Introduction to Computer Networks

TCP Reliability Support (I)

https://pages.cs.wisc.edu/~mgliu/CS640/S25/index.html

Ming Liu
mgliu@cs.wisc.edu

Outline

- Last
  - TCP Connection Management (II)
- Today
  - TCP Reliability Support (I)
- Announcements
  - Quiz3 in class on 04/03/2025

Recap: UDP Issues

- **#1: Arbitrary communication**
  - **Senders and receivers can talk to each other in any ways**
- #2: No reliability guarantee
  - Packets can be lost/duplicated/reordered during transmission
  - A checksum is not enough
- #3: No resource management
  - Each channel works as an exclusive network resource owner
  - No adaptive support for the physical networks and applications

What is the goal of TCP reliability mechanisms?

What is the goal of TCP reliability mechanisms?

Byte stream @sender = Byte stream @receiver

What is the goal of TCP reliability mechanisms?

Byte stream @sender = Byte stream @receiver

⬇

- #1: TCP segments are delivered with no loss/duplication
- #2: TCP segments are delivered in order
- #3: The sender is not over-running the receiver capability

What is the goal of TCP reliability mechanisms?

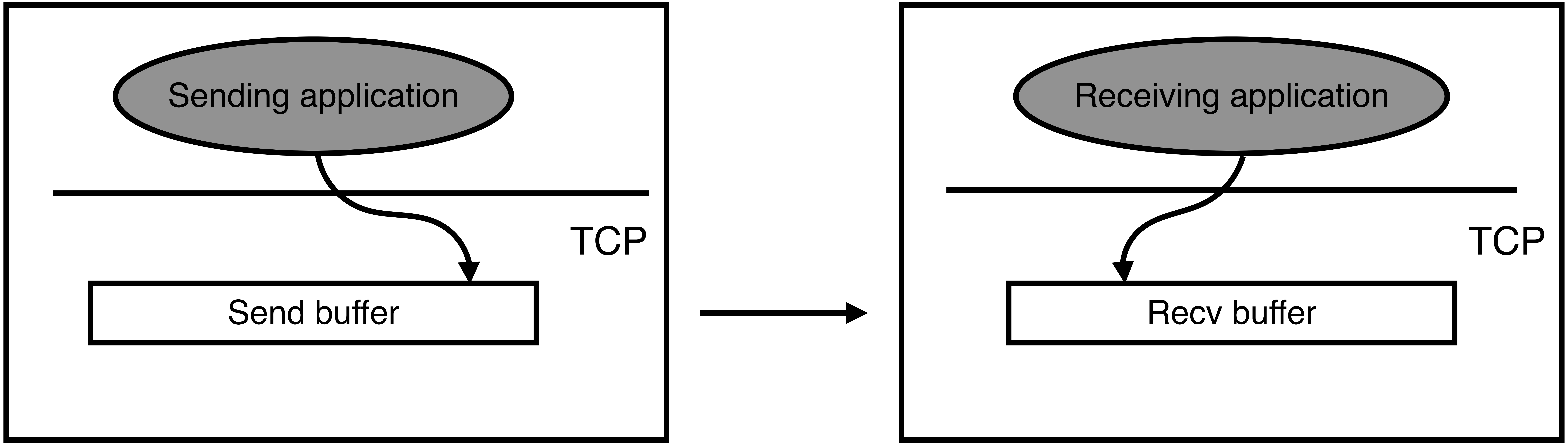
Byte stream @sender = Byte stream @receiver

⬇

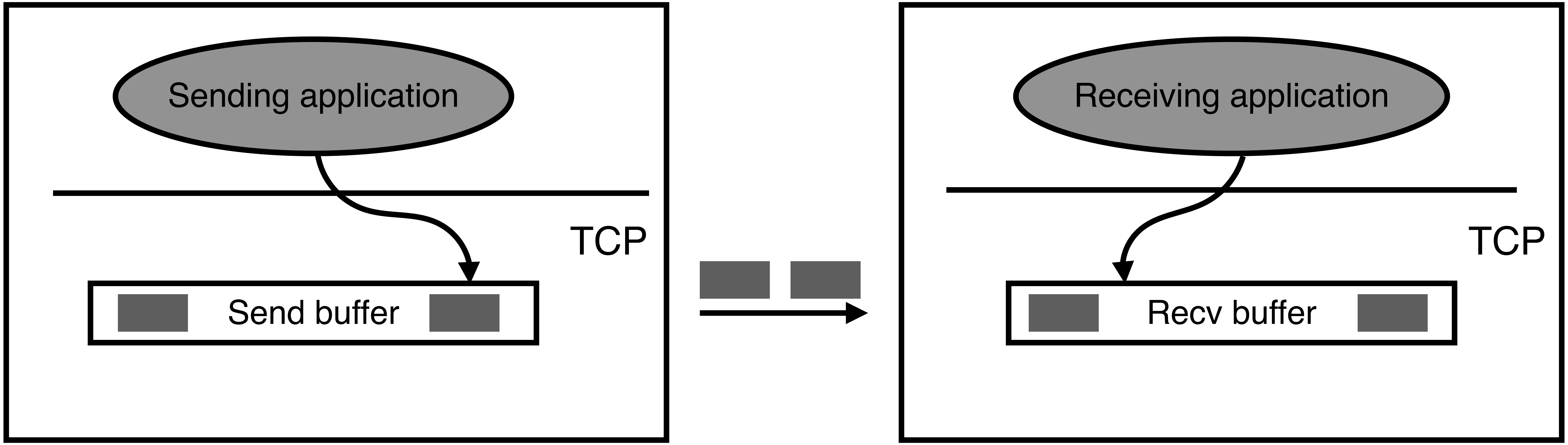
- #1: TCP segments are delivered with no loss/duplication
- #2: TCP segments are delivered in order
- #3: The sender is not over-running the receiver capability

TCP Segment: The smallest data transmission unit under TCP, consisting of a (segment) header and a data payload.

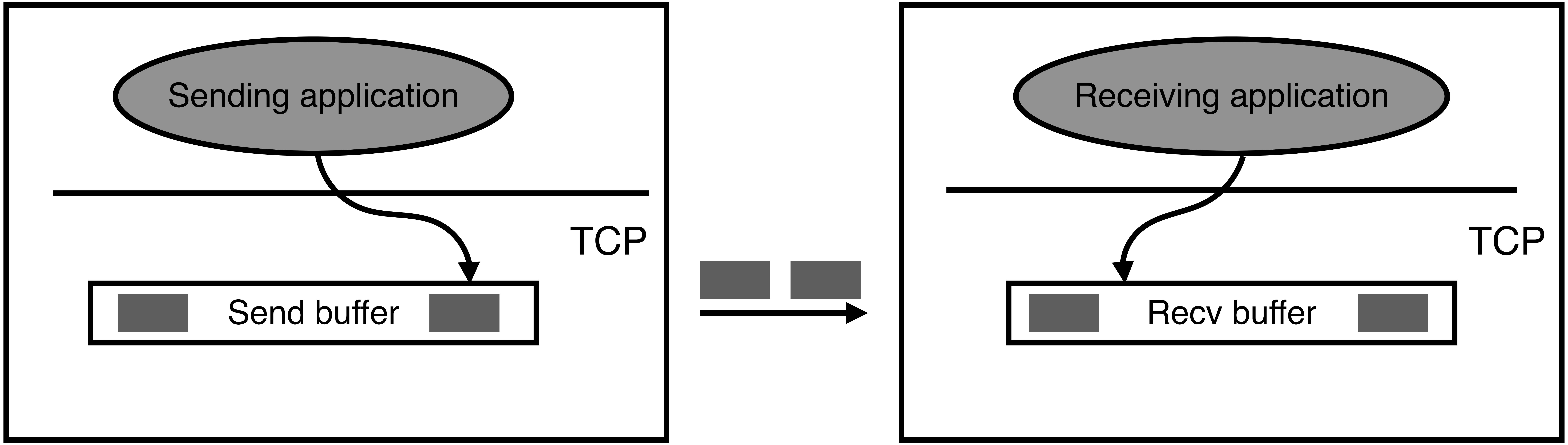
A TCP Send/Recv Example



A TCP Send/Recv Example

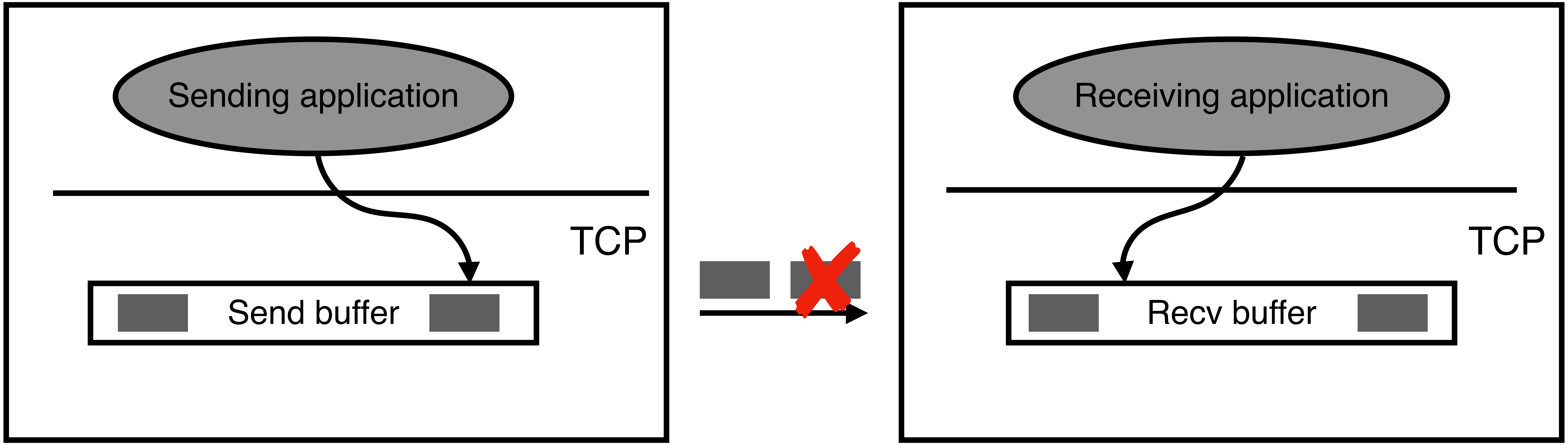


A TCP Send/Recv Example

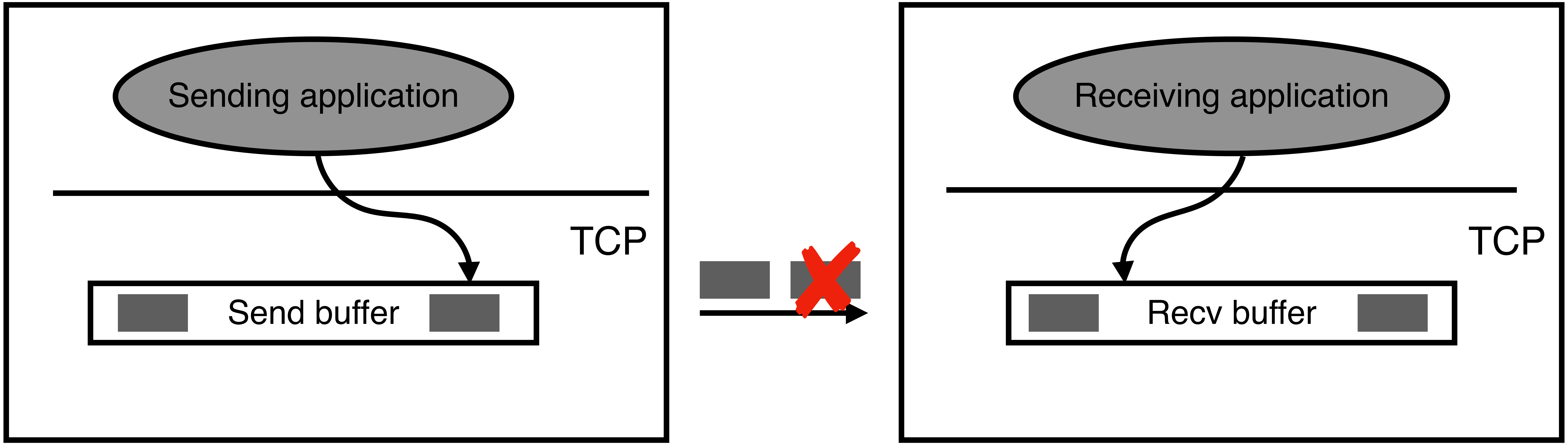


Send/Recv buffer is fixed-sized
(i.e., MaxSendBuffer and MaxRcvBuffer)

Issue #1: Segment Loss

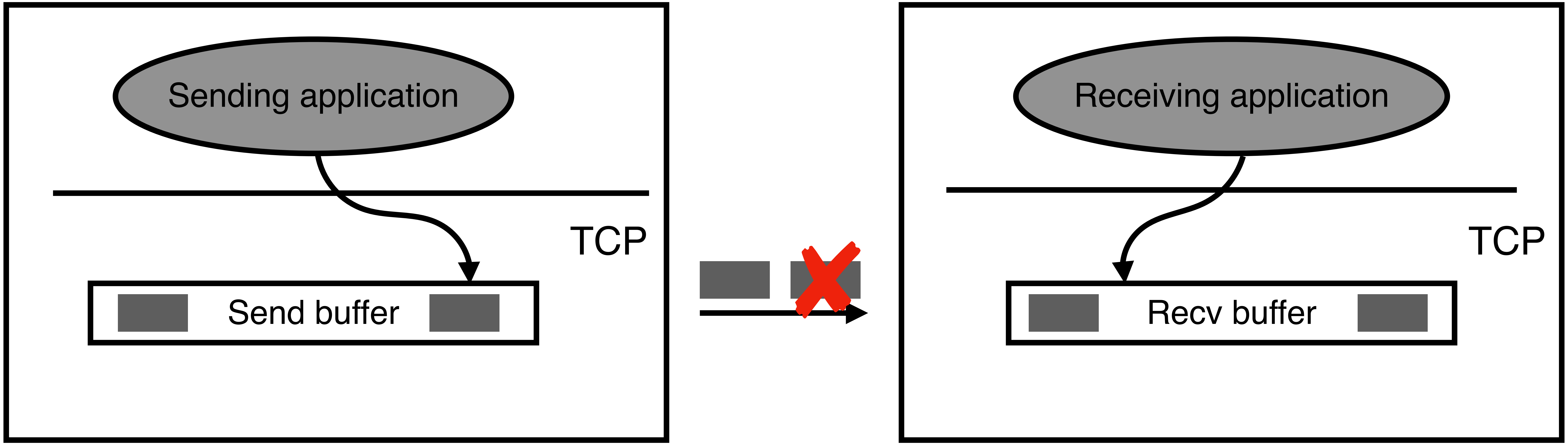


Issue #1: Segment Loss



- How do we know a segment is missing?
- How do we recover a missing segment?

Issue #1: Segment Loss



- How do we know a segment is missing? => Detection
- How do we recover a missing segment? => Correction

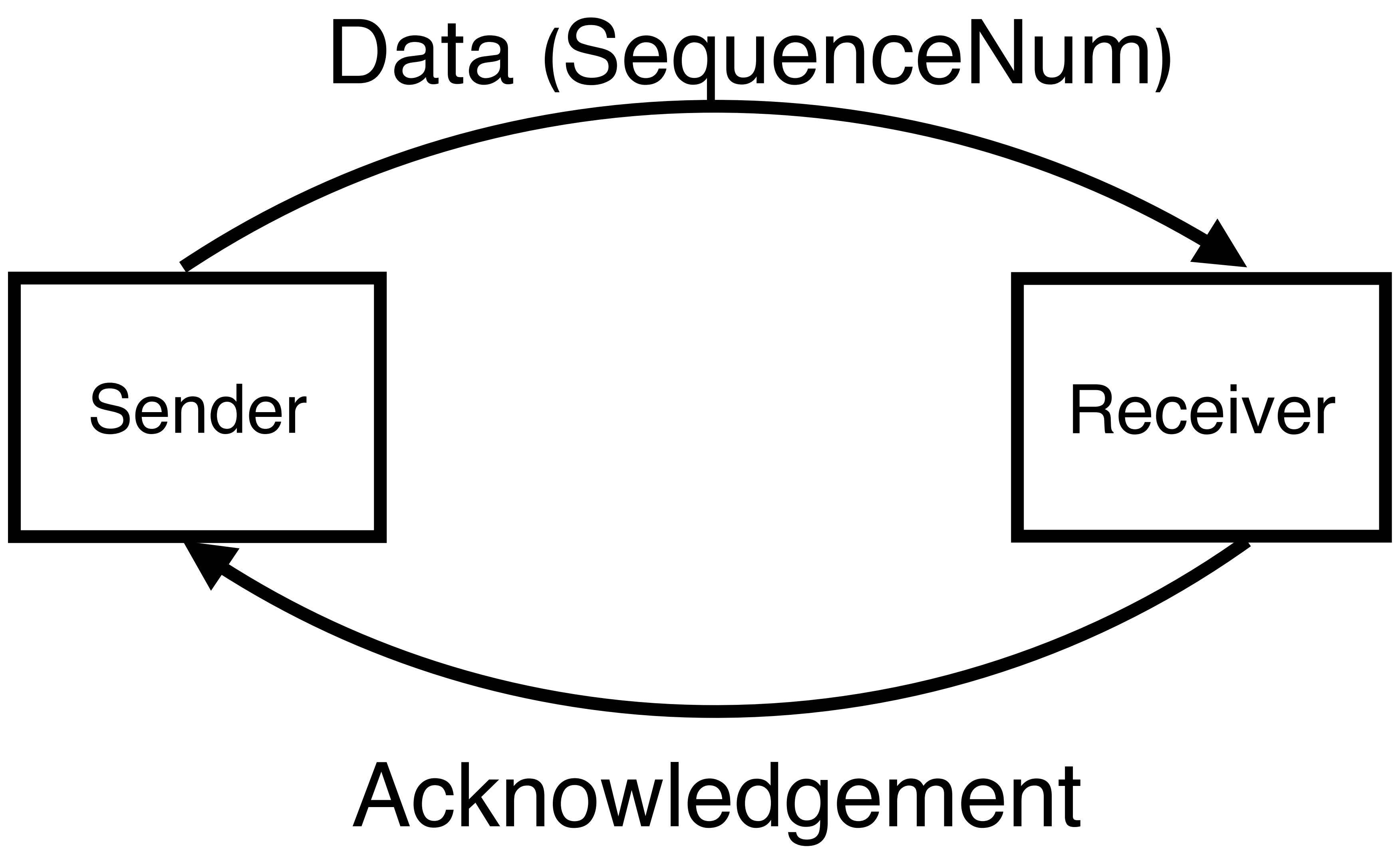
Detecting a Missing Segment

- Challenge: no in-network observability
  - We will revisit this in **L23 (TCP in-network support)**

- Solution: use host-side indirect signals
  - Sender: check if the transmitted packets are confirmed by the receiver
  - Receiver: check if the byte stream misses any segments

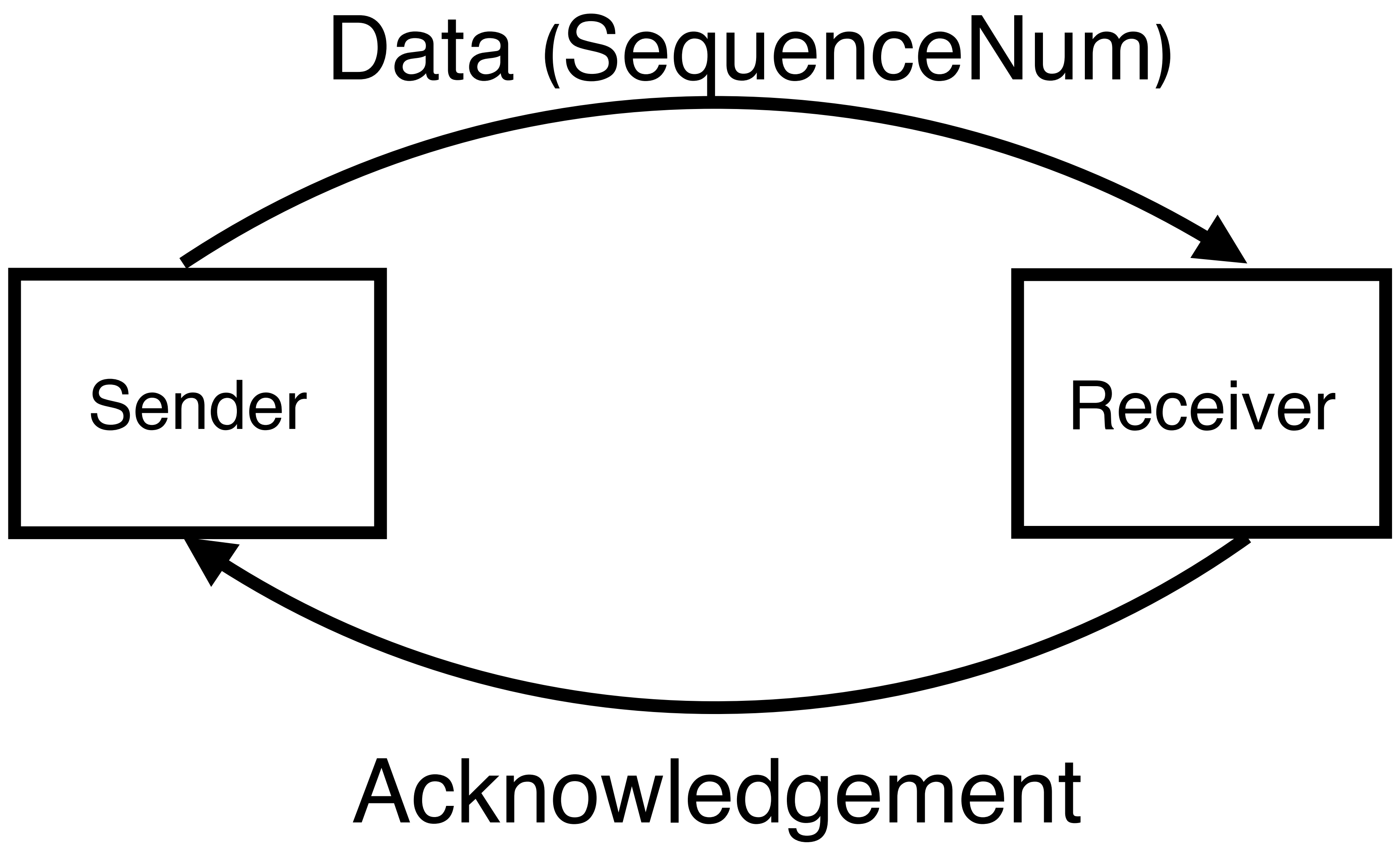
Sender-side Detection — Acknowledgment

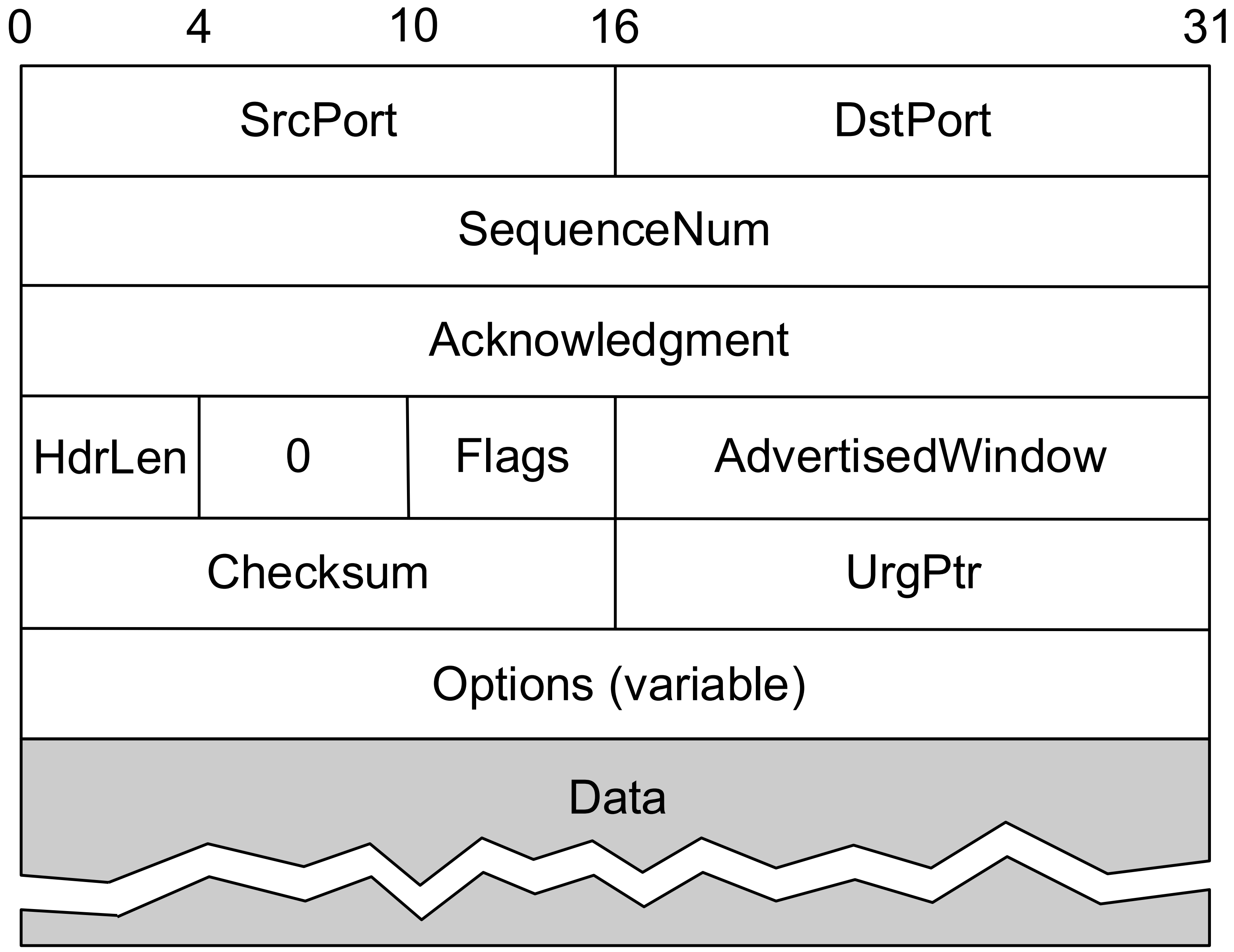
- Acknowledgment
  - Ask the receiver to send back an ACK when a segment is received
  - A missing ACK indicates a missing segment



Sender-side Detection — Acknowledgment

- Acknowledgment
  - Ask the receiver to send back an ACK when a segment is received
  - A missing ACK indicates a missing segment
- TCP realization
  - Making an ACK segment is simply changing a field in the header
  - Data can be piggybacked in ACKs





SrcPort | DstPort

SequenceNum

Acknowledgment

HdrLen | 0 | Flags | AdvertisedWindow

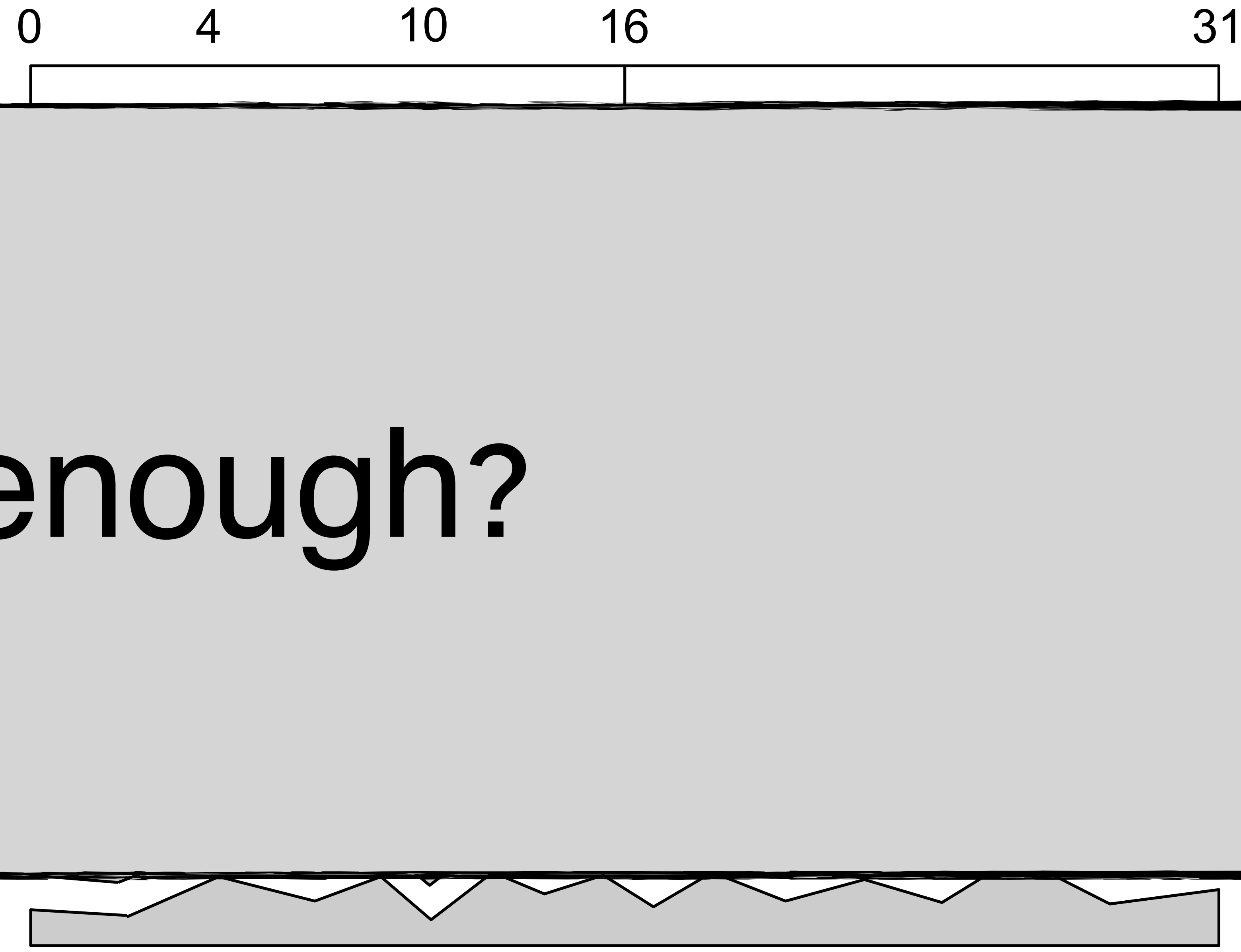
Checksum | UrgPtr

Options (variable)

Data

Sender-side Detection — Acknowledgment

- Acknowledgment
  - Ask the receiver to send back an ACK when a segment is received
  - A missing ACK indicates a missing segment
- TCP realization
  - Making an ACK segment is simply changing a field in the header
  - Data can be piggybacked in ACKs

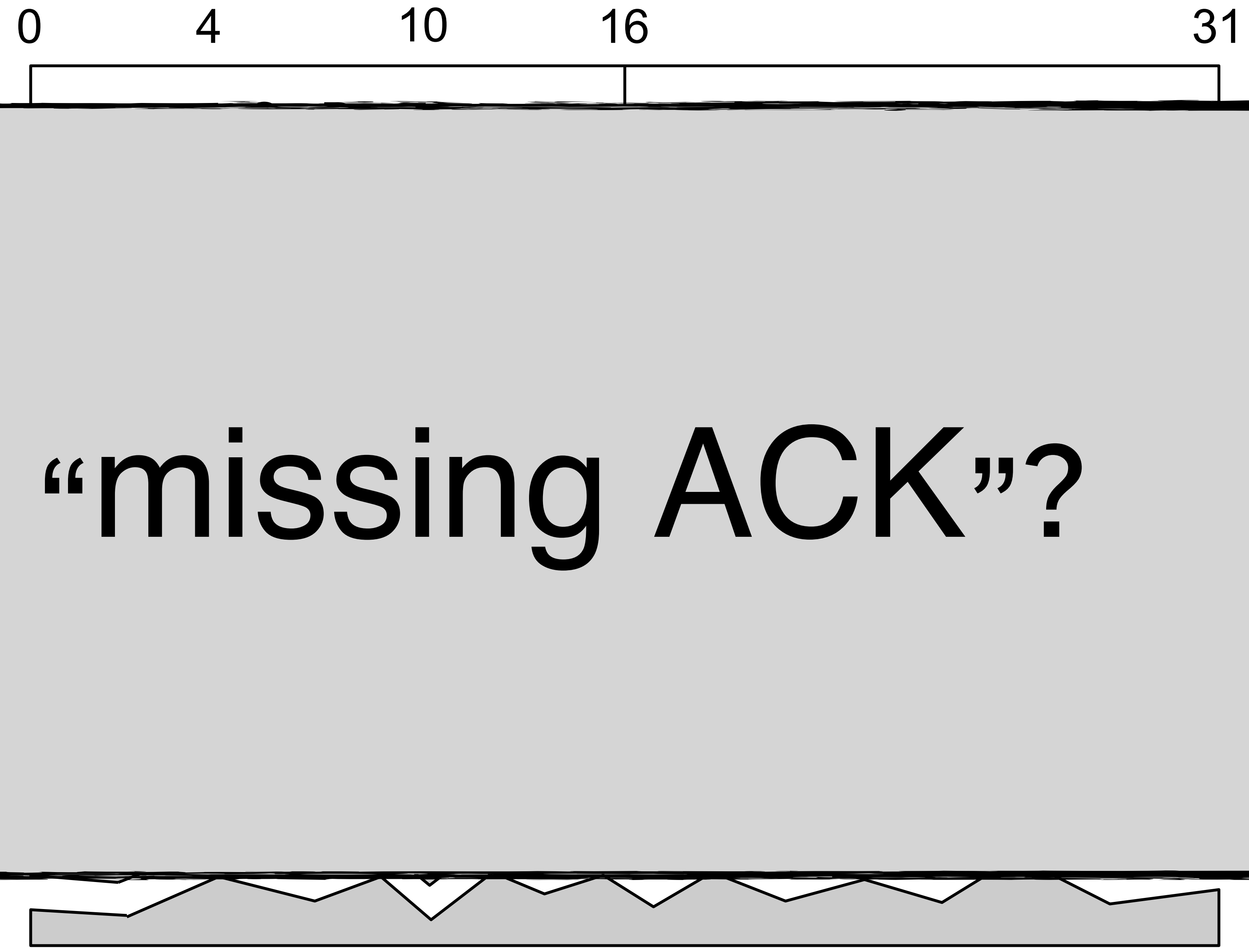


Is this good enough?

Acknowledgement

Sender-side Detection — Acknowledgment

- Acknowledgment
  - Ask the receiver to send back an ACK when a segment is received
  - A missing ACK indicates a missing segment
- TCP realization
  - Making an ACK segment is simply changing a field in the header
  - Data can be piggybacked in ACKs



How can we define a “missing ACK”?

Acknowledgement

Sender-side Detection — Timeout

- Timeout
  - A signal that a segment that was sent but has not received its ACK within a specified time frame (threshold)

- A missing segment
  - The corresponding ACK is not received before the timeout is triggered

Sender-side Detection — Timeout

- Timeout
  - A signal that a segment that was sent but has not received its ACK within a specified time frame (threshold)
- A missing segment
  - The corresponding ACK is not received before the timeout is triggered

But how to set the timer (threshold)?

The Art of Timeout Threshold

- This is tricky because the network condition is dynamic

- Approach: use RTTs to estimate the timeout period
  - RTT = the delay between transmission and receipt of packets between the sender and the receiver
  - Measure the RTT of each segment online and dynamically adjust the threshold
  - One RTT is not sufficient since we need to capture the network dynamics

EWMA for RTT Estimation

- EWMA = Exponentially Weighted Moving Average

- EWMA mechanisms
  - #1: Measure SampleRTT for each segment/ACK pair
  - #2: Compute the weighted average of RTT
    - $EstimatedRTT = \alpha \times EstimatedRTT + \beta \times SampleRTT, where\ \alpha + \beta = 1$
    - $0.8 \leq \alpha \leq 0.9$
    - $0.1 \leq \beta \leq 0.2$
  - #3: Set timeout based on the EstimatedRTT
    - $TimeOut = 2 \times EstimatedRTT$

Summary

- Today
  - TCP reliability support (I)

- Next lecture
  - TCP reliability support (II)