CS120: Computer Networks

Lecture 5. ACK

Zhice Yang

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.

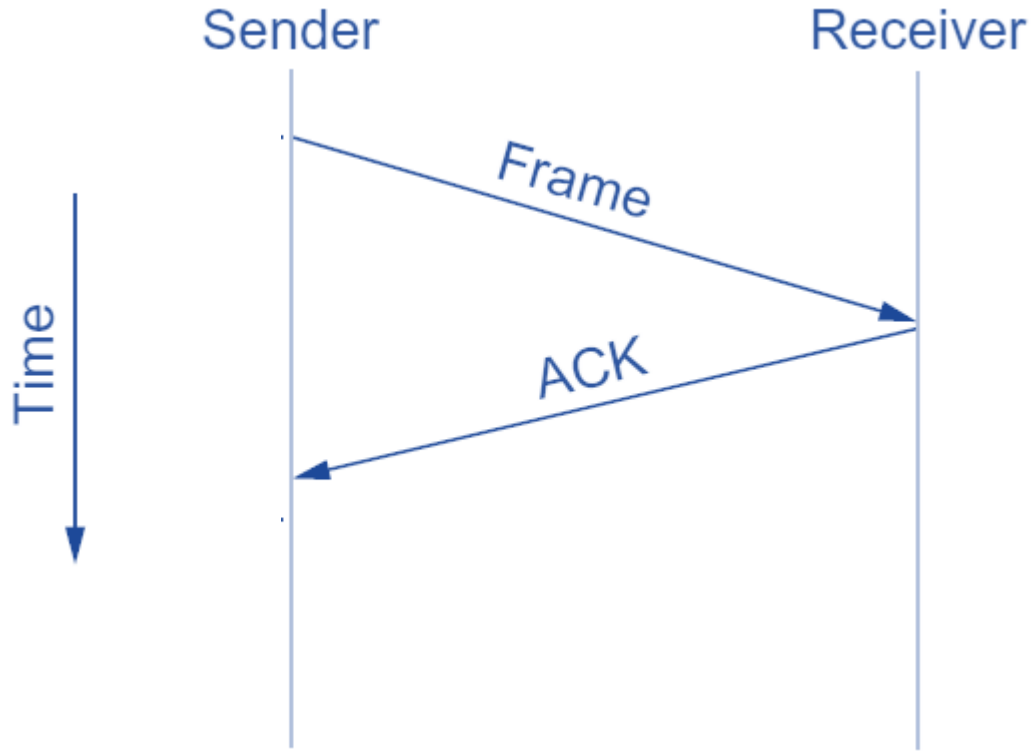
Have you heard that ?

ACK Schemes

- Stop-and-Wait
- Sliding Window

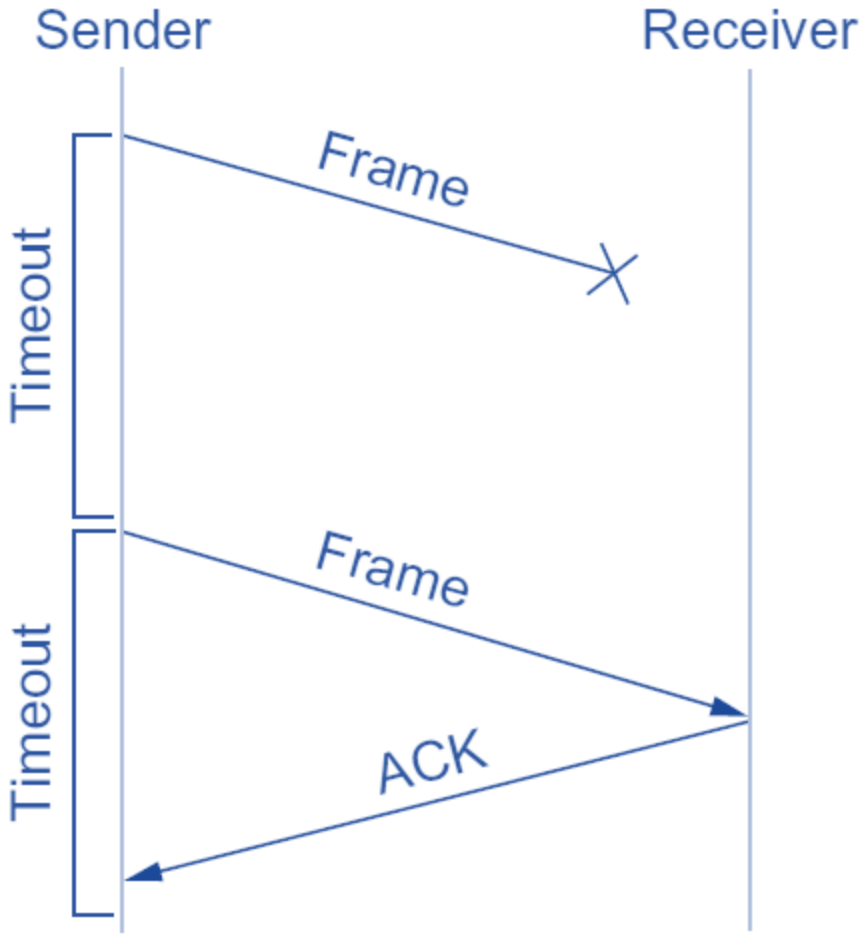
Stop-and-Wait

- Case 0: (understanding the timeline)



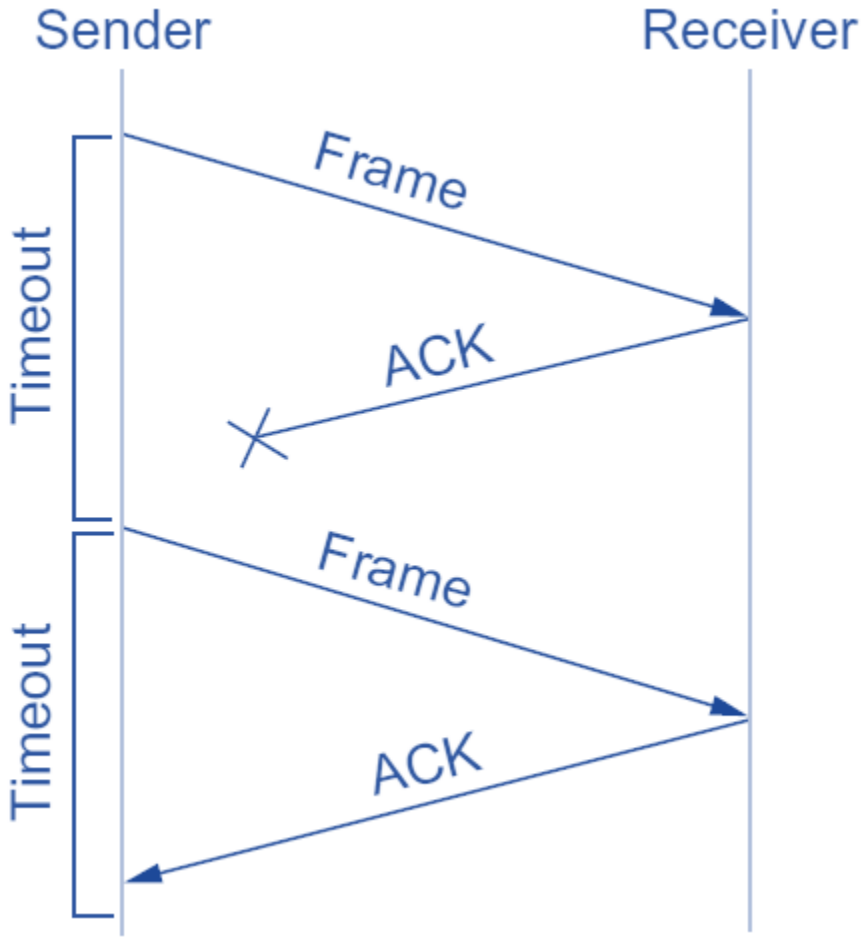
Stop-and-Wait

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



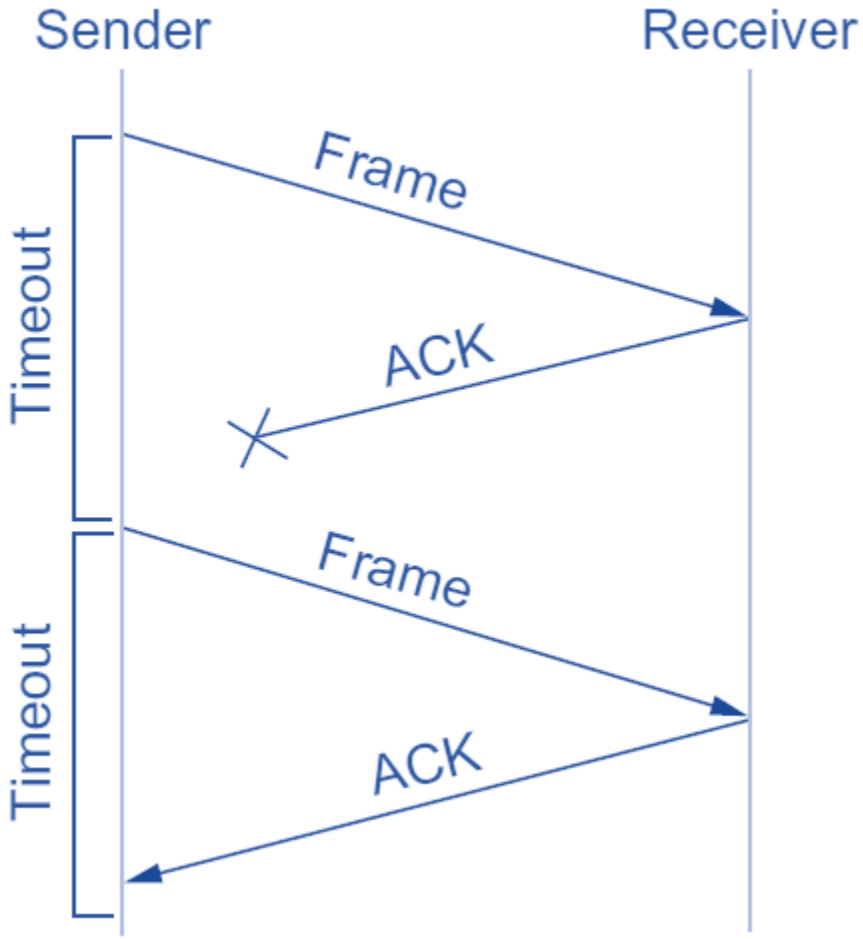
Stop-and-Wait

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



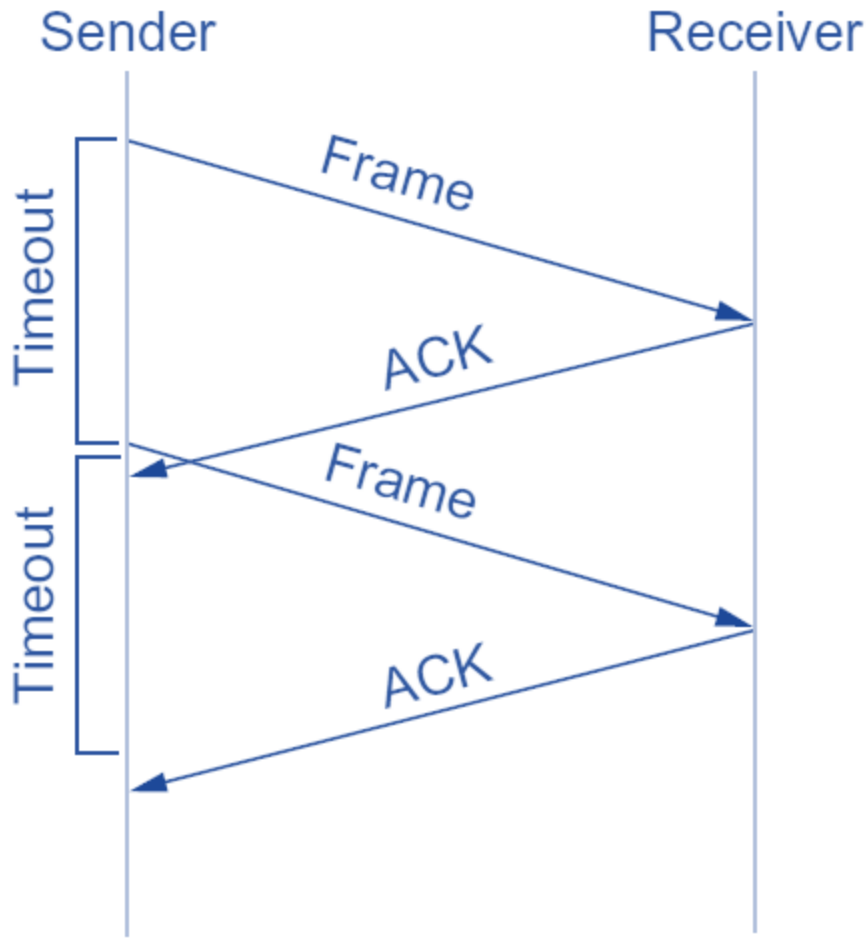
Stop-and-Wait

- Case 2: ACK Loss
  - Sender time out
  - Sender retransmits
- Duplicated Frames
  - Solution: frame number



Stop-and-Wait

- Case 3: ACK Late
  - Sender time out
  - Sender retransmits

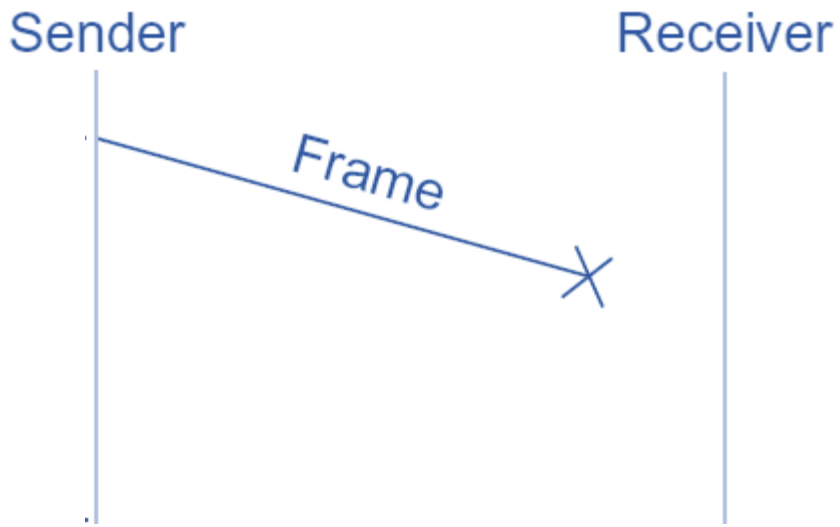


Demo: Stop-and-Wait

- https://www2.tkn.tu-berlin.de/teaching/rn/animations/gbn_sr/

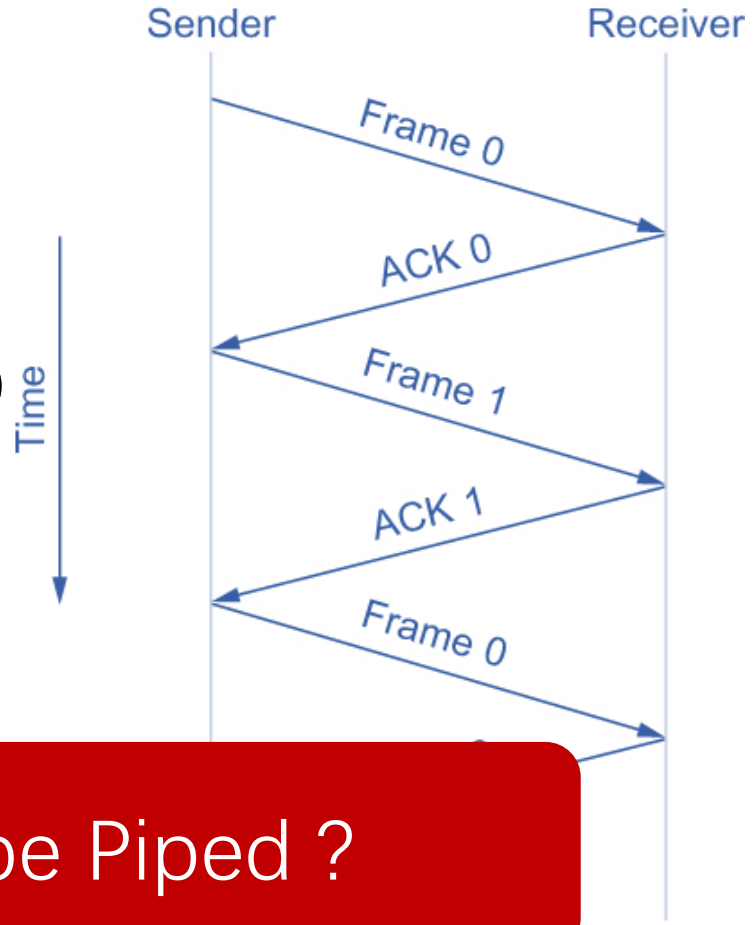
How about NACK ?

- Negative ACK
  - Receiver sends NACK to indicate frame loss, e.g., indicating sequence number
  - If frame loss is after sender's idle
    - The receiver has no way to notice the loss



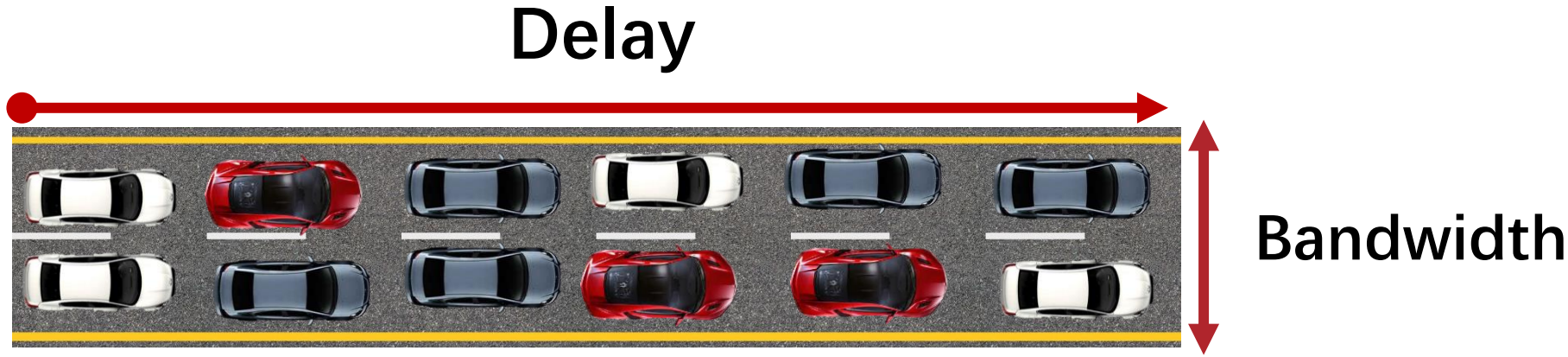
Stop-and-Wait

- Efficiency Problem
  - 1.5Mbps bandwidth
  - 45ms RTT
  - 1KB frame
    - Effective Rate = 1024*8/(1024*8/1.5Mbps+45ms)
      about 160kbps
- Solution
  - Pipeline



How Many Frames Can be Piped ?

Delay × Bandwidth



Delay × Bandwidth

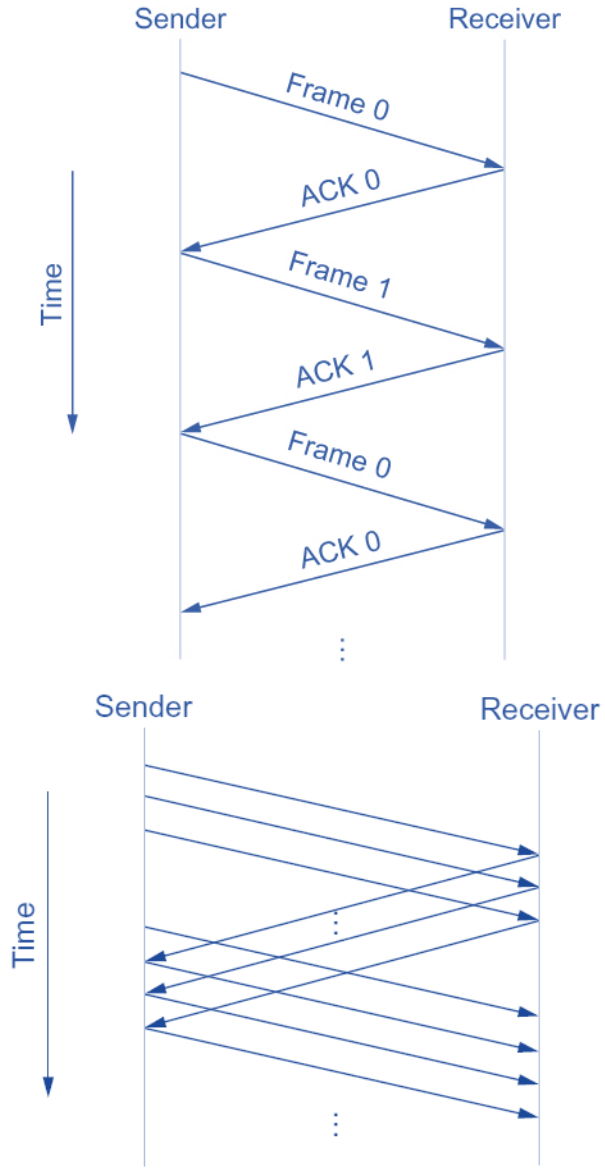
- Use to judge whether the link has been fully utilized

Table 1.1 Sample Delay × Bandwidth Products

Link type	Bandwidth (typical)	One-way distance (typical)	Round-trip delay	RTT × Bandwidth
Dial-up	56 kbps	10 km	87 μs	5 bits
Wireless LAN	54 Mbps	50 m	0.33 μs	18 bits
Satellite	45 Mbps	35,000 km	230 ms	10 Mb
Cross-country fiber	10 Gbps	4,000 km	40 ms	400 Mb

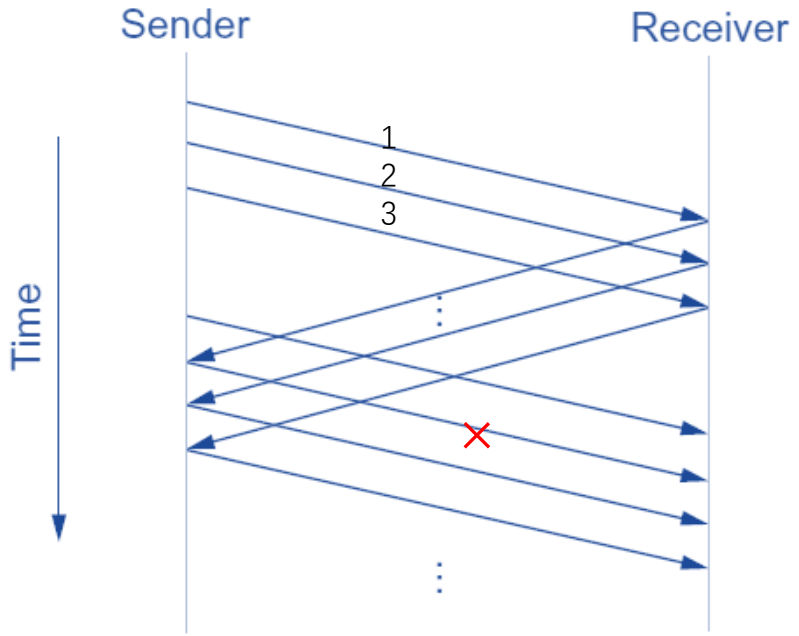
Delay × Bandwidth

- Efficiency Problem
  - 1.5Mbps bandwidth
  - 45ms RTT
  - 1KB Frame
    - Effective Rate = 160kbps
- Solution
  - Pipeline
  - Full pipe situation:
    - 1.5Mbps*45ms/1KB = 8 frames in flight



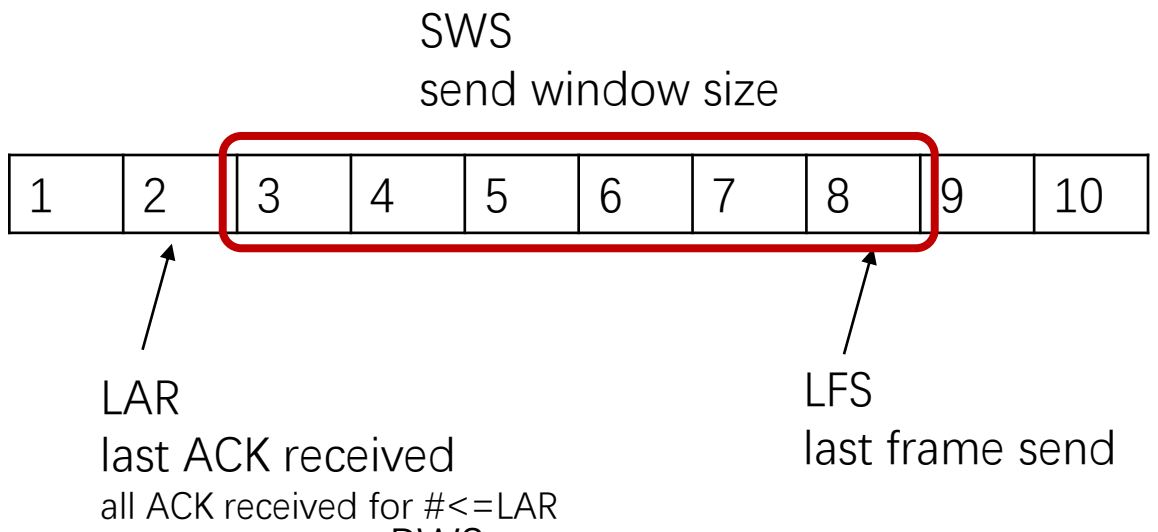
Sliding Window – pipelined transmitting

- Sender Buffer
  - Retransmit
- Receiver Buffer
  - Handle out-of-order frames
- Sequence Number
  - Frame ID

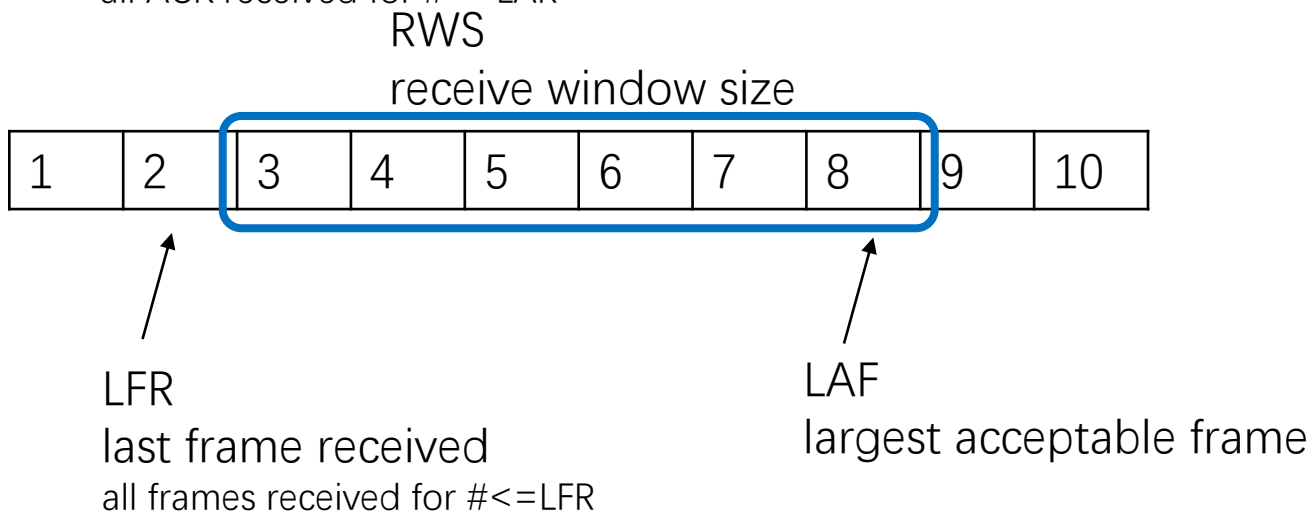


Sliding Window

Sender Buffer:

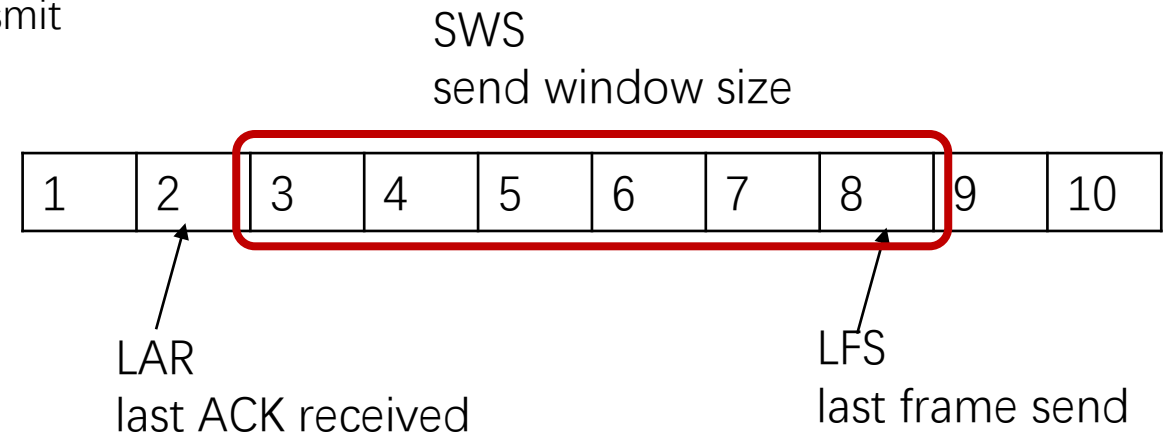


Receiver Buffer:



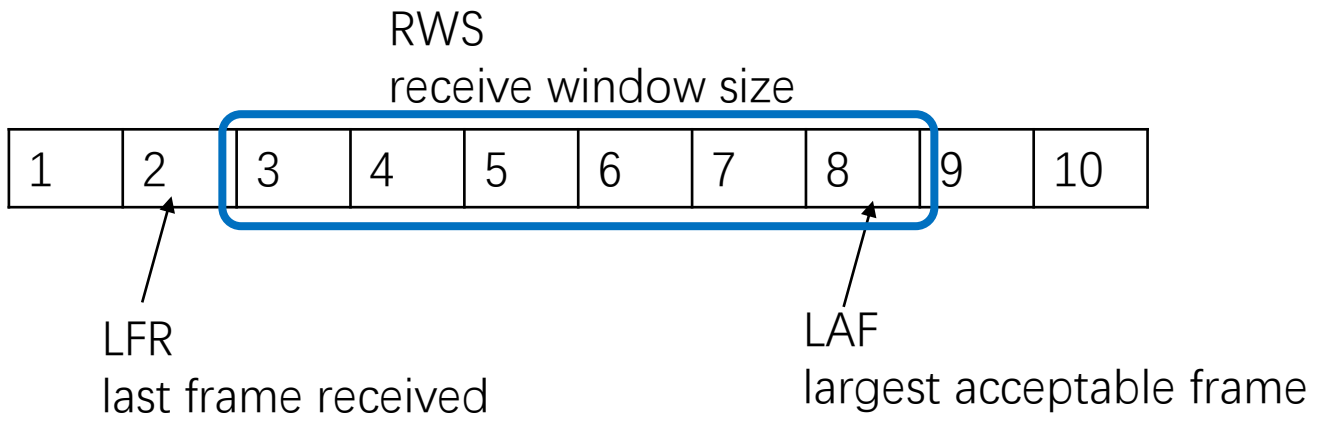
Sliding Window

- Sender Protocol
  - Always maintain LFS – LAR <= SWS
  - When an ACK with sequence number #SeqNum arrives
    - If #SeqNum ⩽ LAR or #SeqNum > LFS
      - No action
    - If LFR < #SeqNum ⩽ LAF
      - Move LAR to #SeqNum, increase LFS to send new frame
  - Associate a timer with each frame sender transmits
    - If timeout
      - Retransmit



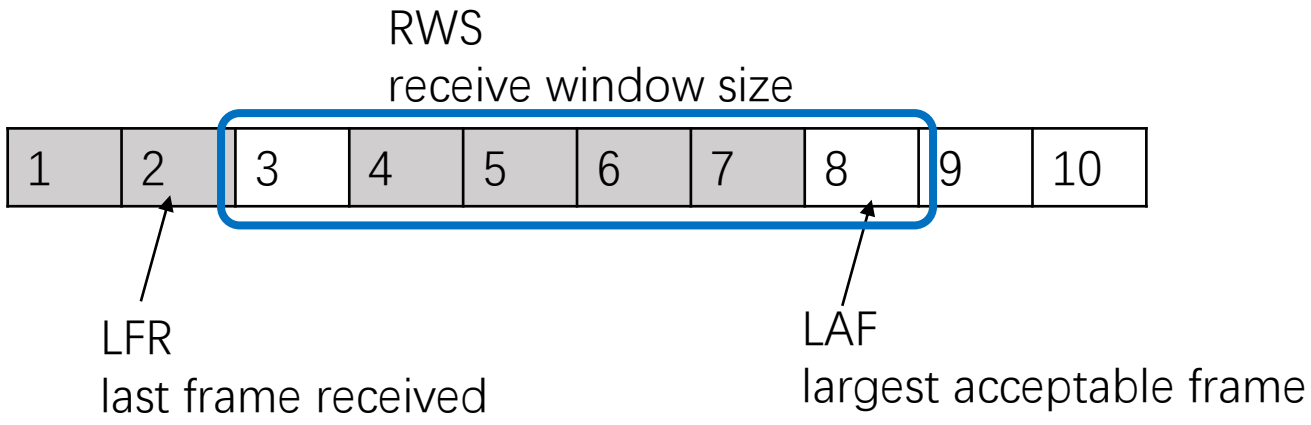
Sliding Window

- Receiver Protocol
  - Always maintain LAF – LFR <= RWS
  - When a frame with sequence number #SeqNum arrives
    - If #SeqNum ⩽ LFR or #SeqNum > LAF
      - Discard frame, send accumulative ACK.
    - If LFR < #SeqNum ⩽ LAF
      - Accept frame, send accumulative ACK, modify LFT and LAF.

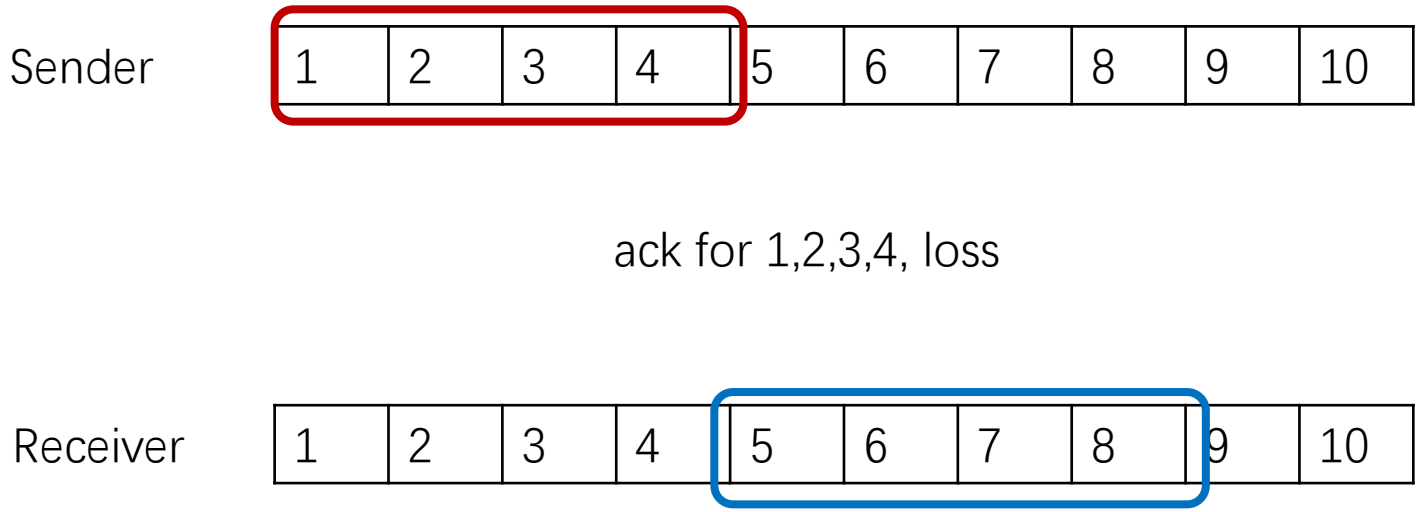


Accumulative Ack

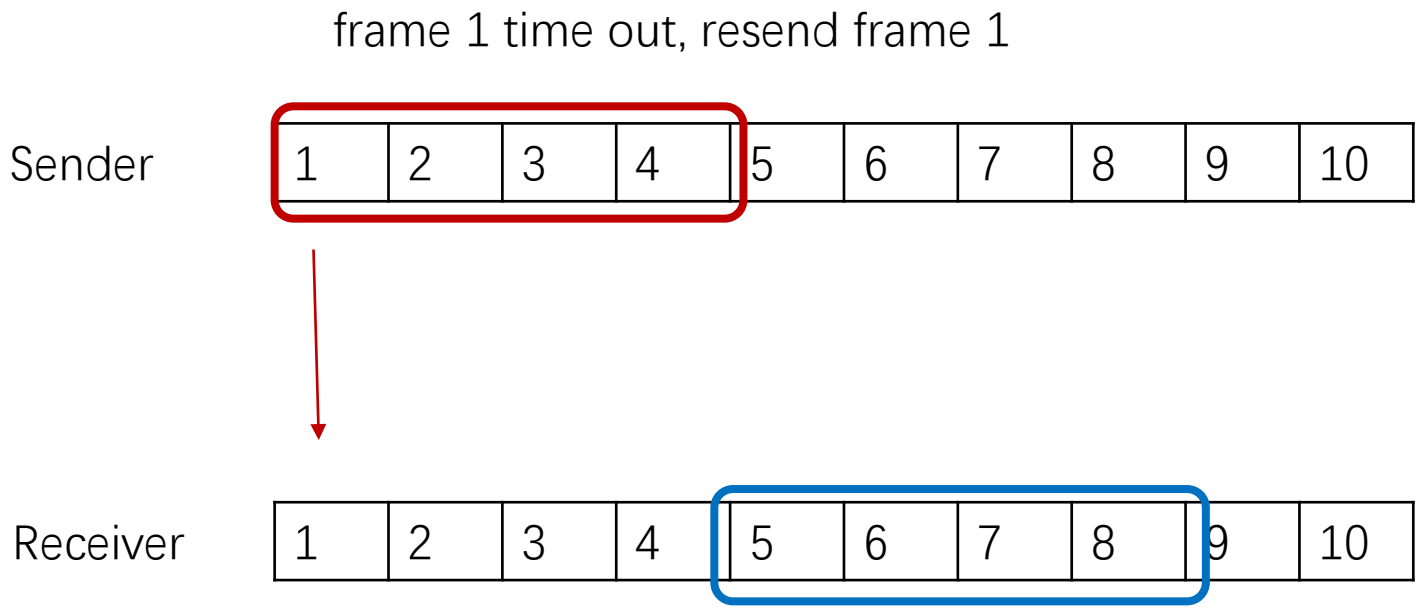
- If receive frame #3
  - Ack #7, move LFR to 7, move LAF to 13
- If receive frame #8
  - Ack #2



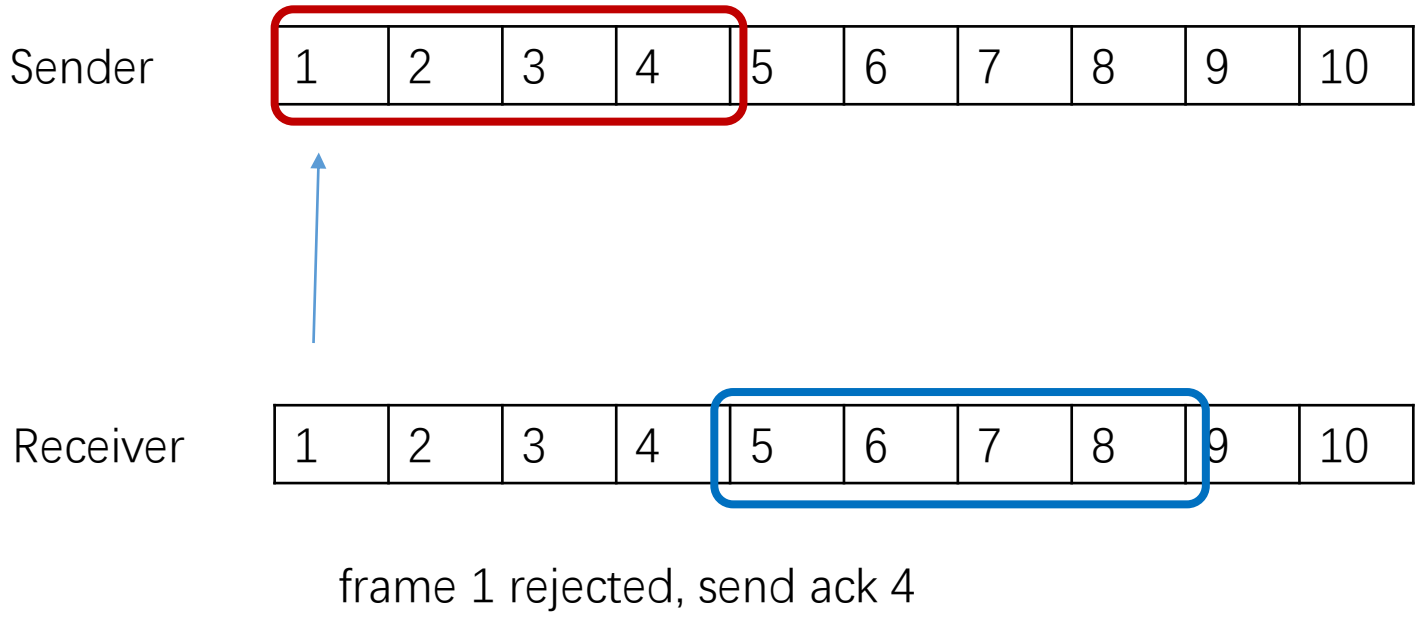
ACK Loss


Sender
1
2
3
4
5
6
7
8
9
10
ack for 1,2,3,4, loss
Receiver
1
2
3
4
5
6
7
8
9
10

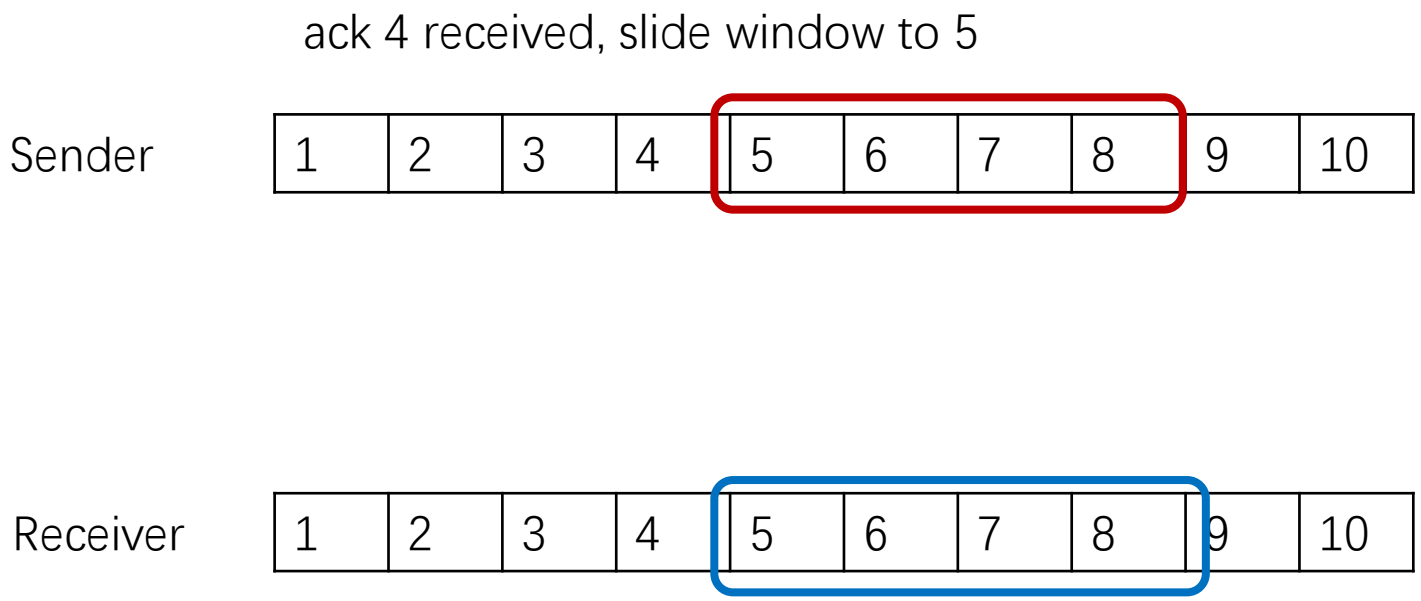
ACK Loss


frame 1 time out, resend frame 1
Sender
1
2
3
4
5
6
7
8
9
10
Receiver
1
2
3
4
5
6
7
8
9
10

ACK Loss


Sender
1
2
3
4
5
6
7
8
9
10
Receiver
1
2
3
4
5
6
7
8
9
10
frame 1 rejected, send ack 4

ACK Loss

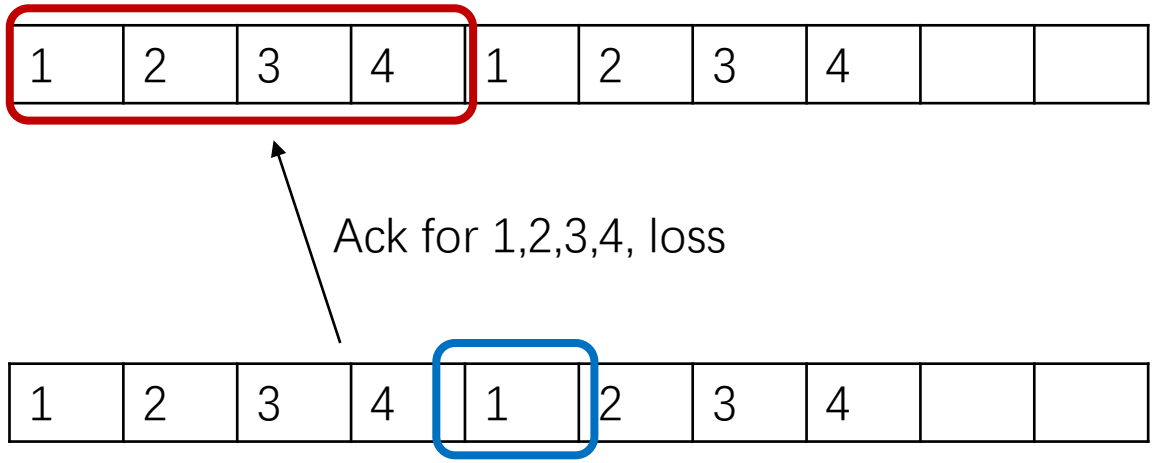

ack 4 received, slide window to 5
Sender
1
2
3
4
5
6
7
8
9
10
Receiver
1
2
3
4
5
6
7
8
9
10

Sliding Window

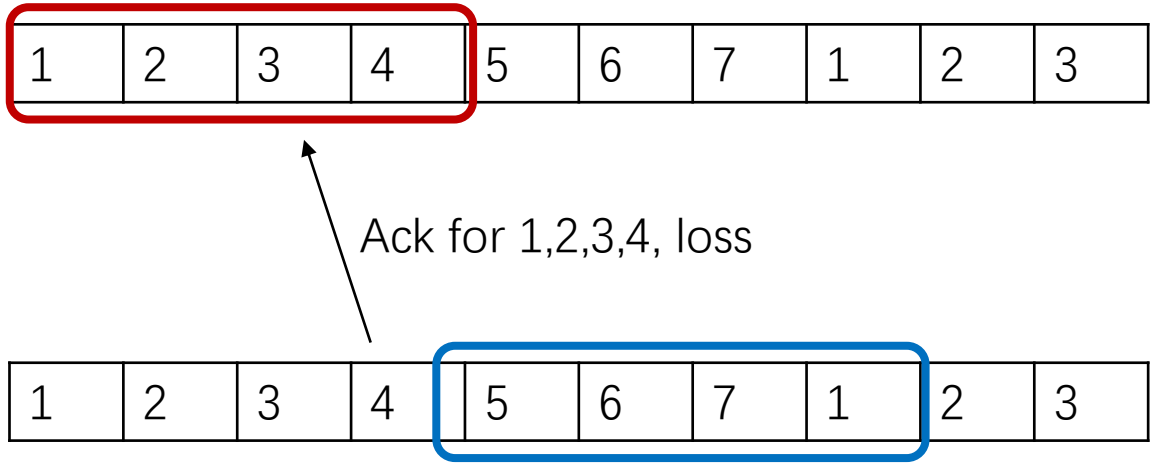
- Determine Window Size
  - Send Window Size: Pipeline depth
    - Delay × Bandwidth
  - Receive Window Size: Flow control

Sliding Window

- Determine SeqNum Range
  - if Receive Window Size == 1
    - MaxSeqNum >= Send Window Size+1



  - if Send Window Size == Receive Window Size
    - MaxSeqNum >= 2* Send Window Size



Demo

- Sliding Window code in TCP
  `/net/ipv4/`
- Change Sliding Window Scheme
  - 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]`
- https://www2.tkn.tu-berlin.de/teaching/rn/animations/gbn_sr/

Reference

- Textbook 1.5.2
- Textbook 2.5