

TCP UDP

TCP (Transmission Control Protocol)

전송을 제어하는 프로토콜(규약)

→ 인터넷상에서 데이터를 메시지의 형태로 보내기 위해 **IP**와 함께 사용하는 프로토콜

TCP 와 IP는 **Pair**로 주로 사용됨(TCP/IP)

IP : 데이터의 배달부

TCP : 패킷을 추적 및 관리하는 관리자

패킷(Packet)

인터넷 내에서 데이터를 보내기 위한 경로배정
(라우팅)을 효율적으로 하기 위해서 데이터를
여러 개의 조각으로 나누어 전송하게 됨
“조각 == 패킷”

TCP 특징

1. 연결 지향 방식으로 패킷 교환 방식을 사용한다(가상 회선 방식이 아님)
2. 3-way handshaking 과정을 통해 연결을 설정하고 4-way handshaking을 통해 해제한다.
3. 흐름 제어 및 혼잡 제어
4. 높은 신뢰성 보장
5. UDP보다 전송 속도가 느리다
6. 전이중(Full-Duplex), 점대점(Point to Point) 방식

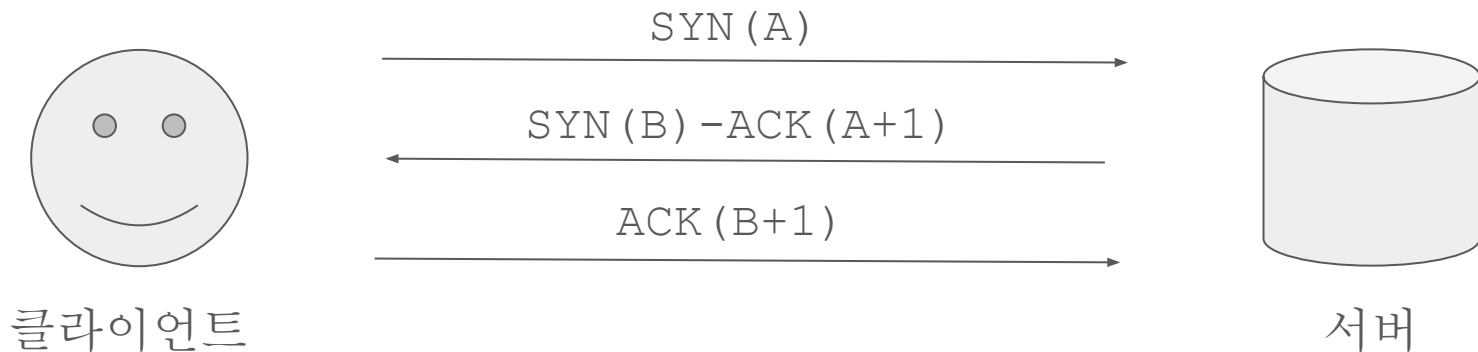
연결 지향 방식?

패킷을 전송하기 위한 논리적 경로를 배정한다

3-way handshaking?

정확한 데이터 전송을 보장하기 위해 목적지와 수신지를 지정하여 세션을 수립하는 방식

3-way handshaking?

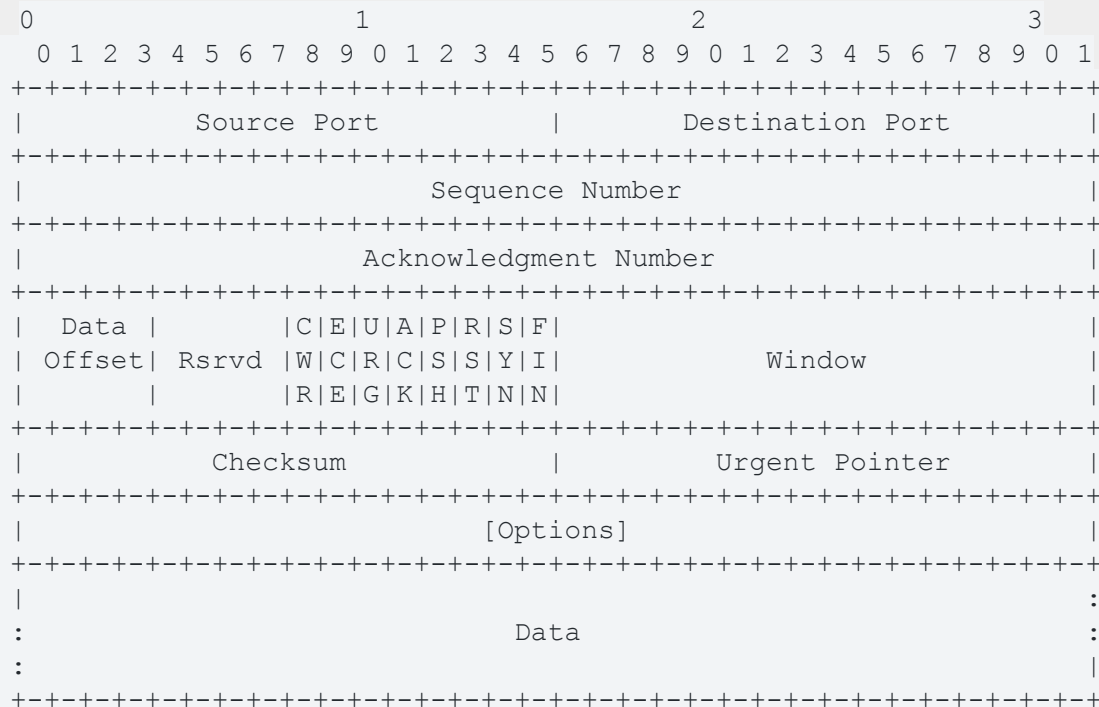


1. **SYN**: 클라이언트가 서버에게 **SYN(synchronize)** 플래그가 설정된 메시지를 보냅니다. 이때 **SYN**은 클라이언트의 임의 시퀀스 번호 **A**를 포함합니다.
2. **SYN-ACK**: 서버는 클라이언트의 **SYN**에 응답하여 **SYN-ACK** 메시지를 보냅니다. 서버는 자신만의 새로운 시퀀스 번호 **B**를 포함한 **SYN**을 전송하고, 클라이언트의 시퀀스 번호 **A**에 1을 더한 **ACK(acknowledgment)**를 함께 보냅니다(**A+1**).
3. **ACK**: 클라이언트는 서버로부터 받은 **SYN**에 대해 **ACK** 메시지를 전송합니다. 이때 **ACK**는 서버의 시퀀스 번호 **B**에 1을 더한 값(**B+1**)을 포함합니다.

TCP 통신 과정

1. 데이터 스트림에서 받은 데이터를 일정 단위로 분할
2. 분할된 데이터 단위에 **TCP** 헤더를 붙여서 **TCP 세그먼트**를 생성
3. **TCP** 세그먼트를 **IP** 데이터그램으로 변환
(IP 데이터그램은 인터넷 통신에 사용되는 데이터 패킷)
4. IP 데이터그램을 수신 애플리케이션에 전송

TCP 세그먼트



Source Port: 데이터를 발송하는

애플리케이션의 포트 번호

Destination Port: 데이터를 수신하는

애플리케이션의 포트 번호

Sequence Number(SYN): TCP 통신

과정에서 데이터를 일정 단위로 분할된

데이터의 순서

Acknowledgment Number(ACK):

데이터를 수신하는 애플리케이션

입장에서, 다음으로 받고 싶은 TCP

세그먼트의 **Sequence Number**

TCP Server 특징

1. 서버소켓은 연결만을 담당
2. 연결 과정에서 반환된 클라이언트 소켓은 데이터의 송수신에 사용된다
3. 서버와 클라이언트는 1: 1 연결된다
4. 스트림 전송으로 전송 데이터의 크기가 무제한이다
5. 패킷에 대한 응답을 해야하기 때문에(시간 지연, CPU소모) 성능이 낮다.
6. Streaming 서비스에 불리하다.(손실된 경우 재전송 요청)

TCP 패킷 추적 관리 방식

데이터는 패킷 단위로 나누어 같은 목적지(IP 계층)으로 전송됨

ex) 서울(발신지) → 부산(수신지)로 가야하는 사람 A, B, C가 존재

셋은 순차적으로 목적지까지 도달해야 한다.

A, B, C가 순차적으로 가야 하는 상황에서 B가 길을 잘못 들어서 분실됨

하지만 목적지에서는 A, B, C가 모두 필요한지 모르고 A, C만 보고 다 왔다고 착각할 수 있음

그렇기 때문에 TCP에서는 A, B, C 패킷에 1, 2, 3이라는 번호를 부여하여 패킷의 분실 확인과 같은 처리를 하여 목적지에서 재조립

TCP 사용 환경

연속성보다 신뢰성있는 전송이 중요할 때에 사용하는 프로토콜

ex) 파일 전송

UDP(User Datagram Protocol)

사용자 데이터그램 프로토콜(규약)

→ 데이터를 데이터그램 단위로 처리하는 프로토콜

비연결형 프로토콜

TCP와 다르게 연결을 위해 할당되는 논리적인 경로 X

→ 각각의 패킷은 다른 경로로 전송되고, 각각의 패킷은 독립적인 관계를 지니게 됨.

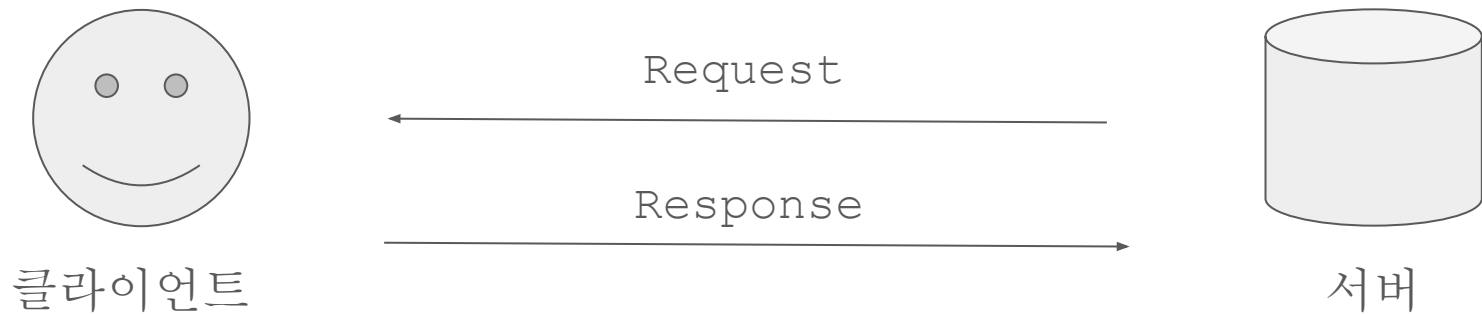
UDP 특징

1. 비연결형 서비스로 데이터그램 방식을 제공한다.
2. 정보를 주고 받을 때 정보를 보내거나 받는다는 신호 절차를 거치지 않는다.
3. UDP 헤더의 CheckSum 필드를 통해 최소한의 오류만 검출한다.
4. 신뢰성이 낮다.
5. TCP보다 속도가 빠르다

UDP Server 특징

1. UDP에는 연결 자체가 없어서(**connect** 함수 불필요) 서버 소켓과 클라이언트 소켓의 구분이 없다
2. 소켓을 활용해 **IP**와 **PORT** 기반으로 데이터를 전송한다
3. 서버와 클라이언트는 **1:1, 1:N, N:M** 연결 가능
4. 데이터그램(메세지) 단위로 전송되며, 그 크기는 **65535Byte**, 크기가 초과하면 잘라서 보낸다.
5. 흐름제어(**flow control**)이 없어서 패킷이 제대로 전송되었는지, 오류가 없는지 확인할 수 가 없다.
6. 파일 전송과 같은 신뢰성이 필요한 서비스보다는 성능이 중요시 되는 경우 사용된다.

UDP 통신 방식



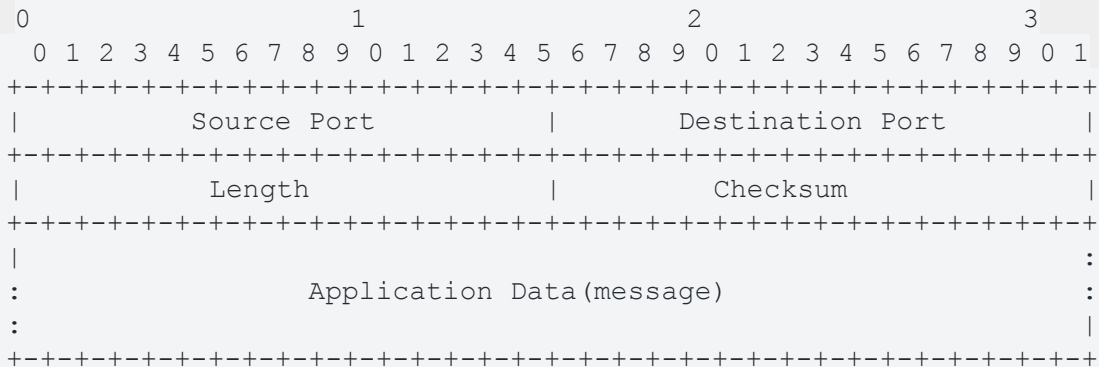
흐름 제어

데이터를 송신하는 곳과 수신하는 곳의 데이터 처리 속도를 조절하여 수신자의 오버플로우를 방지하는 것

혼잡 제어

네트워크 내의 패킷 수가 넘치게 증가하지 않도록 방지하는 것

UDP 세그먼트



UDP 사용 환경

신뢰성보다는 연속성이 중요한 서비스에 적합

ex) 실시간 서비스(Streaming)