CS2105

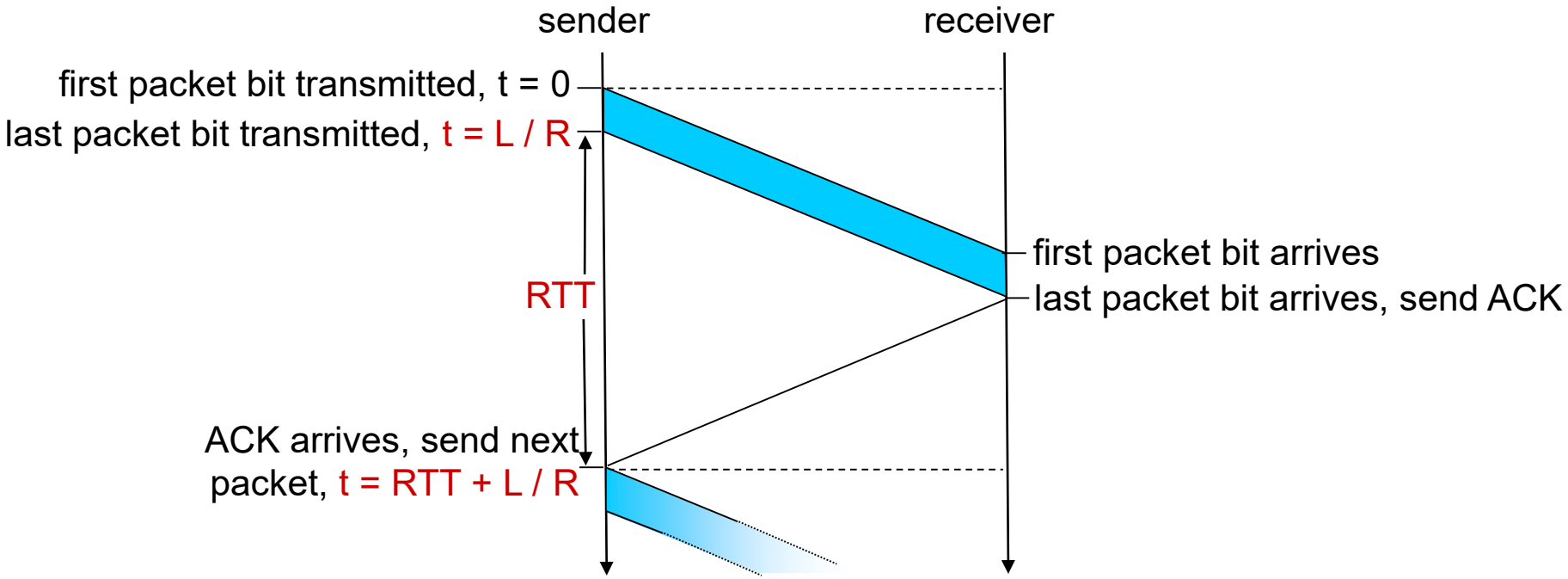
An Awesome Introduction to Computer Networks

Lecture 5 discussion

Department of Computer Science
School of Computing

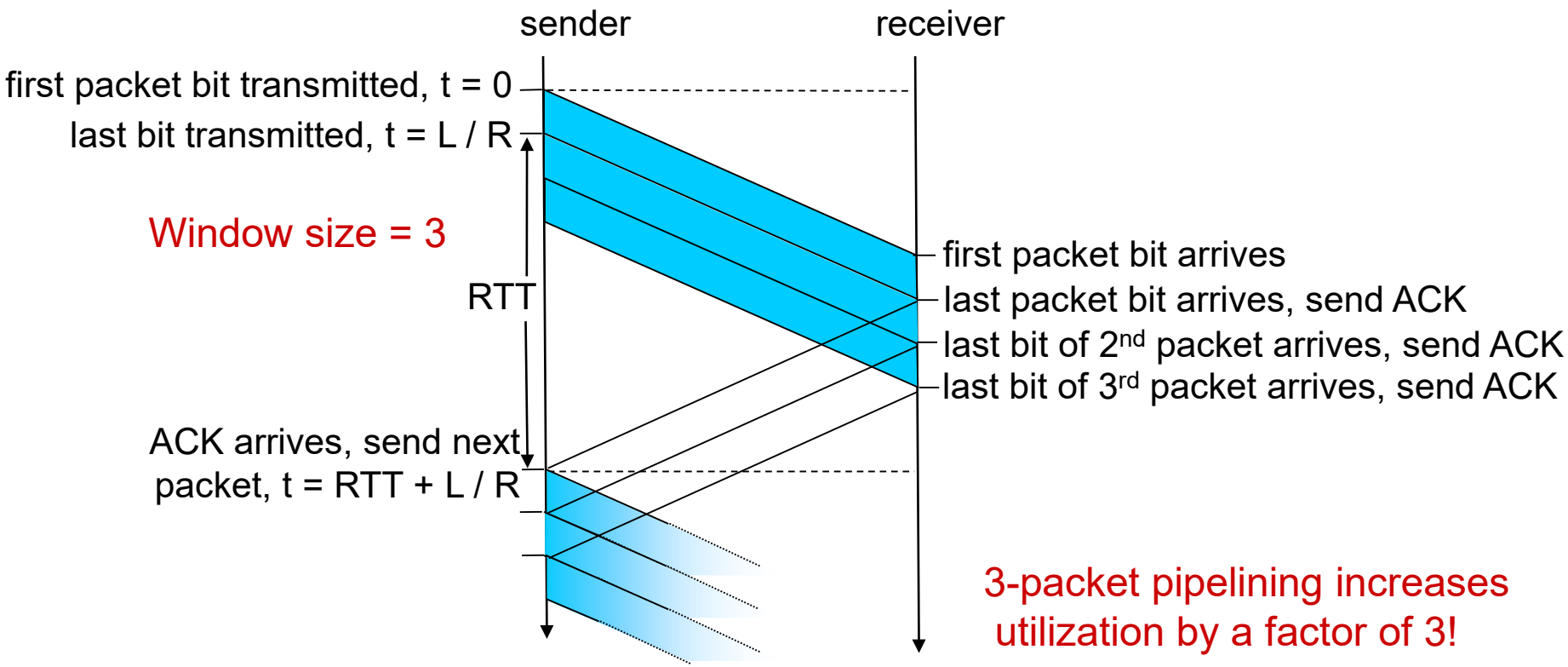
rdt 3.0: Stop-and-wait Operation

- Network protocol limits use of physical resources!



$$U_{sender} = \frac{L / R}{RTT + L/R} = \frac{0.008}{30 + 0.008} = 0.00027$$

Pipelining: Increased Utilization



3-packet pipelining increases utilization by a factor of 3!

$$U_{sender} = \frac{3 * L / R}{RTT + L/R} = \frac{0.024}{30 + 0.008} = 0.00081$$

GBN vs. SR

Go-back-N

- Sender can have up to *N* unACKed packets in pipeline
- Receiver only sends *cumulative ACKs*
  - Out-of-order packets discarded
- Sender sets timer for the oldest unACKed packet
  - when timer expires, retransmit ***all*** unACKed packets

Selective Repeat

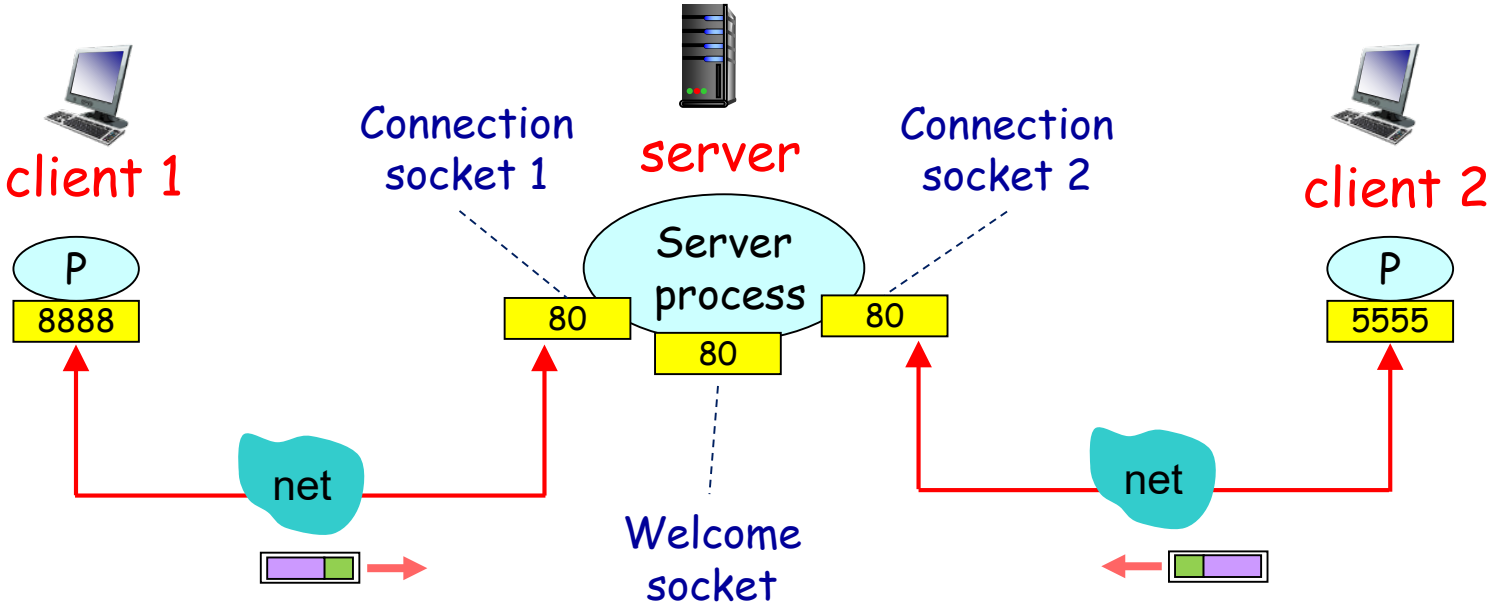
- Sender can have up to *N* unACKed packets in pipeline
- Receiver sends *individual ACK* for each packet
  - Out-of-order packets buffered
- Sender maintains timer for ***each*** unACKed packet
  - when timer expires, retransmit only that unACKed packet

TCP Overview [RFC 793, 1122, … 2581 …]

- Point-to-point:
  - One sender, one receiver.
- Connection-oriented:
  - handshaking (exchange of control messages) before sending app data.
- Full duplex service:
  - bi-directional data flow in the same connection
- Reliable, in-order *byte steam:*
  - use sequence numbers to label bytes

Connection-oriented De-mux

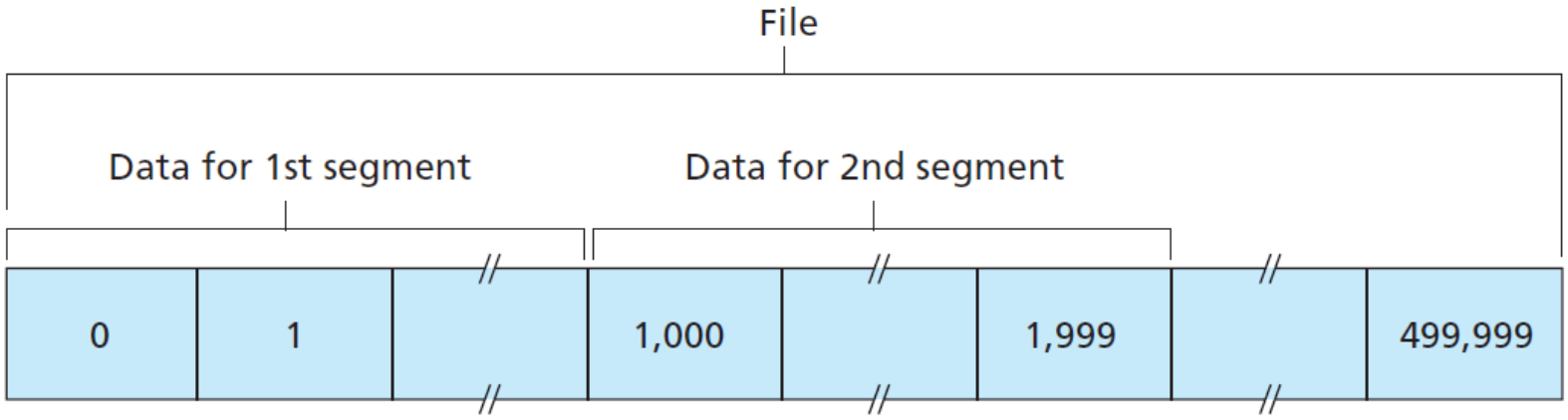
- A TCP connection (socket) is identified by 4-tuple:
  - (srcIPAddr, srcPort, destIPAddr, destPort)
  - Receiver uses all four values to direct a segment to the appropriate socket.



source port #	dest port #
sequence number	
ACK number	
checksum	

TCP Sequence Number

- "Byte number" of the first byte of data in a segment.
- Example: send a file of 500,000 bytes; MSS is 1,000 bytes.

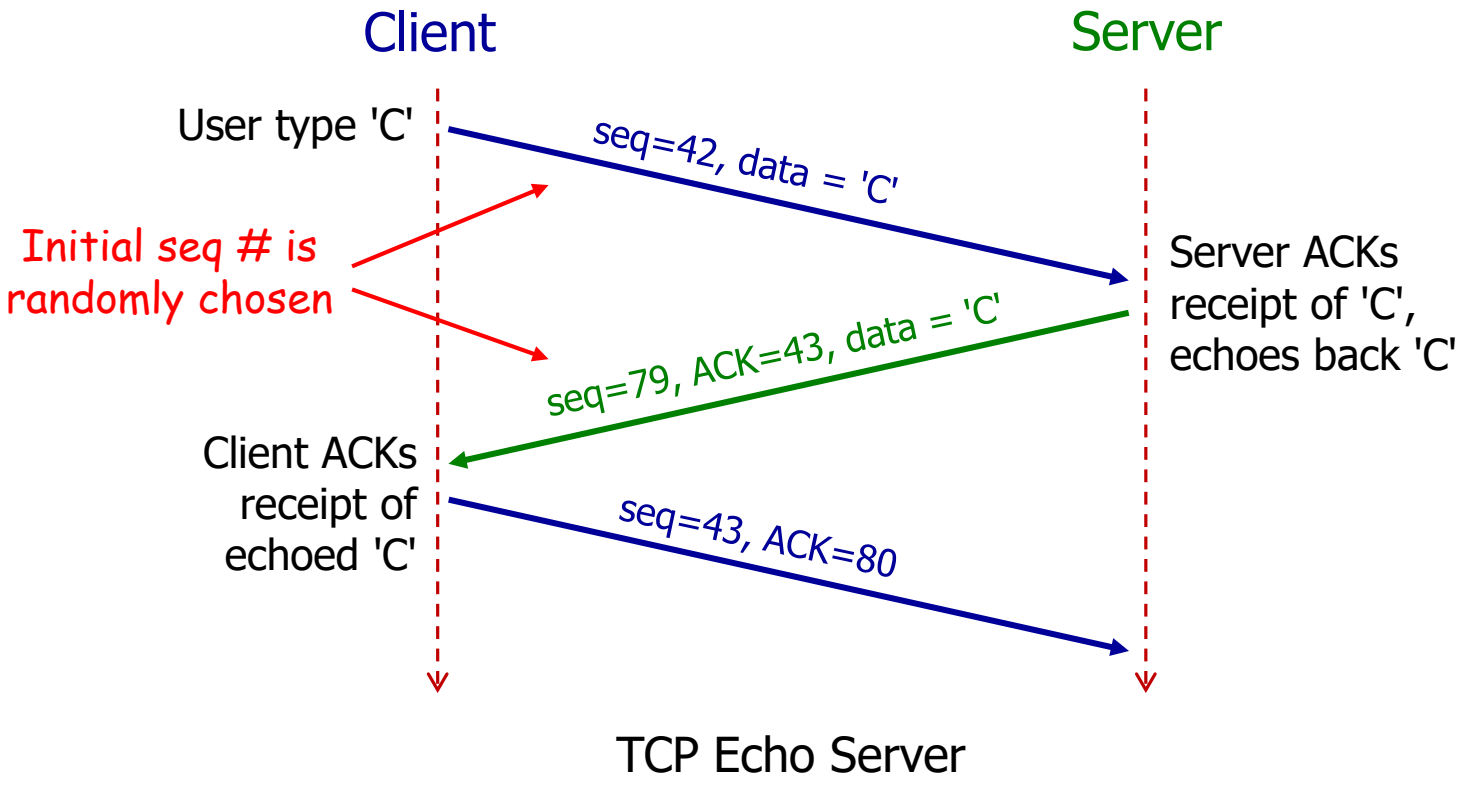


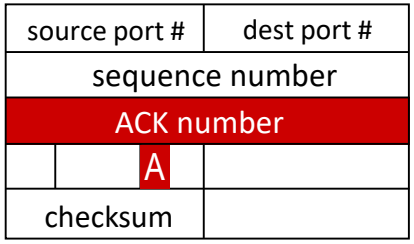
Dividing file data into TCP segments

- Seq. # of 1st TCP segment: 0, 2nd TCP segment: 1,000, 3rd TCP segment: 2,000, 4th TCP segment: 3,000, etc.

Example: TCP Echo Server

- TCP (and also UDP) is a full duplex protocol
  - bi-directional data flow in the same TCP connection.
- Example:





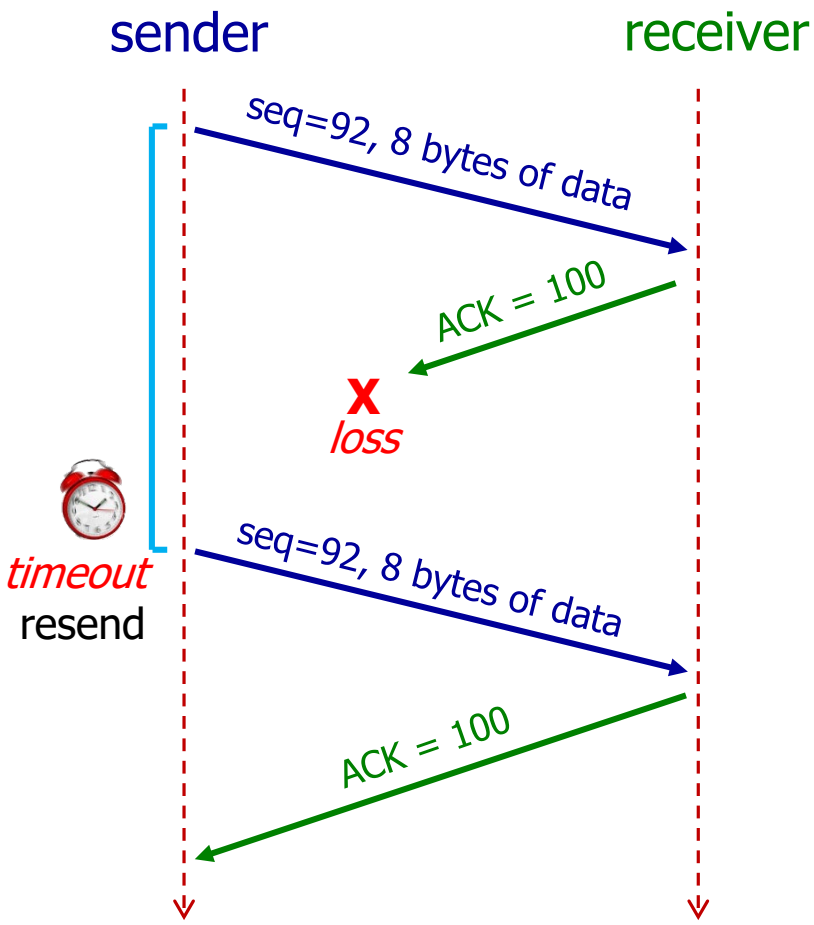
TCP ACK Number

- Seq # of the next byte of data expected by receiver.

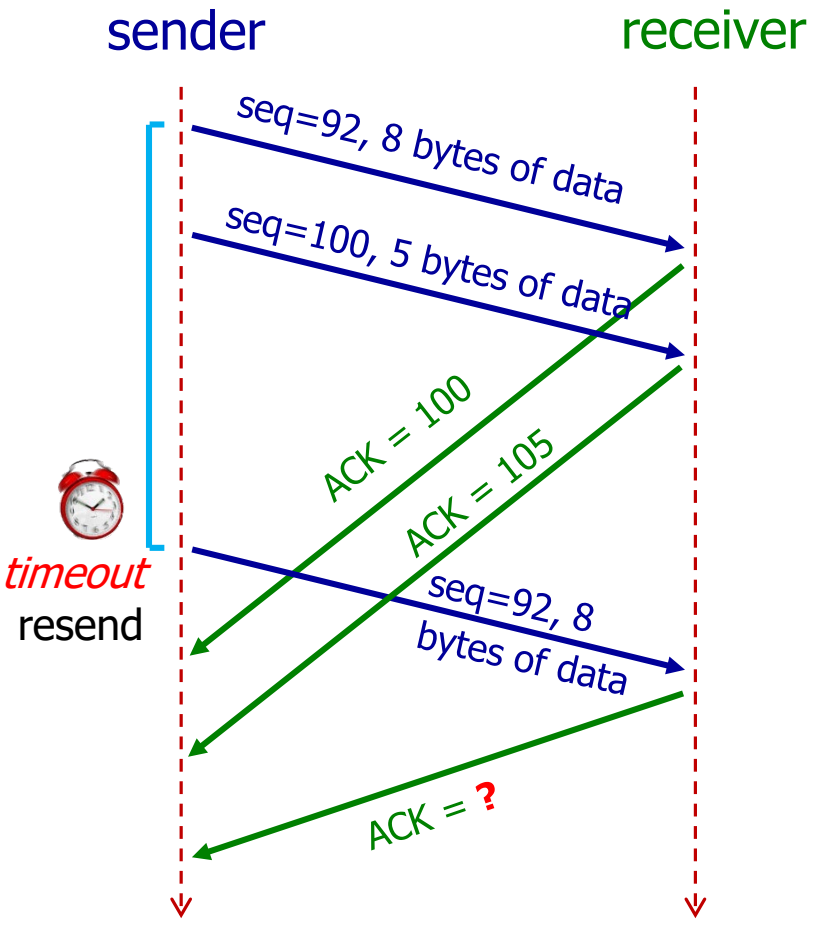
Sequence number of a segment	Amount of data carried	Corresponding ACK number
0	1,000	1,000
1,000	1,000	2,000
2,000	1,000	3,000
3,000	1,000	4,000
…	…	…

- TCP ACKs up to the first missing byte in the stream (cumulative ACK).
  - Note: TCP spec doesn't say how receiver should handle out-of-order segments - it's up to implementer.

TCP Timeout / Retransmission



a) Lost ACK

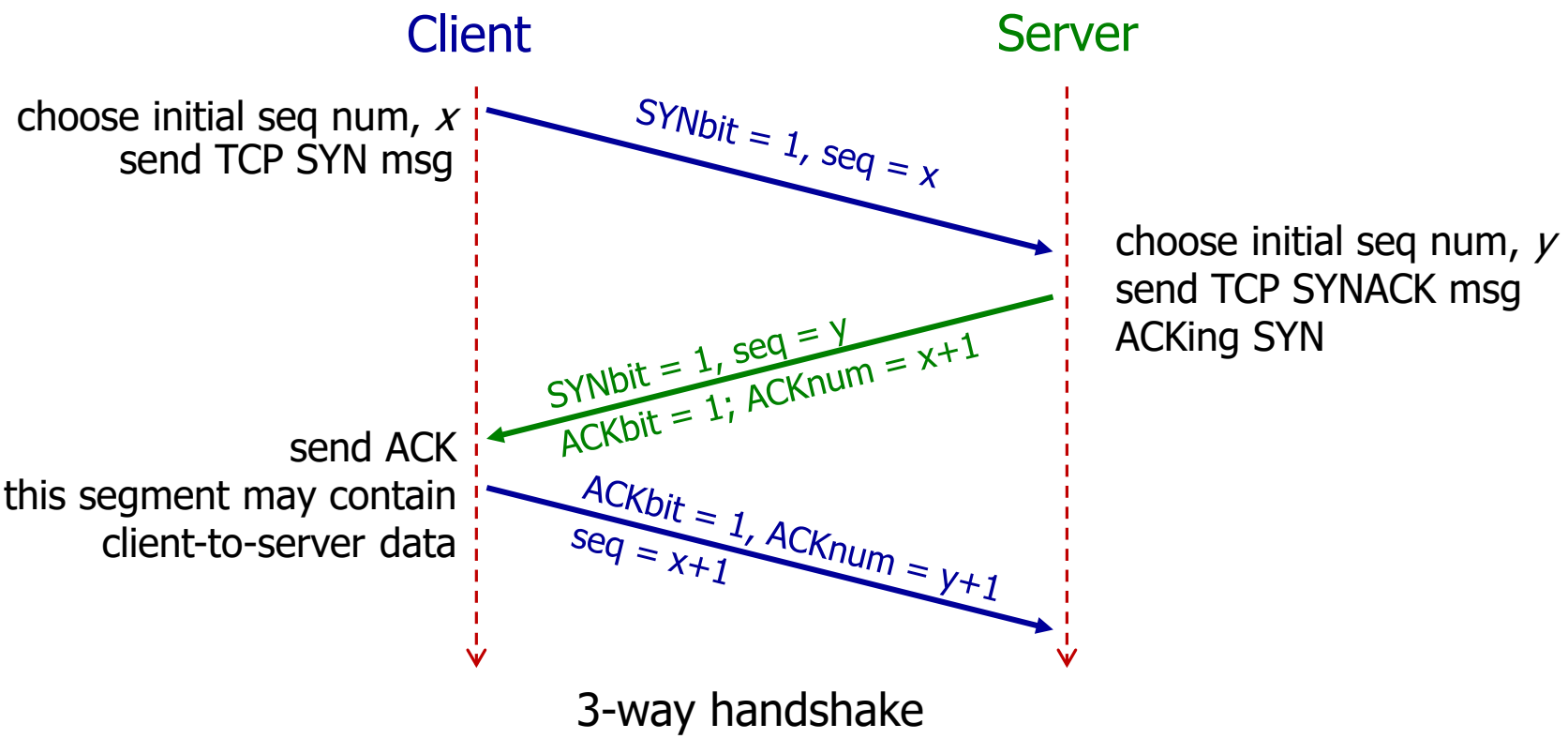


b) premature timeout

source port #	dest port #
sequence number	
ACK number	
A S	
checksum	

Establishing Connection

- Before exchanging app data, TCP sender and receiver "shake hands".
  - Agree on connection and exchange connection parameters.



3-way handshake

Q1

1s complement is used as checksum in _____.

A. TCP but not UDP

B. UDP but not TCP

C. Both TCP and UDP

Q2

Which of the following does UDP implement or guarantee?

A. Flow control
B. Congestion control
C. Minimum throughput
D. Maximum end-to-end delay
E. None of the above

Q3

A TCP sender transmits 5 segments with the sequence numbers 1000, 2000, 3000, 4000 and 5000. After a short while, the sender receives 4 acknowledgements with the ACK numbers 2000, 2000, 2000 and 5000.

What is the sequence number of the TCP segment that is lost?

A. 2000

B. 3000

C. 4000

D. 5000