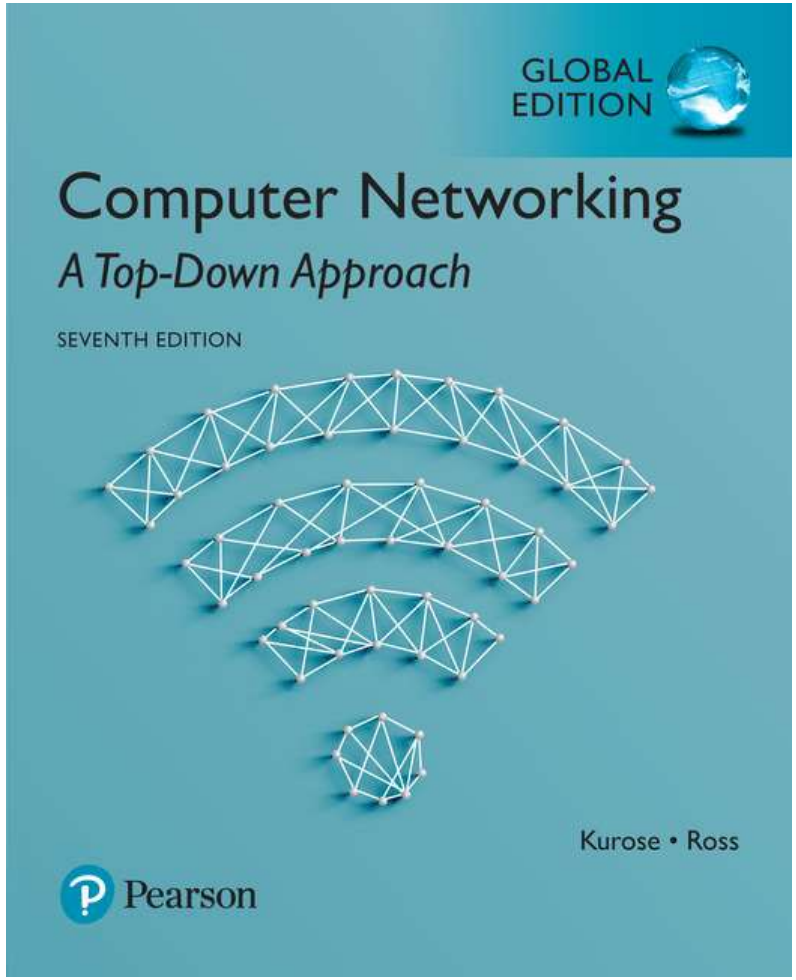
제14강
TCP 연결, 세그먼트, 타임아웃 설정

Computer Networking: A Top Down Approach

컴퓨터 네트워크
(2019년 1학기)

박승철교수

한국기술교육대학교
컴퓨터공학부



Pre-study Test :

1) TCP의 특징이 아닌 것은?

① Point-to-Point 통신만 지원한다.
② 바이트 단위로 순서번호를 매긴다.
③ 양방향 전송을 위해 클라이언트와 서버는 2개의 소켓을 설정한다.
④ 세그먼트의 최대크기는 제한된다.

2) TCP 세그먼트의 헤더에 포함될 필요가 없는 것은?

① 출발지 포트번호
② 순서 번호
③ 수신 윈도우
④ 송신 윈도우

3) i 번째 데이터 세그먼트의 순서 번호가 K이고 사용자 데이터 크기가 L이면 (i+1) 번째 세그먼트의 순서 번호는?

① K
② K+1
③ K+L
④ K+L+1

4) 수신된 세그먼트의 순서 번호가 K이고 세그먼트의 크기는 N 바이트이다.
확인번호는 무엇인가?

① K
② K+N-1
③ K+N
④ K+N-20

5) 재전송 타이머의 타임아웃 크기가 너무 작으면 어떤 문제가 발생하는가?

① 불필요한 대기 시간이 작아서 효율이 높아진다.
② 불필요한 재전송이 너무 자주 발생한다.
③ 정상 수신 확인에 유리하다.
④ 패킷 손실을 빨리 확인할 수 있다.

6) 재전송 타임아웃 설정에 사용할 가장 바람직한 RTT값은?

① 현재 RTT 측정값
② 전체 RTT의 평균값
③ 현재 RTT 측정값에 높은 가중치를 둔 가중 평균값
④ 과거 RTT 측정값에 높은 가중치를 둔 가중 평균값

7) 재전송 타임아웃 주기를 어떻게 설정하여야 하는가?

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

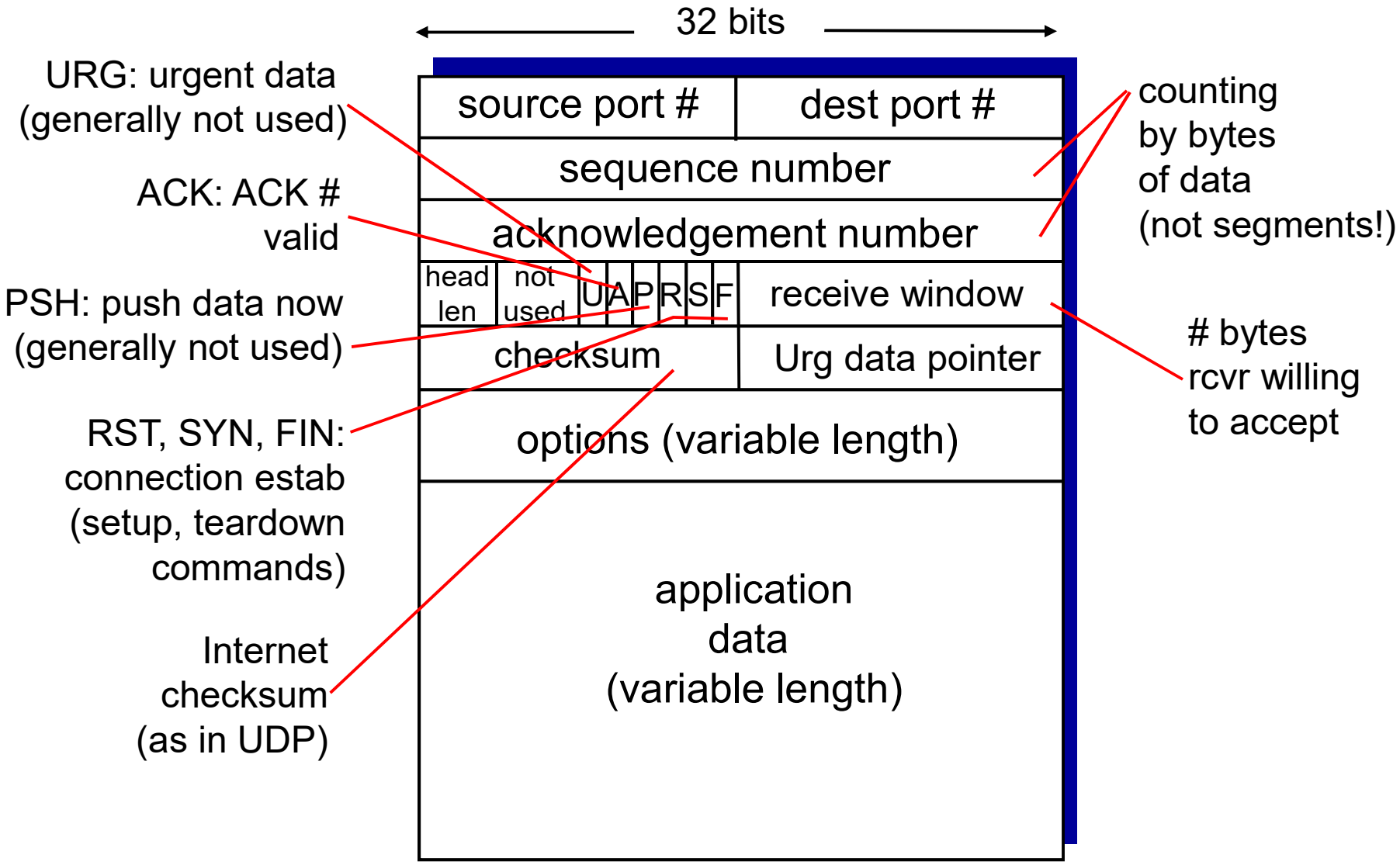
3.6 principles of congestion control

3.7 TCP congestion control

TCP: Overview RFCs: 793,1122,1323, 2018, 2581

- point-to-point:
  - one sender, one receiver
- reliable, in-order *byte steam:*
  - no "message boundaries"
- pipelined:
  - TCP congestion and flow control set window size
- full duplex data:
  - bi-directional data flow in same connection
  - MSS: maximum segment size
- connection-oriented:
  - handshaking (exchange of control msgs) inits sender, receiver state before data exchange
- flow controlled:
  - sender will not overwhelm receiver

TCP segment structure



TCP seq. numbers, ACKs

sequence numbers:

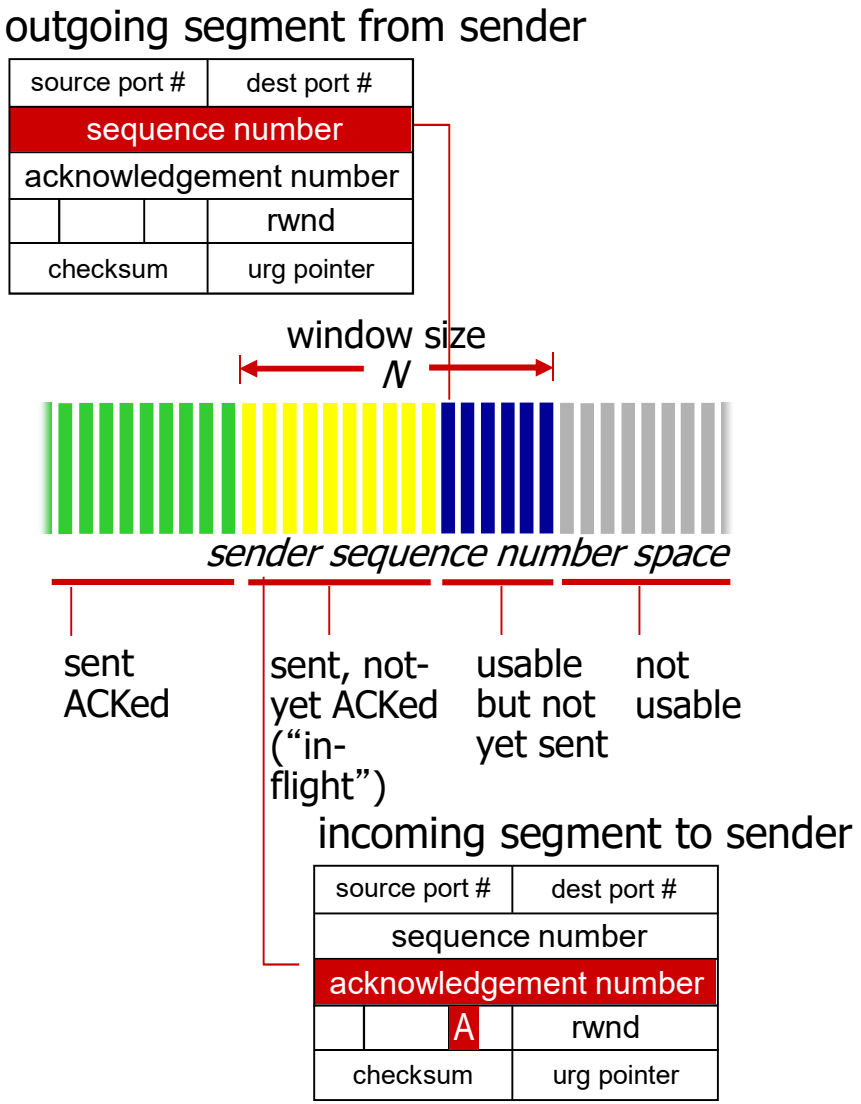
- byte stream "number" of first byte in segment's data

acknowledgements:

- seq # of next byte expected from other side
- cumulative ACK

Q: how receiver handles out-of-order segments

- A: TCP spec doesn't say, - up to implementor



TCP seq. numbers, ACKs

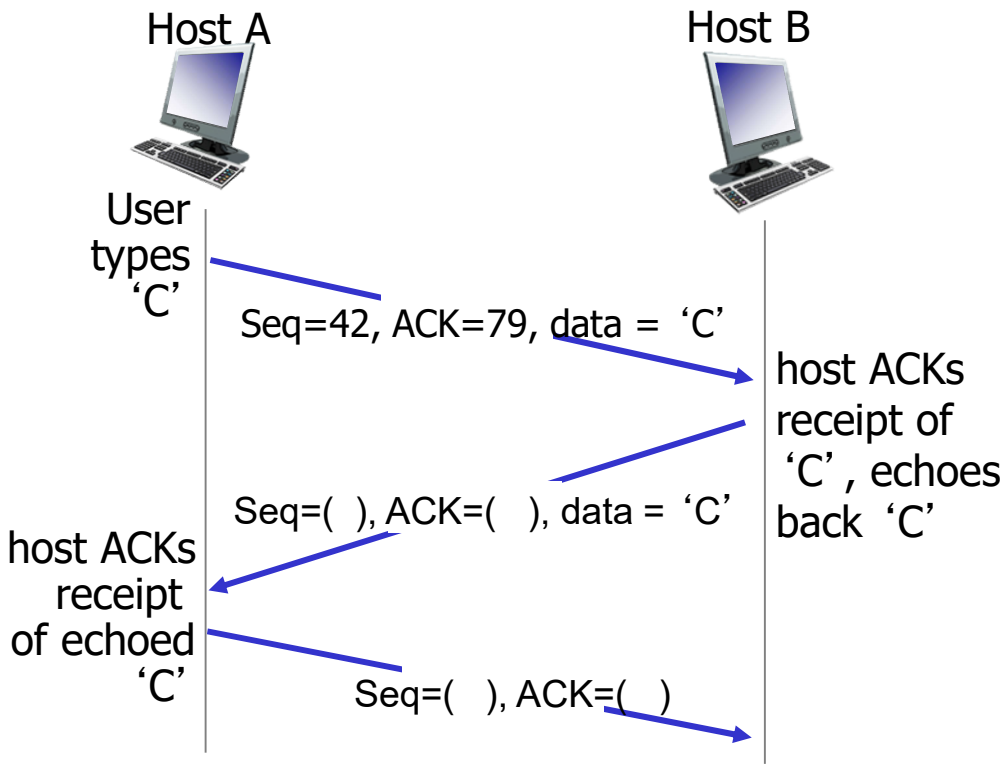
- 순서 번호 부여
  - ① 바이트 단위로 순서 번호 부여
  - ② 세그먼트 데이터의 첫 번째 바이트 번호

➢ i 번째 데이터 세그먼트의 순서 번호가 K이고 사용자 데이터 크기가 L이면 (i+1) 번째 세그먼트의 순서 번호는?

TCP seq. numbers, ACKs

- 확인 번호 부여
  - ① 다음에 수신할 세그먼트의 순서 번호 표시(수신한 데이터의 마지막 바이트 번호가 아님)
  - ② 수신된 세그먼트의 순서 번호가 K이고 세그먼트의 크기는 N 바이트이다. 확인번호는 무엇인가?

TCP seq. numbers, ACKs



simple telnet scenario

TCP round trip time, timeout

Q: how to set TCP timeout value?

- longer than RTT
  - but RTT varies
- *too short:* premature timeout, unnecessary retransmissions
- *too long:* slow reaction to segment loss

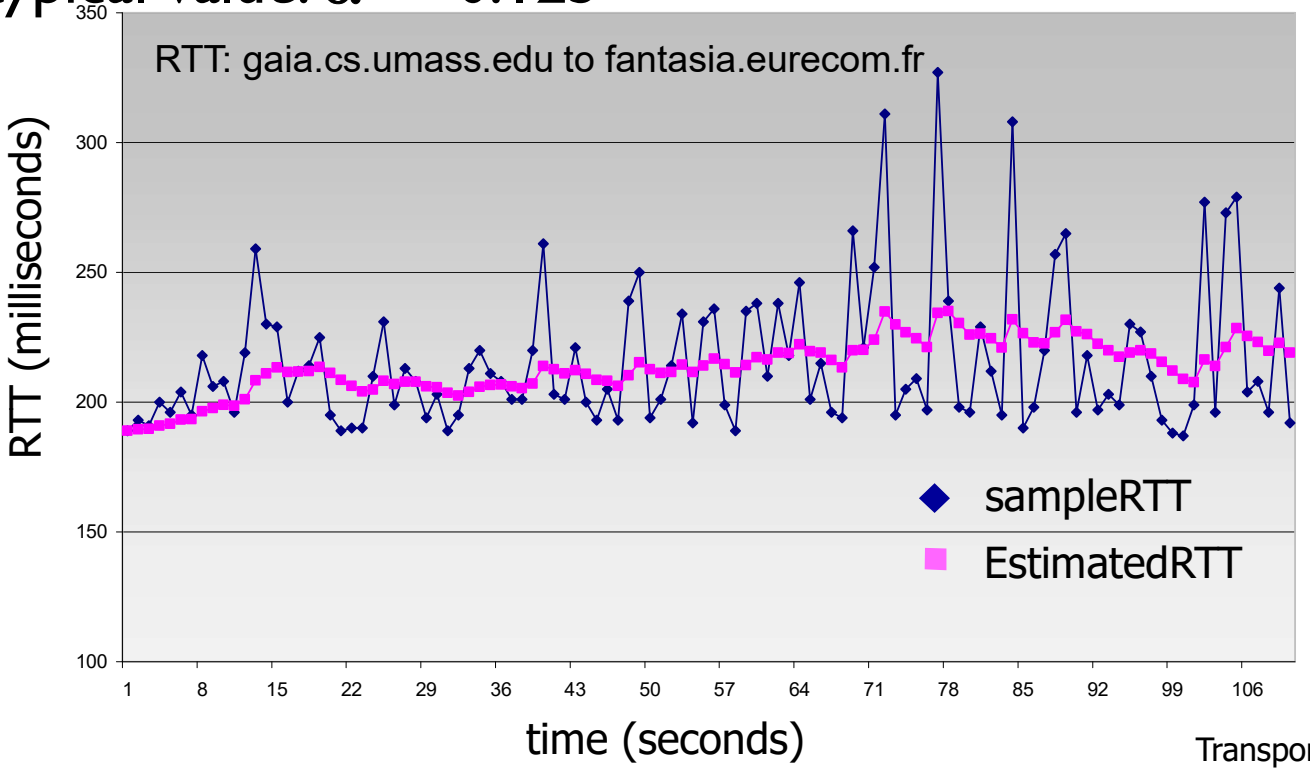
Q: how to estimate RTT?

- **`SampleRTT`**: measured time from segment transmission until ACK receipt
  - ignore retransmissions
- **`SampleRTT`** will vary, want estimated RTT "smoother"
  - average several *recent* measurements, not just current **`SampleRTT`**

TCP round trip time, timeout

EstimatedRTT = (1- α)*EstimatedRTT + α*SampleRTT

- exponential weighted moving average (지수이동가중평균)
- influence of past sample decreases exponentially fast
- typical value: α = 0.125



TCP round trip time, timeout

- 초기 EstimatedRTT를 E0라 하고, 새롭게 측정되는 샘플 RTT를 순서대로 S1, S2, S3라 한다.

① S3를 측정한 시점에 S1, S2, S3가 EstimatedRTT에 반영되는 비율 계산식을 구하라.

TCP round trip time, timeout

- timeout interval: **`EstimatedRTT`** plus "safety margin"
  - large variation in **`EstimatedRTT`** -> larger safety margin
- estimate SampleRTT deviation from EstimatedRTT:

$$\texttt{DevRTT} = (1-\beta)*\texttt{DevRTT} + \beta*|\texttt{SampleRTT}-\texttt{EstimatedRTT}|$$

(typically, $\beta = 0.25$)

$$\texttt{TimeoutInterval} = \texttt{EstimatedRTT} + 4*\texttt{DevRTT}$$

estimated RTT

"safety margin"

* Check out the online interactive exercises for more examples: http://gaia.cs.umass.edu/kurose_ross/interactive/

TCP round trip time, timeout

```
DevRTT = (1-β)*DevRTT +
              β*|SampleRTT-EstimatedRTT|
```

문제 : 위의 식을 해석하시오.

After-study Test :

1) TCP의 특징이 아닌 것은?

① Point-to-Point 통신만 지원한다.
② 바이트 단위로 순서번호를 매긴다.
③ 양방향 전송을 위해 클라이언트와 서버는 2개의 소켓을 설정한다.
④ 세그먼트의 최대크기는 제한된다.

2) TCP 세그먼트의 헤더에 포함될 필요가 없는 것은?

① 출발지 포트번호
② 순서 번호
③ 수신 윈도우
④ 송신 윈도우

3) i 번째 데이터 세그먼트의 순서 번호가 K이고 사용자 데이터 크기가 L이면 (i+1) 번째 세그먼트의 순서 번호는?

① K
② K+1
③ K+L
④ K+L+1

4) 수신된 세그먼트의 순서 번호가 K이고 세그먼트의 크기는 N 바이트이다.
확인번호는 무엇인가?

① K
② K+N-1
③ K+N
④ K+N-20

5) 재전송 타이머의 타임아웃 크기가 너무 작으면 어떤 문제가 발생하는가?

① 불필요한 대기 시간이 작아서 효율이 높아진다.
② 불필요한 재전송이 너무 자주 발생한다.
③ 정상 수신 확인에 유리하다.
④ 패킷 손실을 빨리 확인할 수 있다.

6) 재전송 타임아웃 설정에 사용할 가장 바람직한 RTT값은?

① 현재 RTT 측정값
② 전체 RTT의 평균값
③ 현재 RTT 측정값에 높은 가중치를 둔 가중 평균값
④ 과거 RTT 측정값에 높은 가중치를 둔 가중 평균값

7) 재전송 타임아웃 주기를 어떻게 설정하여야 하는가?