Computer Networks (CS F303)

BITS Pilani
Pilani Campus

Virendra Singh Shekhawat
Department of Computer Science and Information Systems

innovate achieve lead

BITS Pilani

Pilani Campus

Second Semester 2020-2021

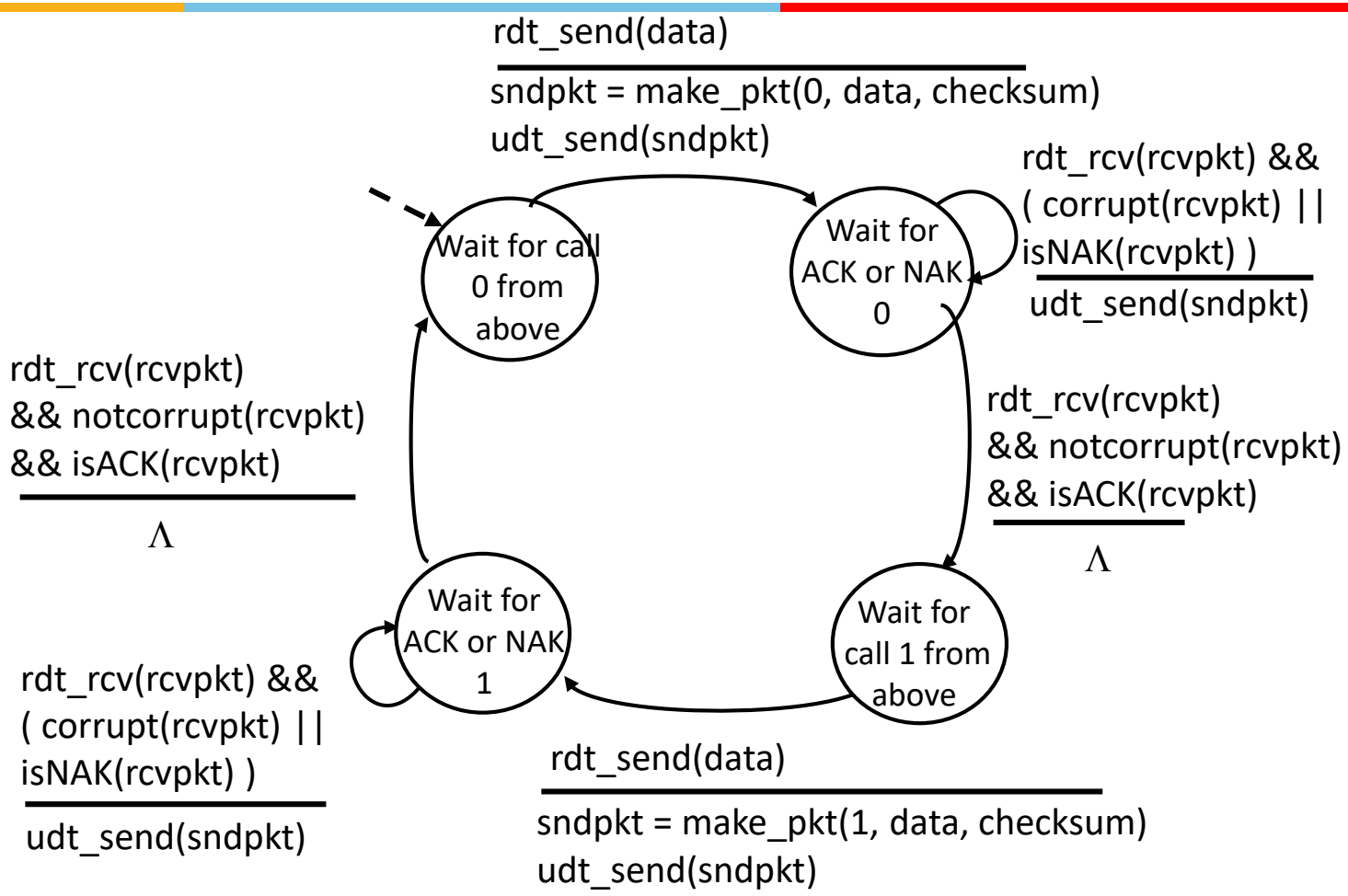
Module-3 <Transport layer>

Lecture: 12

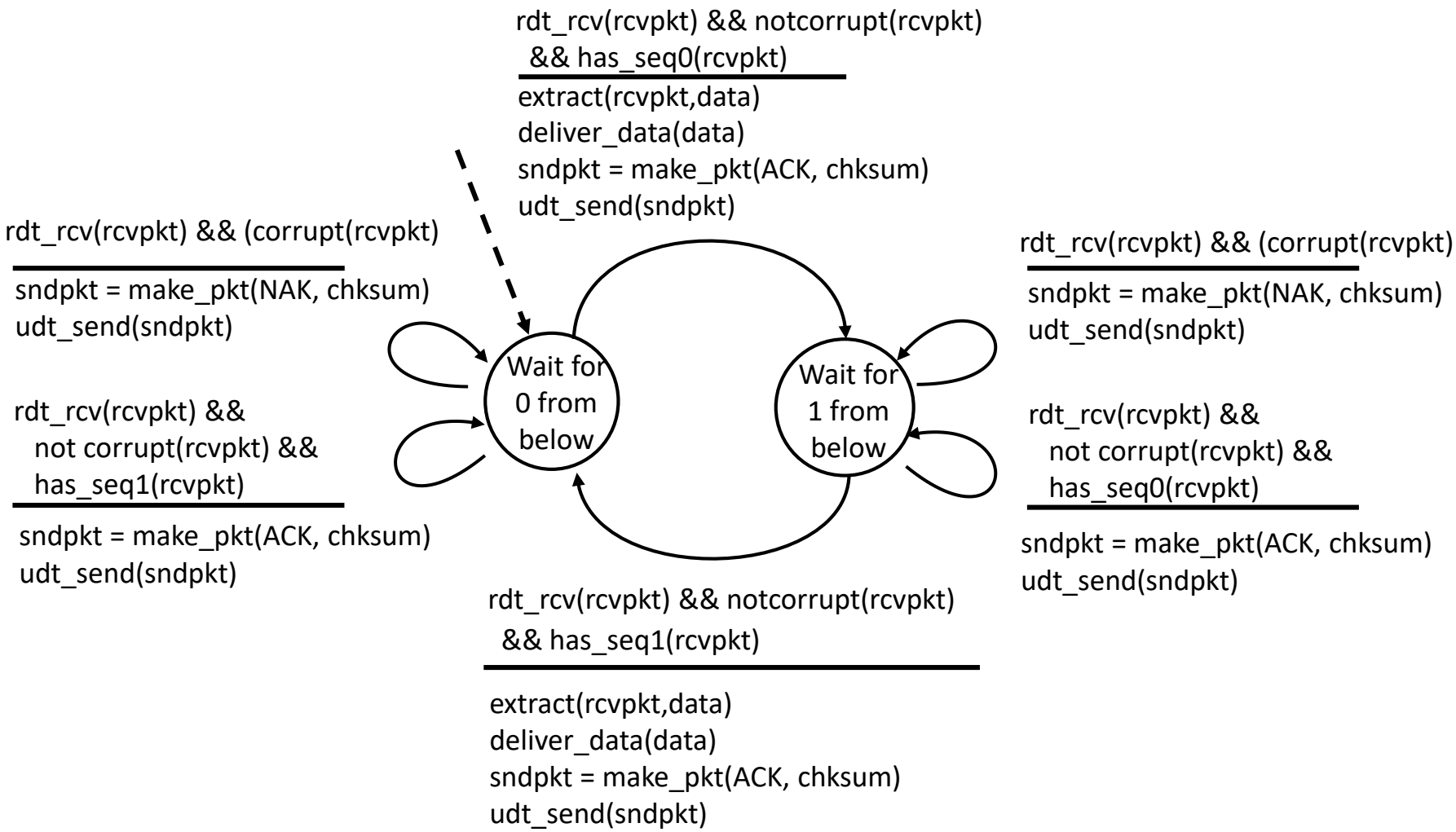
Topics

- Transport Layer
  - Reliable data transfer (Protocol design)
    - Stop and Wait vs. Pipelining (Sliding Window)
    - Go Back N and Selective Repeat Protocols
  - Flow control
  - Congestion control

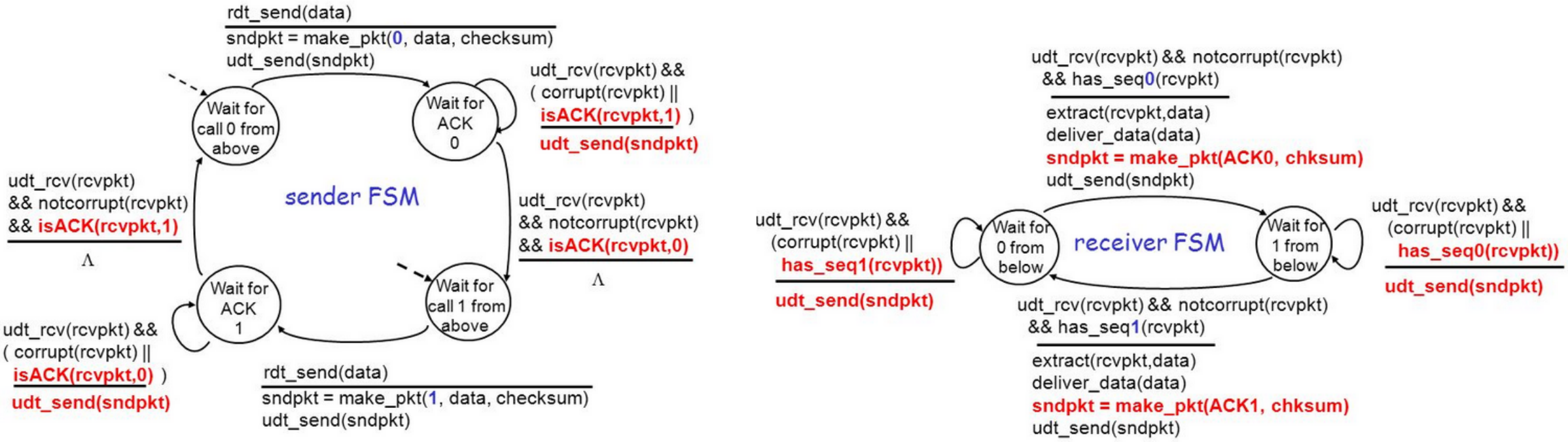
rdt2.1: Sender, handles garbled ACK/NAKs



rdt2.1: Receiver, handles garbled ACK/NAKs



rdt2.2: NAK Free Protocol



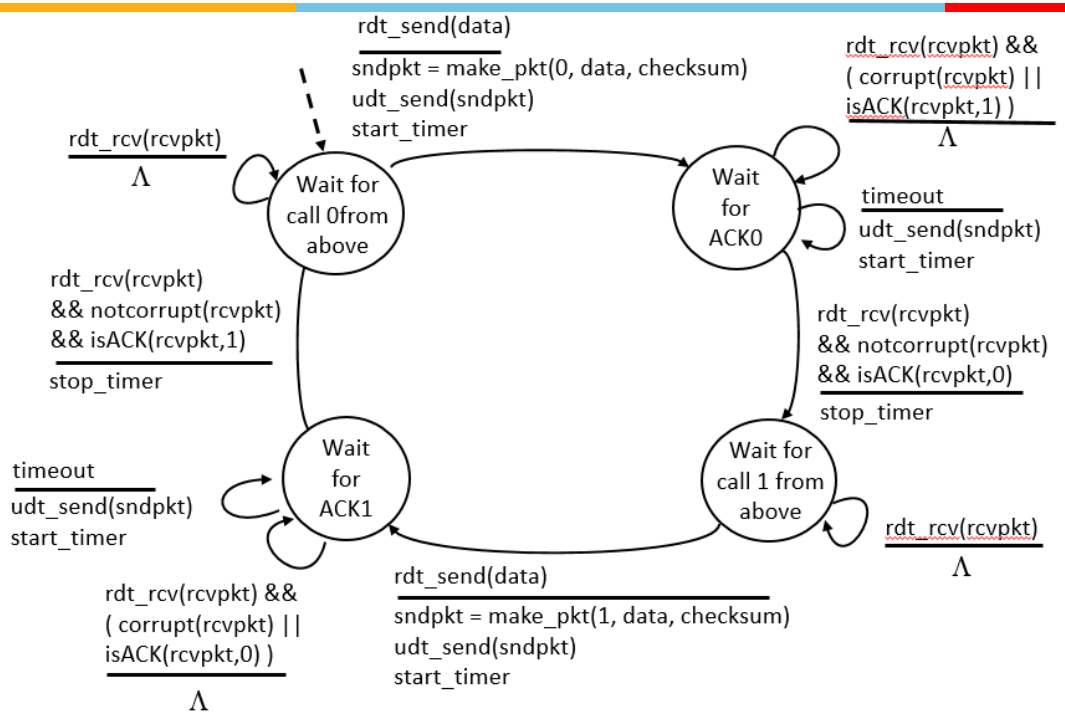
- Same functionality as rdt2.1, using ACKs only
- Instead of NAK, receiver sends ACK for last pkt received OK
  - Receiver must *explicitly* include **seq #** of pkt being **ACKed**

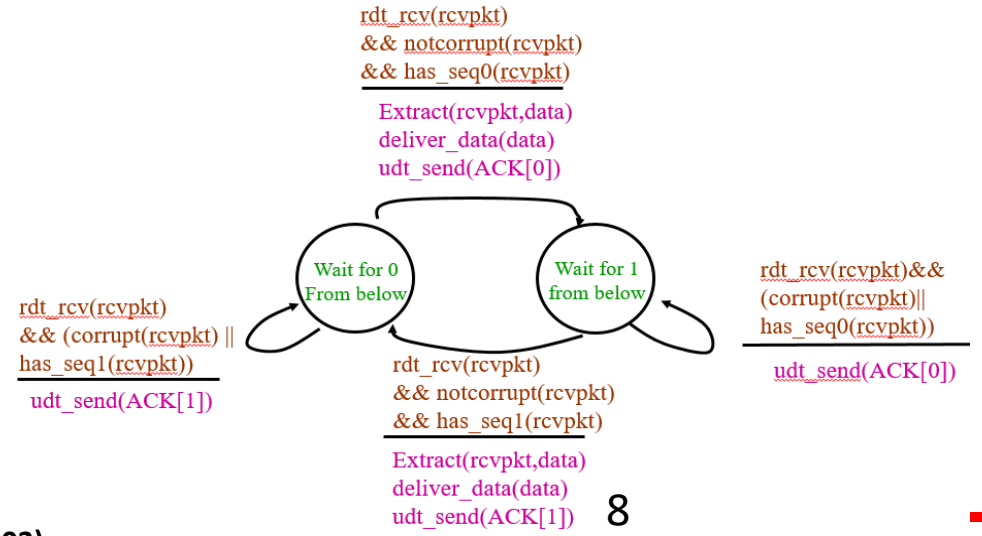
rdt3.0: Channels with **errors** *and* **loss**

- New assumption: Underlying channel can also lose packets (data or ACKs)
  - Checksum, seq. #, ACKs, retransmissions will be of help, but not enough

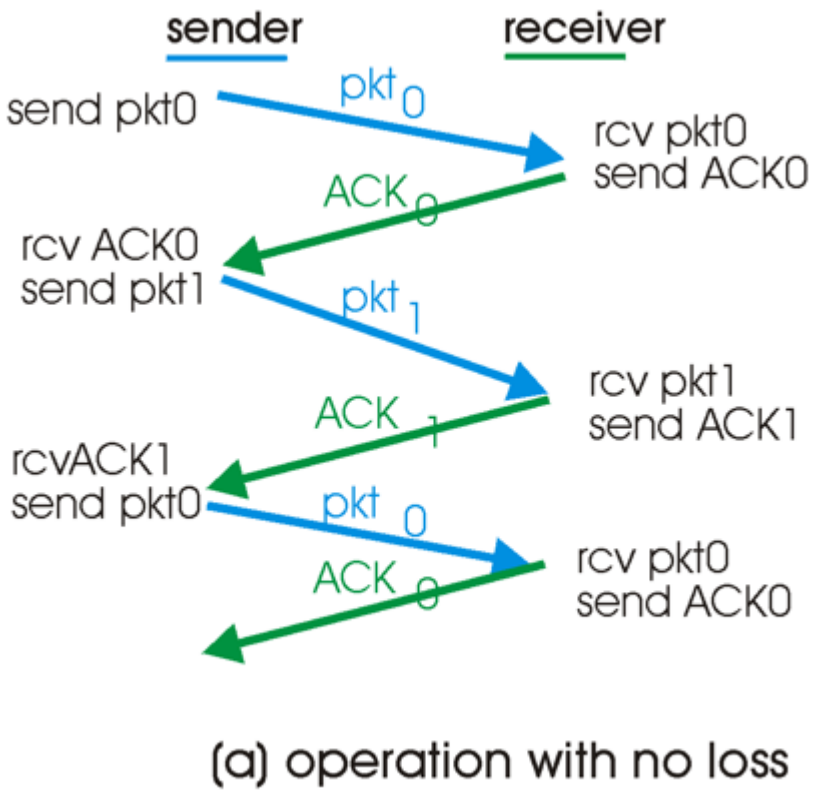
- Approach: Sender waits "reasonable" amount of time for ACK
  - Retransmits if no ACK received in this time
  - If pkt (or ACK) just delayed (not lost):
    - Retransmission will be duplicate, but use of seq. #'s already handles this
    - Receiver must specify seq # of pkt being ACKed
  - Requires countdown timer

rdt 3.0 Sender and Receiver FSM

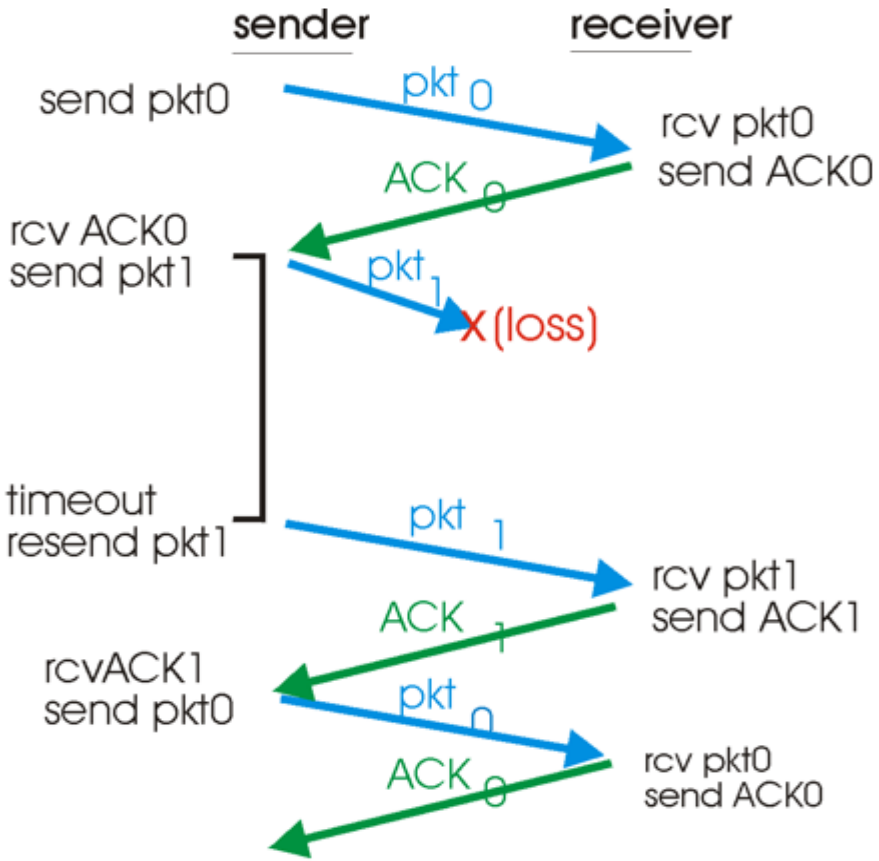




rdt3.0 in action

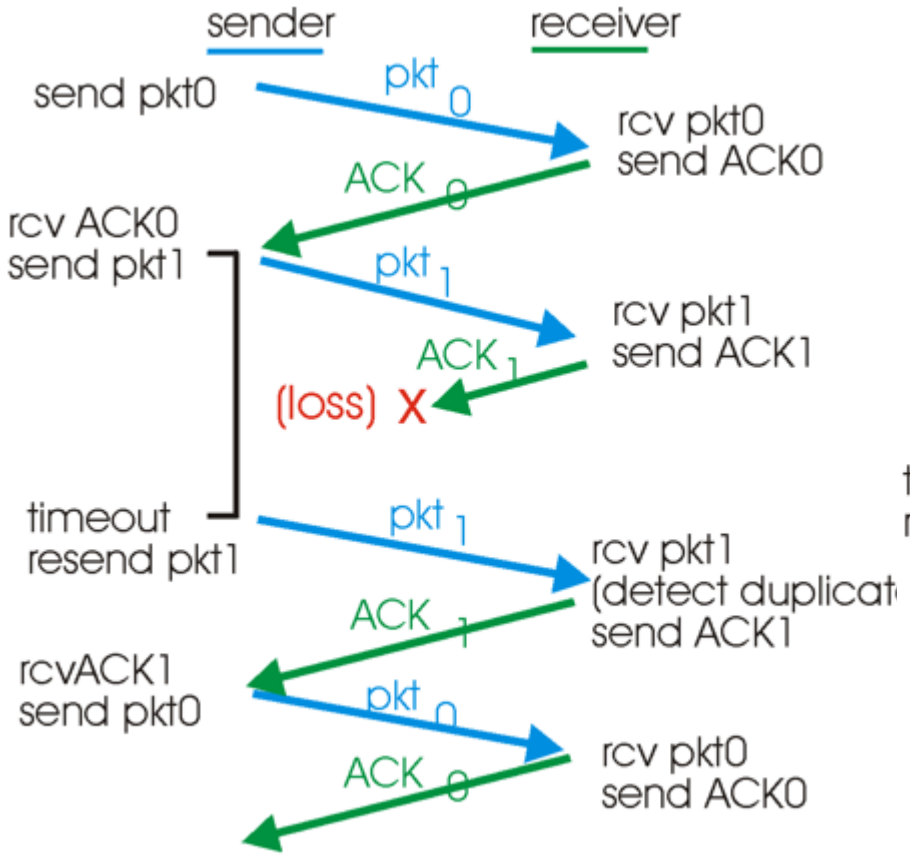


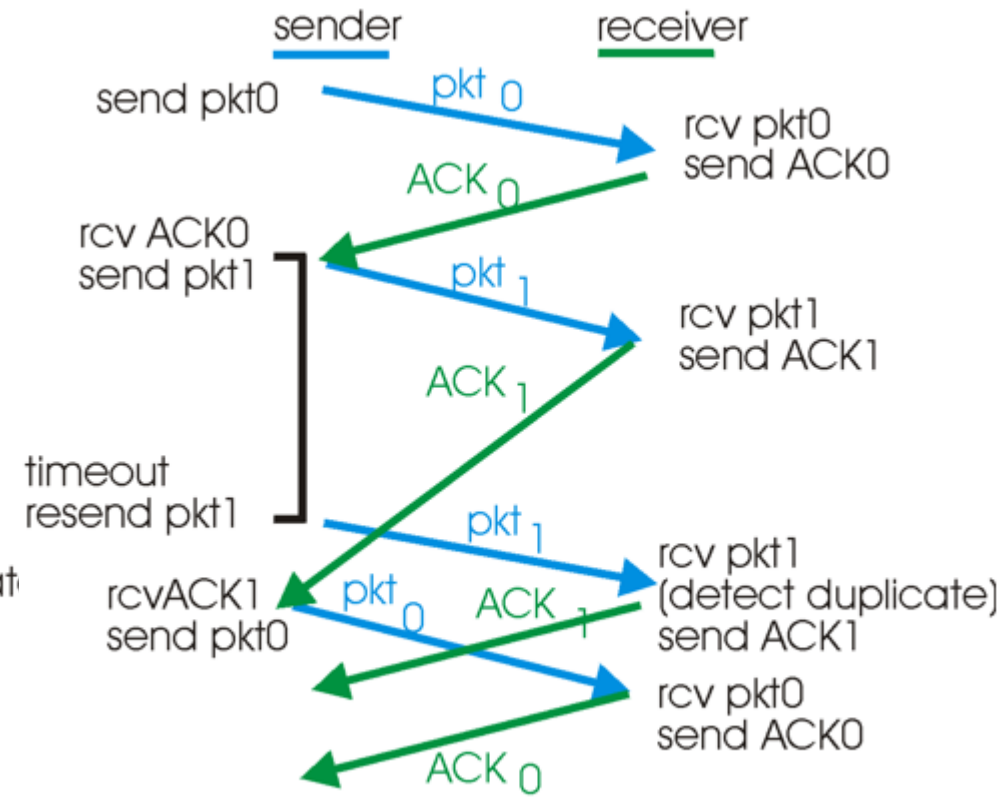
(a) operation with no loss



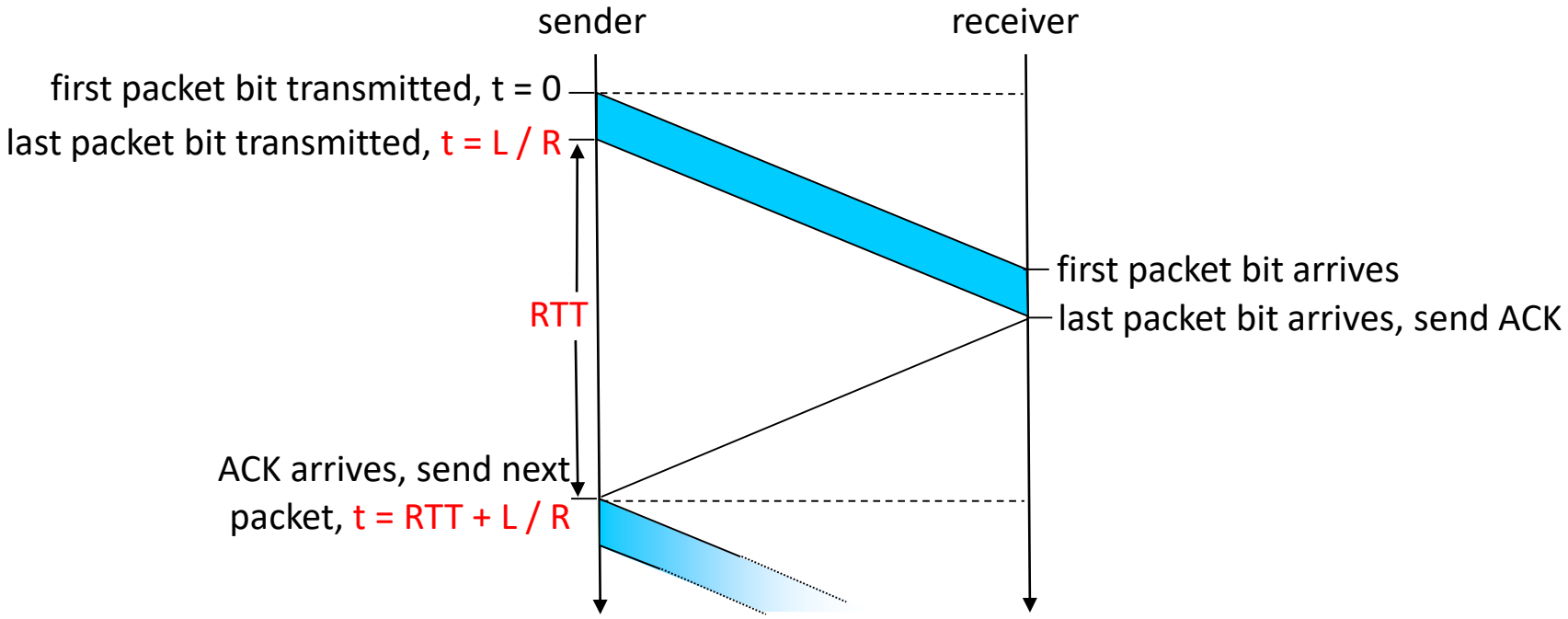
(b) lost packet

rdt3.0 (Lost ACK and Premature Timeout)





rdt3.0: Performance

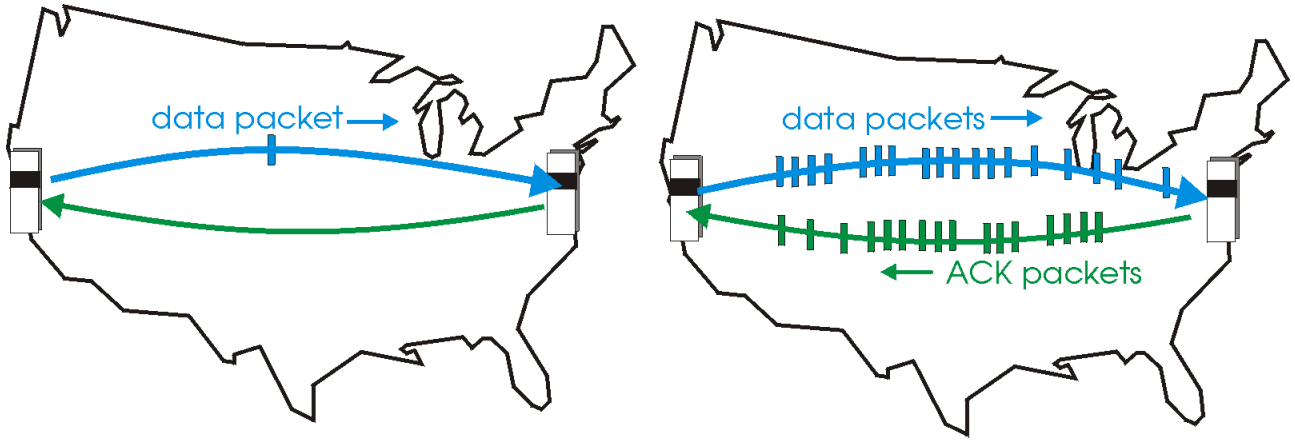


Example: 1 Gbps link, 15 ms end to end prop. delay, 1KB packet:

$$U_{sender} = \frac{L/R}{RTT + L/R} = \frac{.008}{30.008} = 0.00027$$

Pipelined Protocols

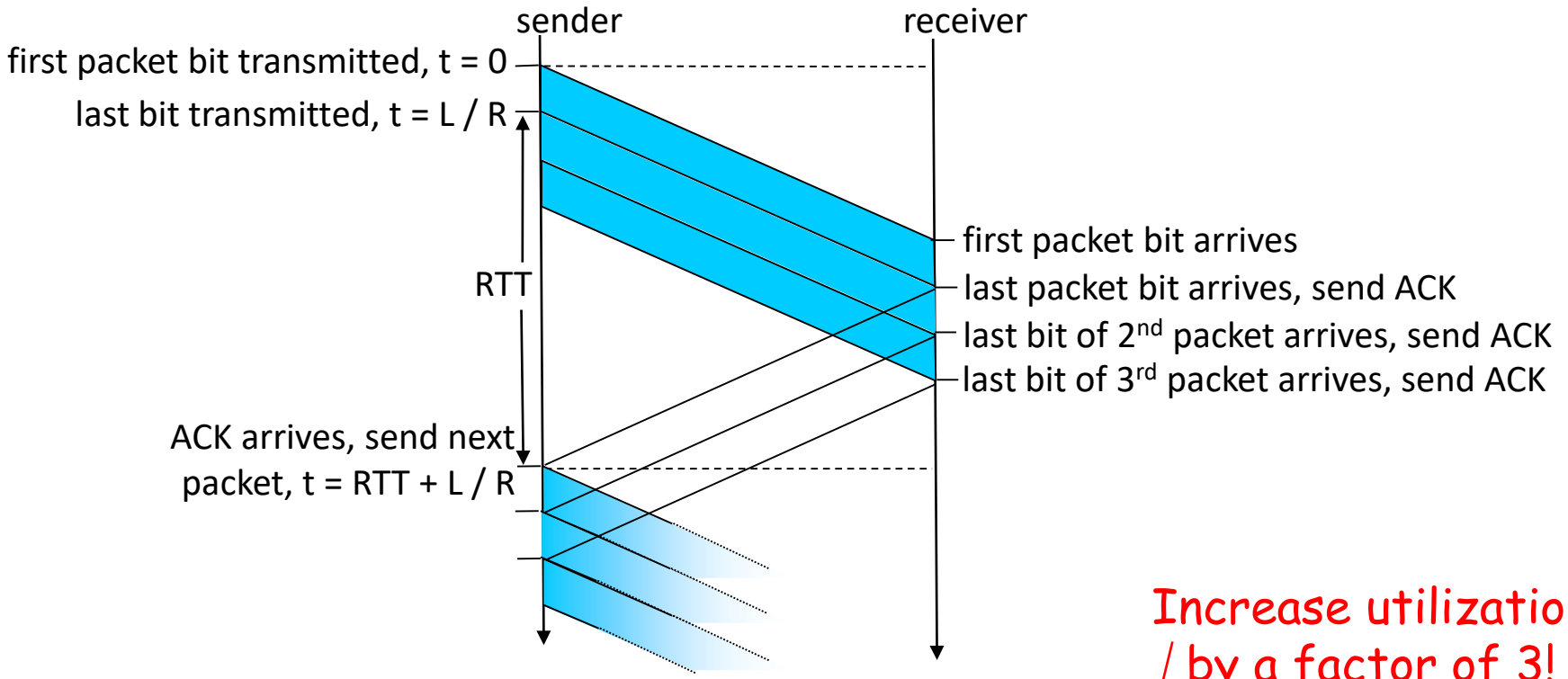
- Pipelining: Sender allows multiple, “in-flight”, yet-to-be-acked pkts
  - Range of sequence numbers must be increased
  - Buffering at sender and/or receiver



(a) a stop-and-wait protocol in operation

(b) a pipelined protocol in operation

Pipelining: Increased Utilization



$$U_{sender} = \frac{3 * L / R}{RTT + L / R} = \frac{.024}{30.008} = 0.0008$$

Increase utilization by a factor of 3!

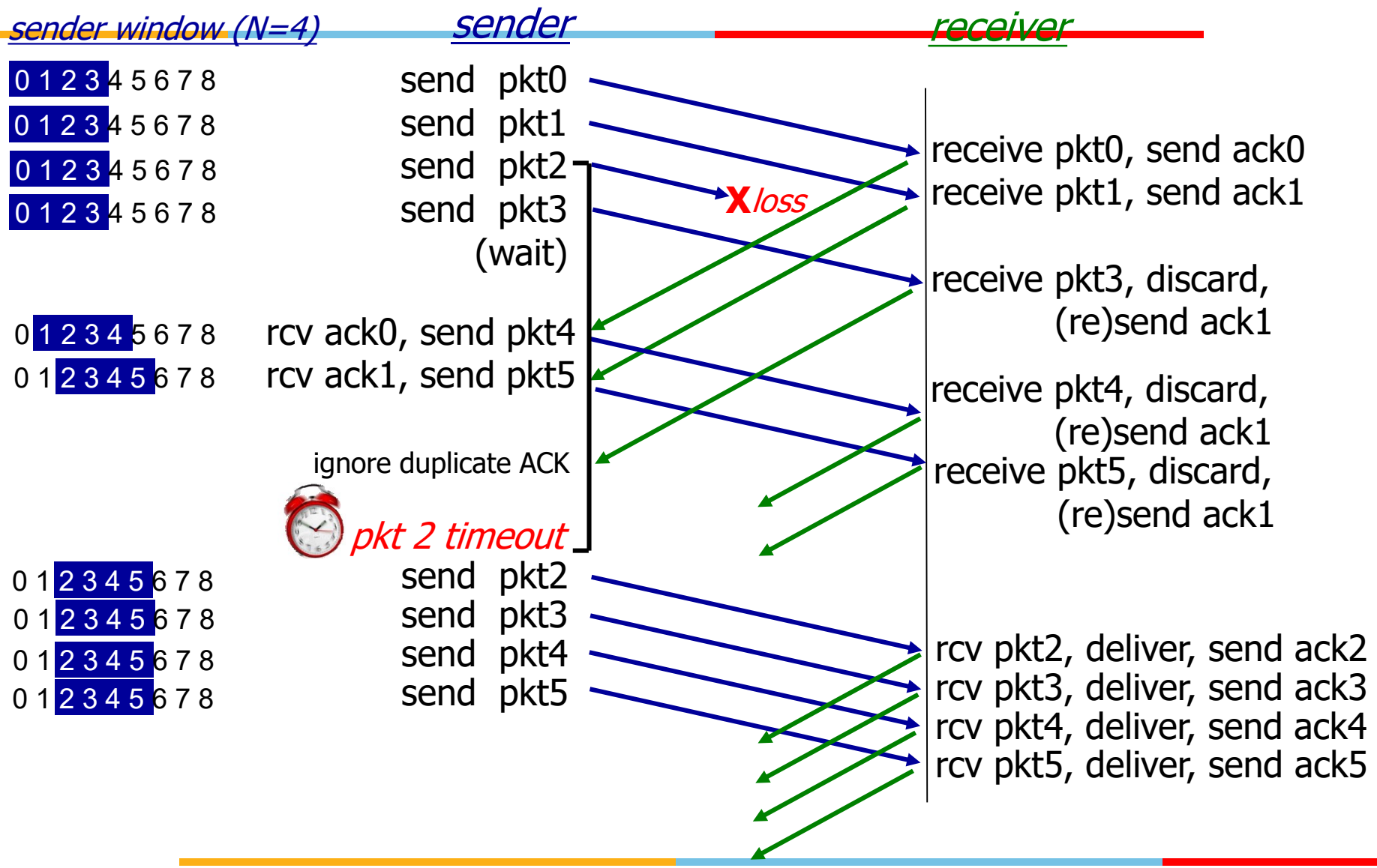
Pipelining Protocols Requirements

- The range of sequence numbers must be increased
  - Multiple in-transit packets
- Packet Buffering is required at both sides. Why?
- Range of sequence numbers needed?
- Two basic approaches
  - Go-Back-N (GBN)
  - Selective Repeat (SR)

GBN Protocol

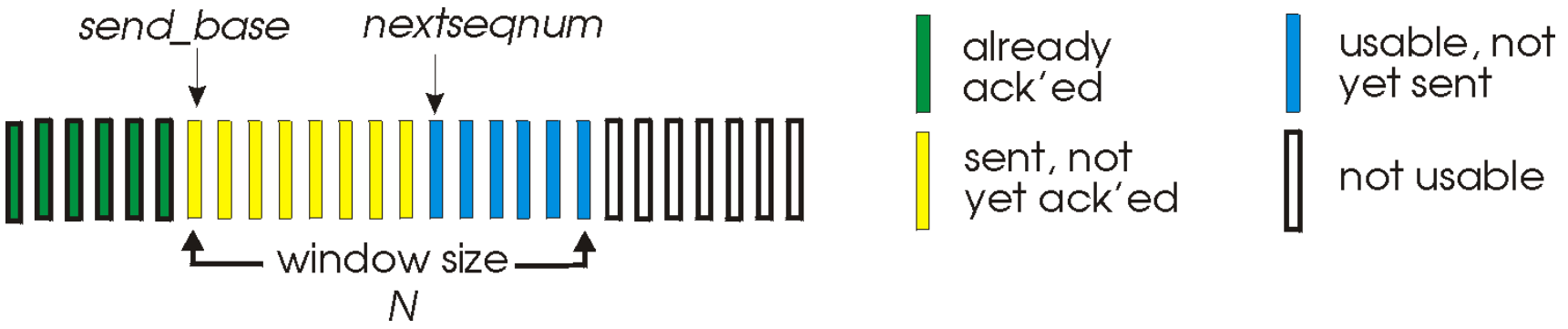
- Sender is allowed to transmit maximum **N** pkts without waiting for ACK
- Receiver only sends cumulative ACK
  - Doesn't receive out of order packet!
- Sender has timer for oldest unacked packet
  - When timer expires, retransmit *all* unacked packets

GBN in action

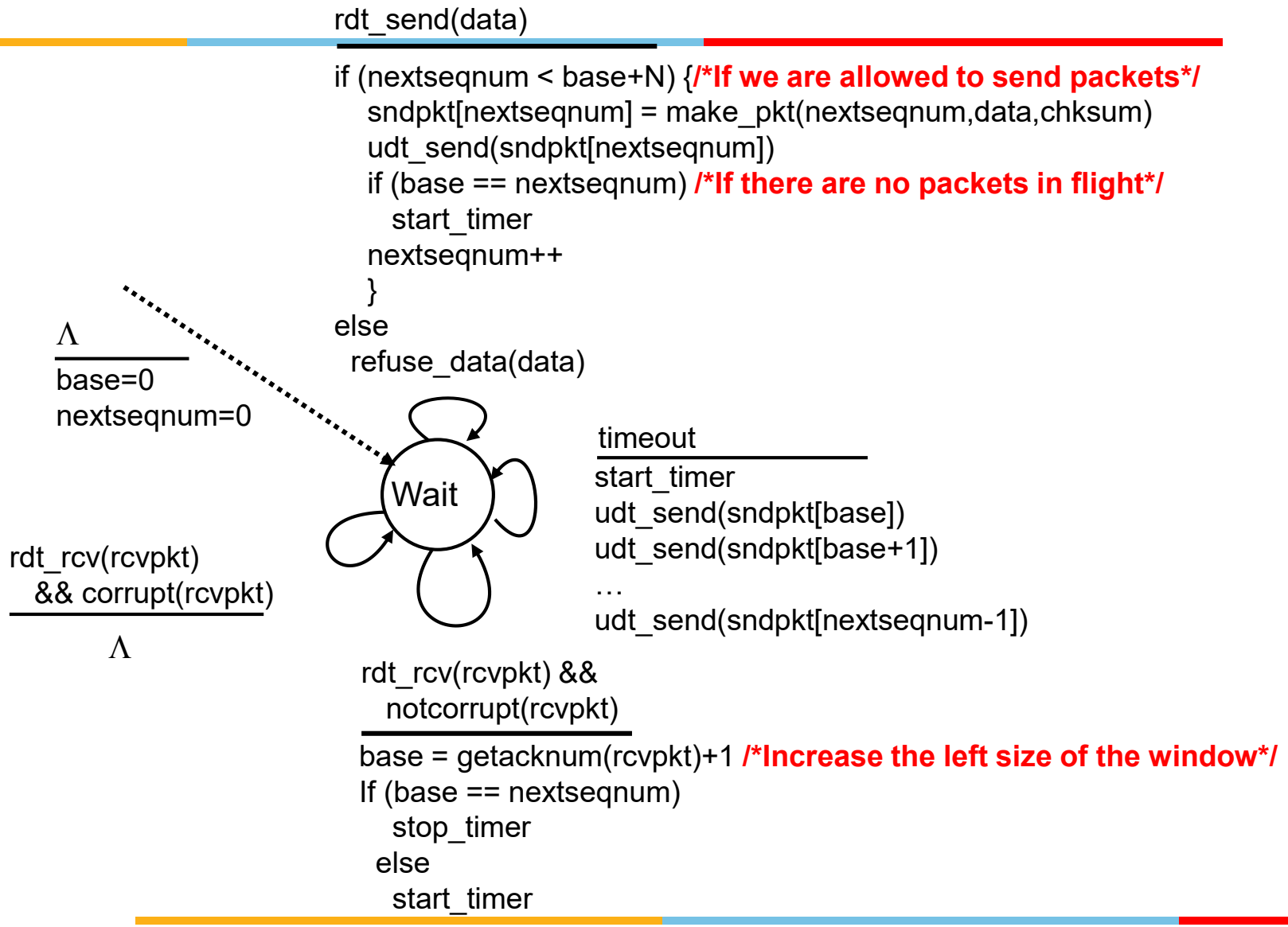


GBN Sender

- K-bit seq # in pkt header (modulo 2^K arithmetic)
- A "window" of upto N, consecutive unack'ed pkts allowed

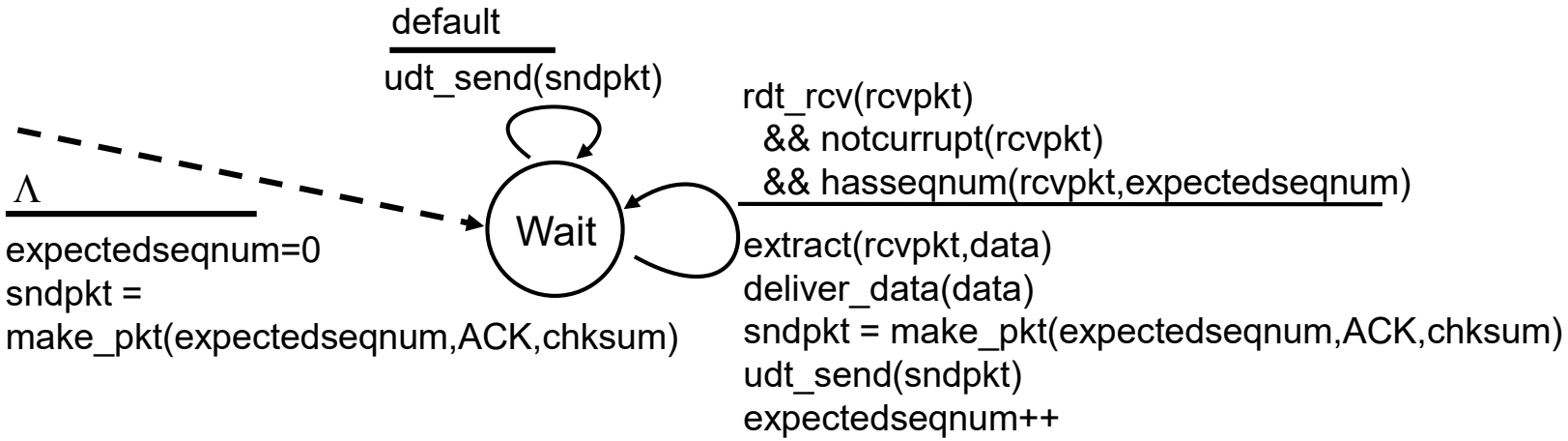


GBN Sender FSM



GBN Receiver FSM

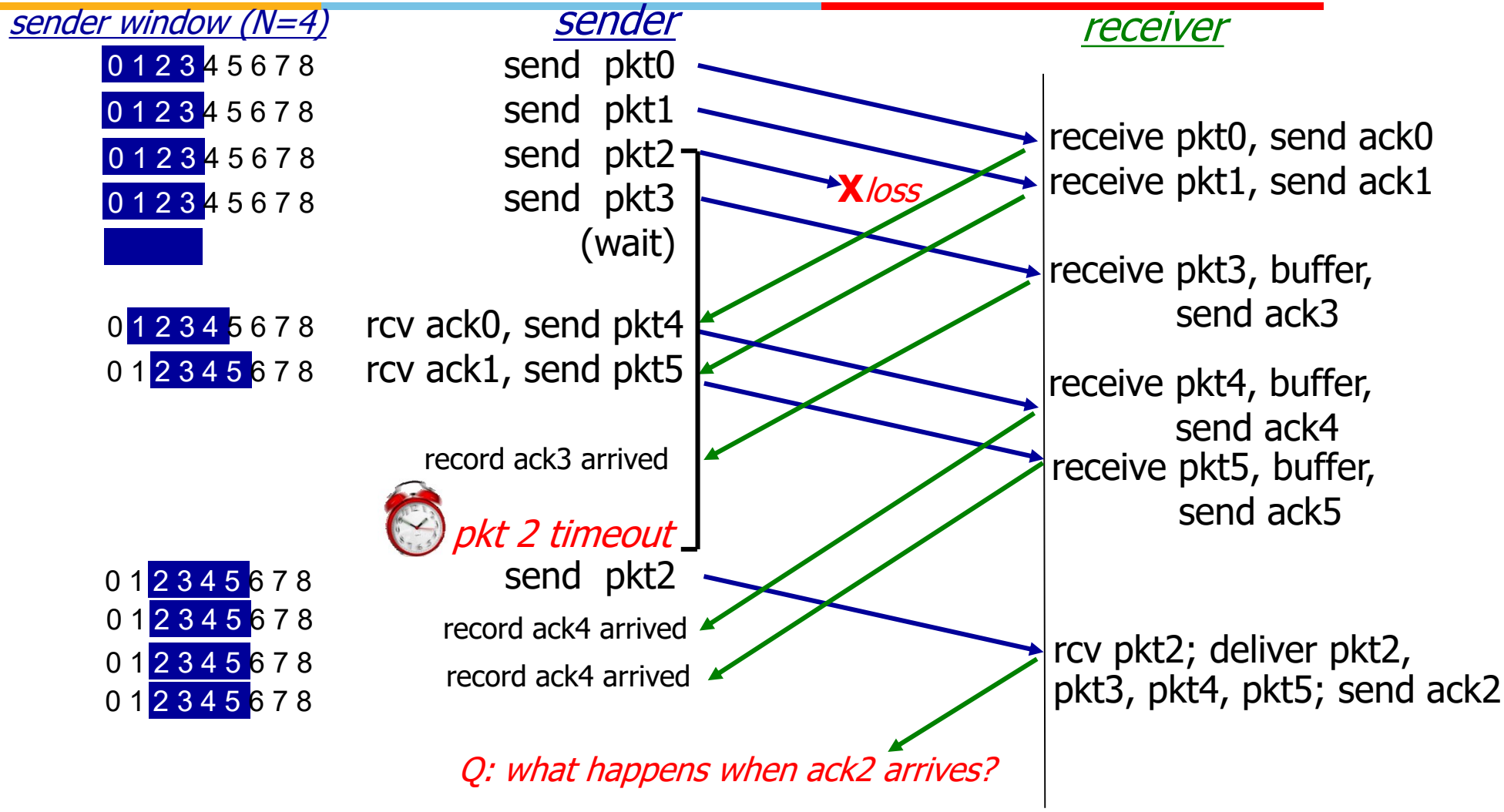
- Always send ACK for correctly-received pkt with highest ***in-order*** seq #
  - Need only to remember "**expectedseqnum**"
- If out-of-order pkt arrived
  - Discard it
  - Re-ACK pkt with the highest ***in-order*** seq #



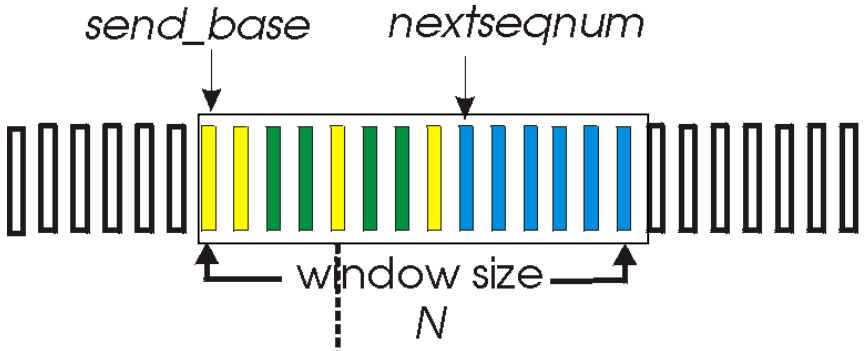
Selective Repeat (SR) Protocol

- Receiver *individually* acknowledges all correctly received pkts
  - Buffers pkts, as needed, for eventual in-order delivery to upper layer
- Sender only resends pkts for which ACK not received
  - Sender keeps timer for each unACKed pkt
- Retransmit only that unacked packet for which ***timer expires***

Selective Repeat in Action



SR Protocol: Windows



(a) sender view of sequence numbers

already ack'ed

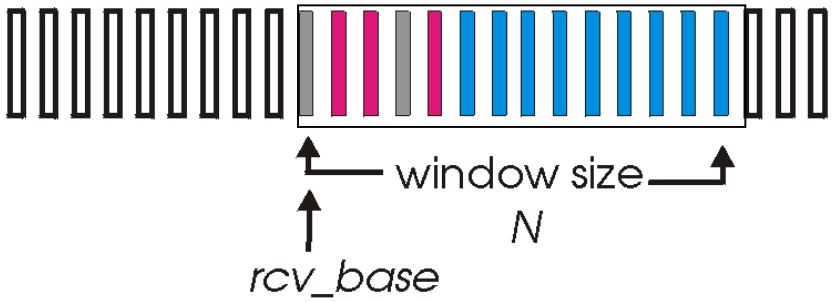
sent, not yet ack'ed

usable, not yet sent

not usable

Events at Sender

- Data from above
- Timeout
- ACK(n) in [sendbase,sendbase+N-1]



(b) receiver view of sequence numbers

out of order (buffered) but already ack'ed

Expected, not yet received

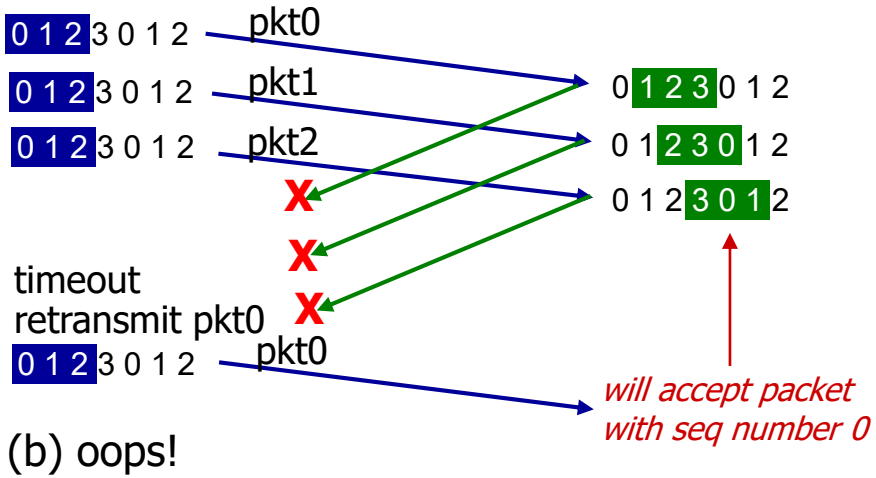
acceptable (within window)

not usable

Events at Receiver

- ✓ Pkt n in [rcvbase, rcvbase+N-1]
- ✓ Pkt n in [rcvbase-N,rcvbase-1]
- ✓ Otherwise

Selective Repeat Dilemma



(b) oops!

Relation between Window Size and Sequence Number

- Sequence numbers range for K bits
  - 0 to 2^K-1
- What should be the window size N for
  - Selective Repeat
  - Go Back N

Thank You!