

Chapter 06. TCP와 NAT

TCP 통신

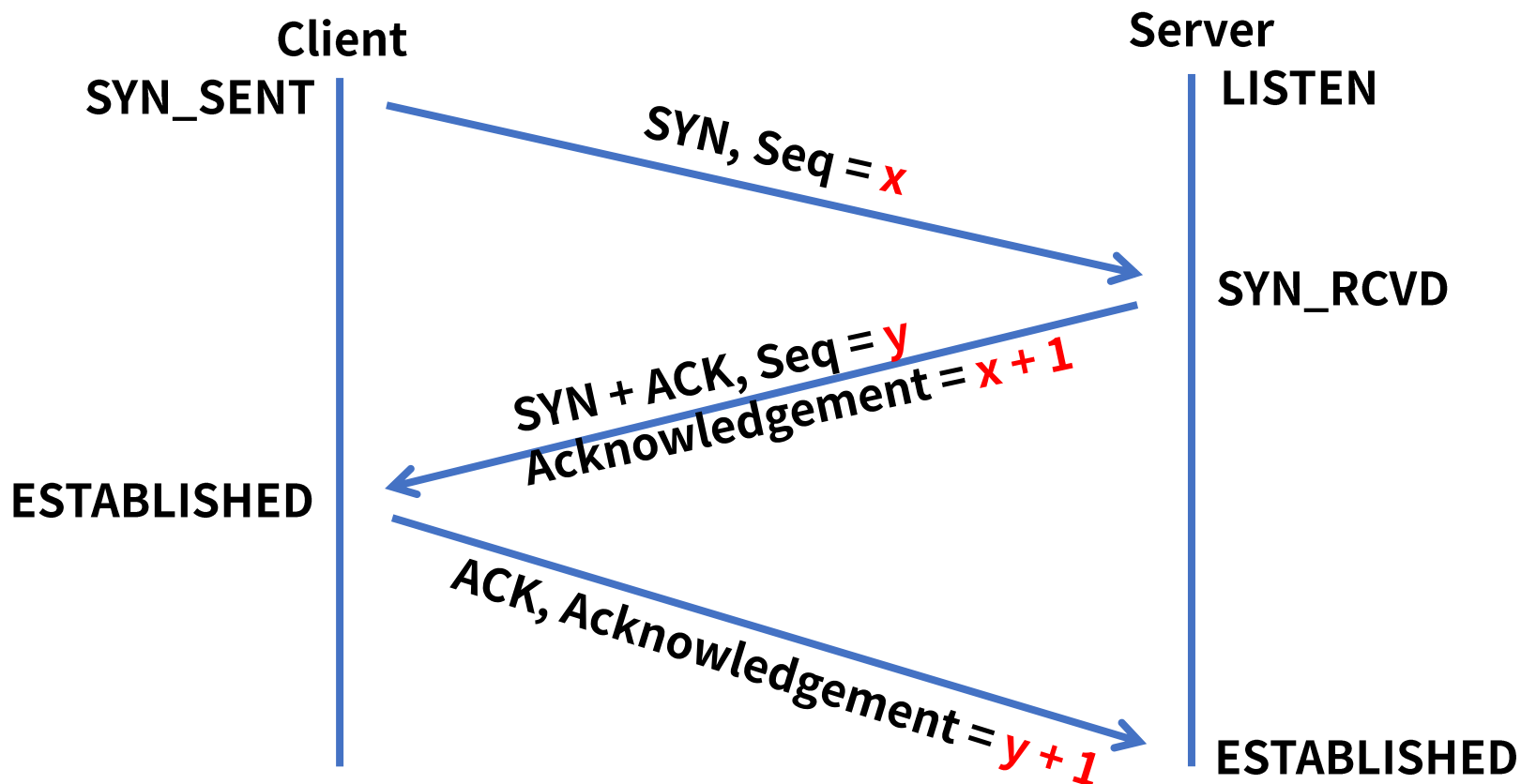
목차

- TCP 통신 과정
- 흐름 제어
- 혼잡 제어

TCP 통신 과정

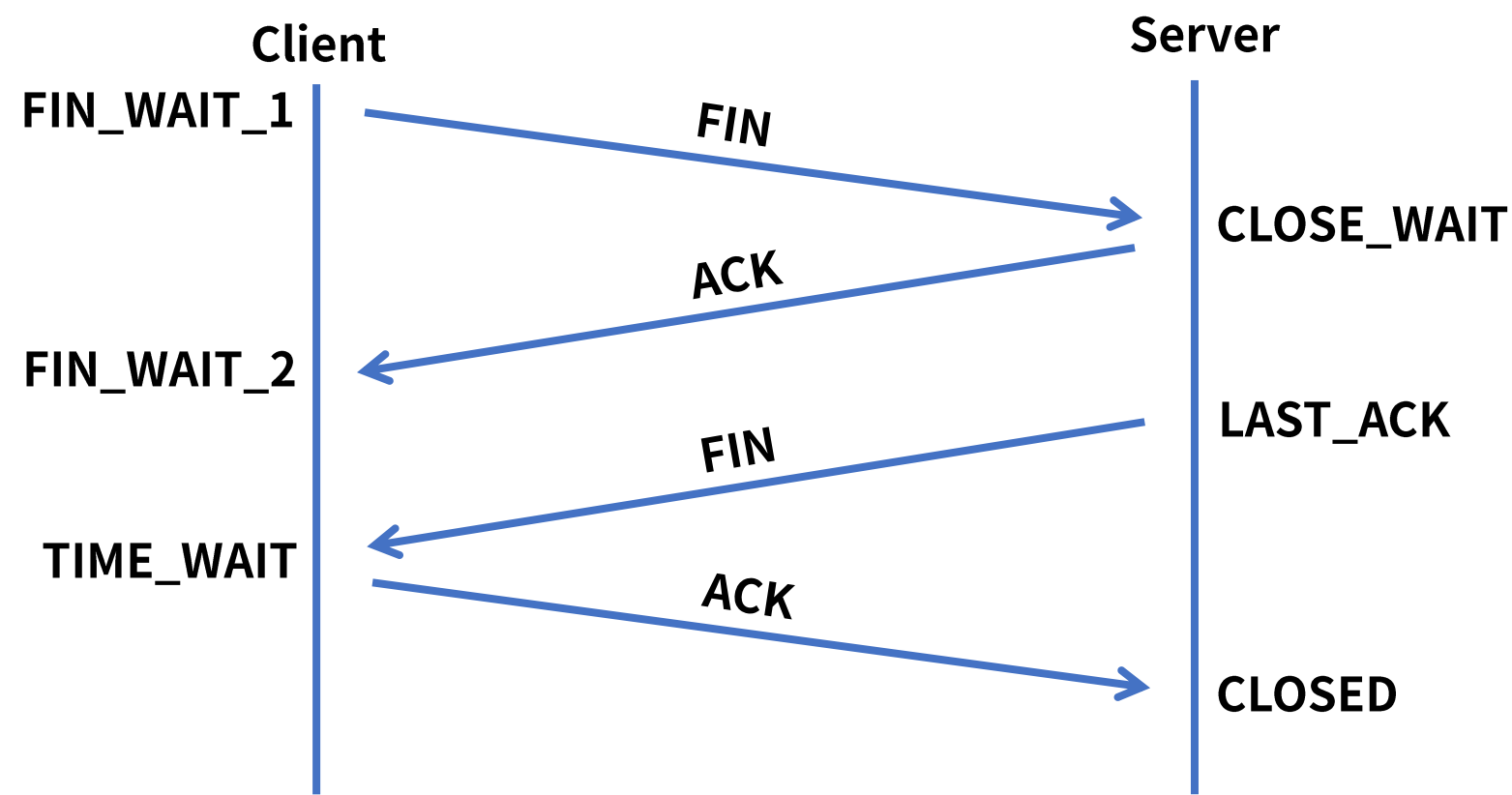
- 3 way handshake

TCP는 연결 지향 프로토콜로 두 호스트가 통신하기 전에 연결을 위한 관계를 수립



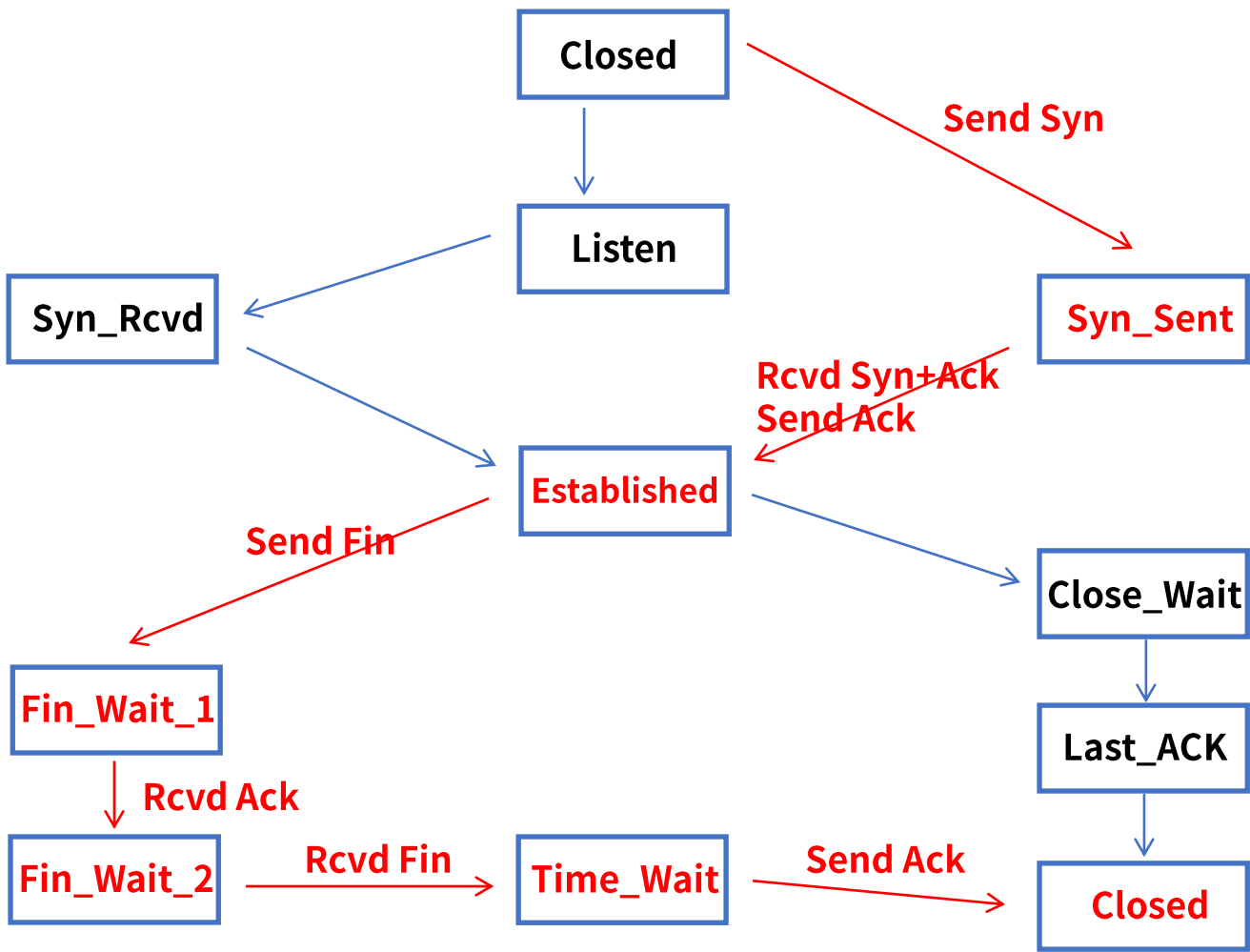
TCP 통신 과정

- 4 way handshake
연결을 종료



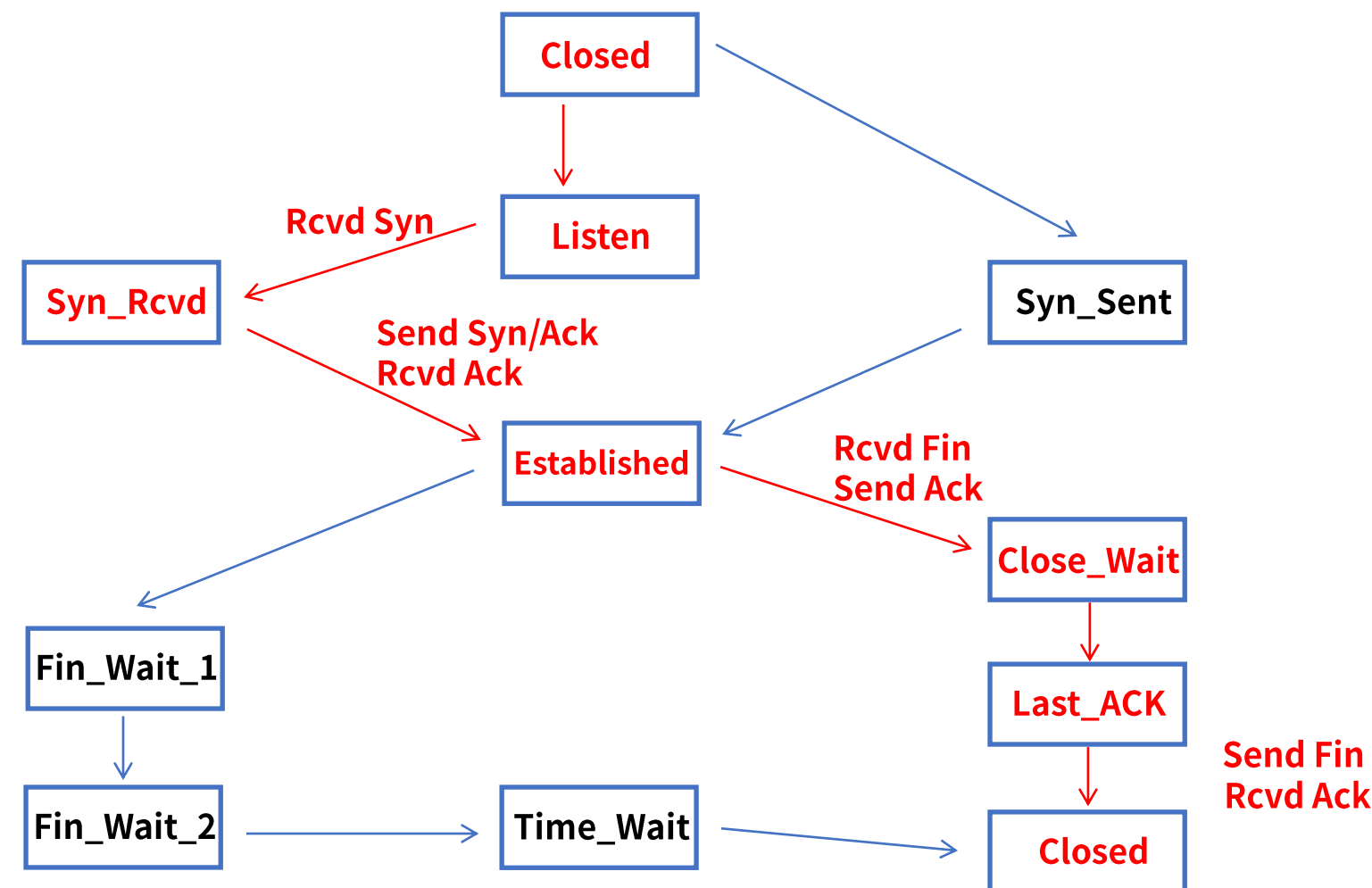
TCP 통신 과정

- TCP 상태 전이도 - Client



TCP 통신 과정

- TCP 상태 전이도 - Server



TCP 통신 과정

- TCP 타이머 - Retransmission

송신측이 패킷을 매번 전송할 때 카운트

RTO(Retransmission Timeout) 내 ACK 응답이 오지 않으면 재전송
RTO는 RTT(Round Trip Time)에 따라서 가변적으로 변함

SRTT(Smoothed Round-Trip Time), RTTVAR (Round-Trip Time Variation)
 $\alpha=1/8$, $\beta=1/4$, R = 측정된 RTT 값, G = clock granularity

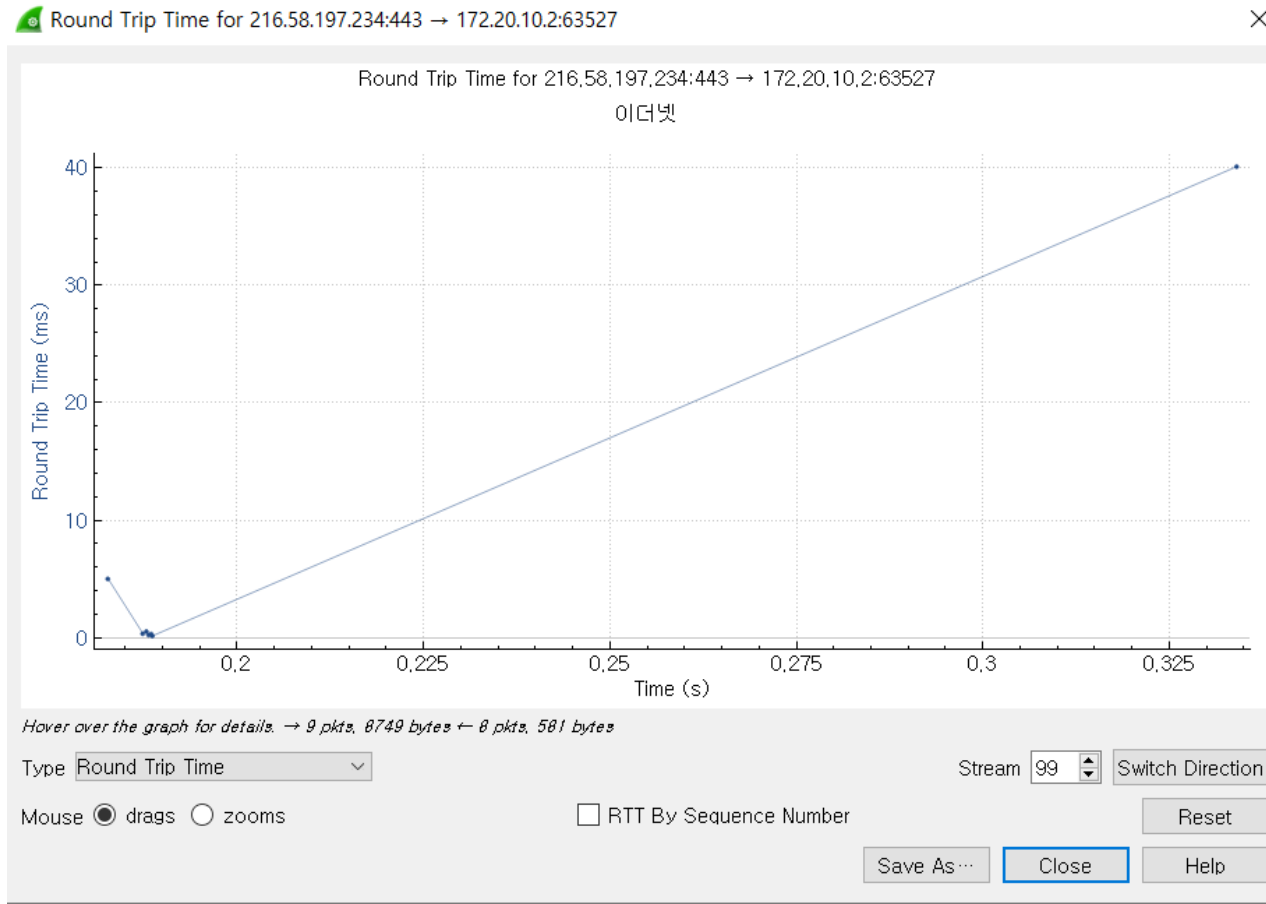
$$\text{RTTVAR} = (1 - \beta) * \text{RTTVAR} + \beta * |\text{SRTT} - R|$$
$$\text{SRTT} = (1 - \alpha) * \text{SRTT} + \alpha * R$$
$$\text{RTO} = \text{SRTT} + \max(G, 4 * \text{RTTVAR})$$

<https://tools.ietf.org/rfc/rfc6298.txt>

TCP 통신 과정

- RTT - PCAP

Wireshark > Menu > Statistics > TCP Stream Graph > Round Trip Time graph



TCP 통신 과정

- TCP 타이머 - Persistence

윈도우 사이즈 관련 타이머

수신측에서 용량 부족으로 윈도우 사이즈 없음을 보내고 다시 용량에 여유가 생기면 송신측에 요청

중간에서 윈도우 사이즈 > 0 을 보내는 ACK이 유실되면 서로 통신 간 문제 발생

수신측 윈도우 사이즈 = 0 을 보낼 경우 Persistence 타이머 가동 - RTO

Persistence 타이머가 종료되면 Probe(ACK 재전송 요청)를 보내고 타이머 재 가동

다시 타이머가 종료되기 전에 ACK을 수신 못하면 시간을 2배로 늘리고 Probe 재 전송

타이머의 임계치는 60초

TCP 통신 과정

- **TCP 타이머 - Time waited**

TCP 연결 종료 후에 특정 시간만 연결을 유지

MSL(Maximum Segment Lifetime) = 120초, TIME_WAIT = 2MSL

다른 연결이 맺어진 상태에서 이전 연결의 지연/중복 패킷 도착으로 인한 문제 발생

- **TCP 타이머 - Keepalive**

TCP 연결 유지 타이머

TCP 연결을 맺고 수신측에서 2시간 동안 송신하는 패킷이 없으면 수신측은 75초 단위로 Probe 전송

Probe 9개를 보내고 응답이 없으면 연결 종료

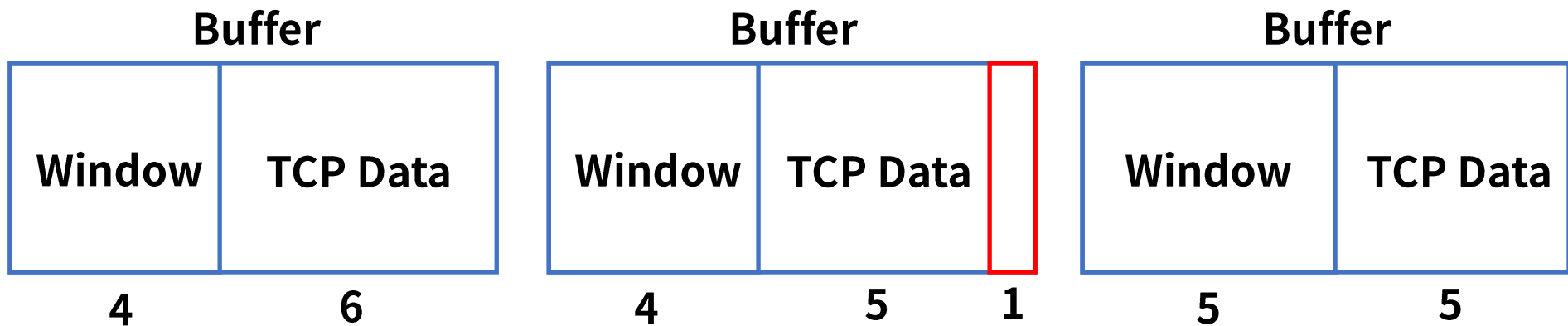
Probe 9개 이전에 응답이 있으면 타이머는 재 설정됨

흐름 제어

- **Flow Control**

송신과 수신측의 데이터 처리 속도 차이를 해결

Sliding Window 기법, Window = 데이터 전송을 위한 버퍼



1. 버퍼 사이즈 중 Window = 4
2. TCP Data 중 1을 프로세스에서 Read 처리
3. Window Size = 5, 송신측에서 Size 확인 후 데이터 전송

윈도우 사이즈 = 마지막 수신한 데이터 - 프로세스가 처리한 데이터

혼잡 제어

- **Congestion Control**

수신측으로 유입되는 트래픽의 양이 정해진 대역폭을 넘어가지 않도록 제어

1. AIMD(Additive Increase/Multiplicative Decrease)

패킷 전송시 문제 없으면 Windows size 1씩 증가, 타임아웃 또는 loss 시 패킷 속도 1/2 감소
초기에 높은 대역폭 사용 불가, 미리 혼잡 상태 감지 불가

2. Slow Start

패킷 전송시 문제 없으면 Windows size 2배씩 증가, 혼잡 상태 발생시 1로 변경
사전 혼잡 상태를 기록하고 Windows size 절반까지 2배씩 증가 후 1씩 증가

3. Fast Retransmit - TCP Tahoe / Fast Recovery - TCP Reno

수신측에서 먼저 와야 하는 패킷이 오지 않고 다음 패킷이 오게 되어도 송신측에 ACK을 보냄
송신측은 타임아웃 시간을 기다리지 않고 중복된 순번의 패킷을 3개 받으면 재전송

4. 개선된 Fast Retransmit / Fast Recovery

TCP New Reno, SACK(TCP Tahoe + Selective Retransmit)

Wrap up

- TCP는 초기 연결은 3 way handshake로 정상적인 연결 종료는 4 way handshake를 사용한다
- TCP 타이머는 Retransmission, Persistence, Time waited, Keepalive가 있다
- Flow Control은 송신과 수신측의 데이터 처리 속도 차이를 해결하는 방법
- Congestion Control은 수신측으로 유입되는 트래픽의 양이 정해진 대역폭을 넘어가지 않도록 제어