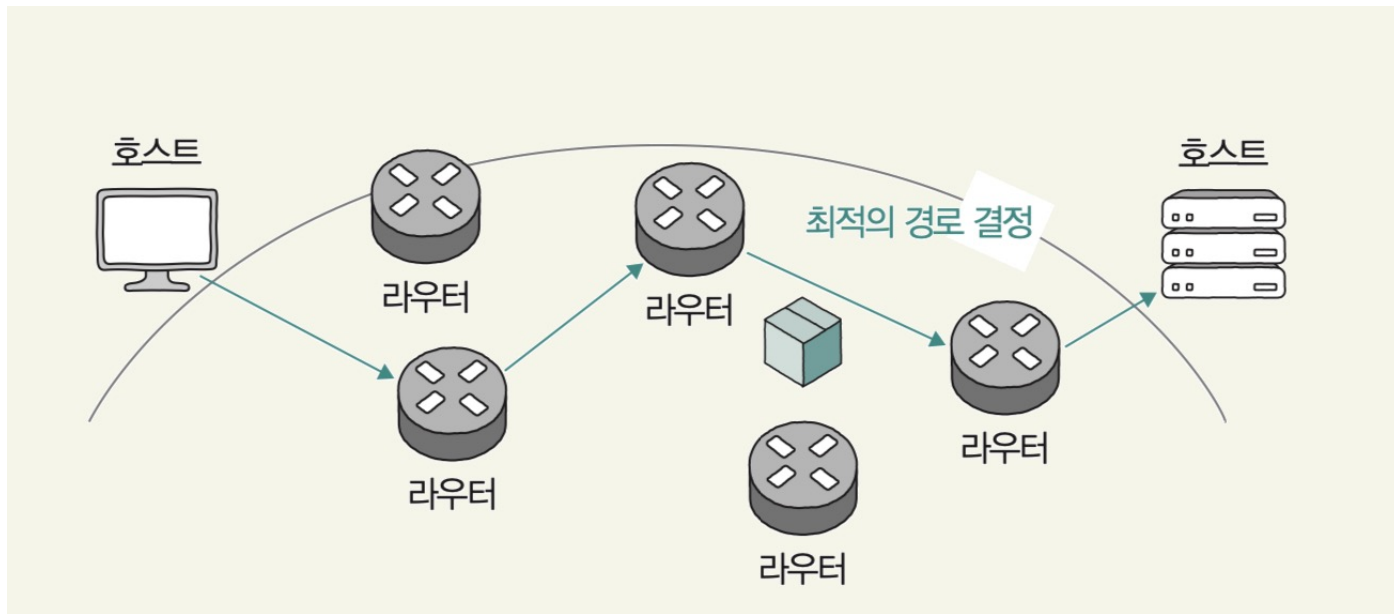


전송 계층 - TCP와 UDP

장동호

지난 시간 복습

- 네트워크 계층 = 네트워크-네트워크 사이의 통신

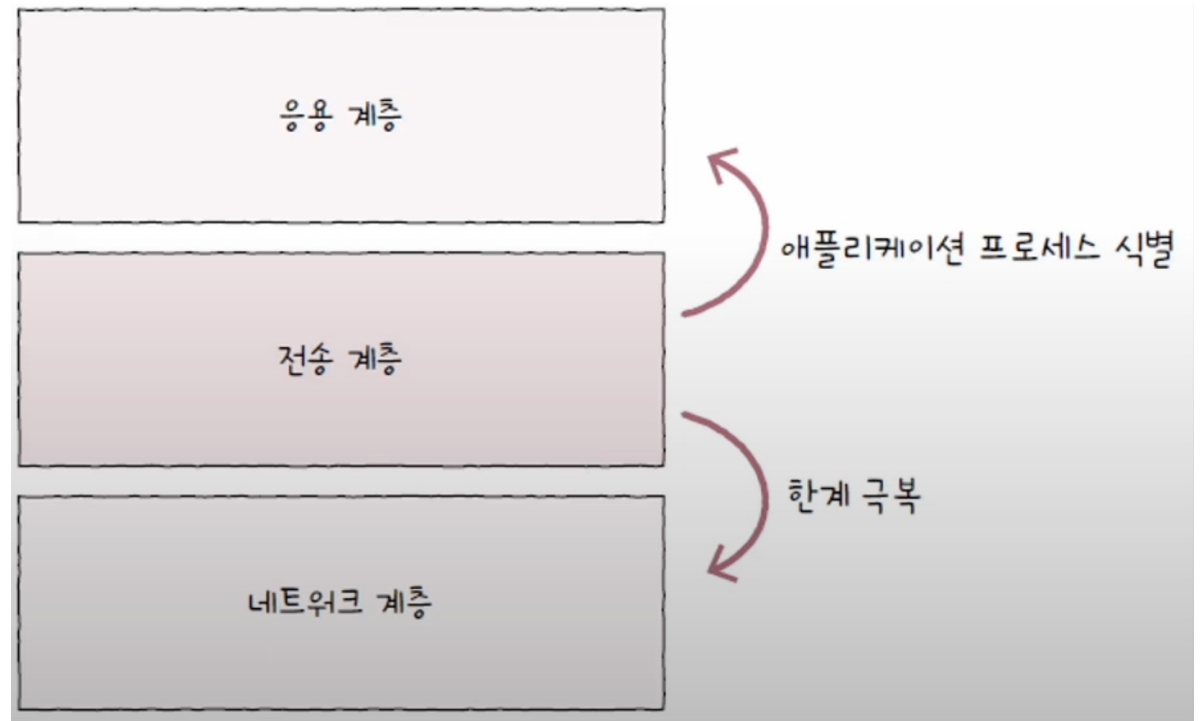


IP 프로토콜

- 비연결형 서비스
- 데이터 손실 가능성
- 순서 보장 없음
- 흐름 제어나 혼잡 제어 미지원

전송 계층

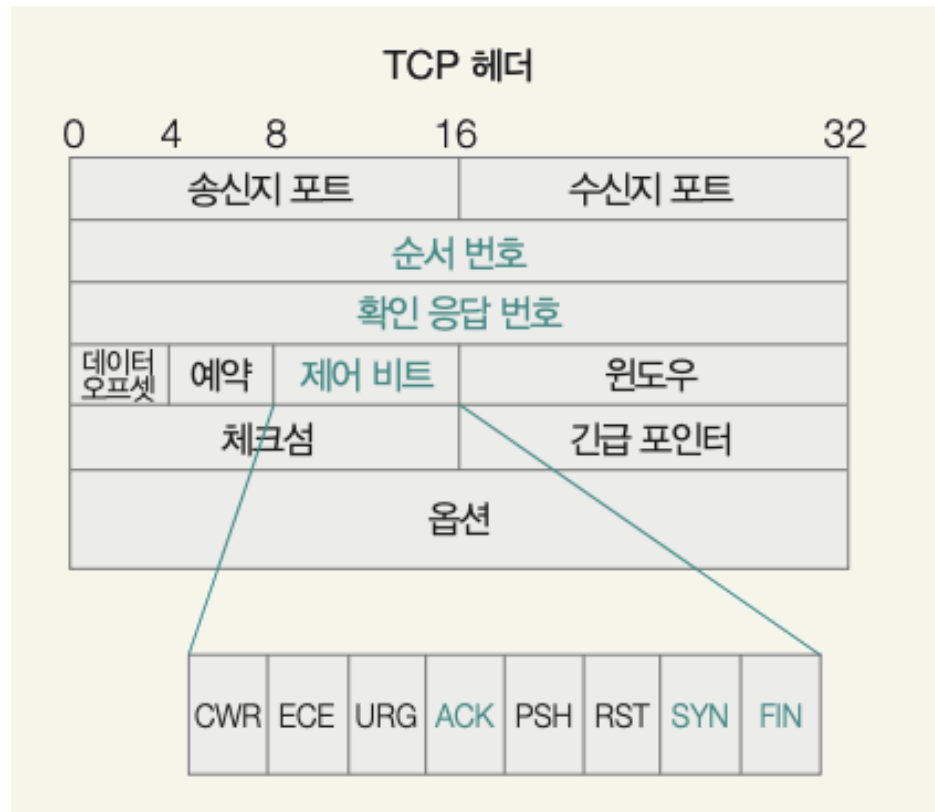
- 전송 계층 = 호스트와 호스트 종단 간 통신
 - IP 한계 보완
 - 응용 계층 프로세스 식별



포트를 통한 프로세스 식별

- IP 주소: 호스트 식별
- 포트 번호: 애플리케이션 프로세스 식별
- 192.168.0.15:8000

포트를 통한 프로세스 식별



포트 번호의 세 가지 유형

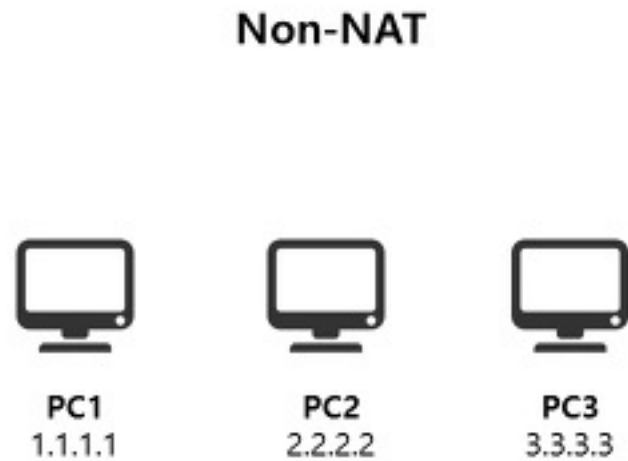
유형	범위	설명
Well-known 포트	0 ~ 1023	유명하고 범용적인 프로토콜이 사용하는 고정 포트
Registered 포트	1024 ~ 49151	특정 기업, 소프트웨어에서 사용하는 등록 포트
Dynamic/Private 포트	49152 ~ 65535	클라이언트에서 임시로 사용하는 포트 (자유롭게 사용 가능)

프로토콜	포트 번호
HTTP	80
HTTPS	443
FTP	20, 21
SSH	22
DNS	53

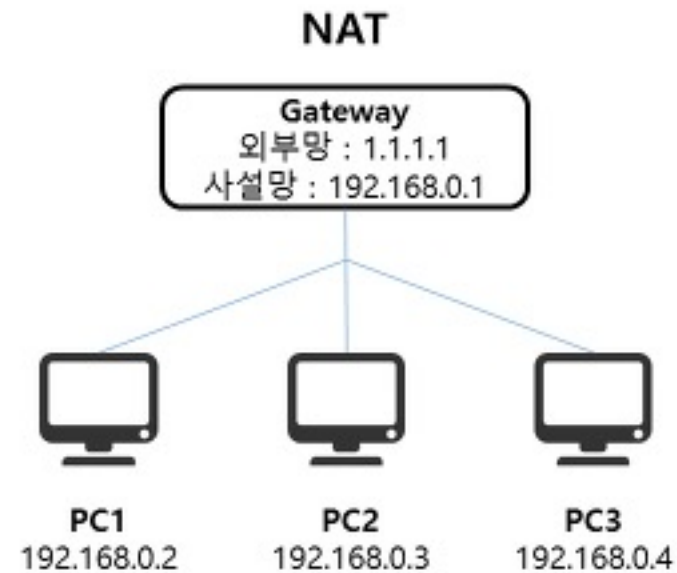
프로그램	포트 번호
MySQL	3306
Redis	6379
Microsoft SQL Server	1433
OpenVPN	1194
HTTP 대체 포트	8080

NAT

- 공인 IP 주소와 사설 IP 주소 간 변환을 위해 사용하는 기술



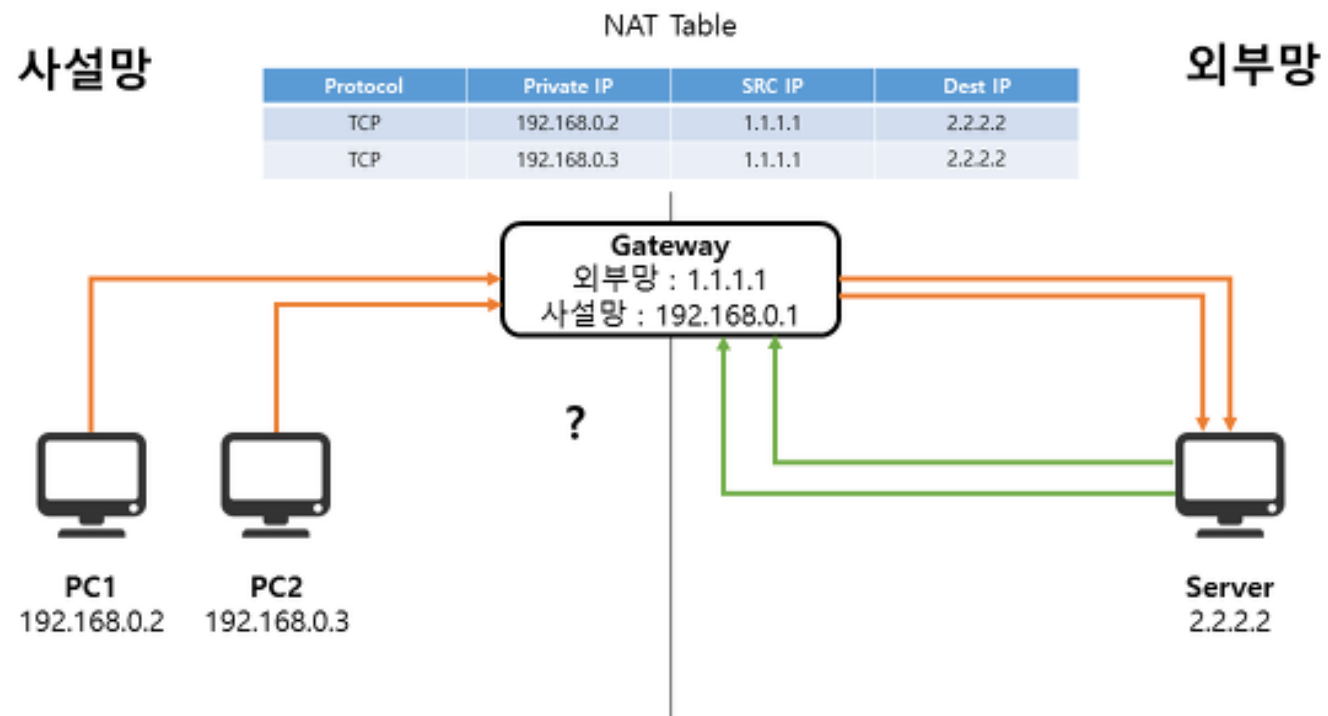
각 기기마다 공인IP 사용



게이트웨이만 공인IP 사용

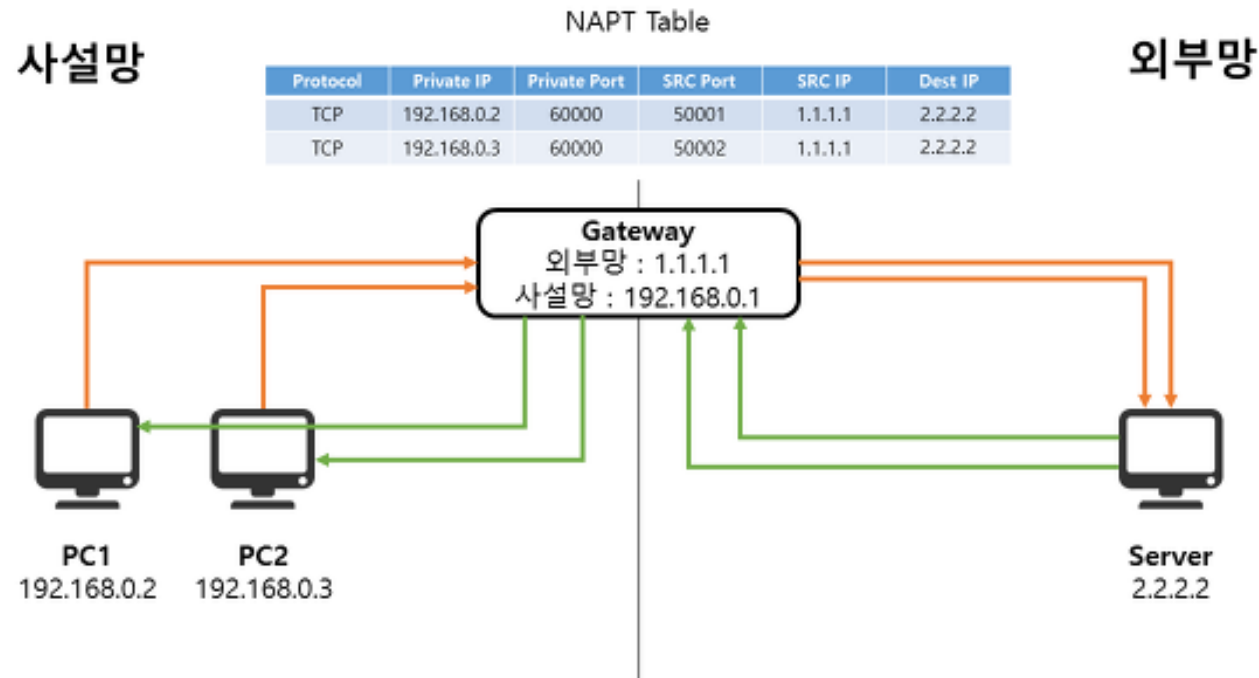
NAPT

- 문제: 여러 개의 사설망 내의 기기가 동시에 요청을 보내면?



NAPT

- 문제: 여러 개의 사설망 내의 기기가 동시에 요청을 보내면?

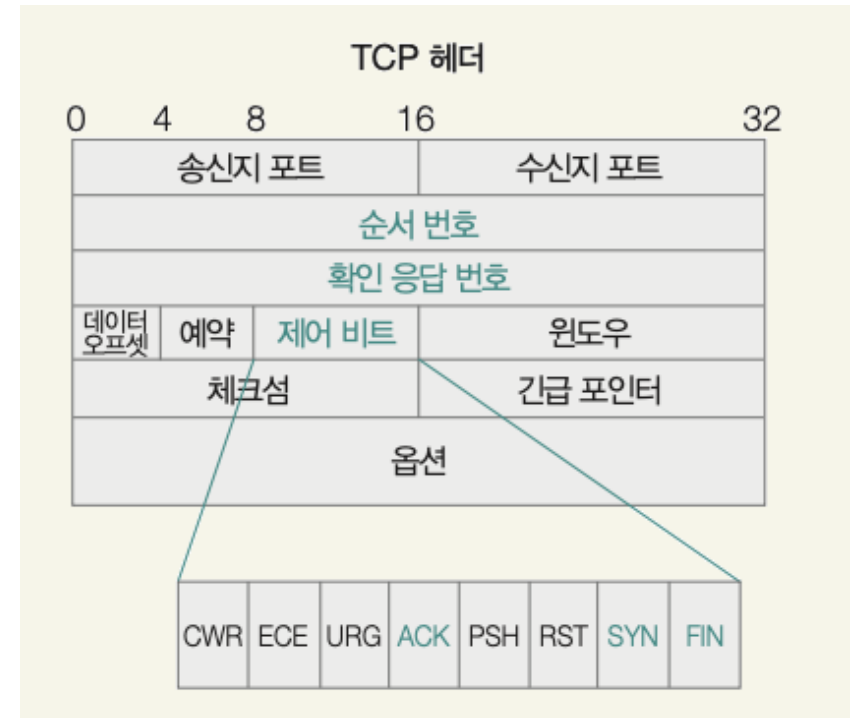


TCP vs UDP

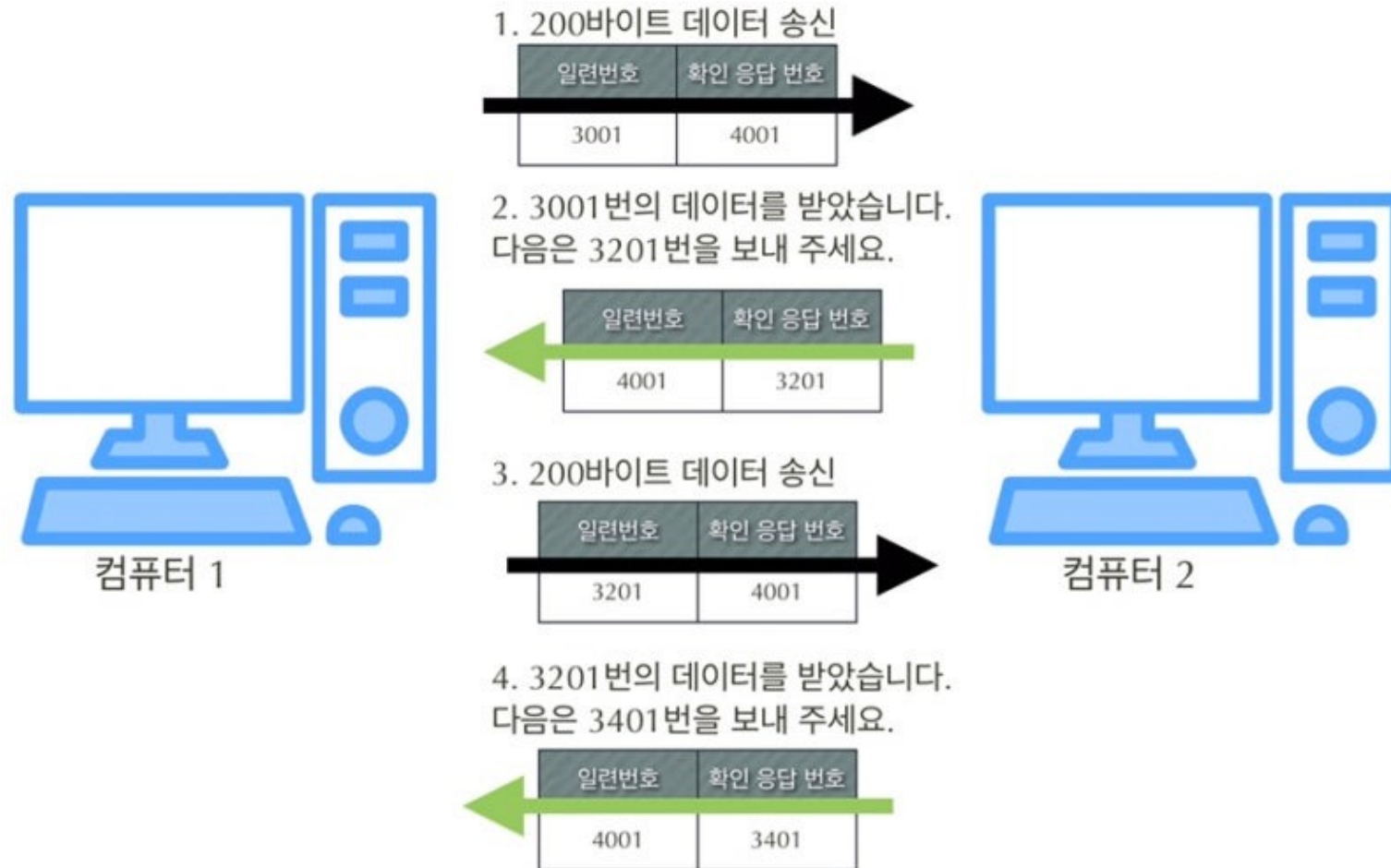
프로토콜	특징
TCP (Transmission Control Protocol)	연결 지향적, 신뢰성 보장, 순서 보장, 느리지만 안전함
UDP (User Datagram Protocol)	비연결형, 빠르지만 신뢰성 없음, 실시간 서비스에 적합

TCP 세그먼트 구조

- 순서 번호: 세그먼트 첫 바이트에 매겨진 번호
- 확인 응답 번호: 다음으로 수신하길 기대하는 순서 번호
- 제어 비트 (ACK, SYN, FIN)



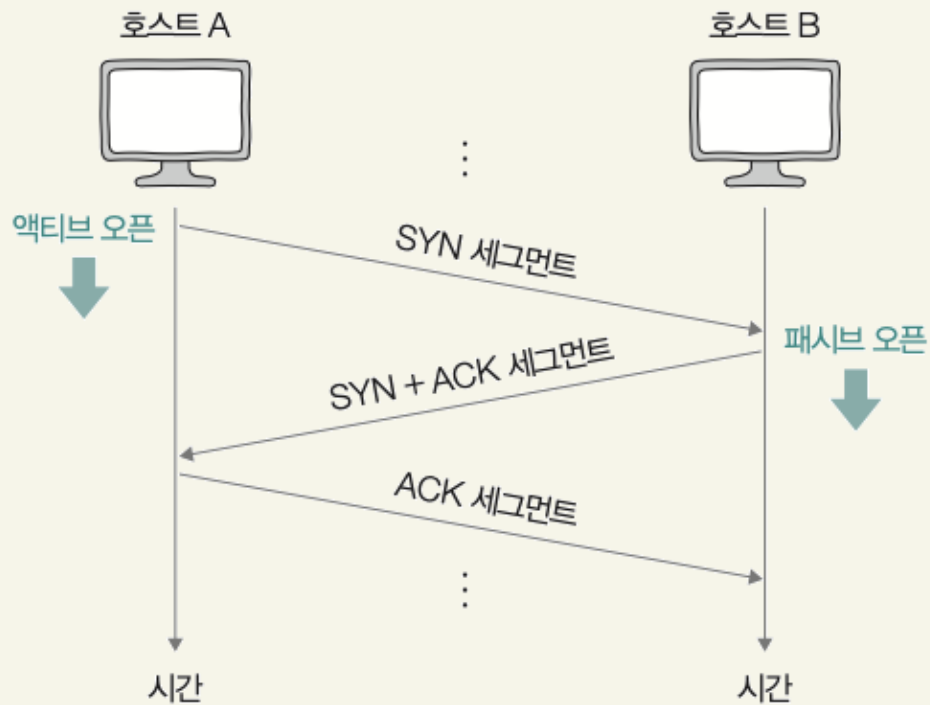
TCP 통신 흐름



TCP 연결부터 종료까지

1. 연결 수립 (3-way Handshake)
2. 데이터 송수신
3. 연결 종료 (4-way Handshake)

연결 수립: 쓰리 웨이 핸드셰이크



송수신 방향	세그먼트 종류	세그먼트에 포함된 주요 정보	비유
A → B	SYN 세그먼트	- 호스트 A의 초기 순서 번호 - 1로 설정된 SYN 비트	"연결 시작합니다."
B → A	SYN + ACK 세그먼트	- 호스트 B의 초기 순서 번호 - 호스트 A가 전송한 세그먼트에 대한 확인 응답 번호 - 1로 설정된 SYN 비트 - 1로 설정된 ACK 비트	"네, 확인했습니다. 연결 시작해요!"
A → B	ACK 세그먼트	- 호스트 A의 다음 순서 번호 - 호스트 B가 전송한 세그먼트에 대한 확인 응답 번호 - 1로 설정된 ACK 비트	"네, 확인했습니다."

데이터 송수신

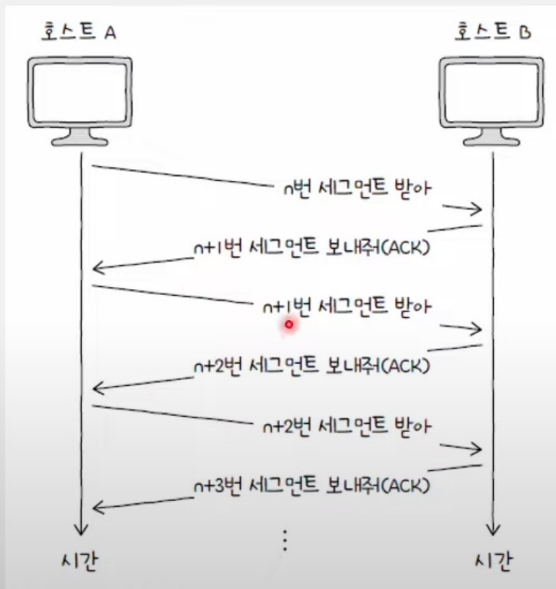
1. 재전송을 통한 오류 제어
2. 흐름 제어
3. 혼잡 제어

1. 재전송을 통한 오류 제어

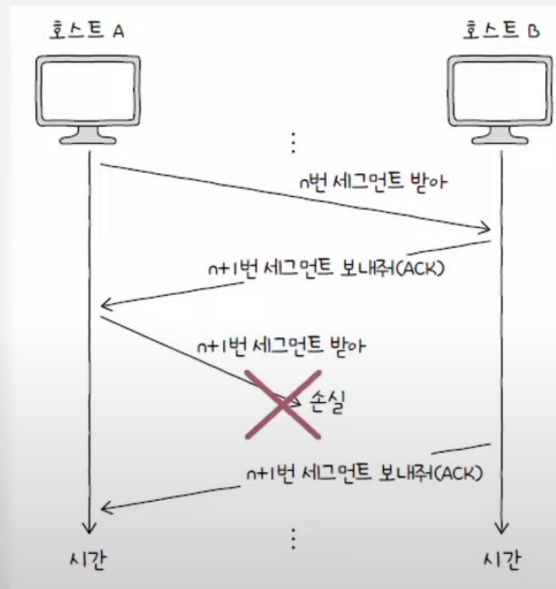
• TCP는 '언제 잘못 전송된 세그먼트가 있음'을 인지할까?

1. 중복된 ACK 세그먼트가 도착

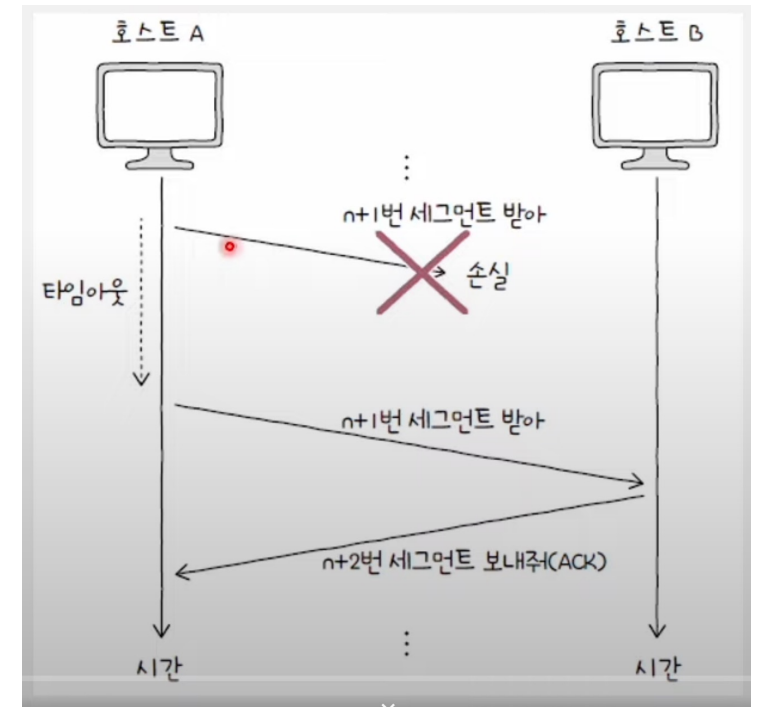
2. 타임아웃 발생



송수신이 올바르게 이루어진 경우

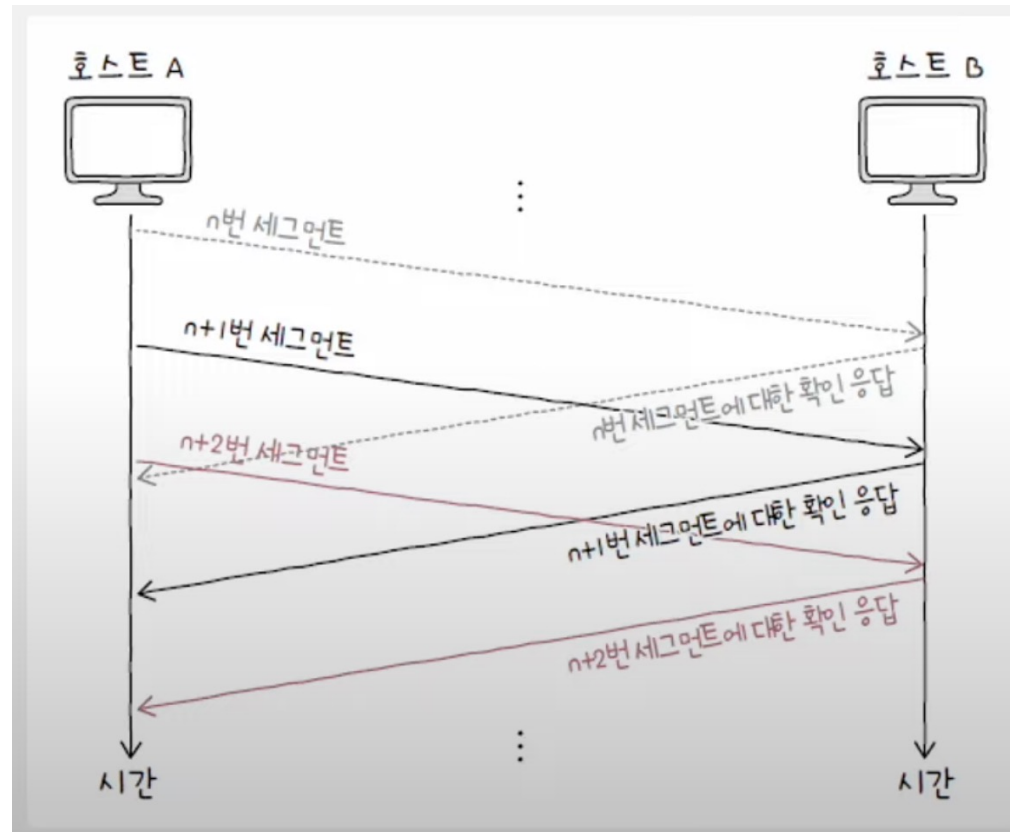


수신 호스트 측이 받은 세그먼트의 순서 번호
중에서 일부가 누락된 경우



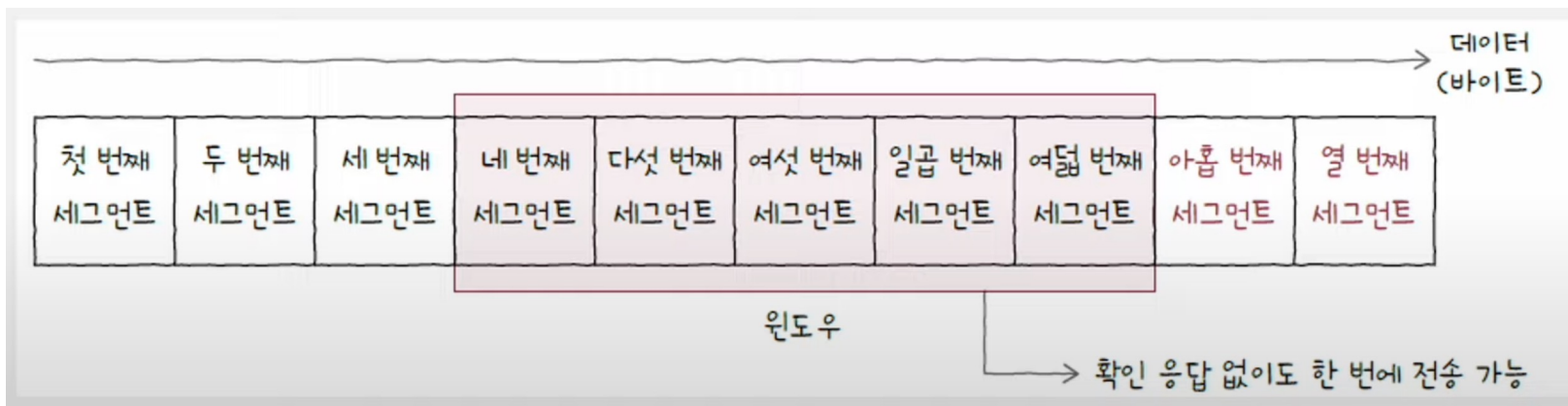
파일프라이닝 전송

- 사실 한 번에 송신하고 한 번에 각 세그먼트에 대한 확인 응답을 받는다.



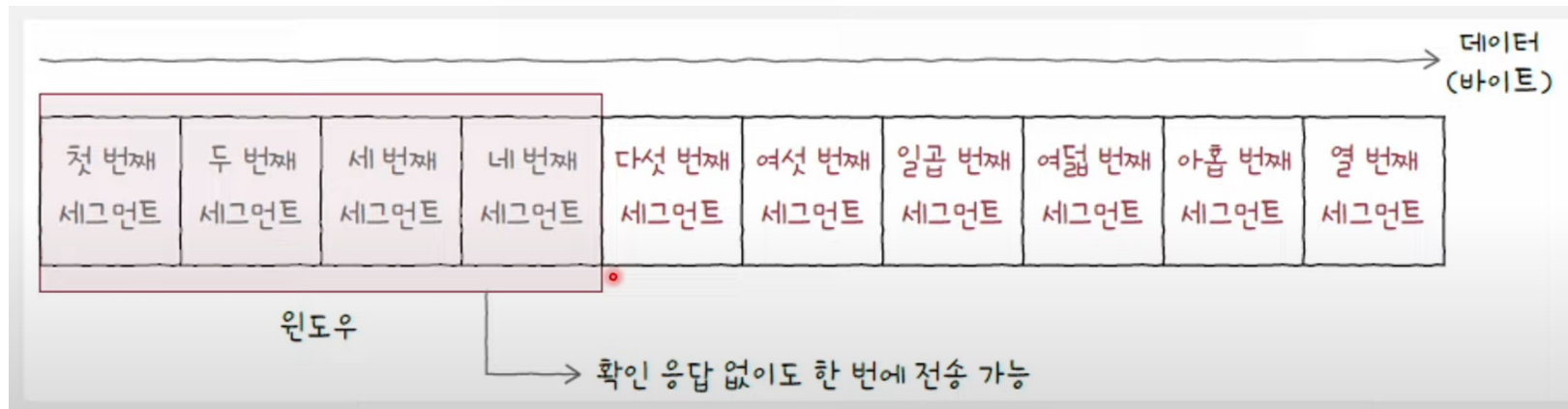
2. 흐름 제어

- 슬라이딩 윈도우: TCP 흐름 제어 기법
 - 윈도우: 송신 호스트가 파이프라이닝할 수 있는 최대량
즉, 윈도우의 크기만큼 확인 응답을 받지 않고도 한 번에 전송 가능하다는 의미



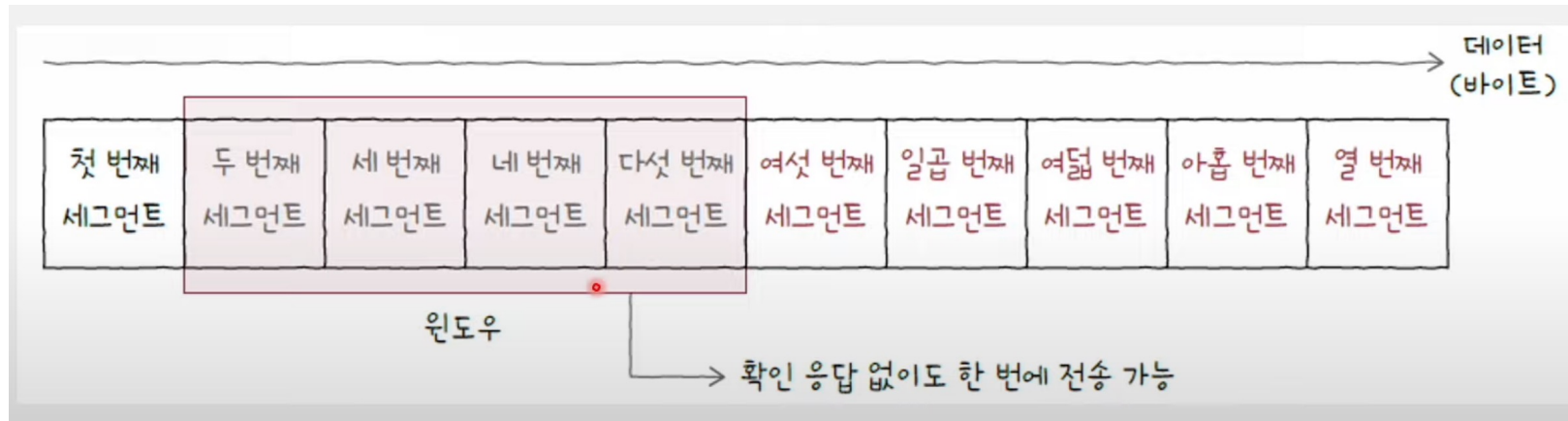
2. 흐름 제어

- 슬라이딩 윈도우: TCP 흐름 제어 기법



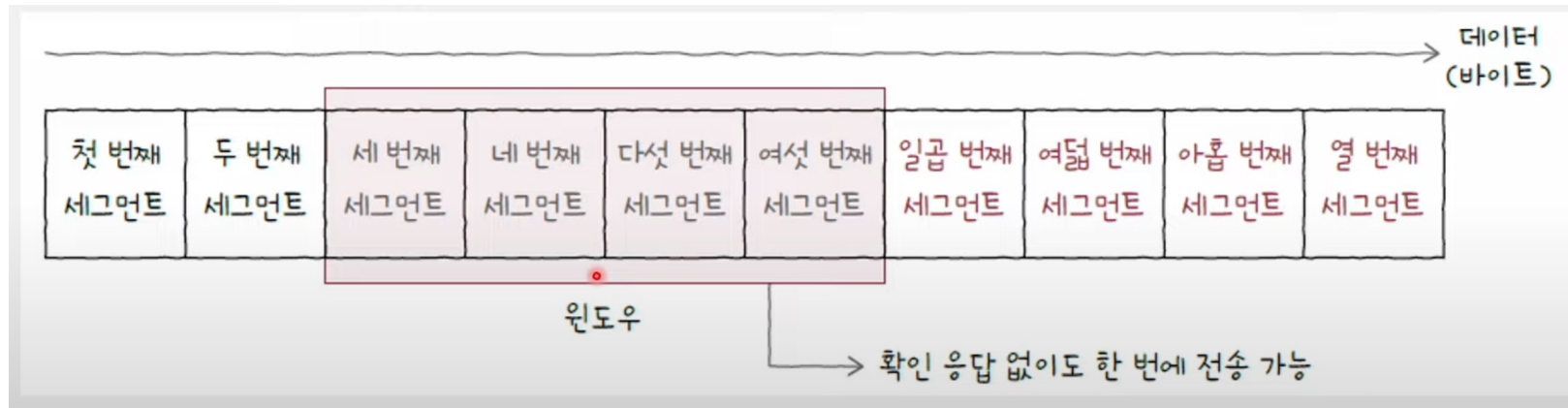
2. 흐름 제어

- 슬라이딩 윈도우: TCP 흐름 제어 기법



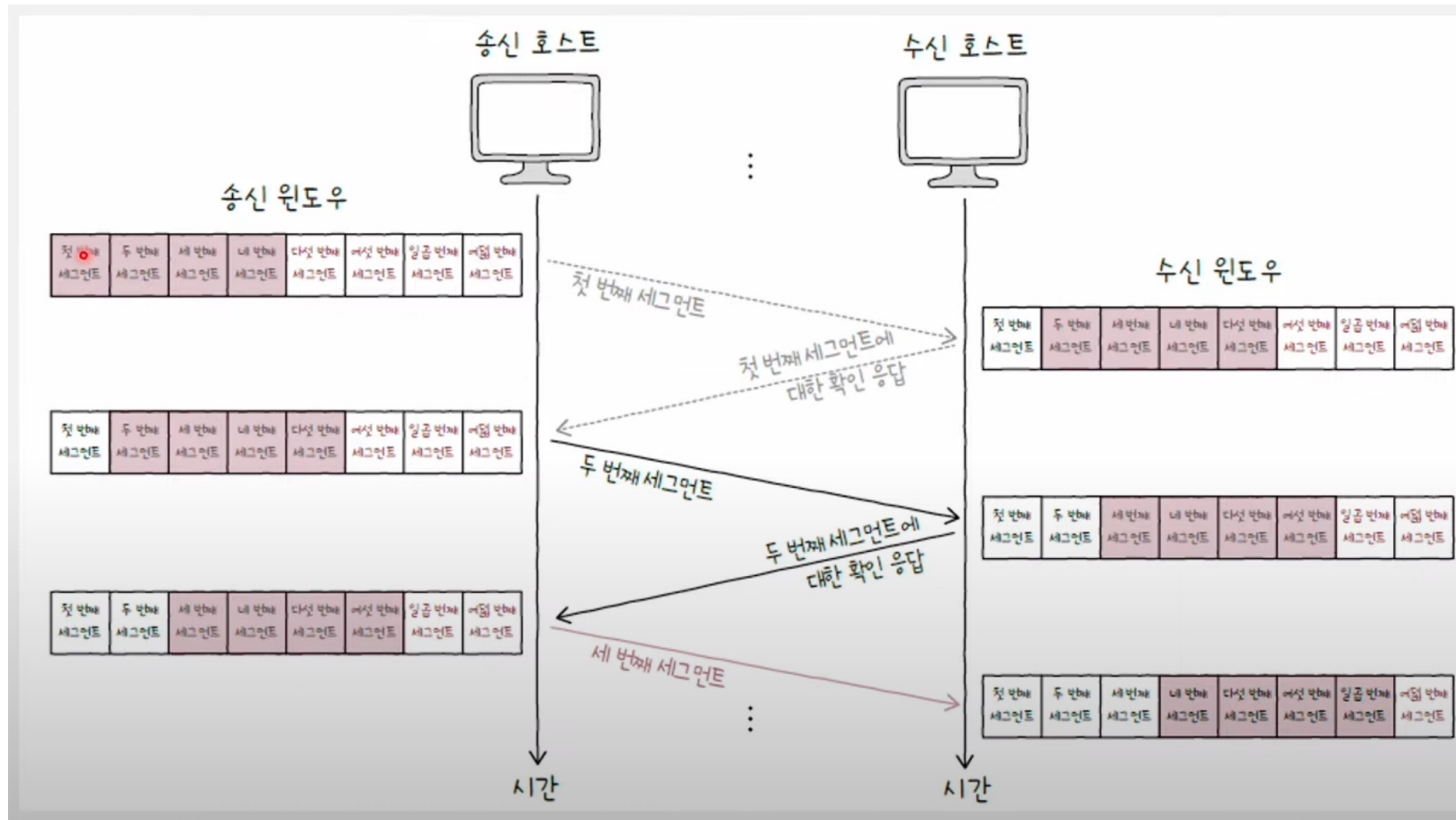
2. 흐름 제어

- 슬라이딩 윈도우: TCP 흐름 제어 기법



2. 흐름 제어

- 슬라이딩 윈도우: TCP 흐름 제어 기법



3. 혼잡 제어

- 혼잡: 많은 트래픽으로 인해 패킷의 처리 속도가 늦어지거나 유실될 우려가 있는 네트워크 상황
- 흐름 제어의 주체가 수신 호스트라면 혼잡 제어의 주체는 송신 호스트

3. 혼잡 제어

- 혼잡 윈도우: 혼잡 없이 전송할 수 있을 법한 데이터양
 - 수신 윈도우는 수신 호스트가 헤더로 알려줌
 - 혼잡 윈도우는 송신 호스트가 알아서 직접 계산 해야함

3. 혼잡 제어

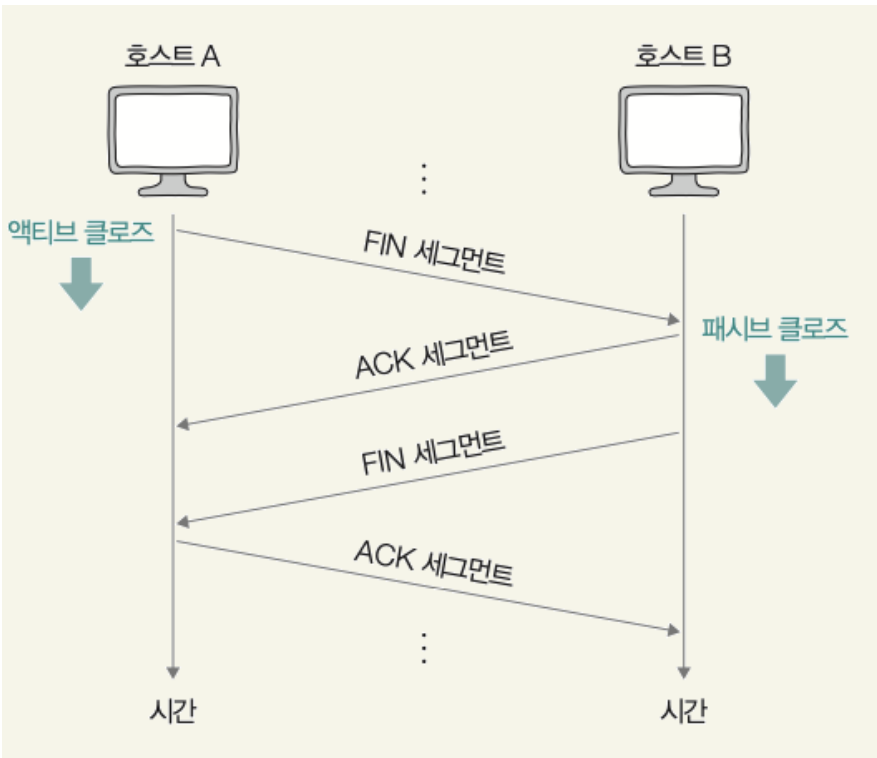
- TCP 혼잡 제어 알고리즘: AIMD (Additive Increase/Multiplicative Decrease)
 - '합으로 증가, 곱으로 감소'
 - 혼잡이 감지되지 않는다면 - 혼잡 윈도우를 RTT(Round Trip Time)마다 1씩 선형적으로 증가
 - 혼잡이 감지되면 - 혼잡 윈도우를 절반으로 떨어뜨리는 동작을 반복



3. 혼잡 제어

- TCP는 혼잡 기준이 뭔데? = 오류 제어랑 동일!
 - 중복된 ACK 세그먼트를 수신했을 때
 - 타임아웃이 발생했을 때

연결 종료: 포 웨이 핸드셰이크



송수신 방향	세그먼트 종류	세그먼트에 포함된 주요 정보	비유
A → B	FIN 세그먼트	- 1로 설정된 FIN 비트	"연결 끊을게요."
B → A	ACK 세그먼트	- 호스트 A가 전송한 세그먼트에 대한 확인 응답 번호 - 1로 설정된 ACK 비트	"네, 확인했습니다."
B → A	FIN 세그먼트	- 1로 설정된 FIN 비트	"이제 연결 끊어요."
A → B	ACK 세그먼트	- 호스트 B가 전송한 세그먼트에 대한 확인 응답 번호 - 1로 설정된 ACK 비트	"네, 확인했습니다."

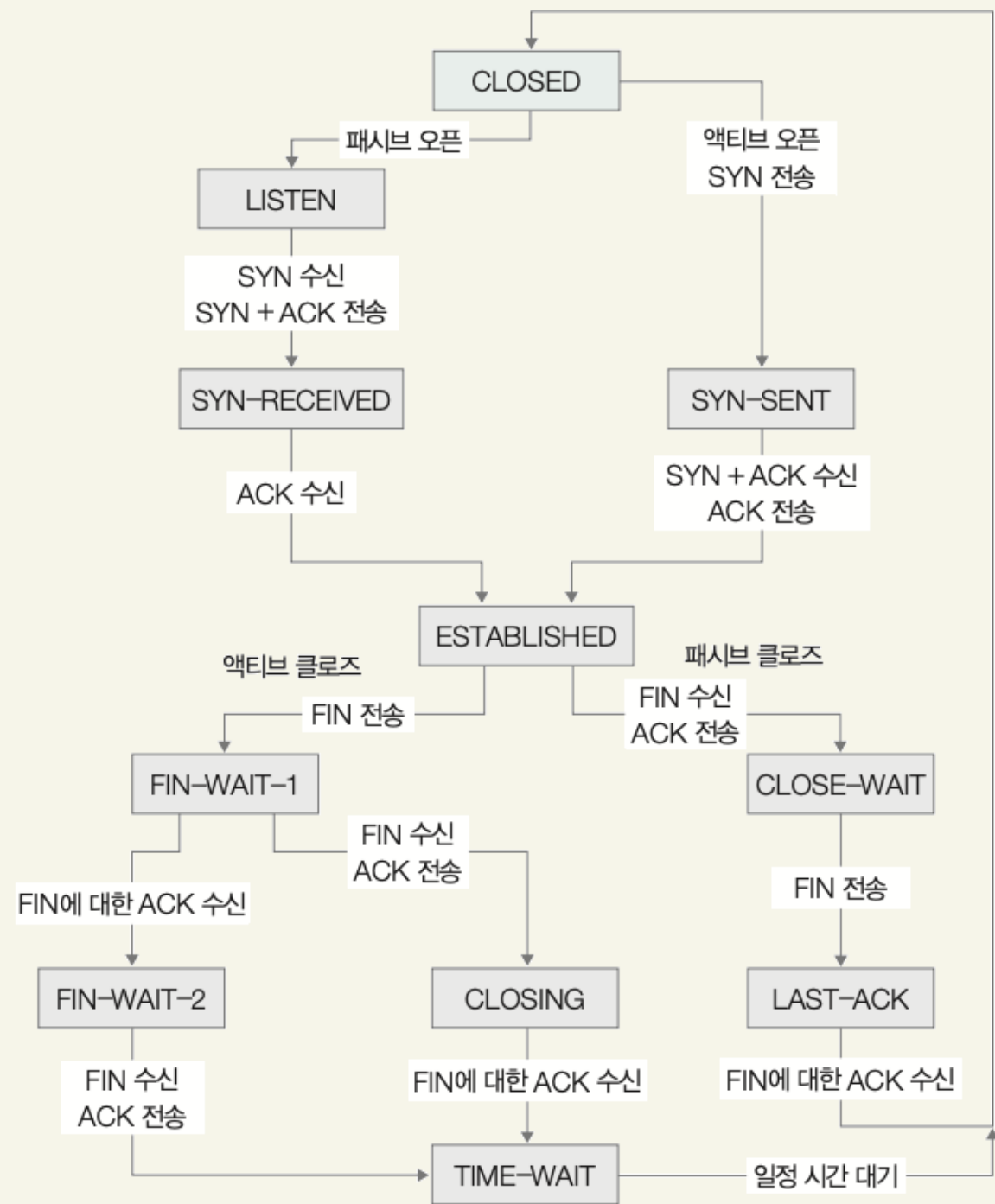
TCP의 상태 관리

- TCP는 스테이트풀 프로토콜(stateful protocol)
- 상태(state) = 현재 어떤 통신 과정에 있는지?

```
? netstat -a -p tcp
```

```
Active Internet connections (including servers)
```

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	(state)
tcp4	0	0	192.168.0.14.49965	kix07s06-in-f10..https	ESTABLISHED
tcp4	0	0	192.168.0.14.49964	49.50.167.165.http-alt	SYN_SENT
tcp4	0	0	192.168.0.14.49962	ec2-52-196-128-1.https	ESTABLISHED
tcp4	0	0	192.168.0.14.49953	ec2-43-206-164-1.https	ESTABLISHED
tcp4	0	0	192.168.0.14.49948	server-18-67-51-.https	ESTABLISHED
tcp4	0	0	192.168.0.14.49946	207.164.110.34.b.https	ESTABLISHED

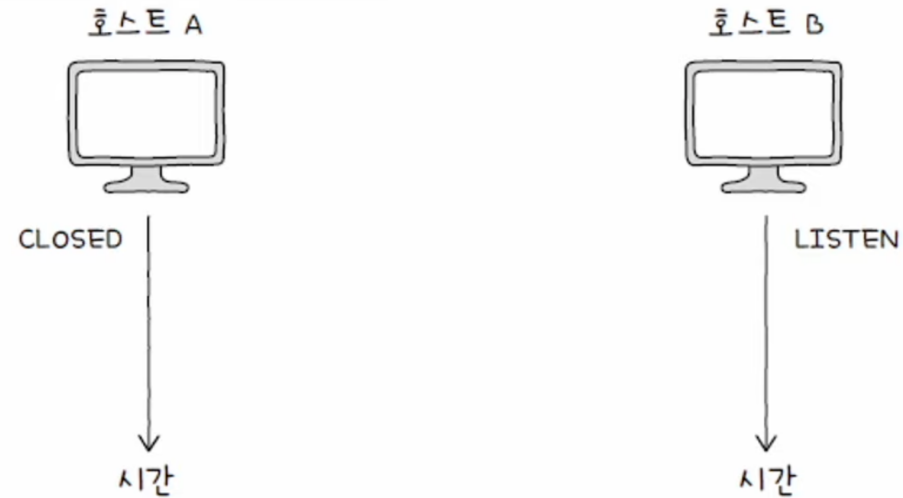


TCP의 상태 관리

1. 연결이 수립되지 않았을 때 주로 활용되는 상태
2. 연결 수립 과정에서 주로 활용되는 상태
3. 연결 종료 과정에서 주로 활용되는 상태

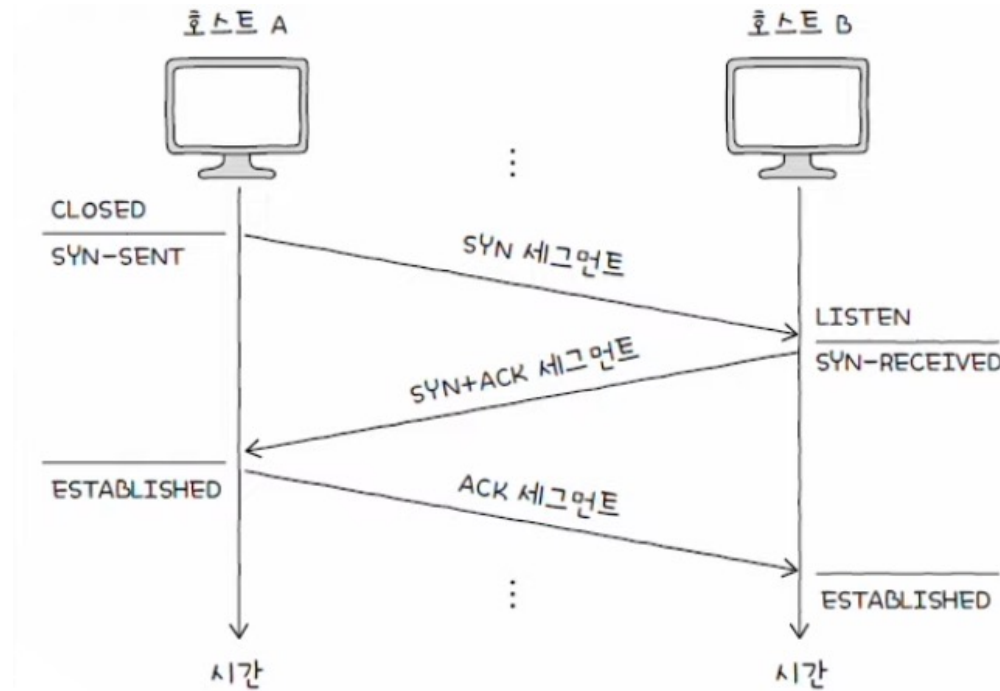
상태 분류	주요 상태
1	CLOSED, LISTEN
2	SYN-SENT, SYN-RECEIVED, ESTABLISHED
3	FIN-WAIT-1, CLOSE-WAIT, FIN-WAIT-2, LAST-ACK, TIME-WAIT, CLOSING

1. 연결이 수립되지 않았을 때



상태	설명
CLOSED	아무런 연결이 없는 상태
LISTEN	일종의 연결 대기 상태(SYN 세그먼트를 기다리는 상태)

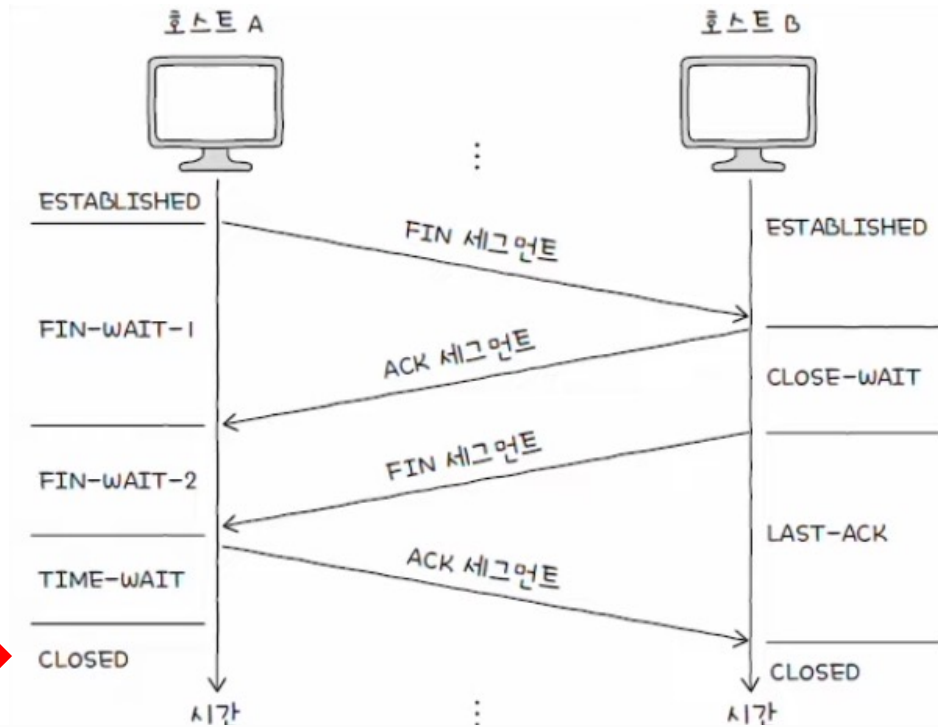
2. 연결 수립 과정에서 주로 활용될 때



상태	설명
SYN-SENT	액티브 오픈 호스트가 SYN 세그먼트를 보낸 뒤, 그에 대한 응답인 SYN + ACK 세그먼트를 기다리는 상태(연결 요청 전송)
SYN-RECEIVED	패시브 오픈 호스트가 SYN + ACK 세그먼트를 보낸 뒤, 그에 대한 ACK 세그먼트를 기다리는 상태(연결 요청 수신)
ESTABLISHED	쓰리 웨이 핸드셰이크가 끝난 뒤 데이터를 송수신할 수 있는 상태(연결 수립)

3. 연결 종료 과정에서 주로 활용될 때

액티브 클로즈 호스트는 일정 시간을 기다린 뒤 CLOSED 상태가 된다. 일정 시간을 대기하는 이유는 마지막 ACK 세그먼트가 올바르게 전송되지 않았을 수 있고, 이 경우 재전송이 필요하기 때문이다.



상태	설명
FIN-WAIT-1	액티브 클로즈 호스트가 FIN 세그먼트로 연결 종료 요청을 보낸 상태(연결 종료 요청 전송)
CLOSE-WAIT	FIN 세그먼트를 받은 패시브 클로즈 호스트가 그에 대한 응답으로 ACK 세그먼트를 보낸 후 대기하는 상태(연결 종료 요청 승인)
FIN-WAIT-2	FIN-WAIT-1 상태에서 ACK 세그먼트를 받은 상태
LAST-ACK	CLOSE-WAIT 상태에서 FIN 세그먼트를 전송한 뒤 대기하는 상태
TIME-WAIT	액티브 클로즈 호스트가 마지막 ACK 세그먼트를 전송한 뒤 접어드는 상태