

# 프론트 엔드 개발자가 알아야 하는 컴퓨터 공학 지식

## 컴퓨터 네트워크

컴퓨터 네트워크 | 프론트 엔드 개발자가 알아야 하는 CS 지식

강사 나동빈

# 프론트 엔드 개발자가 알아야 하는 컴퓨터 공학 지식

## 컴퓨터 네트워크

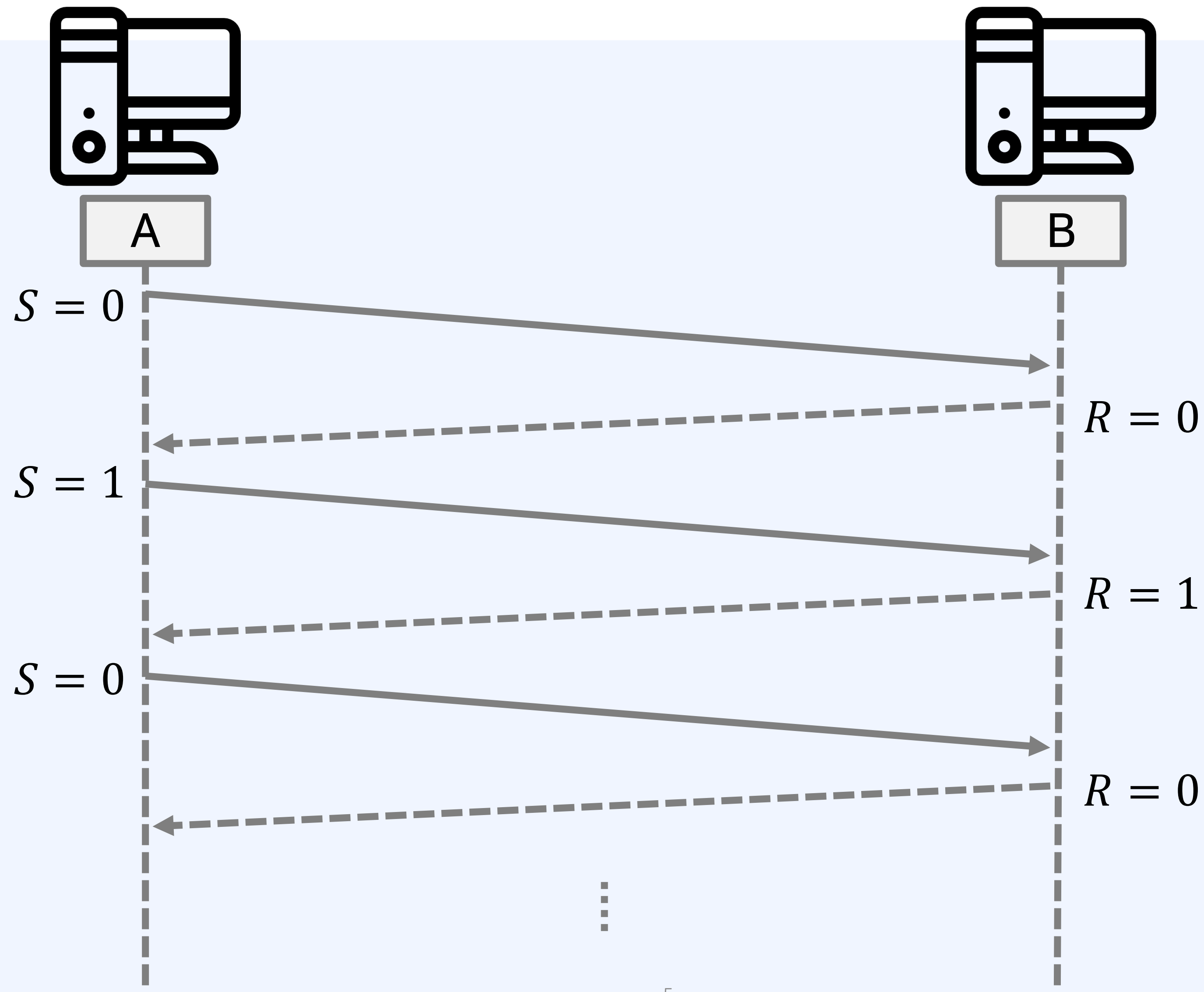
## TCP/IP 흐름 제어

- 데이터를 받는 수신 측이 송신 측보다 데이터 처리 속도가 느릴 때 문제가 발생한다.
  - 송신 측이 데이터를 너무 빠르게 보내면, 수신 측의 용량을 초과한 패킷이 소실될 수 있다.
- TCP/IP 흐름 제어는 그러한 문제를 완화한다.
  - 흐름 제어를 통해 TCP 버퍼(TCP 세그먼트 보관 메모리)의 크기 차이를 완화한다.

## TCP/IP 흐름 제어 방법 ① Stop and Wait

1. 수신 측은 송신 측이 전송한 패킷에 대하여 매번 확인 응답(ACK)을 보낸다.
  2. 송신 측은 ACK를 확인하면, 그 다음 패킷을 전송한다.
- 패킷을 하나씩 보내기 때문에, 속도 측면에서 비효율적으로 동작한다.

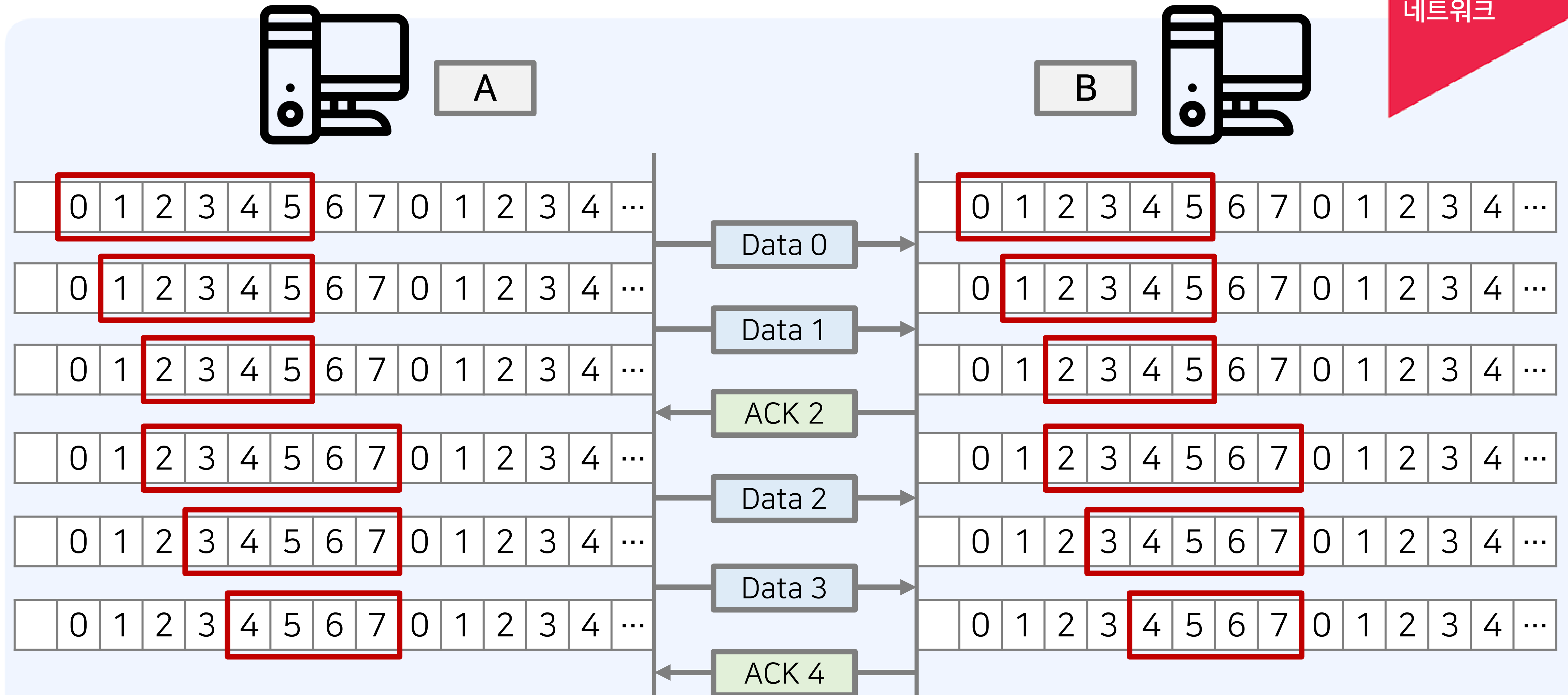
## TCP/IP 흐름 제어 방법 ① Stop and Wait



## TCP/IP 흐름 제어 방법 ② Sliding Window

- 수신 측에서 설정한 윈도우 크기만큼 송신 측이 패킷을 전달한다.
  - 초기 윈도우 크기는 호스트 간의 3 Way Handshaking 과정에서 수신 측이 결정한다.
- 수신 측이 확인 응답(ACK)을 보낼 때 TCP 헤더에 담아서 전송한다.

## TCP/IP 흐름 제어 방법 ② Sliding Window



## TCP/IP 혼잡 제어

- 라우터가 처리할 수 있는 데이터의 양을 초과하면, 데이터가 손실될 수 있다.
- 송신 측에서는 라우터가 처리하지 못한 데이터에 대해 재전송하므로, 네트워크가 혼잡해질 수 있다.
- 따라서, 네트워크의 **오버플로우(overflow)**를 **예방**하기 위해, 패킷 수를 조절한다.

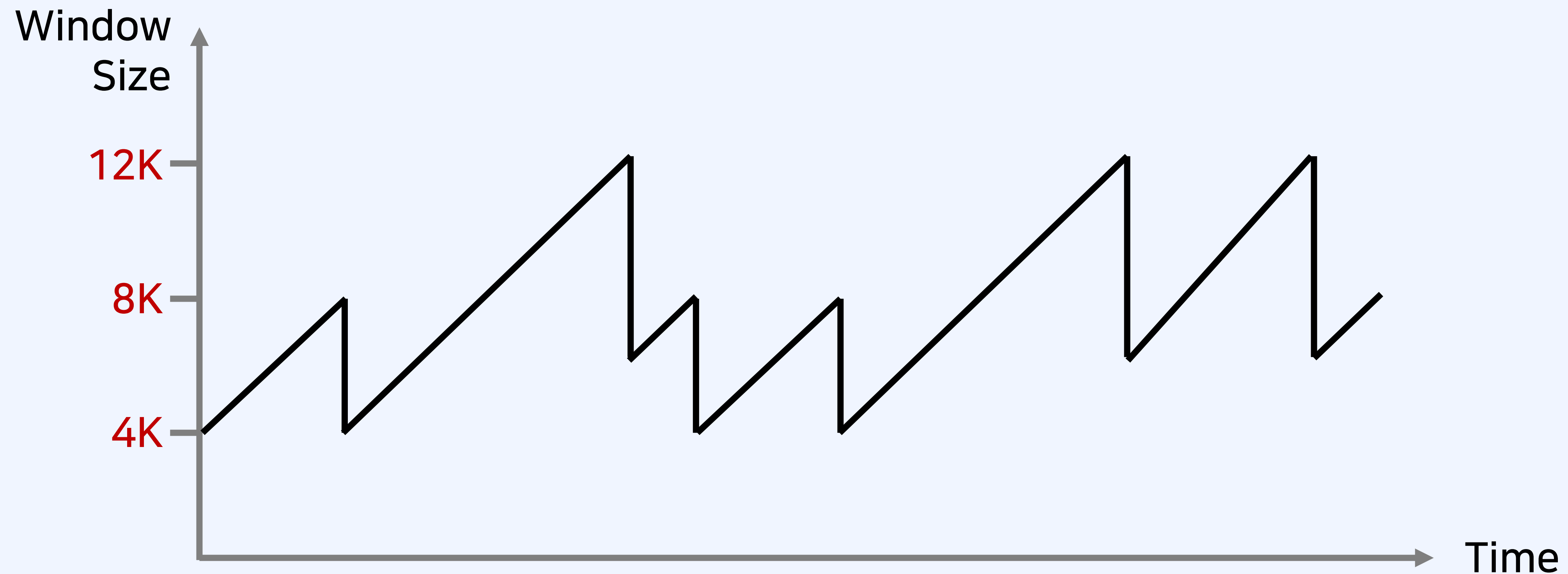


## TCP/IP 혼잡 제어 방법 ① AIMD

- 합 증가/곱 감소(Additive Increase/Multiplicative Decrease) 방식이다.
  1. 처음에 패킷을 하나씩 보내며, 문제가 없다면 윈도우의 크기를 1씩 증가시키며 전송한다.
  2. 전송에 실패했을 때, 윈도우 크기를 반으로 감소시킨다.→ 윈도우의 크기가 선형적으로 증가하여 네트워크의 대역폭을 모두 활용하기에 오랜 시간이 소요될 수 있다.

## TCP/IP 혼잡 제어 방법 ① AIMD

1. 처음에 패킷을 하나씩 보내며, 문제가 없다면 윈도우의 크기를 1씩 증가시킨다.
2. 전송에 실패했을 때, 윈도우 크기를 반으로 감소시킨다.



## TCP/IP 혼잡 제어 방법 ② Slow Start

- 윈도우의 크기를  $\{1, 2, 4, 8, \dots\}$  형식으로 지수적으로 증가시키는 방식이다.
- [기본 아이디어] 혼잡 상태가 감지되었을 때, 윈도우 크기를 1로 감소시킨다.
- 보낸 패킷 데이터에 대하여 **확인 응답(ACK)**이 정상적으로 도착하였을 때 윈도우 크기를 증가시킨다.

## TCP/IP 혼잡 제어 방법 ③ Fast Recovery

- 다양한 방식을 적절히 조합한 다양한 혼잡 제어 정책들이 존재한다.
- 예시) Slow Start 방식으로 동작하다가, 혼잡 상태가 한 번 감지된 이후에는 AIMD 방식으로 동작하도록 할 수 있다.