Computer Networks
COL 334/672

Reliability in TCP

Slides adapted from KR

Sem 1, 2025-26

Recap

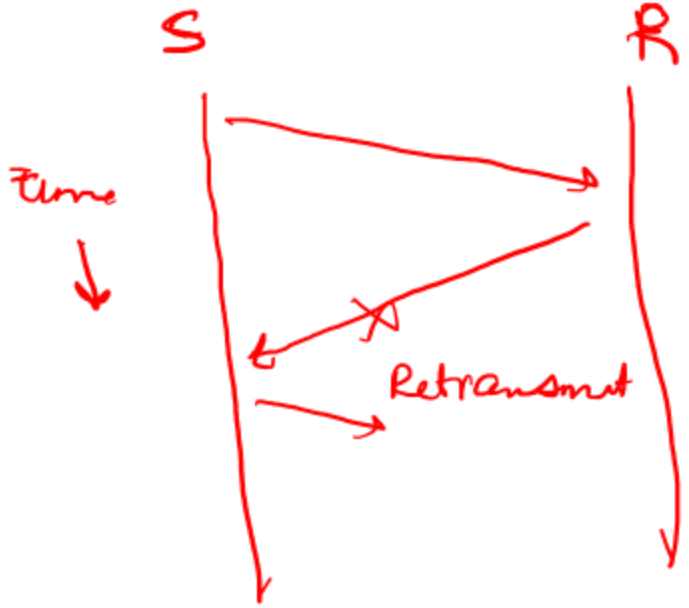
Reliability in Transport layer

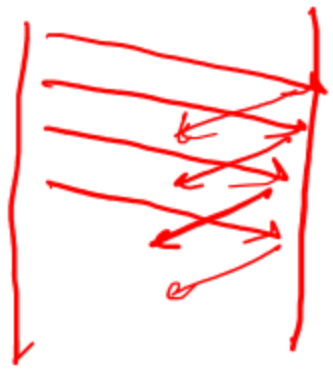
① Forward Error correction (Don't use it)

② Automatic Repeat Request (ARQ)

↳ Stop & wait

↳ Pipelining or Sliding window protocols



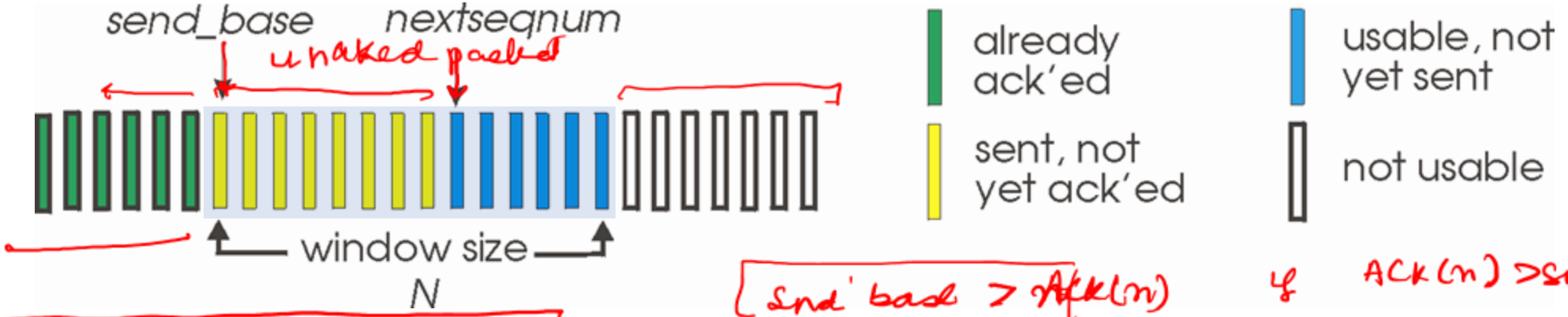


ACK every packet independentally
or ACK only in-order packet

→ ① Go BACK N
② Selective Repeat

Go-Back-N: sender

- SWS of N packets
- k-bit seq # in pkt header

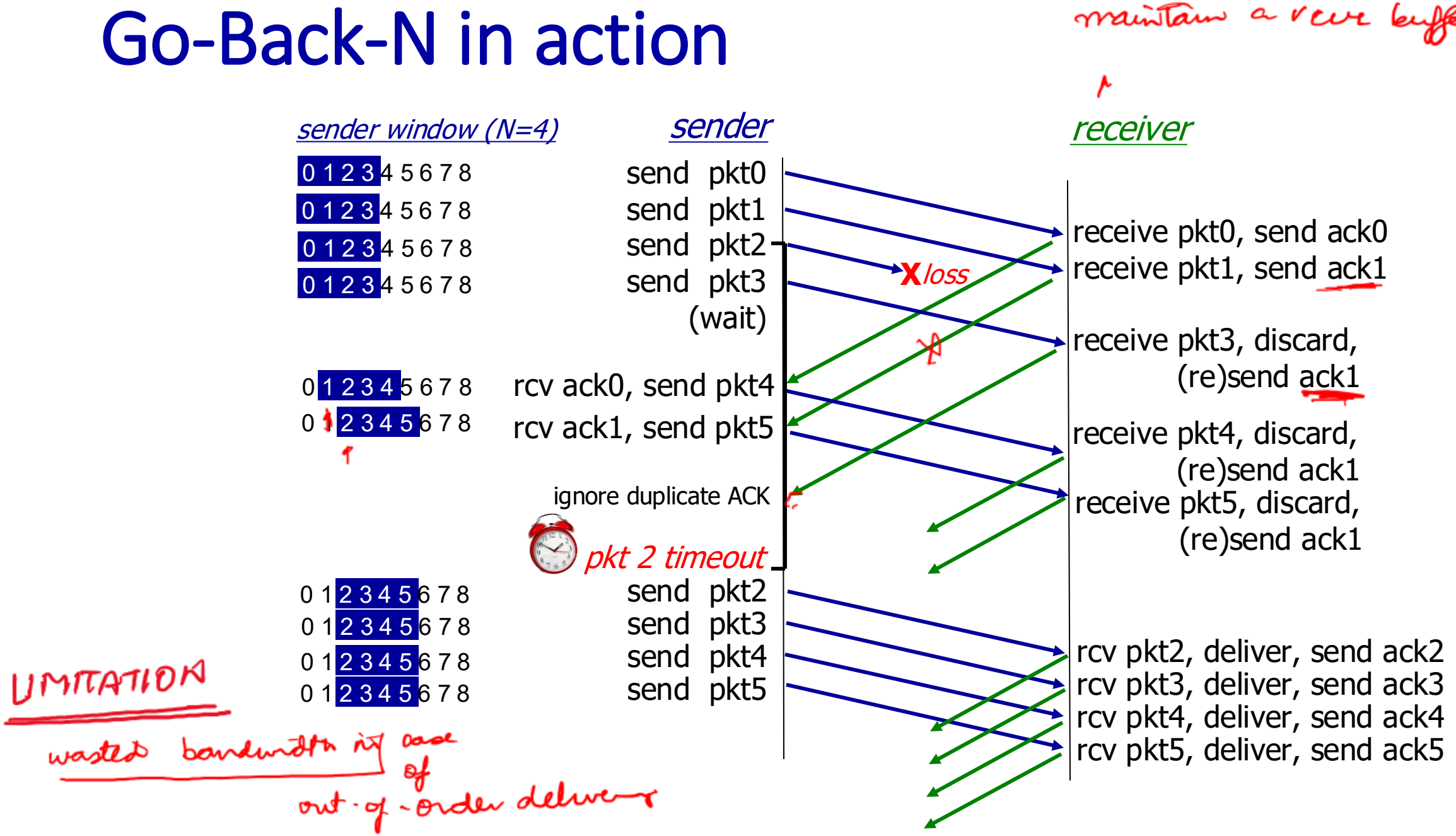


- *cumulative ACK:* ACK(*n*): ACKs all packets up to, including seq # *n*
  - on receiving ACK(*n*): move window forward to begin at *n+1*
- timer for oldest in-flight packet
- *timeout(n):* retransmit packet n and all higher seq # packets in window

Go-Back-N: receiver

- ACK-only: always send ACK for correctly-received packet so far, with highest *in-order* seq #
  - may generate duplicate ACKs
  - need only remember `rcv_base`
- on receipt of out-of-order packet:
  - discard (don't buffer) the packet
  - re-ACK pkt with highest in-order seq #

Go-Back-N in action



Selective repeat: the approach

- *pipelining*: *multiple* packets in flight
- *receiver individually ACKs* all correctly received packets
  - buffers packets, as needed, for in-order delivery to upper layer
- sender:
  - maintains (conceptually) a timer for each unACKed pkt
    - timeout: retransmits single unACKed packet associated with timeout
  - maintains (conceptually) "window" over *N* consecutive seq #s
    - limits pipelined, "in flight" packets to be within this window

Selective repeat: sender and receiver

sender

data from above:

- if next available seq # in window, send packet

timeout(*n*):

- resend packet *n*, restart timer

ACK(*n*) in [sendbase,sendbase+N-1]:

- mark packet *n* as received
- if n smallest unACKed packet, advance window base to next unACKed seq #

receiver

packet *n* in [rcvbase, rcvbase+N-1]

- send ACK(*n*)
- out-of-order: buffer
- in-order: deliver (also deliver buffered, in-order packets), advance window to next not-yet-received packet

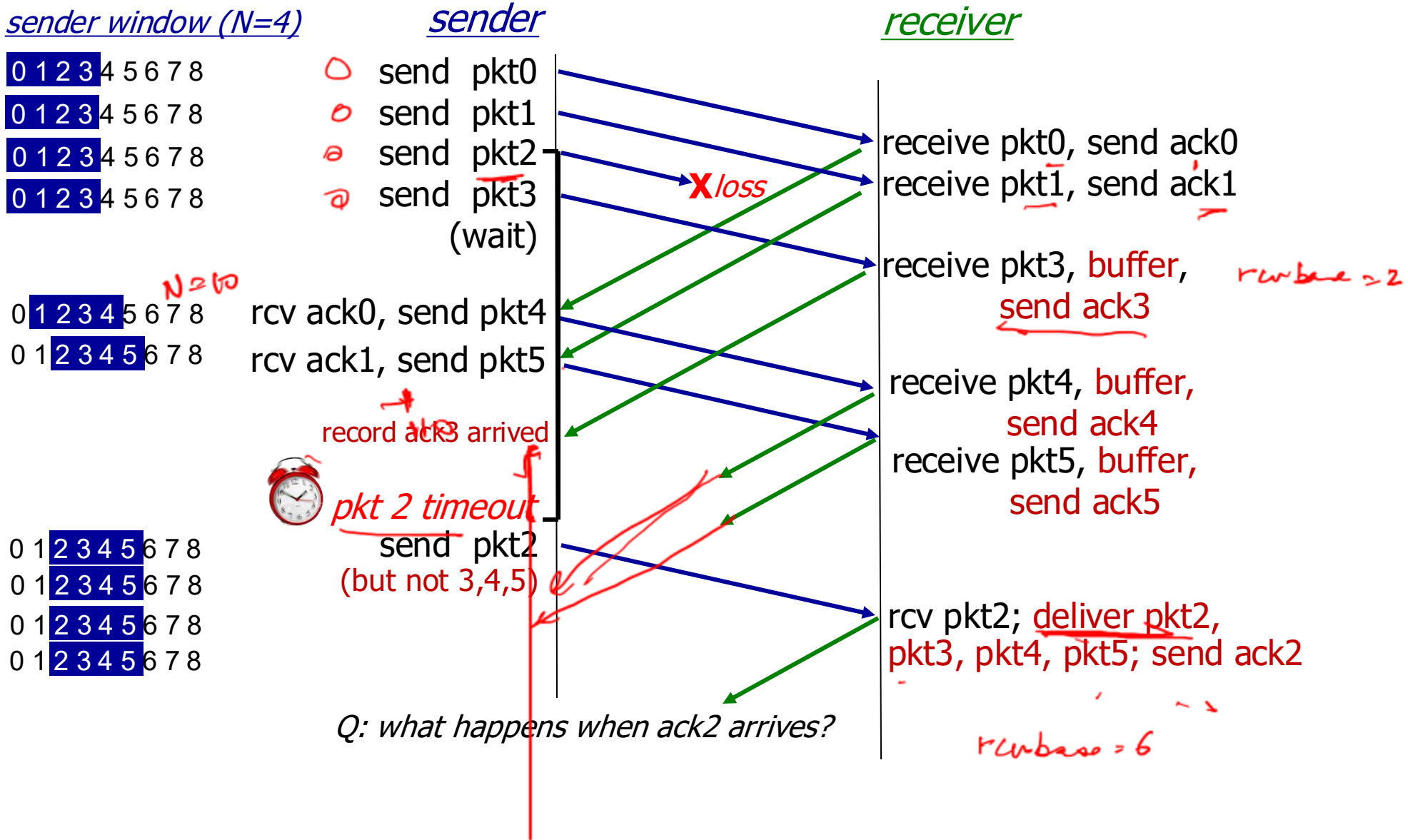
packet *n* in [rcvbase-N,rcvbase-1]

- ACK(*n*)

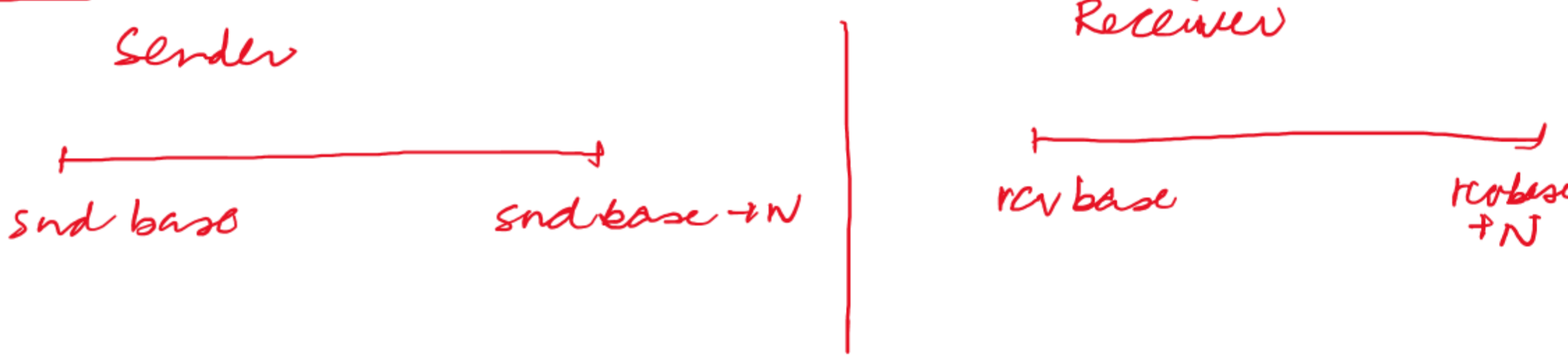
otherwise:

- ignore

Selective Repeat in action



Claim :



CLAIM

snd base $\geq$ rcv base - N

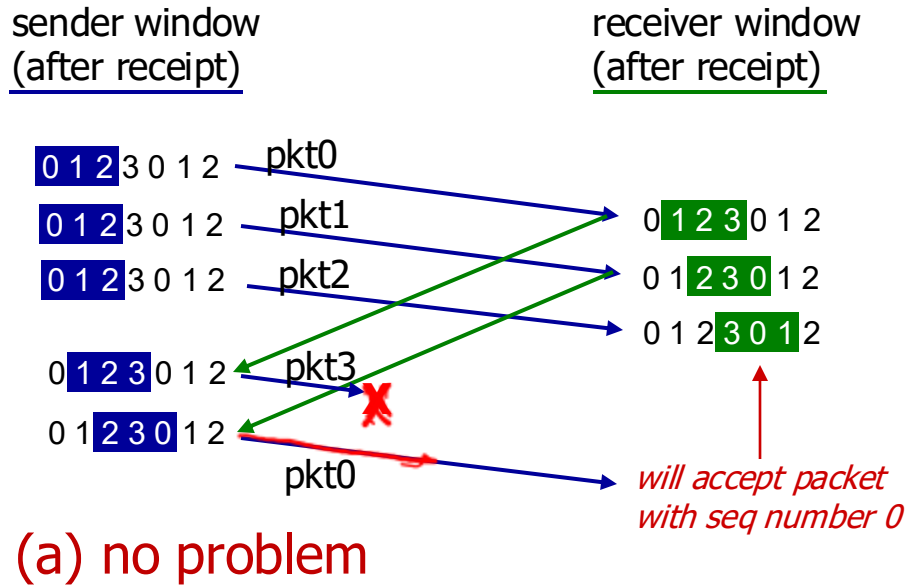
E.g. if N = 10, rcv base = 100

there is no way possible that send base < 90, else sender would not have sent the packet with seq # 100

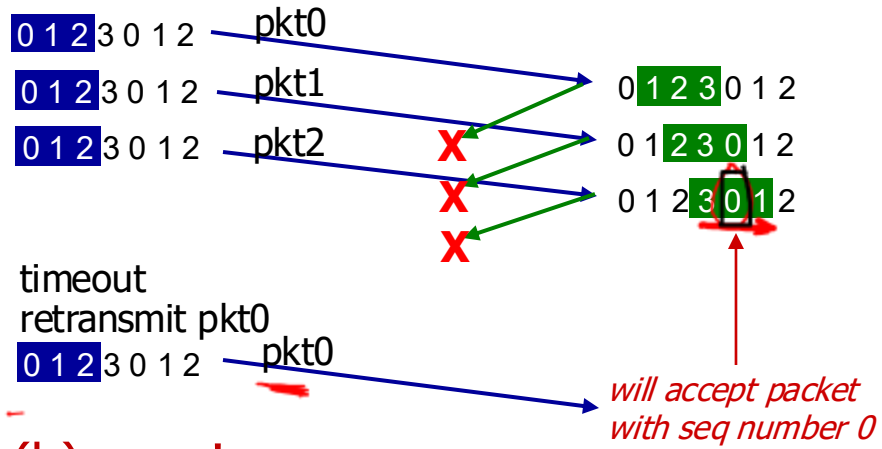
Selective repeat: a dilemma!

example:

- seq #s: 0, 1, 2, 3 (base 4 counting)
- window size=3



(a) no problem



(b) oops!

Selective repeat: a dilemma!

example:

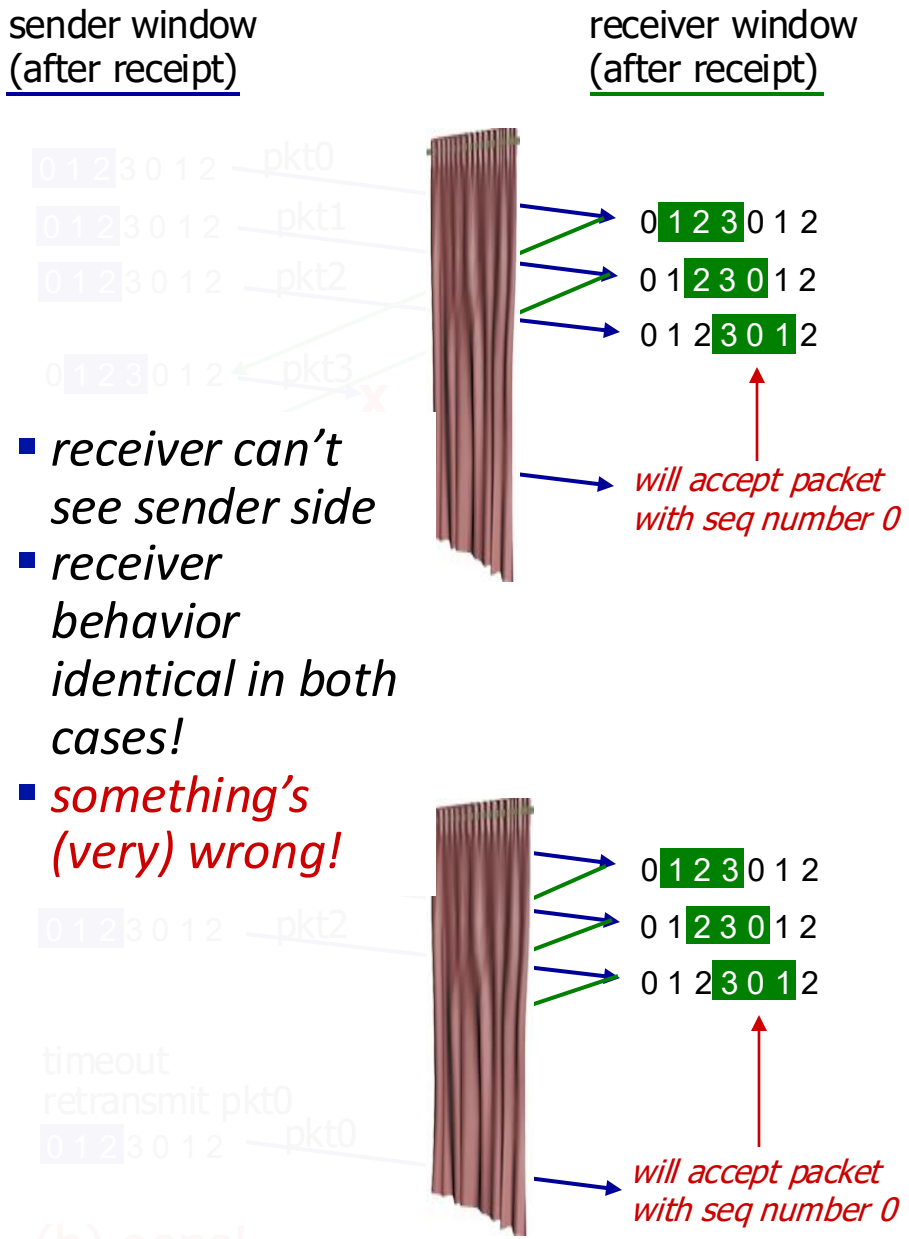
- seq #s: 0, 1, 2, 3 (base 4 counting)
- window size=3

Q: what relationship is needed between sequence # size and window size to avoid problem in scenario (b)?

Maximum discrepancy = $2N$

$2N \leq 2^k$

$N \leq 2^{k-1}$



How does TCP implement reliability?

FACTS ABOUT TCP

1. TCP is a connection-oriented protocol
   - ↳ Establishment & Termination protocols
2. Byte stream protocol
   - ↳ seq #s are bytes : not packets
3. Bidirectional data transfer
   - ↳ seq #s needed for both sender & Receiver
   - ↳ ACK #s needed for both sender & Receiver