Network

1. TCP와 UDP의 헤더를 비교해주세요

TCP 헤더

Source Port (16 bit)			Destination Port (16 bit)
Sequence Number (32 bit)			
Acknowledgment Number (32 bit)			
Offset (4 bit)	Reserved (4 bit)	Flags (8 bit)	Window size (16 bit)
Checksum (16 bit)			Urgent Pointer (16 bit)
Option (0~40 byte)			

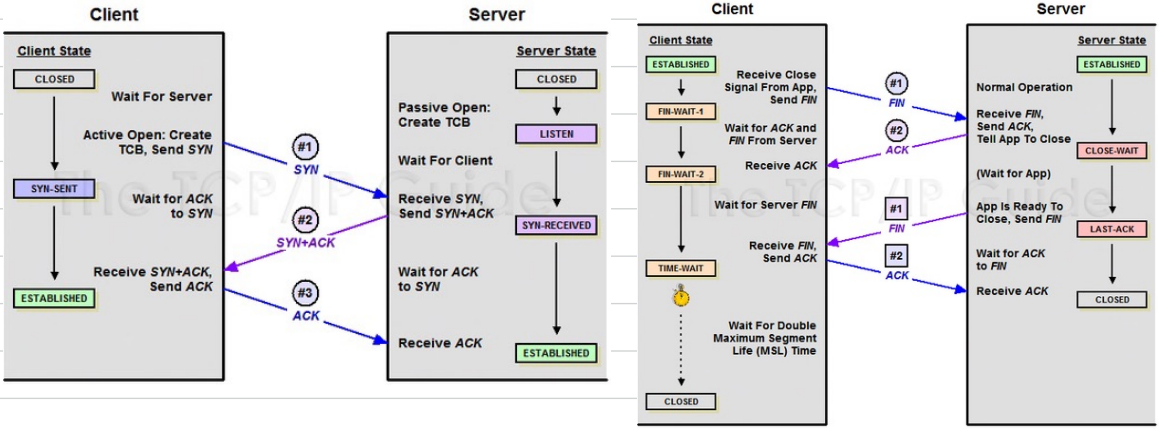
TCP 소켓은 4개의 서로 다른 요소들의 결합에 의해 식별된다. (Source IP, Source Port, Destination IP, Destination Port) 따라서 IP를 제외한 출발지/도착지 포트번호 (각 16 bit)와 Sequence Number, ack Number를 합쳐 기본적으로 20 byte의 헤더를 가지게 되며 옵션을 포함하면 최대 60 byte의 헤더를 가질 수 있다. 다른 출발지 주소를 가지는 세그먼트는, 다른 소켓을 통하여 프로세스에 전달된다. UDP와 다르게, TCP 세그먼트는 출발지 주소가 다르면 다른 소켓으로 전달된다.

UDP 헤더

Source Port (16 bit)	Destination Port (16 bit)
Length (16 bit)	Checksum (16 bit)

애플리케이션 데이터는 UDP 데이터그램의 데이터 필드에 위치한다. UDP 헤더는 2바이트(16비트)씩 구성된 4개의 필드를 가지게 된다. UDP헤더는 출발지 포트 번호, 목적지 포트 번호, 체크섬, 길이로 이루어져 있다.

2. TCP의 3-way handshake와 4-way handshake를 비교 설명하세요.



handshake란 호스트간의 데이터를 전송하기 전에 먼저 정확한 전송을 보장하기 위해 상대방 컴퓨터와 사전에 세션을 수립하는 과정을 의미한다.

3-way handshake는 TCP의 연결을 초기화 할 때 사용한다. 양쪽 모두 데이터를 전송할 준비가 되었다는 것을 보장하고, 실제로 데이터 전달이 시작하기전에 한쪽이 다른 쪽이 준비되었다는 것을 알수 있도록 한다. 양쪽 모두 상대편에 대한 초기 순차일련변호를 얻을 수 있도록 한다. 절차는 다음과 같다.

1. 접속 요청 프로세스가 연결 요청 메시지 전송한다.(SYN)
2. 접속 요청을 받은 프로세스가 요청을 수락한다는 확인 메시지를 보낸다. (ACK) 동시에 접속 요청을 받은 프로세스도 접속 요청을 한 프로세스에 연결 요청을 보낸다.(SYN) → (SYN + ACK)
3. 마지막으로 접속 요청 프로세스가 수락 확인을 보내 연결을 맺는다.(ACK)

> 단순히 응답을 주고받는데 2-way Handshake면 충분해보이지 않는가? 왜 3-way 일까?

TCP/IP 통신은 양방향성 connection 이다. 위의 그림의 1번 과정에서 클라이언트가 연결 요청을 SYN으로 보내면, 서버는 클라이언트가 요청한 SYN에 대한 대답(ACK)과 함께, 자신도 연결하겠다는 요청의 의미로 SYN을 보내고, 클라이언트로부터 요청에 대한 대답(과정 3)을 받아야한다. 이 과정은 2-way handshaked에서는 성립될 수 없다.

4-way handshake는 세션을 종료하기 위해 수행되는 절차이다. 구체적인 과정은 다음과 같다.

1. 클라이언트가 연결을 종료하겠다는 FIN 플래그를 전송한다.
2. 서버는 일단 확인메시지를 보내고 자신의 통신이 끝날때까지 기다리는데, 이 상태가 `TIME_WAIT` 상태이다.
3. 서버가 통신이 끝났으면 연결이 종료되었다고 클라이언트에게 FIN 플래그를 전송한다.
4. 클라이언트는 확인했다는 메시지를 보낸다.

> `CLOSE_WAIT` 와 `TIME_WAIT` 상태란 무엇일까?

`TIME_WAIT` 상태로 대기하는 이유는, 세션 종료후, 혹시나 네트워크에 아직 라이브 패킷이 존재할수도 있기때문이다.

> 용어

- **SYN**(Synchronization): 연결요청, 세션을 설정하는데 사용되며 초기에 시퀀스 번호를 보낸다.
- **ACK**(Acknowledgement): 보낸 시퀀스 번호에 TCP 계층에서의 길이 또는 양을 더한 것과 같은 값을 ACK에 포함하여 전송한다.
- **FIN**(Finish) : 세션을 종료시키는데 사용되며 더 이상 보낸 데이터가 없음을 표시한다.