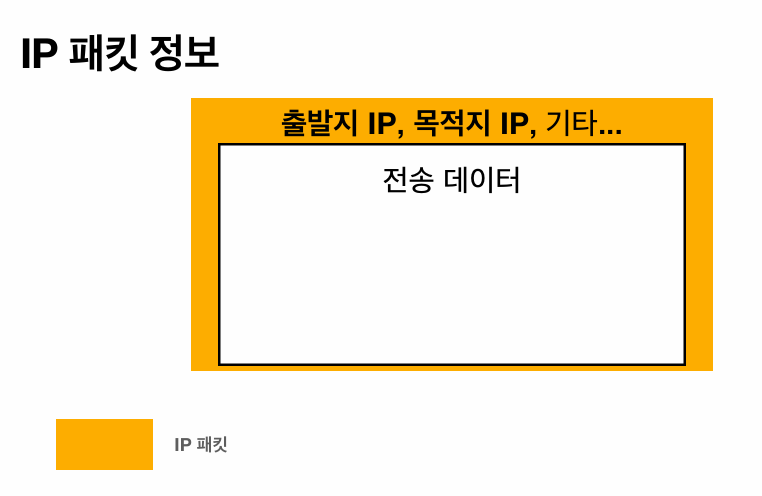


Socket라이브라러를 통해 OS에 입력값을 전달함.

Hello 정보에 TCP 정보 생성해서 넘김.

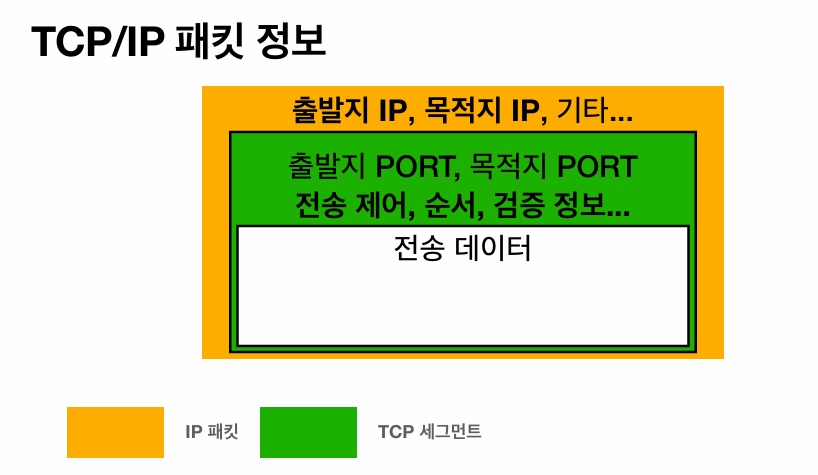
IP 패킷을 추가해서 또 넘김.

LAN카드를 통해 나갈 때 Ethernet frame을 담아서 보냄. (Mac 주소 같은 물리적인 주소 포함)

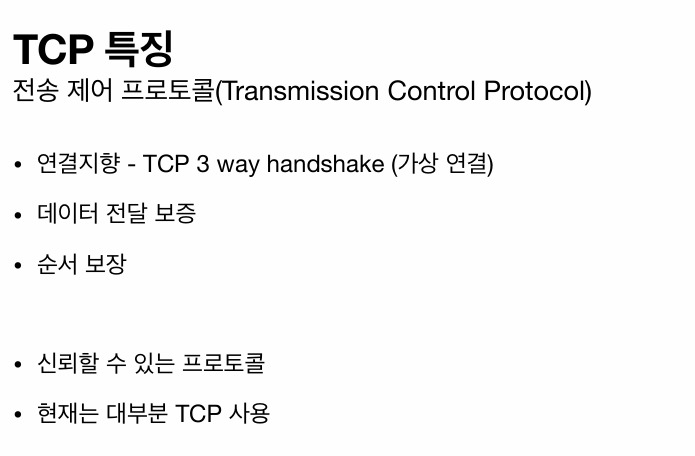


패킷 : 수화물을 뜻하는 패키지와 덩어리를 뜻하는 버킷의 합성어

택배박스에 데이터 넣는 것과 비슷

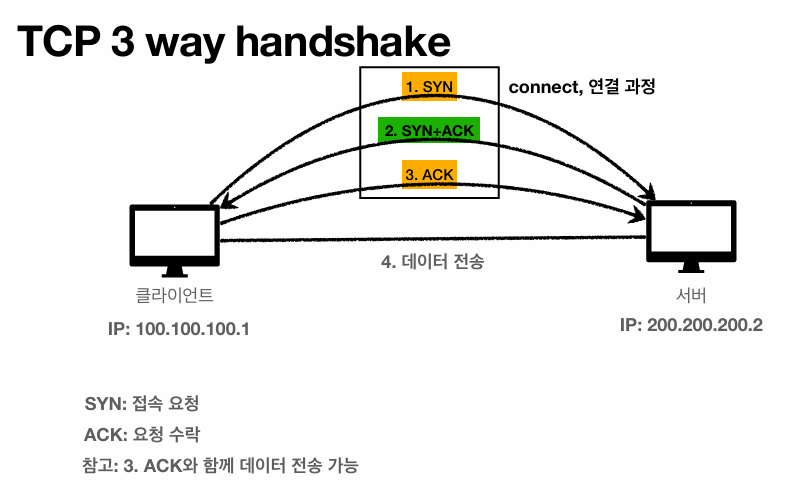


TCP에는 전송 제어 정보, 순서 정보, 검증정보 등이 있어서 신뢰있는 연결 및 데이터 전송이 가능함.



무조건 연결을 하고 메시지를 보내게 됨.

메시지를 보냈는데 패킷이 누락되었으면 누락되었다는 것을 알 수 있음.



클라이언트에서 서버에 SYN이란 메시지를 보냄.

server에서 ACK와 SYN이란 메시지를 보냄.

클라이언트에서 ACK보냄.

SYN : Syncronize

Ack : Acknowledge

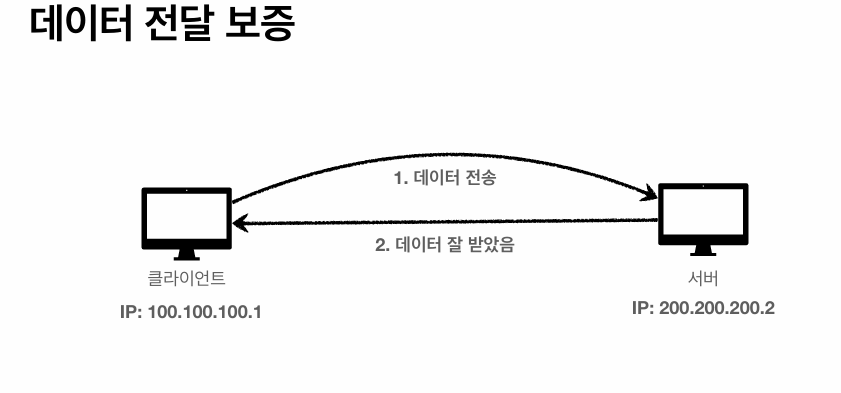
클라이언트도 서버를 믿을 수 있고, 서버도 클라이언트를 믿을 수 있음.

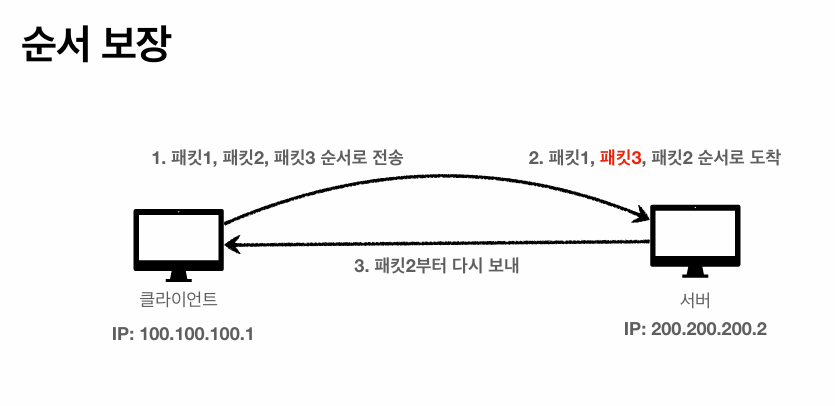
메시지를 3번 전달받으면서 연결이 되었다는 것을 알 수 있음.

서버에서 ACK라는 메시지를 보내지 않으면, 클라이언트에서 연결이 되지 않은 것을 인식하고 데이터를 보내지 않음.

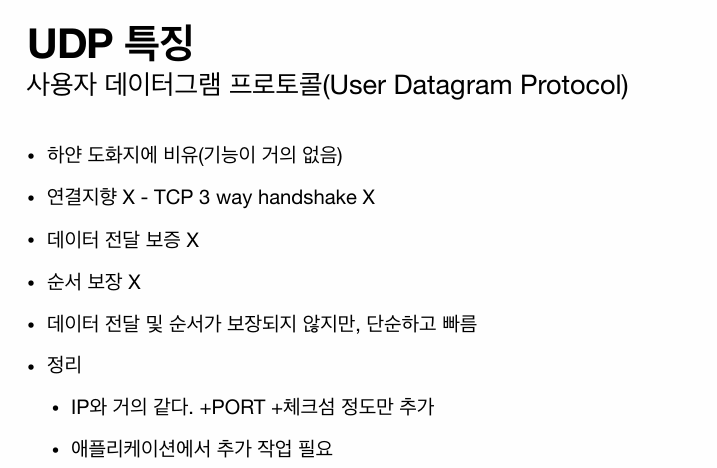
TCP 연결이 되었다는 것은 개념적으로 연결이 되었음을 의미함. 논리적인 연결이 된 것임.

물리적으로 직접 연결된 것은 아님. 중간에 수많은 노드들이 연결을 시켜주지만, 어떤 노드들을 거쳐서 연결되었는지는 모름. 전용 랜선이 연결되는 것이 아님.





서버에서 패킷 순서가 잘못된 것을 알면 잘못 보낸 패킷부터 다시 보내라고 요청함.



메시지가 제대로 맞는지 검증해주는 데이터 정도만 추가되어있음.

UDP를 쓰는 이유?

TCP는 데이터도 추가적으로 더 들어가고, 검증작업이 들어가기 때문에 전송 속도가 느림.

하지만 최적화를 위해서 TCP를 수정할 수는 없음.

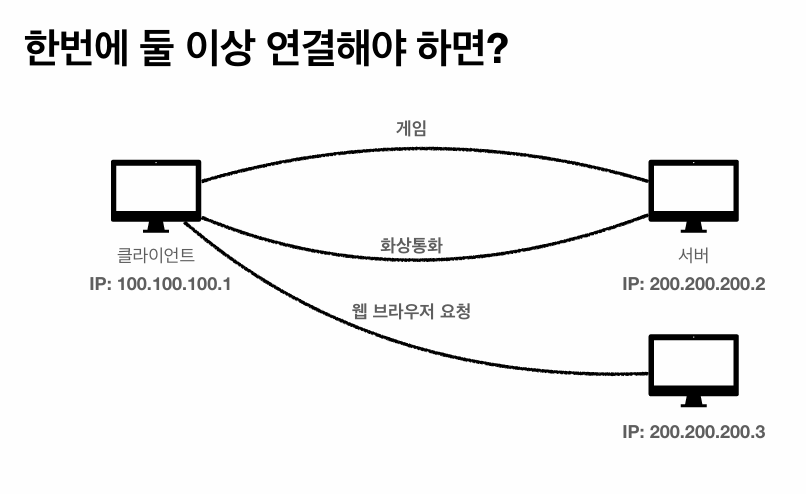
TCP를 그대로 쓰고, UDP를 사용하되, 내가 원하는 대로 수정하면 됨.

TCP를 90퍼센트 쓰고 있는데, 요새 UDP 가 다시 뜨고 있음.

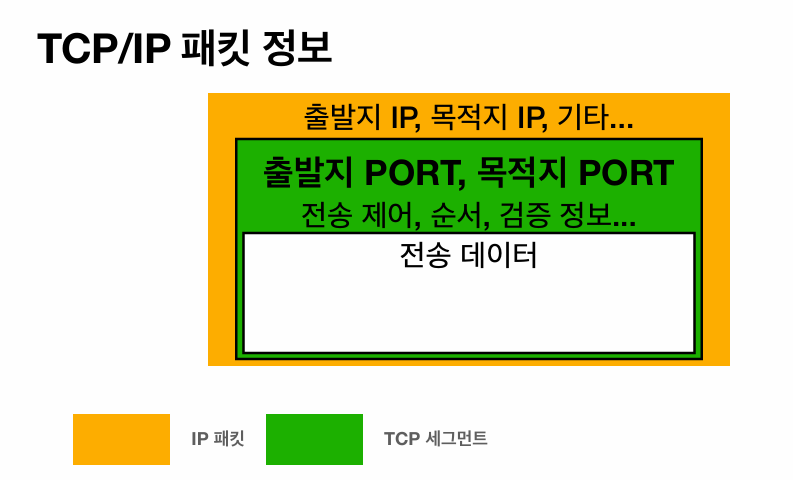
HTTP3에서는 TCP의 3HandShaking을 줄이고 최적화해보기 위해 UDP를 사용함.



항구라는 뜻.



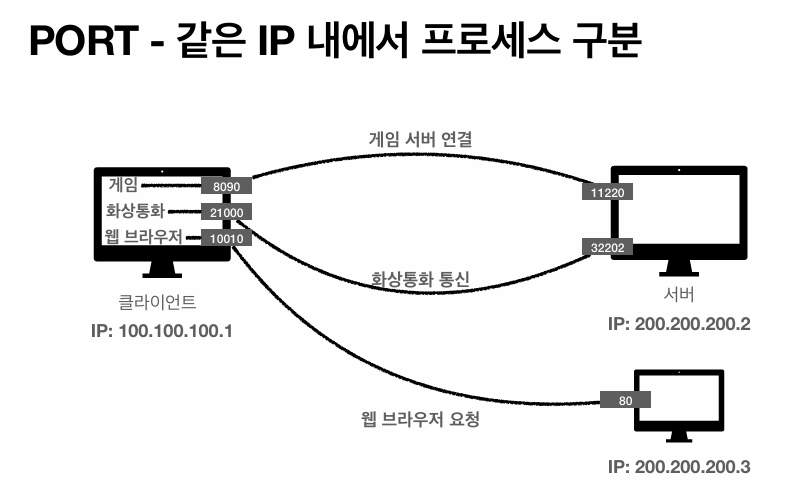
클라이언트가 여러 개의 서버와 통신을 해야한다. 패킷들이 내 ip를 향해서 오게 됨. ip만 가지고 어떤 애플리케이션으로 들어온 요청인지 알 수 없음. 보낼 때도 똑같음.



IP 목적지 서버를 찾는 용도

PORT 목적지 서버의 애플리케이션





패킷 보낼 때 서버에 포트정보를 패킷에 담아 보내기 때문에 응답 서버에서도 포트를 알고 응답데이터를 모낼 수 있다.

IP 가 아파트, 포트는 몇동 몇호를 의미함.

