

# [TCP/IP] 전송 계층 (Transport Layer) - TCP, UDP

☰ 태그	
🕒 날짜	@2023년 12월 15일 오전 12:05

▼ 목차

- [전송 계층이란?](#)
- [TCP \(Transmission Control Protocol\)](#)
  - [TCP란?](#)
  - [특징](#)
  - [TCP 3-way handshake](#)
  - [연결 설정 방식](#)
- [UDP](#)
  - [UDP란?](#)
  - [UDP를 사용하는 이유](#)
- [정리](#)

## 전송 계층이란?

	TCP/IP 4계층	주요 프로토콜	역할
4층	응용 계층	HTTP, DNS, FTP, ...	애플리케이션에 맞추어 통신
3층	전송 계층	TCP, UDP, ...	IP와 애플리케이션을 중개해 데이터를 확실하게 전달
2층	인터넷 계층	IP, ICMP, ARP, RARP	네트워크 주소로 기반으로 데이터 전송
1층	네트워크 접근 계층	Ethernet, wifi, ...	컴퓨터를 물리적으로 네트워크에 연결해 기기 간 전송

전송 계층은 프로세스 간의 신뢰성 있는 데이터 전송을 담당하는 계층입니다.

**연결 지향 통신, 신뢰성, 흐름제어, 다중화**와 같은 서비스를 제공합니다.

전송 계층의 프로토콜에는 **TCP**와 **UDP**, DCCP, SCTP가 있습니다.

TCP와 UDP는 2계층에서 동작하는 IP 와 4계층에서 동작하는 애플리케이션(http 등)을 중개하는 역할을 합니다.

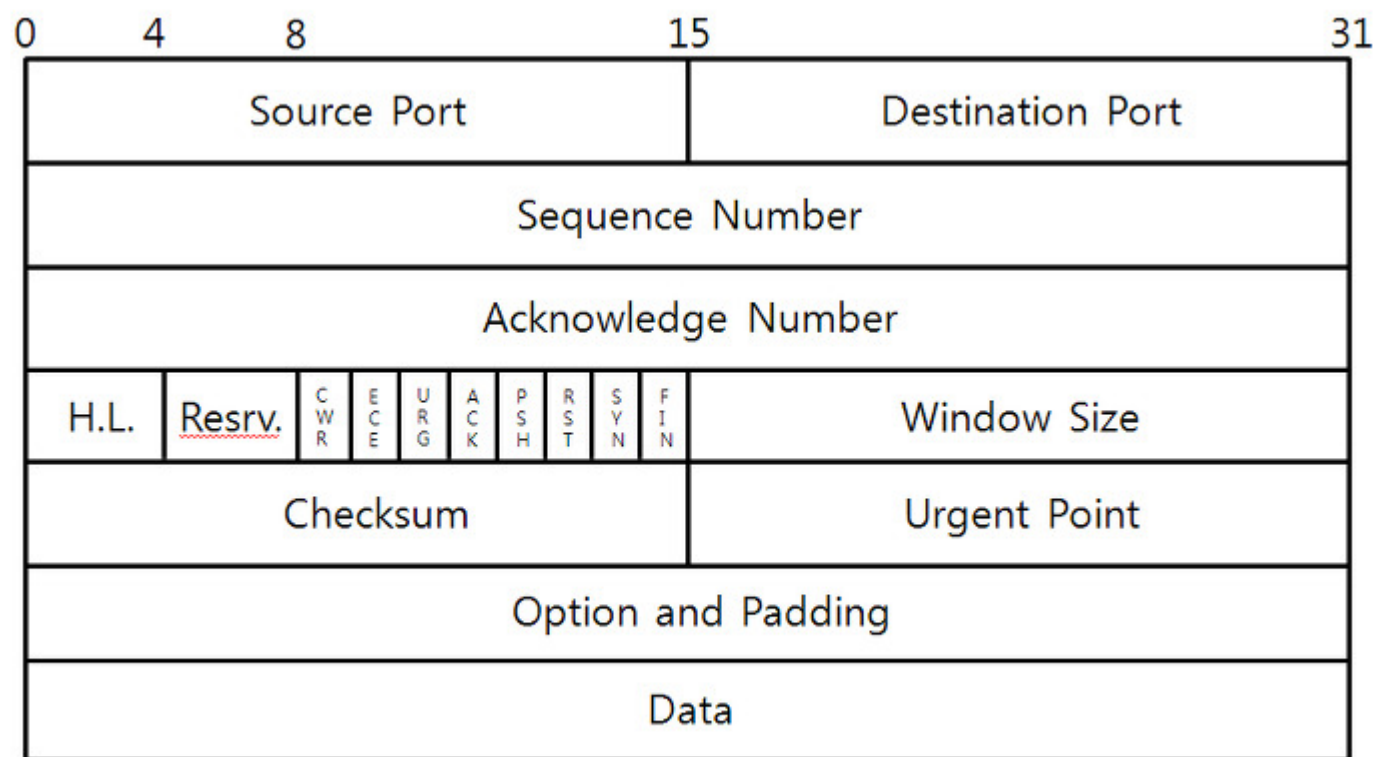
## TCP (Transmission Control Protocol)

### TCP란?

TCP는 신뢰성있는 연결지향형 프로토콜입니다.

신뢰성있다는 말은 그 패킷에 대한 오류처리나 재전송따위로 에러를 복구하는 것을 말합니다. 그때문에 TCP의 헤더에 붙는 정보가 많습니다.

▼ 헤더 구조



[그림 1] TCP 구조

## 특징

- 연결지향

데이터 전송 전 클라이언트와 서버 간의 연결 설정이 이루어집니다.

연결은 3-way handshake로 이루어집니다.

(TCP 세션을 종료할 때는 4-way handshake로 이루어집니다)

- 신뢰성 보장

TCP는 손상되거나 없어지거나 중복되거나 네트워크 계층에서 순서가 틀어져서 전달된 데이터를 복구합니다. 즉, TCP는 신뢰성을 실현하기 위해 적극적 수신, 통지, 재전송 체계를 사용합니다.

- 흐름 제어

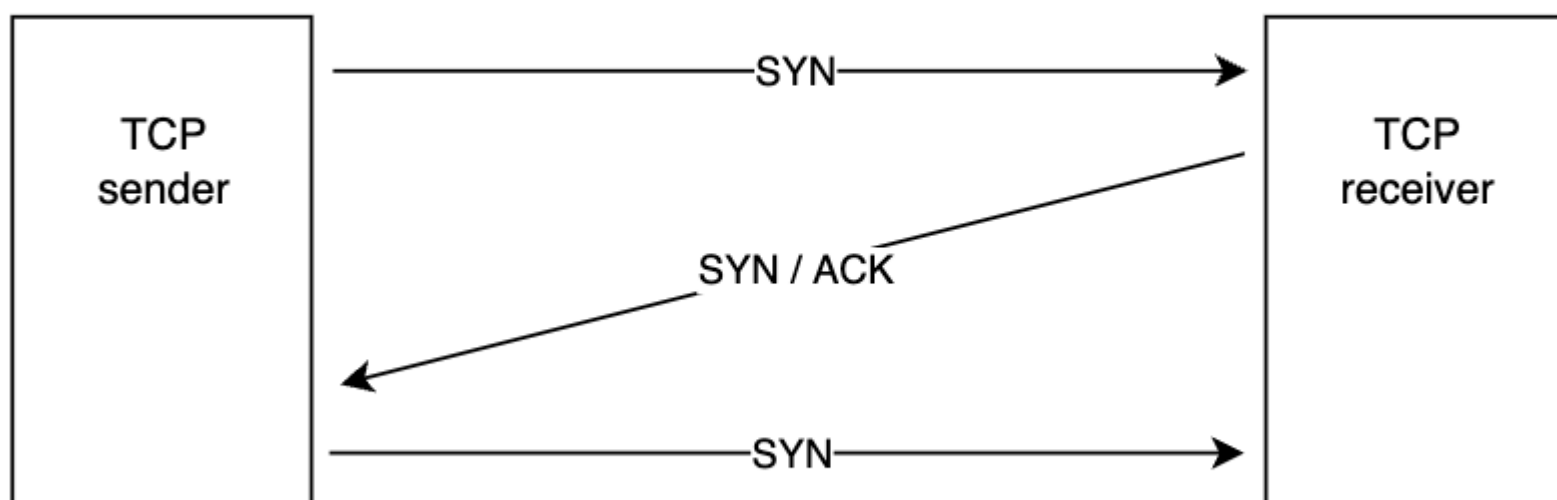
TCP 데이터 세그먼트를 송수신하는 컴퓨터는 CPU와 네트워크 대역폭의 차이 때문에 서로 다른 데이터 속도로 작동할 수 있습니다.

결국 수신자가 처리할 수 있는 것처럼 훨씬 더 빠른 속도로 송신자가 데이터를 보낼 가능성이 많다. TCP는 송신자가 보낸 데이터의 양을 제어하는 흐름 제어 메커니즘을 구현합니다.

- 혼잡 제어

## TCP 3-way handshake

### TCP 3-way handshake

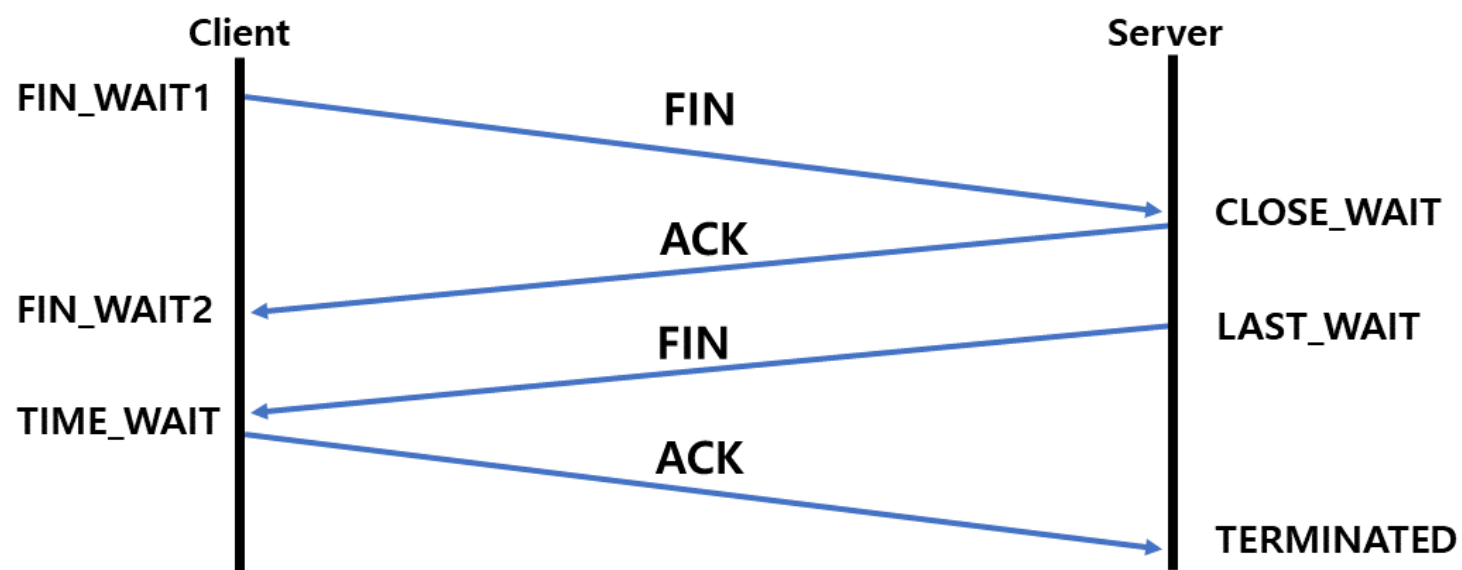


- TCP 3-way handshake 는 양 끝단의(end to end) 기기의 신뢰성 있는 데이터 통신을 위해, TCP 방식이 연결을 설정하는 방식

## 연결 설정 방식

- Step 1 (SYN): 처음으로, sender는 receiver와 연결 설정을 위해, segment를 랜덤으로 설정된 SYN(Synchronize Sequence Number) 와 함께 보냅니다. 이 요청은 receiver에게 sender가 통신을 시작하고 싶다고 알립니다.
- Step 2(SYN / ACK): receiver 는 받은 요청을 바탕으로 SYN/ACK 신호 세트를 응답합니다. Acknowledgement(ACK) 응답으로 보내는 segment가 유효한 SYN요청을 받았는지를 의미합니다.
- Step 3(ACK): 마지막 단계에서, sender는 받은 ACK를 receiver에게 전송을 하면서, 신뢰성 있는 연결이 성립되었다는 사실을 sender와 receiver 양쪽에서 알 수 있고, 실제 데이터 전송이 시작되게 됩니다.

### ▼ 4-way handshake



1. 클라이언트가 FIN flag를 보낸다.
2. 서버가 ACK flag로 답변한다.
3. 서버가 통신을 중단한 준비가 되었다면, FIN flag를 클라이언트에게 보낸다.
4. 클라이언트는 서버가 보낸 FIN flag에 대해 ACK flag를 보내고 통신이 중단 된다.

### 💡 연결을 끊어낼 때는 4 way로 하는 이유

연결을 시작할 때는 서로 통신 중이 아닌 상태에서 handshake를 시도하지만, 끝낼 때는 이미 상호 통신 중이기 때문에 한쪽에서 일방적으로 끊어낼 수 없다.

반대편도 끝낼 준비를 마치고, 통신이 종료되어야하기 때문에 4-way handshake하는 것이다.

## UDP

### UDP란?

비신뢰성 비연결형 프로토콜입니다.

패킷을 잃거나 오류가 있어도 대처하지 않는 것을 말합니다. 따라서 UDP헤더는 간단한 구조를 갖고 있습니다.

## UDP를 사용하는 이유

### 1. 애플리케이션의 정교한 제어 가능

- TCP의 경우 receiver가 전송 받을 준비가 될 때까지 세그먼트를 반복적으로 재전송 → latency ↑
- 실시간 전송에 대한 요구가 큰 애플리케이션 들은 높은 latency를 지양하여 약간의 데이터 손실을 감수한다. 대신 개발자 스스로가 이를 보완하기 위해 애플리케이션에 추가 기능을 구현할 수 있다.

### 2. 연결 설정에 무관

- TCP 3-way handshake 가 없는 UDP는 예비 과정 없이 바로 전송을 시작
- 설정 단계에서 발생하는 지연이 없는 만큼, 반응 속도가 빠르다
- TCP 가 신뢰성을 위해 많은 파라미터와 정보 전달이 필요함과 비교해 UDP는 연결 설정 관리를 하지 않기 때문에 어떠한 파라미터도 기록하지 않는다.
- 이때문에 서버에서도 TCP와 비교에 더 많은 클라이언트 수용이 가능하다

## 정리

	TCP	UDP
서비스 타입	연결 지향적 프로토콜	데이터그램 지향적 프로토콜 (비연결형 프로토콜)
신뢰성	데이터 전송 표적 기기까지의 전송을 보장O (신뢰성 👍)	데이터 전송 표적 기기까지의 전송을 보장X (비신뢰성)
순서 보장	전송하는 패킷 순서 보장O (신뢰성 👍)	패킷 순서 보장X, 패킷 순서를 보장하고 싶으면 애플리케이션 레이어에서 관리되어야 함
속도	UDP에 비해 느림	TCP에 비해 빠름, 단순하고 더 효율적인 속도 (속도 👍)

- **TCP (Transmission Control Protocol)** : 신뢰성있는 연결지향형 프로토콜. 신뢰성있다는 말은 그 패킷에 대한 오류처리나 재전송따위로 에러를 복구하는 것을 말합니다. 그때문에 TCP의 헤더에 붙는 정보가 많습니다.
- **UDP (User Datagram Protocol)** : 비신뢰성 비연결형 프로토콜. 패킷을 잃거나 오류가 있어도 대처하지 않는 것을 말합니다. 따라서 UDP헤더는 간단한 구조를 갖고 있습니다.
- TCP → 통신 신뢰성을 높이는 기능 구현 → 데이터의 신뢰성을 필요로하는 애플리케이션에 사용
- UDP → 높은 속도, 효율성 → 빠른 속도나 실시간 통신이 중요한 애플리케이션에 사용
- HTTP는 모든 데이터를 제대로 송수신이 가능해야 하는 특성상, TCP를 사용

## 참고

<https://musclebear.tistory.com/2>

<https://www.geeksforgeeks.org/why-tcp-connect-termination-need-4-way-handshake/>