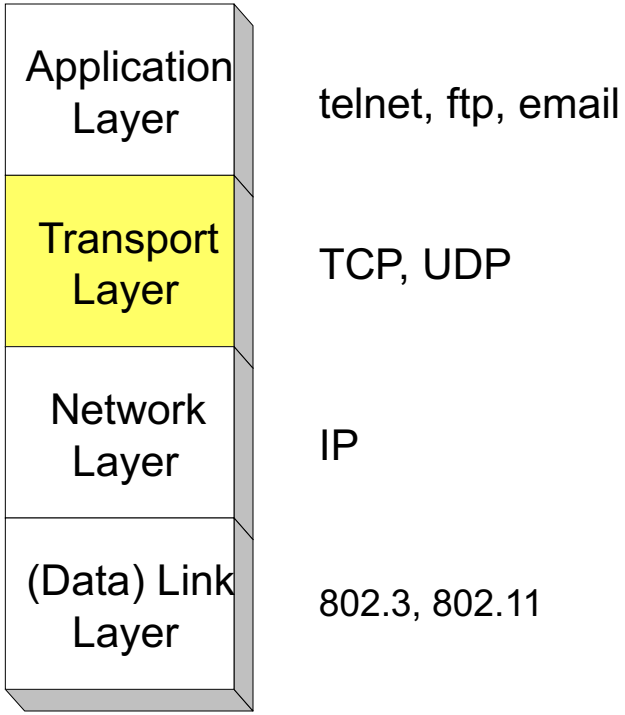
Transport layer: Reliable Transmission

ECE 50863 – Computer Network Systems

Transport Layer

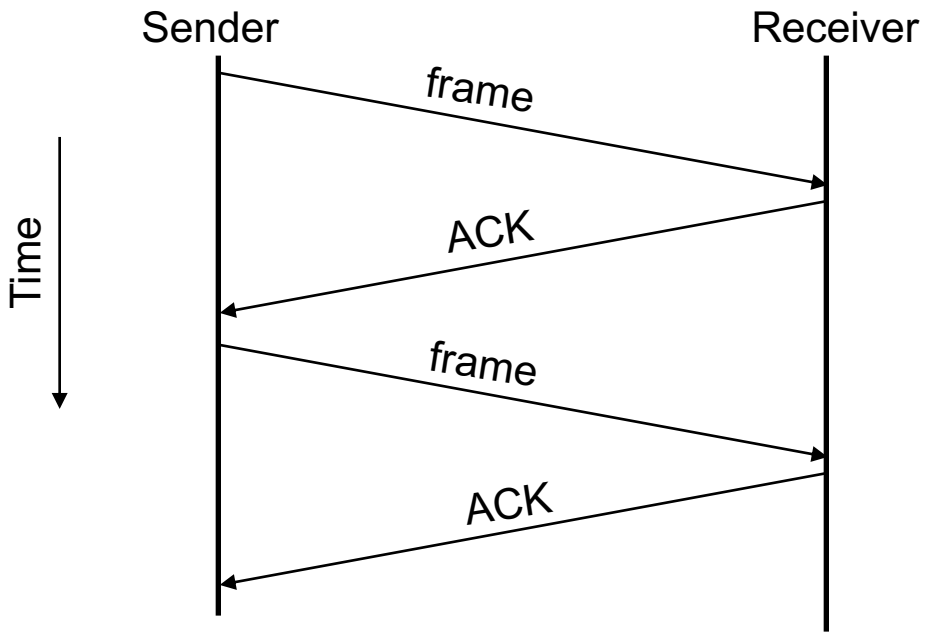


Reliable Transmission

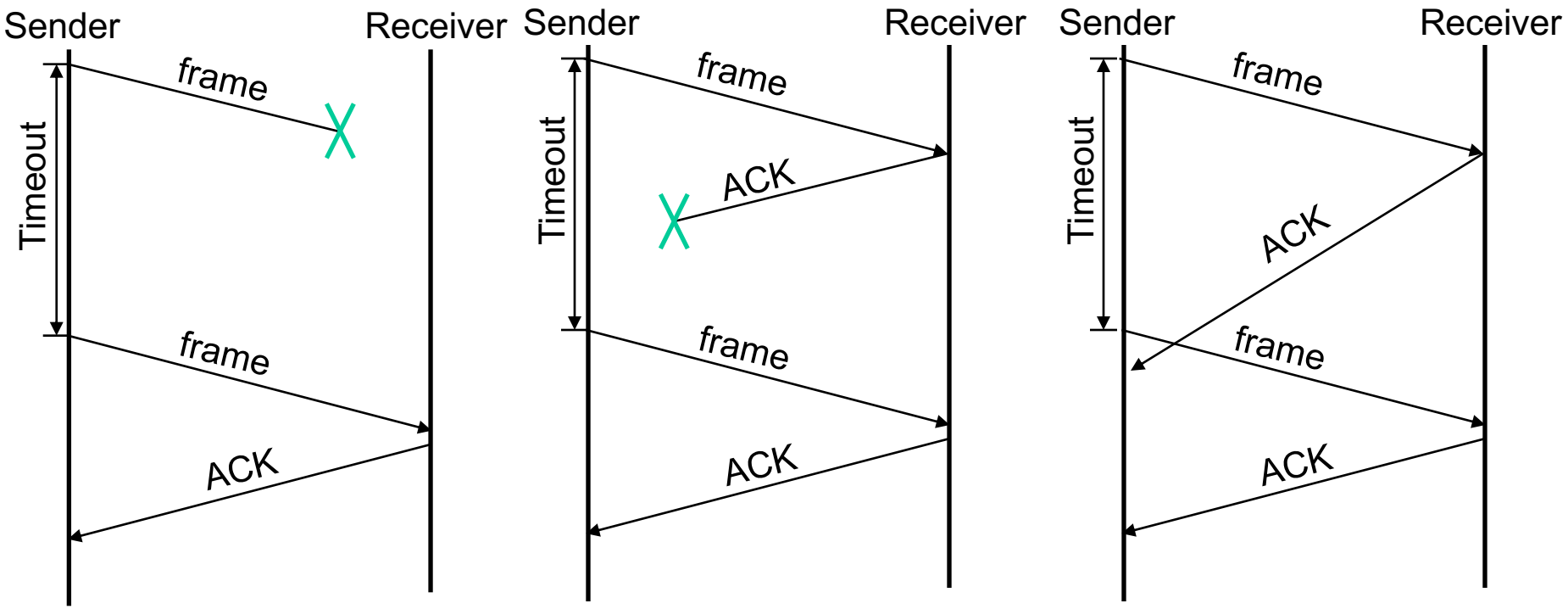
- Use two basic techniques:
  - Acknowledgements (ACKs)
  - Timeouts
- Two examples:
  - Stop-and-go
  - Sliding window

Stop-and-Go

- Receiver: send an acknowledge (ACK) back to the sender upon receiving a packet (frame)
- Sender: excepting first packet, send a packet only upon receiving the ACK for the previous packet



What Can Go Wrong?



Frame lost → resent it on Timeout

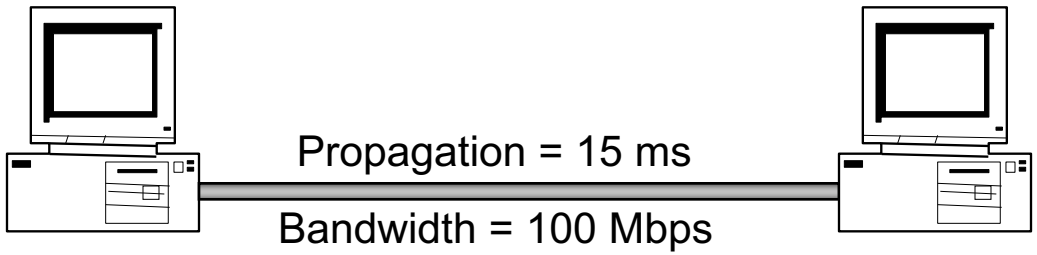
ACK lost → resent packet

Need a mechanisms to detect duplicate packet

ACK delayed → resent packet

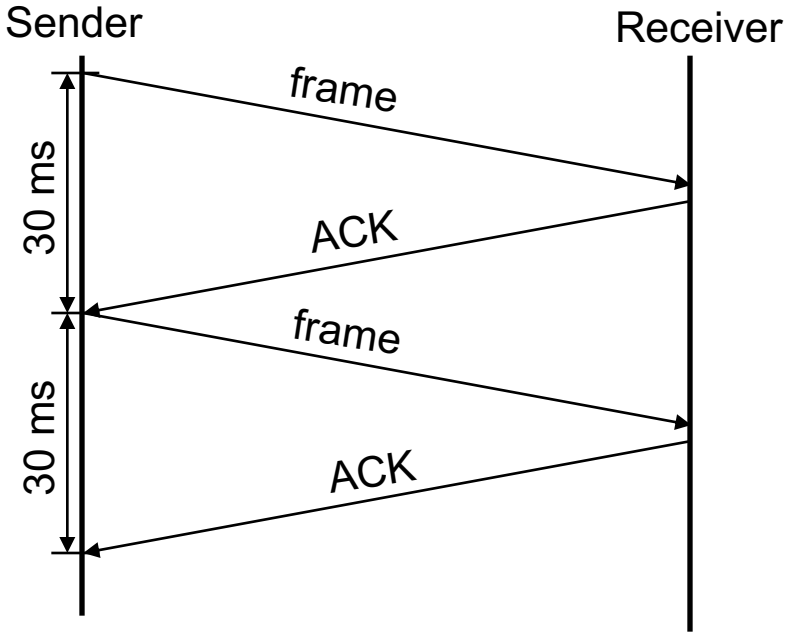
Stop-and-Go Disadvantage

- May lead to inefficient link utilization
- Example: assume
  - One-way propagation = 15 ms
  - Bandwidth = 100 Mbps
  - Packet size = 1000 bytes → transmit = $(8*1000)/10^8$ = 0.08ms
  - Neglect queue delay → Latency = approx. 15 ms; RTT = 30 ms

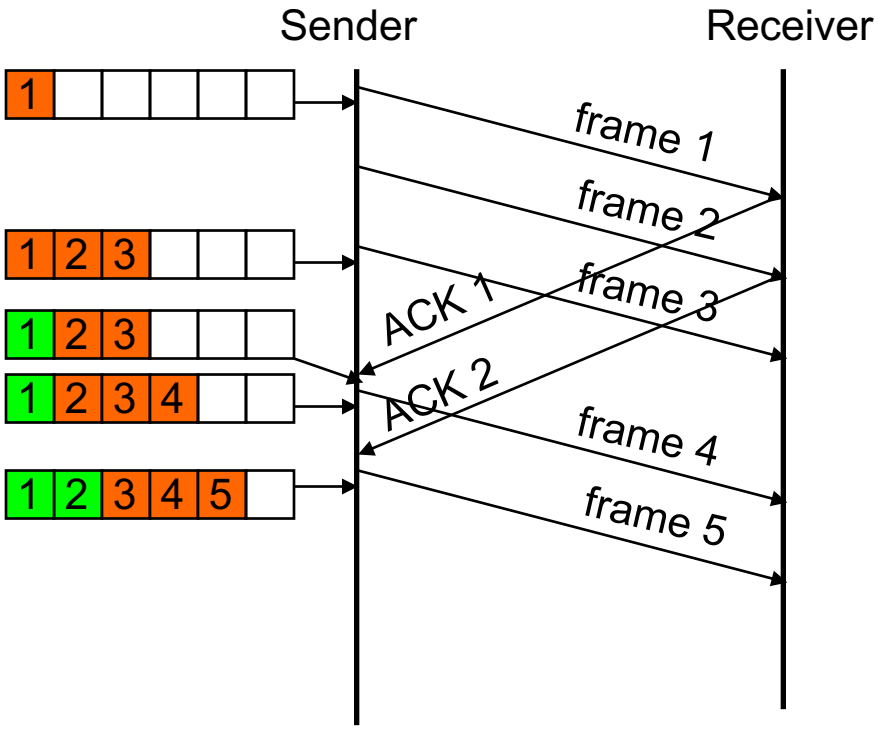


Stop-and-Go Disadvantage (cont'd)

- Send a message every 30 ms → Throughput = (8*1000)/0.03 = 0.2666 Mbps
- Thus, the protocol uses less than 0.3% of the link capacity!



Sliding Window Approach



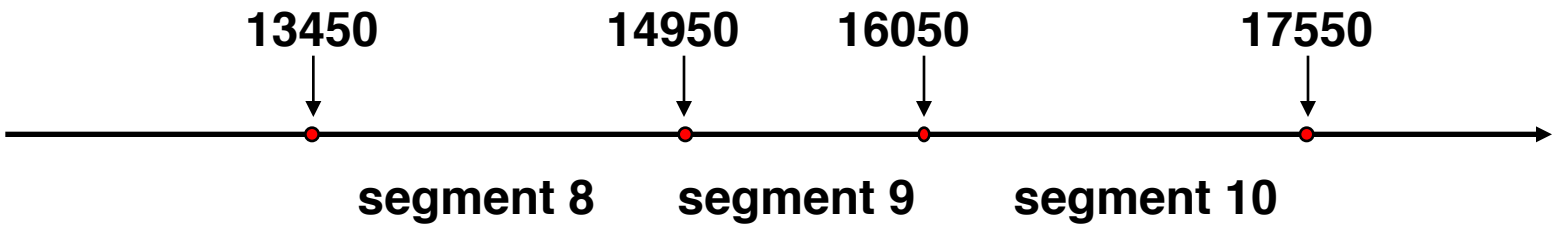
Sliding window protocol

Send multiple packets without waiting for ACK.

Sliding window size: Max number of packets that can be sent without ACK being received (3 in figure)

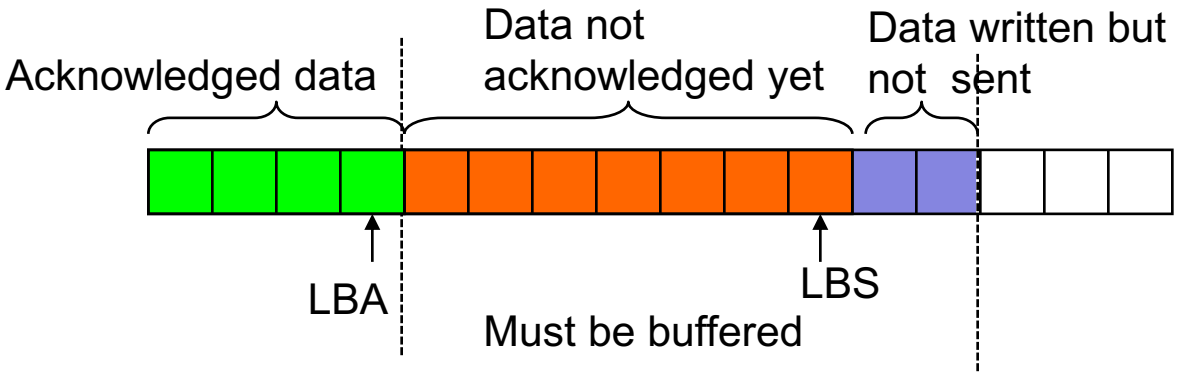
Sequence Numbers in TCP

- Each byte in byte stream is numbered.
  - 32 bit value
- TCP breaks up the byte stream in segments
- Each segment has a sequence number.
  - Indicates where it fits in the byte stream



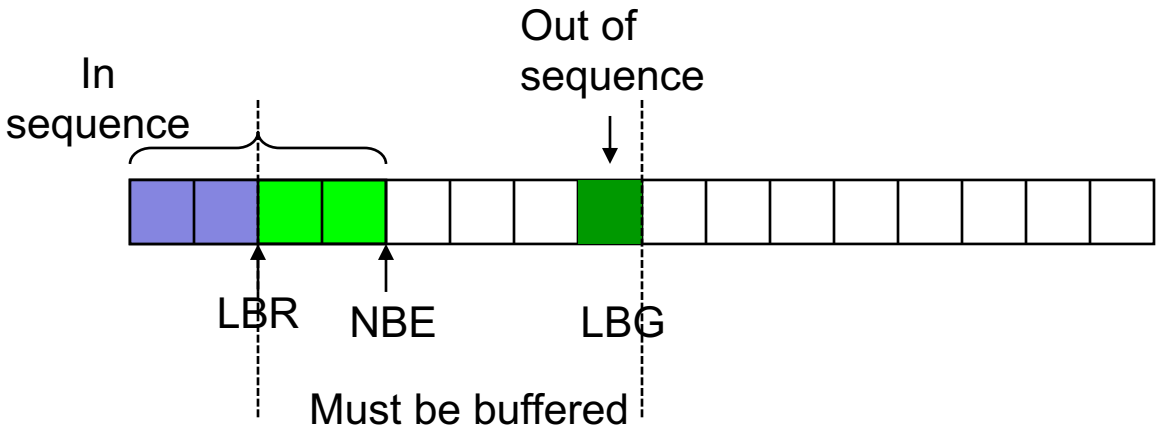
Sliding Window Protocol: Sender

- TCP: operates at a byte stream level (rather than packet level)
- Sender maintains a window of sequence numbers
  - SWS (sender window size) – maximum data that can be sent without receiving an ACK
  - LBA (last byte acknowledged)
  - LBS (last byte sent)
- TCP sender side socket buffer:
  - Data sent but not acknowledged
  - Data written by application but not sent yet



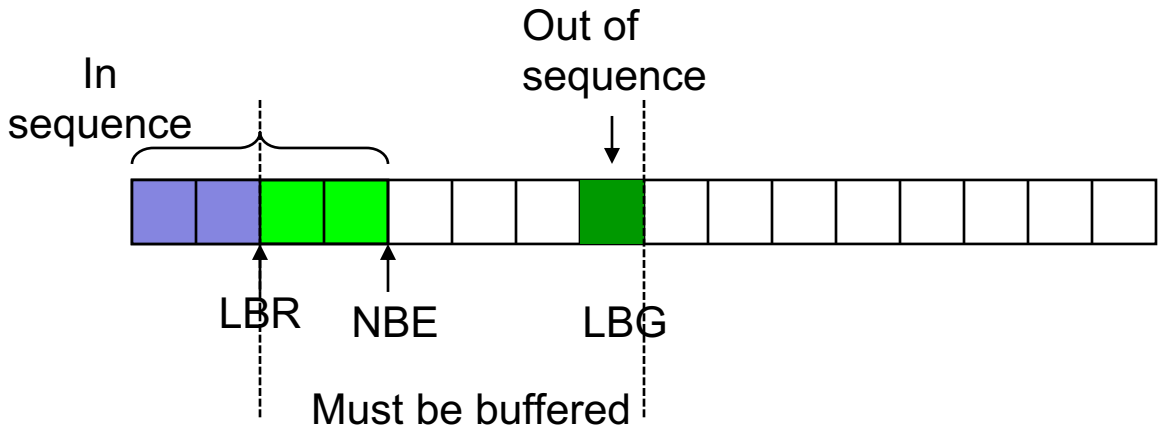
Sliding Window Protocol: Receiver

- TCP maintains a receive socket buffer
  - Stores data that arrived out-of-order (cannot be given to application yet)
  - Also stores data that arrived in-order but not yet read by application (slow process)
- Receiver maintains a window of sequence numbers
  - NBE (next byte expected – all previous bytes received in sequence
  - LBR (last byte read by application)
  - LBG (last byte got by receiver)



Sliding Window Protocol: Receiver

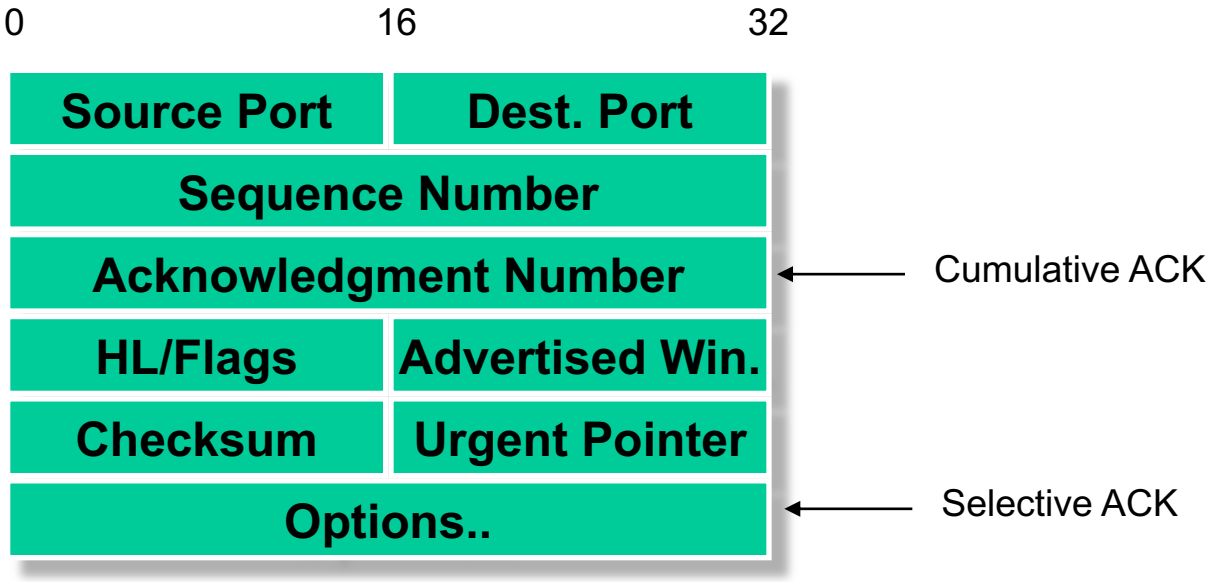
- If incoming byte < NBE
  - Discard packet
- Else
  - Accept packet (provided it fits in the buffer)
  - ACK largest byte that all previous bytes were received
    - Cumulative ACK scheme



Cumulative vs. Selective ACK

- Imagine receiver already got and ACK'd packets 1,2 and 3. (Packets for simplicity).
- Next receiver gets packet 6.
- Cumulative ACK scheme:
  - ACK packet 3 again (or say 4 is expected)
  - Approach taken by basic versions of TCP.
- Selective ACK
  - In addition to ACKing packet 3, also ACKs packet 6
  - This is an extra option added to TCP header (would increase header size).
  - Upto 3 out-of-order segments can be acknowledged in addition to cumulative ACK.

TCP Header


0
16
32
Source Port
Dest. Port
Sequence Number
Acknowledgment Number
HL/Flags
Advertised Win.
Checksum
Urgent Pointer
Options..
Cumulative ACK
Selective ACK