



# CS120: Computer Networks

## **Lecture 5. ACK**

Zhice Yang

# The ACK Mechanism

- For ACK Transmitter: an acknowledgement (ACK for short) is a small control frame that a protocol sends back to its peer saying that it has received the earlier frame
- For ACK Receiver: The receipt of an acknowledgement indicates to the sender of the original frame that its frame was successfully delivered.



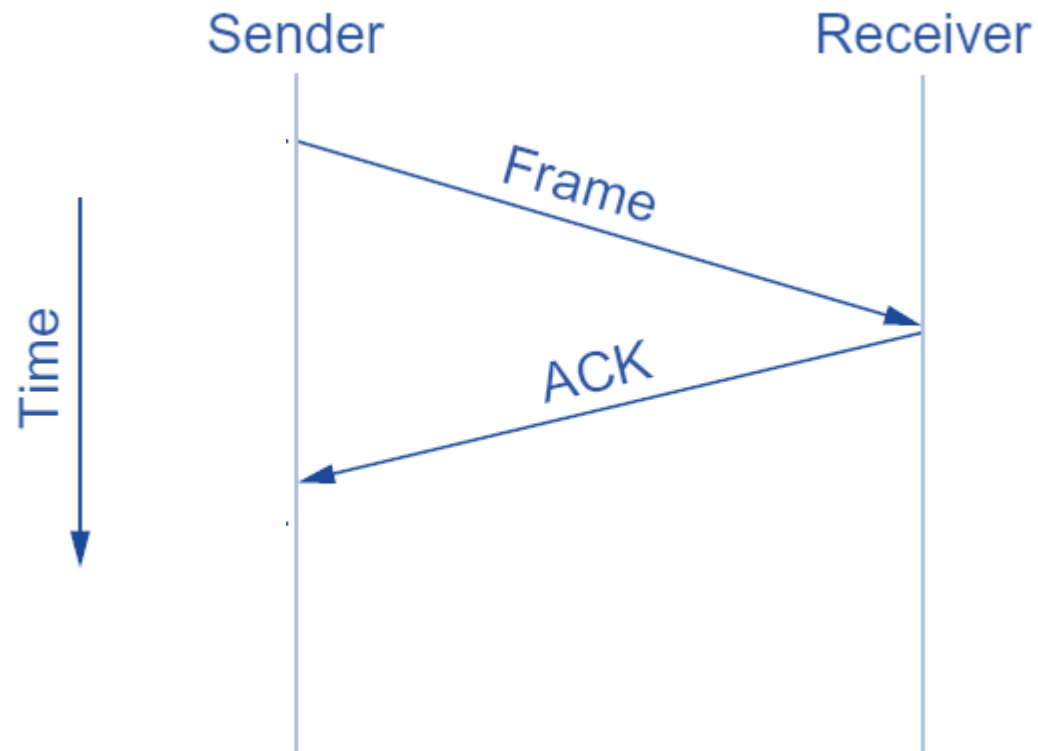
Have you heard that ?

# ACK Schemes

- Stop-and-Wait
- Sliding Window

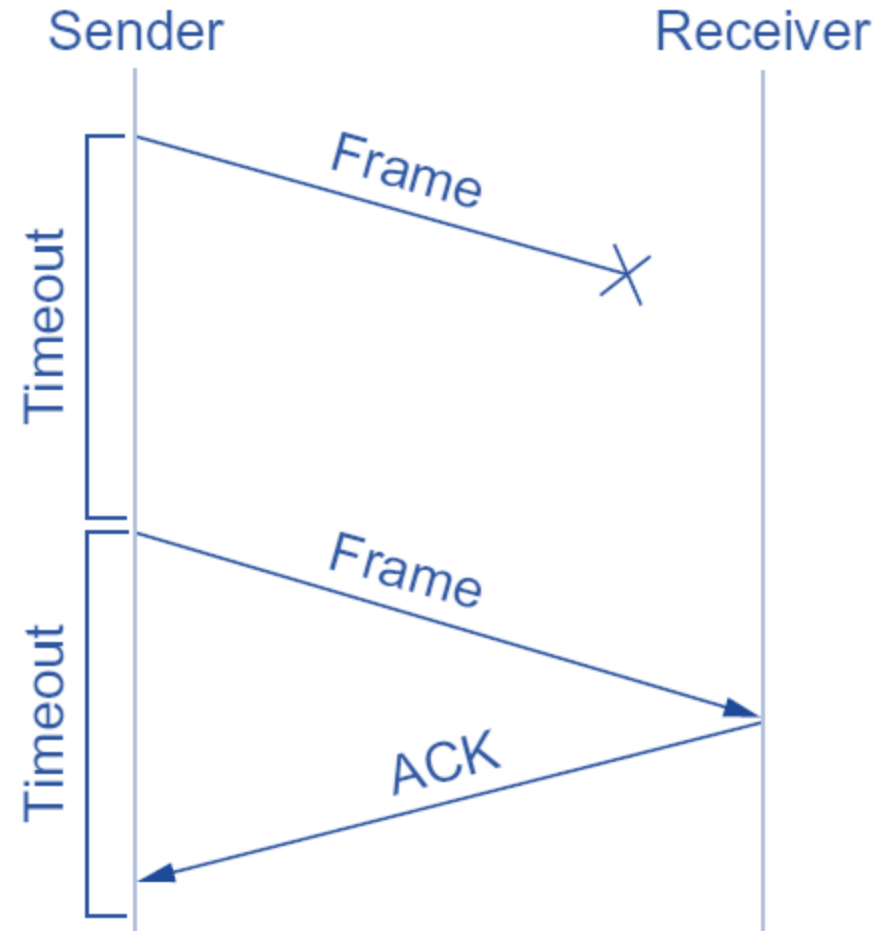
# Stop-and-Wait

- Case 0: (understanding the timeline)



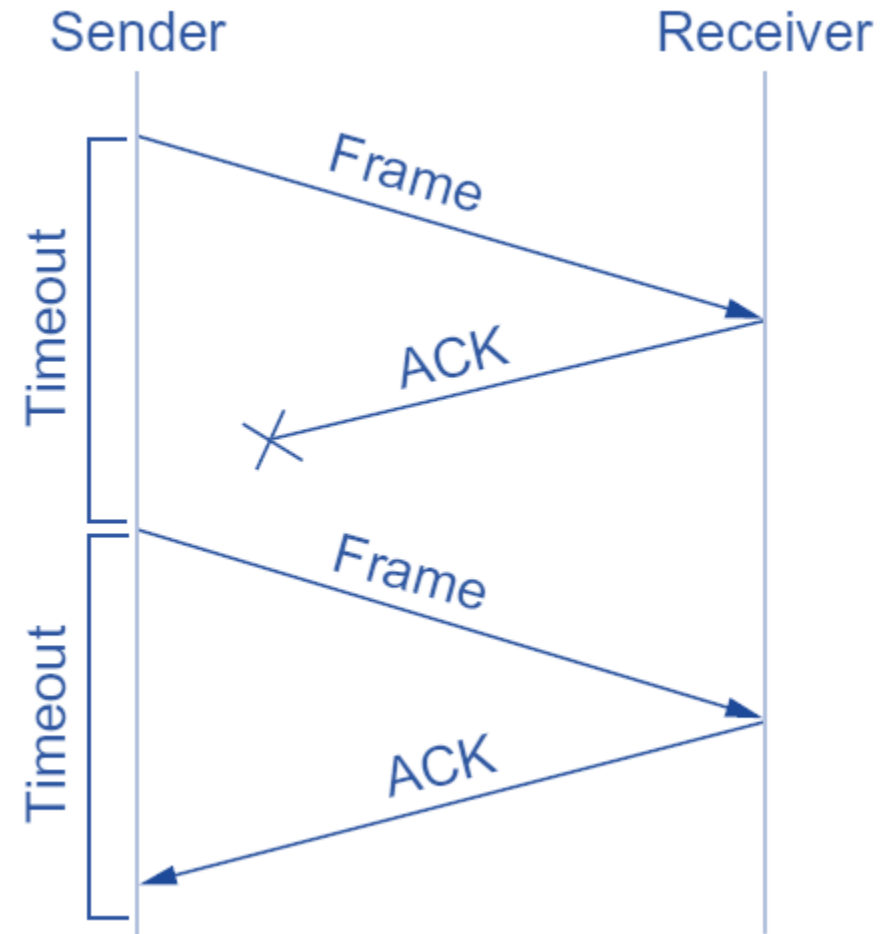
# Stop-and-Wait

- Case 1: Frame Loss
  - Sender time out
  - Sender retransmits



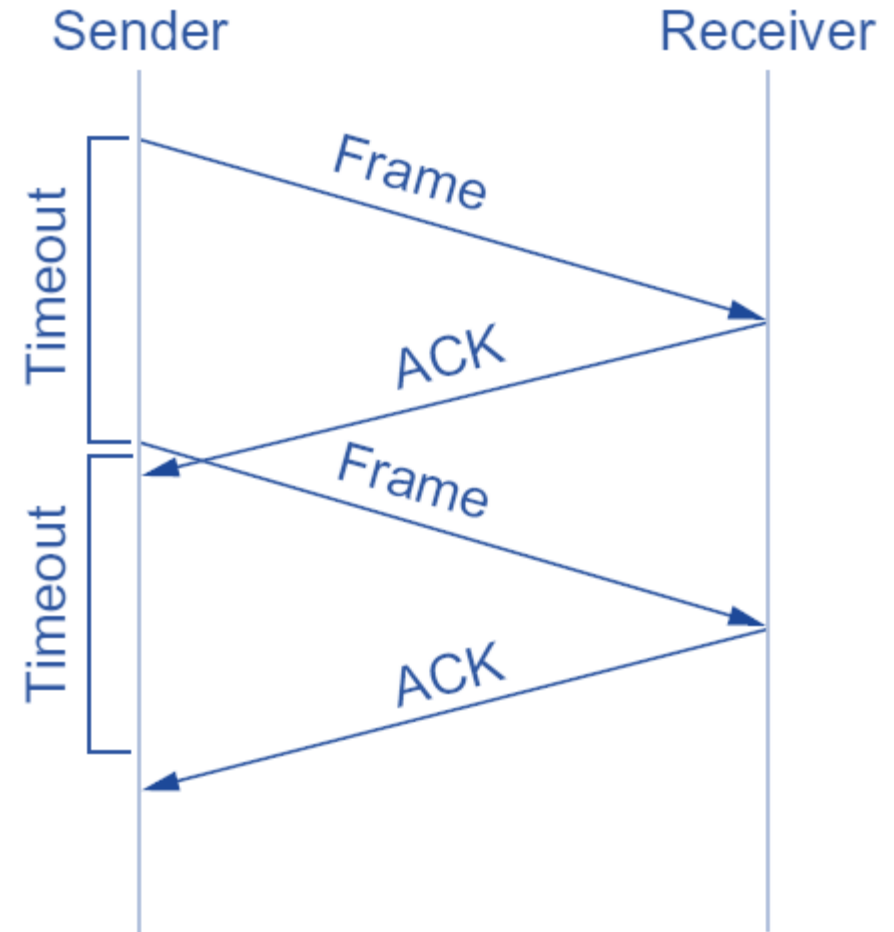
# Stop-and-Wait

- Case 2: ACK Loss
  - Sender time out
  - Sender retransmits



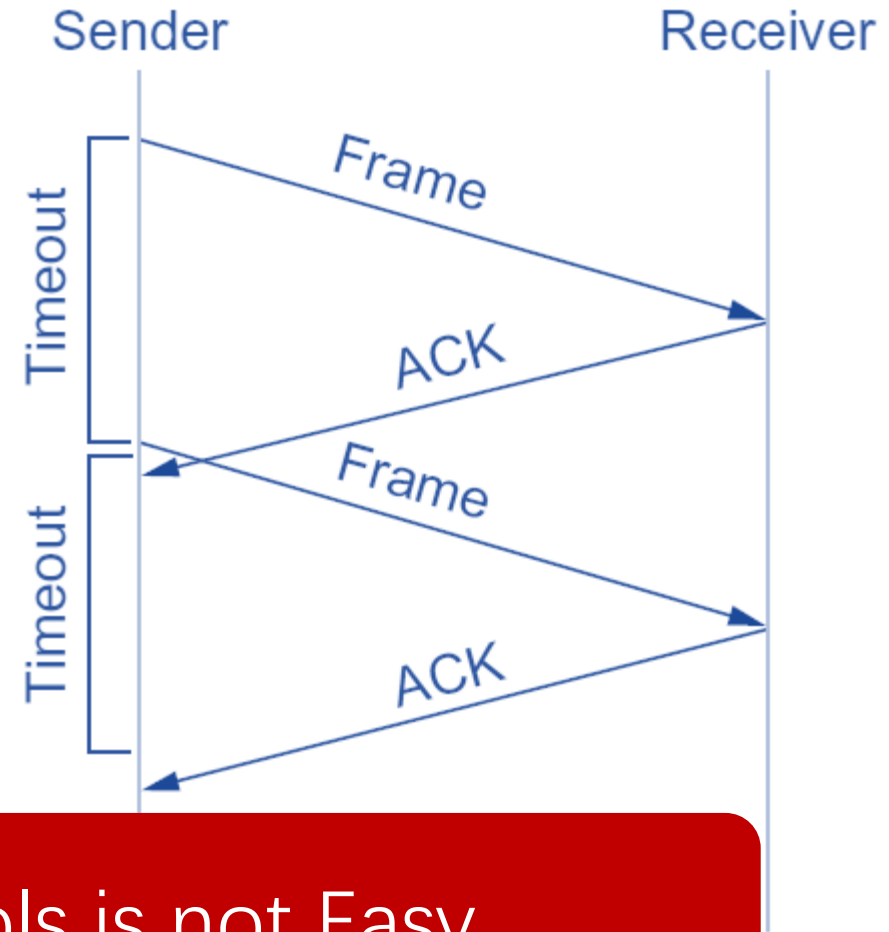
# Stop-and-Wait

- Case 3: ACK Late
  - Sender time out
  - Sender retransmits
  - Receiver may receive same frames



# Stop-and-Wait

- Case 3: ACK Late
  - Sender time out
  - Sender retransmits
  - Receiver may receive same frames
- Duplicated Frames
  - Receiver: Frame number
  - Sender: Timeout duration
    - How



Designing Protocols is not Easy

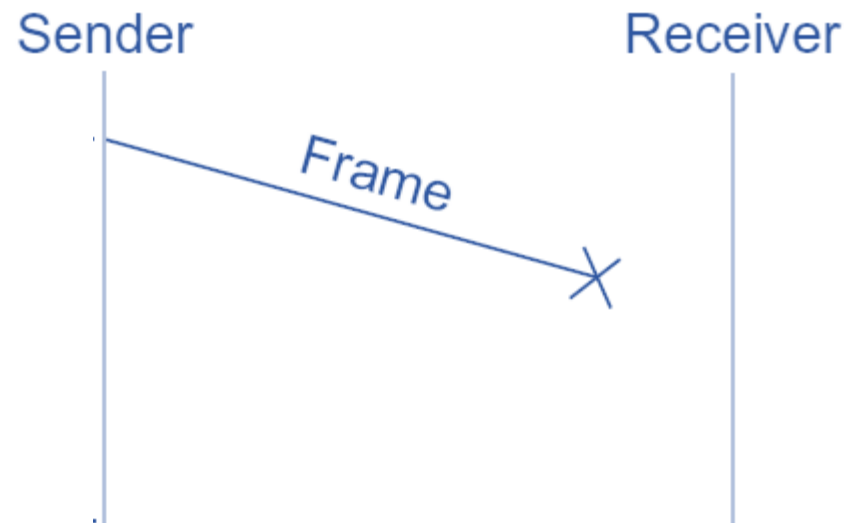


# Demo: Stop-and-Wait

- [http://www2.rad.com/networks/2004/sliding\\_window](http://www2.rad.com/networks/2004/sliding_window)

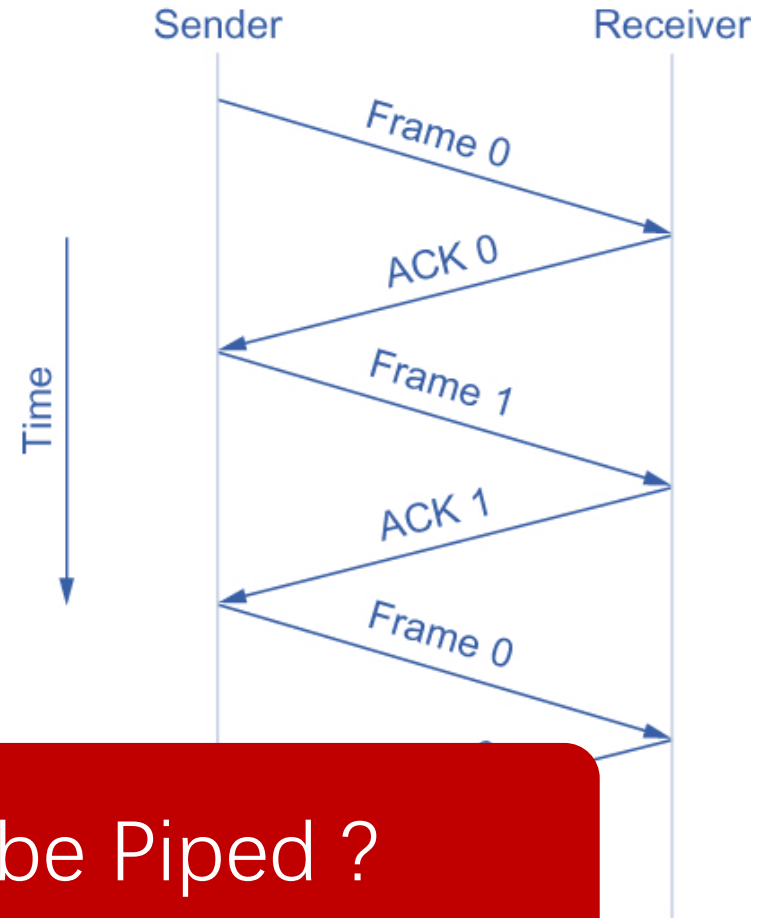
# How about NACK ?

- Negative ACK
  - Receiver send NACK to indicate frame loss through sequence number
  - If frame loss is after sender's idle
    - The receiver has no way to notice the loss



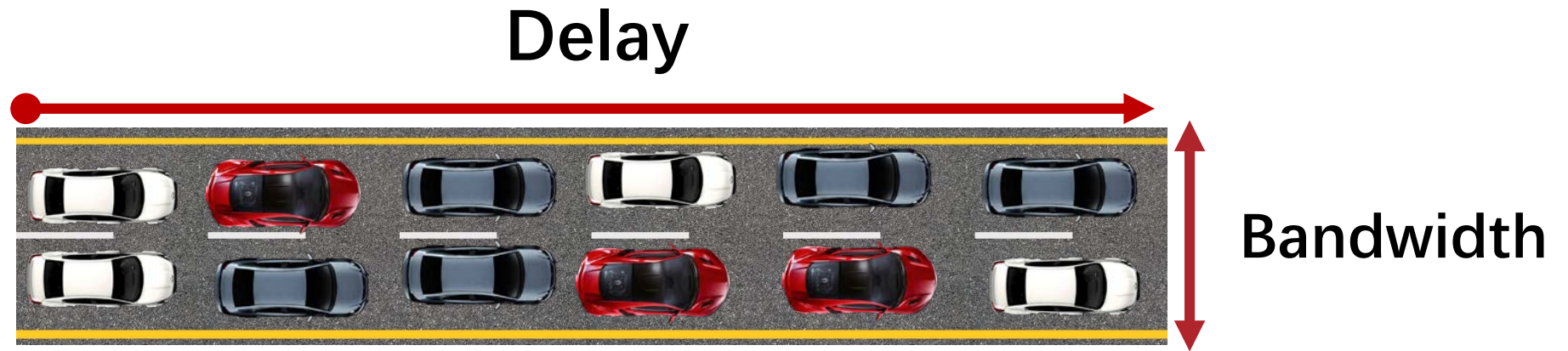
# Stop-and-Wait

- Efficiency Problem
  - 1.5Mbps bandwidth
  - 45ms RTT
  - 1KB frame
    - Effective Rate =  $1024 \times 8 / (1024 \times 8 / 1.5M + 45ms)$
- Solution
  - Pipeline



How Many Packets Can be Piped ?

Delay  $\times$  Bandwidth



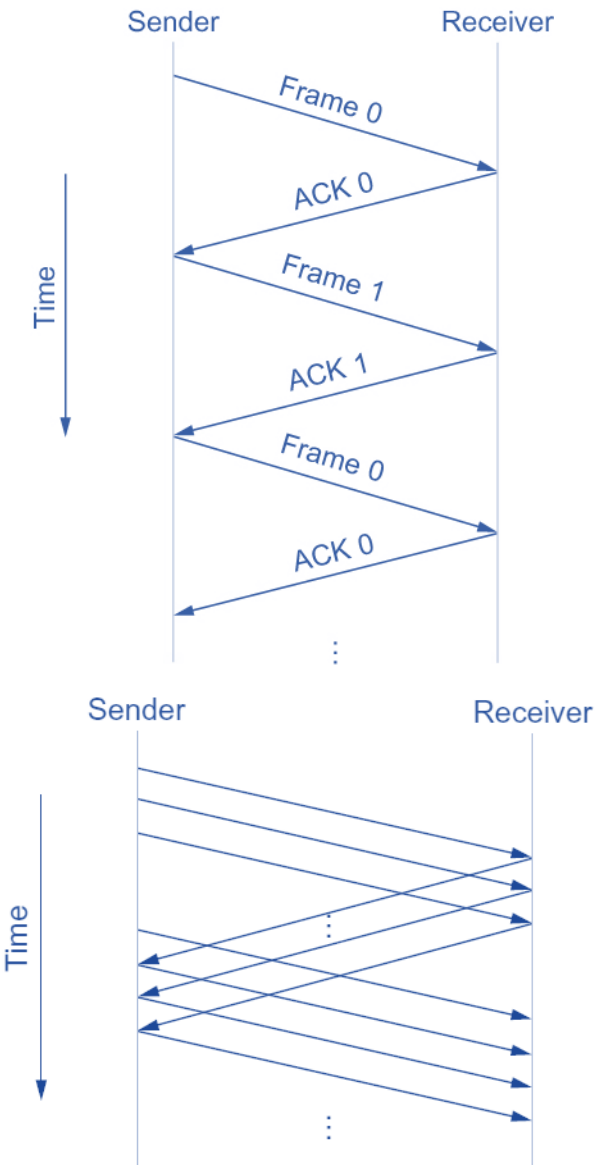
# Delay $\times$ Bandwidth

- Quantify the utilization of the link

Table 1.1 Sample Delay $\times$ Bandwidth Products				
Link type	Bandwidth (typical)	One-way distance (typical)	Round-trip delay	RTT $\times$ Bandwidth
Dial-up	56 kbps	10 km	87 $\mu$ s	5 bits
Wireless LAN	54 Mbps	50 m	0.33 $\mu$ s	18 bits
Satellite	45 Mbps	35,000 km	230 ms	10 Mb
Cross-country fiber	10 Gbps	4,000 km	40 ms	400 Mb

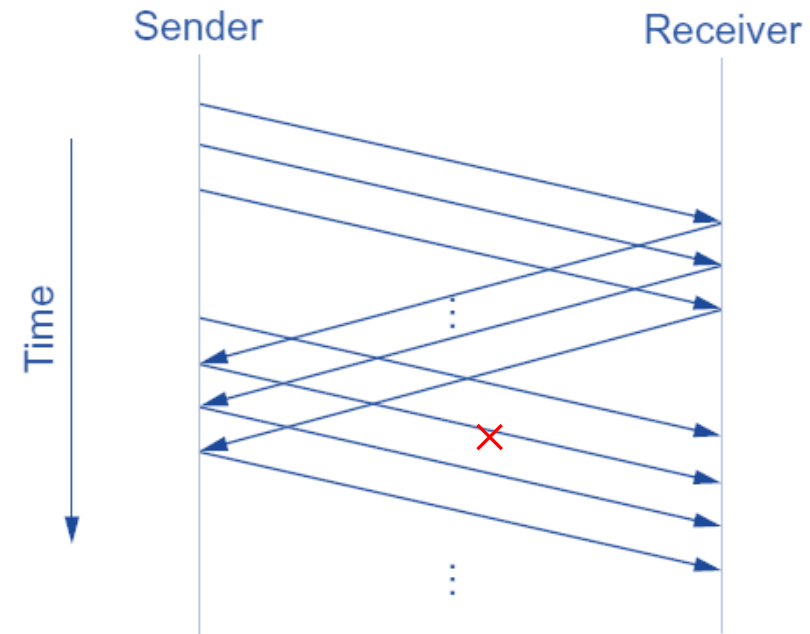
# Delay $\times$ Bandwidth

- Efficiency Problem
  - 1.5Mbps bandwidth
  - 45ms RTT
  - 1KBbps
    - Effective Rate =  $1024 \times 8 / (1024 \times 8 / 1.5\text{M} + 45\text{ms})$
- Solution
  - Pipeline
  - Full pipe
    - $1.5\text{Mbps} \times 45\text{ms} / 1\text{kB} = 8$  frames in flight



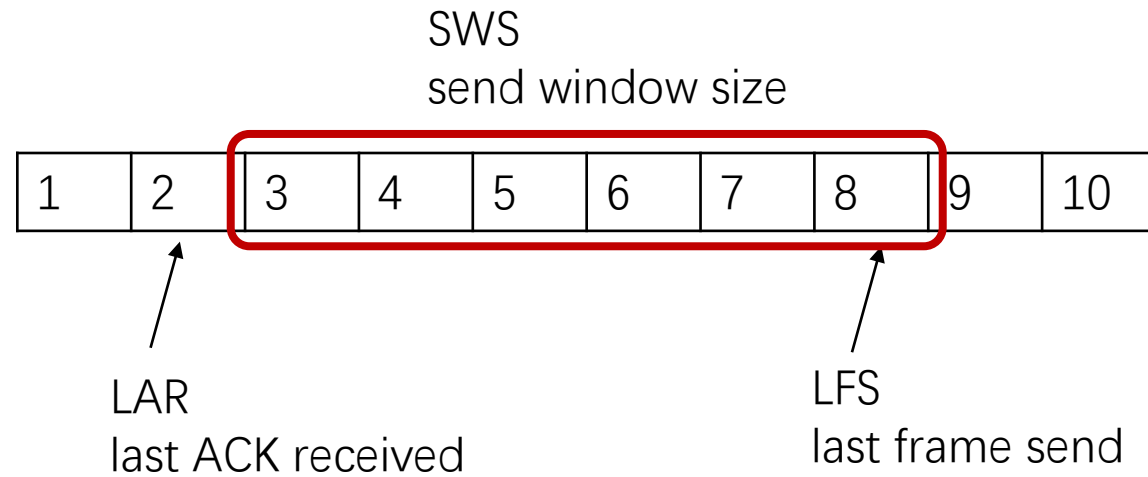
# Delay $\times$ Bandwidth

- Sender Buffer
  - Retransmit
- Receiver Buffer
  - Handle out-of-order frames

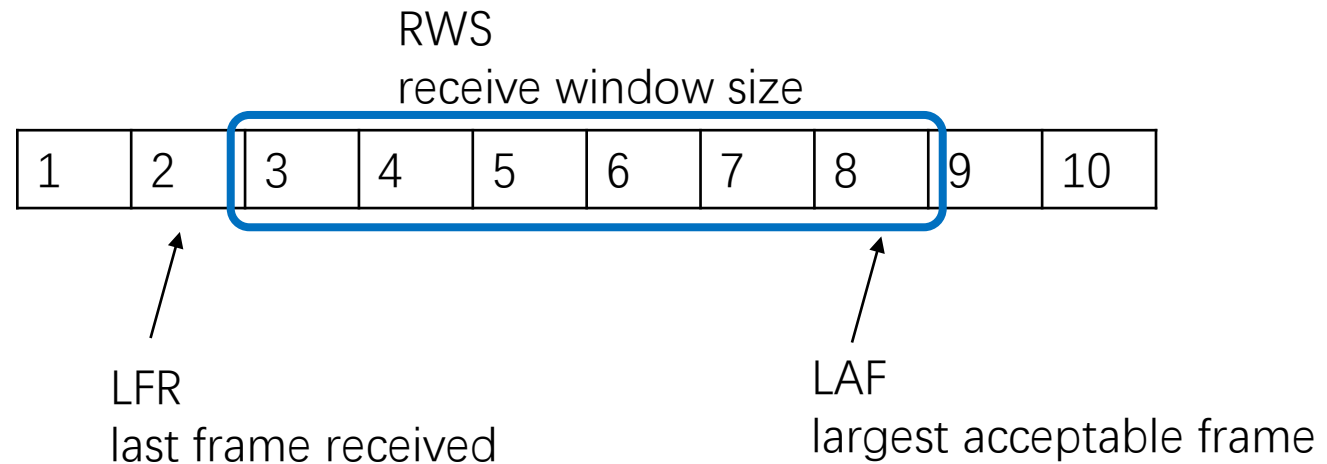


# Sliding Window

Sender Buffer:



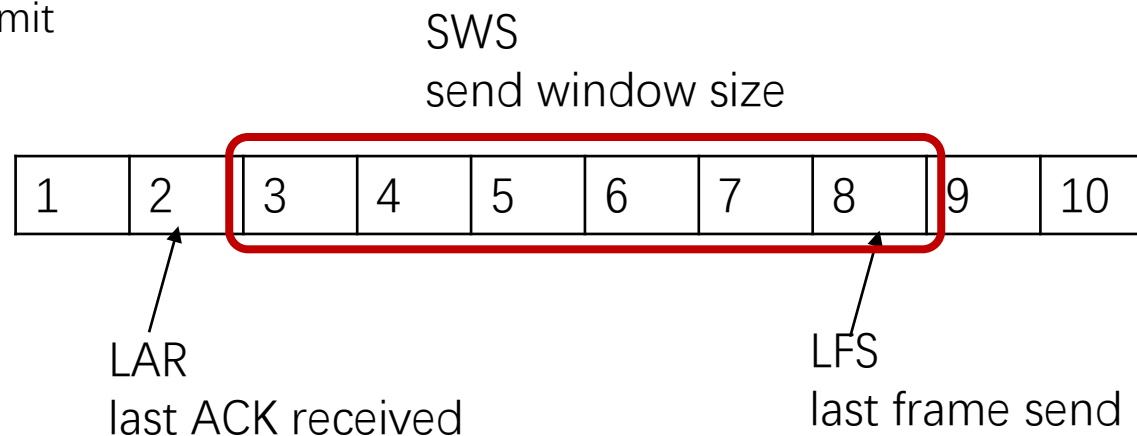
Receiver Buffer:





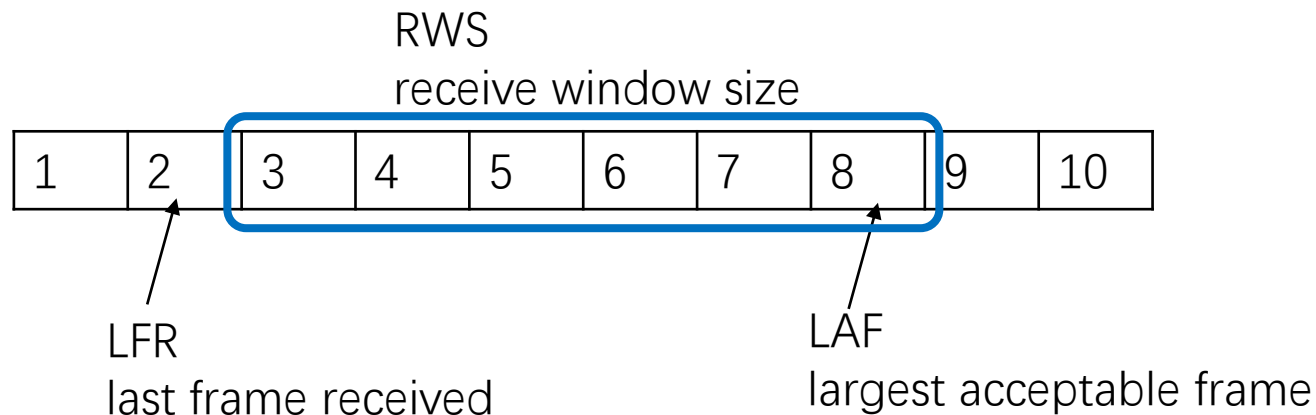
# Sliding Window

- Sender Protocol
  - Always maintain  $LFS - LAR \leq SWS$
  - When an ACK with sequence number #SeqNum arrives
    - If  $\#SeqNum \leq LAR$  or  $\#SeqNum > LFS$ 
      - No action
    - If  $LFR < \#SeqNum \leq LAF$ 
      - Move LAR to #SeqNum
  - Associate a timer with each frame sender transmits
    - If timeout
      - Retransmit



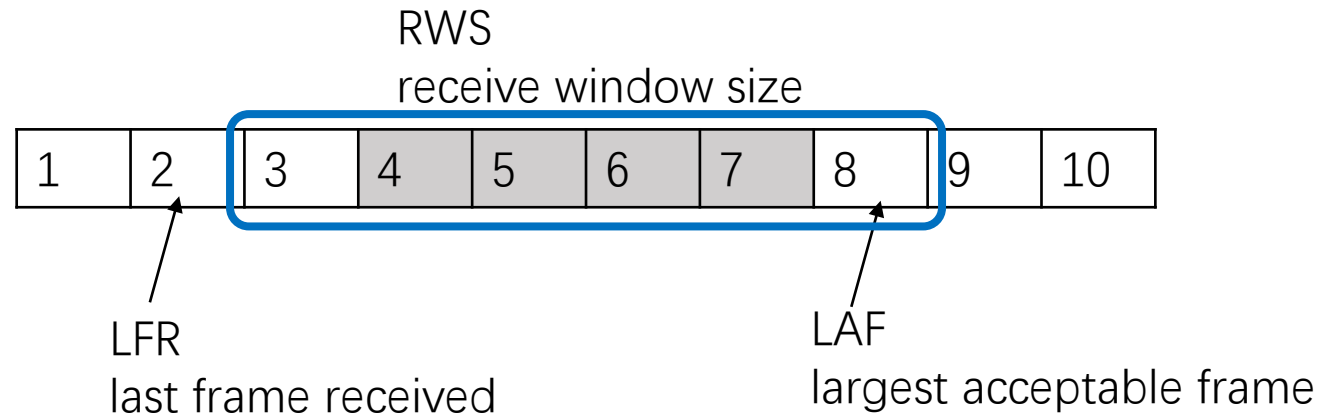
# Sliding Window

- Receiver Protocol
  - Always maintain  $LAF - LFR \leq RWS$
  - When a frame with sequence number  $\#SeqNum$  arrives
    - If  $\#SeqNum \leq LFR$  or  $\#SeqNum > LAF$ 
      - Discard frame
    - If  $LFR < \#SeqNum \leq LAF$ 
      - Accept frame
      - Accumulative Ack

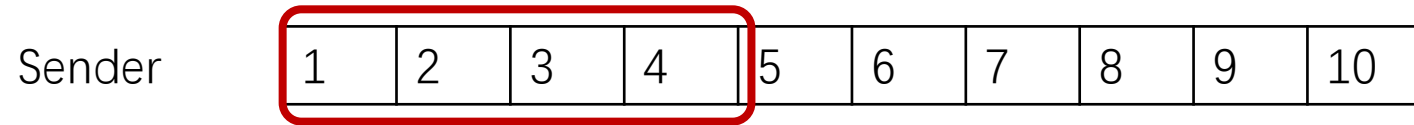


# Accumulative Ack

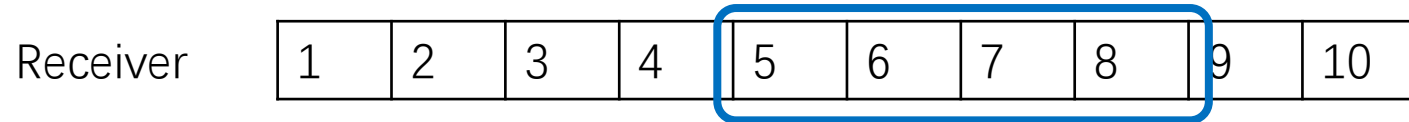
- If frame #3 is received
  - Ack #7, move LFR to 7, move LAF to 13
- If frame #8 is received
  - Opt1: no action
  - Opt2: Ack #2



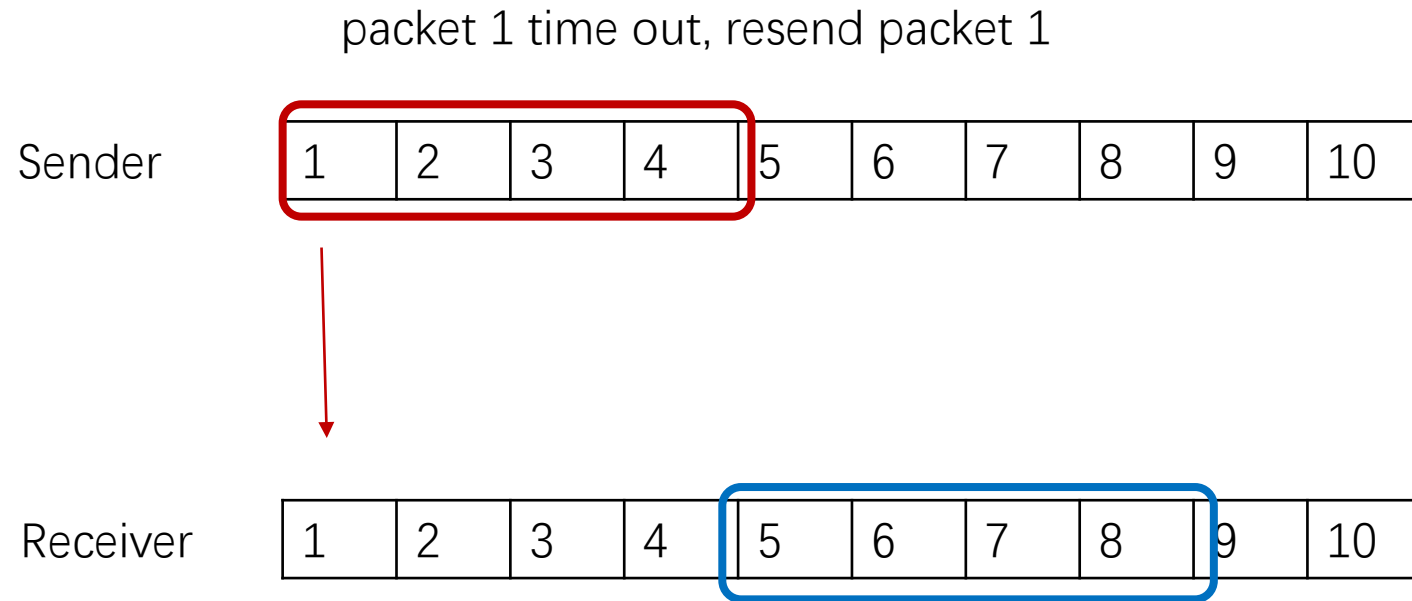
# ACKs Lost



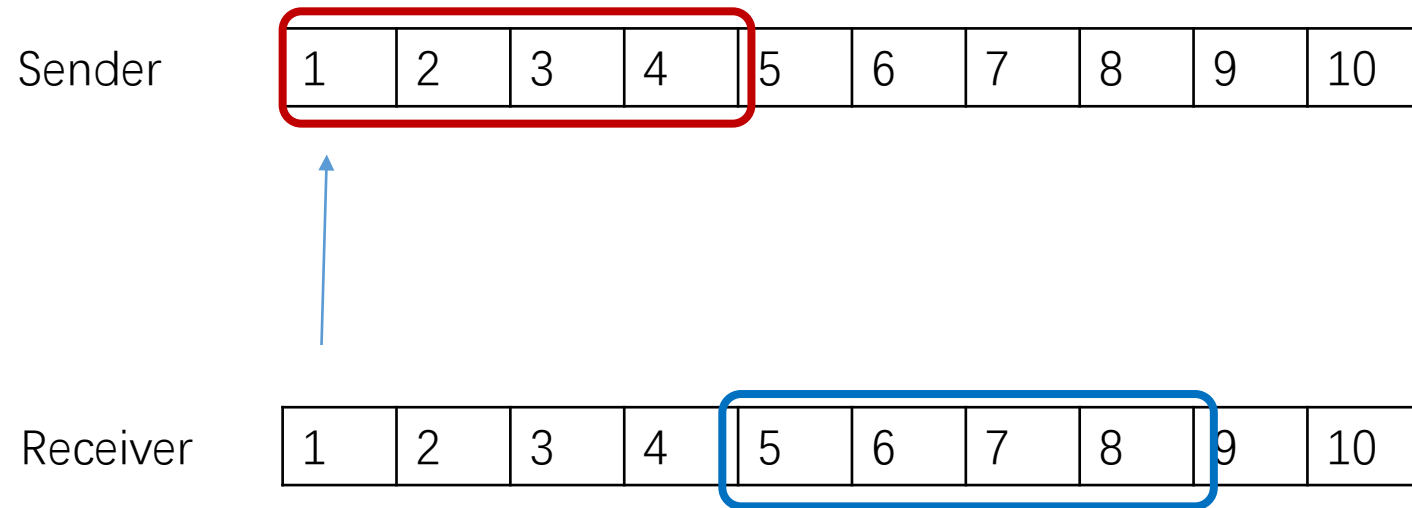
ack for 1,2,3,4, lost



# ACKs Lost



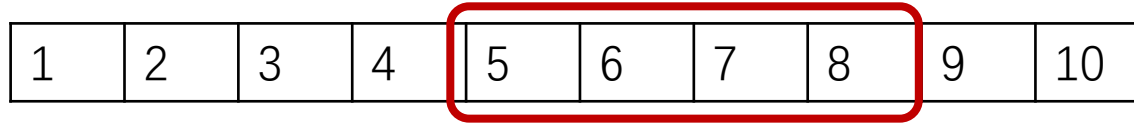
# ACKs Lost



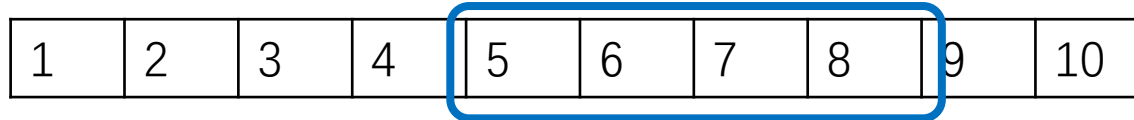
# ACKs Lost

ack 4 received, slide window to 5

Sender

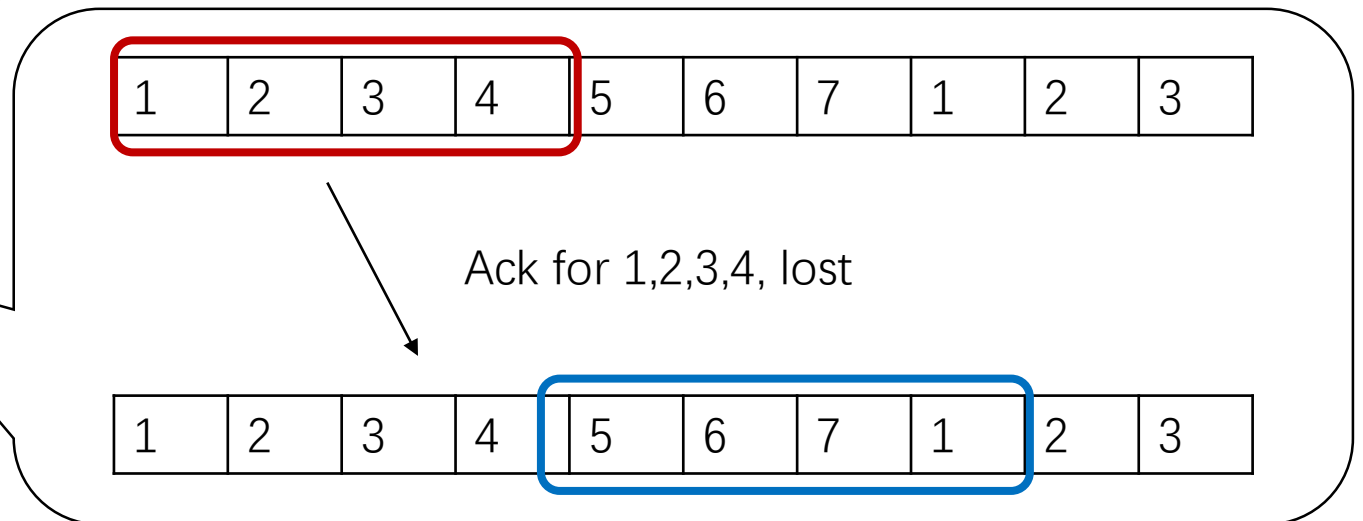
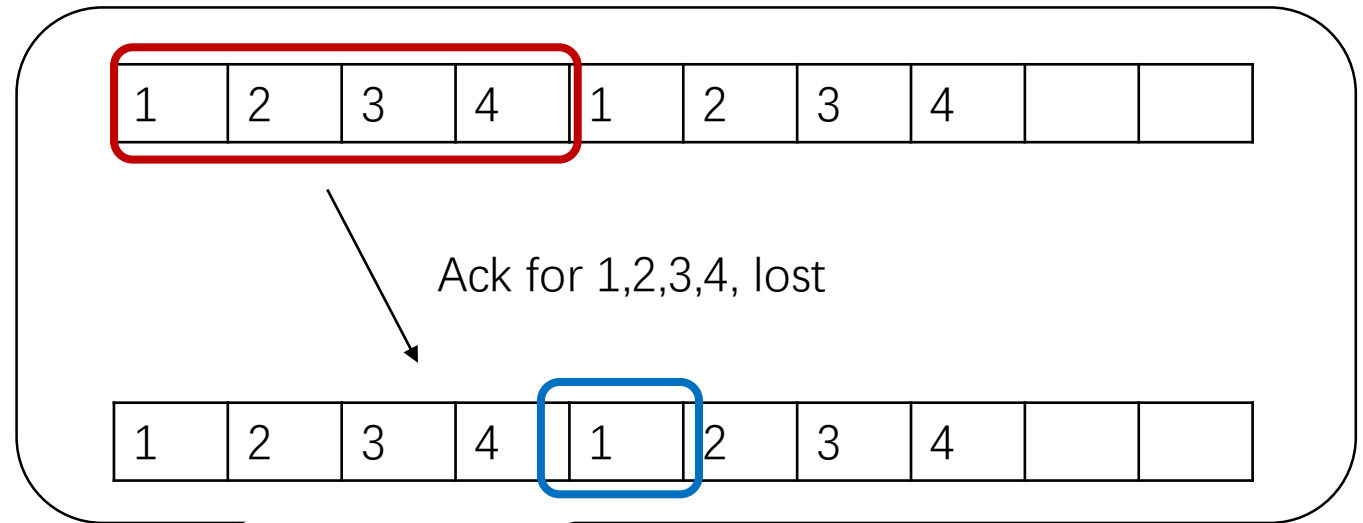


Receiver



# Sliding Window

- Determine Window Size
  - SWS: Pipeline depth
    - Delay  $\times$  Bandwidth
  - RWS: Flow control
- Determine SeqNum range
  - if RWS = 1
    - $\text{MaxSeqNum} \geq \text{SWS} + 1$
  - if SWS = RWS
    - $\text{MaxSeqNum} \geq 2 * \text{SWS}$





# Sliding Window

- Functionalities
  - Pipelined ACK
  - Preserving Frame Order
  - Flow control
    - Adjust the rate of the sender at the receiver side

# Demo

- Sliding Window code in TCP  
/net/ipv4/
- Change Sliding Window
  - Show current congestion control scheme  
`cat /proc/sys/net/ipv4/tcp_congestion_control`
  - Show/change available congestion control scheme  
`sysctl net.ipv4.tcp_available_congestion_control[=XX]`
- [http://www2.rad.com/networks/2004/sliding\\_window](http://www2.rad.com/networks/2004/sliding_window)

# Reference

- Textbook 1.5.2
- Textbook 2.5