



# COMPUTER NETWORKS

---

**Animesh Giri**

Department of Computer Science & Engineering

# COMPUTER NETWORKS

---

## Transport Layer

**Animesh Giri**

Department of Computer Science & Engineering

# COMPUTER NETWORKS

---

## Connection-oriented transport: TCP Flow control

**Animesh Giri**

Department of Computer Science & Engineering

# COMPUTER NETWORKS

## In this segment

---

- TCP flow control



# COMPUTER NETWORKS

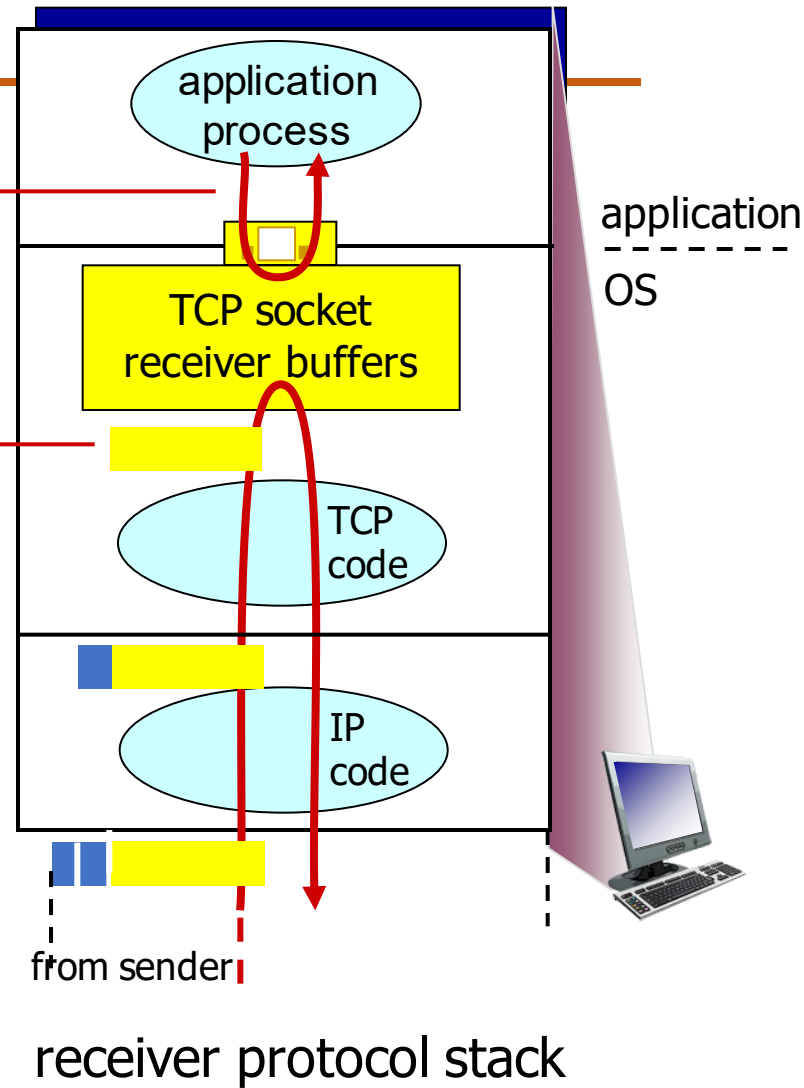
## TCP flow control

application may  
remove data from  
TCP socket buffers ....

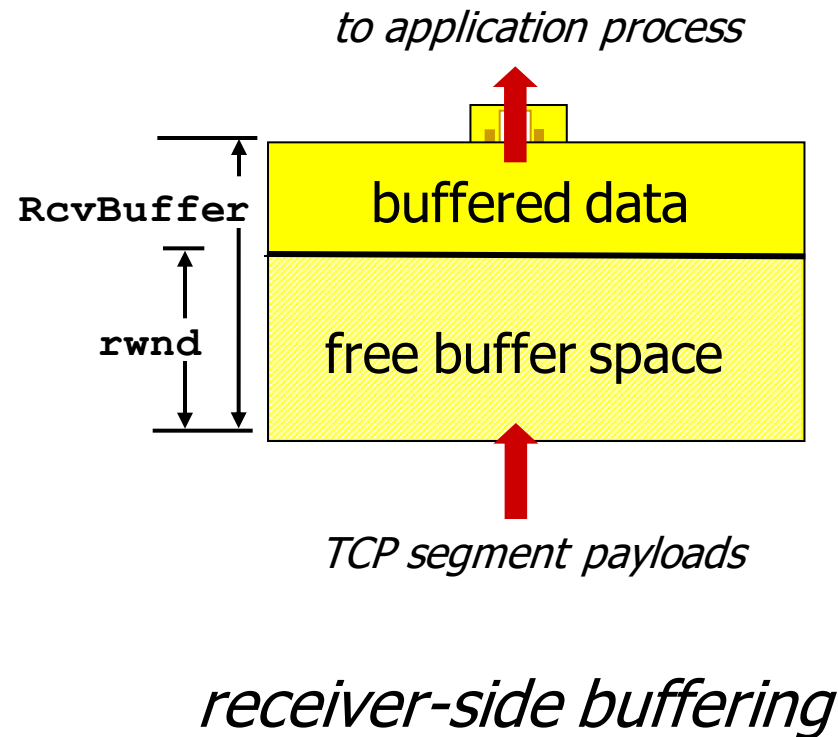
... slower than TCP  
receiver is delivering  
(sender is sending)

### *flow control*

receiver controls sender, so  
sender won't overflow  
receiver's buffer by transmitting  
too much, too fast



- receiver “advertises” free buffer space by including **rwnd** value in TCP header of receiver-to-sender segments
  - **RcvBuffer** size set via socket options (typical default is 4096 bytes)
  - many operating systems autoadjust **RcvBuffer**
- sender limits amount of unacked (“in-flight”) data to receiver’s **rwnd** value
- guarantees receive buffer will not overflow



# COMPUTER NETWORKS

---

## Connection-oriented transport: TCP Connection Management

**Animesh Giri**

Assistant Professor, Department of Computer Science & Engineering

# COMPUTER NETWORKS

## In this segment

---

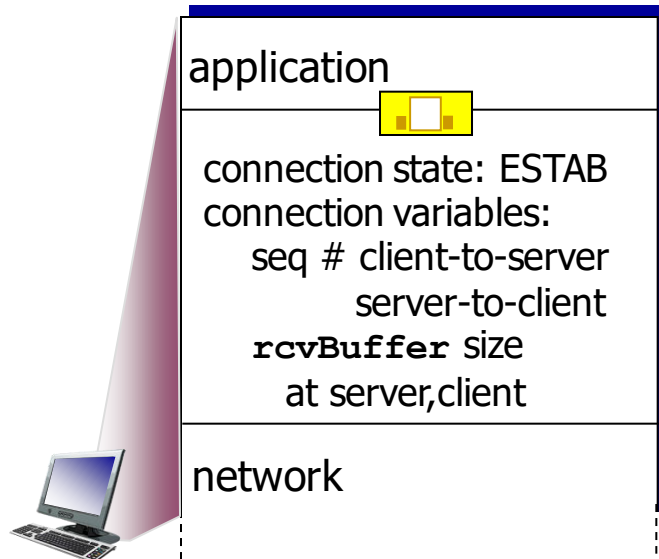
- Connection Management
- Agreeing to establish a connection
- TCP 3-way handshake
- TCP 3-way handshake: FSM
- TCP: closing a connection



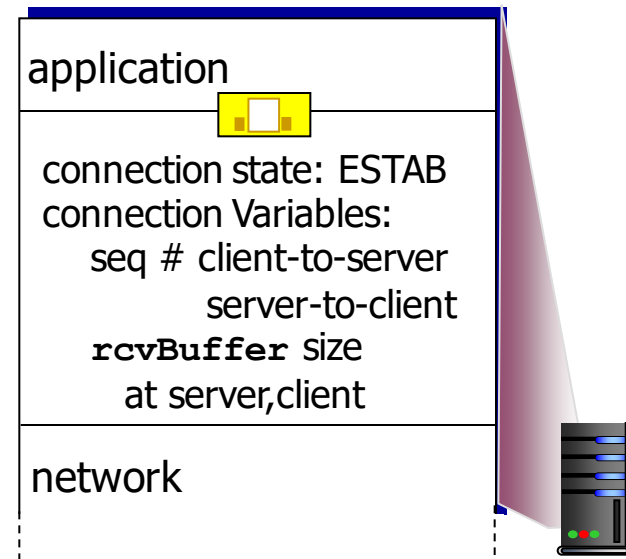


before exchanging data, sender/receiver “handshake”:

- agree to establish connection (each knowing the other willing to establish connection)
- agree on connection parameters

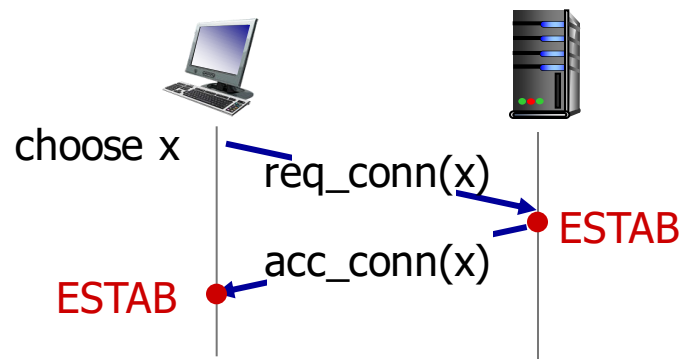
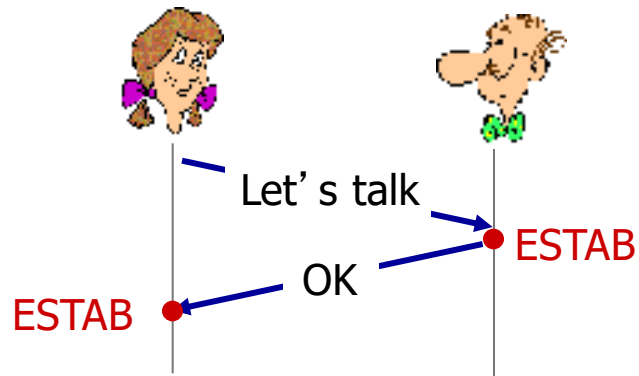


```
Socket clientSocket =  
    newSocket("hostname", "port  
    number");
```



```
Socket connectionSocket =  
    welcomeSocket.accept();
```

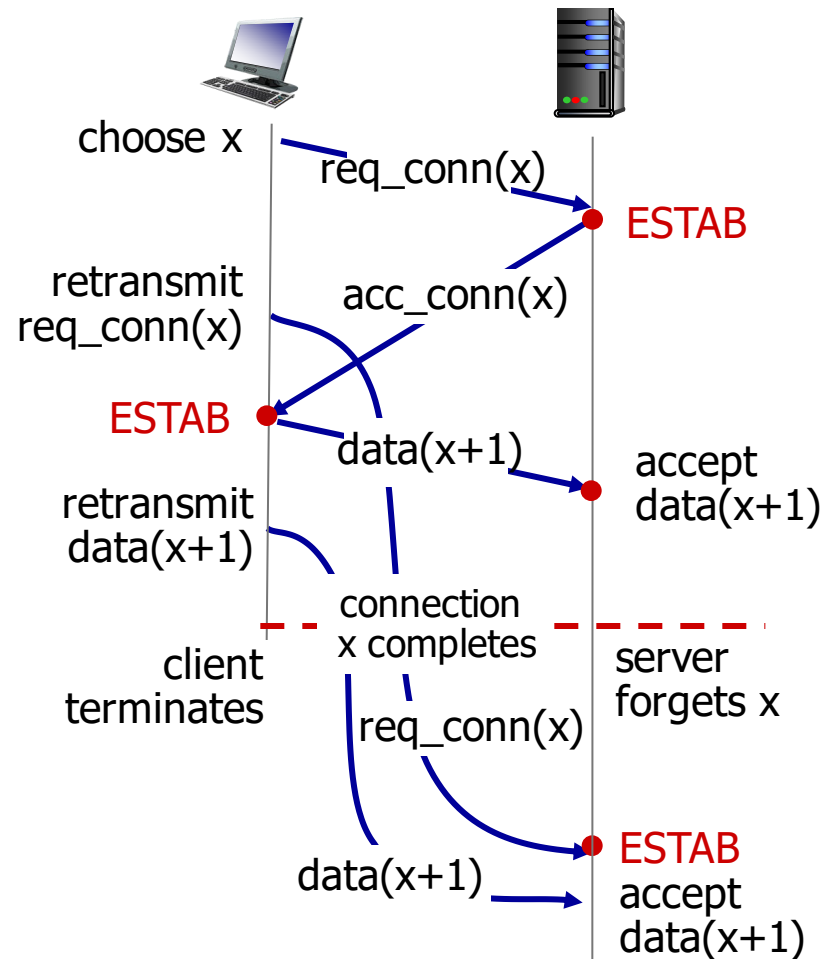
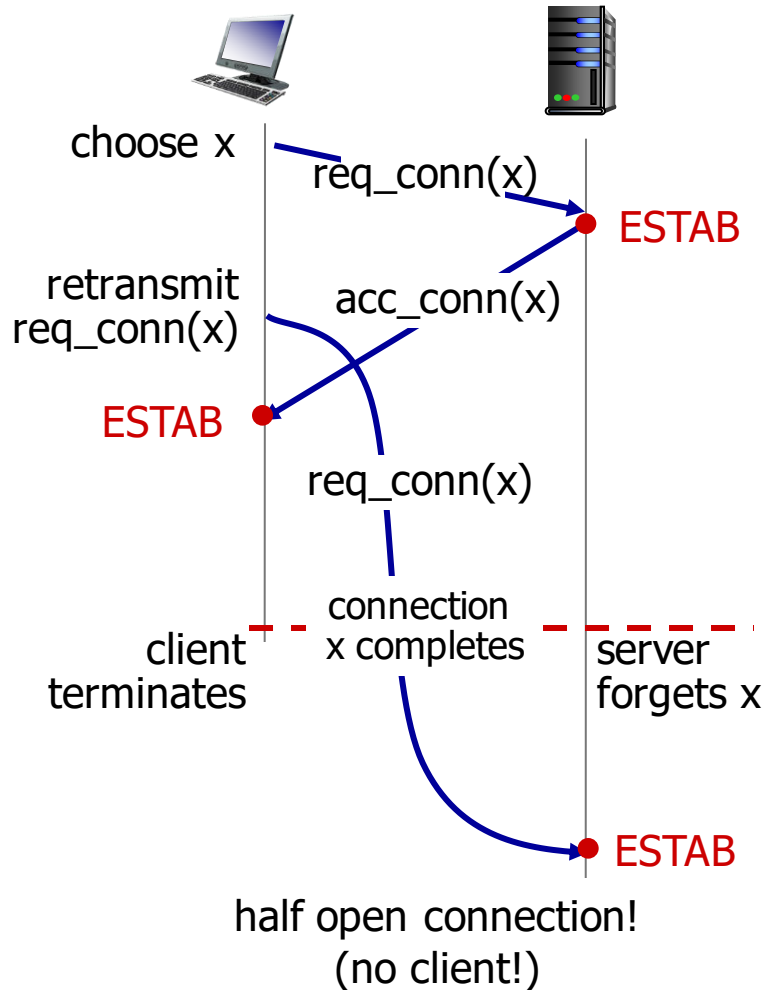
2-way handshake:



Q: will 2-way handshake always work in network?

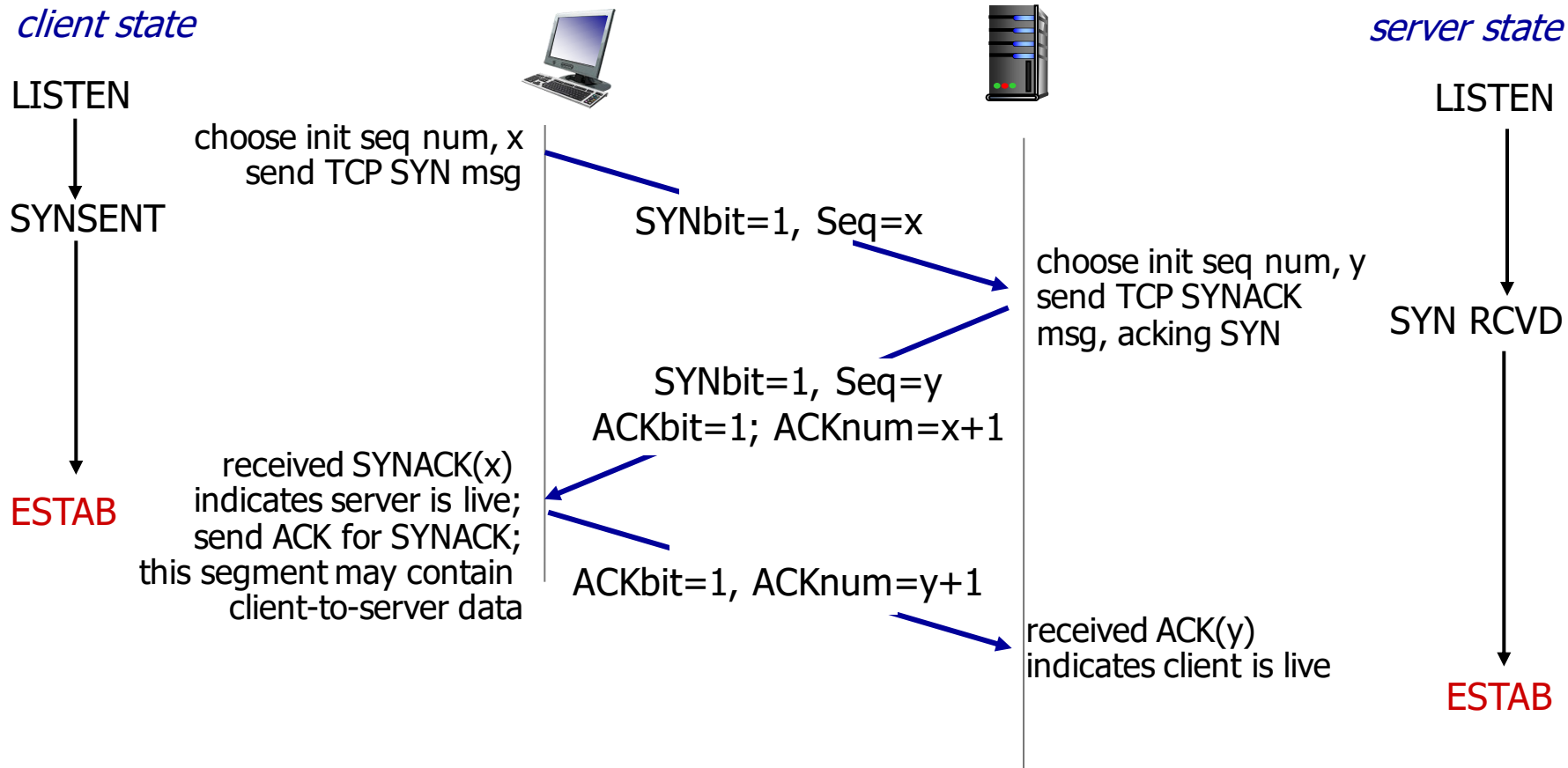
- variable delays
- retransmitted messages (e.g. req\_conn(x)) due to message loss
- message reordering
- can't "see" other side

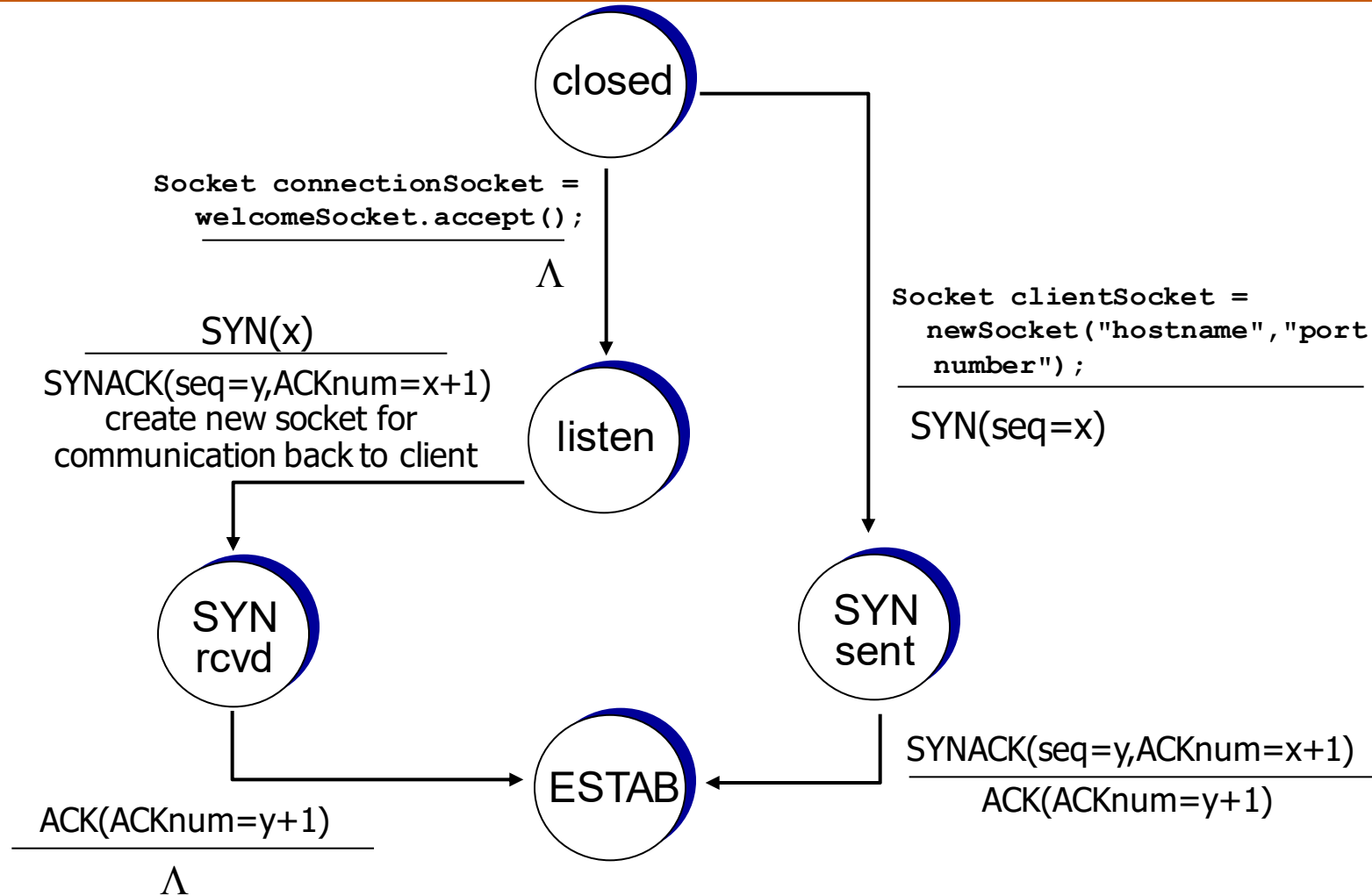
### 2-way handshake failure scenarios:



# COMPUTER NETWORKS

## TCP 3-way handshake





- client, server each close their side of connection
  - send TCP segment with FIN bit = 1
- respond to received FIN with ACK
  - on receiving FIN, ACK can be combined with own FIN
- simultaneous FIN exchanges can be handled

# COMPUTER NETWORKS

## TCP: closing a connection

*client state*

ESTAB

`clientSocket.close()`

FIN\_WAIT\_1

can no longer  
send but can  
receive data

FIN\_WAIT\_2

wait for server  
close

TIMED\_WAIT

timed wait  
for  $2 \times \text{max}$   
segment lifetime

CLOSED



FINbit=1, seq=x

ACKbit=1; ACKnum=x+1

FINbit=1, seq=y

ACKbit=1; ACKnum=y+1

can still  
send data

can no longer  
send data

*server state*

ESTAB

CLOSE\_WAIT

LAST\_ACK

CLOSED



**THANK YOU**

---

**Animesh Giri**

Department of Computer Science & Engineering

**[animeshgiri@pes.edu](mailto:animeshgiri@pes.edu)**

**+91 80 66186603**