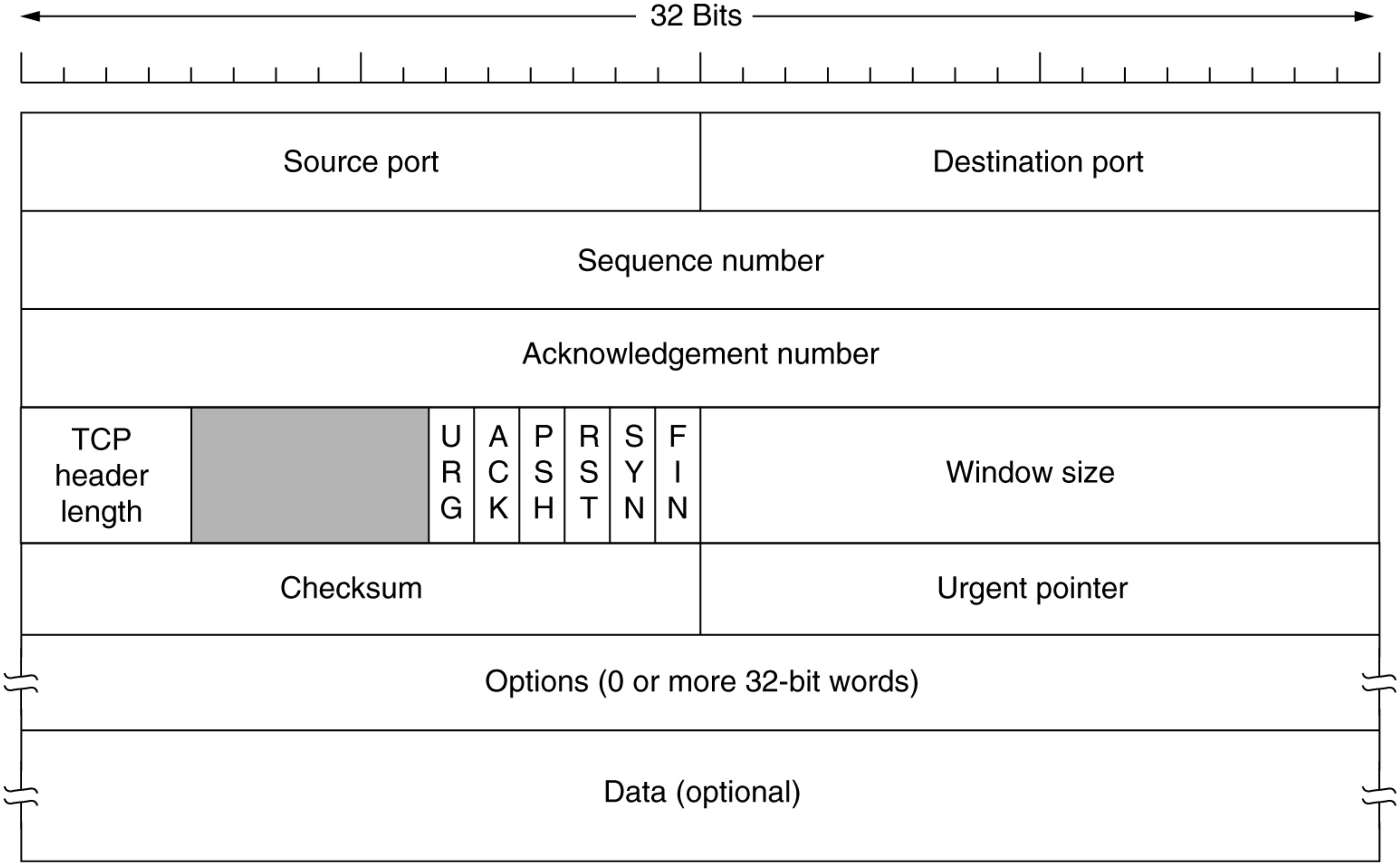
Computer Communication Networks

TCP Flow Control

Review: TCP basics

- Services offered by TCP
  - use services offered by IP
  - TCP protocol mechanisms
    - to fill the gap
- TCP connection management
  - TCP connection establishment
  - TCP connection release

TCP packet header



32 Bits

Source port

Destination port

Sequence number

Acknowledgement number

TCP header length

URG

ACK

PSH

RST

SYN

FIN

Window size

Checksum

Urgent pointer

Options (0 or more 32-bit words)

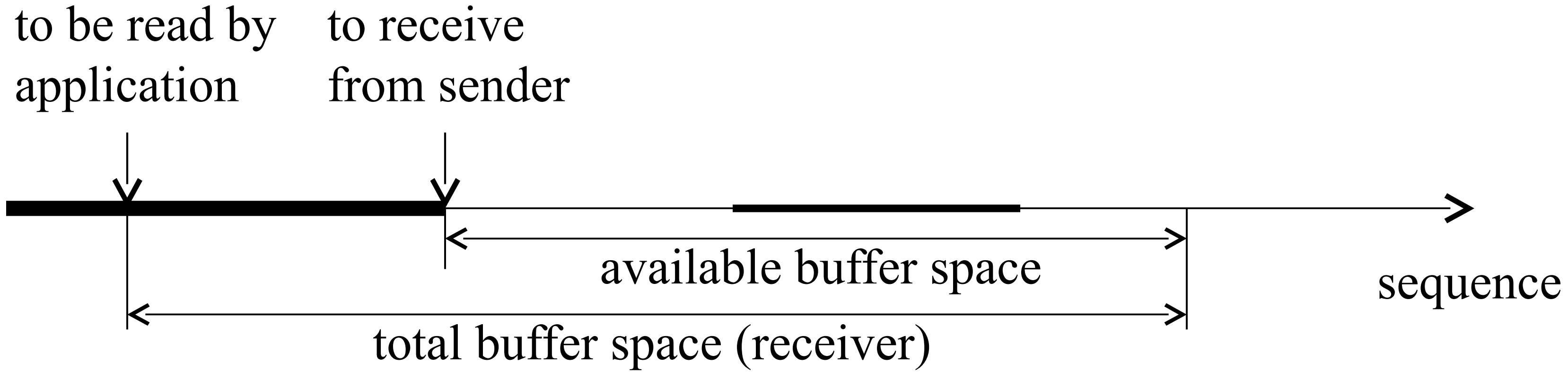
Data (optional)

TCP flow control

- Purpose
  - avoid a fast sender to overflow a slow receiver
    - overflow: exceed receiver’s available buffer space
- Approach
  - let the receiver tell
    - how much available space I have
  - receiver window size (16-bit) in TCP header
    - advertised window size
    - in bytes!

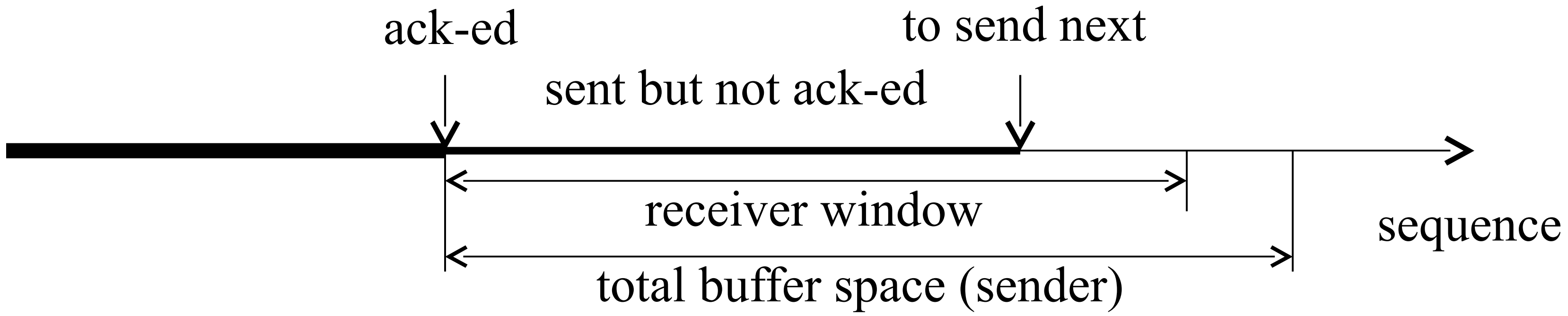
TCP receiver's view

- Sequence space
  - acknowledgment number
    - the next ***continuous*** byte to receive from the sender
  - receiver window
    - available buffer space at receiver



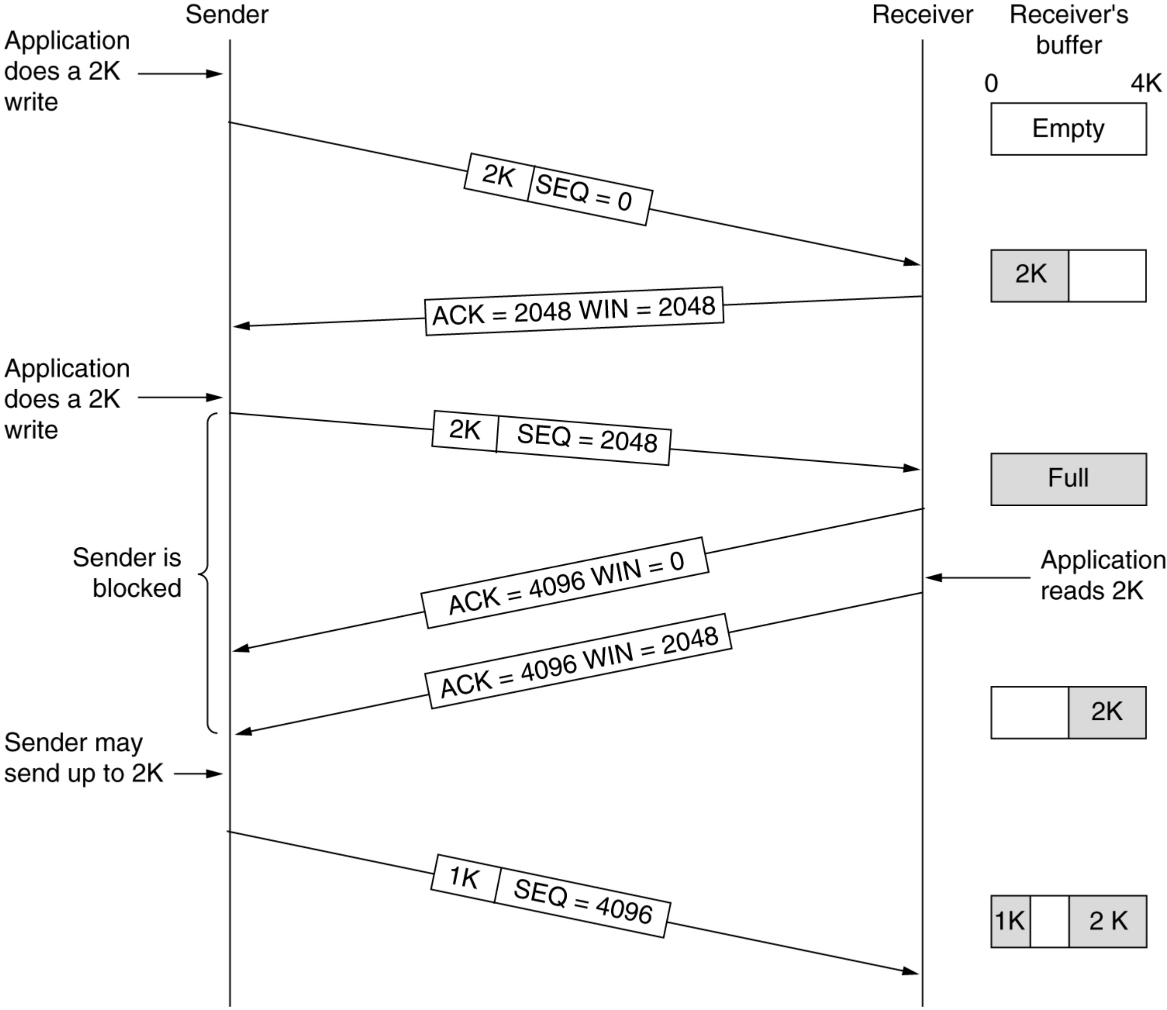
TCP sender’s view

- Sequence space
  - sequence number
    - the first byte in the payload
  - sender window
    - min {receiver window, total buffer space}



Between sender and receiver

- Window control
  - – sliding window
    - acknowledgment
  - – variable window
    - window size



Window and sequence space

- Window space (16-bit)
  - maximum window size 2^{16}-1: ~64K bytes!
  - achievable throughput limit: win/rtt
- Sequence space (32-bit)
  - t_1: time to consume the sequence space
  - t_2: time a packet can “live” in the network
  - $t_1 > t_2$: no reuse; $t_1 > 2\ t_2$: always ordered
- Explore further: TCP large window

Summary

- TCP flow control
  - purpose
  - mechanism
    - sliding variable window: seqno, ackno, win
- Explore further
  - TCP large window
  - in tcpdump (or Ethereal/Wireshark)
    - time sip:spt > dip:dpt: P **144**:192 (48) **ack** 321 **win** 16022

Next

- TCP error control