



Computer Networks

TCP Connection Management

Jianping Pan
Fall 2022

Why?

How?

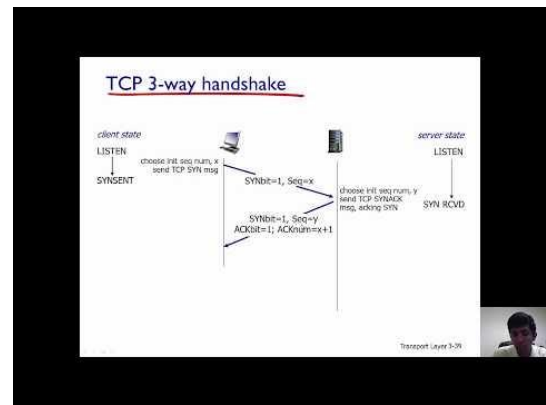
What?

Hierarchically

in a top-down way

for

lectures/tutorials/labs



10/11/22

CSc 361

1

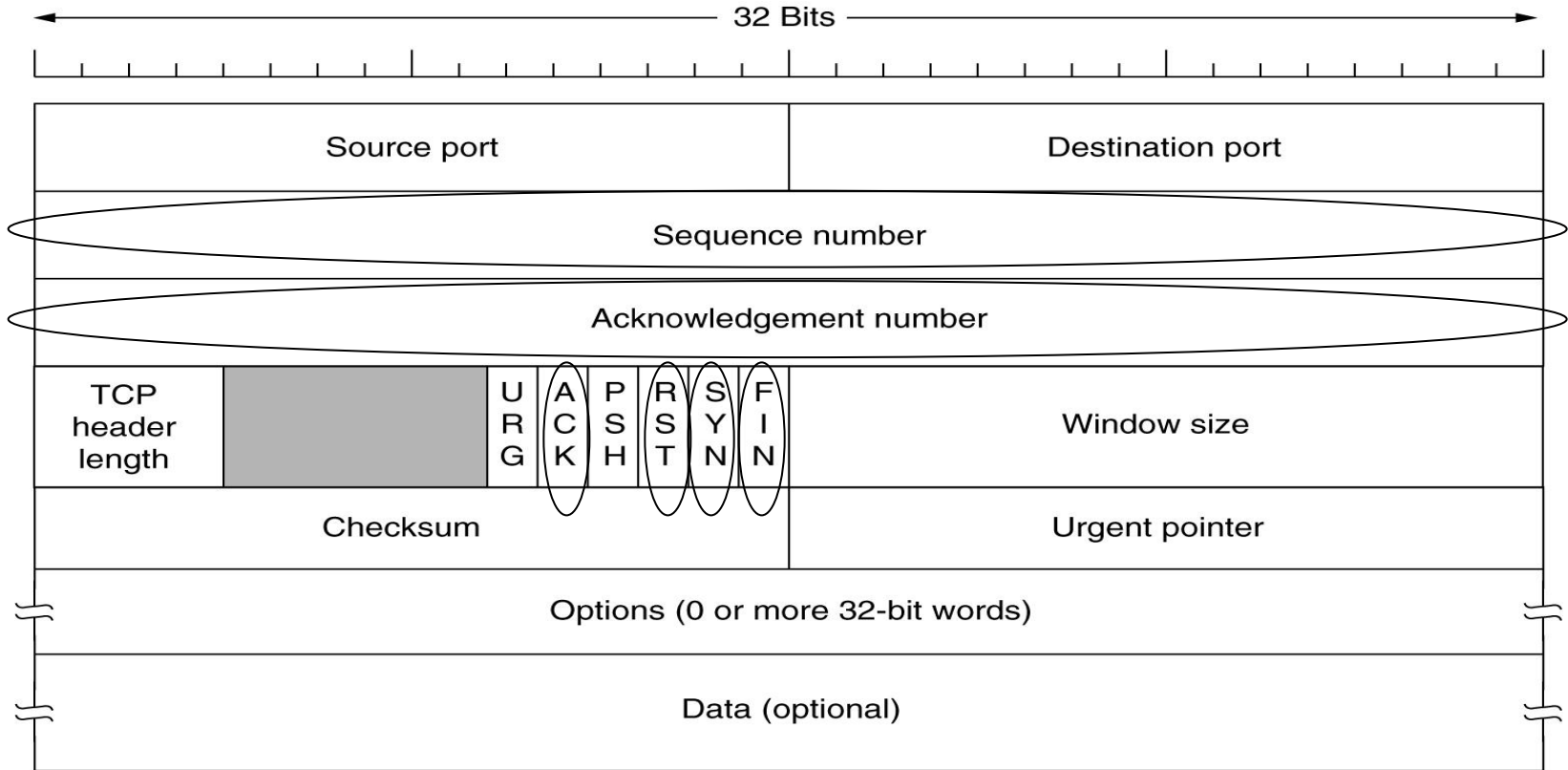
* the teaching team is NOT on discord; please use brightspace for Q&A and online participation; how about uvic teams?

Our course reps

- Thanks to our diverse student volunteers!
 - B01: Addie Finke <addiefinke>
 - B02: Preet Toor <preett23>
 - B03: Victor Kamel <vkamel>
 - B04: Nubia De La Torre <nubiadelatorre>
 - B05: Samantha Carter <abound>
 - B06: Serena Wollersheim <serenawollersheim>
- AAA: Aggregate, Amplify and Anonymize
 - we will e-meet them next week
 - we do welcome student feedback directly too

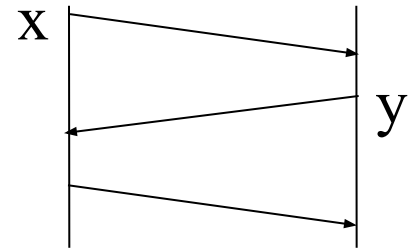


Last lecture: TCP packet header



Last lecture: Sequence number

- TCP sequence number (32-bit)
 - byte sequence for the **first** byte in payload
 - exception: SYN/FIN sequence number
 - random initial sequence number
 - exchanged during 3-way handshake
 - sequence number rollover
- Acknowledgment number (32-bit)
 - byte sequence for the **next** byte to expect



Last lecture: Control flags

- URG: urgent pointer meaningful
- ACK: acknowledgment number meaningful
- PSH: logic message boundary
- RST: connection reset
- SYN: synchronization (conn. establishment)
- FIN: finish (graceful connection release)
 - stay tuned: “TCP connection management”

Review: TCP

- Service provided by TCP
 - connection-oriented
 - reliable data transfer
- Service used by TCP
 - IP: connectionless, unreliable
- TCP packet header (*we will catch up too*)
- TCP protocol mechanisms
 - **connection management**
 - flow, error and congestion control

Today's topics

- Connection management
 - how to construct a **connection-oriented** service over **connectionless** services
 - TCP: connection-oriented
 - IP: connectionless
 - TCP connection management
 - connection **establishment**
 - three-way handshake
 - connection **release**
 - TCP connection **state machine**

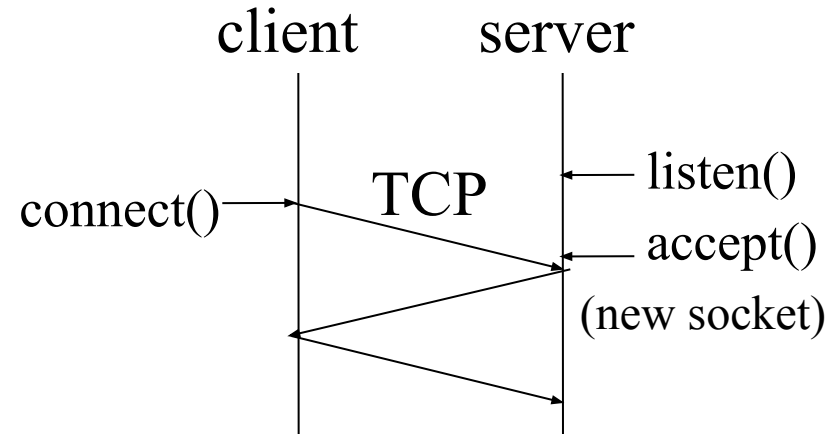
Socket interface

- Server
 - listen()

- Client
 - connect()

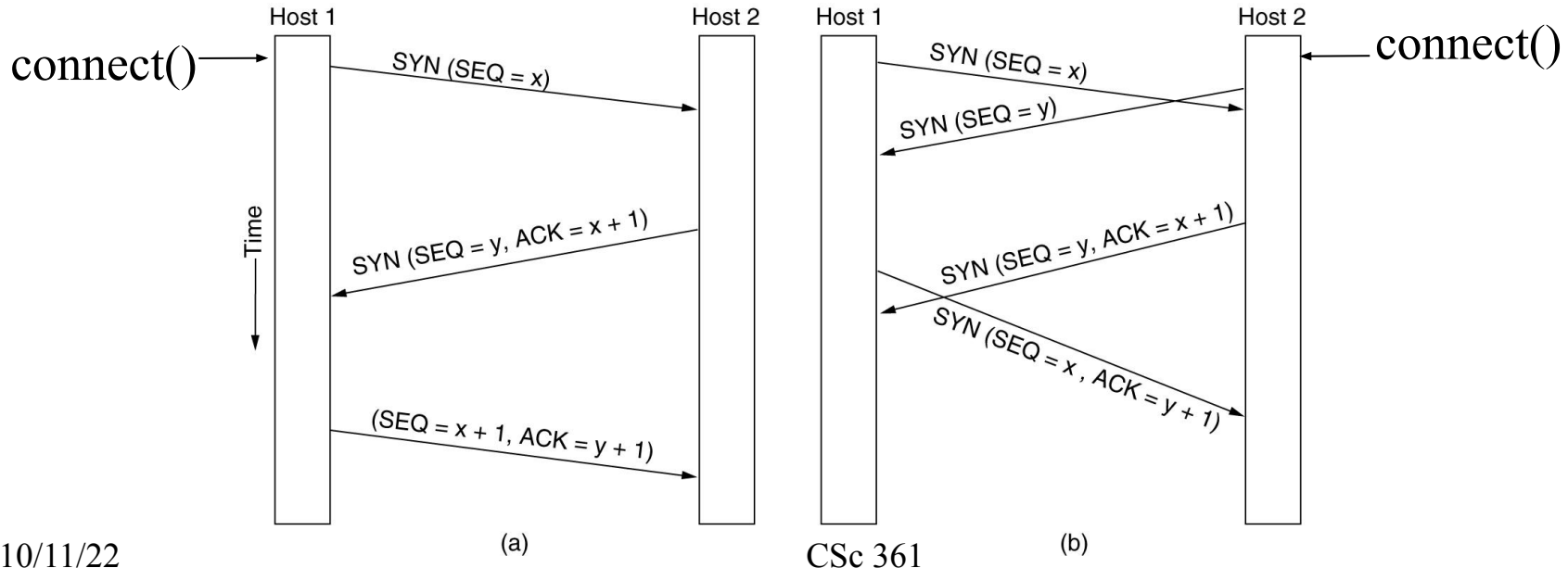
- Server
 - accept()

- The purpose of 3-way handshake
 - connection establishment
 - initial sequence number exchange
 - TCP option negotiation



Connection establishment

- Three-way handshake
 - and simultaneous establishment attempts



10/11/22

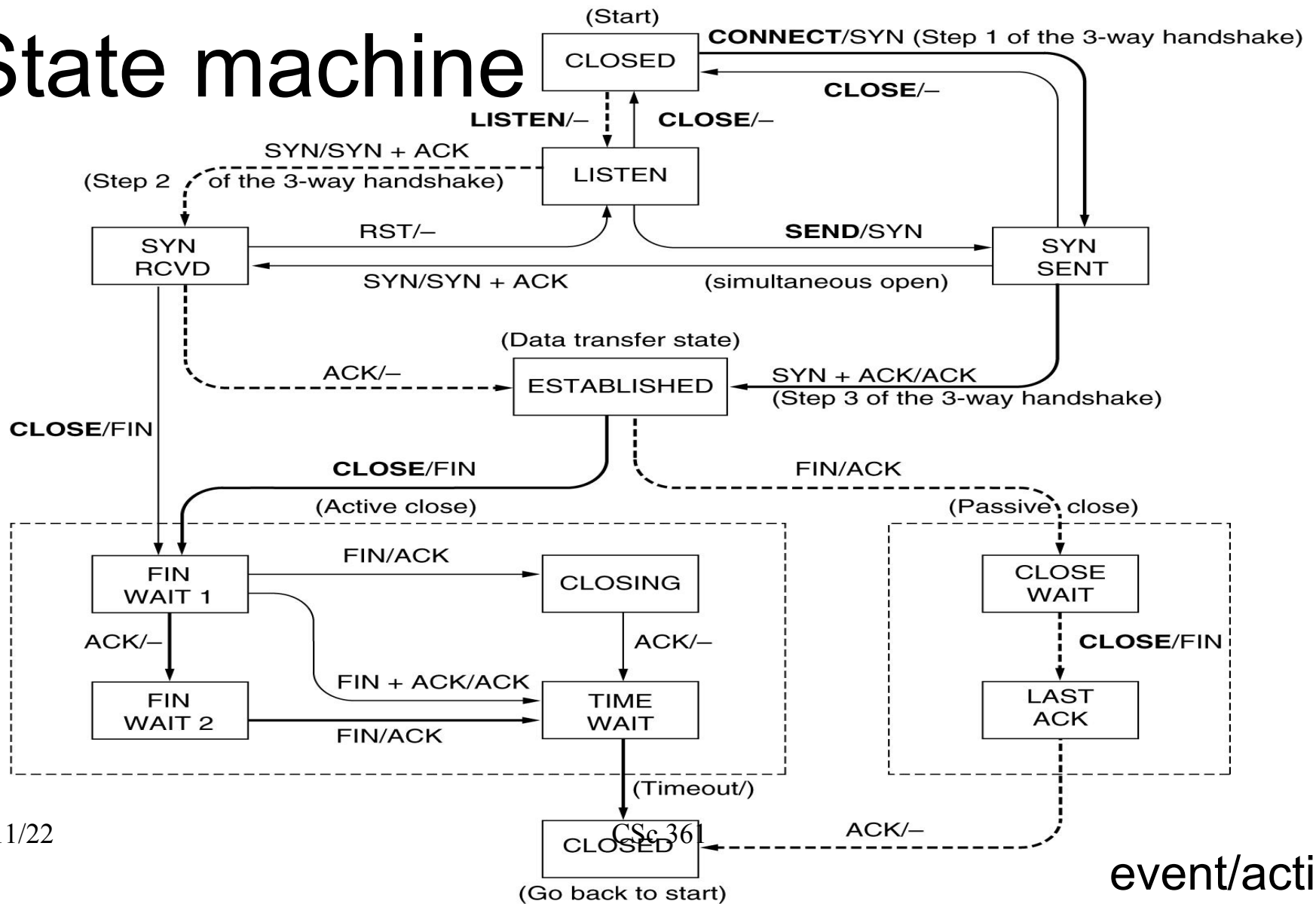
CSc 361

9

* TCP fast open?

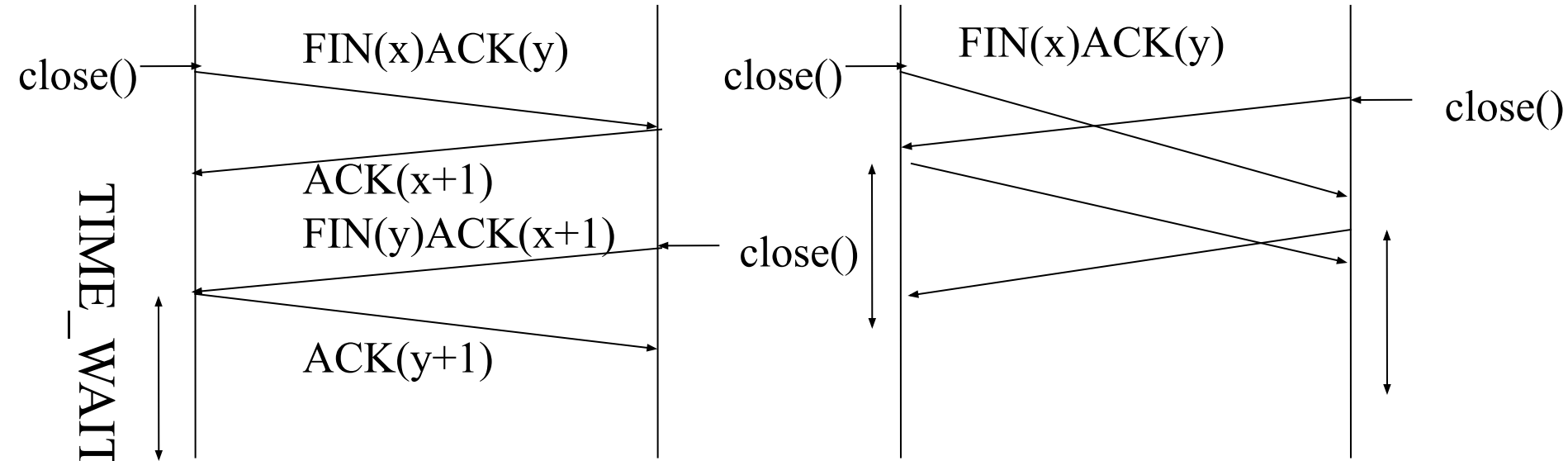
Q: non-ACK packet?

State machine



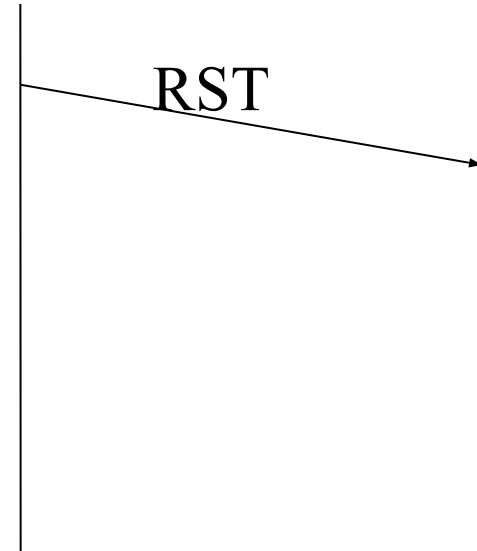
Connection release

- 2-way handshake in each direction
 - and simultaneous release attempts

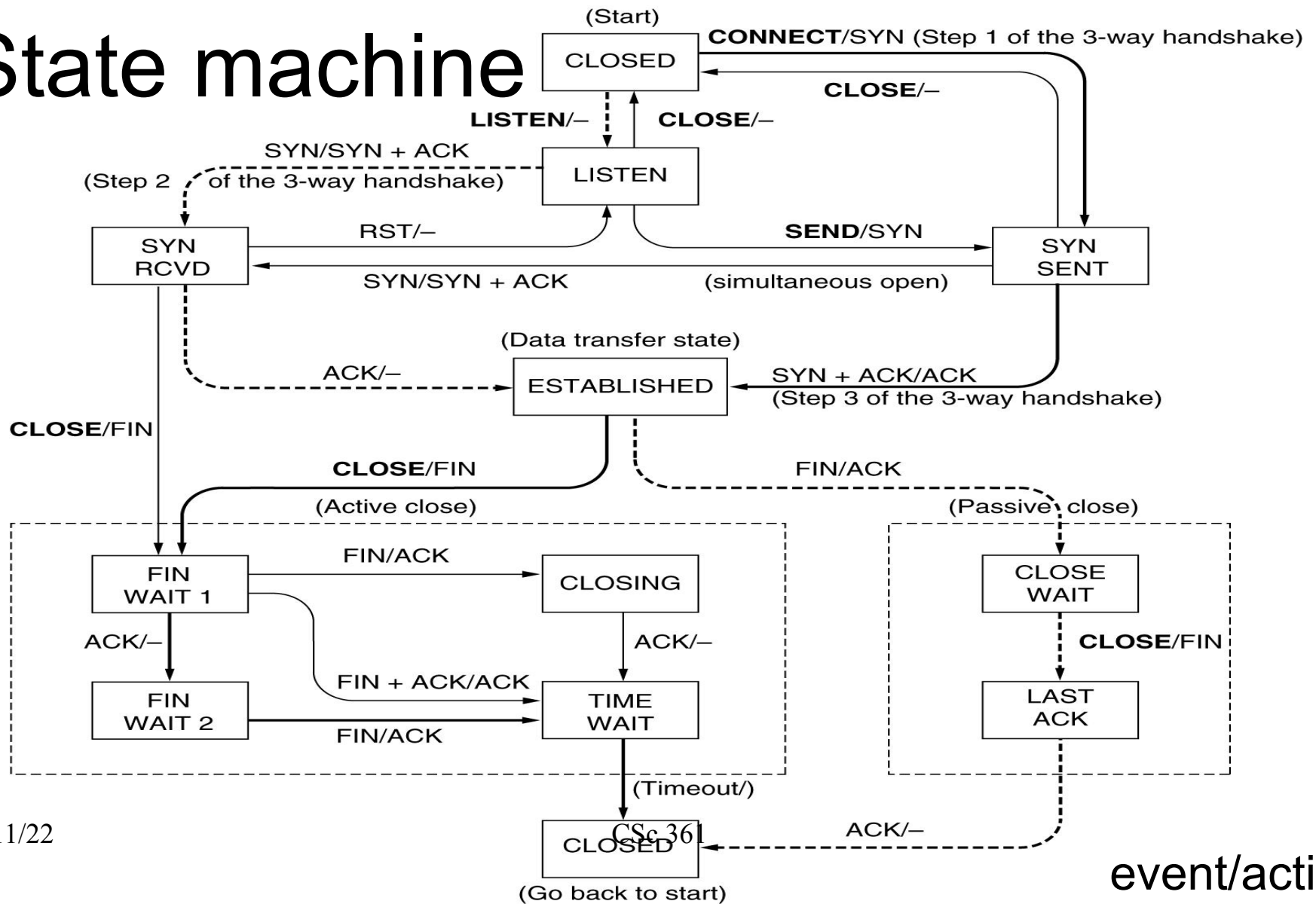


Connection reset

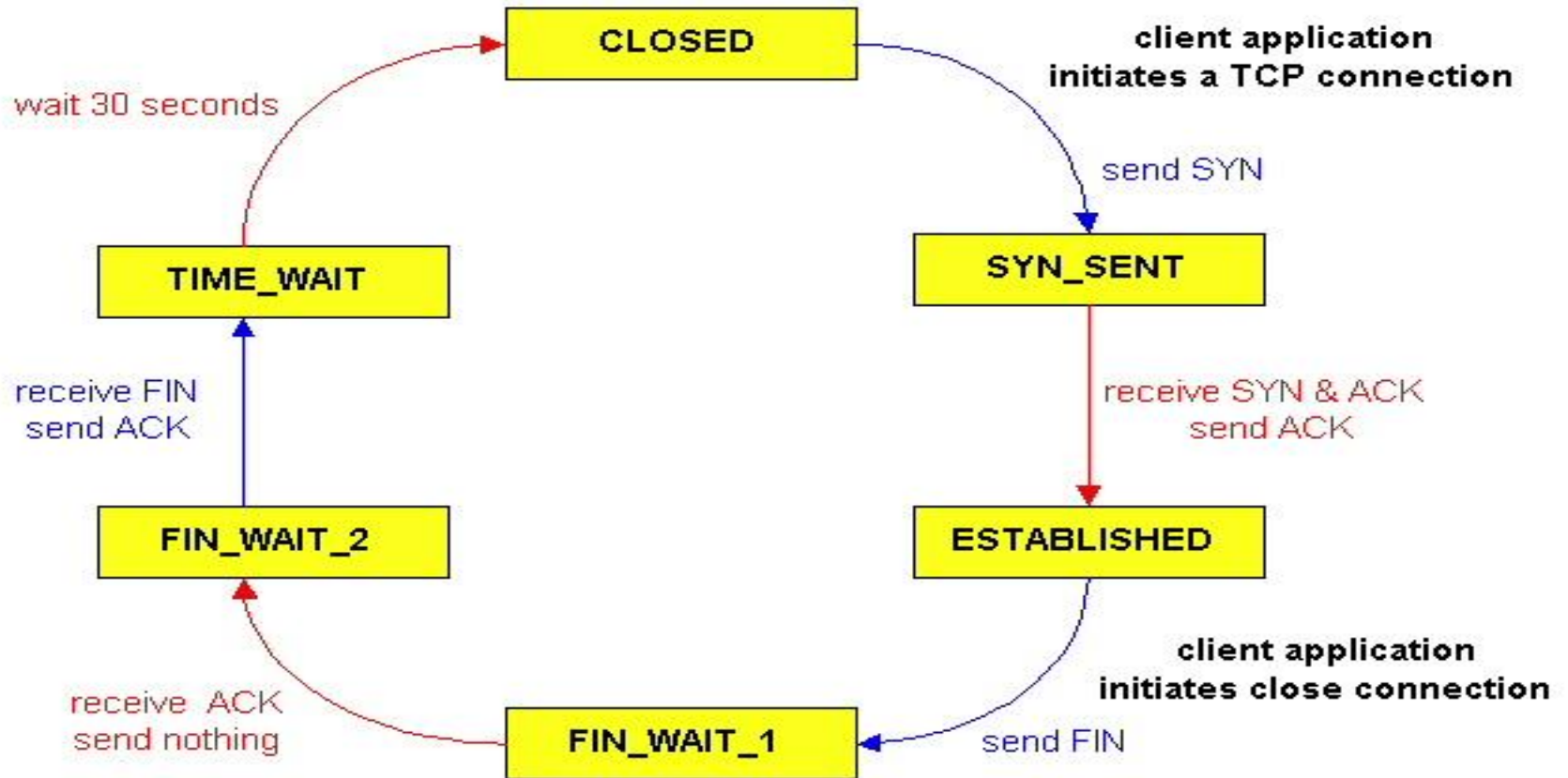
- Indicate protocol error
 - go back to CLOSED or LISTEN state
- Reset triggers
 - any packets received in CLOSED
 - except receiving a RESET
 - unacceptable ACK received
 - e.g., acknowledging seqno not used yet
 - unacceptable segments received
 - e.g., seqno above receiver's window



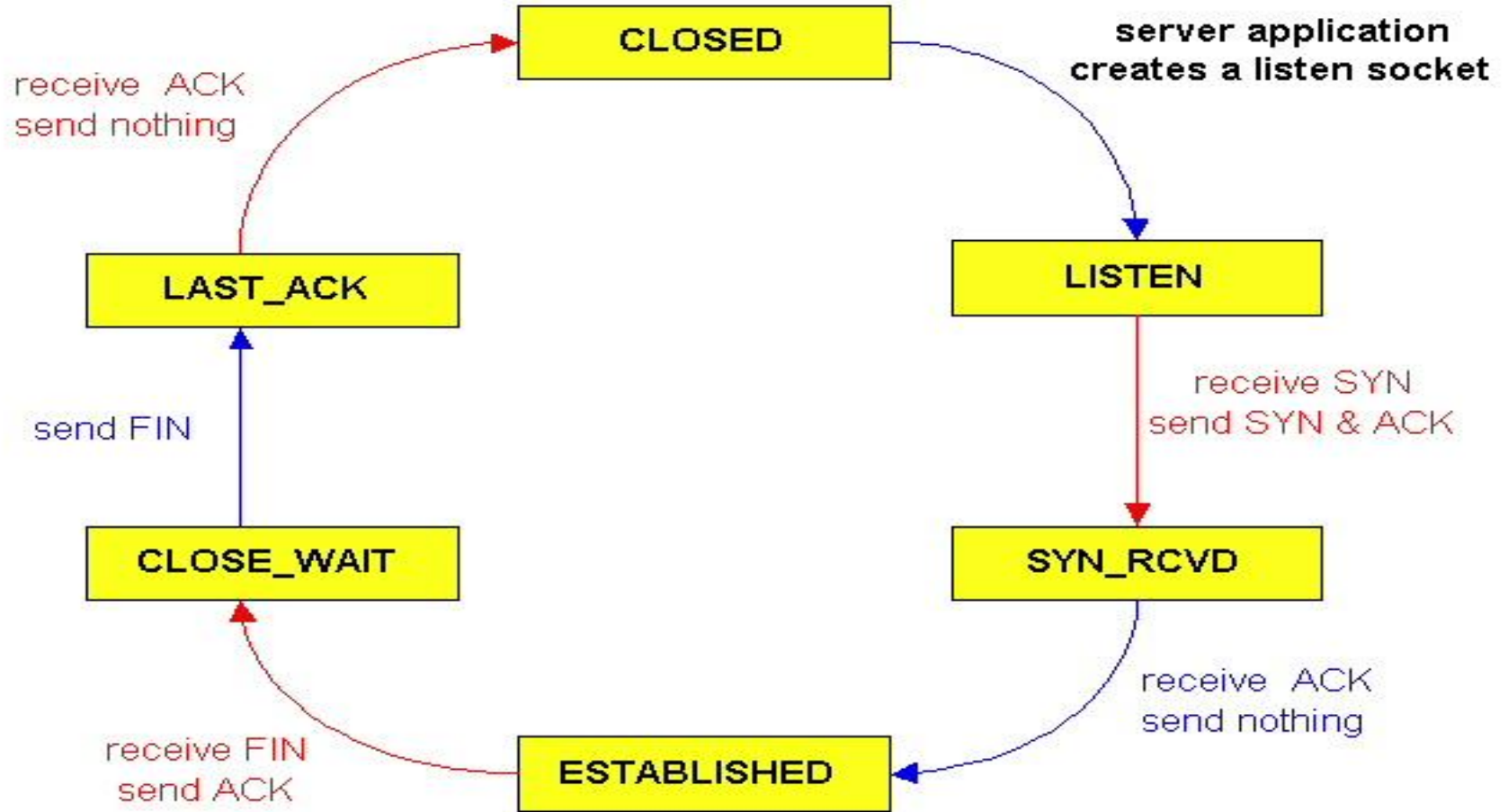
State machine



Client-side view



Server-side view



This lecture

- TCP connection management
 - connection establishment
 - connection release
 - state machine
- Explore further
 - Qs on previous slides
 - RFC793 and RFC1122
 - How do you do your RDP connection?

Next lecture

- TCP flow control
 - read KR4: Computer Networking
 - Chapter 3 (all sections required this month)

TCP seq. numbers, ACKs

sequence numbers:

- byte stream “number” of first byte in segment’s data

acknowledgements:

- seq # of next byte expected from other side
- cumulative ACK

Q: how receiver handles out-of-order segments

- A: TCP spec doesn’t say, - up to implementor

Transport Layer 3-25