

# CPSC 317 COMPUTER NETWORKING

2023W2: Transport – Day 8 – TCP wrapup

1

# READING

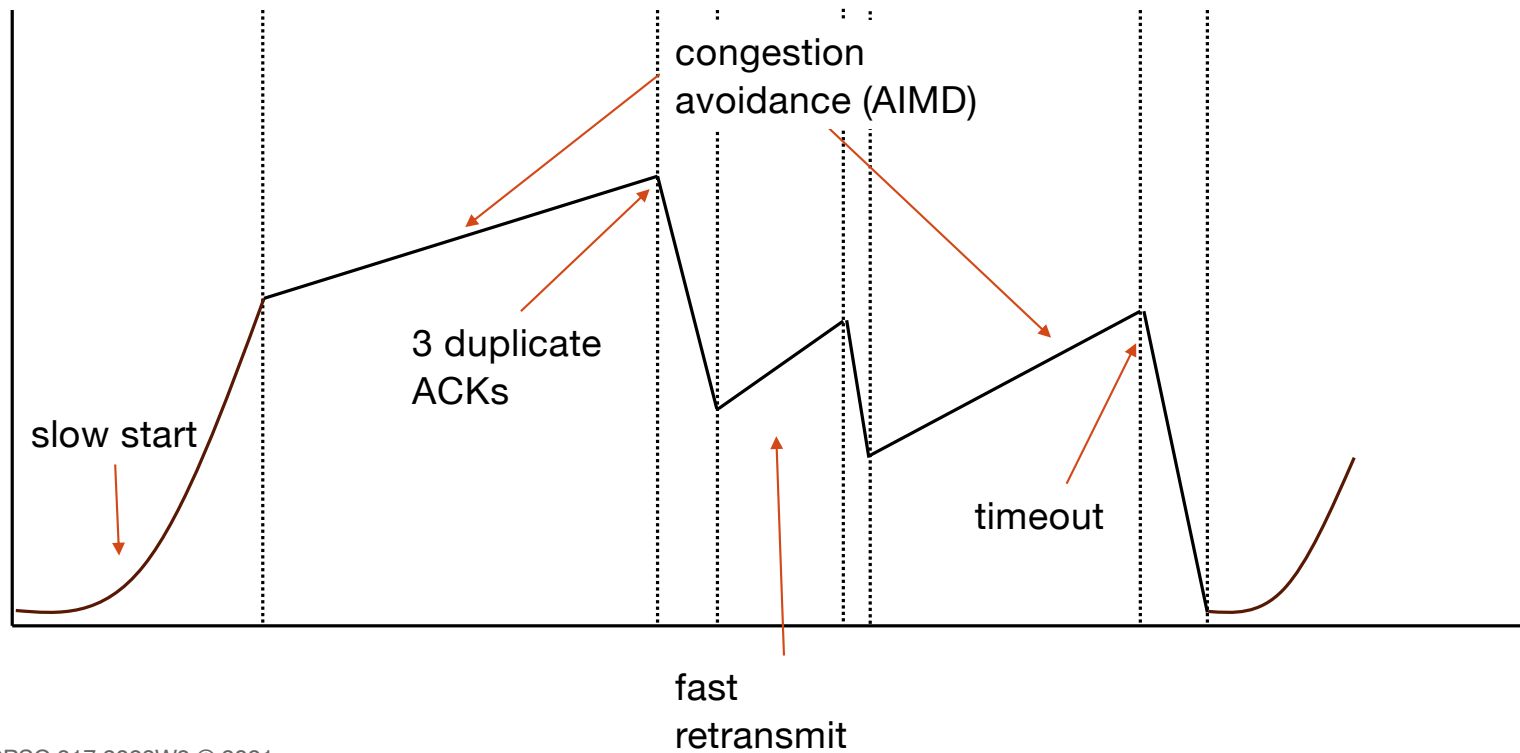
- Reading: 3.5.6

# LEARNING GOALS

## TCP

- Understand TCP congestion management
- Trace how TCP sets up a connection
- Trace how TCP terminates a connection
- Be aware of TCP's many complications

# TCP CONGESTION MANAGEMENT

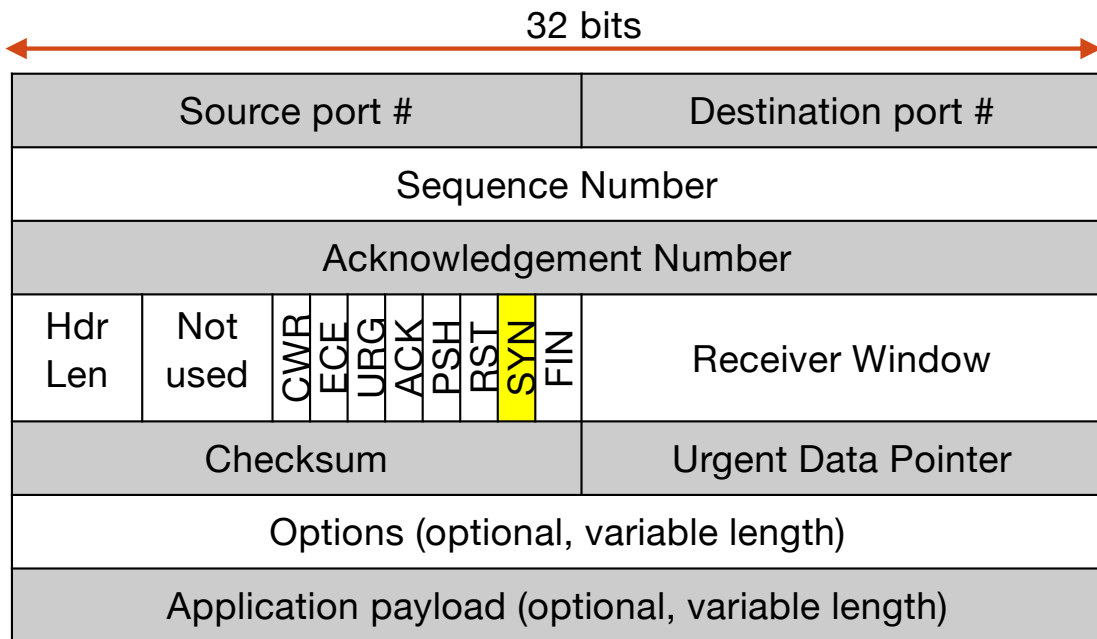


# TCP CONNECTION ESTABLISHMENT

- TCP uses a “three-way handshake” to establish connections
- Client sends an initial SYN message
  - Initial sequence number for client→server is specified
- Server responds with a SYN/ACK message
  - Client→server sequence number is confirmed in ACK
  - Server→client initial sequence number is specified
- Client sends an ACK message
  - Server→client sequence number is confirmed in ACK

# TCP SEGMENT FORMAT

- SYN – conn. start, synchronize seq#
- FIN – conn. close, no new data from sender
- RST – reset conn. (rejected connection)
- ACK – indicate that ACK # is set
- PSH – sender pushed the accumulated send buffer
- URG – segment contains “urgent” data; combined with urgent data pointer, points to the data to be delivered before other data.
- ECE – ECN-Echo, indicate ECN capability
- CWR – congestion window reduced



*For information only, you will not be tested on the format of the header*

# TCP CONNECTION ESTABLISHMENT

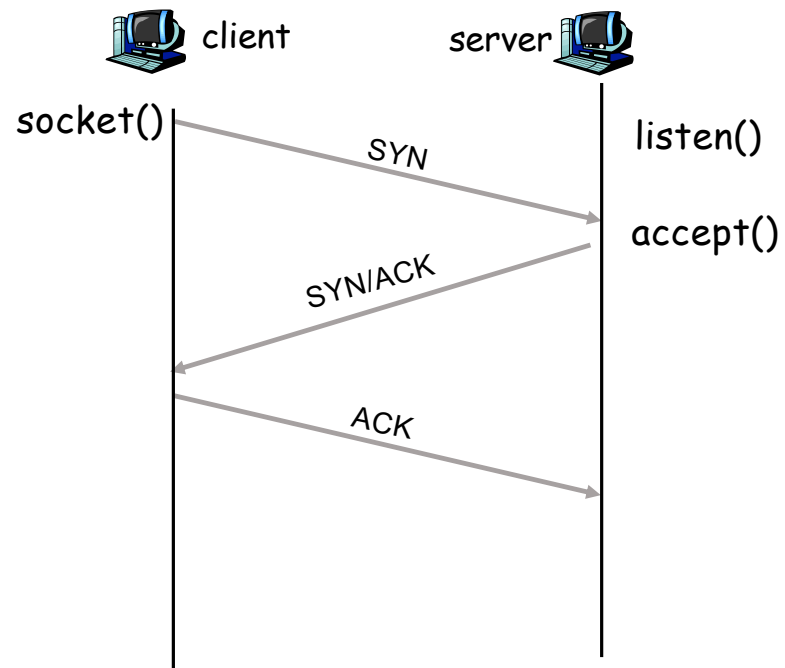
## Opening a connection:

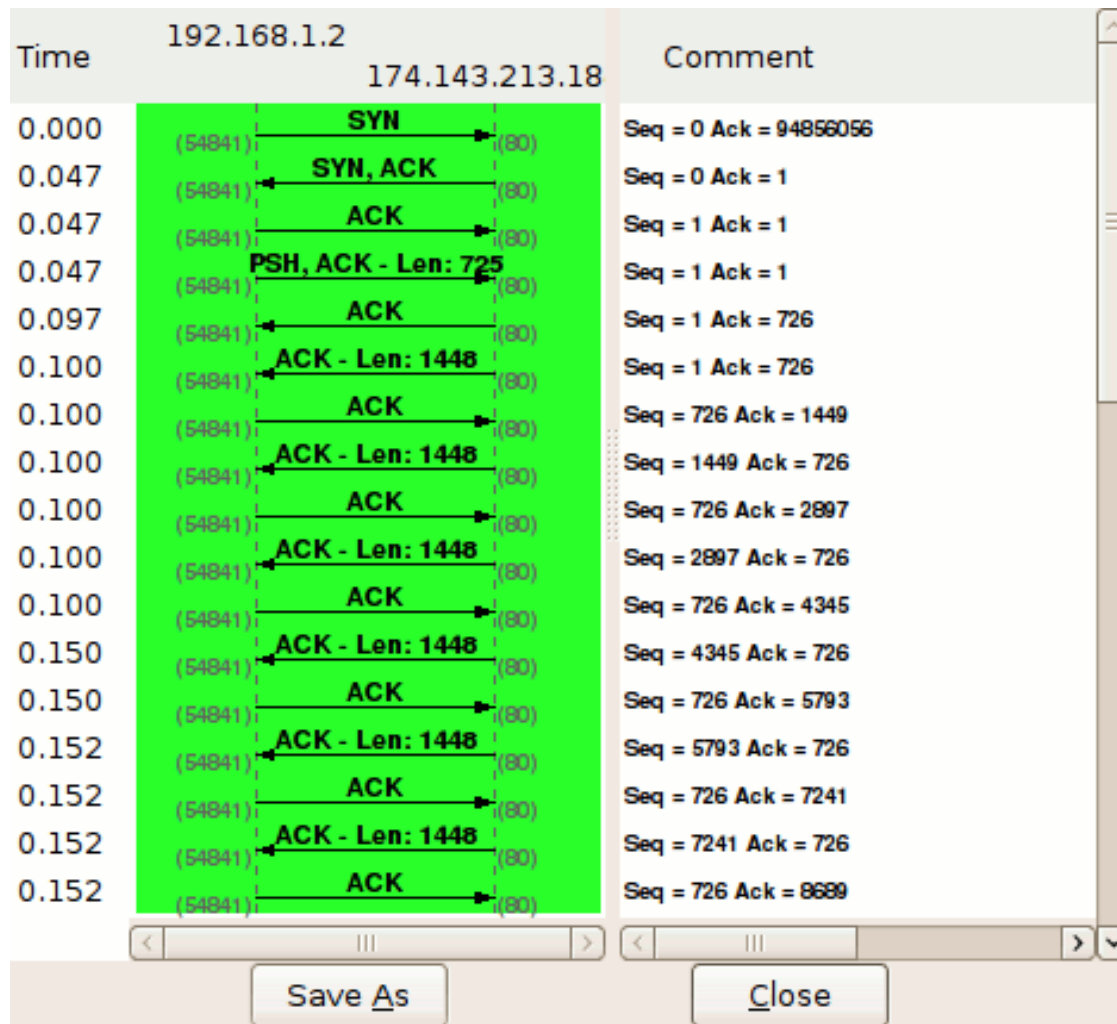
Client opens connection

**Step 1:** client end system sends TCP SYN control segment to server with an initial sequence number (ISN)

**Step 2:** server receives SYN, replies with SYN/ACK with server ISN

**Step 3:** client receives SYN/ACK, replies with ACK and may include data







Wireshark · Flow · knockknock2.pcap

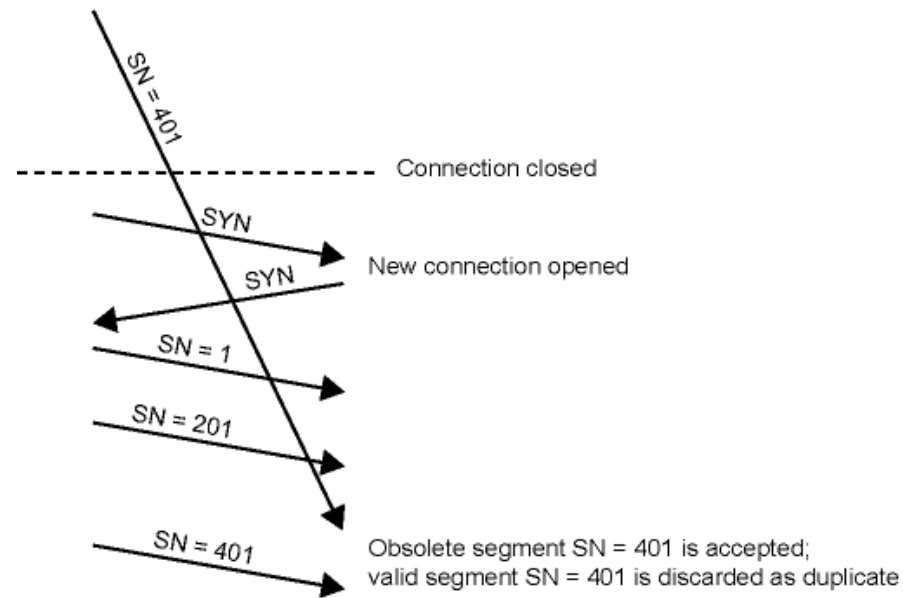
Time	127.0.0.1	Comment
0.000000	9989 ← 53602	TCP: 53602 → 9989 [SYN] Seq=0 Win=65535 ...
0.026307	53602 → 9989	TCP: 9989 → 53602 [SYN, ACK] Seq=0 Ack=1 ...
0.026349	9989 ← 53602	TCP: 53602 → 9989 [ACK] Seq=1 Ack=1 Win=...
0.026366	53602 → 9989	TCP: [TCP Window Update] 9989 → 53602 [A...
0.032040	53602 → 9989	TCP: 9989 → 53602 [PSH, ACK] Seq=1 Ack=1 ...
0.032072	9989 ← 53602	TCP: 53602 → 9989 [ACK] Seq=1 Ack=15 Win...
0.037536	9989 ← 53602	TCP: 53602 → 9989 [PSH, ACK] Seq=1 Ack=15...
0.037570	53602 → 9989	TCP: 9989 → 53602 [ACK] Seq=15 Ack=14 Wi...
0.037922	53602 → 9989	TCP: 9989 → 53602 [PSH, ACK] Seq=15 Ack=1...
0.037939	9989 ← 53602	TCP: 53602 → 9989 [ACK] Seq=14 Ack=22 Wi...
0.038094	9989 ← 53602	TCP: 53602 → 9989 [PSH, ACK] Seq=14 Ack=...
0.038173	53602 → 9989	TCP: 9989 → 53602 [ACK] Seq=22 Ack=26 Wi...

1 node, 22 items

Limit to display filter    Flow type: All Flows    Addresses: Any

Help    Reset Diagram    Export    Close

# SINGLE INITIAL SEQUENCE NUMBER PROBLEM



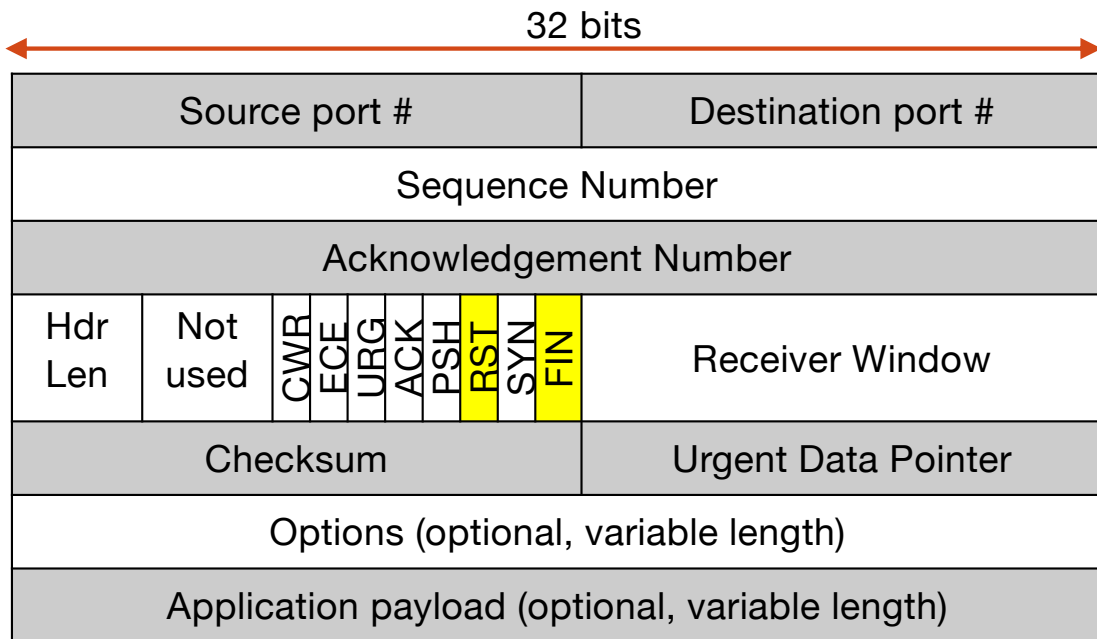
Solution: use randomly chosen initial sequence numbers

# TCP CONNECTION TERMINATION

- The side that wants to terminate sends FIN message
  - Other side responds with ACK
- The other side will also send a FIN message
  - It may not send it immediately, since it may have more data to send
  - The peer also responds with an ACK
- Alternative: connection reset (RST message)
  - Usually used if other side misbehaves or if disconnection or too many timeouts are detected

# TCP SEGMENT FORMAT

- SYN – conn. start, synchronize seq#
- FIN – conn. close, no new data from sender
- RST – reset conn. (rejected connection)
- ACK – indicate that ACK # is set
- PSH – sender pushed the accumulated send buffer
- URG – segment contains “urgent” data; combined with urgent data pointer, points to the data to be delivered before other data.
- ECE – ECN-Echo, indicate ECN capability
- CWR – congestion window reduced



*For information only, you will not be tested on the format of the header*

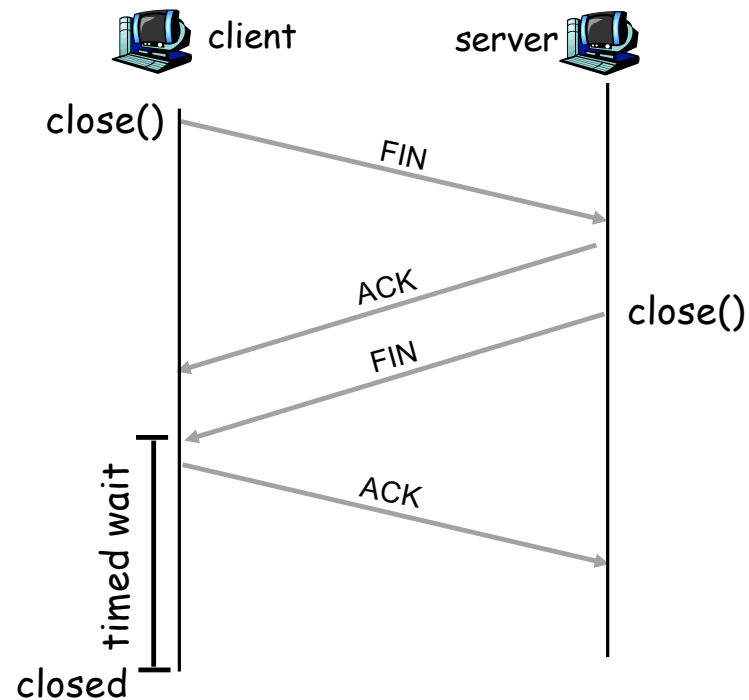
# TCP CONNECTION TERMINATION

## Closing a connection:

Client closes connection

**Step 1:** client end system sends TCP FIN control segment to server

**Step 2:** server receives FIN, replies with ACK. Some time later, it closes connection, sends FIN.



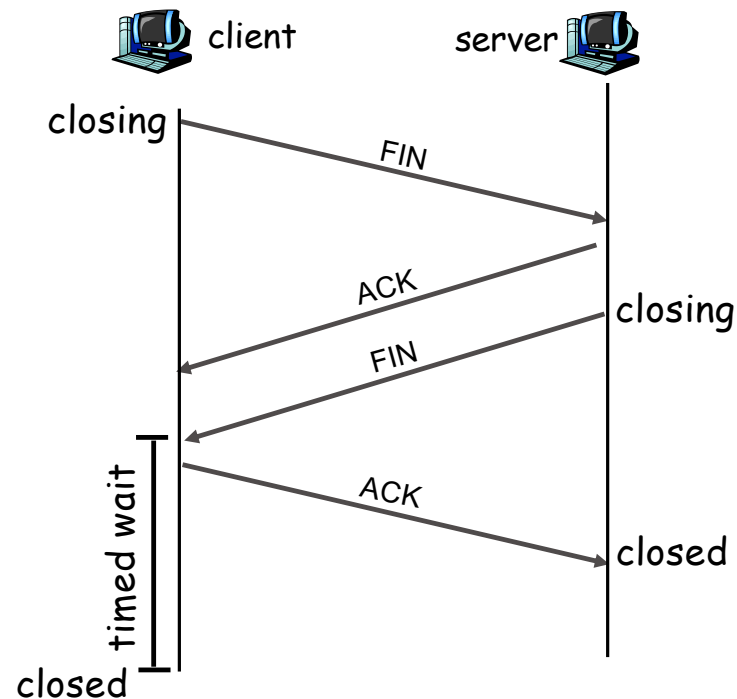
# TCP CONNECTION TERMINATION

**Step 3:** client receives FIN, replies with ACK.

- Enters “timed wait” - will respond with ACK to received FINs

**Step 4:** server receives ACK.  
Connection closed.

**Note:** with small modification, can handle simultaneous FINs.



Wireshark · Flow · knockknock2.pcap

Time	127.0.0.1	Comment
0.037939	9989 ← 53602	TCP: 53602 → 9989 [ACK] Seq=14 Ack=22 Wi...
0.038094	9989 ← 53602	TCP: 53602 → 9989 [PSH, ACK] Seq=14 Ack=...
0.038173	53602 → 9989	TCP: 9989 → 53602 [ACK] Seq=22 Ack=26 Wi...
0.042678	53602 → 9989	TCP: 9989 → 53602 [PSH, ACK] Seq=22 Ack=...
0.042696	9989 ← 53602	TCP: 53602 → 9989 [ACK] Seq=26 Ack=78 Wi...
0.043157	9989 ← 53602	TCP: 53602 → 9989 [PSH, ACK] Seq=26 Ack=...
0.043173	53602 → 9989	TCP: 9989 → 53602 [ACK] Seq=78 Ack=28 Wi...
0.043390	53602 → 9989	TCP: 9989 → 53602 [PSH, ACK] Seq=78 Ack=...
0.043403	9989 ← 53602	TCP: 53602 → 9989 [ACK] Seq=28 Ack=83 Wi...
0.043980	53602 → 9989	TCP: 9989 → 53602 [FIN, ACK] Seq=83 Ack=2...
0.043994	9989 ← 53602	TCP: 53602 → 9989 [ACK] Seq=28 Ack=84 Wi...
0.044034	9989 ← 53602	TCP: 53602 → 9989 [FIN, ACK] Seq=28 Ack=8...
0.044053	53602 → 9989	TCP: 9989 → 53602 [ACK] Seq=84 Ack=29 Wi...

1 node, 22 items

Limit to display filter      Flow type: All Flows      Addresses: Any

Help    Reset Diagram    Export    **Close**

# LOTS OF OTHER COMPLICATIONS

- Slow start congestion window details
- Nagle's algorithm (delayed sends)
- Delayed ACKs (only send half the ACKs)
- “Silly Window Syndrome”



# IN-CLASS ACTIVITY

- ICA48