

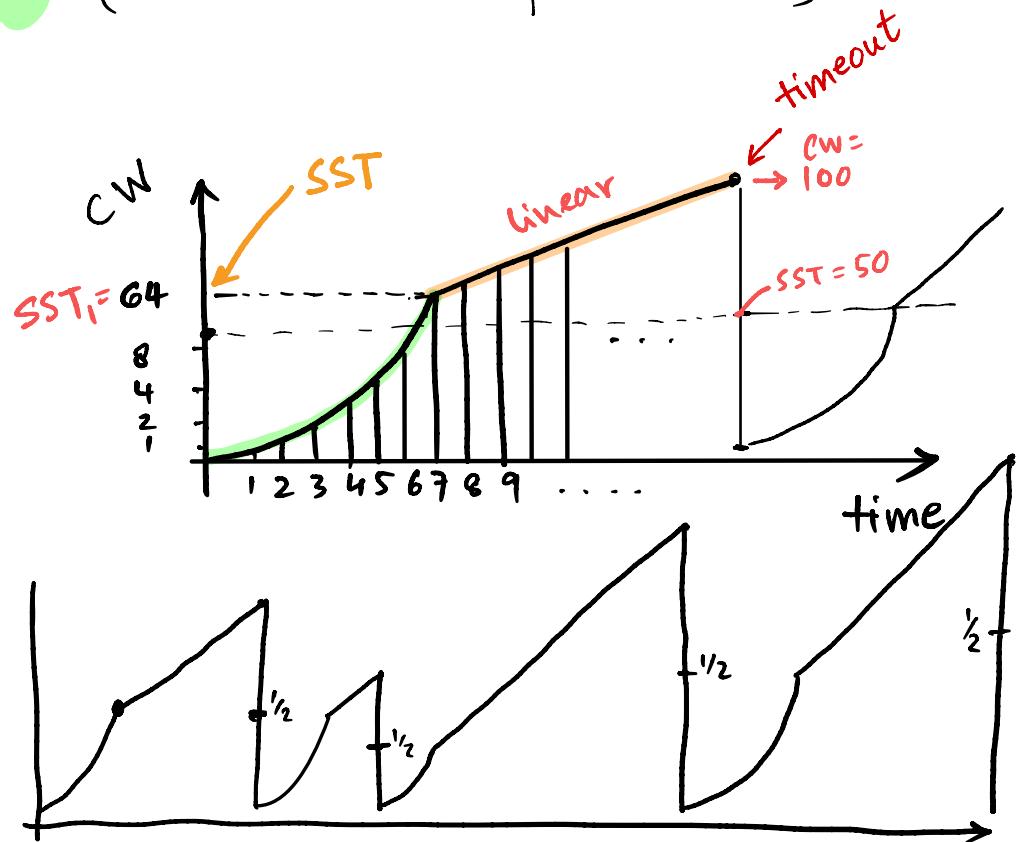
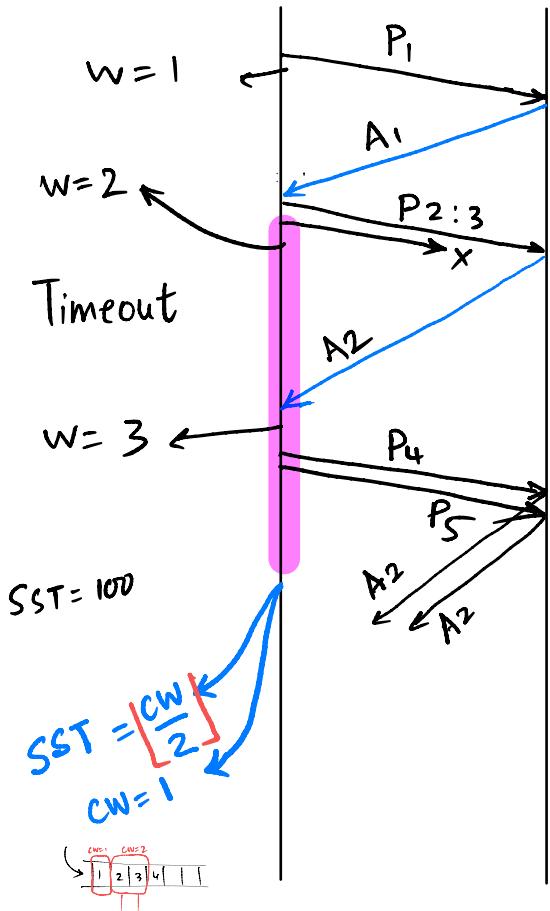
# Lecture 14

- Packet loss → AIMD + SST adaptation
- Fast Recovery
- Examples
- State Diagram
- saw tooth
- single timer

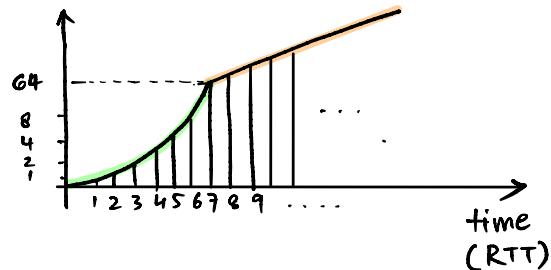
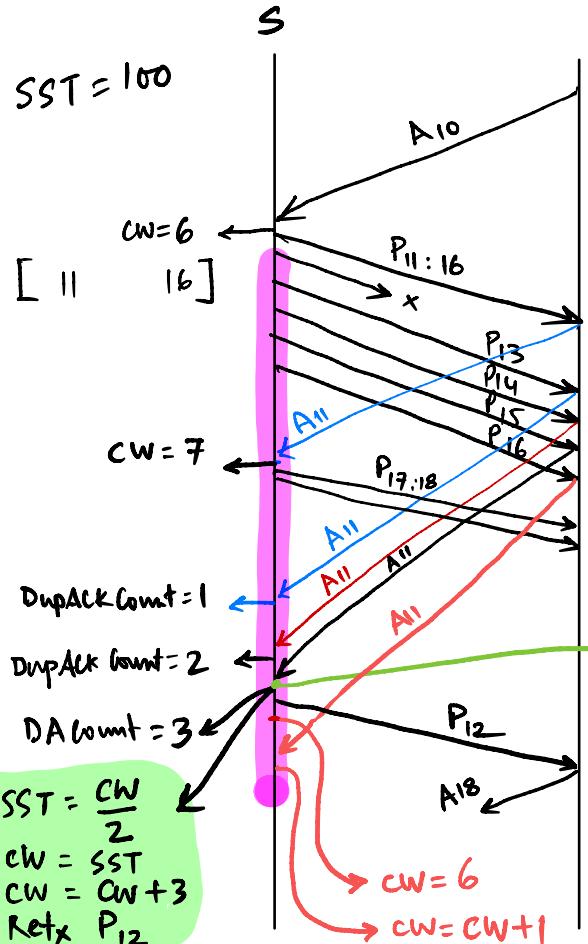
Packet Losses

: AIMD

( Additive increase Multiplicative decrease)



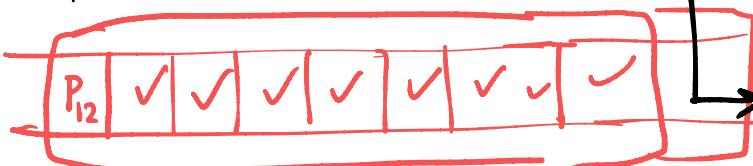
## Packet Losses : Fast Recovery



Cong - Avoidance  
 $CW \geq SST$

Upon receiving new ACK  
 ↳ get out of Fast Recovery

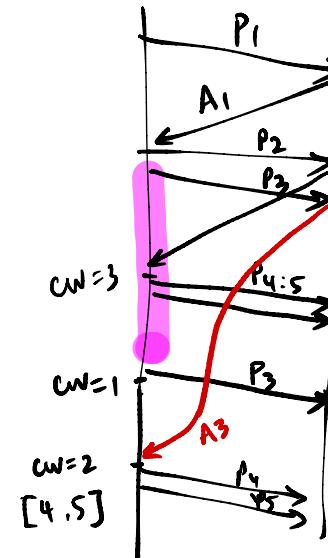
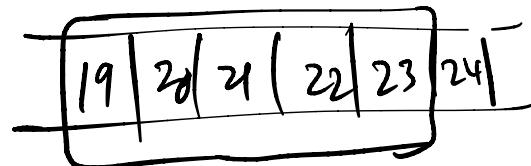
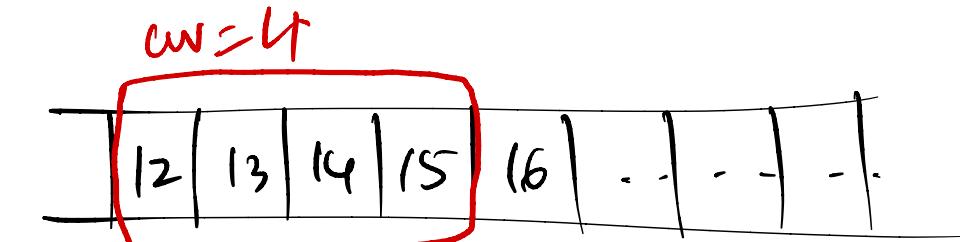
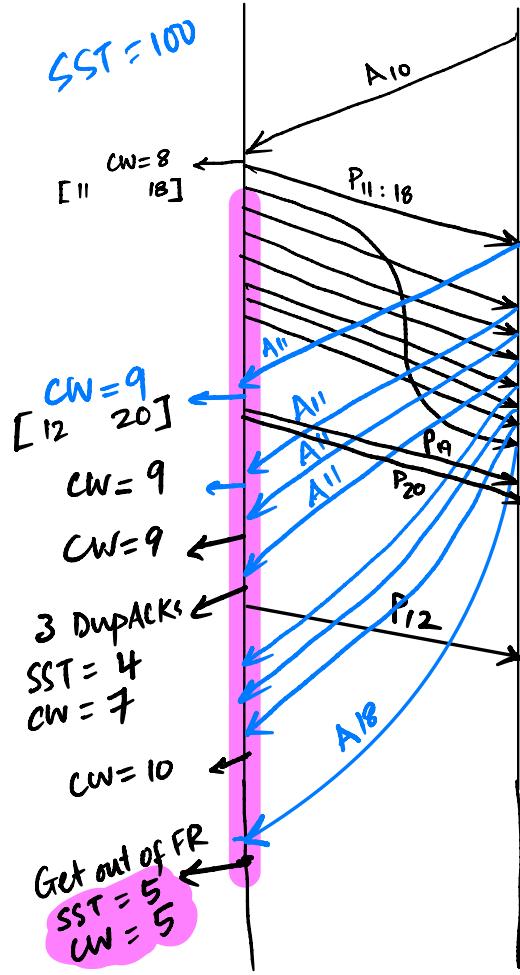
Fast Recovery !  
 Don't panic but take precautions -



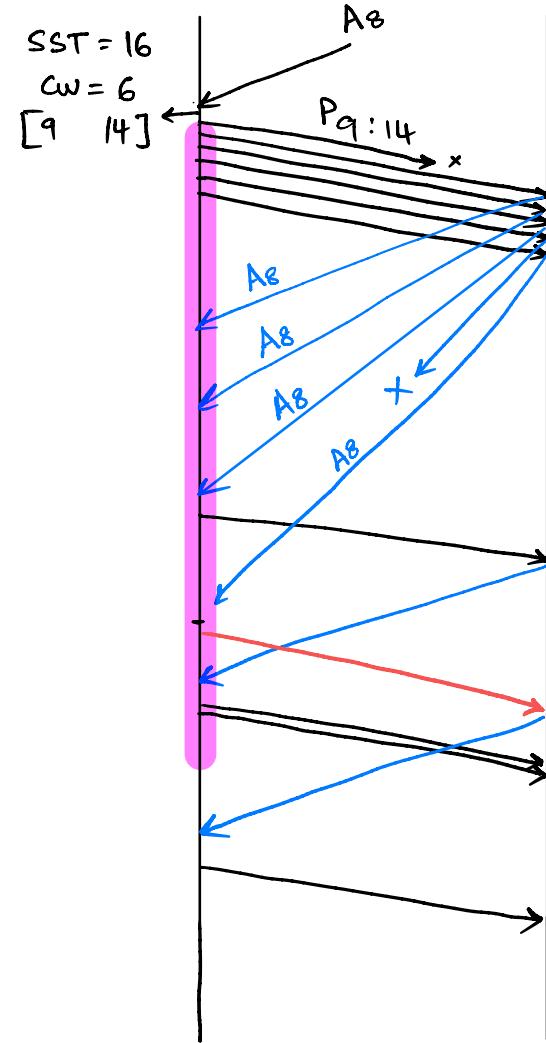
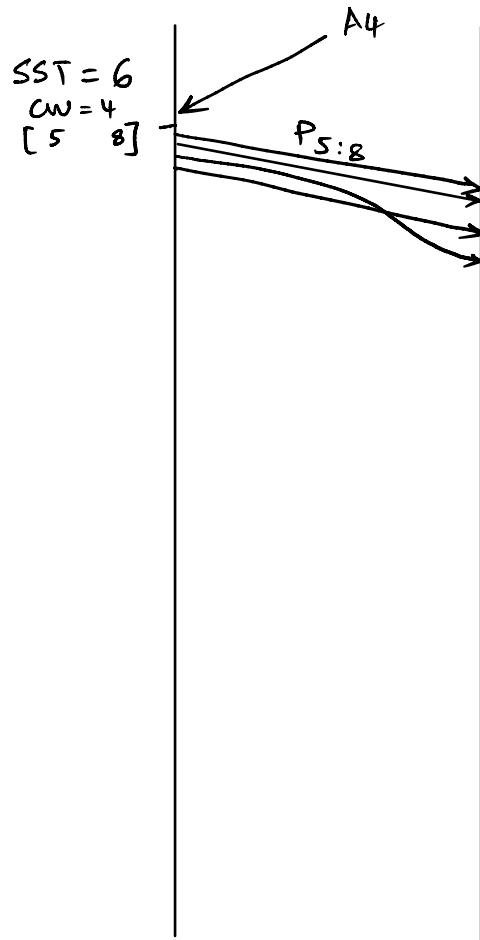
DupACK count = 0

CW = SST

## Fast Recovery : DupACKs keep coming



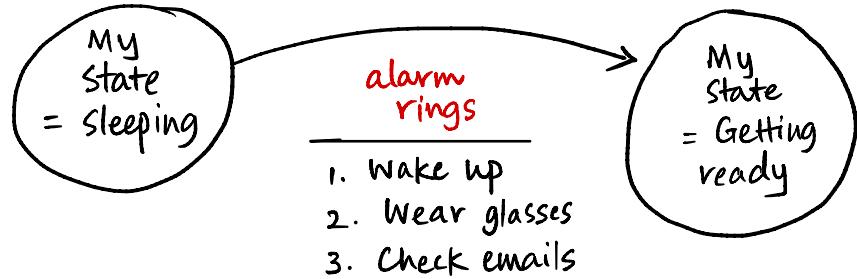
## More examples



## Protocol State Diagram



## State Transition Diagram



# TCP Protocol : State Diagram

**Socket connection**

new ACK

- $CW = CW + 1$
- Send pkts based on  $CW$
- $DupACK = 0$

- $CW = 1$
- $SST = 64$
- $DupACK = 0$

Slow Start

DupACK

- $DupACK++$

timeout

- $SST = \frac{CW}{2}$
- $CW = 1$
- $DupACK = 0$
- Retx  $CW$ -base packet

$DupACK == 3$

- $SST = CW/2$
- $CW = SST + 3$
- Retx  $CW$ -base
- Tx new pkt if allowed

$CW \geq SST$

X

timeout

- $SST = CW/2$
- $CW = 1$
- $DupACK = 0$
- Retx  $CW$ -base pkt

Fast Recovery

new ACK

- $CW = CW + \frac{1}{\lceil \frac{1}{CWS} \rceil}$

- $DupACK = 0$
- Transmit based on  $CW$

Congestion avoidance

$DupACK$

- $DupACK++$

new ACK

- $DupACK = 0$
- $CW = SST$
- Tx as allowed

$DupACK$

- $CW = CW + 1$
- Tx as allowed

$DupACK == 3$

- $SST = CW/2$
- $CW = SST + 3$
- Retx  $CW$ -base
- Transmit new pkt if allowed