


Computer Networks

CMSC 417 : Spring 2024

COMPUTER SCIENCE
UNIVERSITY OF MARYLAND

Topic: TCP Flow-control and Congestion-control (Textbook chapter 5 & 6)

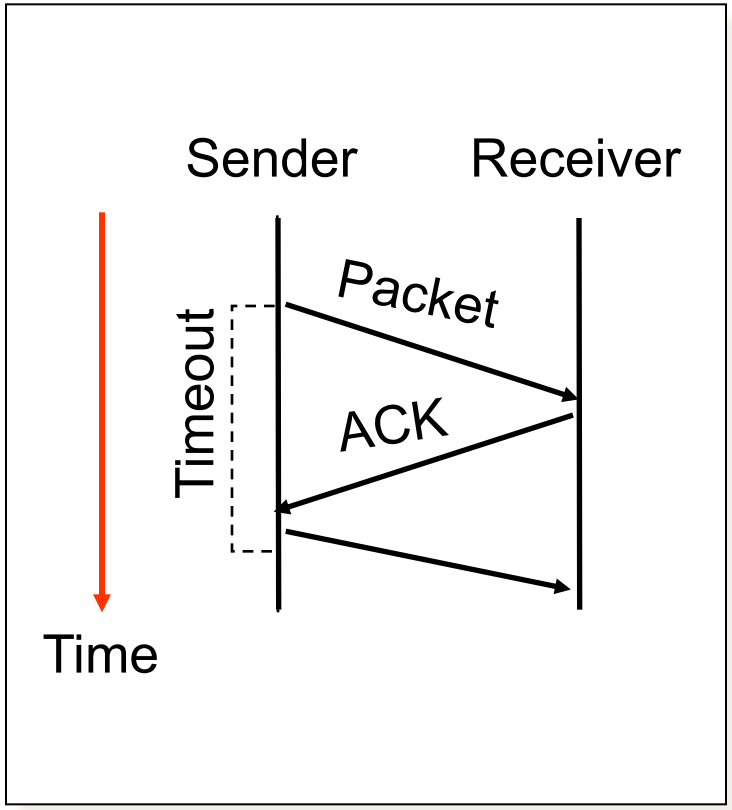
Nirupam Roy

Tu-Th 2:00-3:15pm
CSI 2117

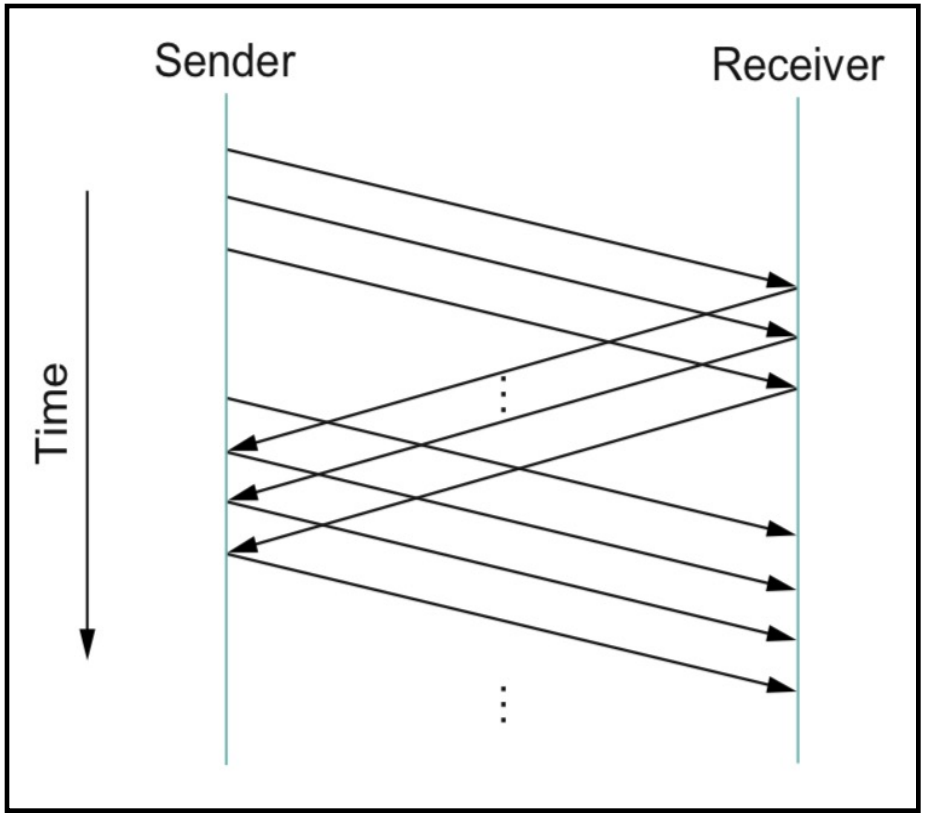
March 14th, 2024




Stop-and-wait

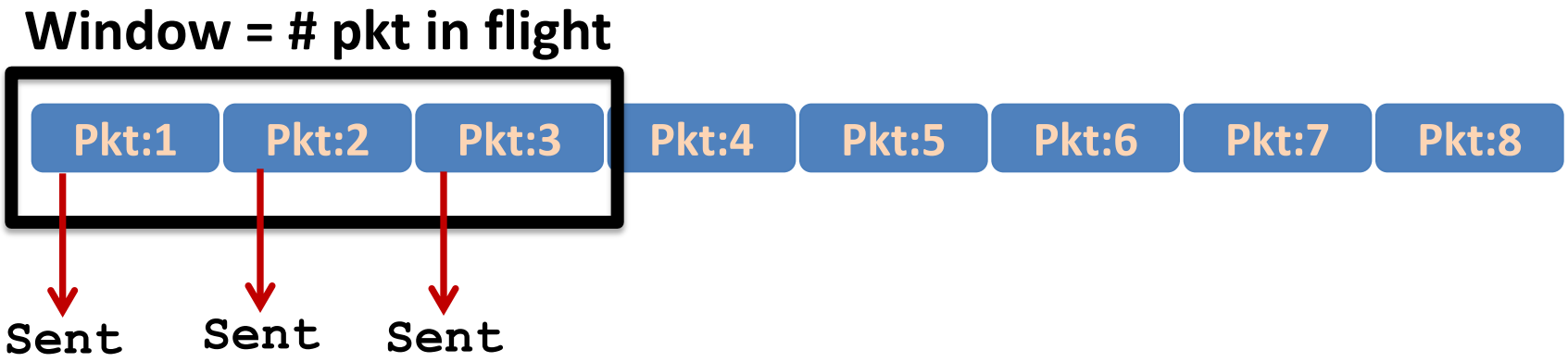


Sliding Window

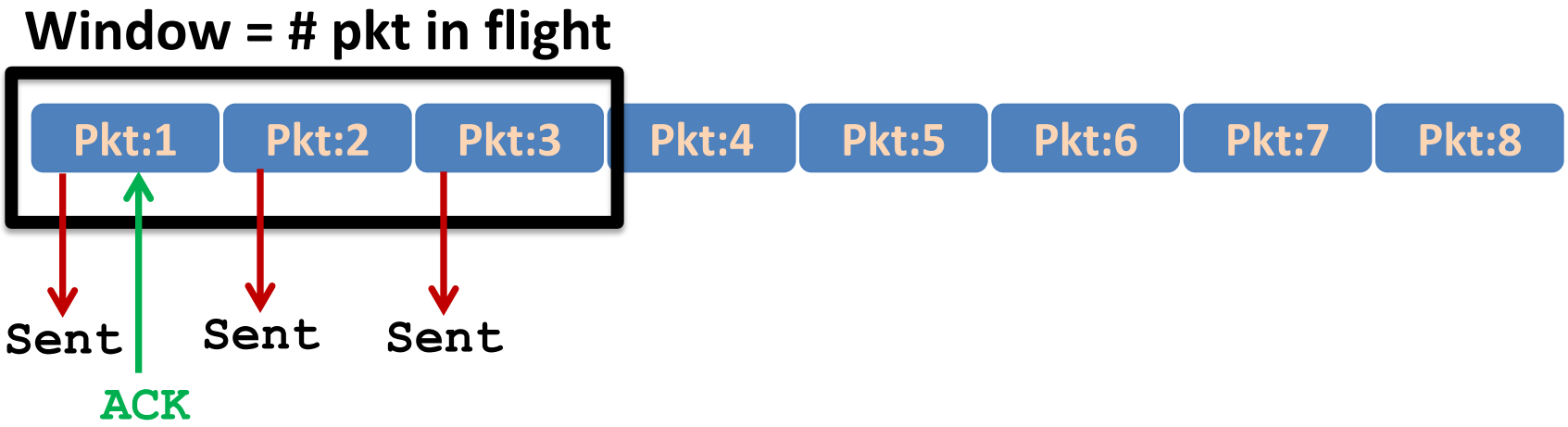


Sliding Window Protocol (recap)

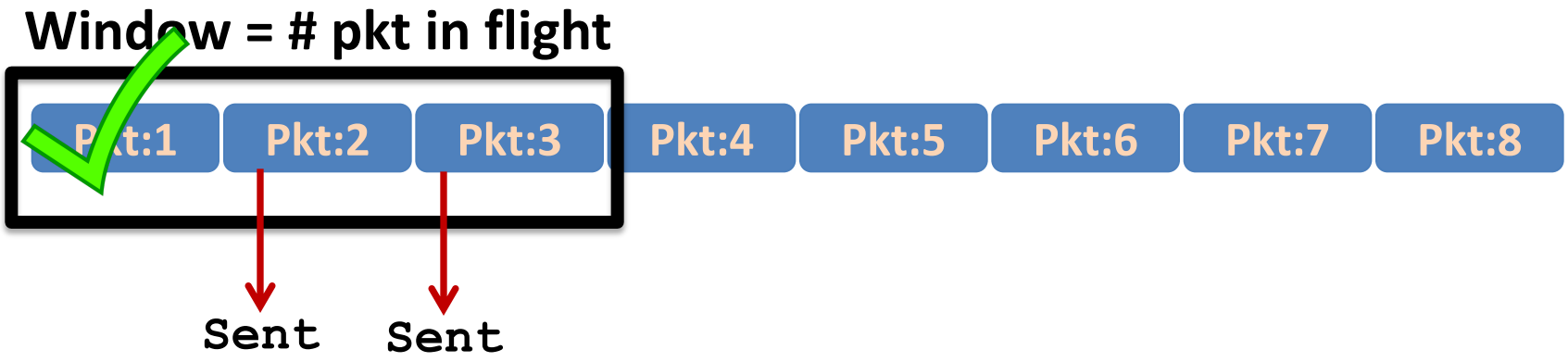
Sliding Window: A series of packets to be sent reliably



Sliding Window: A series of packets to be sent reliably

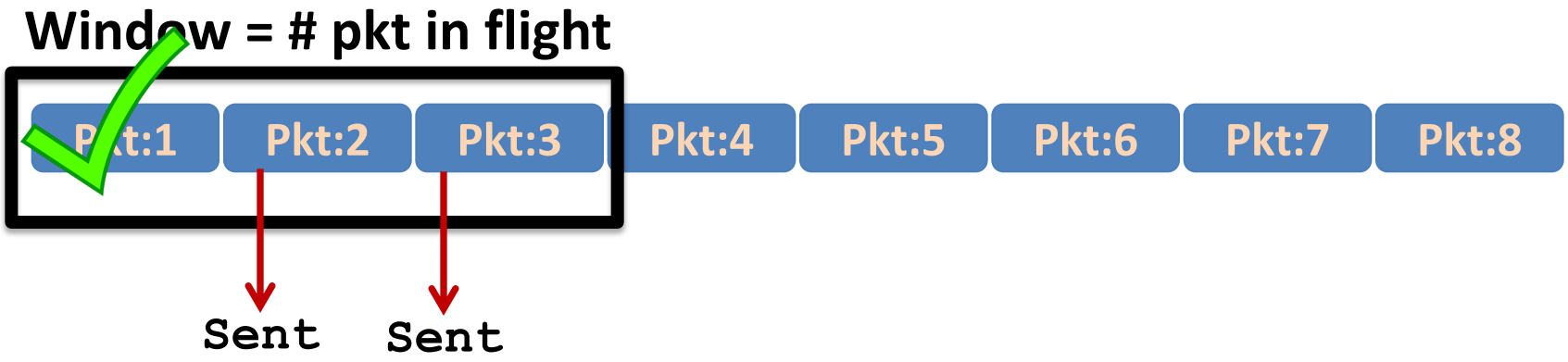


Sliding Window: A series of packets to be sent reliably



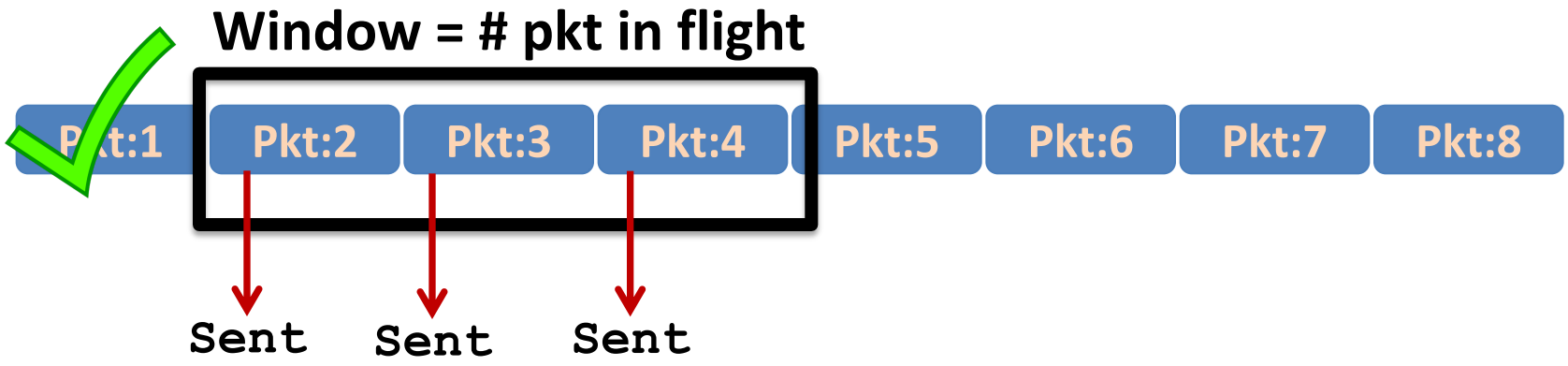
Packet 1 is delivered

Sliding Window: A series of packets to be sent reliably



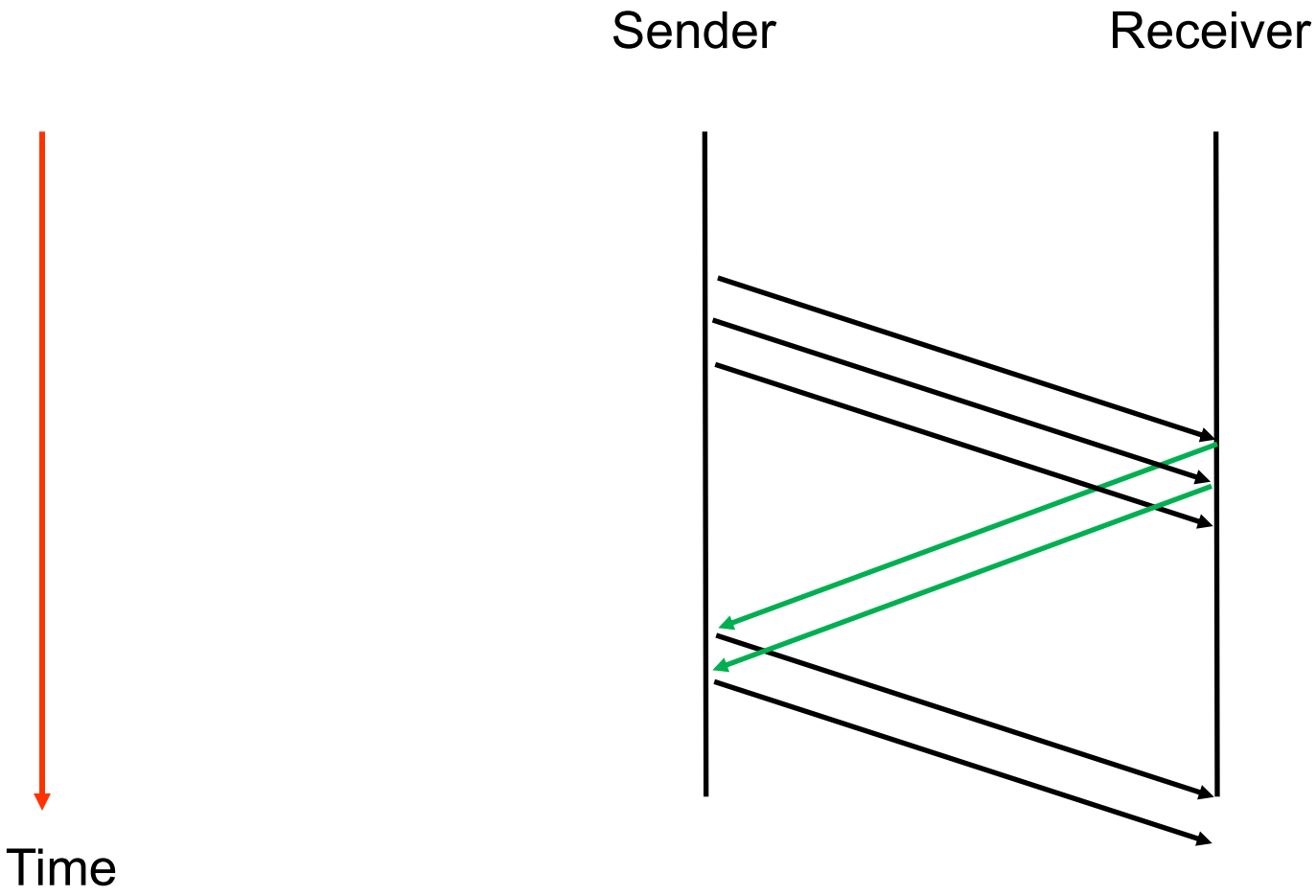
Packet 1 is delivered. Slide the window.

Sliding Window: A series of packets to be sent reliably

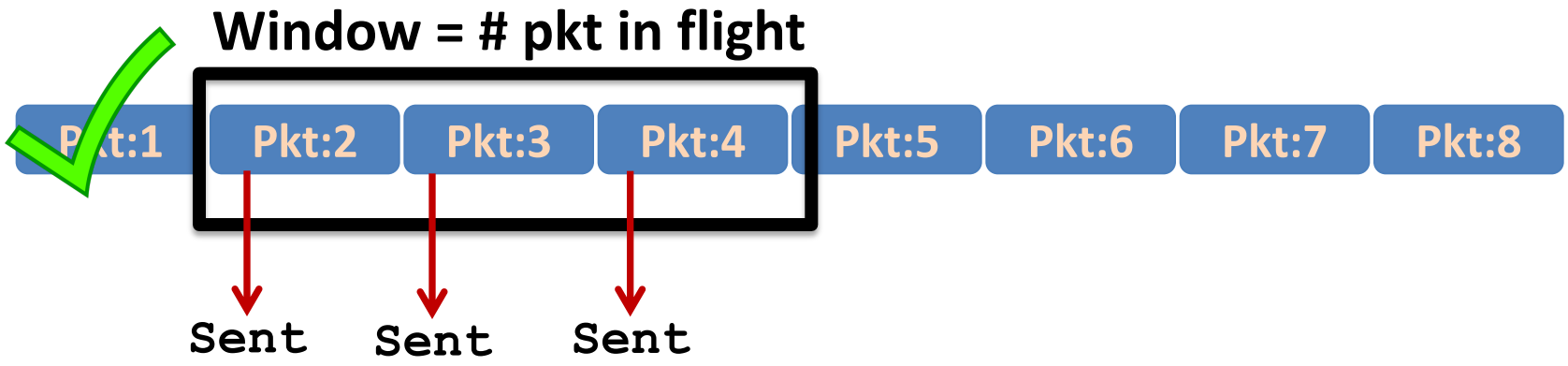


Packet 1 is delivered. Slide the window.

Sliding Window: A series of packets to be sent reliably



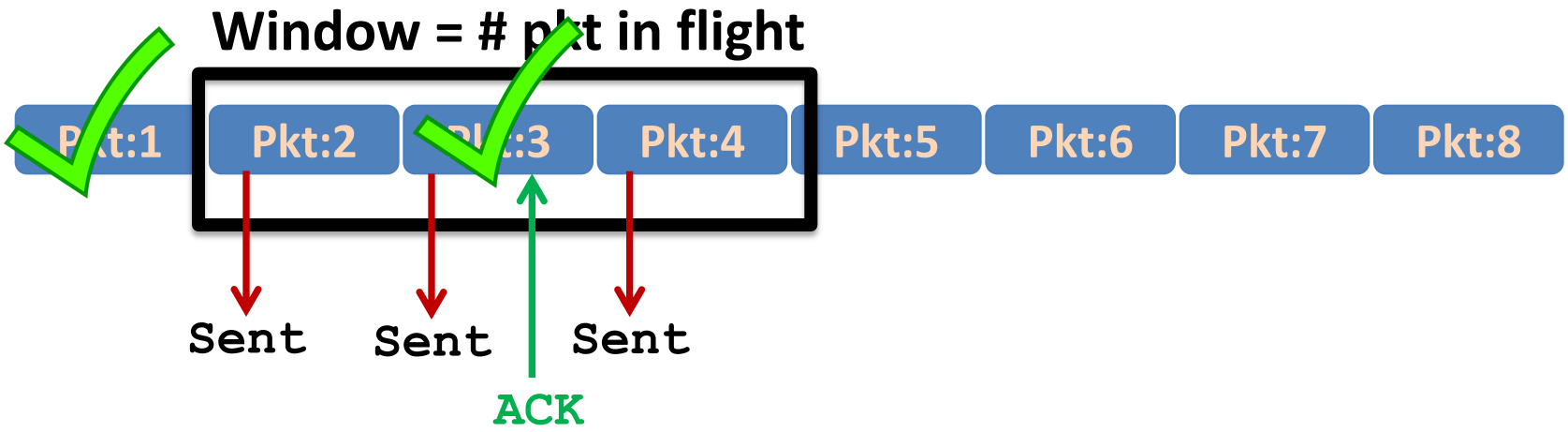
Sliding Window: A series of packets to be sent reliably



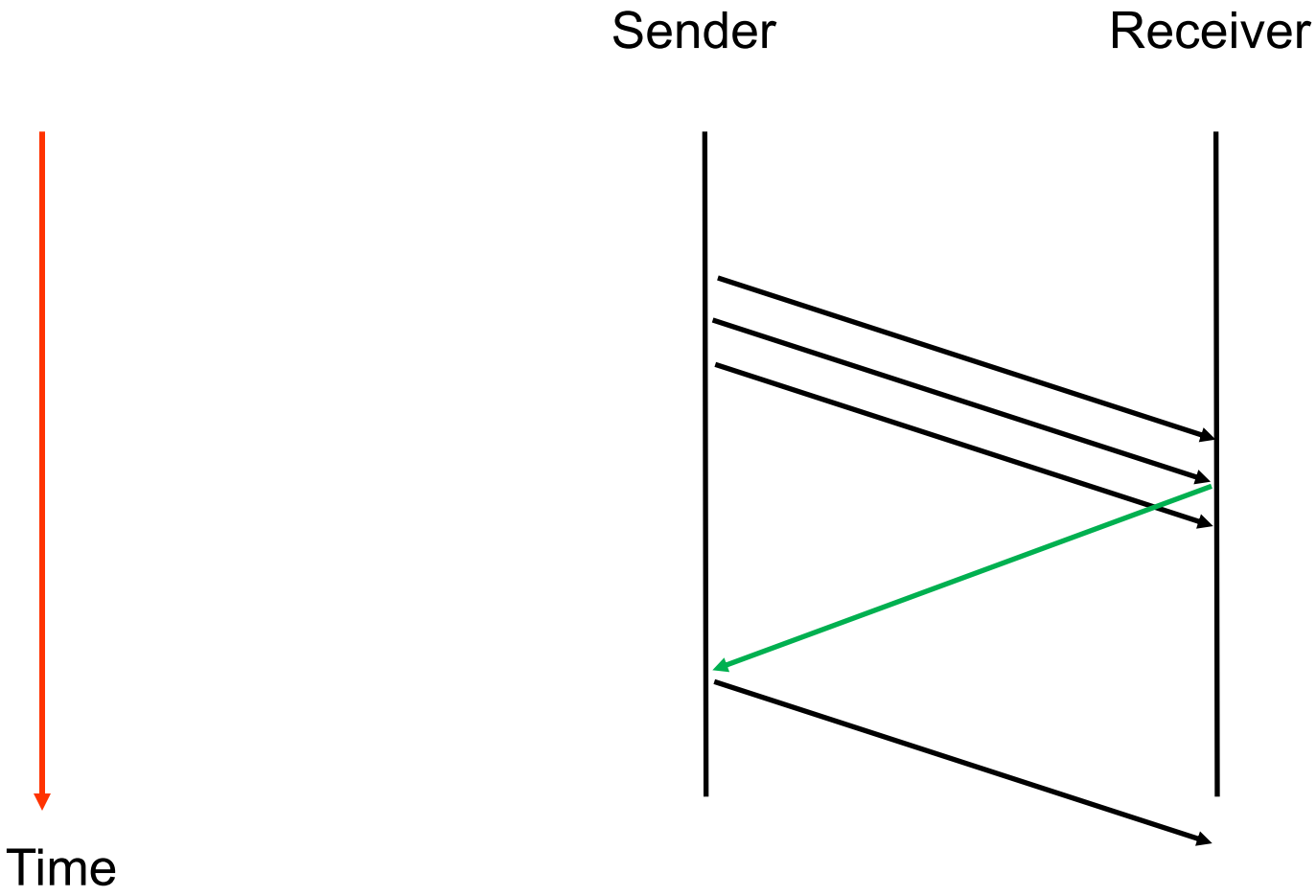
Packet 1 is delivered. Slide the window.

Sliding Window: A series of packets to be sent reliably

Should we advance the window in the following scenario?

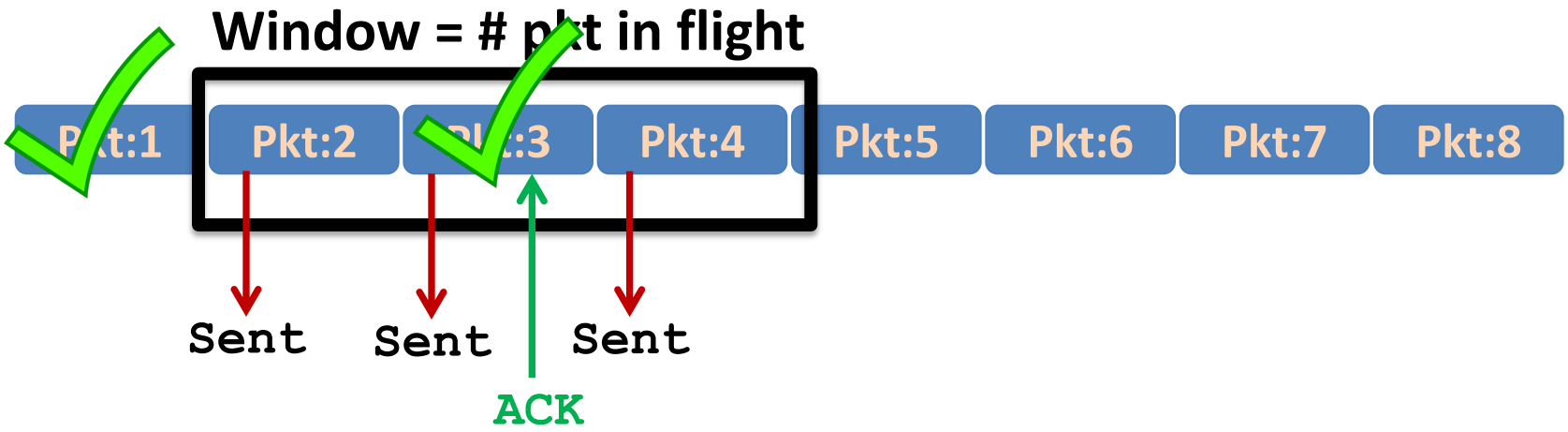


Sliding Window: A series of packets to be sent reliably



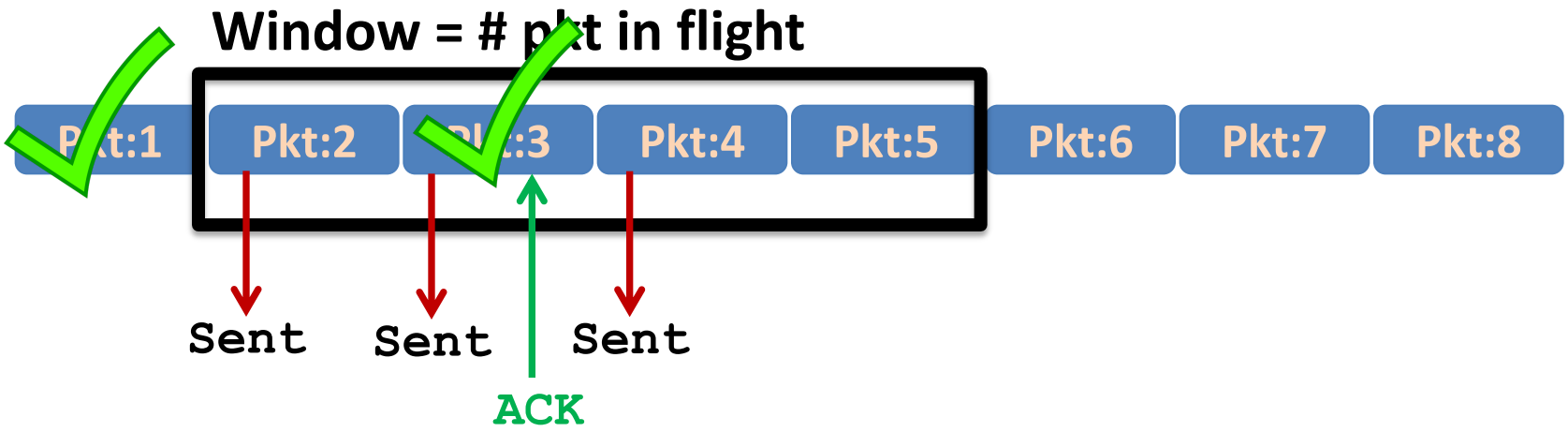
Sliding Window: A series of packets to be sent reliably

Should we advance the window in the following scenario?



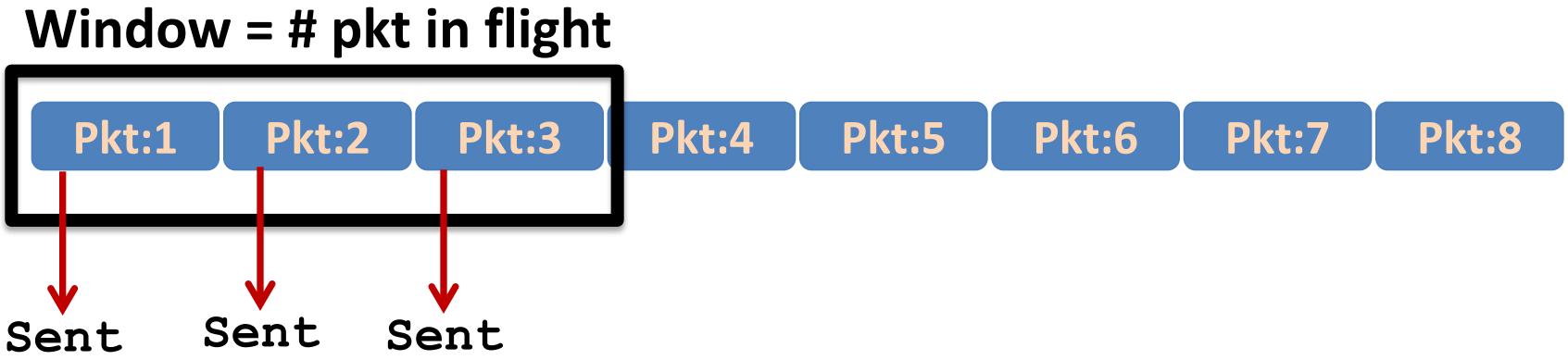
Sliding Window: A series of packets to be sent reliably

Should we advance the window in the following scenario?



Note that the window size is not changed!

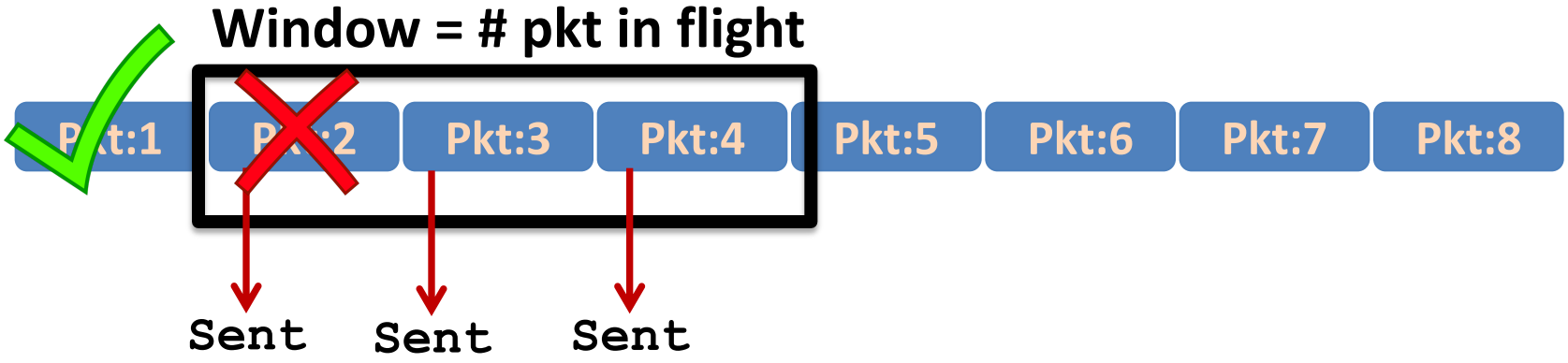
Sliding Window: A series of packets to be sent reliably



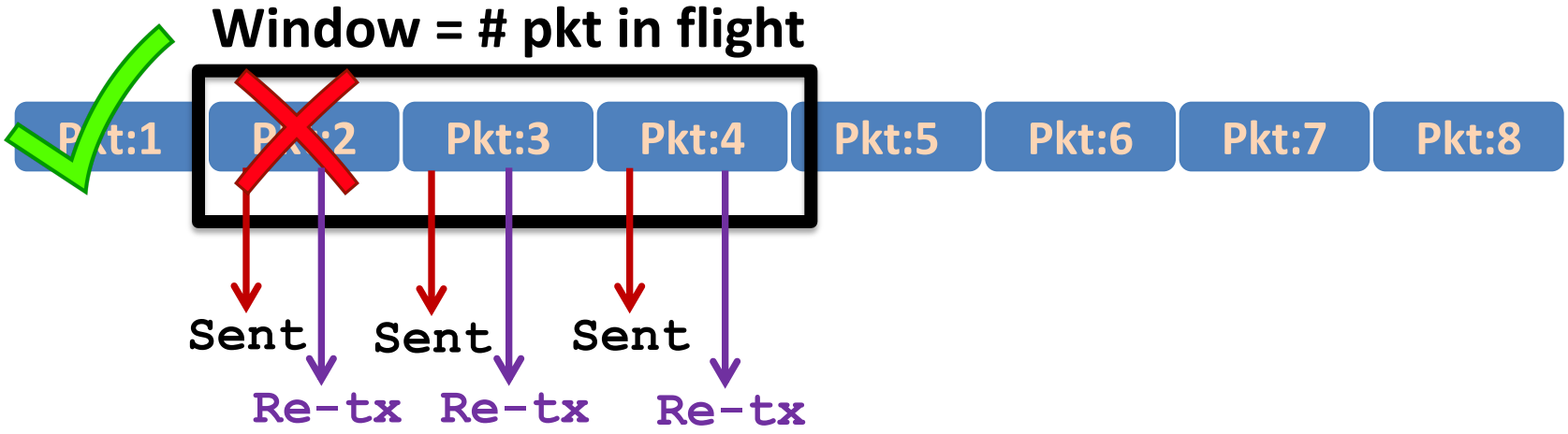
Can you make sliding window behave like stop-and-wait?

Sliding Window Protocol (dealing packet loss)

Sliding Window: A series of packets to be sent reliably



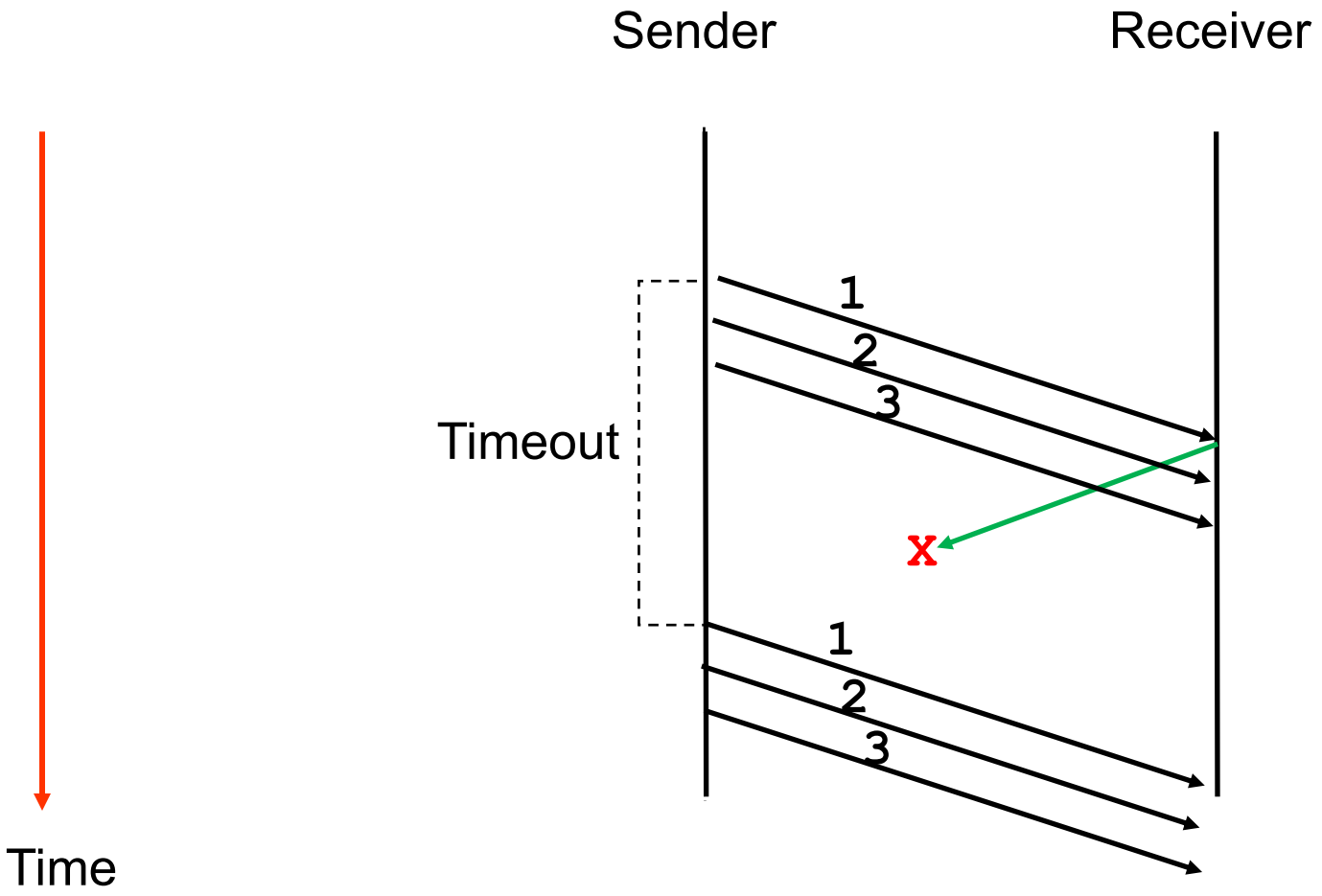
Sliding Window: A series of packets to be sent reliably



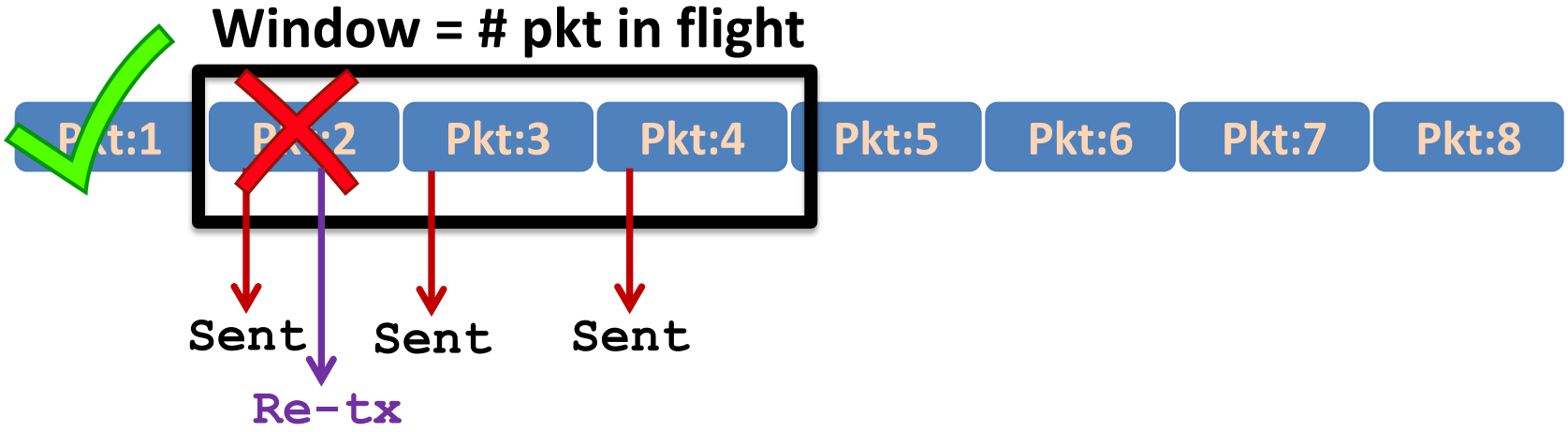
(1) ARQ strategy 1: Retransmit all un-ACKed packets

Sliding Window: A series of packets to be sent reliably

(1) Go-back-N (Sender side)



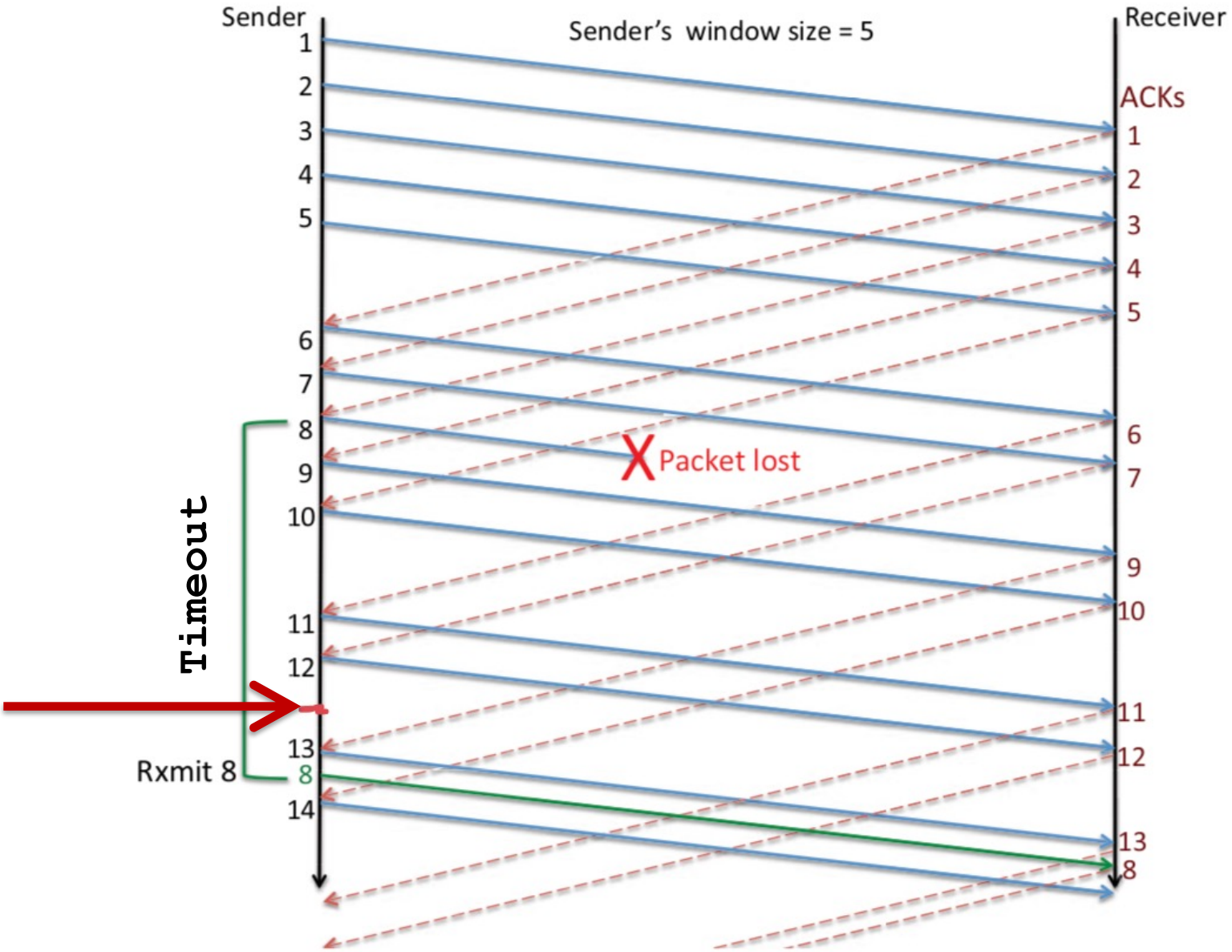
Sliding Window: A series of packets to be sent reliably



(2) ARQ strategy 2: Retransmit only the lost packet

(2) Selective retransmit (Sender side)

Sliding Window: Lost Packet/ACK



TCP Cumulative ACK strategy (receiver side)

Cumulative ACK:

ACK the largest in-sequence packet received.

What will be ACK sequence if the receive sequence is: 5, 7, 8, 6 ?

TCP Cumulative ACK strategy (receiver side)

Cumulative ACK:

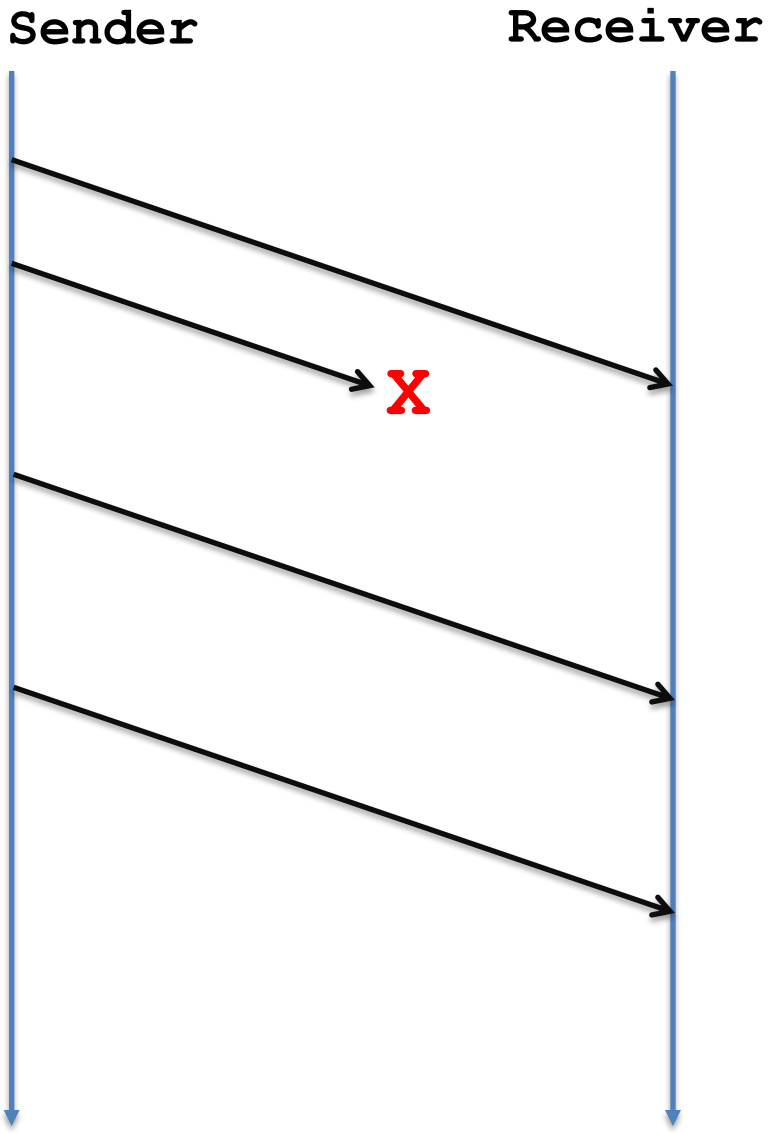
ACK the largest in-sequence packet received.

What will be ACK sequence if the receive sequence is: 5, 7, 8, 6 ?

ACK: 6, 6, 6, 9

TCP Cumulative ACK strategy (receiver side)

(A more precise example)



TCP Cumulative ACK strategy (receiver side)

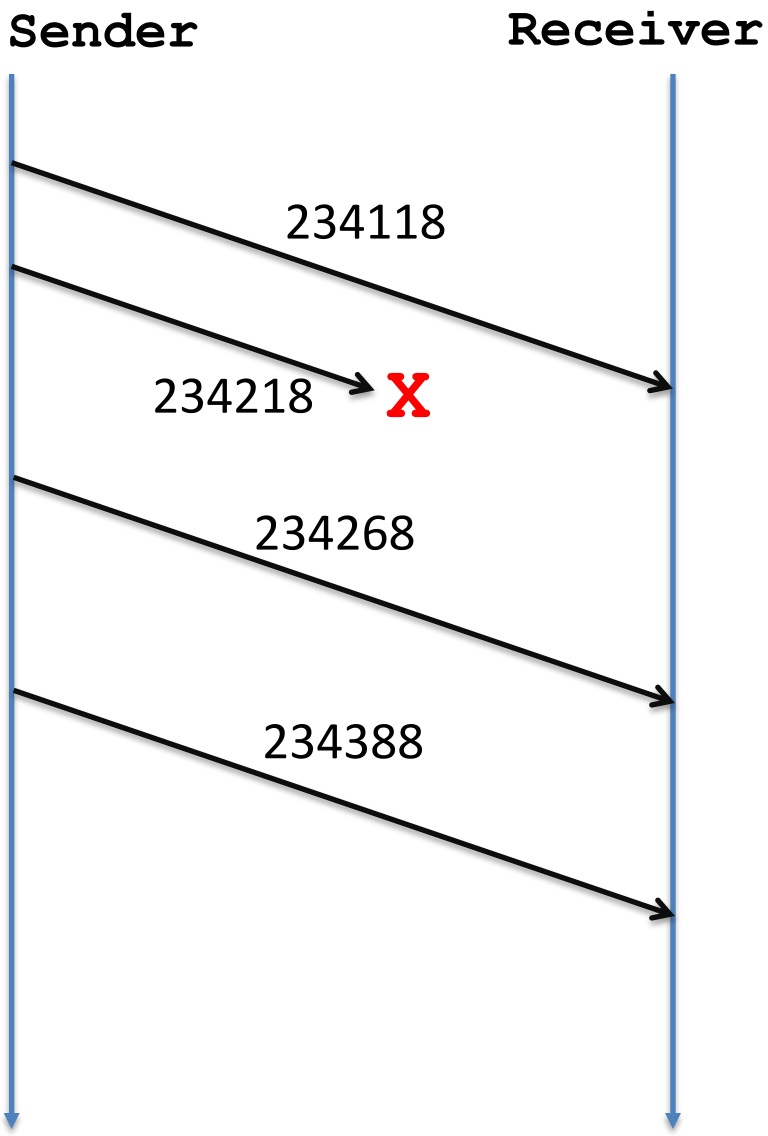
(A more specific example)

Seq. #	234118
ACK flag	N/A
ACK #	N/A
Data Len	100 bytes

Seq. #	234218
ACK flag	N/A
ACK #	N/A
Data Len	50 bytes

Seq. #	234268
ACK flag	N/A
ACK #	N/A
Data Len	120 bytes

Seq. #	234388
ACK flag	N/A
ACK #	N/A
Data Len	100 bytes



TCP Cumulative ACK strategy (receiver side)

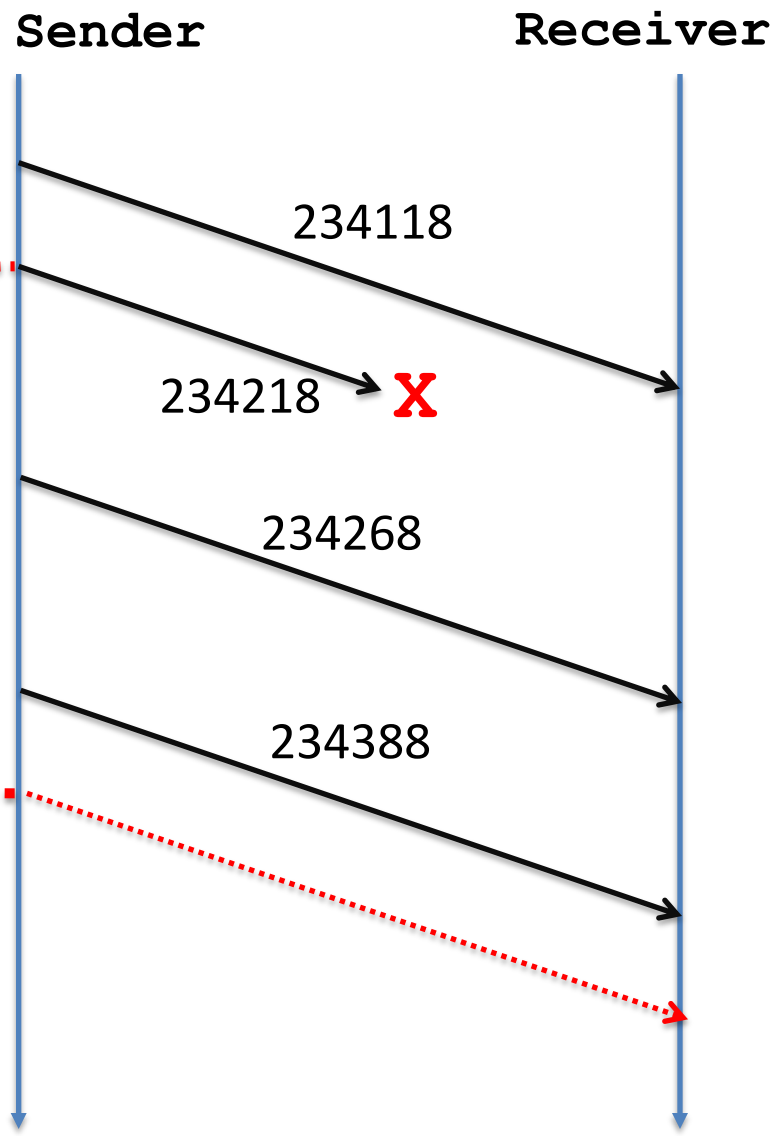
(A more specific example)

Seq. #	234118
ACK flag	N/A
ACK #	N/A
Data Len	100 bytes

Seq. #	234218
ACK flag	N/A
ACK #	N/A
Data Len	50 bytes

Seq. #	234268
ACK flag	N/A
ACK #	N/A
Data Len	120 bytes

Seq. #	234388
ACK flag	N/A
ACK #	N/A
Data Len	100 bytes



TCP Cumulative ACK strategy (receiver side)

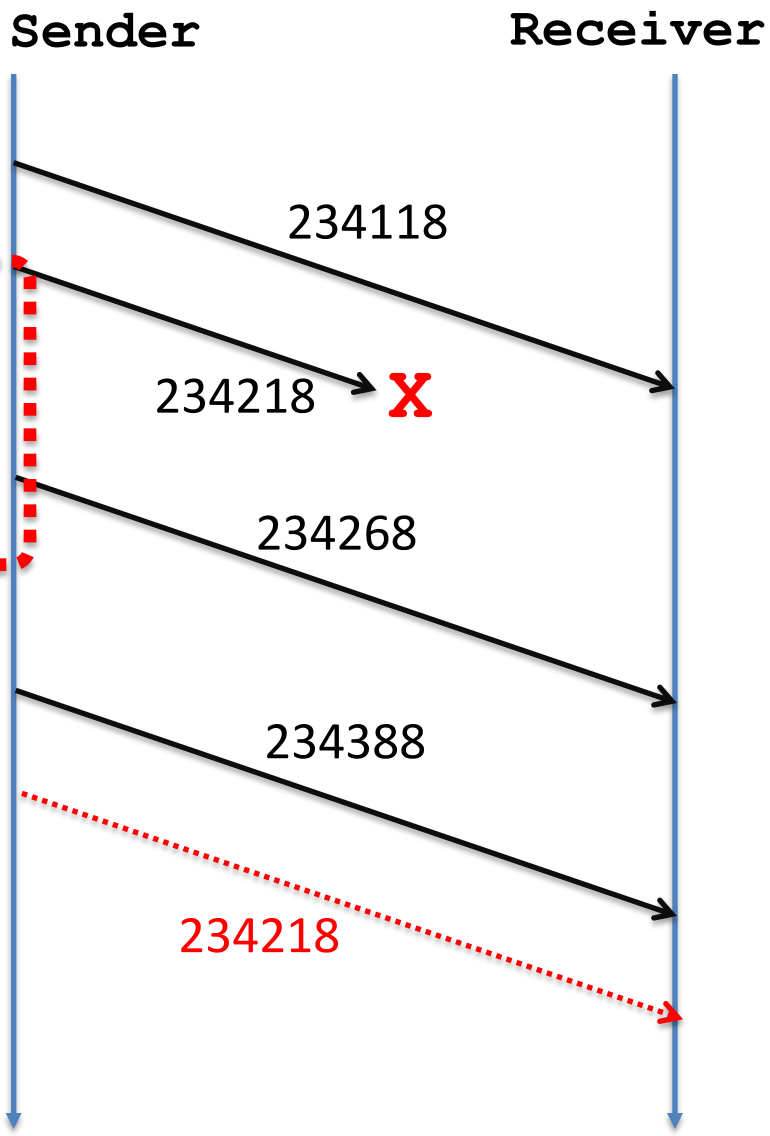
(A more specific example)

Seq. #	234118
ACK flag	N/A
ACK #	N/A
Data Len	100 bytes

Seq. #	234218
ACK flag	N/A
ACK #	N/A
Data Len	50 bytes

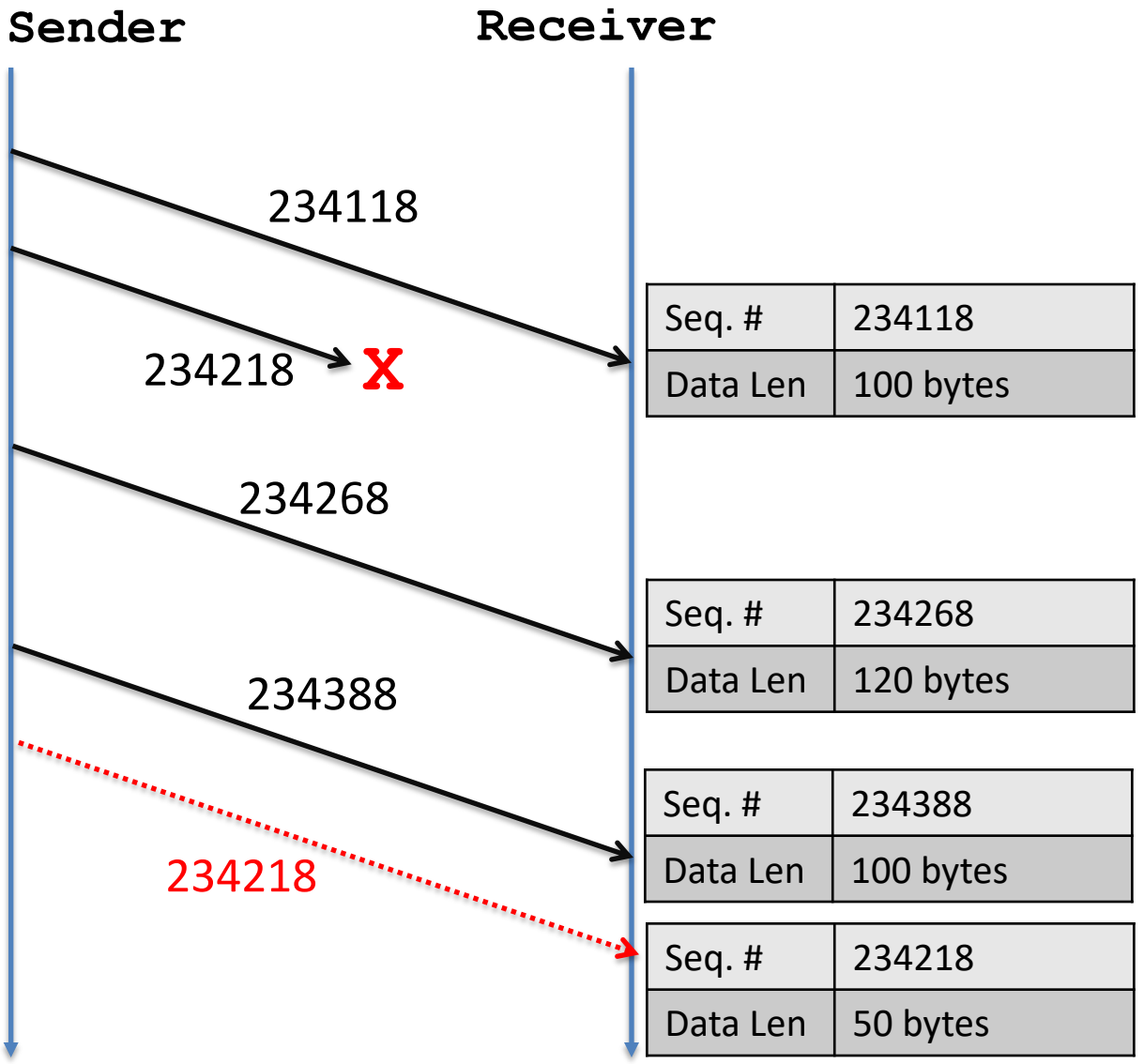
Seq. #	234268
ACK flag	N/A
ACK #	N/A
Data Len	120 bytes

Seq. #	234388
ACK flag	N/A
ACK #	N/A
Data Len	100 bytes



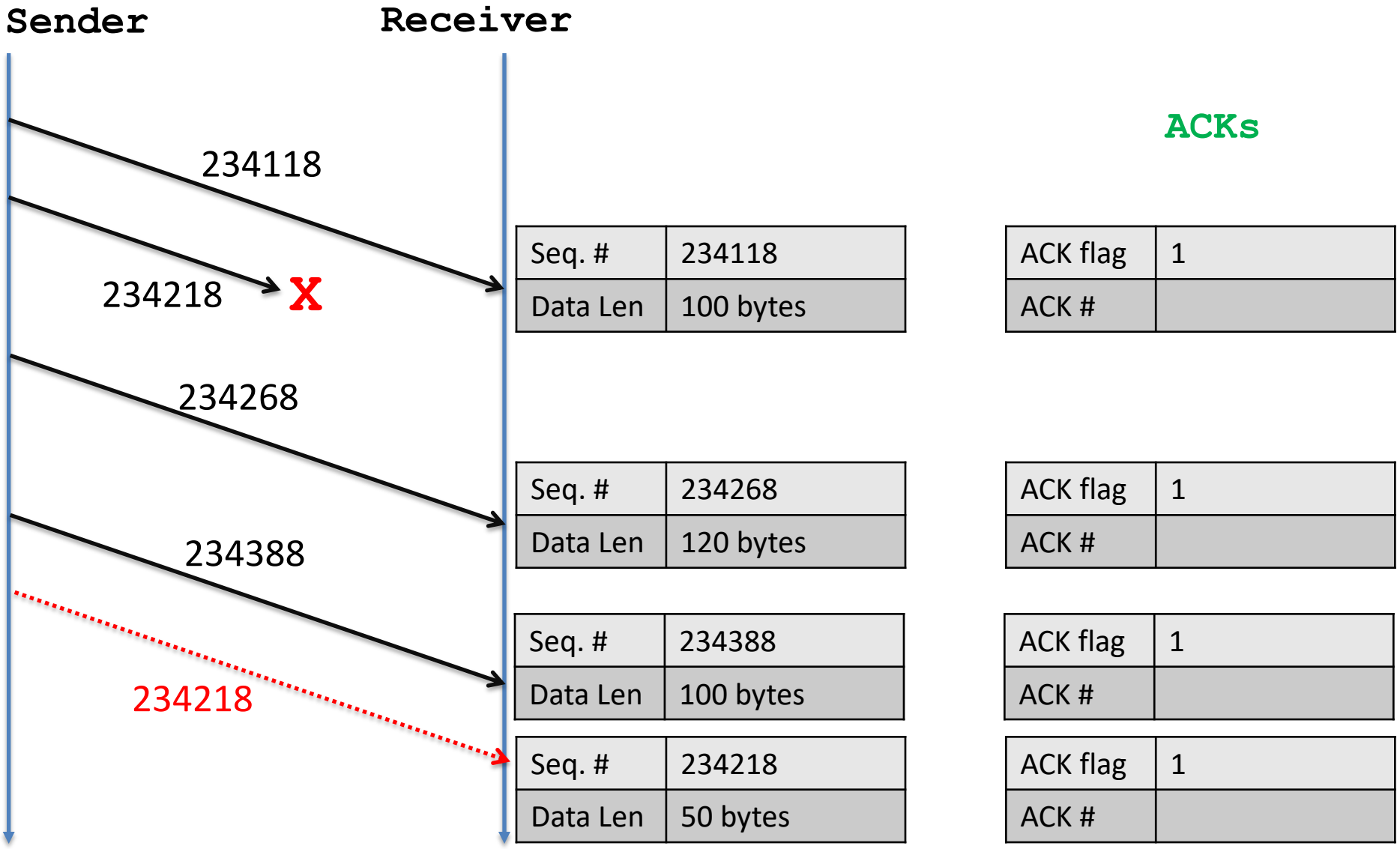
TCP Cumulative ACK strategy (receiver side)

(A more specific example)



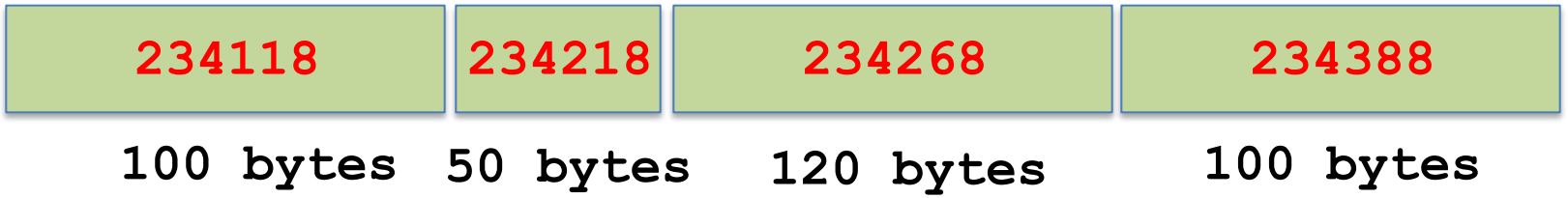
TCP Cumulative ACK strategy (receiver side)

(A more specific example)

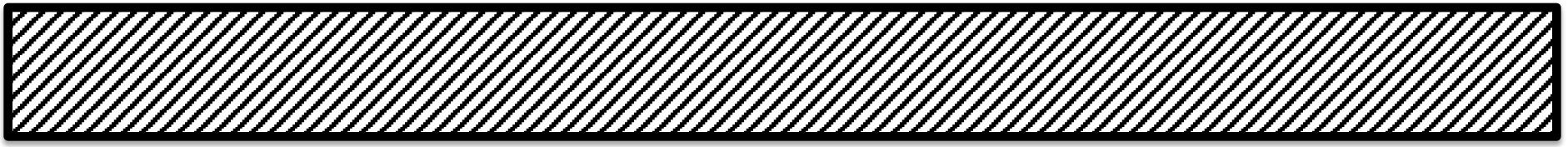


TCP Cumulative ACK strategy (receiver side)

Sender buffer:

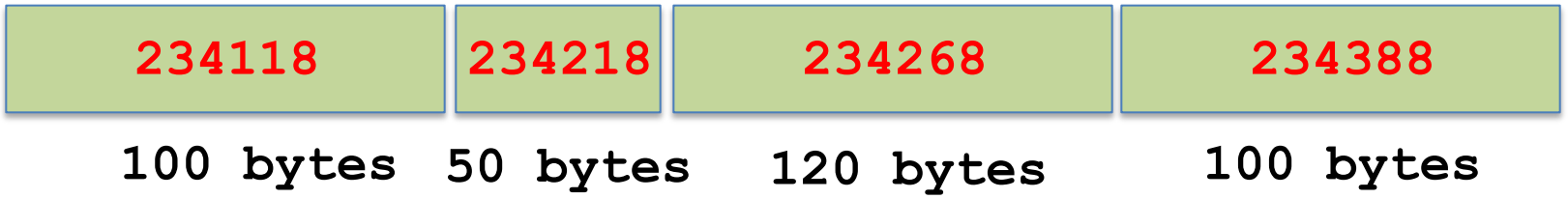


Receiver buffer:

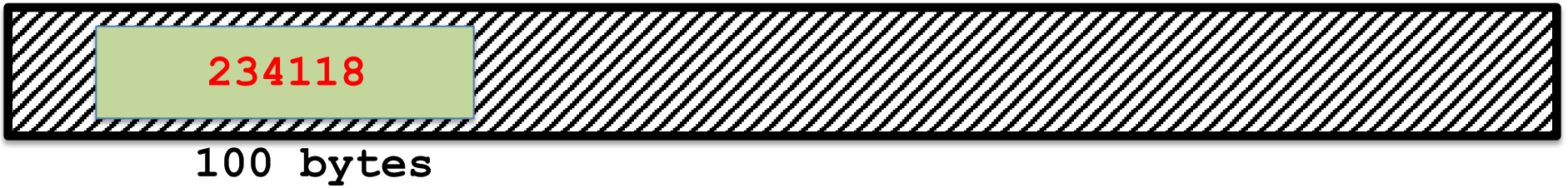


TCP Cumulative ACK strategy (receiver side)

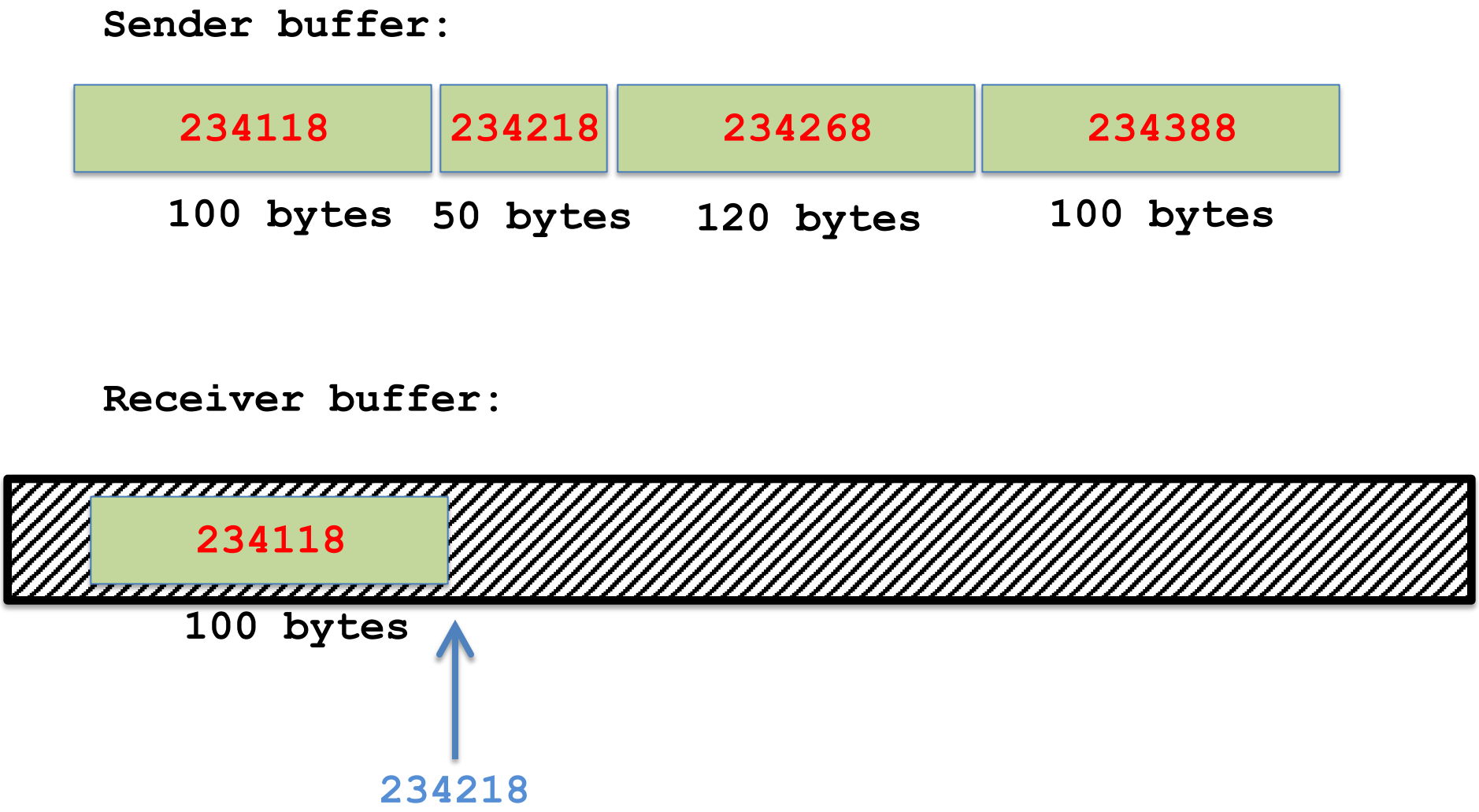
Sender buffer:



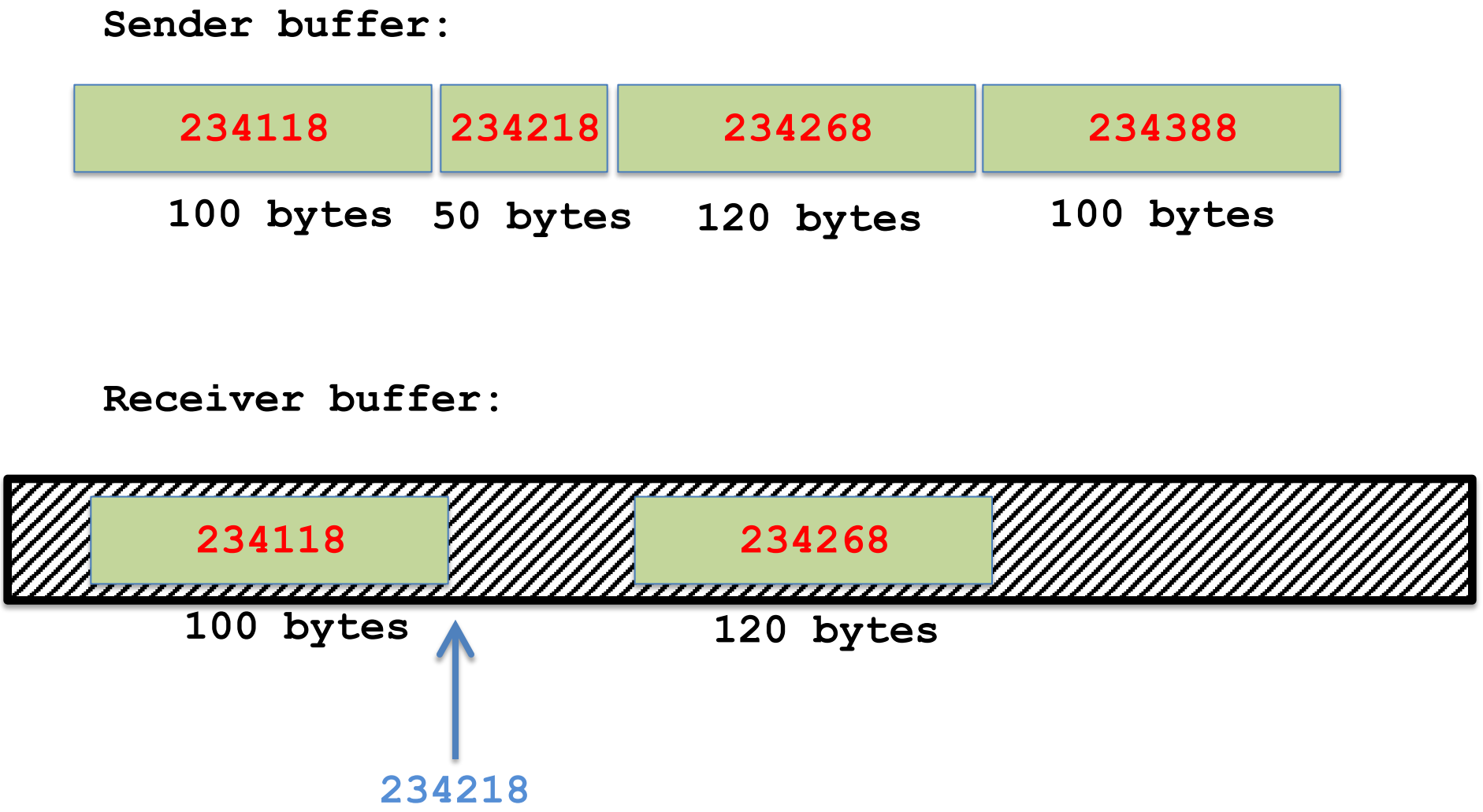
Receiver buffer:



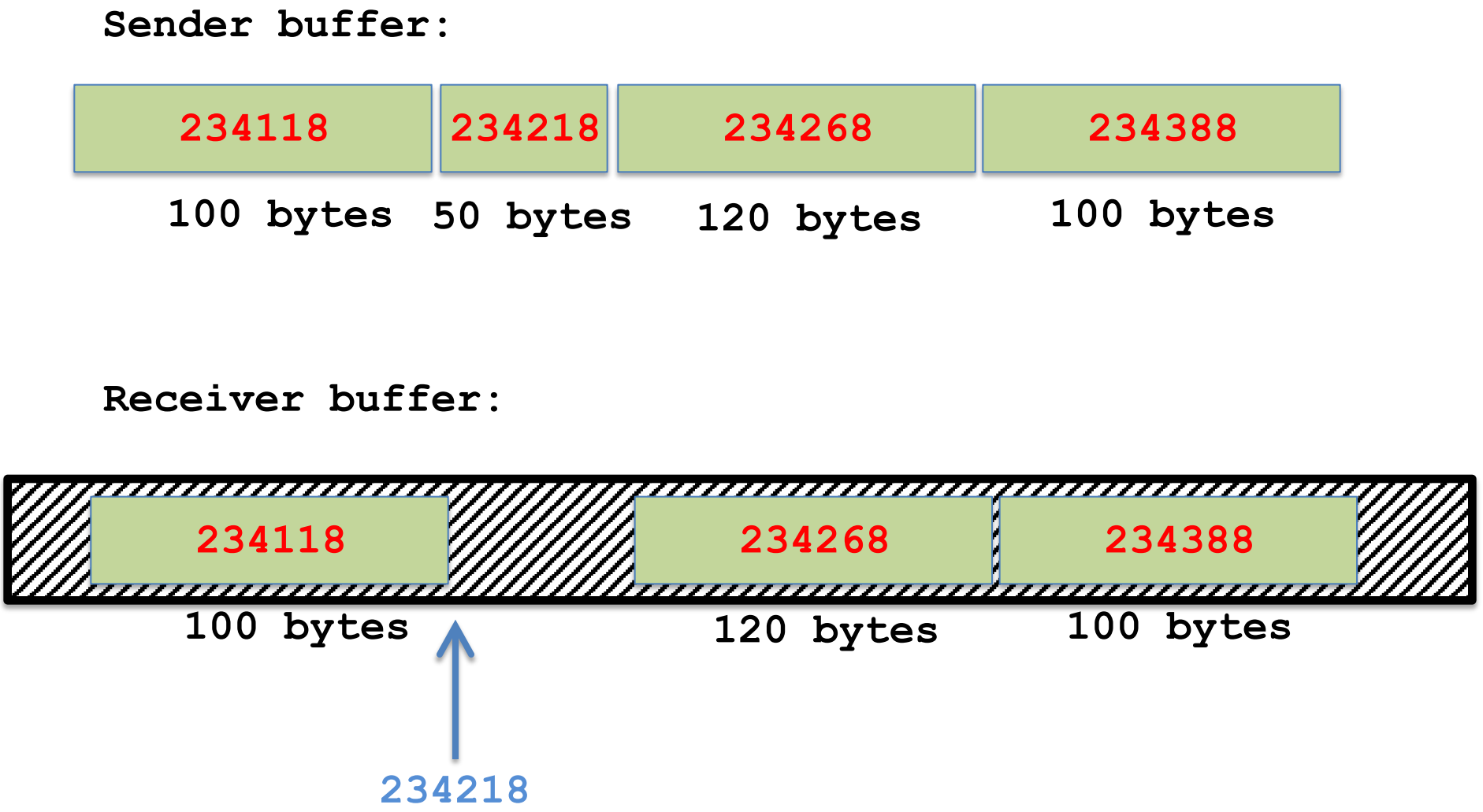
TCP Cumulative ACK strategy (receiver side)



TCP Cumulative ACK strategy (receiver side)

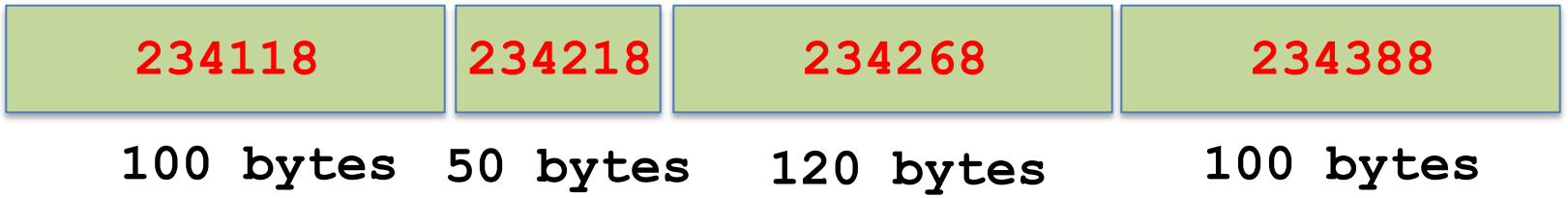


TCP Cumulative ACK strategy (receiver side)

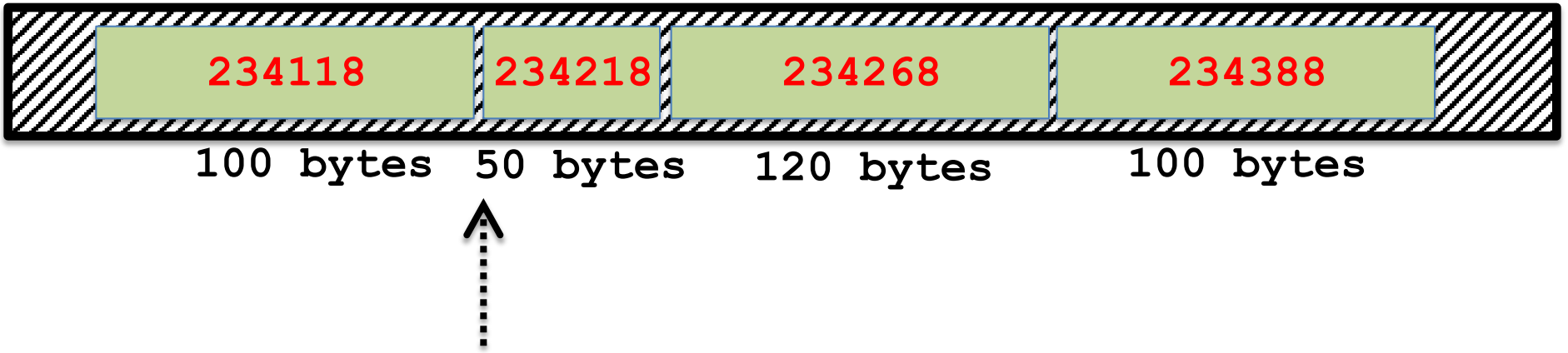


TCP Cumulative ACK strategy (receiver side)

Sender buffer:

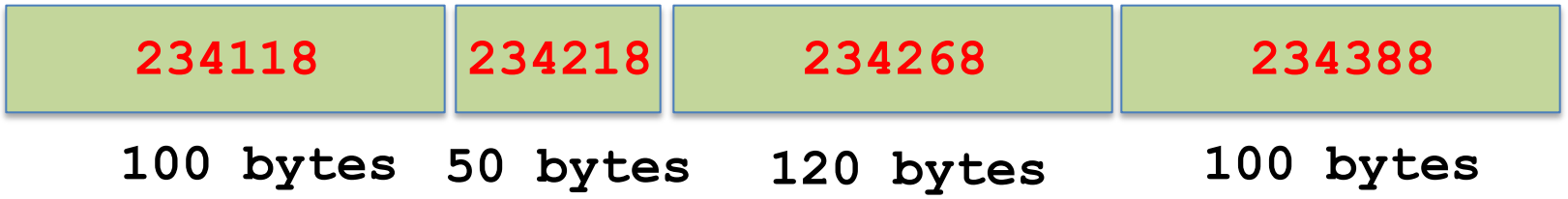


Receiver buffer:

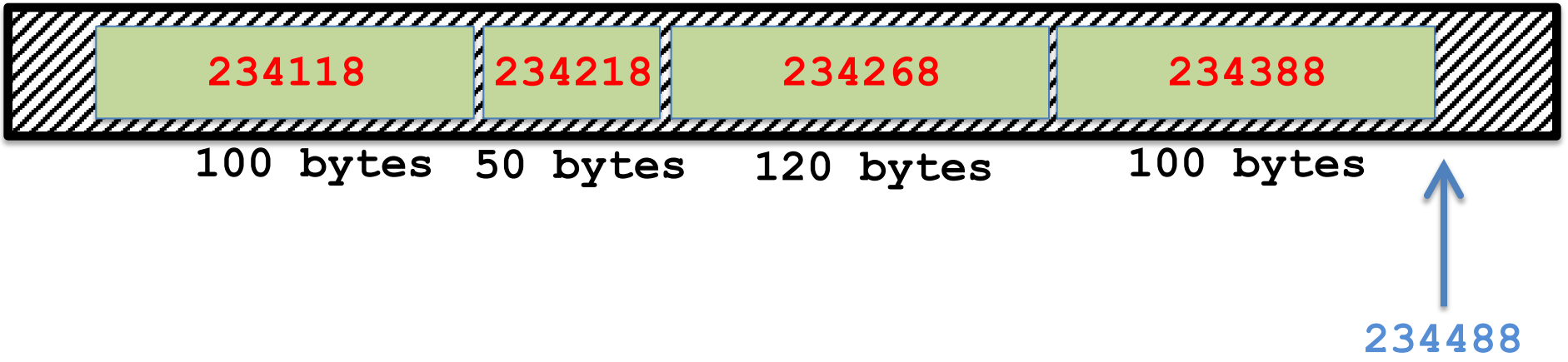


TCP Cumulative ACK strategy (receiver side)

Sender buffer:

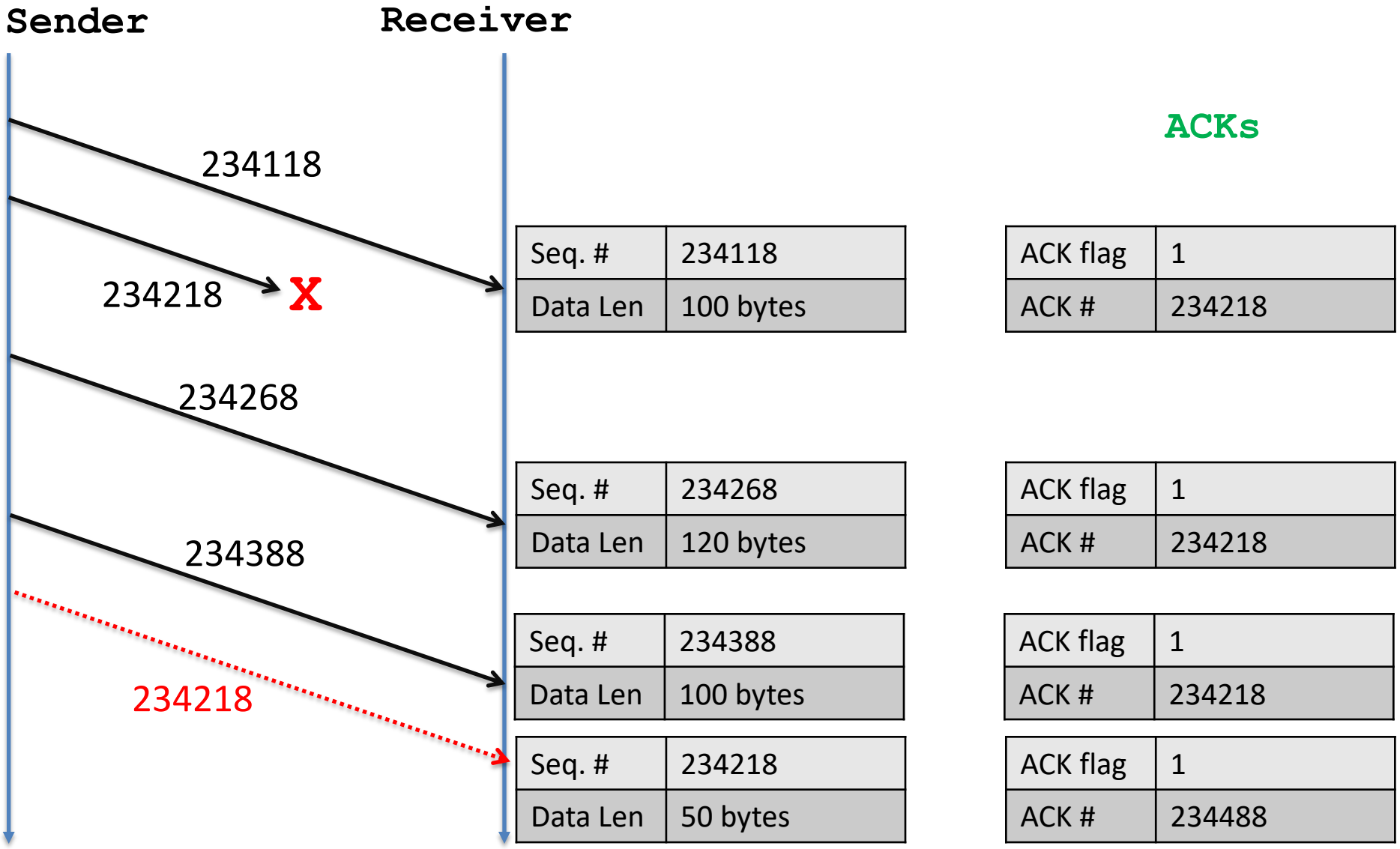


Receiver buffer:



TCP Cumulative ACK strategy (receiver side)

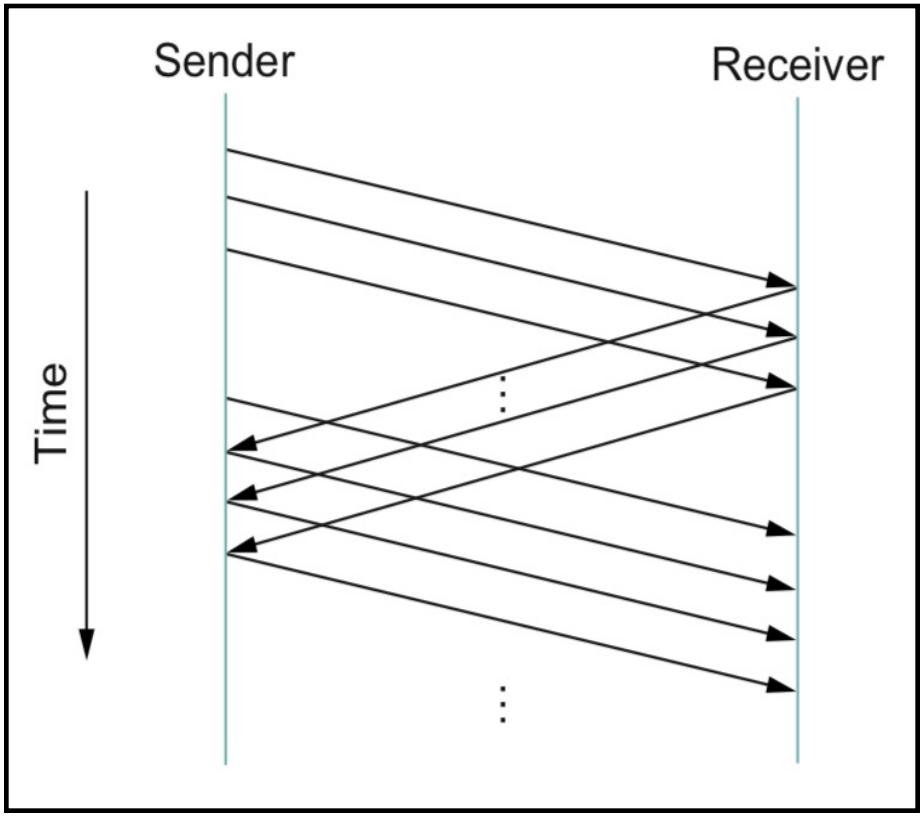
(A more specific example)



Flow Control:
TCP Sliding Window

Motivation: Maximum utilization of the resources without overwhelming the receiver.

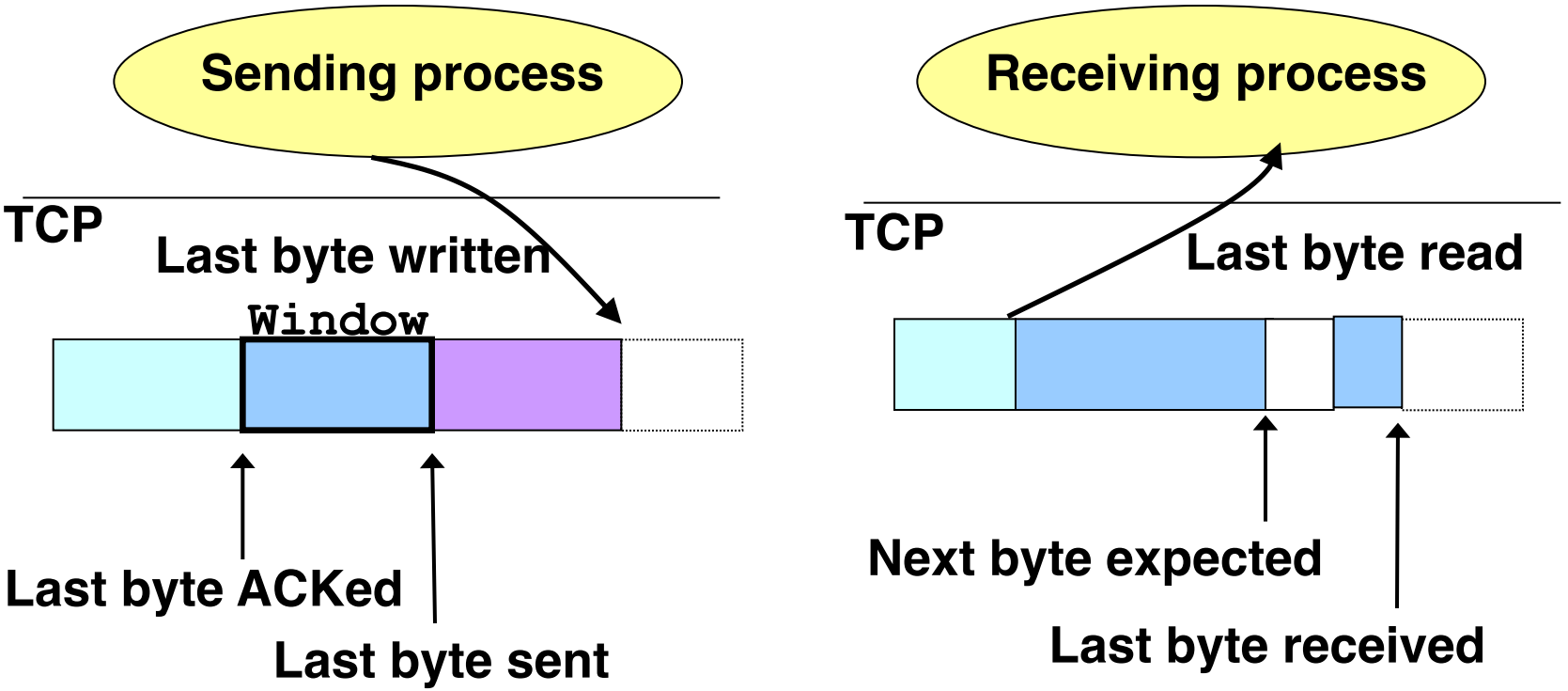
Sliding Window



Receiver should have enough space to buffer/save the received data !!

Sliding Window: Implementation

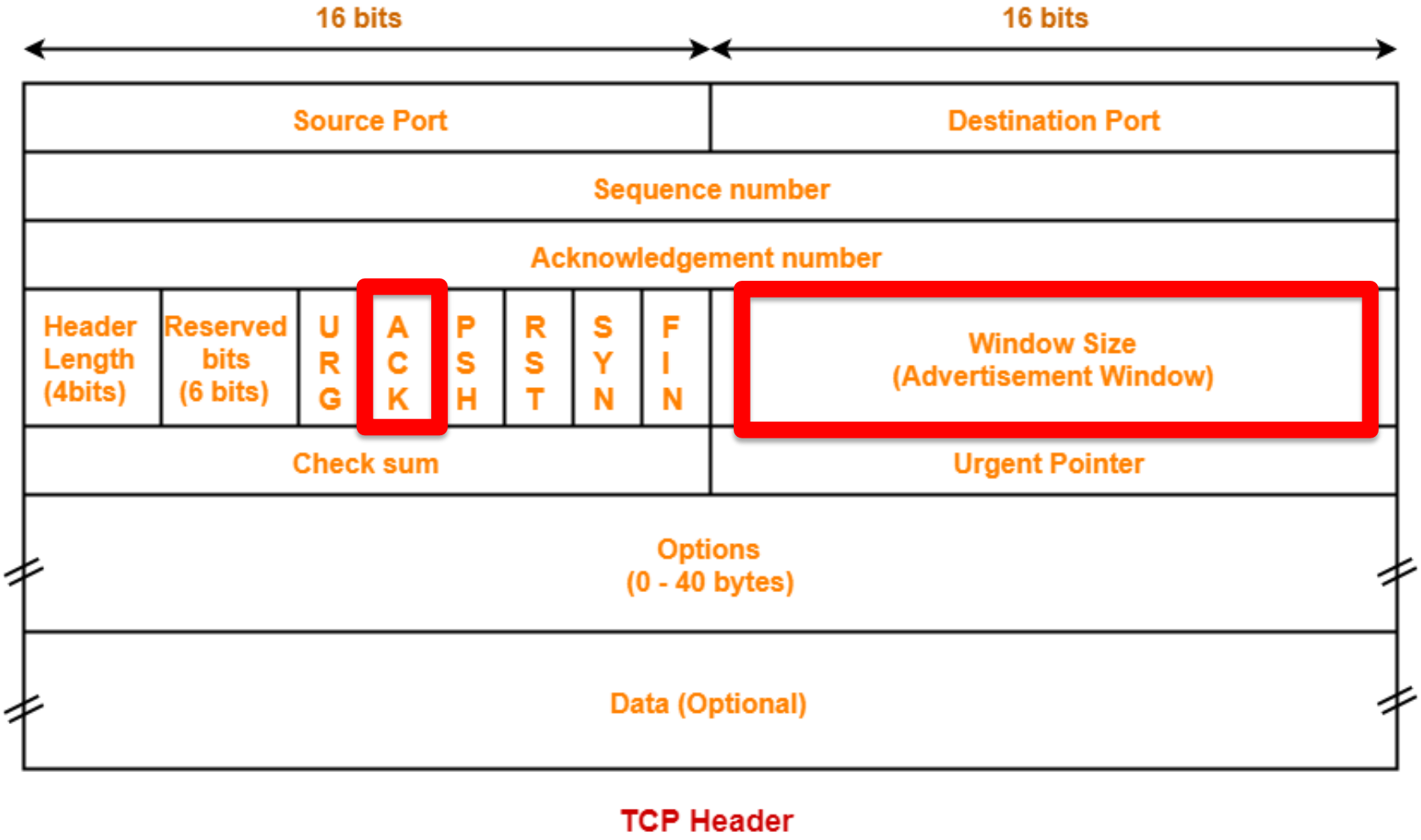
- Allow a larger amount of data "in flight" for better channel utilization, but should not exceed receiver's capacity.



Receiver Buffering: Flow control

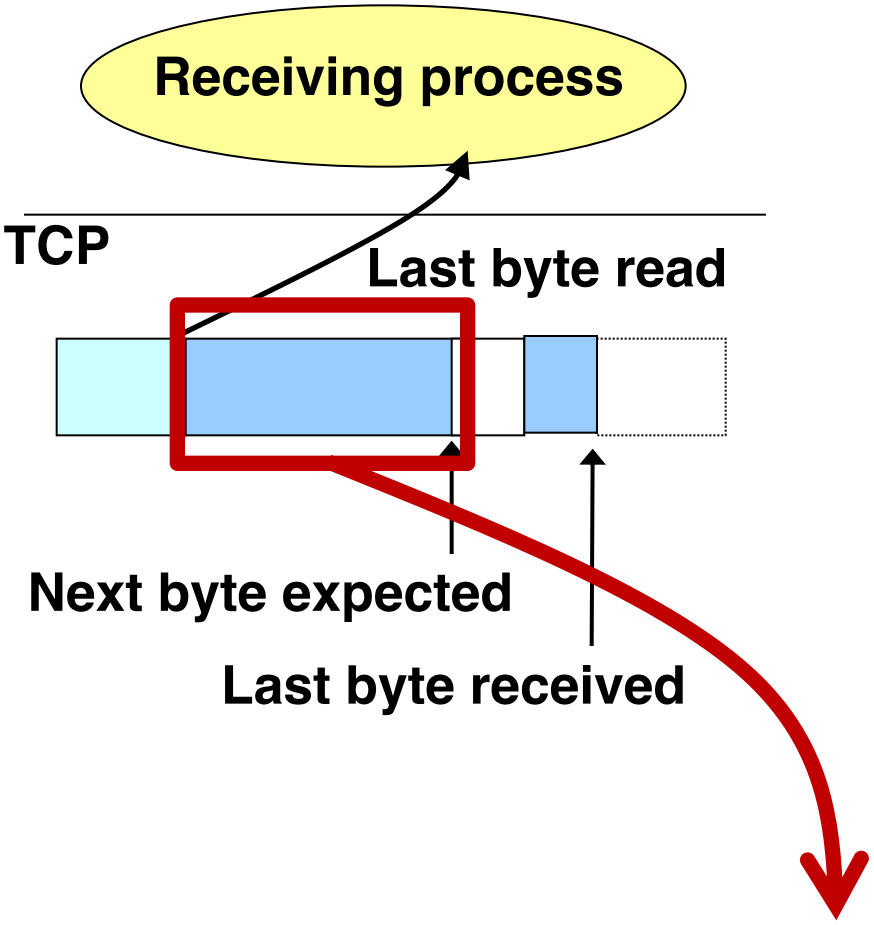
- Receive window size
  - Amount that can be sent without acknowledgment
  - Receiver must be able to store this amount of data
- Receiver tells the sender the window
  - Tells the sender the amount of free space left (i.e. sends an Advertised_Window size to the sender)

How to send the Advertised_Window size



TCP Header

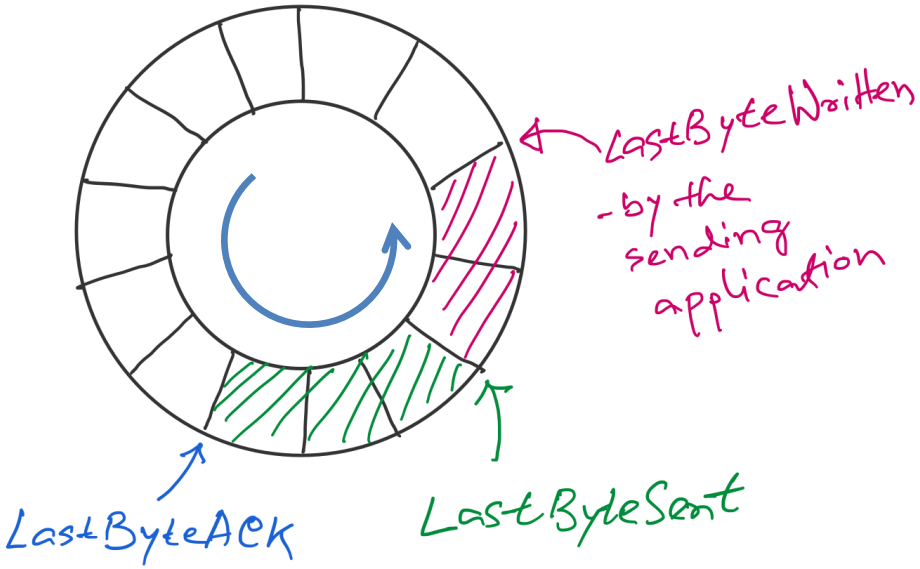
Calculate Advertised_Window (Receiver side)



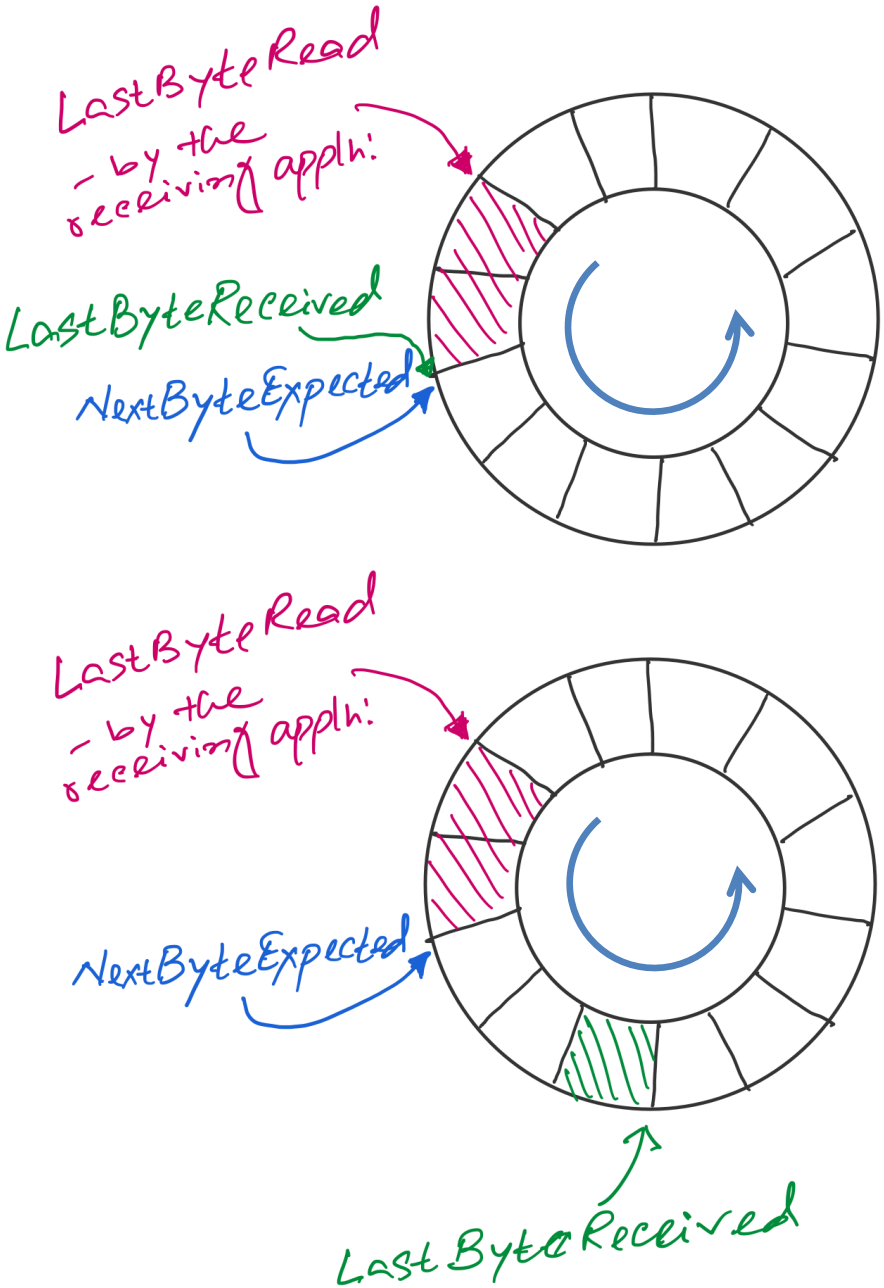

```
AdvertisedWindow = MaxRcvBuffer - (bytes used)

AdvertisedWindow = MaxRcvBuffer -
                        ((NextByteExpected-1) - LastByteRead)
```

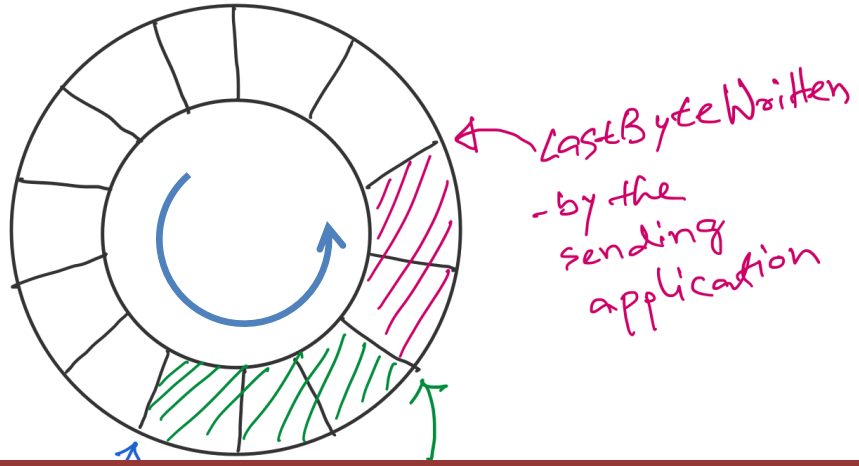
Sender buffer



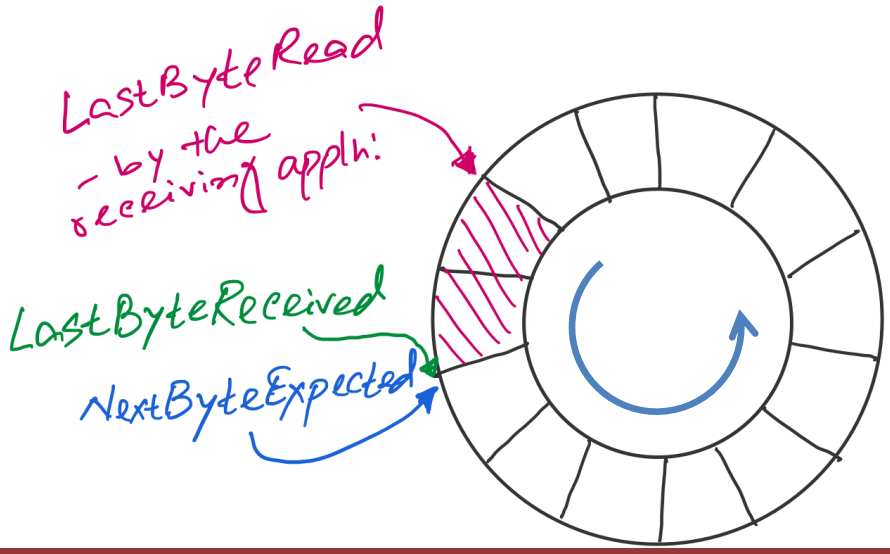
Receiver buffer



Sender buffer



Receiver buffer

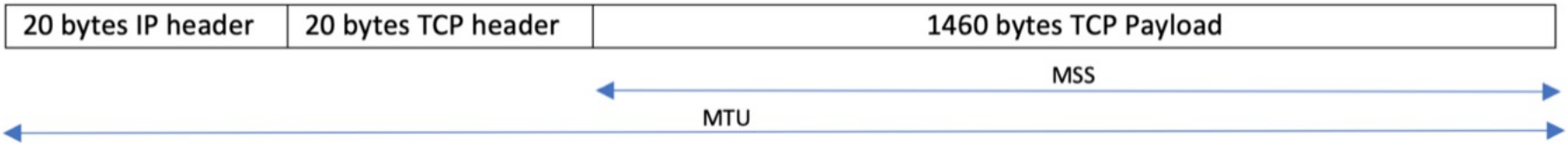


What if the AdvertisedWindow is zero?

>> Periodically send 1byte packets to break the deadlock. (uses the persist timer)

```
EffectiveWindow (calculated at the sender side)
= AdvertisedWindow - (LastByteSent - LastByteACKed)
```

Considering Maximum Segment Size (MSS)



What if…

(1) Adv_window > MSS

(2) Adv_window < MSS