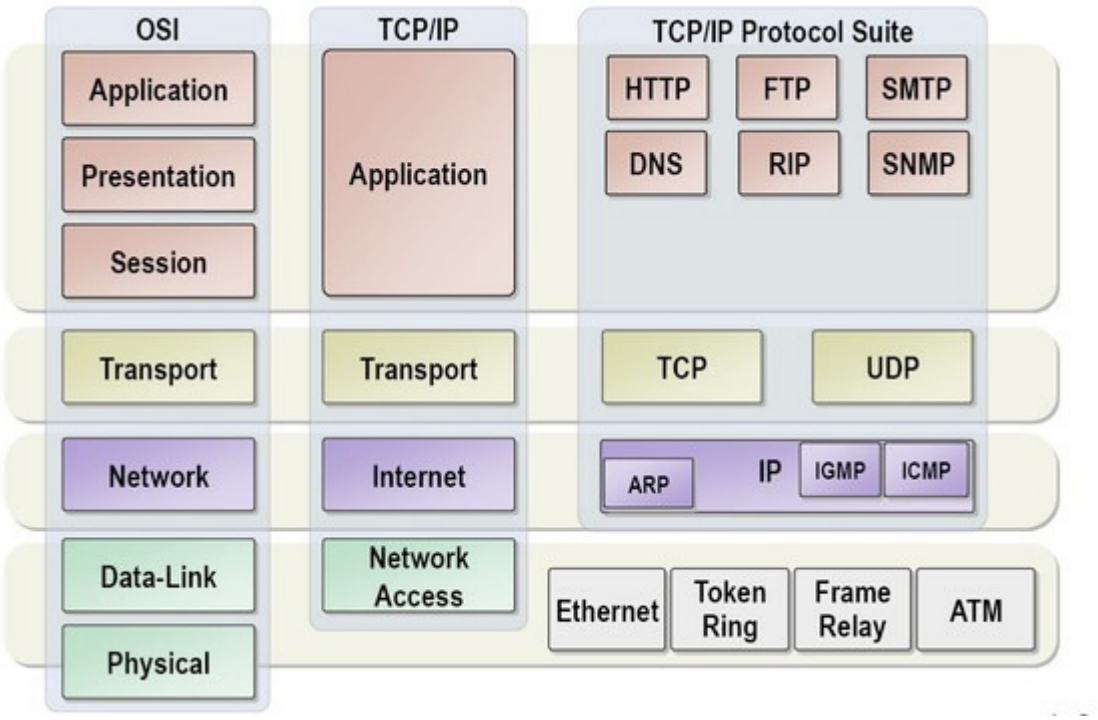
TCP 3 & 4 way handshake

트랜스포트 계층

OSI 7계층 구조에서 트랜스 포트 계층은 end-to-end의 사용자들이 **신뢰성 있는 데이터를 주고 받을 수 있도록 해 주어**, 상위 계층들이 데이터 전달의 유효성이나 효율성을 생각하지 않도록 해준다. 트랜스포트 계층은 인터넷의 기반인 TCP/IP 참조 모델과 일반적인 네트워크 모델인 개방형 시스템 간 상호 접속 (Open Systems Interconnection, OSI) 모두 포함하고 있다.



TCP(Transmission Control Protocol)

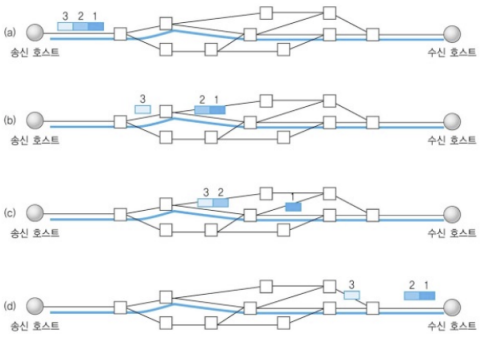
> 인터넷상에서 데이터를 메시지의 형태로 보내기 위해 IP와 함께 사용하는 프로토콜

TCP는 애플리케이션에 **신뢰적이고 연결지향성 서비스**를 제공한다. 일반적으로 TCP와 IP는 함께 사용되며 IP는 배달을, TCP는 **패킷의 추적 및 관리**를 하게 된다.

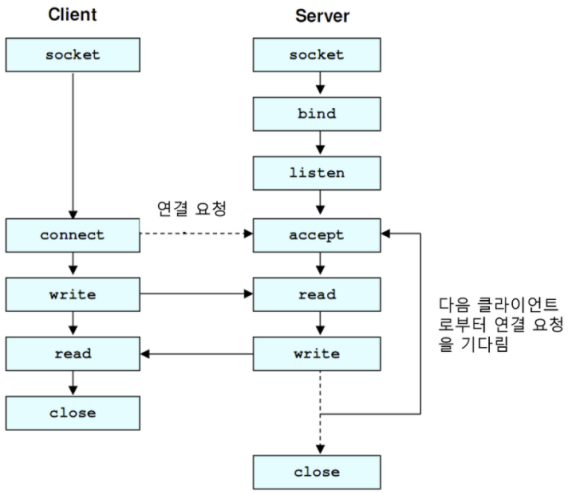
TCP는 **연결형 서비스**이다. 신뢰적인 전송을 보장하기에 handshaking 하고 데이터의 흐름 제어와 혼잡제어를 수행한다. 하지만 이러한 기능 때문에 속도가 느리다.

TCP 특징

- 3- way handshaking 과정을 통해 연결을 설정하고 4-way handshaking을 통해 해제한다.
- 흐름 제어 및 혼잡 제어
- 높은 신뢰성 보장
- UDP보다 속도가 느림
- 전이중(Full-Duplex), 점대점(Point to Point)방식



< 가상회선 패킷 교환 방식 >

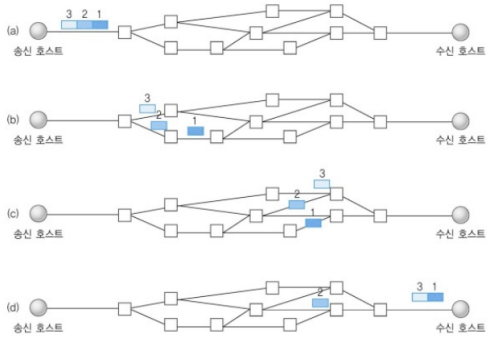


UDP (User Datagram Protocol)

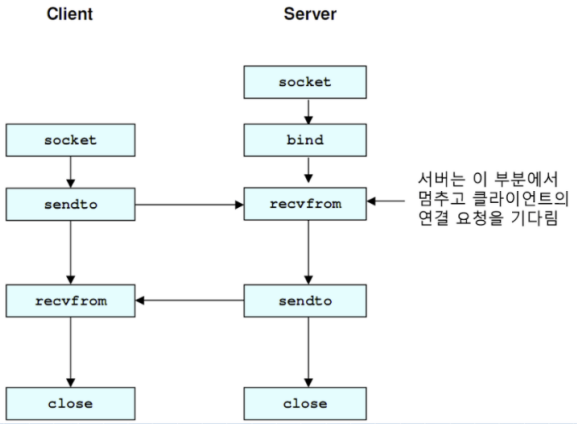
데이터를 **데이터그램 단위로 처리하는 프로토콜**이다(독립적인 관계를 지니는 패킷)

UDP는 **비연결형 프로토콜**이다. 즉, 할당되는 논리적인 경로가 없고 각각의 패킷이 다른 경로로 전송되고 이 각각의 패킷은 독립적인 관계를 지니게 되는데, 이렇게 데이터를 서로 다른 경로로 독립 처리 하는 프로토콜을 UDP라고 한다.

UDP는 연결을 설정하고 해제하는 과정이 존재하지 않는다. 서로 다른 경로로 독립적으로 처리함에도 **패킷에 순서를 부여하여 재조립하거나 흐름제어 및 혼잡제어를 수행하지 않아** 속도가 빠르며 네트워크 부하가 적다는 장점이 있지만 데이터 전송의 신뢰성이 낮다. 연속성이 중요한 실시간 서비스(스트리밍)에 좋다.



< 데이터그램 패킷 교환 방식 >



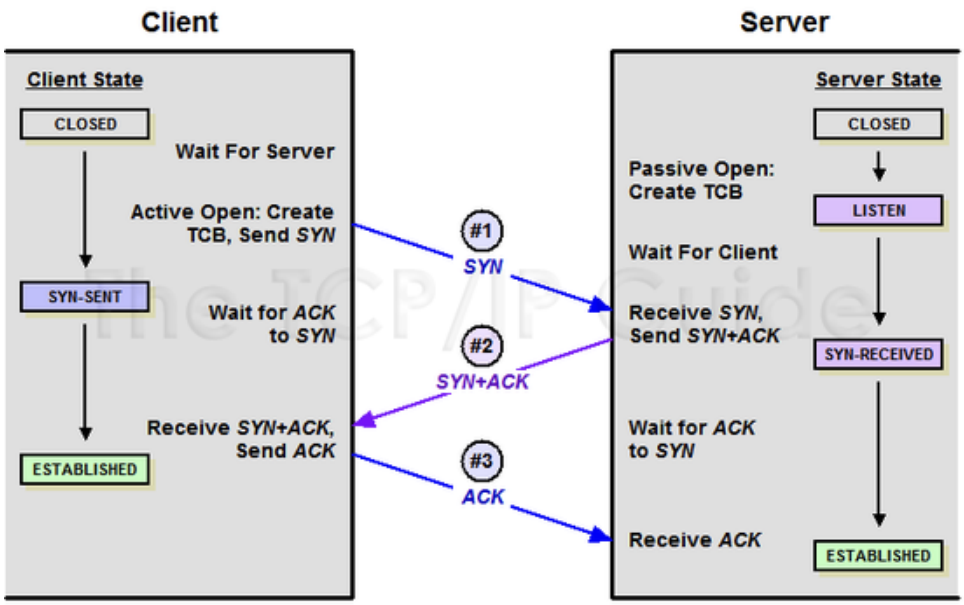
UDP 특징

- 비연결형 서비스로 데이터그램 방식을 제공한다
- 정보를 주고 받을 때 정보를 보내거나 받는다는 신호절차를 거치지 않는다.
- UDP헤더의 CheckSum 필드를 통해 최소한의 오류만 검출한다.
- 신뢰성이 낮다
- TCP보다 속도가 빠르다

TCP의 3-Way Handshake

> TCP 통신을 이용하여 데이터 전송하기 위해 네트워크 **연결을 설정 (Connection Establish)**하는 과정

양쪽 모두 데이터를 전송할 준비가 되었다는 것을 보장하고, **실제로 데이터 전달이 시작하기 전에 한 쪽이 다른 쪽이 준비되었다는 것을 알 수 있도록 한다**. 즉, TCP/IP 프로토콜을 이용해서 통신을 하는 응용 프로그램이 데이터를 전송하기 전에 먼저 정확한 전송을 보장하기 위해 **상대방 컴퓨터와 사전에 세션을 수립하는 과정**을 의미한다.



Client > Server : **TCP SYN**

Server > Client : **TCP SYN, ACK**

Client > Server : **TCP ACK**

- SYN : Synchronize Sequence Numbers
- ACK : Acknowledgement

Step 1

A클라이언트는 B서버에 접속을 요청하는 SYN 패킷을 보낸다.

이때 A클라이언트는 SYN 을 보내고 SYN/ACK 응답을 기다리는 **SYN_SENT** 상태, **B서버는 Wait for Client** 상태이다.

Step 2

B서버는 SYN요청을 받고 A클라이언트에게 요청을 수락한다는 ACK 와 SYN flag 가 설정된 패킷을 발송하고

A가 다시 ACK으로 응답하기를 기다린다. 이때 **B서버**는 **SYN_RECEIVED** 상태가 된다.

Step 3

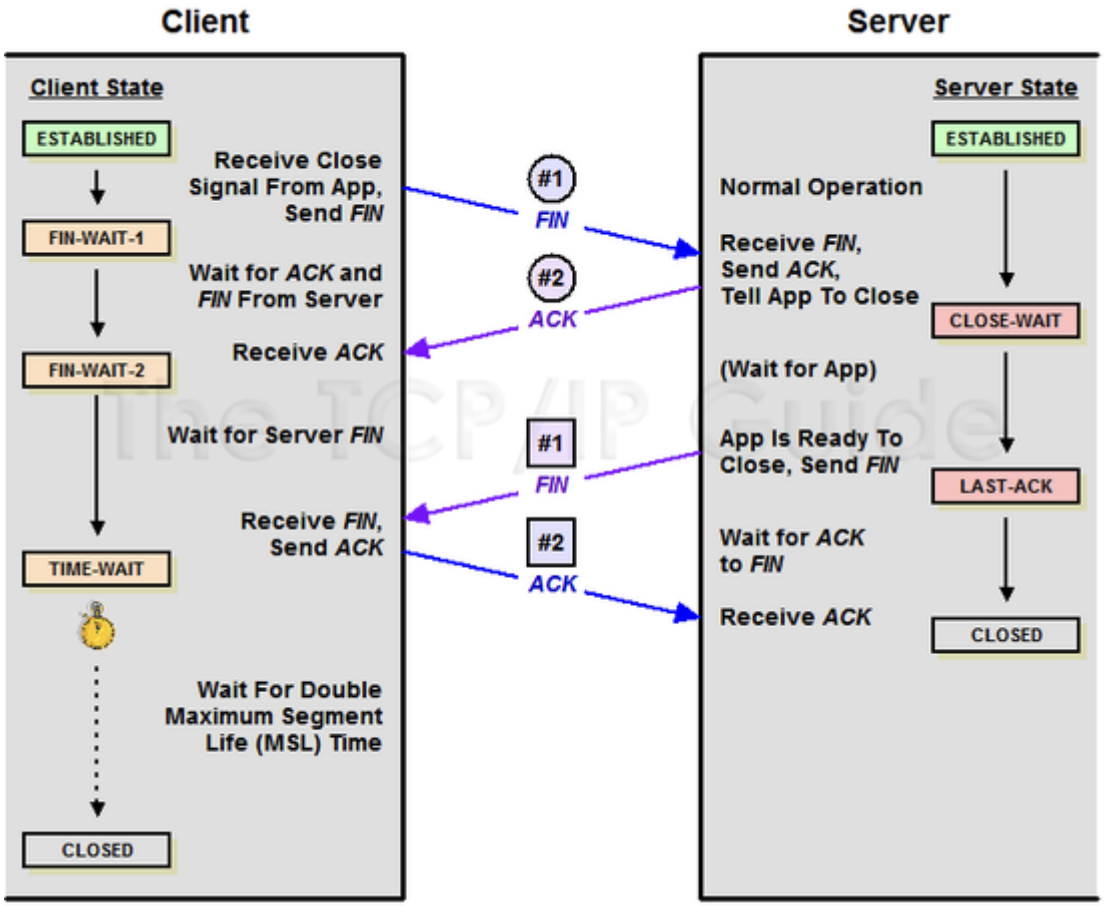
A클라이언트는 B서버에게 ACK을 보내고 이후로부터는 연결이 이루어지고 데이터가 오가게 되는것이다.

이때의 **B서버 상태가 ESTABLISHED** 이다.

위와 같은 방식으로 통신하는것이 신뢰성 있는 연결을 맺어 준다는 TCP의 3 Way handshake 방식이다.

TCP의 4-Way Handshake

> 3-Way handshake는 TCP의 연결을 초기화 할 때 사용한다면 4-Way handshakes는 세션을 종료하기 위해 수행되는 절차이다.



Step 1

클라이언트가 연결을 종료하겠다는 FIN플래그를 전송한다. 이때 **A클라이언트는 FIN-WAIT**
상태가 된다.

Step 2

B서버는 FIN플래그를 받고, 일단 확인메시지 ACK 보내고 자신의 통신이 끝날때까지 기다리는데 이 상태가 **B서버의 CLOSE_WAIT**상태다.

Step 3

연결을 종료할 준비가 되면, 연결해지를 위한 준비가 되었음을 알리기 위해 클라이언트에게 FIN플래그를 전송한다. 이때 B서버의 상태는 **LAST-ACK**이다.

Step 4

클라이언트는 해지준비가 되었다는 ACK를 확인했다는 메시지를 보낸다. **A클라이언트의 상태가 FIN-WAIT ->TIME-WAIT** 으로 변경된다.

A클라이언트는 이러한 현상에 대비하여 Client는 Server로부터 FIN을 수신하더라도 일정시간(디폴트 240초) 동안 세션을 남겨놓고 잉여 패킷을 기다리는 과정을 거치게 되는데 이 과정을 **"TIME_WAIT"**라고 합니다. 일정시간이 지나면, 세션을 만료하고 연결을 종료시키며, **"CLOSE"**상태로 변화합니다.