CS2105

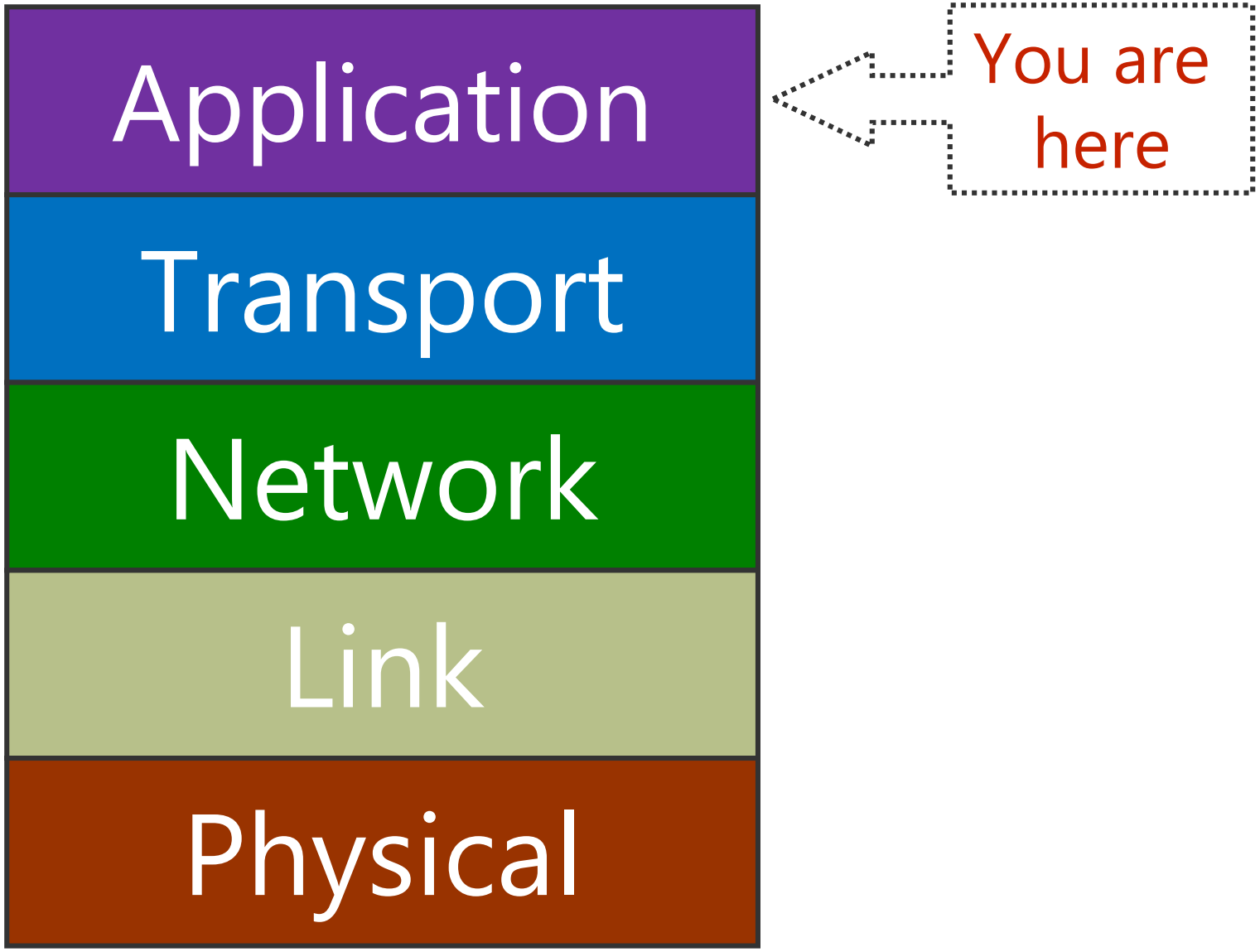
An *Awesome* Introduction to Computer Networks

Lecture 2 discussion

Department of Computer Science
School of Computing

Network Protocols

- The Internet supports various kinds of network applications:
  - Web, VoIP, email, games, e-commerce, social nets, …
- Network applications exchange messages and communicate among peers according to protocols.
  - A protocol defines format and order of messages exchanged and the actions taken after messages are sent or received.


Application
Transport
Network
Link
Physical
You are here

Lectures 2&3: Roadmap

2.1 Principles of Network Applications

2.2 Web and HTTP

2.4 DNS

2.7 Socket programming

} To discuss next week

Kurose Textbook, Chapter 2
(Some slides are taken from the book)

What transport service does an app need?

Data integrity

- some apps (e.g., file transfer, web transactions) require 100% reliable data transfer
- other apps (e.g., audio streaming) can tolerate some data loss

Timing

- some apps (e.g., online interactive games) require low delay to be “effective”

Throughput

- some apps (e.g., multimedia) require minimum amount of bandwidth to be “effective”
- other apps (e.g., file transfer) make use of whatever throughput available

Security

- encryption, data integrity, authentication …

Lectures 2&3: Roadmap

2.1 Principles of Network Applications

2.2 Web and HTTP

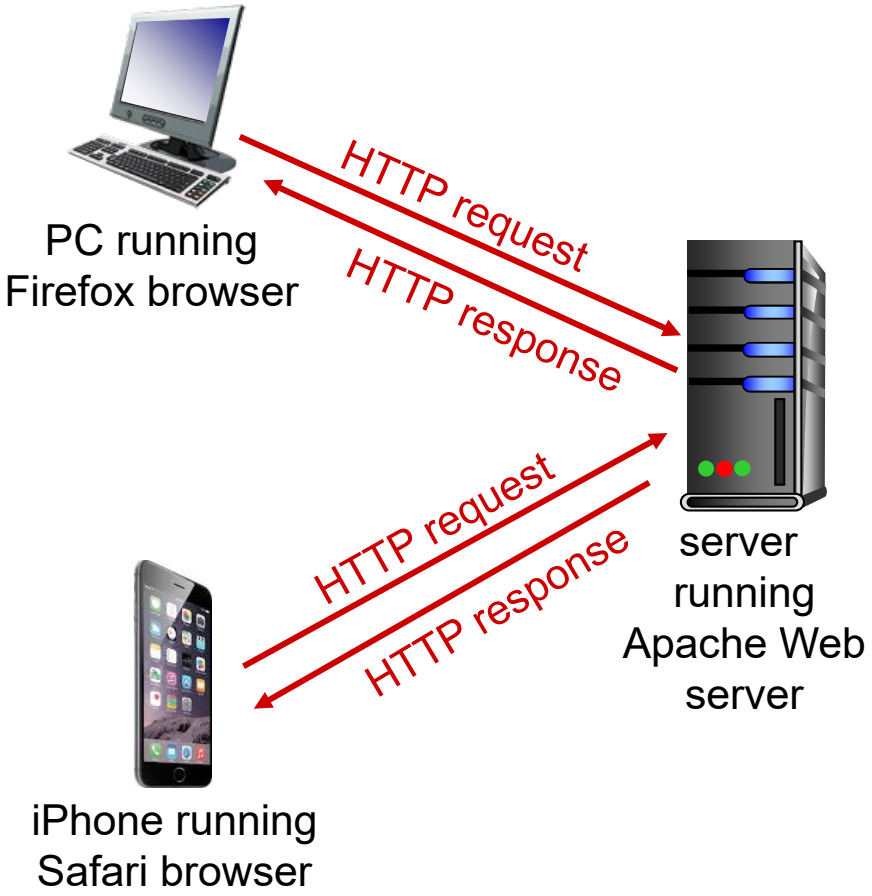
2.4 DNS

2.7 Socket programming

HTTP Overview

HTTP: Hypertext transfer protocol

- Web's application layer protocol
- Client/server model
  - *client:* usually is browser that requests, receives and displays Web objects
  - *server:* Web server sends objects in response to requests
- http 1.0: RFC 1945
- http 1.1: RFC 2616



HTTP Over TCP

HTTP uses TCP as transport service

- Client initiates TCP connection to server.
- Server accepts TCP connection request from client.
- HTTP messages are exchanged between browser (HTTP client) and Web server (HTTP server) over TCP connection.
- TCP connection closed.

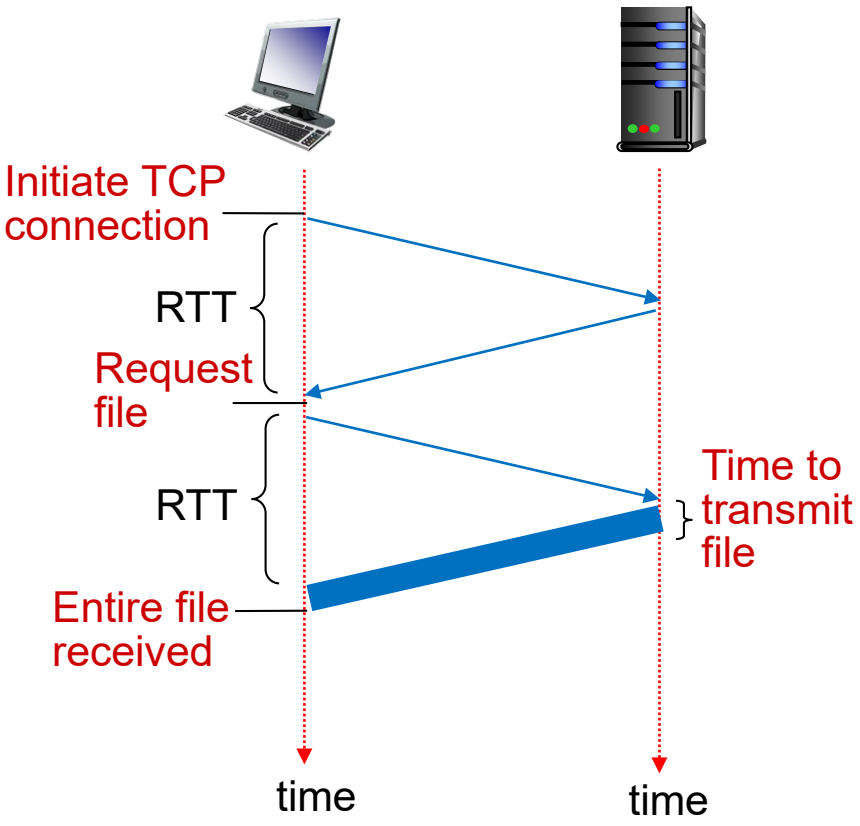
Non-persistent HTTP: Response Time

RTT: time for a packet to travel from client to server and go back

HTTP response time:

- one RTT to establish TCP connection
- one RTT for HTTP request and the first few bytes of HTTP response to return
- file transmission time
- non-persistent HTTP response time =

  2 * RTT+ file transmission time



Two Versions of HTTP

non-persistent HTTP issues:

- requires 2 RTTs per object
- OS overhead for *each* TCP connection
- browsers often open parallel TCP connections to fetch referenced objects

persistent HTTP:

- server leaves connection open after sending response
- subsequent HTTP messages between same client/server sent over the same TCP connection
- moreover, client may send requests as soon as it encounters a referenced object (persistent with pipelining)
  - as little as one RTT for all the referenced objects

Q1

A Web server stores a webpage that comprises a base HTML file and 2 images referenced by the base HTML file. The HTML file is 10 bytes and each image is 20 bytes. A client is connected to the Web server through a direct link of 100 bps. RTT is 100 milliseconds.

Suppose HTTP header is of negligible size. How long does the client take to download the entire webpage, assuming non-persistent and non-parallel HTTP is engaged?

Q2

A Web server stores a webpage that comprises a base HTML file and 2 images referenced by the base HTML file. The HTML file is 10 bytes and each image is 20 bytes. A client is connected to the Web server through a direct link of 100 bps. RTT is 100 milliseconds.

Suppose HTTP header is of negligible size. How long does the client take to download the entire webpage, assuming persistent HTTP with pipelining is engaged?