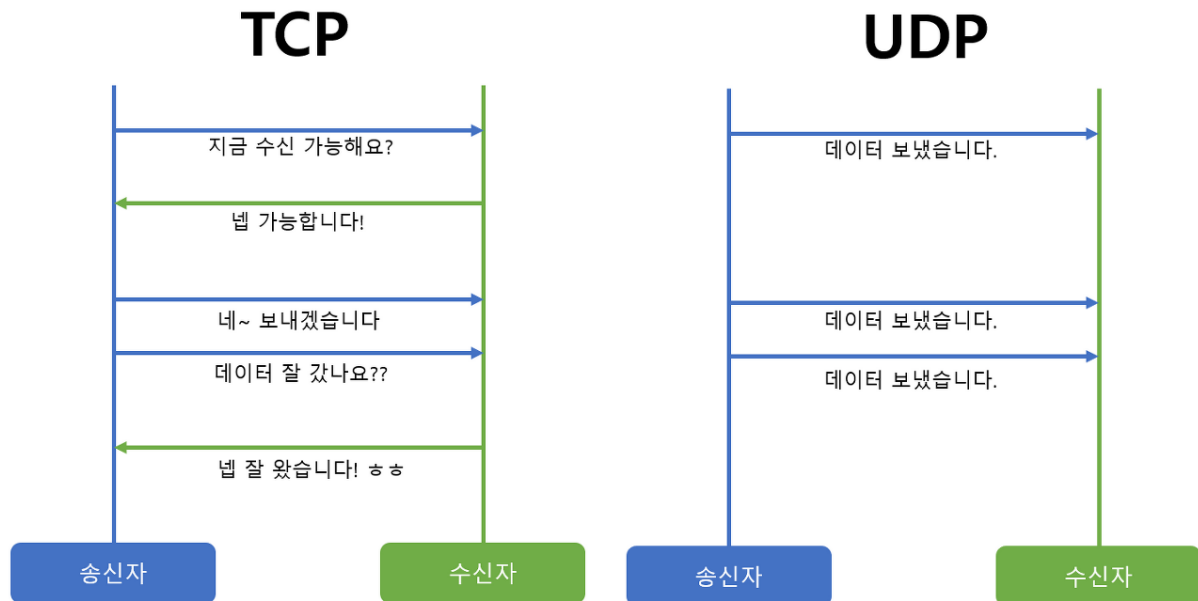


TCP vs UDP



TCP (Transmission Control Protocol, 전송 제어 프로토콜)

인터넷 상에서 데이터를 메시지의 형태로 보내기 위해 IP와 함께 사용하는 프로토콜

일반적으로 TCP와 IP를 함께 사용하는데, IP가 데이터의 배달을 처리한다면 TCP 는 패킷을 추적 및 관리하게 됨

연결형 서비스를 지원하는 프로토콜로 인터넷 환경에서 기본으로 사용함

신뢰성 있고 순서가 보장되는 전송 시스템. 연결 지향형 통신 제공

연결 지향 방식으로 패킷 교환 방식을 사용함

3-way-handshaking 과정을 통해 연결을 설정하고 4-way-handshaking 을 통해 해제함

혼잡 제어, 흐름 제어, 연결 설정 등 각종 오류를 제어함

전이중(Full-Duplex), 점대점(Point-to-Point) 방식

웹 브라우저들이 WWW에서 서버에 연결 할 때 사용되며, 이메일 전송 및 파일 전송에도 사용됨

연결 지향 방식이라는 것은 패킷을 전송하기 위한 논리적 경로를 배정한다는 것

3-way-handshaking 과정은 목적지와 수신지를 확실하게 하여 정확한 전송을 보장하기 위해 세션을 수립하는 과정을 의미함

특징

서버 소켓은 연결만을 담당함

연결과정에서 반환된 클라이언트 소켓은 데이터의 송수신에 사용됨

서버와 클라이언트는 일대일로 연결됨

스트림 전송으로 전송 데이터의 크기가 무제한

패킷에 대한 응답을 해야하기 때문에 (시간지연, CPU 소모) 성능이 낮음

손실된 경우 재전송을 요청하기 때문에 스트리밍 서비스에 불리함

TCP 에서의 패킷 추적 및 관리

데이터는 패킷단위로 나뉘어 같은 목적지(IP계층)으로 전송되는데, 이때 각 패킷에 번호를 부여하여 패킷의 분실 확인과 같은 처리를 하여 목적지에서 재조립함

이런 방식으로 TCP는 패킷을 추적하며, 나뉘어 보내진 데이터를 받고 조립을 할 수 있음

UDP (User Datagram Protocol, 사용자 데이터그램 프로토콜)

데이터를 데이터그램 단위로 처리하는 프로토콜

데이터그램이란 독립적인 관계를 지니는 패킷

비연결형 프로토콜. 즉, 연결을 위해 할당되는 논리적인 경로가 없음

따라서 각 패킷은 다른 경로로 전송되며, 각각의 패킷은 독립적인 관계를 지니게 되는데, 이렇게 데이터를 서로 다른 경로로 독립적으로 처리하게 됨

신뢰성이 낮고 순서가 보장되지 않는 전송 시스템. 비연결 지향형 통신 제공

특징

비연결형 서비스로 데이터그램 방식을 제공함

정보를 주고 받을 때 정보를 보내거나 받는다는 신호절차를 거치지 않음

UDP 헤더의 CheckSum 필드를 통해 최소한의 오류만 검출함

신뢰성이 낮음

TCP보다 속도가 빠름

연결을 설정하고 해제하는 과정이 존재하지 않음

서로 다른 경로로 독립적으로 처리함에도 패킷에 순서를 부여하여 재조립을 하거나 흐름제어 또는 혼잡제어와 같은 기능도 처리하지 않기 때문에 TCP보다 속도가 빠르며 네트워크 부하가 적으나 신뢰성을 보장하지는 못함

⇒ 꼭 필요한 전송만 하고, 오류 제어와 같은 추가 기능을 필요로 하지 않는 애플리케이션에 사용됨

UDP 서버의 특징

UDP에는 연결자체가 없어 서버 소켓과 클라이언트 소켓의 구분이 없음

소켓 대신 IP 를 기반으로 데이터를 전송함

서버와 클라이언트는 1대1, 1대N, N대M 등으로 연결 될 수 있음

데이터그램 단위로 전송되며 그 크기는 65536바이트로, 크기가 초과하면 잘라서 보냄

흐름제어가 없어서 패킷이 제대로 전송되었는지, 오류가 없는지 확인할 수 없음

파일 전송과 같은 신뢰성이 필요한 서비스보다 성능이 중요시 되는 경우에 사용됨

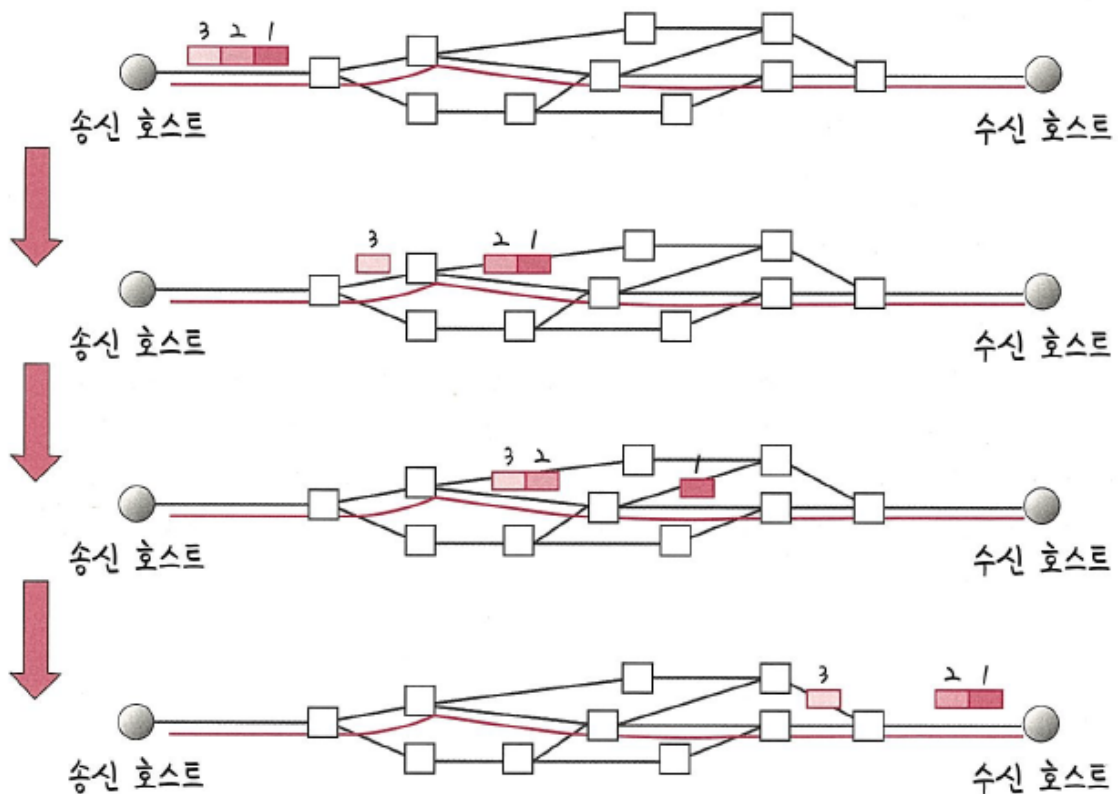
TCP vs UDP 비교 요약

TCP 는 속도보다 정확성을 지향하는 반면, UDP는 정확성보다 속도를 지향함

프로토콜 종류	TCP	UDP
연결 방식	연결형 서비스	비연결형 서비스
패킷 교환 방식	가상 회선 방식	데이터그램 방식
전송 순서	전송 순서 보장	전송 순서가 바뀔 수 있음
수신 여부 확인	수신 여부를 확인함	수신 여부를 확인하지 않음
통신 방식	1:1 통신	1:1 OR 1:N or N:N 통신
신뢰성	높다	낮다
속도	느리다	빠르다

패킷 교환 방식

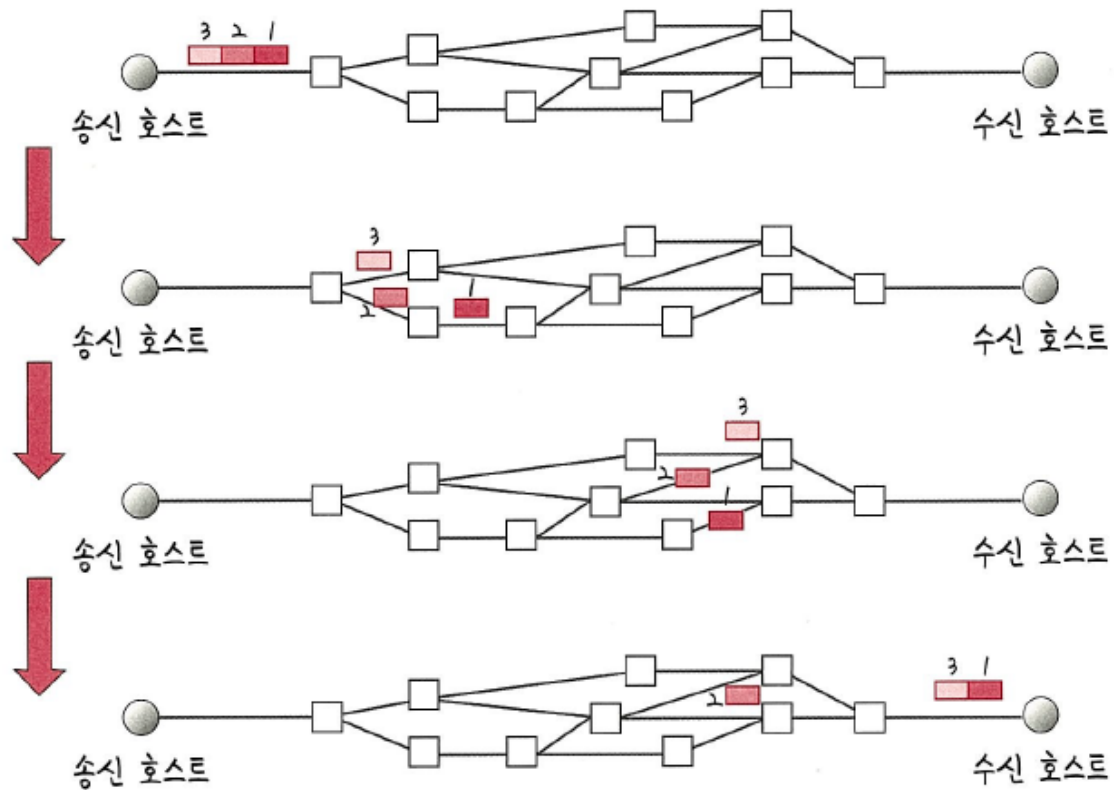
가상회선 패킷 교환 방식



각 패킷에는 가상 회선 식별자가 포함됨

모든 패킷을 전송하면 가상회선이 해제되고, 패킷들은 전송된 순서대로 도착하는 방식

- 데이터그램 패킷 교환 방식



패킷이 독립적으로 이동하며 최적의 경로를 선택하여 이동하는 방식

하나의 메시지에서 분할도니 여러 패킷은 서로 다른 경로로 전송 될 수 있으며, 도착한 순서가 다를 수 있는 방식

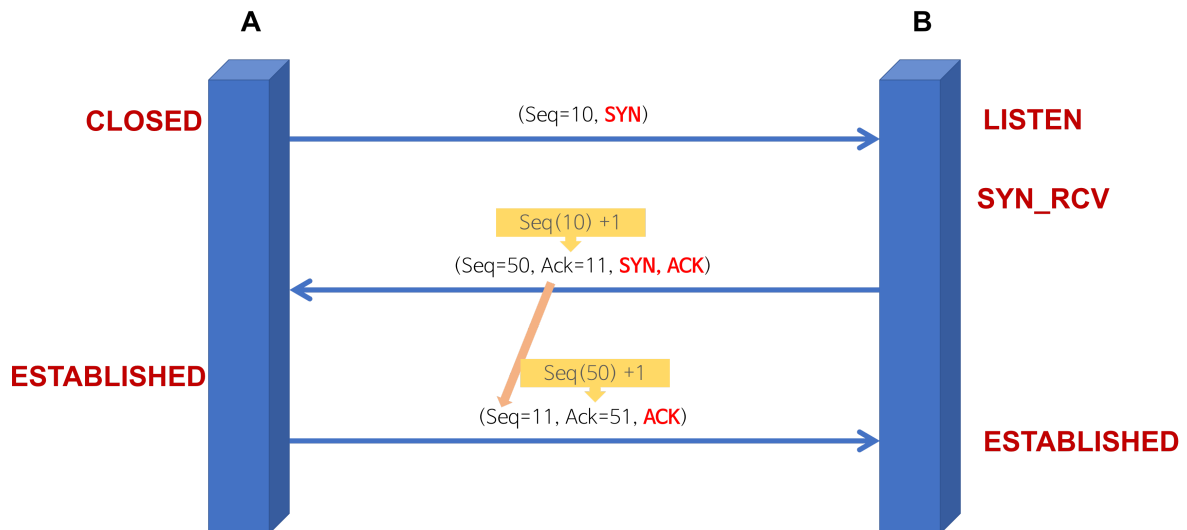
TCP 3-way-handshake & 4-way-handshake

3-way-handshake

TCP 통신을 이용하여 데이터를 전송하기 위해 네트워크 연결을 설정하는 과정

양측 모두 데이터를 전송할 준비가 되어있다는 것을 보장하고, 실제로 데이터 전달이 시작하기 전 한 쪽이 다른 쪽이 준비가 되었다는 것을 알 수 있도록 해야함

즉, 전송 전 정확한 전송을 보장하기 위해 상대방 컴퓨터와 사전에 세션을 수립하는 과정



A 프로세스가 B 프로세스에 연결 요청을 할 때

A는 클라이언트, B는 서버

- **A → B : SYN 단계**

접속을 요청할 프로세스 A가 프로세스 B에게 연결 메시지를 전송함 (SYN, Synchronization, 동기화 비트, 연결 요청 플래그)

송신자 A가 최초로 데이터를 전송할 때, Seq(Sequence Number)를 임의의 랜덤 숫자로 지정하고, SYN 플래그 비트를 1로 설정한 세그먼트를 전송함

포트 상태 — A : CLOSED(or SYN_SENT), B : LISTEN

- **B → A : SYN + ACK 단계**

접속 요청을 받은 프로세스 B가 요청을 수락했으며, 접속을 요청한 프로세스 A에게 CLOSED된 포트를 열어달라는 메시지를 전송함 (ACK, Acknowledgement, “OK” 응답을 알려주는 패킷)

수신자 (프로세스 B)는 ACK Number 필드를 Seq + 1로 지정하고, SYN과 ACK 플래그 비트를 1로 설정한 세그먼트를 전송함

포트 상태 — A : CLOSED(or SYN_SENT), B : SYN_RCV or SYN_RCEV(SYN Recieves, SYN를 받은 상태)

- **A → B : ACK 단계**

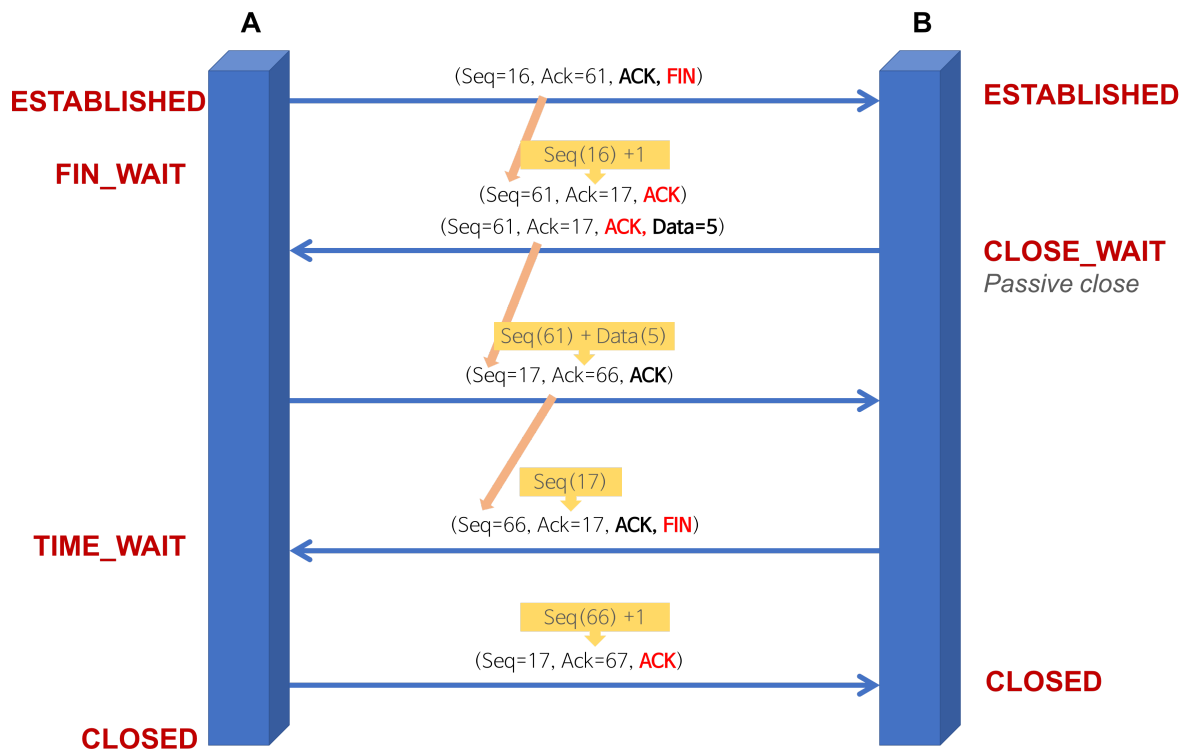
포트 상태 — A : ESTABLISHED(포트 연결), B : SYN_RCV or SYN_RCEV

마지막으로 접속 요청 프로세스 A가 수락 확인을 보내 연결을 맺음 (ACK)

포트 상태 — A : ESTABLISHED, B : ESTABLISHED

4-way-handshake

TCP의 연결을 해제하는 과정



A 프로세스가 B 프로세스에게 연결 해제를 요청 할 때

A는 클라이언트, B는 서버

1. A → B : FIN

프로세스 A가 연결을 종료하겠다는 FIN 플래그를 전송(FIN, Finish, 세션 연결 종료 시 사용하며, 더 이상 전송 할 데이터가 없을을 의미함)

프로세스 B가 FIN 플래그로 응답하기 전까지 연결을 계속 유지

2. B → A : ACK

프로세스 B는 일단 확인 메시지를 보내고 자신의 통신이 끝날 때까지 기다림 (TIME_WAIT 상태)

수신자는 ACK Number 필드를 (Seq + 1) 로 지정하고, ACK 플래그 비트를 1로 설정 한 세그먼트를 전송함

자신이 전송할 데이터가 남아있다면 이어서 계속 전송함

3. **B → A : FiN**

프로세스 B가 통신이 끝났으면 연결 종료 요청에 합의한다는 의미로 프로세스 A에게 FIN 플래그를 전송함

4. **A → B : ACK**

프로세스 A는 확인했다는 메시지를 전송함
