

# 인터넷 네트워크

- 인터넷 통신
- IP(Internet Protocol)
- TCP, UDP
- PORT
- DNS

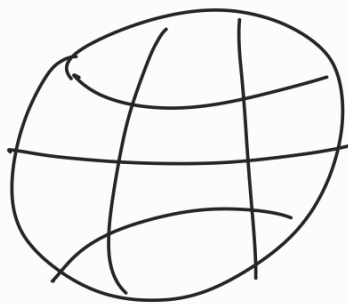
---

## 인터넷 통신

- 인터넷에서 두 컴퓨터는  
어떻게 통신할까?



client

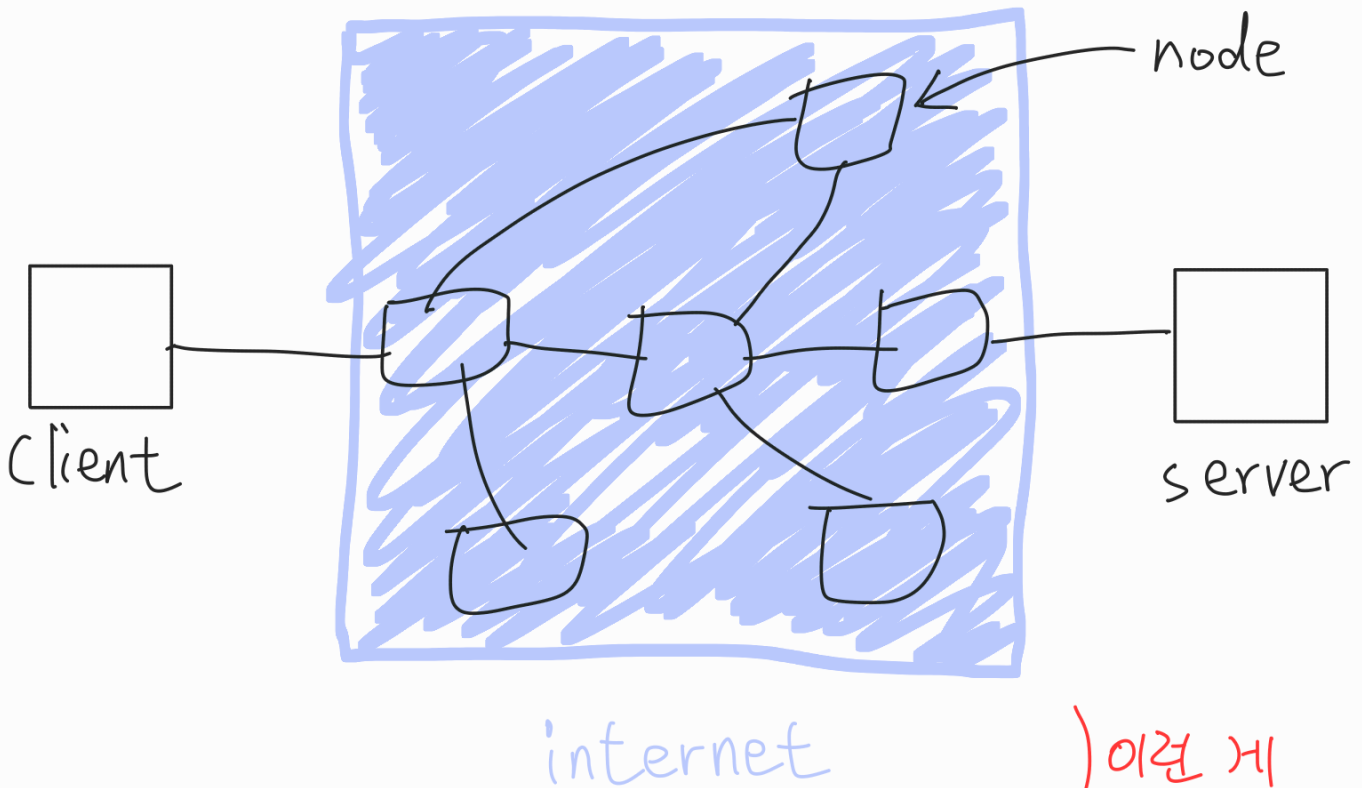


internet



server

· 복잡한 인터넷 망



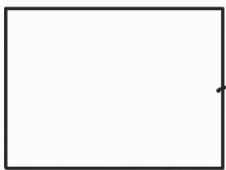
이런 게  
어떻게 흘러가는지  
알기 위해선  
먼저 이것부터!

IP(Internet Protocol)

복잡한 인터넷 망에서는  
“최소한의 규칙”이 있어야 할 것임.

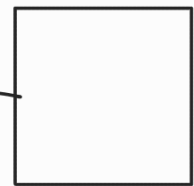
⇒ IP 주소

IP: 100.100.100.1



Client

IP: 200.200.200.2

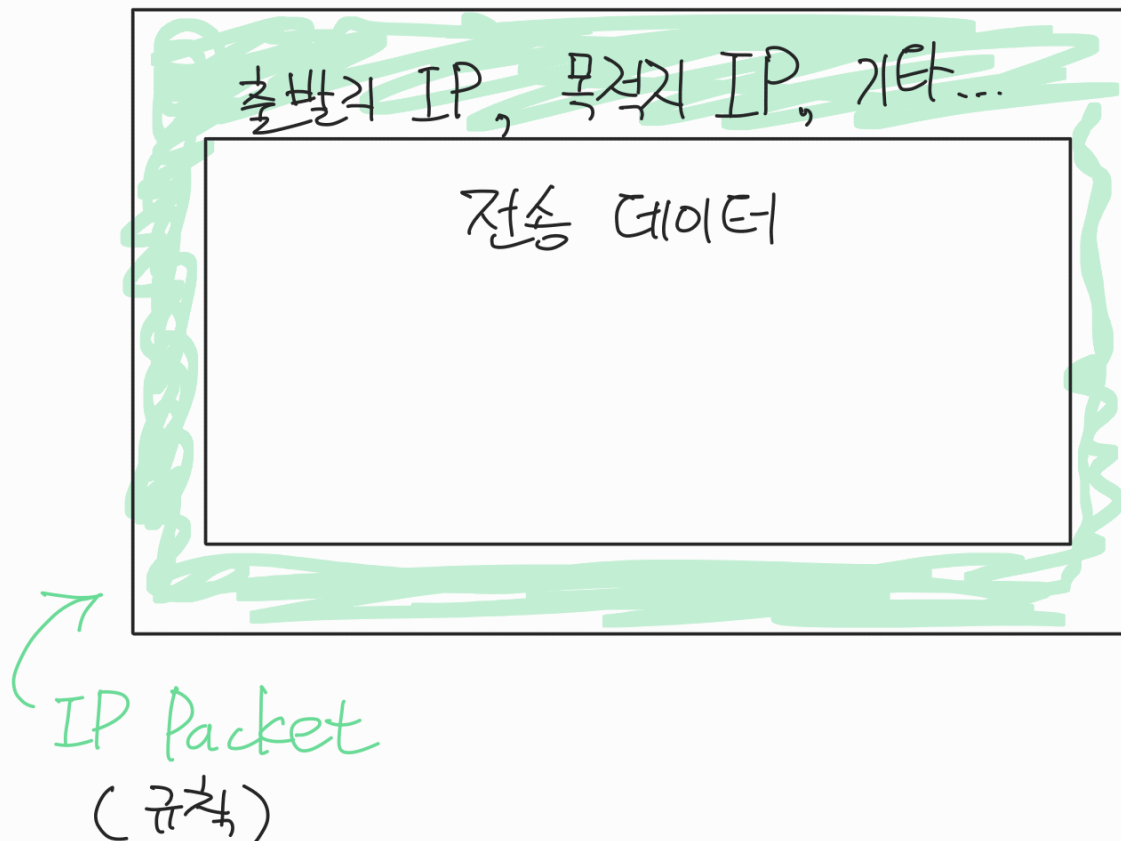


Server

## • IP의 역할

- 지정된 IP address 에 데이터 전달
- 패킷(Packet)이라는 통신 단위로 데이터 전달

## • IP 패킷 정보



이렇게 만들어서

client packet 을 전달함

그 뒤에

Server packet 전달

## • IP 프로토콜의 한계

### - 비연결성

: packet 받을 대상 없거나  
서비스 X 상태여도

packet 전송

### - 비신뢰성

① 중간에 packet 안 오면?

② packet이 순서대로 안 오면?

~~packet~~

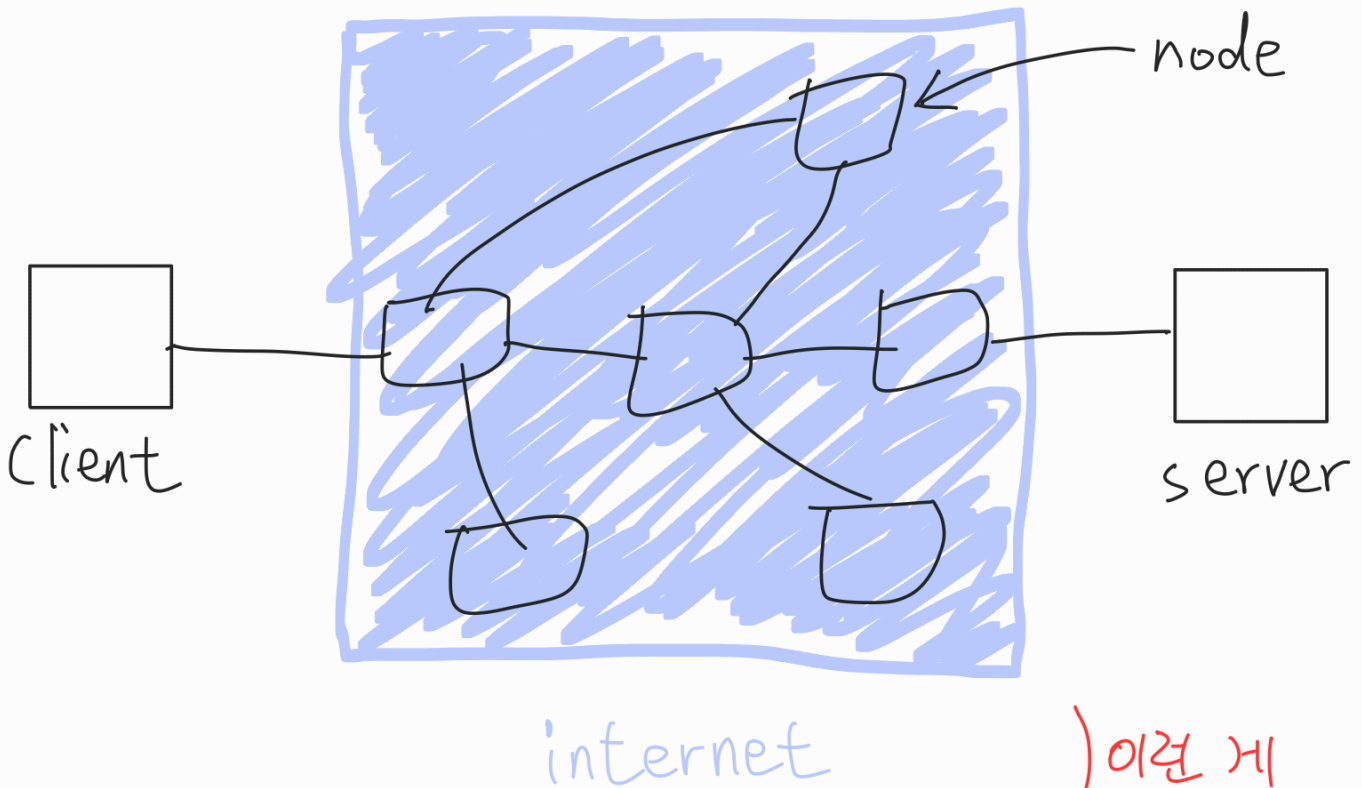
무실 가능성

### - 프로그램 구분

: 같은 IP 사용 server에서 동작하는

app-이 둘 ↑ 이면?

· 복잡한 인터넷 망



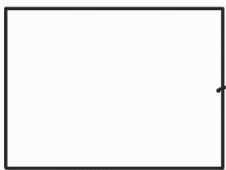
이런 게  
어떻게 흘러가는지  
알기 위해선  
먼저 이것부터!

IP(Internet Protocol)

복잡한 인터넷 망에서는  
“최소한의 규칙”이 있어야 할 것임.

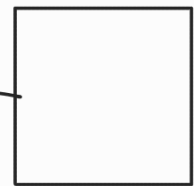
⇒ IP 주소

IP: 100.100.100.1



Client

IP: 200.200.200.2

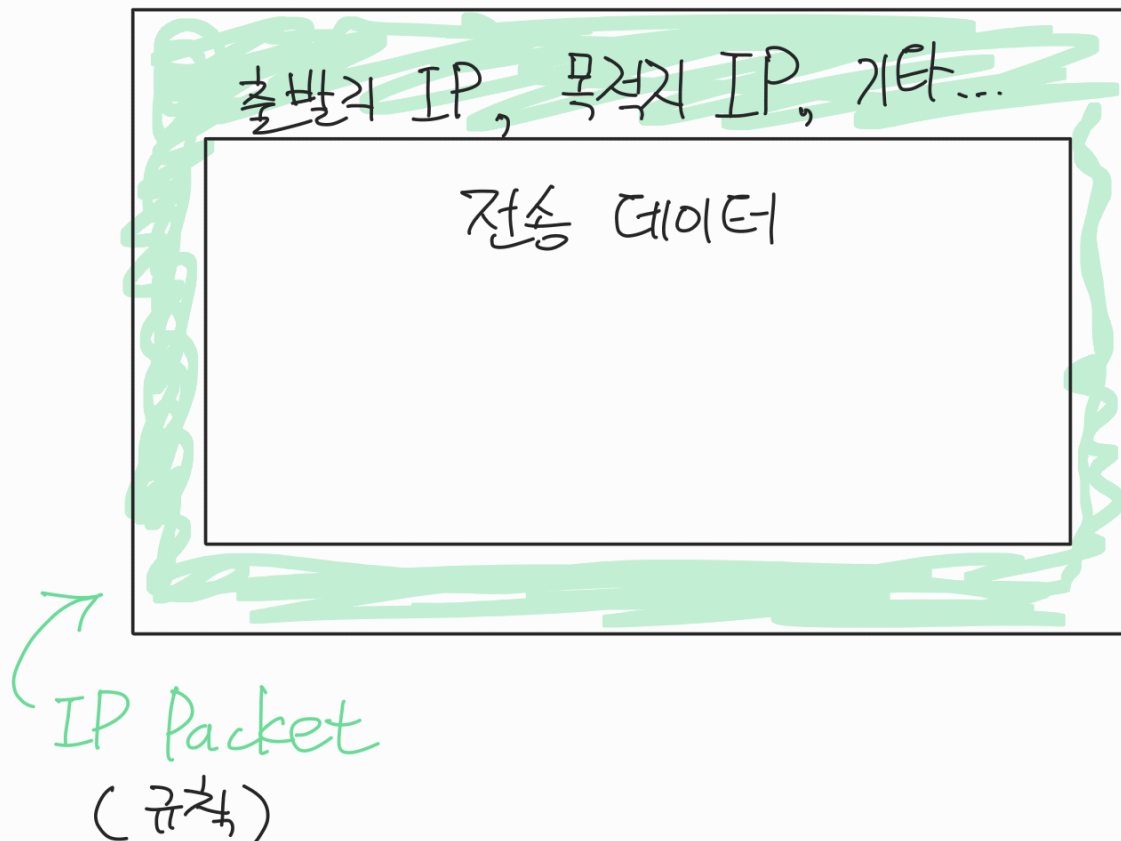


Server

## • IP의 역할

- 지정된 IP address 에 데이터 전달
- 패킷(Packet)이라는 통신 단위로 데이터 전달

## • IP 패킷 정보



이렇게 만들어서

client packet 을 전달함

그 뒤에

Server packet 전달

## • IP 프로토콜의 한계

### - 비연결성

: packet 받을 대상 없거나  
서비스 X 상태여도

packet 전송

### - 비신뢰성

① 중간에 packet 안 오면?

② packet이 순서대로 안 오면?

~~packet~~

무실 가능성

### - 프로그램 구분

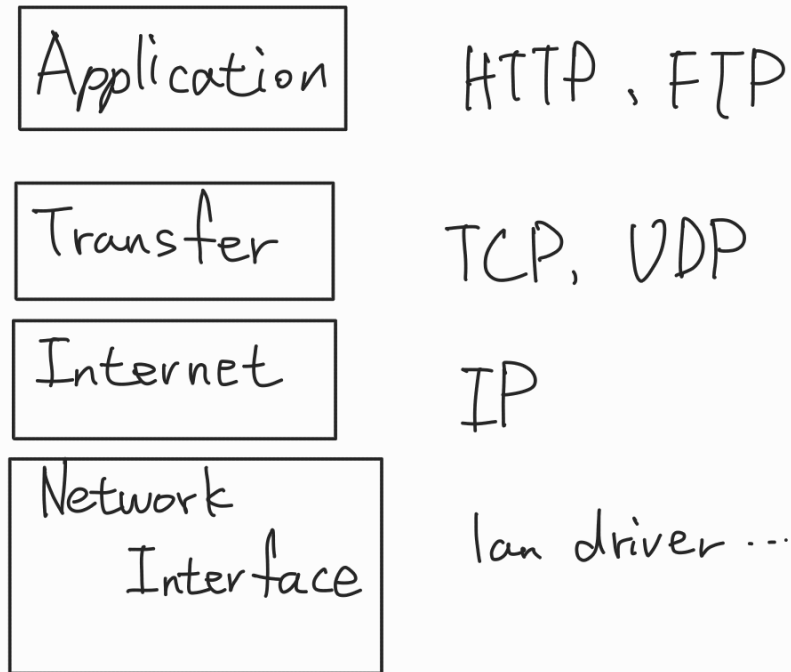
: 같은 IP 사용 server에서 동작하는

app-이 둘 ↑ 이면?

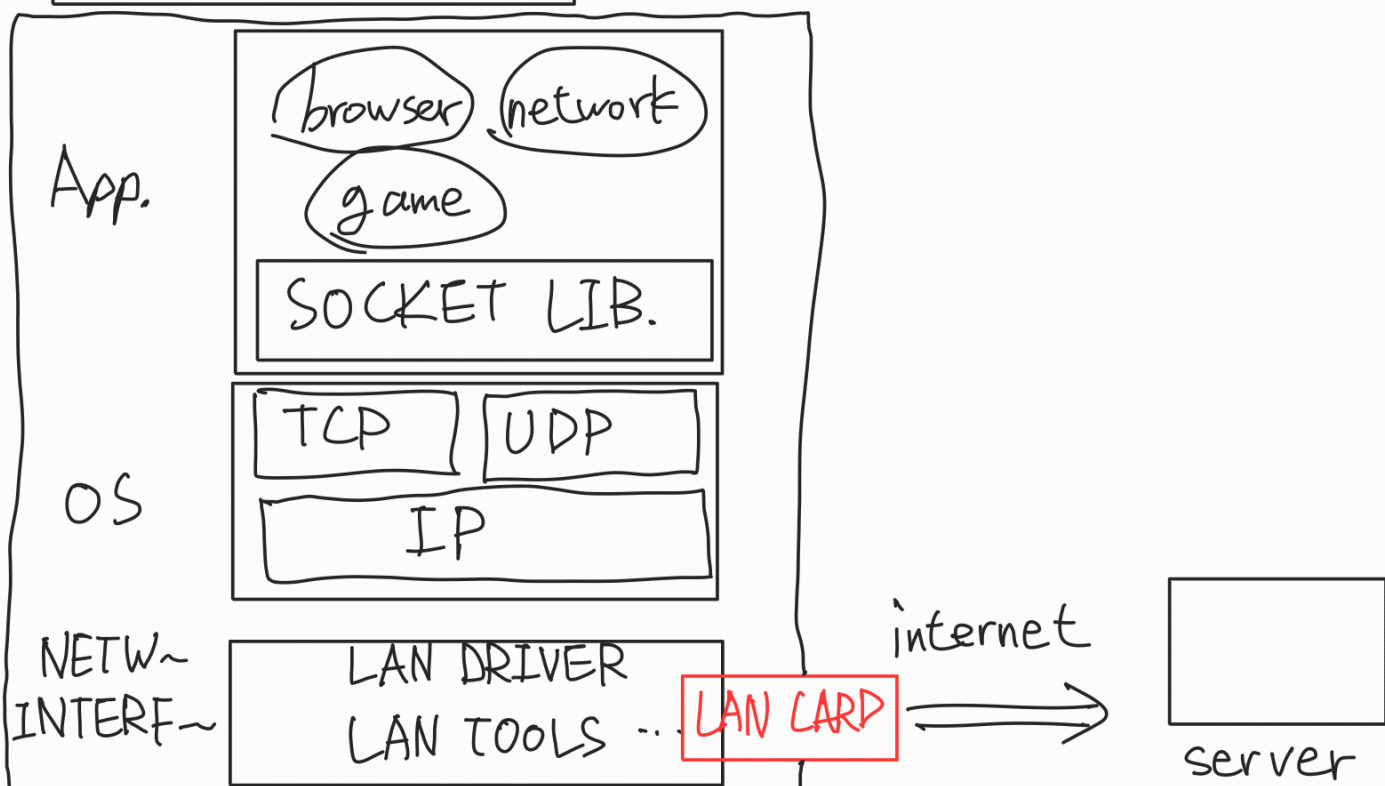
# TCP and UDP

↳ TCP가 이전의 "IP의 한계" 부분을 해결해 줌

## Internet Protocol Stack의 4 Layers

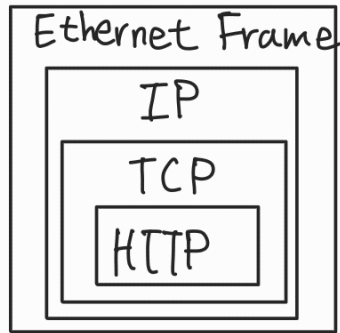


## Protocol Layer

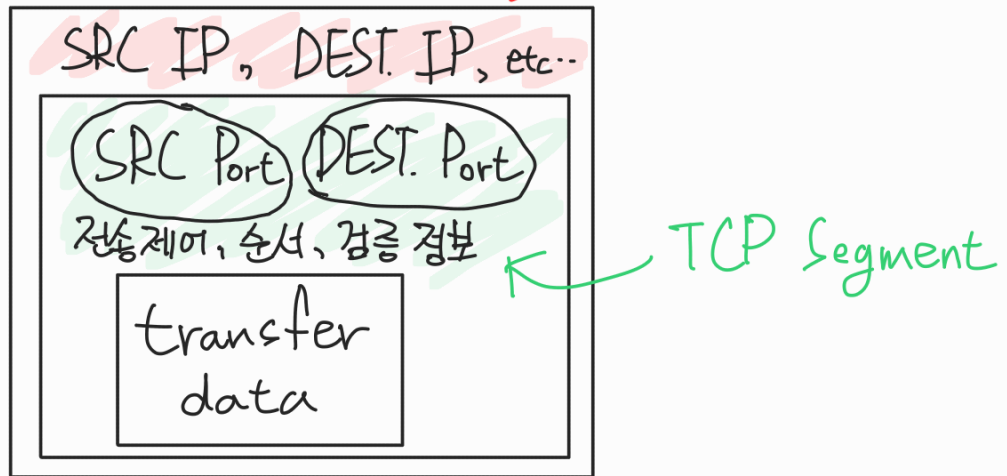




cf)



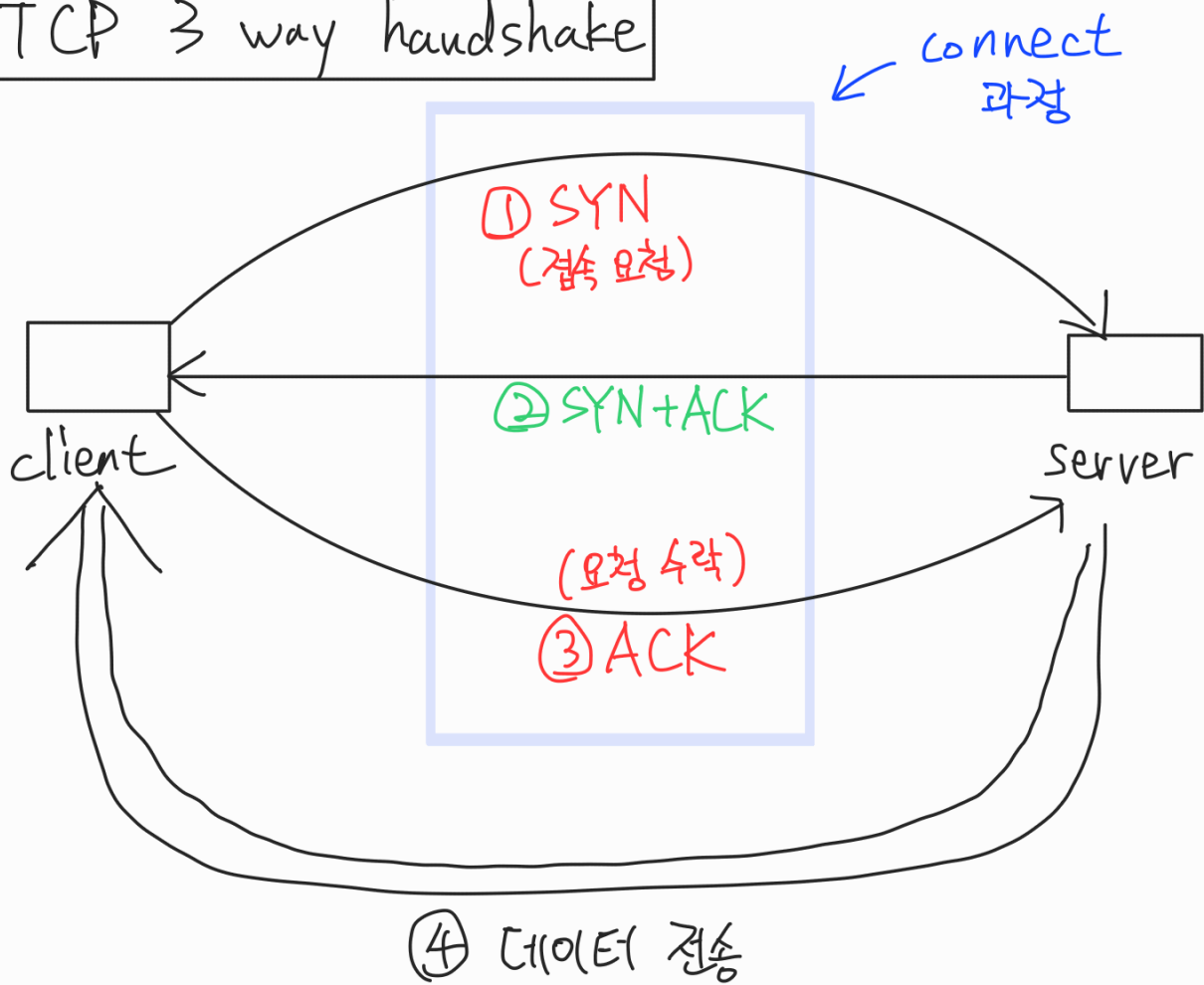
## TCP/IP 패킷 정보



## TCP 특징 (Transmission Control Protocol)

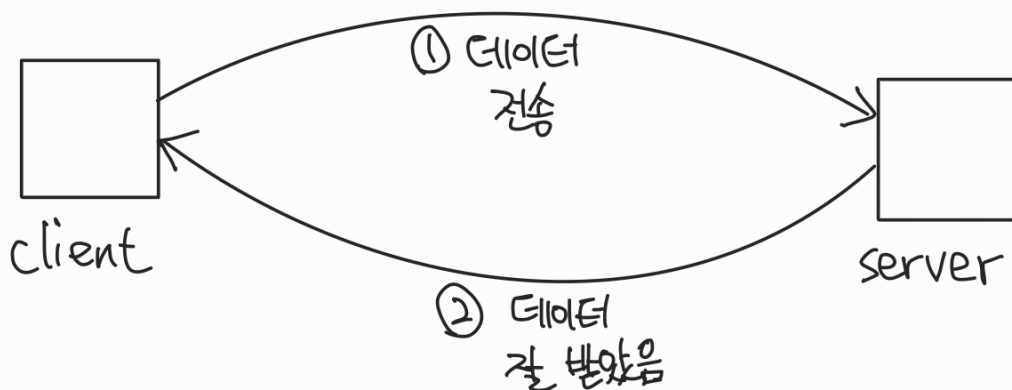
- 연결 지향 (Connection-Oriented)
  - TCP 3-way handshake (가상 연결)
  - ↳ 전송 전에 일단 연결 여부부터!
- 데이터 전달 보증 (receiver: either POS or NEG, sender: has bright CLUE)
- 순서 보장 (Ensures ~ same order)
- 신뢰할 수 있는 프로토콜 (Reliable)
- 현재는 대부분 TCP 사용

# TCP 3 way handshake



★ 이건 "논리적인 측면에서의 연결"임  
(물리적으로까지는 X)

## 데이터 전달 보증



## 순서 보장

packet 순서 잘못되면  
잘못된 것부터 다시 전송!

## UDP 특징 (User Datagram Protocol)

- 하얀 도화지에 비유 (기능 거의 X)
- 연결 지향 X
- 데이터 전달 보증 X
- 순서 보장 X
- But, 단순, 빠름
- IP와 거의 같음 + PORT + CHECKSUM 등
- App. 에서 추가 작업 필요

TCP에는  
손을 못 대므로  
UDP를 이용하여  
여러 ~

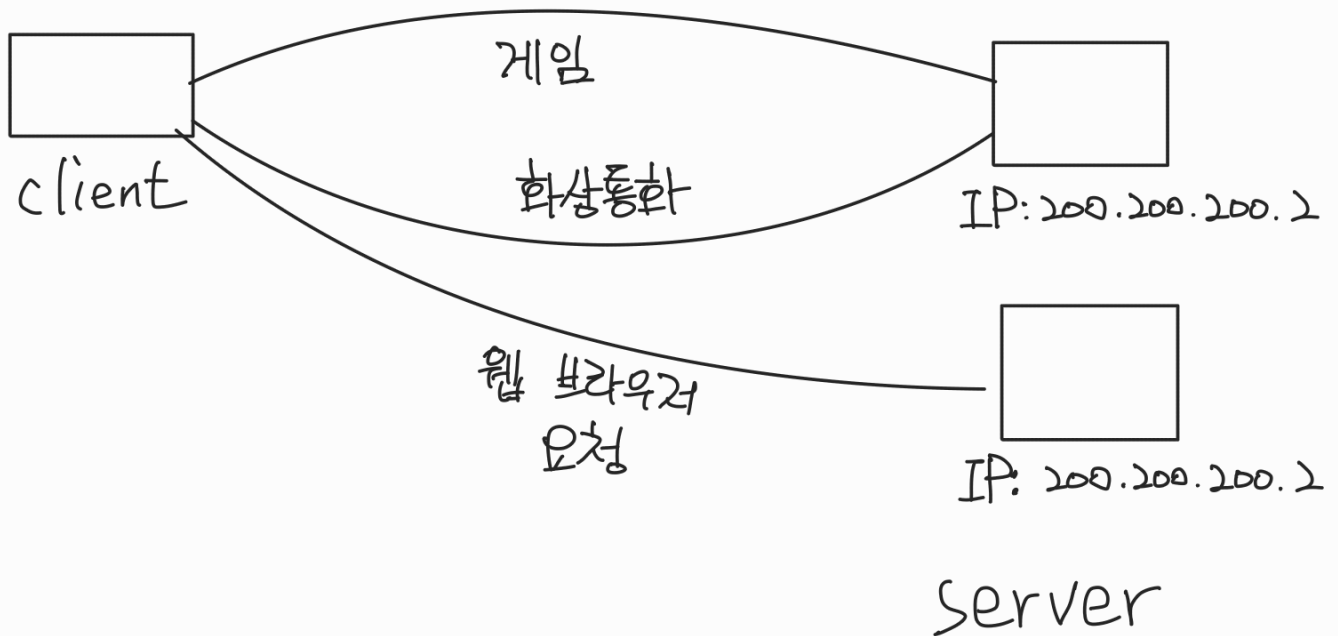
예) HTTP 3에서  
UDP 사용



# PORT

---

한 번에 둘 이상 연결해야 하면?



⇒ PORT

: 같은 IP 내에서  
프로세스 구분

- 0 ~ 65535 할당 가능
- 0 ~ 1023 : 잘 알려진 포트
  - FTP: 20, 21
  - TELNET: 23
  - HTTP: 80
  - HTTPS: 443



# DNS

---

IP는 기억하기 어렵다

⇒ DNS (Domain Name System)

- like 전화번호부
- domain name convert → IP Address