LAB 2

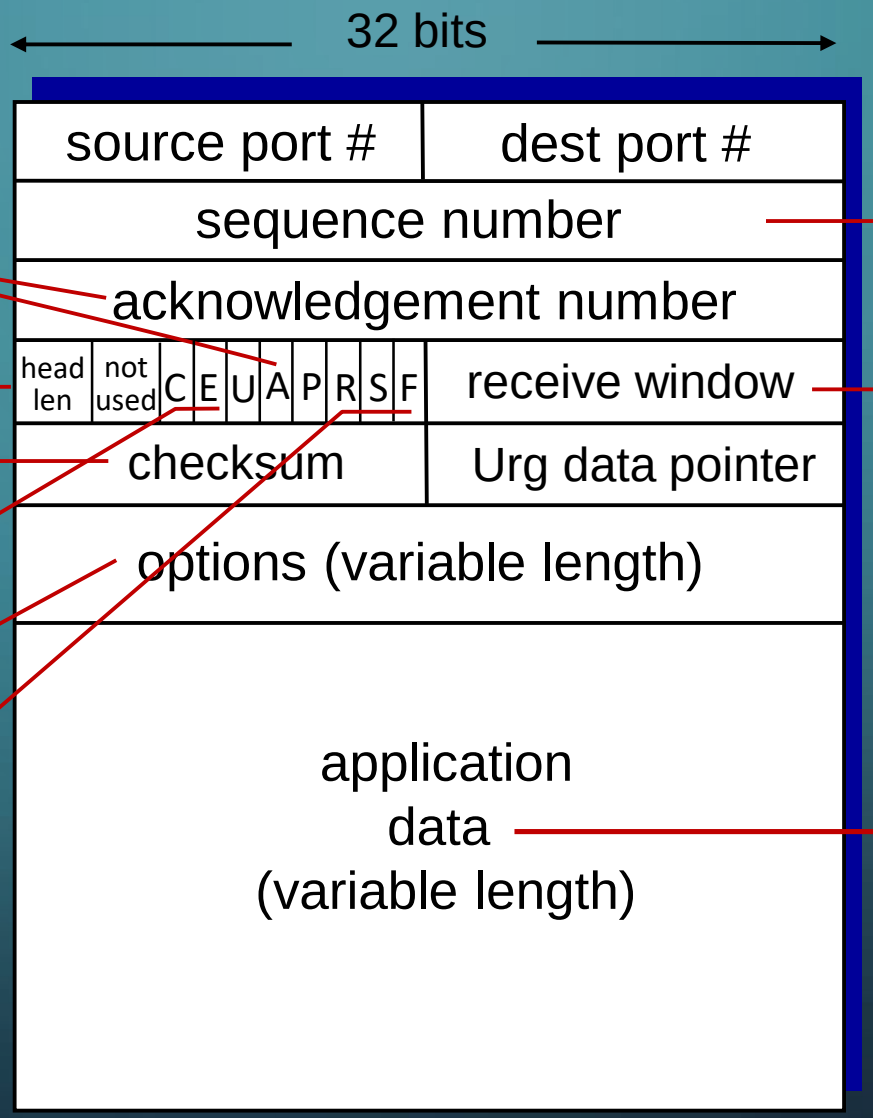
ΣΤΑΥΡΟΥΛΑ ΣΙΑΧΑΛΟΥ

ΣΠΥΡΟΣ ΜΕΓΑΛΟΥ

TCP: OVERVIEW RFCS: 793,1122, 2018, 5681, 7323

- point-to-point:
  - one sender, one receiver
- reliable, in-order *byte steam:*
  - no “message boundaries"
- full duplex data:
  - bi-directional data flow in same connection
  - MSS: maximum segment size
- cumulative ACKs
- pipelining:
  - TCP congestion and flow control set window size
- connection-oriented:
  - handshaking (exchange of control messages) initializes sender, receiver state before data exchange
- flow controlled:
  - sender will not overwhelm receiver

TCP SEGMENT STRUCTURE



ACK: seq # of next expected byte; A bit: this is an ACK

length (of TCP header)

Internet checksum

C, E: congestion notification

TCP options

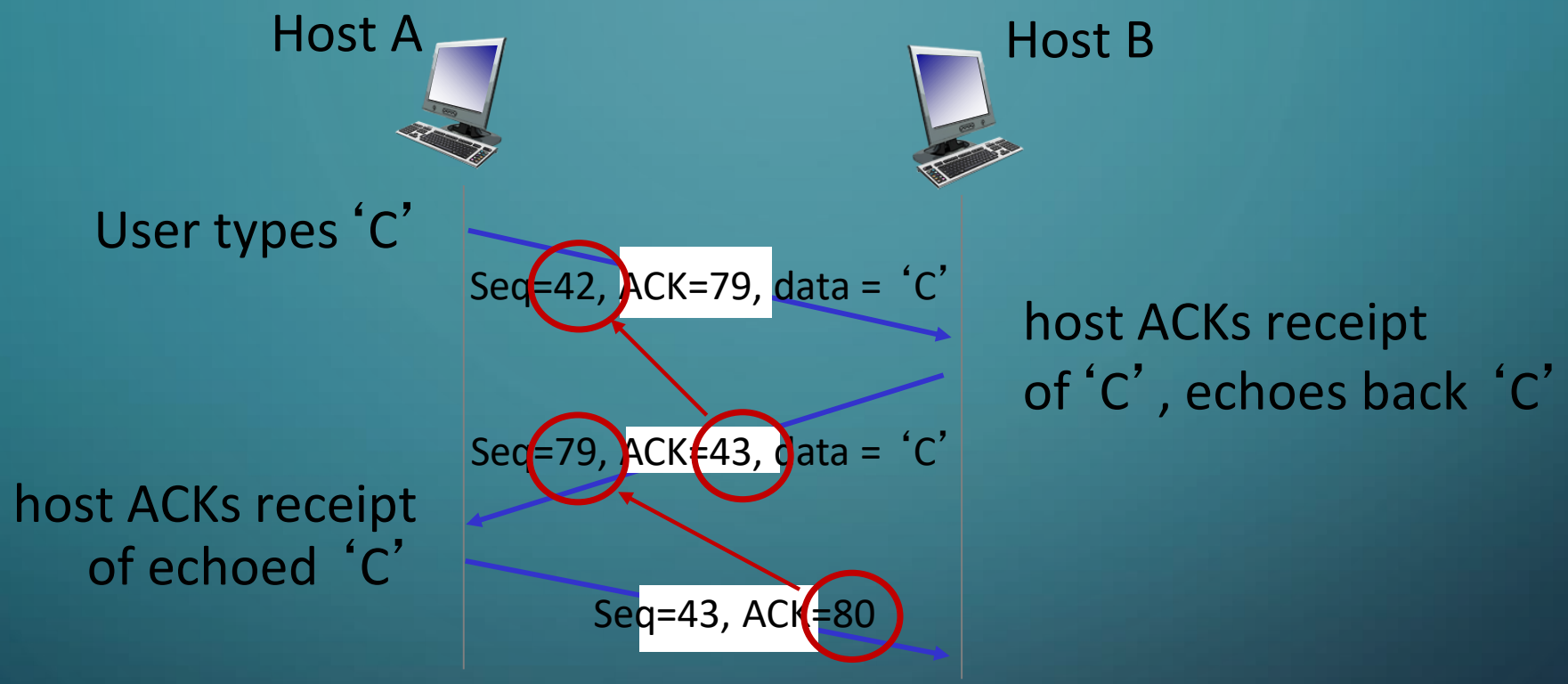
RST, SYN, FIN: connection management

segment seq #: counting bytes of data into bytestream (not segments!)

flow control: # bytes receiver willing to accept

data sent by application into TCP socket

TCP SEQUENCE NUMBERS, ACKS

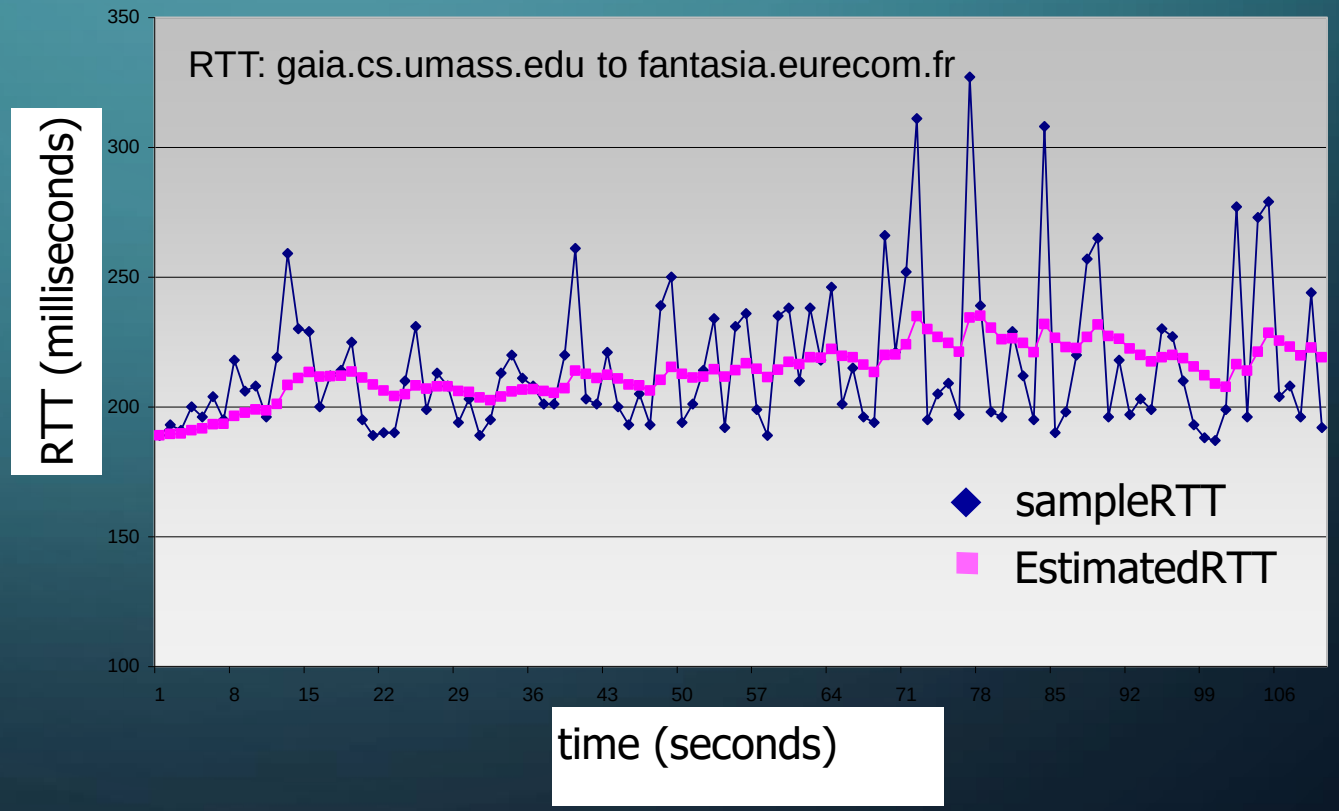


simple telnet scenario

TCP ROUND TRIP TIME, TIMEOUT

EstimatedRTT = (1- α)*EstimatedRTT + α*SampleRTT

- exponential weighted moving average (EWMA)
- influence of past sample decreases exponentially fast
- typical value: α = 0.125



TCP ROUND TRIP TIME, TIMEOUT

- timeout interval: **`EstimatedRTT`** plus "safety margin"
  - large variation in **`EstimatedRTT`**: want a larger safety margin

```
TimeoutInterval = EstimatedRTT + 4*DevRTT
```

estimated RTT

"safety margin"

- **`DevRTT`**: EWMA of **`SampleRTT`** deviation from **`EstimatedRTT`**:

```
DevRTT = (1-β)*DevRTT + β*|SampleRTT-EstimatedRTT|
```

(typically, $\beta = 0.25$)

TCP 3-WAY HANDSHAKE

Client state

```
clientSocket = socket(AF_INET, SOCK_STREAM)
```

LISTEN

```
clientSocket.connect((serverName,serverPort))
```

choose init seq num, x
send TCP SYN msg

SYNSENT

SYNbit=1, Seq=x

choose init seq num, y
send TCP SYNACK
msg, acking SYN

SYNbit=1, Seq=y
ACKbit=1; ACKnum=x+1

received SYNACK(x)
indicates server is live;
send ACK for SYNACK;
this segment may contain
client-to-server data

ESTAB

ACKbit=1, ACKnum=y+1

received ACK(y)
indicates client is live

Server state

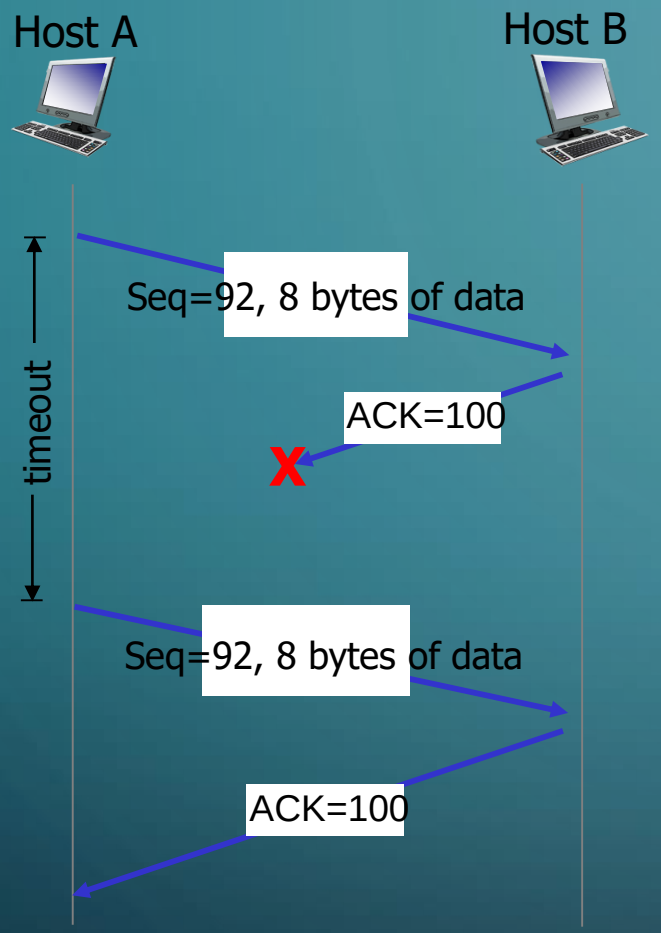
```
serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind(('',serverPort))
serverSocket.listen(1)
connectionSocket, addr = serverSocket.accept()
```

LISTEN

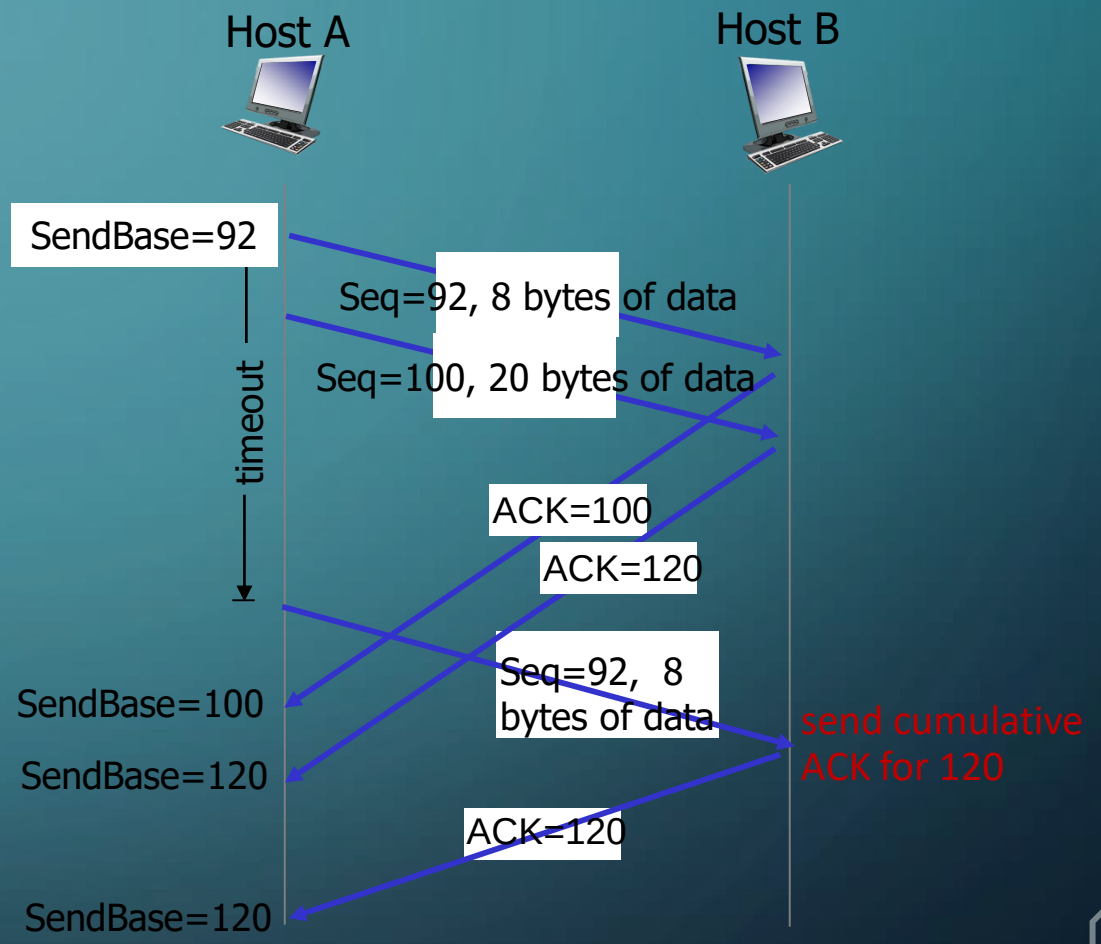
SYN RCVD

ESTAB

TCP: RETRANSMISSION SCENARIOS

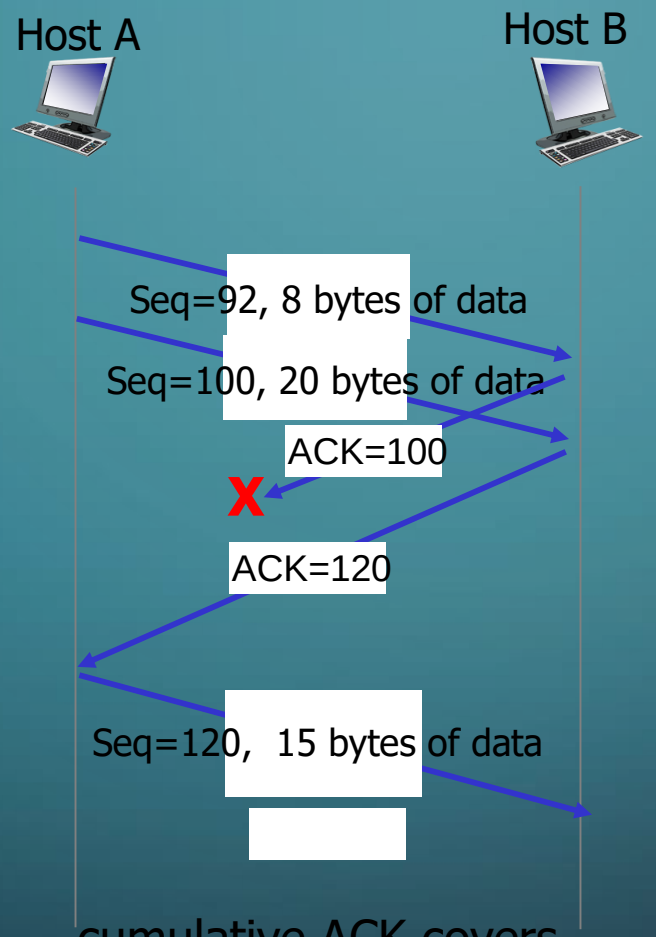


lost ACK scenario



premature timeout

TCP: RETRANSMISSION SCENARIOS



cumulative ACK covers
for earlier lost ACK

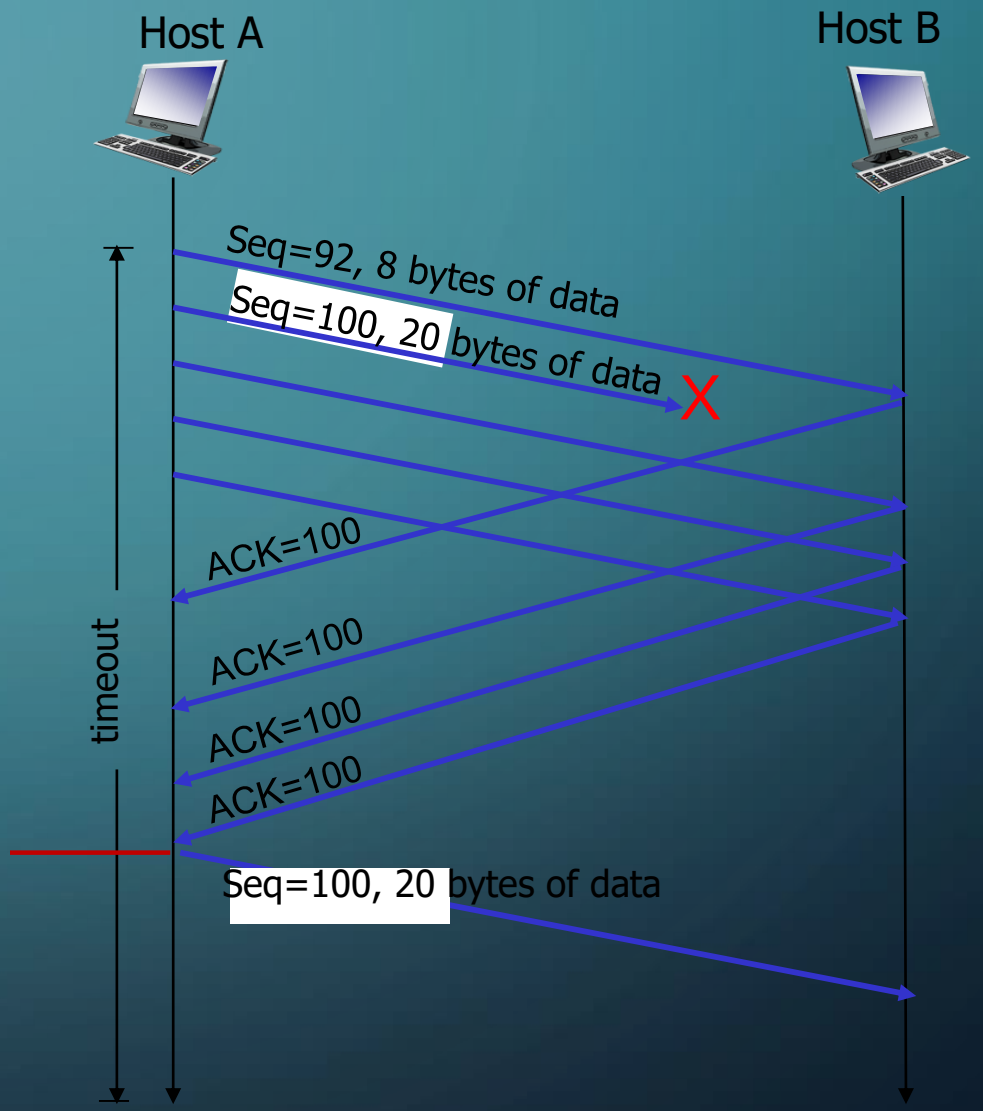
TCP FAST RETRANSMIT

TCP fast retransmit

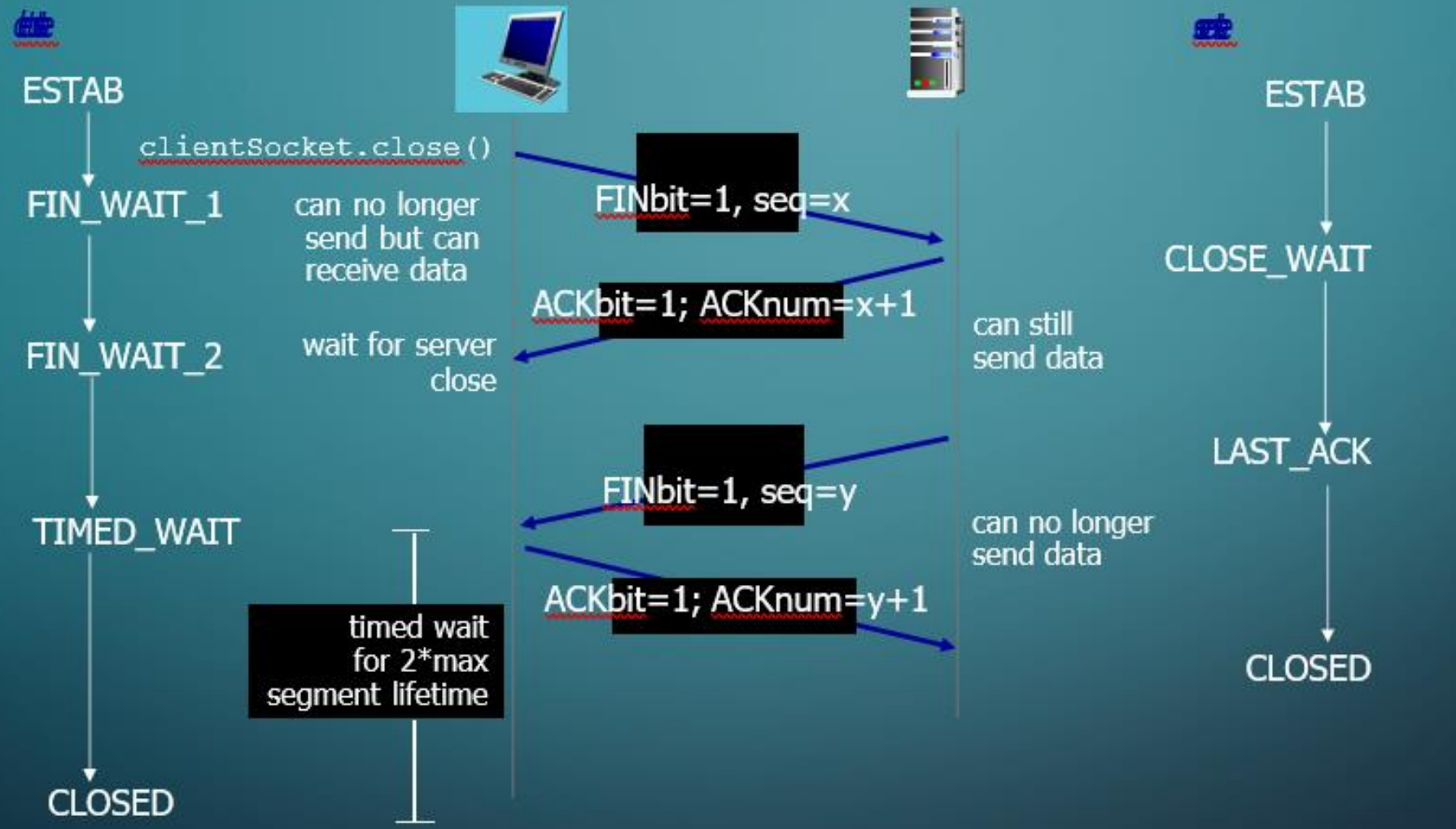
if sender receives 3 additional ACKs for same data ("triple duplicate ACKs"), resend unACKed segment with smallest seq #

- likely that unACKed segment lost, so don't wait for timeout

Receipt of three duplicate ACKs indicates 3 segments received after a missing segment – lost segment is likely. So retransmit!



TCP: ΚΛΕΙΣΙΜΟ ΣΥΝΔΕΣΗΣ



- Ο πελάτης και ο εξυπηρετητής κλείνουν την σύνδεση στέλνοντας το FIN bit = 1
- Ο καθένας απαντάει με ACK μόλις λάβει FIN
  - on receiving FIN, ACK can be combined with own FIN