

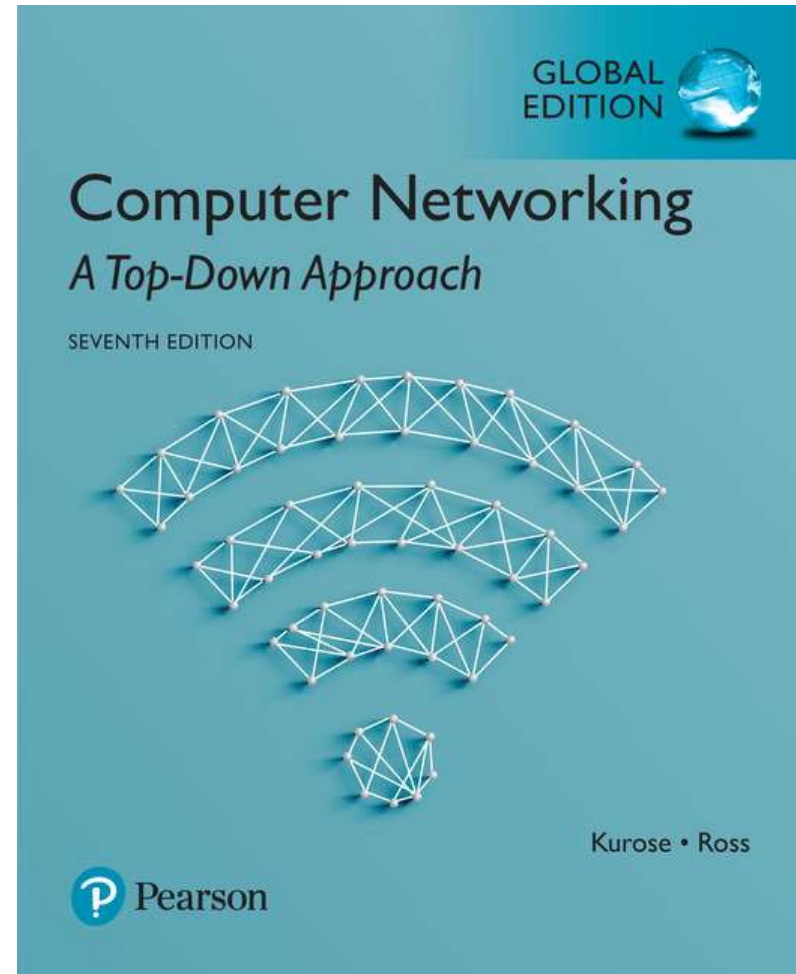
제12강 신뢰 전송의 원리

*Computer Networking: A
Top Down Approach*

컴퓨터 네트워크
(2019년 1학기)

박승철교수

한국기술교육대학교
컴퓨터공학부



Pre-study Test :

1) 손상된 패킷을 검출하는 기술은?

- ① Checksum
- ② Acknowledgement
- ③ Timer
- ④ Retransmission

2) 전송 매체(예, 와이파이 무선링크)상에서 일어날 수 있는 전송 오류는?

- ① 패킷 손실(packet loss)
- ② 패킷 손상(packet corruption)
- ③ 패킷 중복(packet duplicate)
- ④ 패킷 지연(packet delay)

3) 수신자가 정상 수신확인(ACK)과 비정상 수신확인(NAK) 패킷을 전송하고, NAK에 대해 해당 패킷을 재전송하는 오류 복구 방식의 문제점이 아닌 것은?

- ① ACK/ACK 손상이 복구가 불가하다.
- ② 대기 시간이 크다.
- ③ 손실된 패킷 복구가 불가능하다.
- ④ 손상된 패킷 복구가 불가능하다.

4) ACK/NAK 기반의 오류 복구에서 ACK/NAK 손상에 대한 해결책은?

- ① 패킷에 순서번호(sequence number)를 부여한다.
- ② ACK/NAK 손상시 패킷을 재전송한다.
- ③ ACK/NAK 손상시 패킷을 패킷에 순서번호를 부여하고 재전송한다.
- ④ 패킷에 순서번호를 부여하고 ACK/NAK 손상시 패킷을 재전송한다.

5) 송신자가 송신된 패킷의 손실 여부 확인 방법으로 가장 합당한 것은?

- ① 손실된 패킷에 대해 수신자가 NAK을 보내게 한다.
- ② 수신된 패킷에 대해 수신자가 ACK를 보내게 한다.
- ③ 손실된 패킷에 대해 수신자가 NAK을 보내게 하고 일정 시간 동안 기다린다.
- ④ 수신된 패킷에 대해 수신자가 ACK을 보내게 하고 일정 시간 동안 기다린다.

6) 송신된 패킷에 대해 일정 시간 동안 ACK를 기다리고, ACK가 도착하지 않으면 재전송하는 방식(stop-and-wait)의 오류 복구의 문제점이 아닌 것은?

- ① 대기 시간이 크다.
- ② 네트워크 자원 낭비가 크다.
- ③ 손상된 패킷을 복구할 수 없다.
- ④ 패킷이 중복되어 송신될 수 있다.

7) 네트워크 링크의 전송속도가 1Gbps이고 전달 지연시간(propagation delay)가 15ms이다. 길이가 8000비트인 패킷을 stop-and-wait 방식으로 전송할 때 네트워크의 사용율(송신자가 실제로 네트워크를 사용하는 시간 비율)은 얼마인가?

8) 네트워크 사용율을 높이려면 stop-and-wait 대신 어떤 오류 복구 방식을 사용하면 될까?

Chapter 3 outline

3.1 transport-layer services

3.2 multiplexing and demultiplexing

3.3 connectionless transport: UDP

3.4 principles of reliable data transfer

3.5 connection-oriented transport: TCP

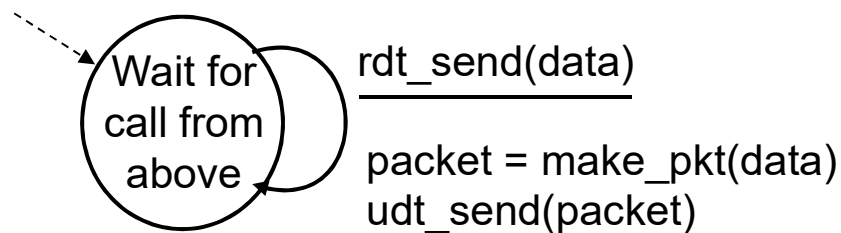
- segment structure
- reliable data transfer
- flow control
- connection management

3.6 principles of congestion control

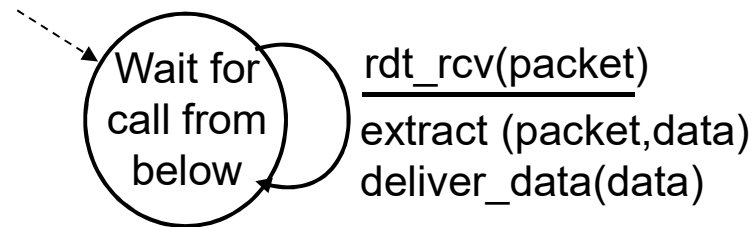
3.7 TCP congestion control

rdt1.0: reliable transfer over a reliable channel

- underlying channel perfectly reliable
 - no bit errors
 - no loss of packets
- separate FSMs for sender, receiver:
 - sender sends data into underlying channel
 - receiver reads data from underlying channel



sender



receiver

rdt2.0: channel with bit errors

- underlying channel may flip bits in packet
 - checksum to detect bit errors
- *the question: how to recover from errors:*
 - *acknowledgements (ACKs):* receiver explicitly tells sender that pkt received OK
 - *negative acknowledgements (NAKs):* receiver explicitly tells sender that pkt had errors
 - sender retransmits pkt on receipt of NAK
- new mechanisms in `rdt2.0` (beyond `rdt1.0`):
 - *error detection*
 - *feedback: control msgs (ACK,NAK) from receiver to sender*

rdt2.0 has a fatal flaw!

what happens if ACK/NAK corrupted?

- sender doesn't know what happened at receiver!
- can't just retransmit: possible duplicate

handling duplicates:

- sender retransmits current pkt if ACK/NAK corrupted
- sender adds *sequence number* to each pkt
- receiver discards (doesn't deliver up) duplicate pkt

stop and wait

sender sends one packet,
then waits for receiver
response

rdt2.1: discussion

sender:

- seq # added to pkt
- two seq. #'s (0,1) will suffice. Why?
- must check if received ACK/NAK corrupted
- twice as many states
 - state must “remember” whether “expected” pkt should have seq # of 0 or 1

receiver:

- must check if received packet is duplicate
 - state indicates whether 0 or 1 is expected pkt seq #
- note: receiver can *not* know if its last ACK/NAK received OK at sender

rdt2.2: a 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
- duplicate ACK at sender results in same action as NAK: **retransmit current pkt**

rdt3.0: channels with errors and loss

new assumption:

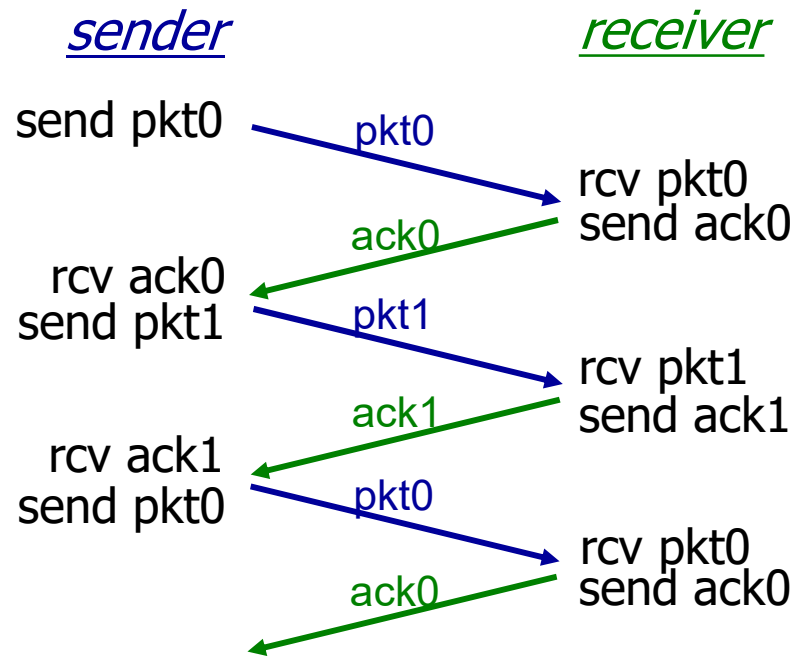
underlying channel can also lose packets (data, ACKs)

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

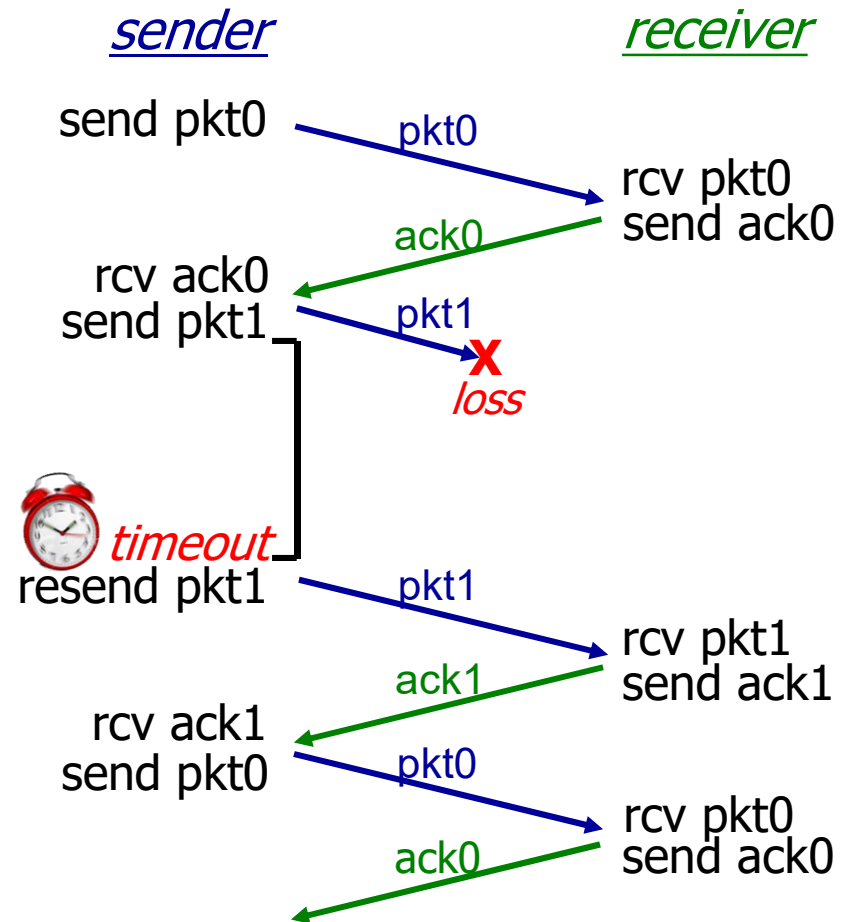
approach: sender waits “reasonable” amount of time for ACK

- 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

rdt3.0 in action

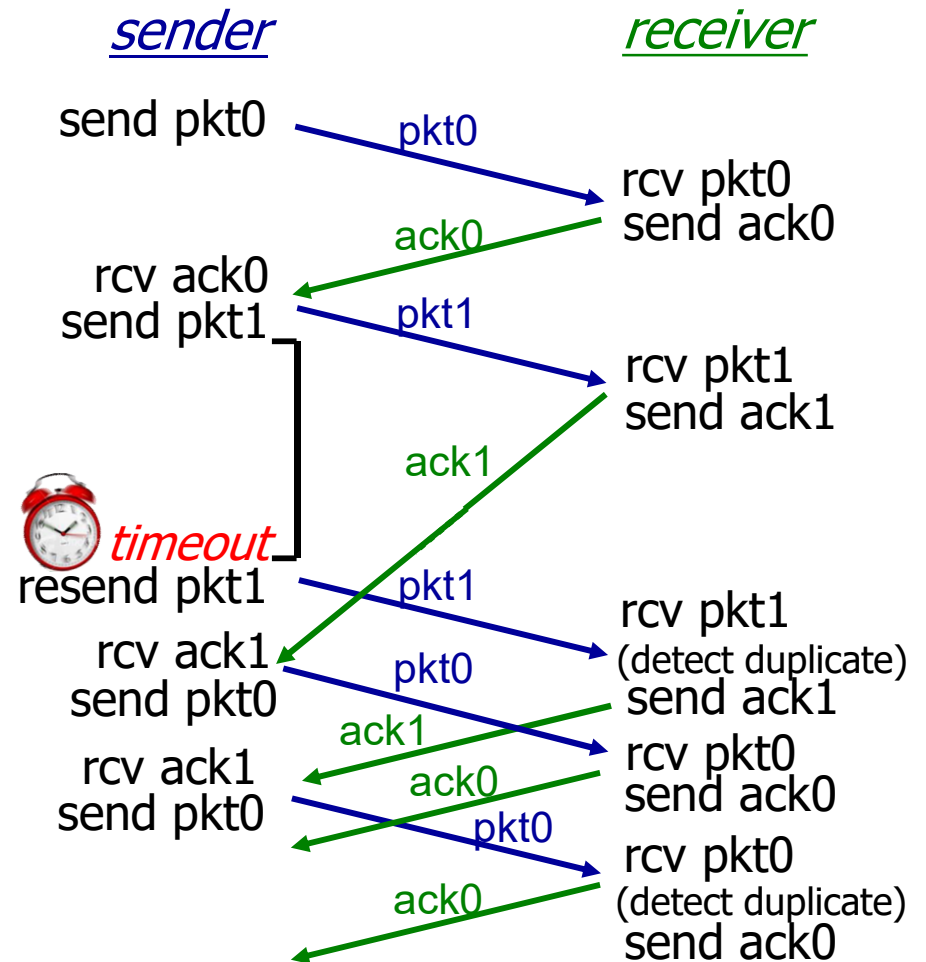
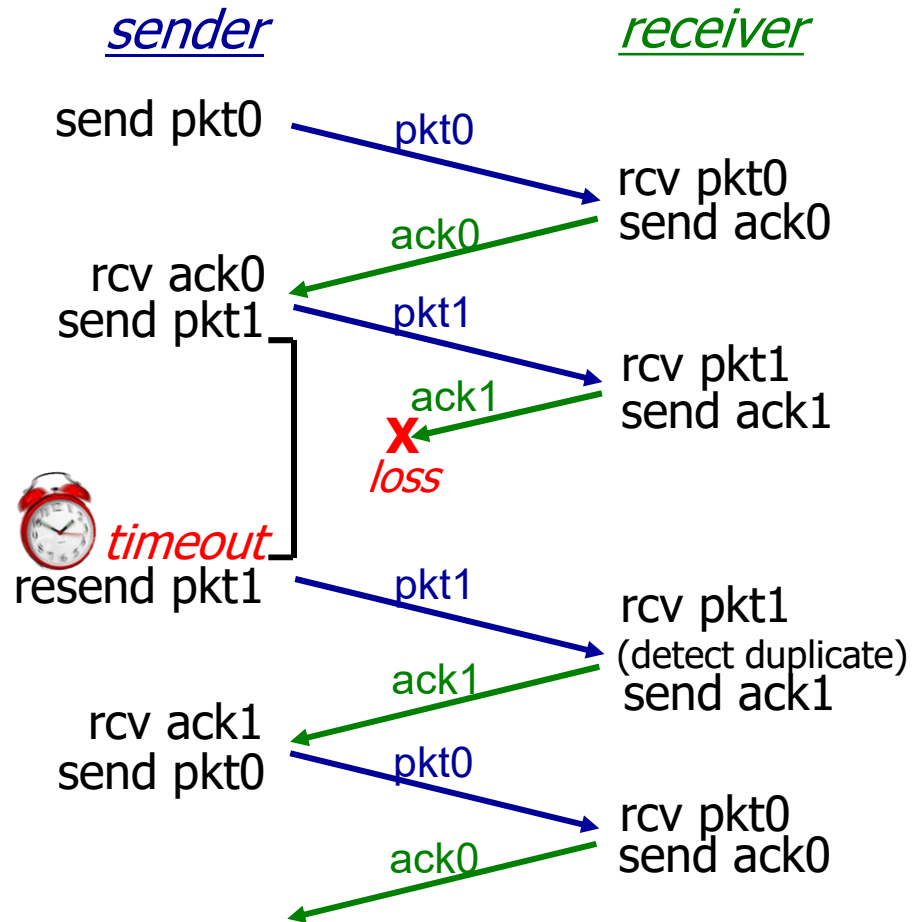


(a) no loss



(b) packet loss

rdt3.0 in action



Performance of rdt3.0

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

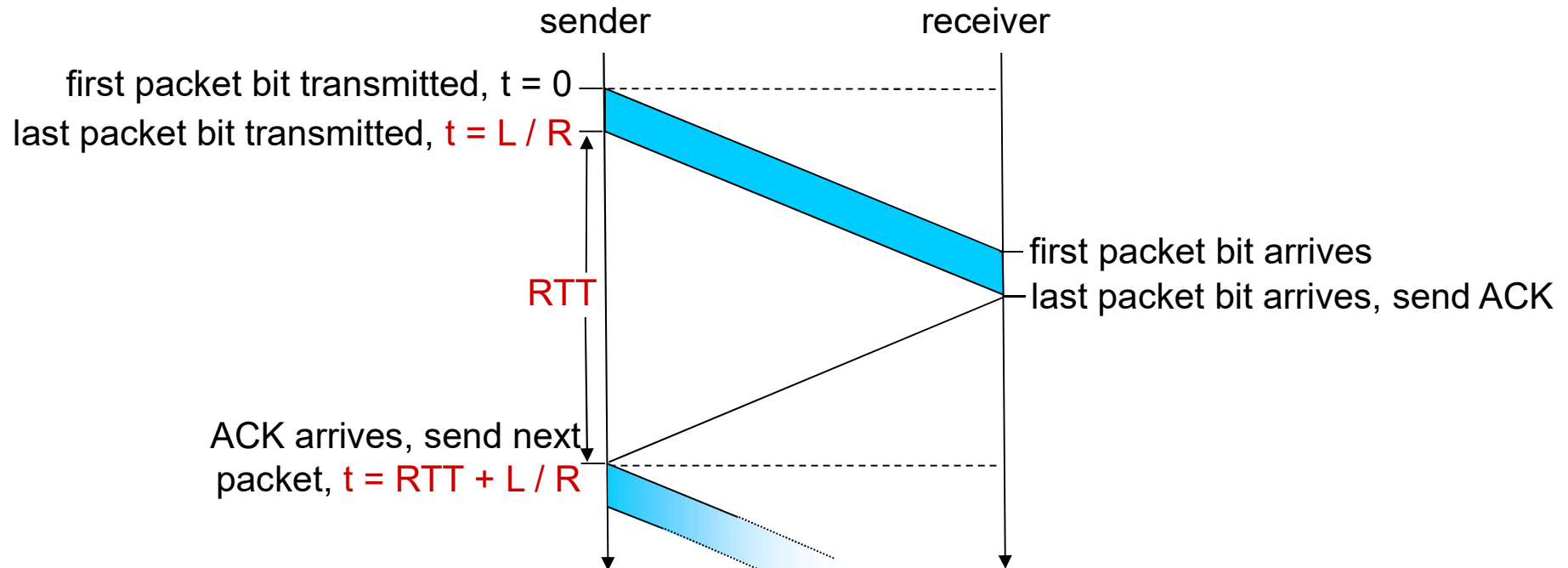
$$D_{trans} = \frac{L}{R} = \frac{8000 \text{ bits}}{10^9 \text{ bits/sec}} = 8 \text{ microsecs}$$

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

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

- 267Kbps effective throughput over 1 Gbps link
- network protocol limits use of physical resources!

rdt3.0: stop-and-wait operation



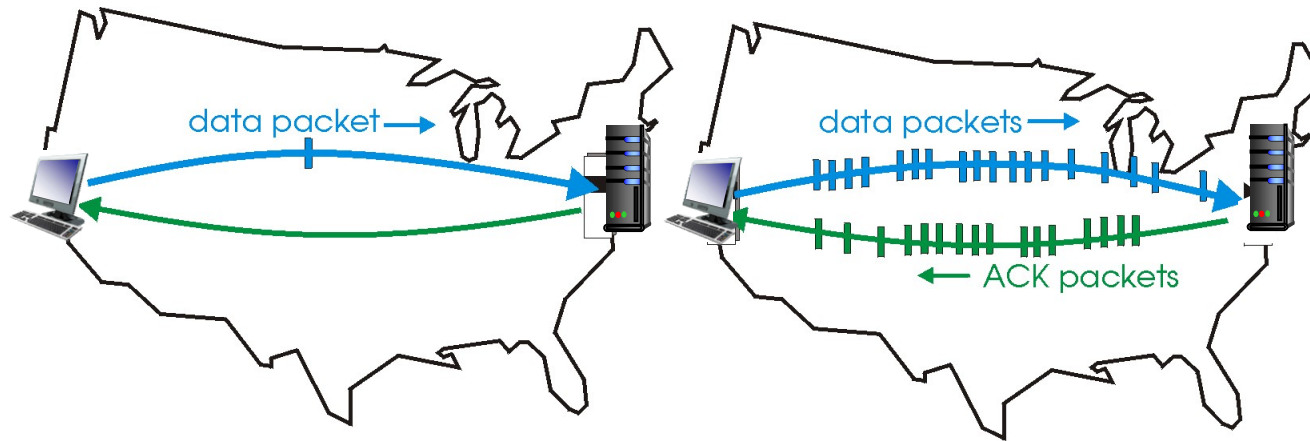
$$U_{\text{sender}} = \frac{L / R}{RTT + L / R} = \frac{.008}{30.008} = 0.00027$$

1Gbps 전송 링크를 사용할 때 유효 전송속도는 얼마인가?

Pipelined protocols

pipelining: sender allows multiple, “in-flight”, yet-to-be-acknowledged pkts

- range of sequence numbers must be increased
- buffering at sender and/or receiver

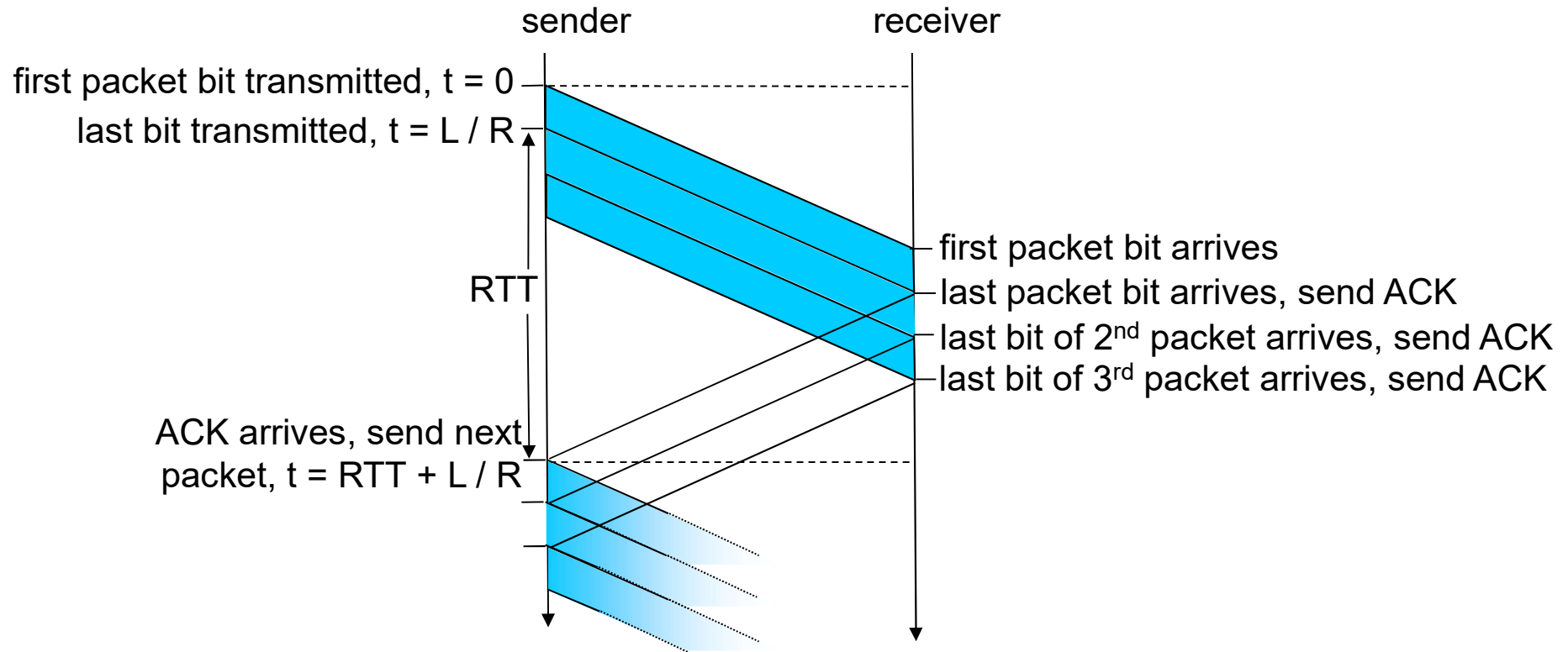


(a) a stop-and-wait protocol in operation

(b) a pipelined protocol in operation

- two generic forms of pipelined protocols: *go-Back-N*, *selective repeat*

Pipelining: increased utilization



$$U_{\text{sender}} = \frac{3L / R}{RTT + L / R} = \frac{.0024}{30.008} = 0.00081$$

After-study Test :

1) 손상된 패킷을 검출하는 기술은?

- ① Checksum
- ② Acknowledgement
- ③ Timer
- ④ Retransmission

2) 전송 매체(예, 와이파이 무선링크)상에서 일어날 수 있는 전송 오류는?

- ① 패킷 손실(packet loss)
- ② 패킷 손상(packet corruption)
- ③ 패킷 중복(packet duplicate)
- ④ 패킷 지연(packet delay)

3) 수신자가 정상 수신확인(ACK)과 비정상 수신확인(NAK) 패킷을 전송하고, NAK에 대해 해당 패킷을 재전송하는 오류 복구 방식의 문제점이 아닌 것은?

- ① ACK/ACK 손상이 복구가 불가하다.
- ② 대기 시간이 크다.
- ③ 손실된 패킷 복구가 불가능하다.
- ④ 손상된 패킷 복구가 불가능하다.

4) ACK/NAK 기반의 오류 복구에서 ACK/NAK 손상에 대한 해결책은?

- ① 패킷에 순서번호(sequence number)를 부여한다.
- ② ACK/NAK 손상시 패킷을 재전송한다.
- ③ ACK/NAK 손상시 패킷을 패킷에 순서번호를 부여하고 재전송한다.
- ④ 패킷에 순서번호를 부여하고 ACK/NAK 손상시 패킷을 재전송한다.

5) 송신자가 송신된 패킷의 손실 여부 확인 방법으로 가장 합당한 것은?

- ① 손실된 패킷에 대해 수신자가 NAK을 보내게 한다.
- ② 수신된 패킷에 대해 수신자가 ACK를 보내게 한다.
- ③ 손실된 패킷에 대해 수신자가 NAK을 보내게 하고 일정 시간 동안 기다린다.
- ④ 수신된 패킷에 대해 수신자가 ACK을 보내게 하고 일정 시간 동안 기다린다.

6) 송신된 패킷에 대해 일정 시간 동안 ACK를 기다리고, ACK가 도착하지 않으면 재전송하는 방식(stop-and-wait)의 오류 복구의 문제점이 아닌 것은?

- ① 대기 시간이 크다.
- ② 네트워크 자원 낭비가 크다.
- ③ 손상된 패킷을 복구할 수 없다.
- ④ 패킷이 중복되어 송신될 수 있다.

7) 네트워크 링크의 전송속도가 1Gbps이고 전달 지연시간(propagation delay)가 15ms이다. 길이가 8000비트인 패킷을 stop-and-wait 방식으로 전송할 때 네트워크의 사용율(송신자가 실제로 네트워크를 사용하는 시간 비율)은 얼마인가?

8) 네트워크 사용율을 높이려면 stop-and-wait 대신 어떤 오류 복구 방식을 사용하면 될까?