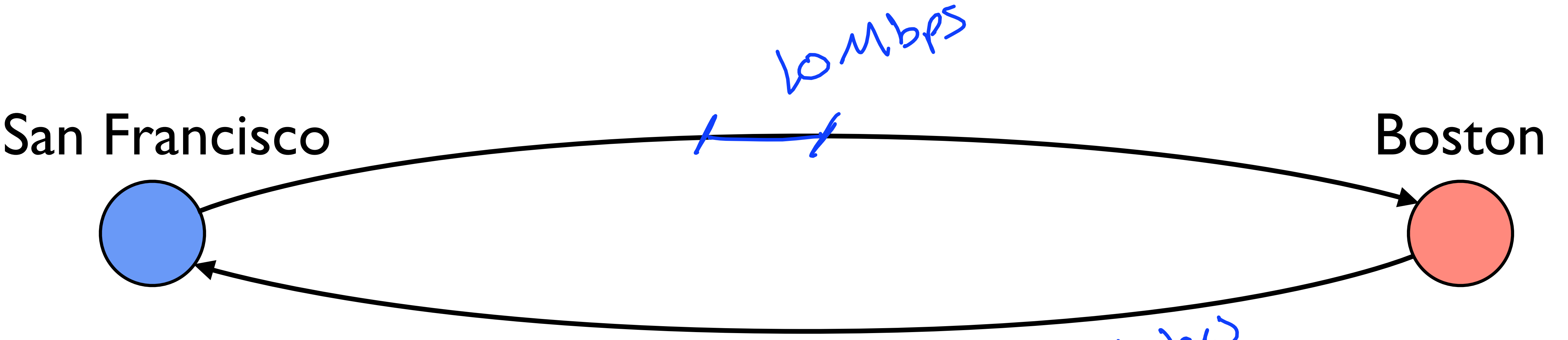
Flow Control II

Sliding Window

Stop and Wait

- At most one packet in flight at any time
- Sender sends one packet
- Receiver sends acknowledgment packet when it receives data
- On receiving acknowledgment, sender sends new data
- On timeout, sender resends current data
- Use 1-bit counter to detect duplicates

Stop and Wait Problem

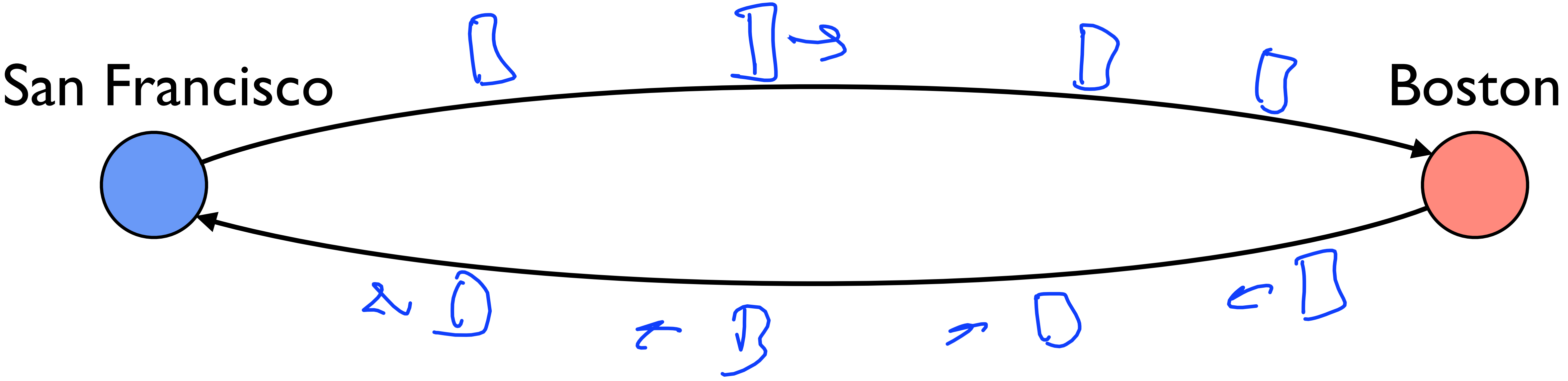


1.5kB or 12 kbits

$1000/50 = 20 \text{pps} \cdot 12 \text{kbits} = \underline{240 \text{kbps}}$

Bottleneck is 10Mbps
RTT is 50ms

Sliding Window

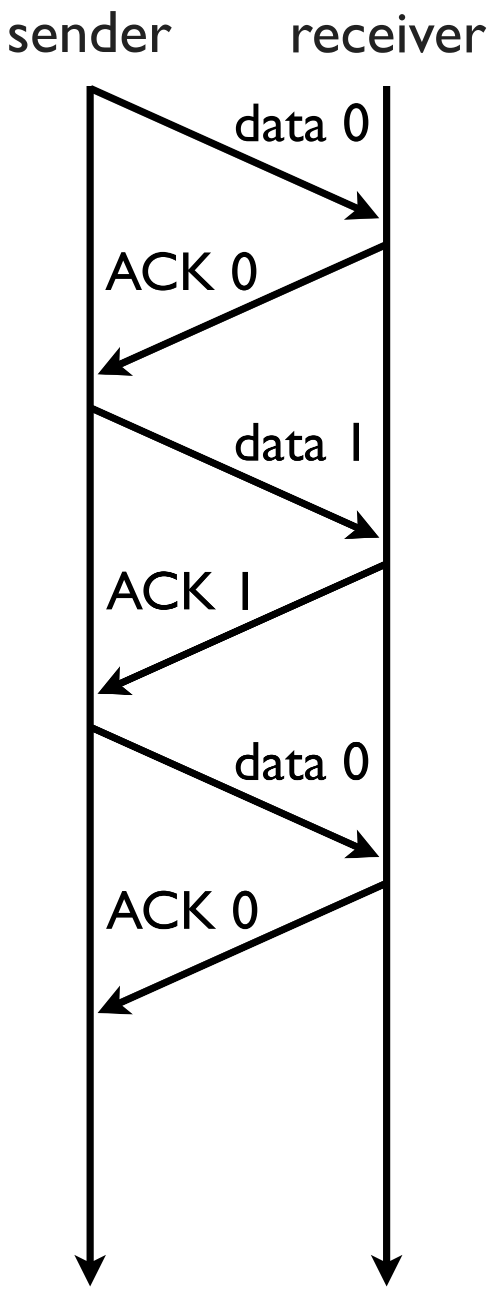


Bottleneck is 10Mbps
RTT is 50ms

- Generalization of stop-and-wait: allow multiple un-acked segments
- Bound on number of un-acked segments, called *window*
- Can keep pipe full

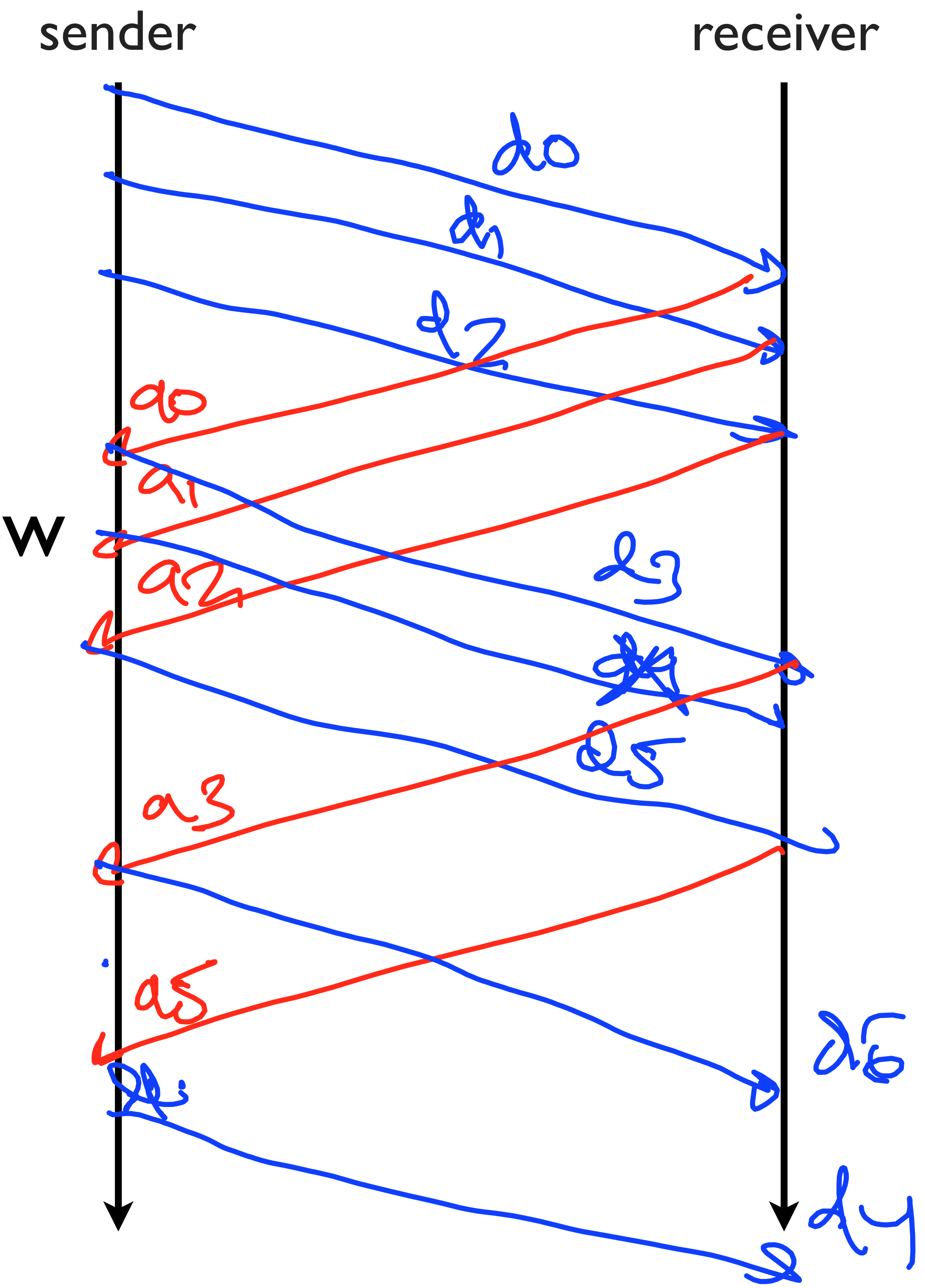
Example Execution

Stop and Wait



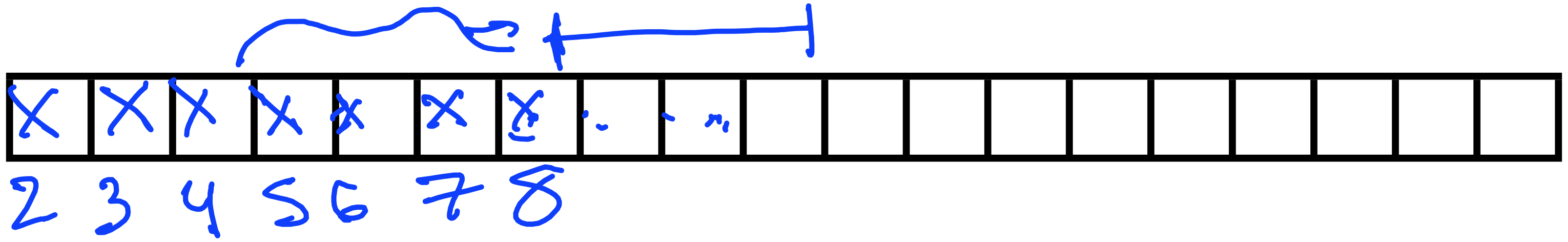
Sliding Window

(3)



Sliding Window Sender

- Every segment has a sequence number (SeqNo)
- Maintain 3 variables
  - Send window size (SWS) = 3
  - Last acknowledgment received (LAR) = 4
  - Last segment sent (LSS)
- Maintain invariant: (LSS - LAR) ≤ SWS
- Advance LAR on new acknowledgment
- Buffer up to SWS segments



Sliding Window Receiver

- Maintain 3 variables
  - Receive window size (RWS)
  - Last acceptable segment (LAS)
  - Last segment received (LSR)
- Maintain invariant: $(\boxed{\text{LAS}} - \underline{\text{LSR}}) \le \underline{\text{RWS}}$
- If received packet is < LAS, send acknowledgment
  - Send *cumulative* acks: if received 1, 2, 3, 5, acknowledge 3
  - NOTE: TCP acks are next *expected* data (e.g., ack 4 in above example)

$\underline{3}$

1, 2, 3

RWS, SWS, and Sequence Space

- RWS ≥ 1, SWS ≥ 1, RWS ≤ SWS
- If RWS = 1, "go back N" protocol, need SWS+1 sequence numbers
- If RWS = SWS, need 2SWS sequence numbers
- Generally need RWS+SWS sequence numbers
  - RWS packets in unknown state (ACK may/may not be lost)
  - SWS packets in flight must not overflow sequence number space

SWS = 3
RWS = 1
1, 2, 3 ... ~~7~~, 8, 9,
7, 8, 9

SWS = 3
RWS = 3
1, 2, 3 ... ~~7~~ 8, 9
7, 8 9

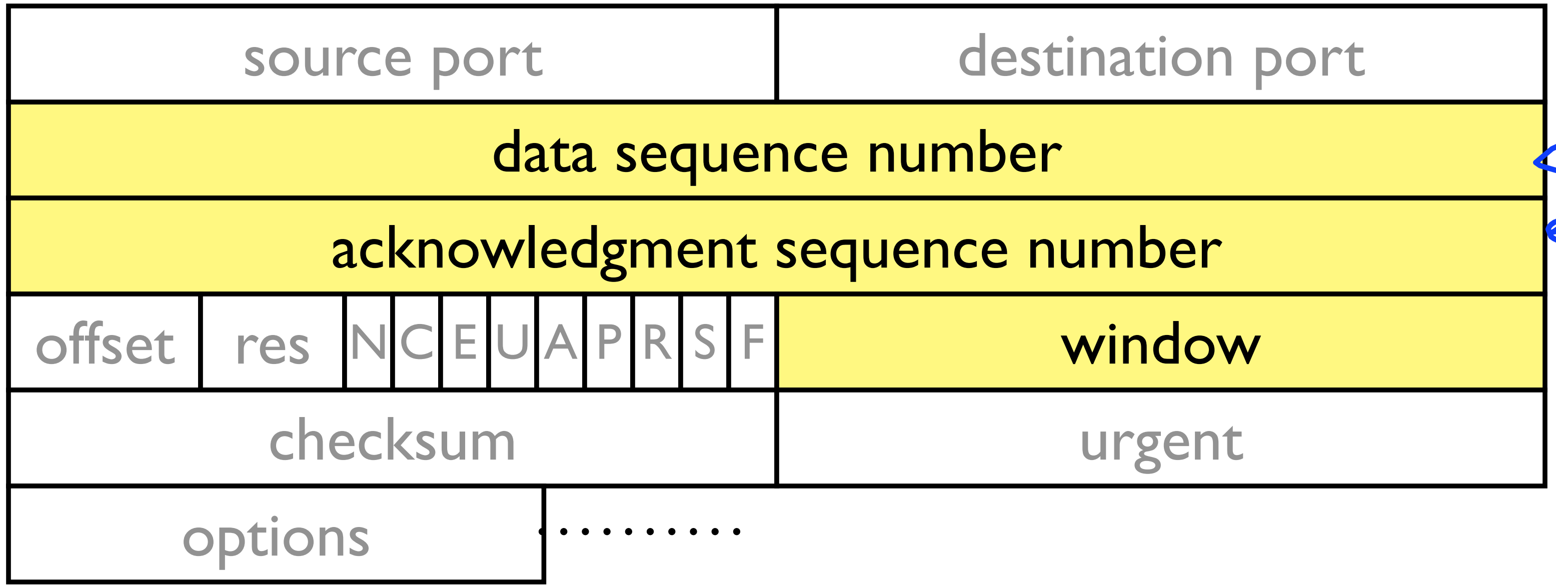
TCP Flow Control

- Receiver advertises RWS using window field
- Sender can only send data up to LAR + window

5,000

20,000

5,001 → 25,000



source port | destination port

data sequence number

acknowledgment sequence number

offset | res | N | C | E | U | A | P | R | S | F | window

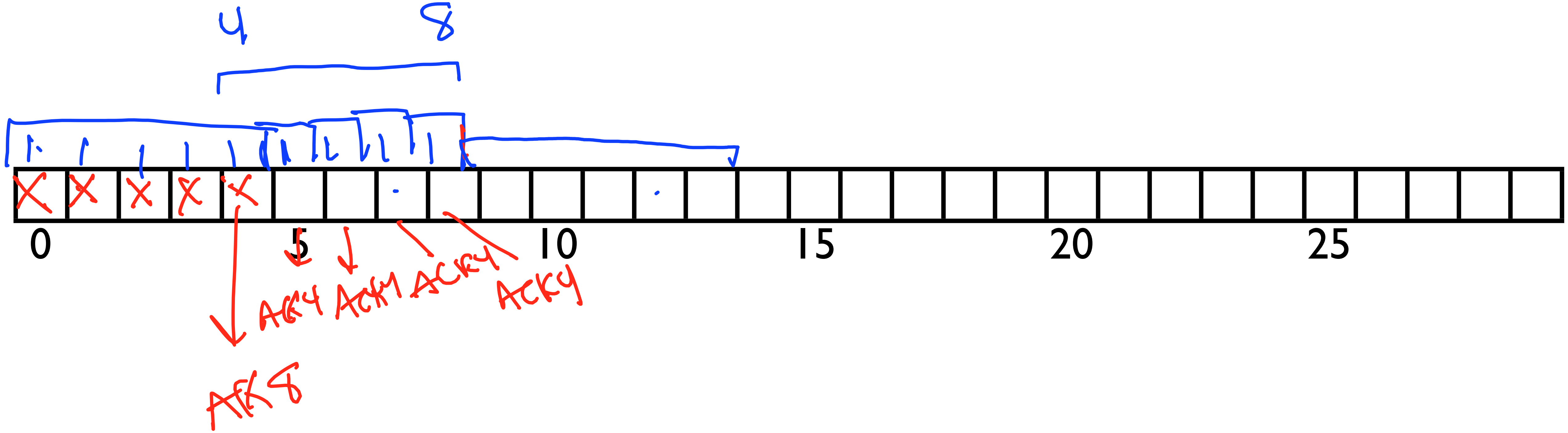
checksum | urgent

options ....

← bytes

Sliding Window Example

RWS = 5
SWS = 5



Sliding Window Flow Control

- Allow a "window" of unacknowledged packets in flight
- When acknowledgment arrives, advance window
- Required sequence number space size depends on window sizes