Midterm

- When: Mar 16 (Tu) 2:05-3:15pm
- Where: Zoom meeting
  - Meeting link same as the class meeting link
  - camera on during the entire exam period
  - your exam will not be graded if you do not join the Zoom meeting
- Open-book and open-notes
  - You are NOT allowed to communicate with each other or search solutions online
- Office hours: Mar 15 (M) 2-3pm

Topics

- Overview (chapters 1-2)
- Interprocess communication (chapters 3-5)
- Time and global states (chapter 14)
- Coordination (chapter 15)

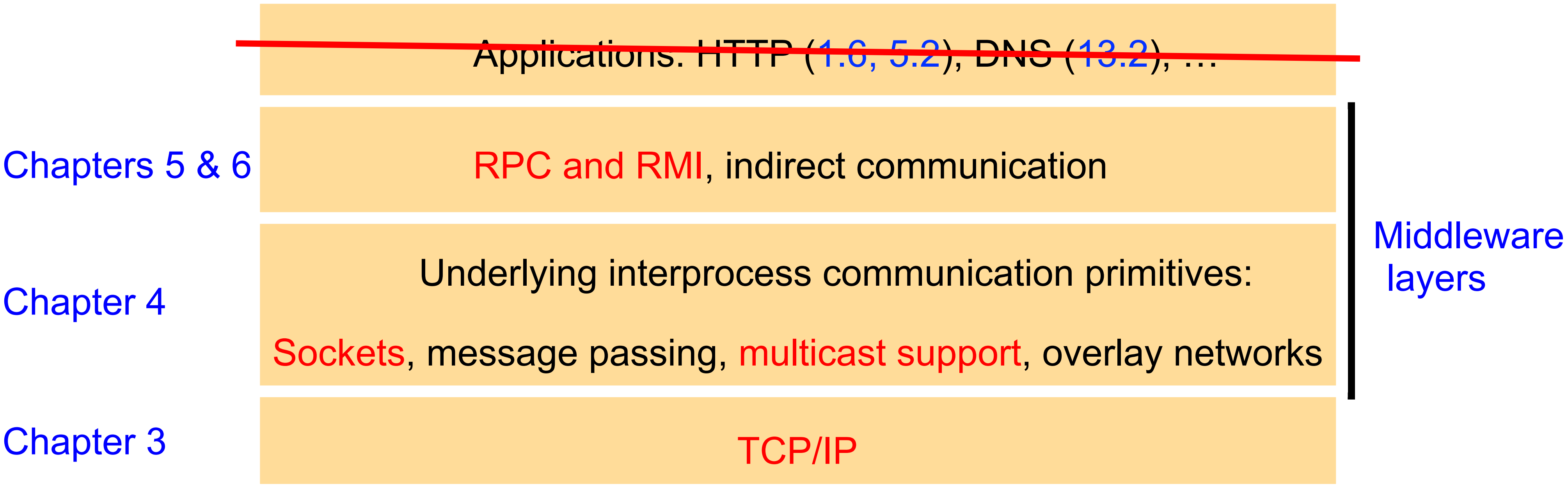
Overview

- What Is a Distributed System?
  - A network of processes: the processes interact with one other to achieve a goal
  - no global clock, fail independently, collective objective, …
- Examples of distributed Systems: Internet, P2P, Cloud, …
- Challenges: heterogeneity, scalability, transparency, …

Overview

- Architectual models
  - Communication entities: processes, objects, components, …
  - Communication paradigms: message passing, multicast, RPC, RMI, …
  - Client-Server vs. P2P
- Fundamental models
  - Synchronous vs. asynchronous systems
  - Common failure types: crash, omission, byzantine

Interprocess Communication



~~Applications: HTTP (1.6, 5.2), DNS (13.2), …~~

Chapters 5 & 6 — RPC and RMI, indirect communication

Chapter 4 — Underlying interprocess communication primitives:

Sockets, message passing, multicast support, overlay networks

Chapter 3 — TCP/IP

Middleware layers

Interprocess Communication

- Internet protocols
  - packet switching, four types of delays
  - IP: routing and forwarding, addressing
  - TCP/UDP: sockets, (de)multiplexing, reliable data transfer
- Socket programming and IP multicast
- RPC and RMI
  - marshalling and unmarshalling
  - call semantics of request-reply communication
  - Implementation of RPC and RMI

Time and global states

- Physical time
  - clock drift and skew
  - synchronization: Cristian’s algorithm, Berkeley algorithm (~~byzantine clocks~~)
- Logical time
  - modeling distributed systems, global states, happened-before model
  - Lamport logical clocks, vector clocks
- Global state
  - cuts, consistent cuts
  - Chandy and Lamport’s snapshot algorithm (FIFO channels, safety & liveness)

Mutual Exclusion

- Correctness: safety, liveness, fairness
- Performance: message complexity, client delay, synchronization delay
- A simple centralized solution, a ring-based solution
- Ricart-Agrawala algorithm, Maekawa’s algorithm

Leader Election

- Correctness: safety, liveness
- Performance: message complexity, turnaround time
- Chang-Roberts algorithm (asynchronous & no failure)
- Bully algorithm (synchronous & crash)

Group Communication

- Why multicast?
- Programming model
- Reliable multicast: integrality, validity, agreement
- Ordered multicast: FIFO ordering, causal ordering, total ordering