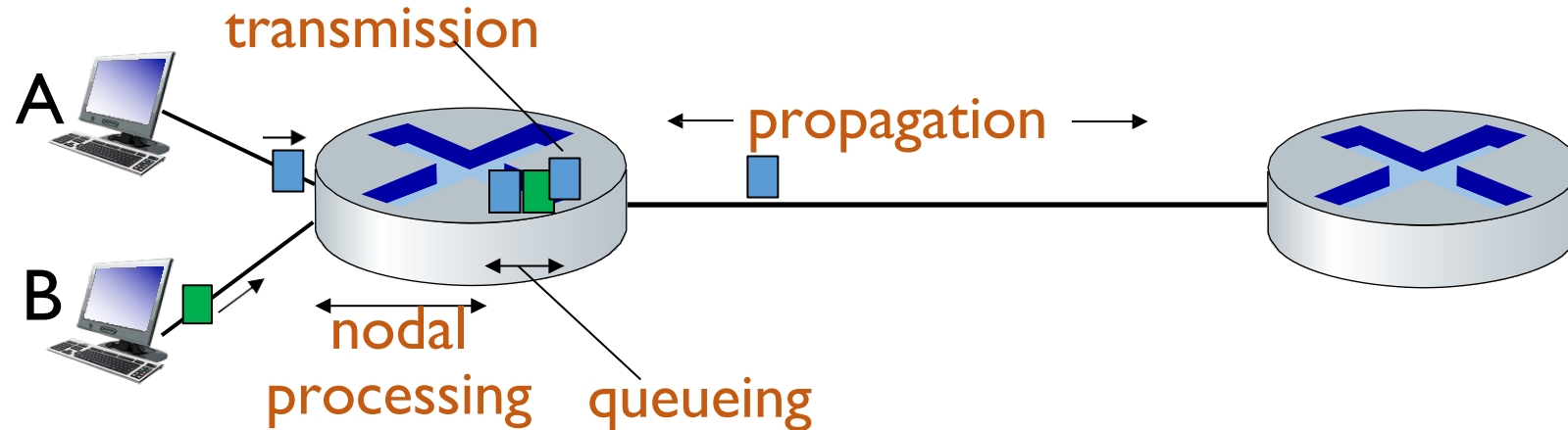
$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

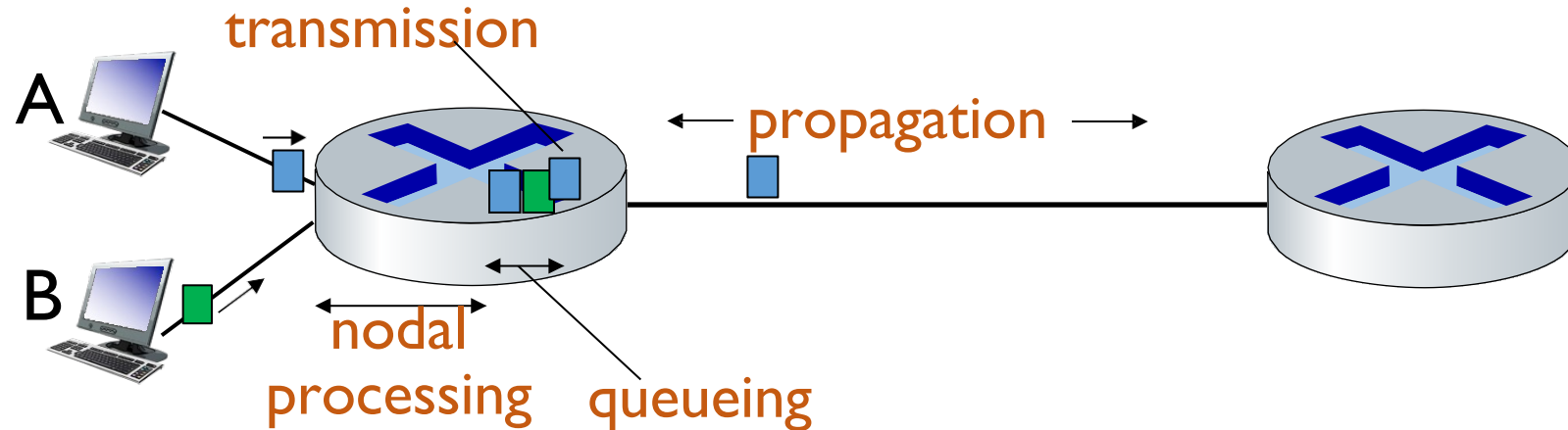d_{proc}: nodal processing
- check bit errors
- determine output link
- typically < microsecs

d_{queue}: queueing delay
- time waiting at output link for transmission
- depends on congestion level of router

Packet delay: four sources

$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

d_{trans}: transmission delay:
- L: packet length (bits)
- R: link transmission rate (bps)
- $d_{trans} = L/R$

d_{prop}: propagation delay:
- d: length of physical link
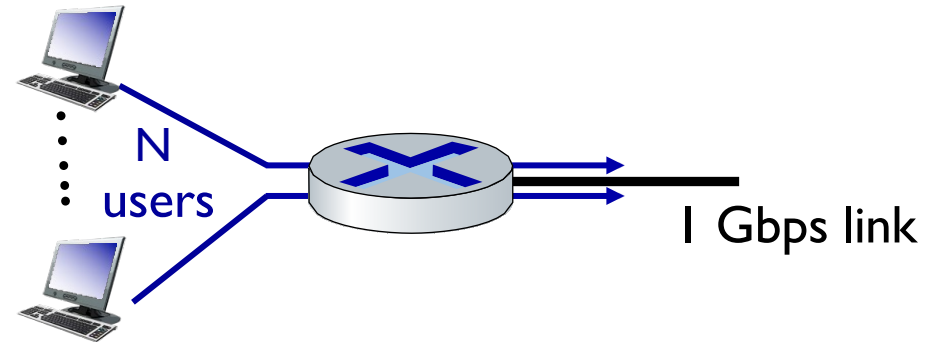- s: propagation speed (~2×10^8 m/sec)
- $d_{prop} = d/s$

d_{trans} and d_{prop}
very different

Packet switching versus circuit switching

example:
- 1 Gb/s link
- each user:
  - 100 Mb/s when "active"
  - active 10% of time (happens randomly)

N users

1 Gbps link

Q: What is the max number of users that can share this network?

- packet switching: ???

A: Need some assumption on link availability guarantee.

Say we guarantee 99.99% link availability for each user.
Failure rate < 0.0001 (When does failure happen?)

What is the number of users such that the probability of more than 10 users being active simultaneously < 0.0001?

Application Layer

Example Protocols

Responsible for

FTP, HTTP, SMTP

Application

application specific needs

TCP, UDP

Transport

process to process data transfer

IP

Network

host to host data transfer across different network

Ethernet, WiFi

Link

data transfer between physically adjacent nodes

802.3 PHY

Physical

bit-by-bit or symbol-by-symbol delivery

14
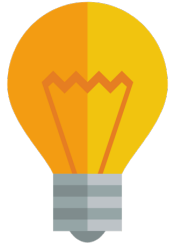
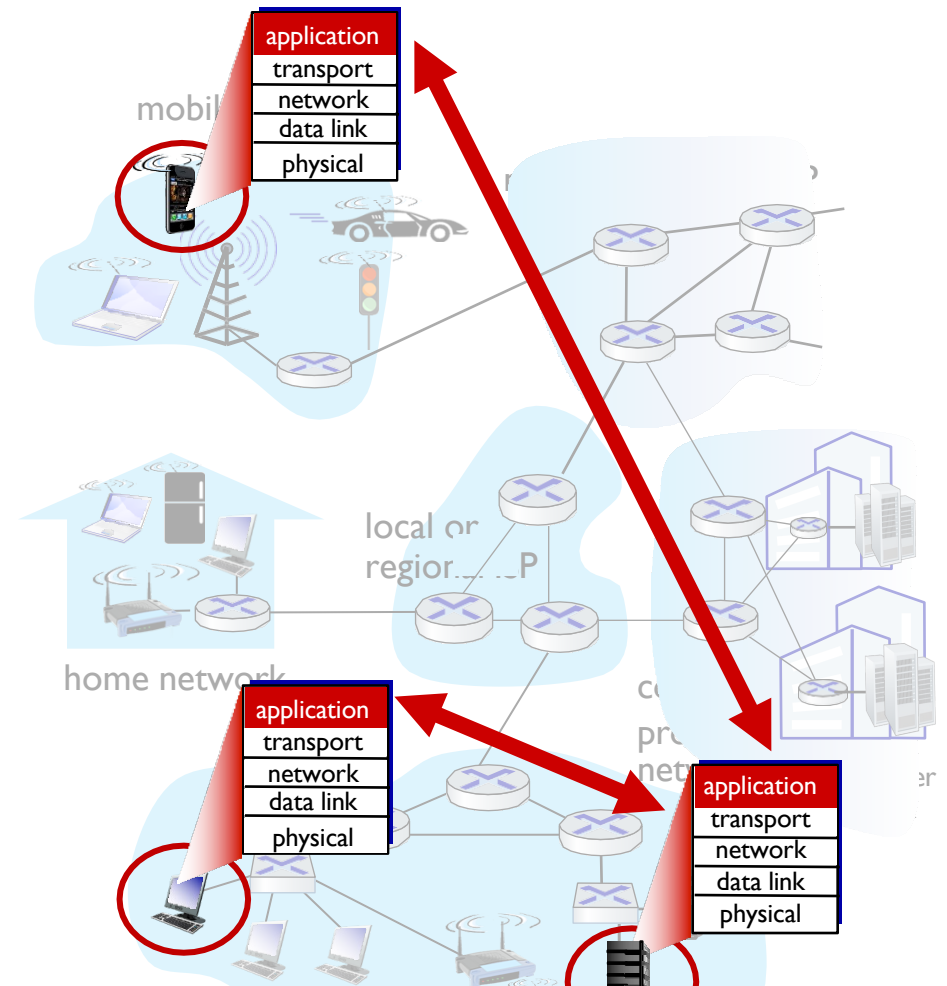# Design

- End to End argument
  - What is it and why they decided this way?
- Stateless protocol vs stateful protocol
- Connection-oriented protocol vs connectionless protocol

Applications only run on the endpoints!

■ Network core devices do NOT run user applications
- No code to write for these ©

■ When developing an app, we only need to consider the two ends
- server/client or peers

This allowed rapid app development and propagation

DNS

- Distributed hierarchical design
- DNS dig
  - Know how to read results
  - Dig's recursive query vs non-recursive query
- Root DNS servers/TLD servers/Authoritative servers
- Caching
- What is the underlying protocol?
  - Why?

HTTP

- Difference between HTTP/1.0, HTTP/1.1 and HTTP/2.0
- Persistent connection vs non-persistent
- Head of line blocking
- Framing
- Framing with priority
- What is the underlying protocol?
  - Why?

HTTP 1.0: Non-persistent HTTP
takes 2 RTT + object transmission time per object!

RTT (definition): Round trip time between client and server
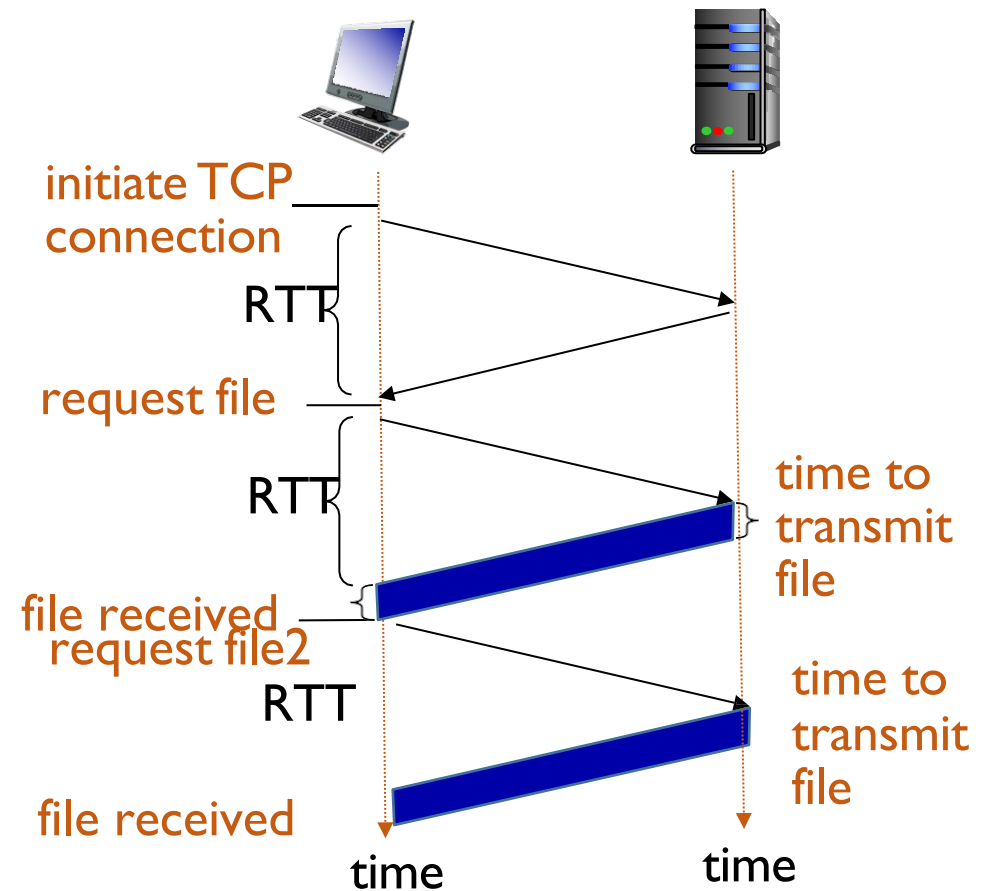
HTTP response time (per object):

- one RTT to initiate TCP connection
- one RTT for HTTP request and first few bytes of HTTP response to return
- object/file transmission time

initiate TCP connection

RTT

request file

RTT

time to transmit file

file received
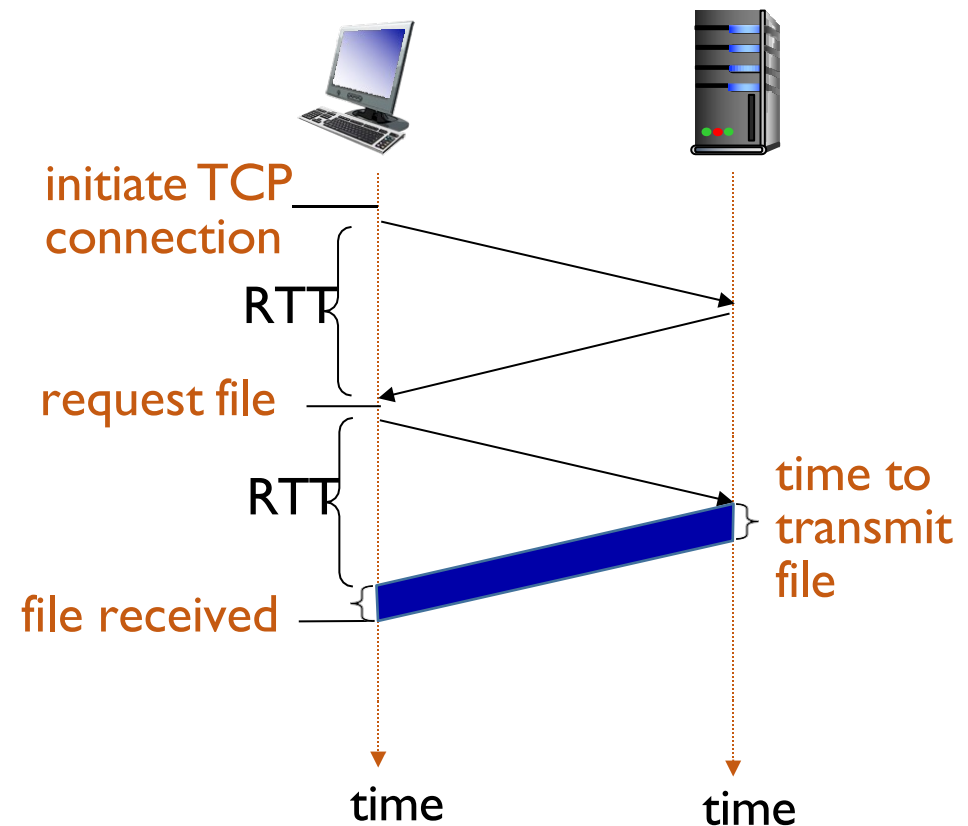
time

time

This was HTTP 1.0

HTTP 1.1: Persistent HTTP reuses the same TCP connection over multiple objects

Initial object 2 RTT
+ subsequent objects 1 RTT
+ all objects TX time

initiate TCP connection

RTT

request file

RTT

time to transmit file

file received
request file2

RTT

time to transmit file

file received

time   time

HTTP/2 mitigates HOL blocking

HTTP/2: objects divided into frames, frame transmission interleaved
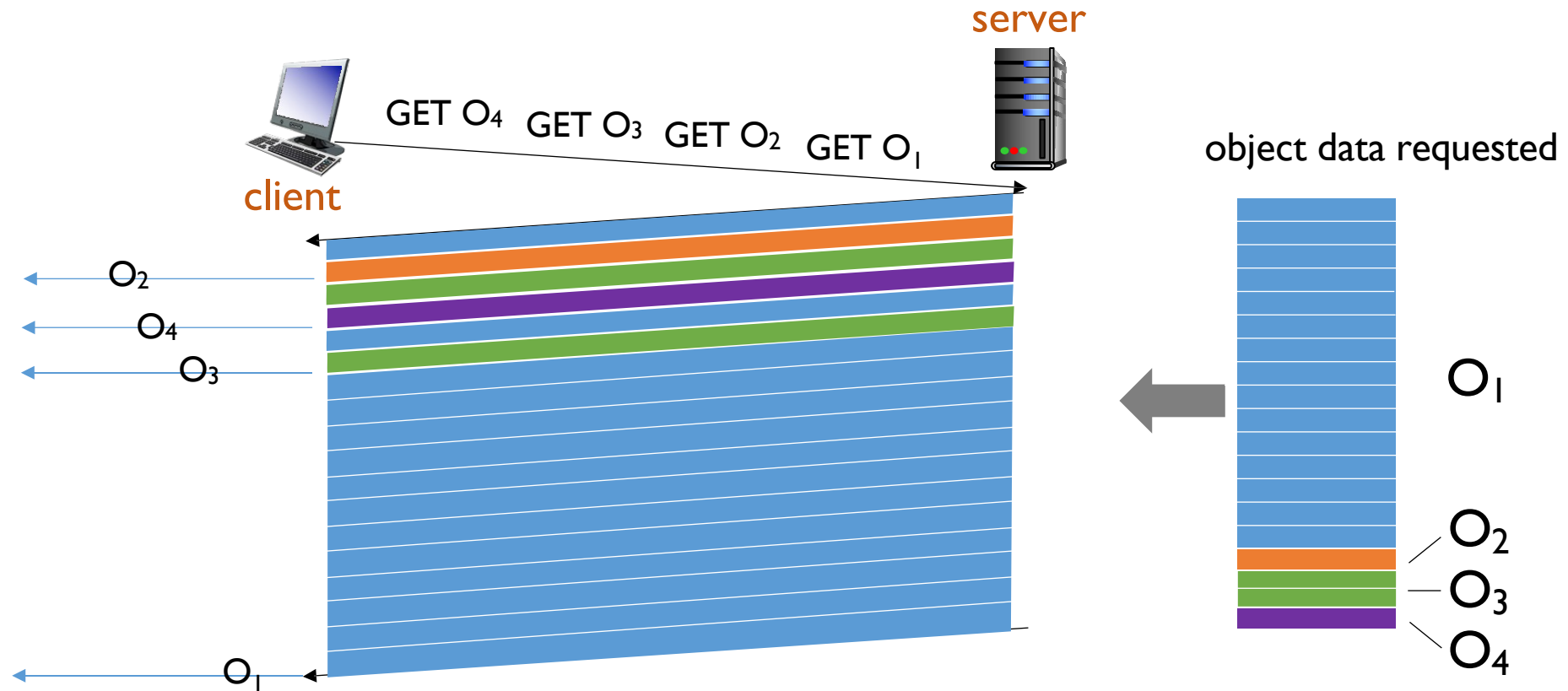
O_2, O_3, O_4 delivered quickly, O_1 slightly delayed

Transport Layer

Example Protocols
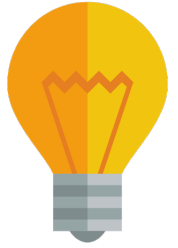
Responsible for

FTP, HTTP, SMTP

Application

application specific needs

TCP, UDP

Transport

process to process data transfer

IP

Network

host to host data transfer across different network

Ethernet, WiFi

Link

data transfer between physically adjacent nodes

802.3 PHY

Physical

bit-by-bit or symbol-by-symbol delivery

23

Transport Layer

- Socket and port

- TCP vs UDP

- Multiplexing and demultiplexing

- RDT

- How TCP works

- Socket programming

What is a Socket?

- process sends/receives messages to/from its socket
- socket analogous to a "door"
  - sending process shoves message out the door
  - sending process relies on transport layer to deliver message to socket at receiving process
  - two sockets involved: one on each end
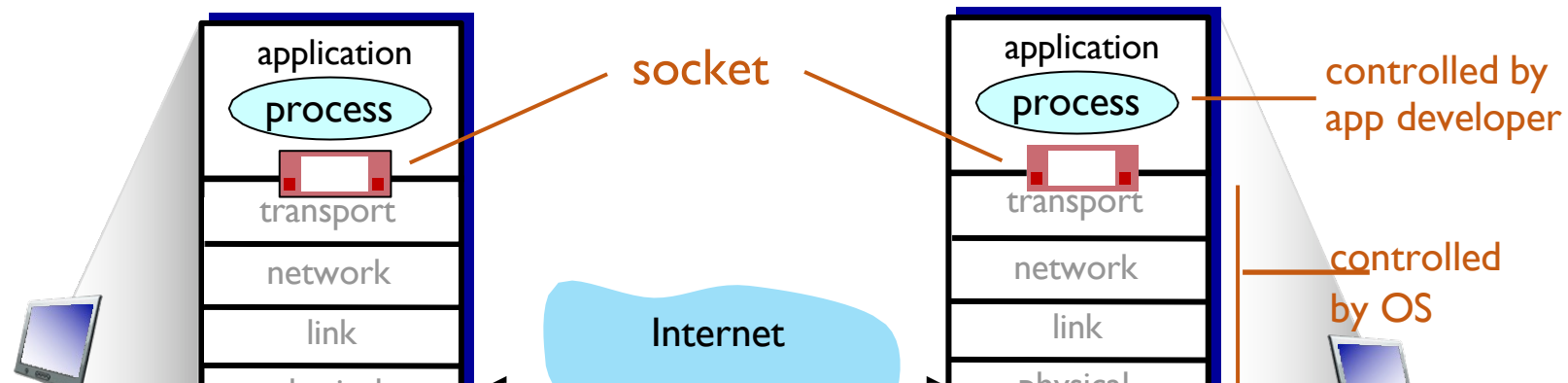
Socket programming question

TCP vs UDP Features

- TCP: Transmission Control Protocol
  - reliable, in-order delivery
  - congestion control
  - flow control
  - Connection-setup
- UDP: User Datagram Protocol
  - unreliable, unordered delivery
  - no-frills extension of "best-effort" IP
- Both has NO guarantee on delay or bandwidth
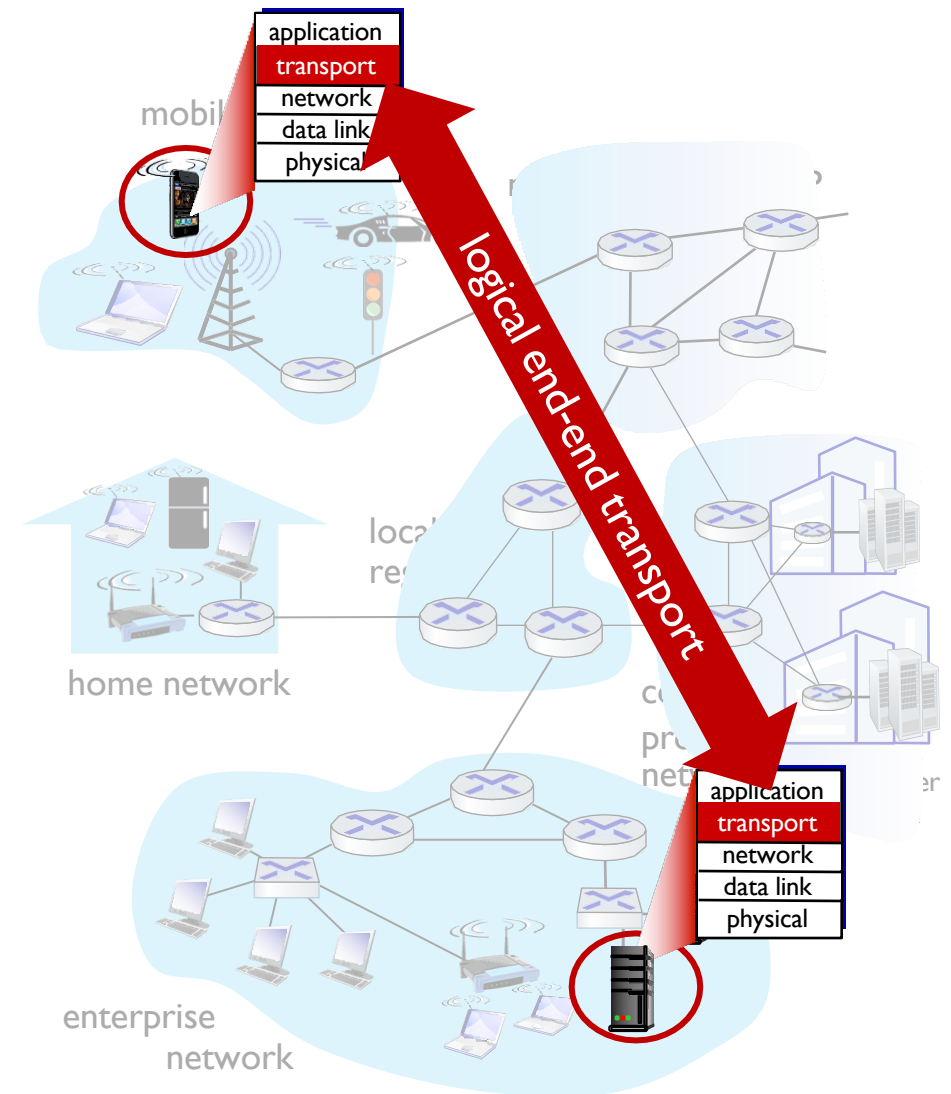
RDT discussions/design choices

- ACK/NAK vs ACK only
- Sequence numbers
- Retransmissions
- Timeout
- Stop and wait

How TCP works

- Sequence number/Ack number
- Retransmissions
- Timeout
- Triple dupe ACK
- TCP flow control
- TCP connection management

Socket programming with multi-threading

- TCP socket program
- Thread
  - Worker thread vs main thread