COMPUTER NETWORKS

Transport Layer

Animesh Giri
Department of Computer Science & Engineering

COMPUTER NETWORKS

Principles of reliable date transfer

Animesh Giri
Department of Computer Science & Engineering

In this segment

- rdt3.0: channels with errors and loss
- rdt3.0 sender
- rdt3.0 in action
- Performance of rdt3.0
- rdt3.0: stop-and-wait operation

rdt3.0: channels with errors and loss

new assumption: underlying channel can also lose packets (data, ACKs)

- checksum, seq. #, ACKs, retransmissions will be of help … but not enough

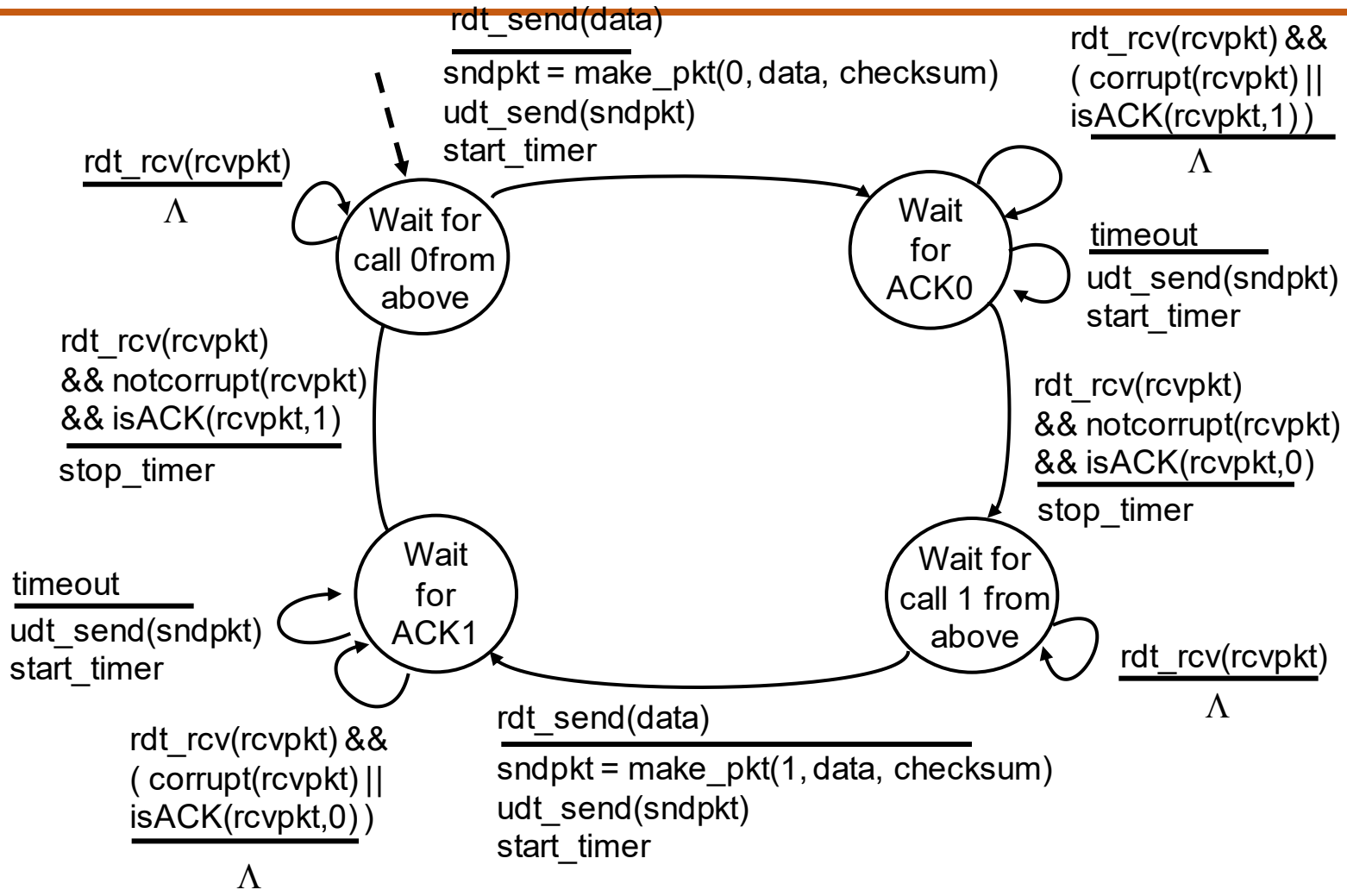
timeout

approach: sender waits "reasonable" amount of time for ACK

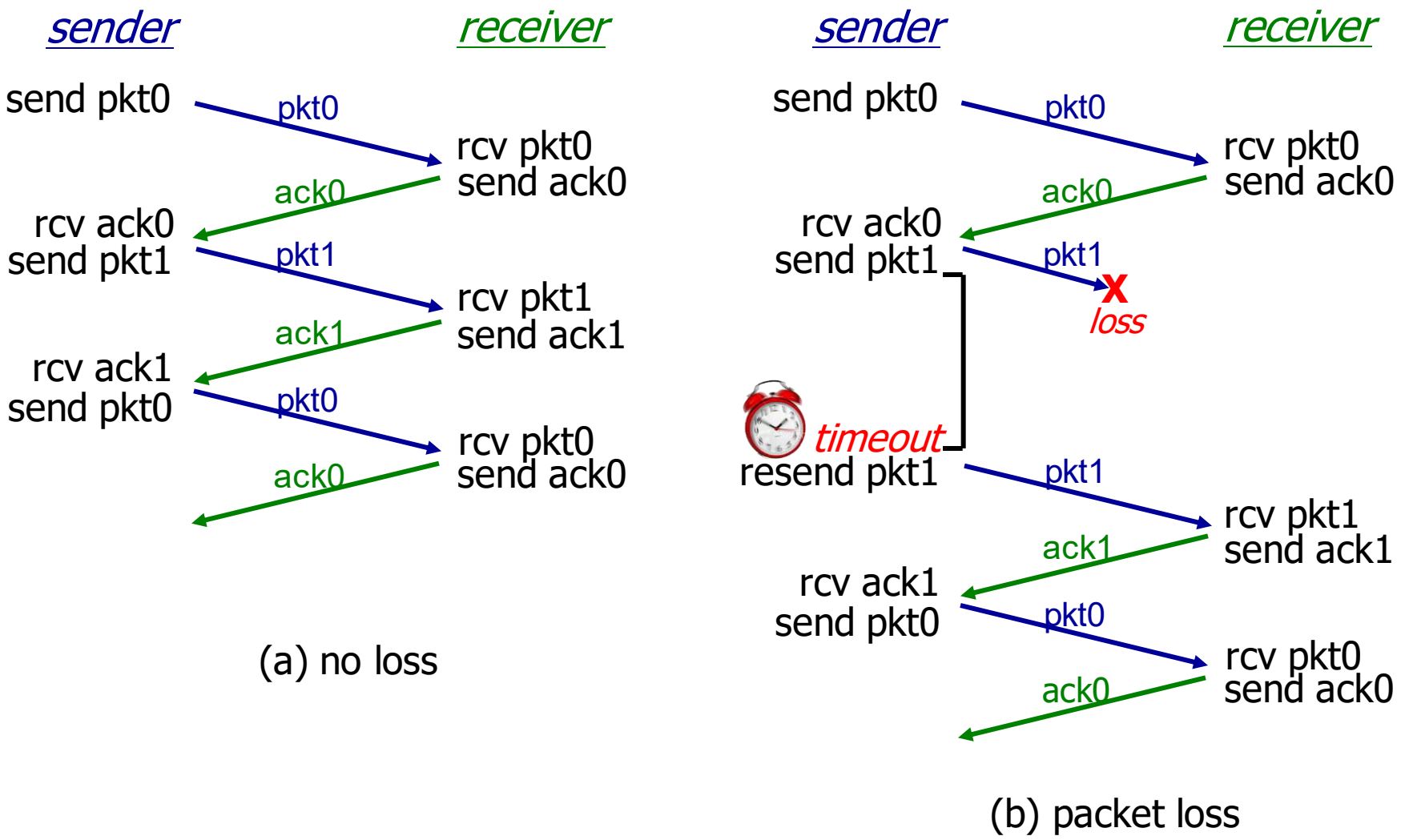
- retransmits if no ACK received in this time
- if pkt (or ACK) just delayed (not lost):
  - retransmission will be duplicate, but seq. #'s already handles this
  - receiver must specify seq # of pkt being ACKed
- requires countdown timer

COMPUTER NETWORKS

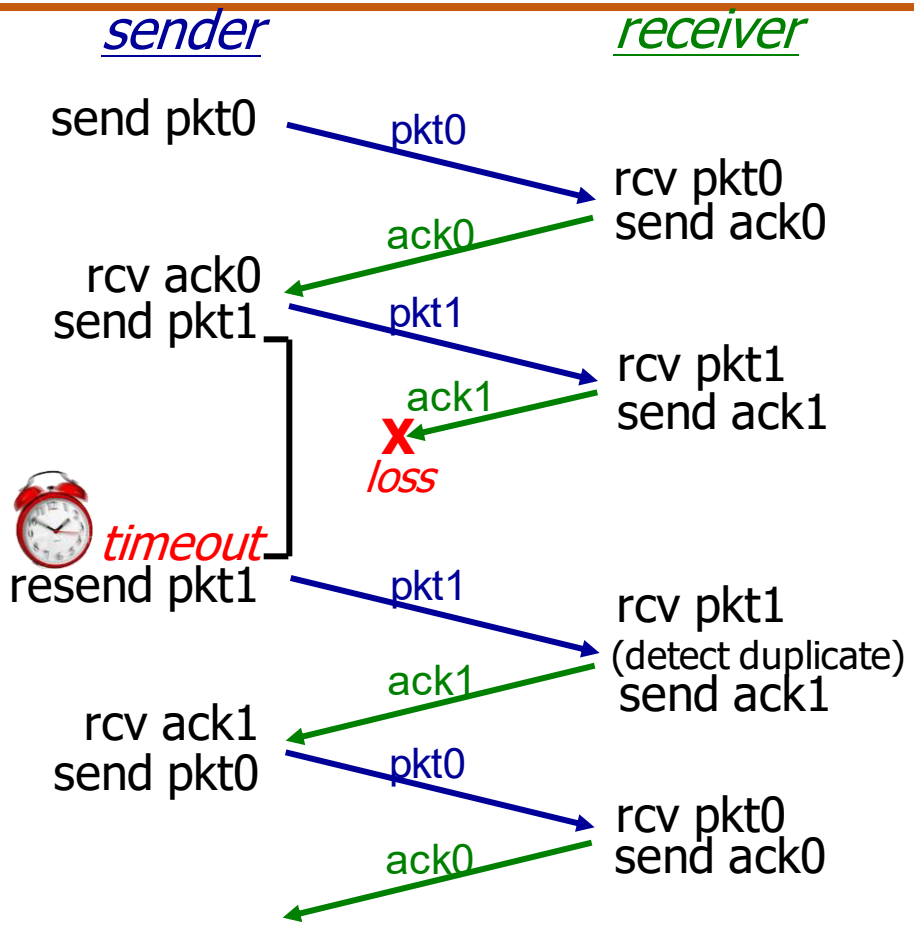
rdt3.0 sender



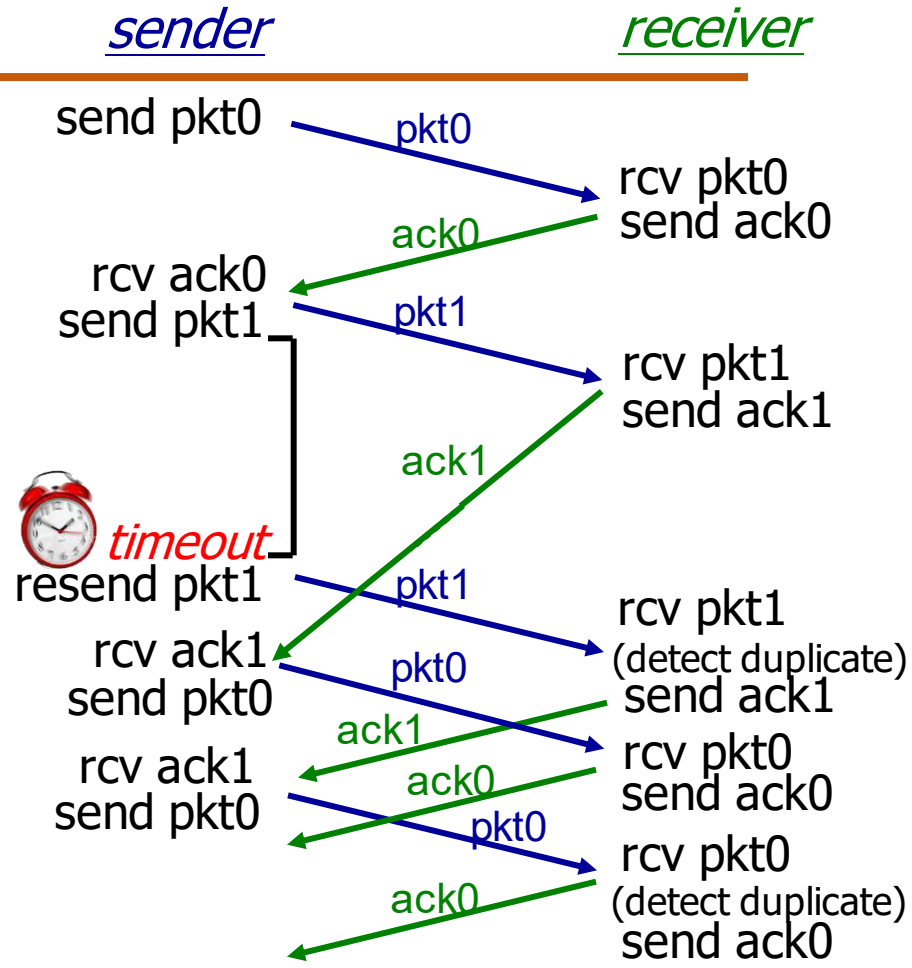
COMPUTER NETWORKS
rdt3.0 in action



rdt3.0 in action



(c) ACK loss



(d) premature timeout/ delayed ACK

Performance of rdt3.0

- rdt3.0 is correct, but performance stinks
- e.g.: 1 Gbps link, 15 ms prop. delay, 8000 bit packet:

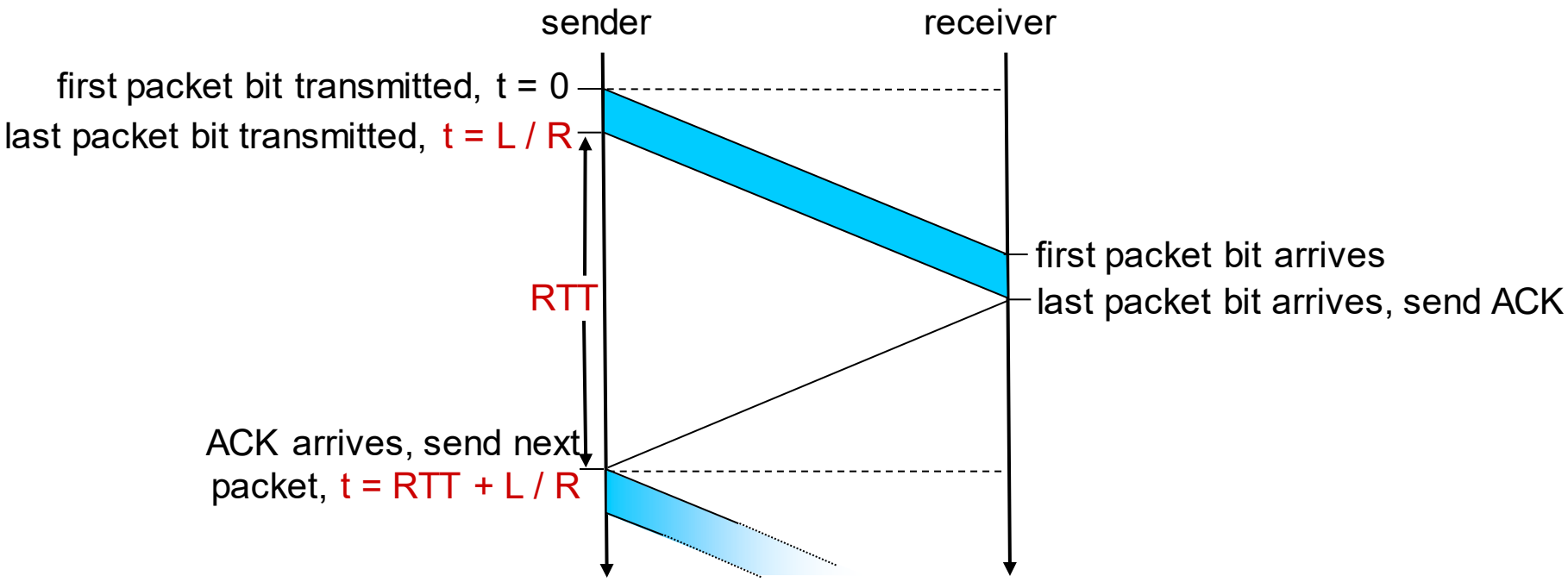
$$D_{trans} = \frac{L}{R} = \frac{8000\ bits}{10^9\ bits/sec} = 8\ microsecs$$

  - U_{sender}: *utilization* – fraction of time sender busy sending

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

    - if RTT=30 msec, 1KB pkt every 30 msec: 33kB/sec thruput over 1 Gbps link
- network protocol limits use of physical resources!

rdt3.0: stop-and-wait operation



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

THANK YOU

Animesh Giri
Department of Computer Science & Engineering

animeshgiri@pes.edu

+91 80 6618 6603