Transport Layer

백설기 3차 스터디

AI6/1반 조해정

지난 주 발표 자료: OSI 7 Layer

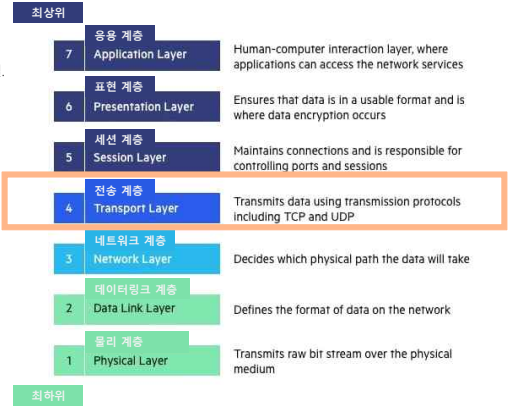
OSI 7 Layer

통신이 일어나는 과정을 **7단계**로 구분해서 정의한 모델.
모든 계층은 **독립적, 상하구조**를 가집니다.

◆상위층
- 응용 계층(**A**pplication Layer)
- 표현 계층(**P**resentation Layer)
- 세션 계층(**Se**ssion Layer)

◆하위층
- 전송 계층(**T**ransport Layer)
- 네트워크 계층(**Ne**twork Layer)
- 데이터링크 계층(**Da**ta Link Layer)
- 물리 계층(**P**hysical Layer)

- 아파서 티내다, 피나다



글, 그림 참조:
https://ko.wikipedia.org/wiki/OSI_%EB%AA%A8%ED%98%95
https://www.imperva.com/learn/application-security/osi-model/

CONTENTS

Transport Layer

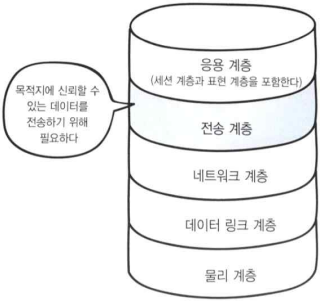
OSI모델의 1~3계층을 거쳐 목적지에 데이터를 보낼 수 있습니다.
네트워크 계층에서 다른 네트워크로 데이터를 전송하려면 라우터가 필요하고,
라우터의 라우팅 기능을 사용하여 데이터를 전송할 수 있지만,

- 수신지 컴퓨터가 인터넷에 존재하는지?
- 수신지 컴퓨터가 존재하더라도 패킷을 수신할 준비가 되어있는지?
- 준비가 되어있더라도 전송 과정에서 패킷이 손상되거나 유실되지는 않았는지?

이 3계층에서는 이런 문제들을 신경 쓰지 않습니다.

IP 프로토콜은 통신하는 호스트 간에 패킷을 전달하기 위해 최선의 노력을 하지만 패킷의
전송 순서나 완전성을 보장하기 않기 때문에 비신뢰형 서비스라고도 합니다.

>> OSI모델의 **전송 계층**은 **목적지에 신뢰할 수 있는 데이터를 전달하기 위해 필요**합니다.

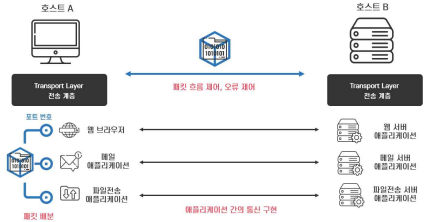


글, 그림 참조:
https://velog.io/@april_5/%EB%84%A4%ED%8A%B8%EC%9B%8C%ED%81%AC-%EA%B5%AC%EC%A1%B0-%EC%9D%B4%ED%95%B4-%EC%A0%84%EC%B6%A1-%EA%B3%84%EC%B8%B5-%EC%8B%A0%EB%A2%B0%ED%95%A0-%EC%88%98-%EC%9E%88%EB%8A%94-%EB%8D%B0%EC%9D%B4%ED%84%B0-%EC%A0%84%EC%86%A1%ED%95%98%EA%B8%B0

Transport Layer

전송 계층의 **두 가지 역할**

- **데이터가 제대로 도착했는지 확인**
- **전송된 데이터의 목적지가 어떤 애플리케이션인지 식별**

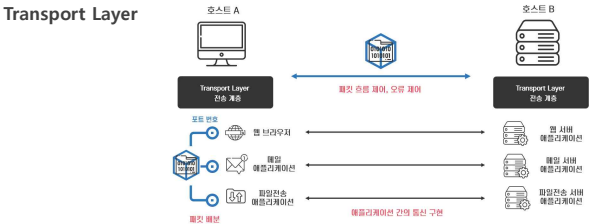


1. 패킷이 전송 과정에서 아무 문제 없이 제대로 수신지 컴퓨터에 도착할 수 있도록 패킷 전송을 제어하는 역할

   전송 계층은 네트워크 혼잡 상황에 따라 패킷의 전송량을 조절하여 **패킷의 흐름을 제어**하고 패킷 전송의 **오류를 점검**해서 수신지 컴퓨터까지 패킷이 **제대로 도착했는지 확인**하는 역할을 합니다.

   다시 말해 전송 계층은 수신지 컴퓨터까지 신뢰할 수 있는 데이터를 전송하기 위해 필요한 계층입니다.

글, 그림 참조:
https://better-together.tistory.com/134



2. 애플리케이션이라는 최종 목적지까지 데이터 전송을 책임지는 역할

데이터를 전송하는 목적은 애플리케이션이 데이터를 처리하여 애플리케이션의 목적에 따른 서비스를 제공하기 위한 것입니다. 따라서 데이터는 수신지 컴퓨터 내의 애플리케이션까지 전송되어야 최종 목적지에 도착하게 됩니다.

전송 계층은 다양한 애플리케이션이 동작하는 컴퓨터 내에서 **어떤 애플리케이션**이 사용하는 데이터인지 **식별**하여 수신지 컴퓨터에 도착한 데이터를 수신지 컴퓨터 내의 애플리케이션에 **배분**하는 역할을 합니다.

네크워크 계층이 **컴퓨터(호스트) 간의 데이터 통신**을, **전송 계층**이 **애플리케이션 간의 데이터 통신**을 구현한다고 볼 수 있습니다.

네트워크 계층이 호스트를 식별하기 위해 IP 주소를 사용하는 것처럼, 전송 계층에서는 애플리케이션을 식별하기 위해 **포트 번호**를 사용합니다.

글, 그림 참조:
https://better-together.tistory.com/134

포트 번호의 종류

포트 번호는 **16비트**로 2의 16승 개의 번호를 사용할 수 있습니다.
IP 주소를 관리하는 ICANN은 다음과 같이 포트 번호를 크게 세 종류로 구분하였습니다.

서버 애플리케이션은 정해진 포트 번호를 할당하지만,
클라이언트 애플리케이션은 정해진 포트 번호 없이 포트 번호가 필요할 때마다 동적으로 할당됩니다. 통신할 때 운영체제에 의해 동적 포트 번호 범위 내에서 사용되지 않고 있는 임의의 번호가 자동으로 할당됩니다.
서버 애플리케이션에 할당된 번호가 **웰 노운 포트와 등록된 포트**이고, **클라이언트** 애플리케이션에 배정된 번호가 **동적 포트**입니다. 동적 포트는 통신이 시작되기 전까지 애플리케이션이 어떤 포트를 사용할 지 알 수 없으며, 사용하는 포트도 통신할 때마다 달라집니다.

글, 그림 참조:
https://better-together.tistory.com/134

Port

포트 번호의 종류

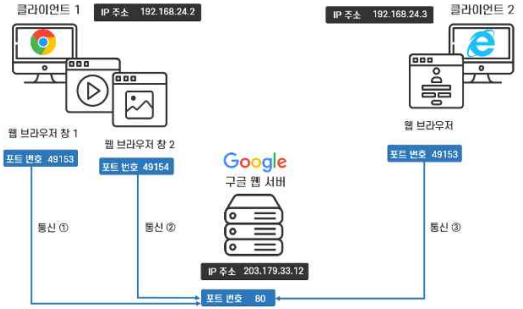
반드시 정해진 포트 번호를 사용해야 한다는 절대적인 규칙이 있는 것은 아니지만, 웰 노운 포트 번호를 다른 용도로 사용하면 혼란을 야기할 수 있으니, 특별한 이유가 없다면 정해진 용도 외에는 사용하지 않는 것이 좋습니다.
대표적으로 HTTP 프로토콜을 사용하는 웹 서비스는 포트 80번을 사용합니다.

포트 번호	서버 애플리케이션
20번	FTP (파일전송)
22번	SSH (원격제어, 보안 기능 추가)
23번	Telent (원격제어)
25번	SMTP (이메일 전송)
80번	HTTP (웹)
110번	POP3 (이메일 수신)
143번	IMAP4 (이메일 수신, 보관 기능 추가)
443번	HTTPS (웹, 보안 기능 추가)

글, 그림 참조:
https://better-together.tistory.com/134

Port

IP 주소와 포트 번호의 조합으로 통신 상대방 식별



IP 주소와 포트 번호의 조합으로 애플리케이션이 통신하는 상대방을 식별할 수 있기 때문에 한 클라이언트가 여러 서버에 접속하더라도, 한 서버에 여러 클라이언트가 접속하더라도 혼선없이 통신이 가능합니다.

글, 그림 참조:
https://better-together.tistory.com/134

전송 계층의 통신 방식과 프로토콜

전송 계층은 수신지까지 문제없이(정확성, 신뢰성) 효율적으로 데이터를 전송해야합니다.
전송 계층의 통신 방식은 두가지로 나뉘고 그에 따른 프로토콜도 두 가지로 구분됩니다.

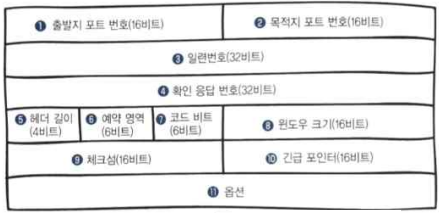
1. **연결형 통신** **>> TCP**
- 신뢰할 수 있고 정확한 데이터를 전달하는 통신입니다.
- 통신 상대방과 확인해 가면서 통신합니다.
- 신뢰성과 정확성이 우선
- 데이터의 정확성과 신뢰성이 중요한 웹이나 이메일에서 사용

2. **비연결형 통신** **>> UDP**
- 효율적으로 데이터를 전송하는 통신입니다.
- 상대방의 확인을 거치지 않고 일방적으로 데이터를 전송하는 방식입니다.
- 효율성이 우선
- 동영상 같이 빠른 데이터 전송으로 원활한 동영상 재생이 중요한 경우 사용

글, 그림 참조:
https://better-together.tistory.com/134

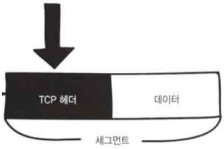
TCP (Transmission Control Protocol, 전송 제어 프로토콜)

전송 계층에서 신뢰할 수 있는 정확한 통신을 제공하는, **신뢰성/정확성**이 우선인 **연결형 통신 프로토콜**입니다.



세그먼트: TCP 헤더가 붙은 데이터

TCP 헤더: TCP로 전송할 때 붙히는 헤더,
목적지까지 데이터를 제대로 전송하기 위해
필요한 정보 포함

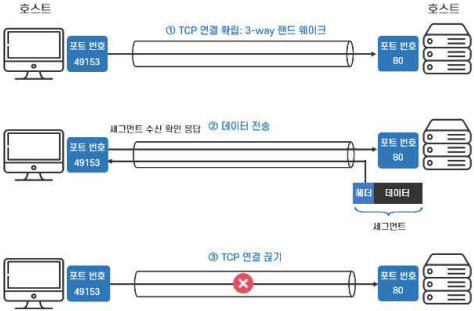


글, 그림 참조:
https://velog.io/@april_5/%EB%84%A4%ED%8A%B8%EC%9B%8C%ED%81%AC-%EA%B5%AC%EC%A1%B0-%EC%9D%84%ED%95%B4-%EC%A0%84%EC%86%A1-%EA%B3%84%EC%B8%B5-%EC%8B%A0%EB%A2%B0%ED%95%A0-%EC%88%98-%EC%9E%88%EB%8A%94-%EB%8D%B0%EC%9D%B4%ED%84%B0-%EC%A0%84%EC%86%A1%ED%95%98%EA%B8%B0

TCP (Transmission Control Protocol, 전송 제어 프로토콜)

TCP 통신 과정:

1. 정상적인 통신이 가능한 상태인지를 확인하는 **연결 확립**
2. **데이터 전송**
3. **연결을 해제**



글, 그림 참조:
https://better-together.tistory.com/134

TCP (Transmission Control Protocol, 전송 제어 프로토콜)

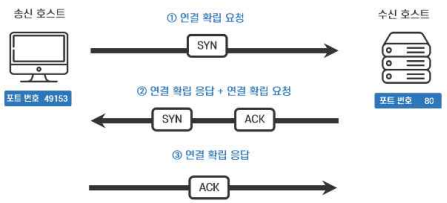
URG	ACK	PSH	RST	SYN	FIN
0	0	0	0	0	0

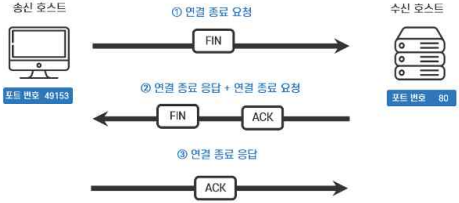
코드 비트 / TCP Flag:

- TCP 헤더의 연결 제어 정보가 기록되고 6비트 구성, 초기값은 0
- **SYN**(Synchronization) : 연결 요청 플래그
- **ACK**(Acknowledgement) : 응답
- RST(Reset) : 제 연결 종료, 비정상적인 세션 연결 끊기
- PSH(Push) : 밀어넣기, 데이터를 바로 7계층의 응용프로그램으로 바로 전달
- URG(Urgent) : 긴급 데이터
- **FIN**(Finish) : 연결 종료 요청

글, 그림 참조:
https://mindgear.tistory.com/206

TCP (Transmission Control Protocol, 전송 제어 프로토콜)





- 연결을 확립하려면 SYN(연결 요청)과 ACK(확인 응답)가 필요
- 연결을 해제하려면 FIN(연결 종료 요청)과 ACK(확인 응답)가 필요
- 신뢰할 수 있는 연결을 하려면 데이터를 전송하기 전에 패킷 교환을 세 번(3-Way Handshake) 확인 한다.

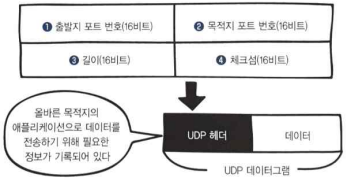
글, 그림 참조:
https://better-together.tistory.com/134

UDP (User Datagram Protocol)

사용자 다이어그램 프로토콜, 전송 계층에서 데이터를 효율적이고 빠르게 보낼 때 사용되는 프로토콜

- 비연결형 통신, 시간이 걸리는 확인 작업을 일일이 하지 않음
- 효율성 중시, TCP와 같은 신뢰성과 정확성을 요구하게 되면 효율 저하
- 데이터를 효율적으로 빠르게 보내는 것이 장점
- 스트리밍 방식으로 전송하는 동영상 서비스와 같은 곳에 적합

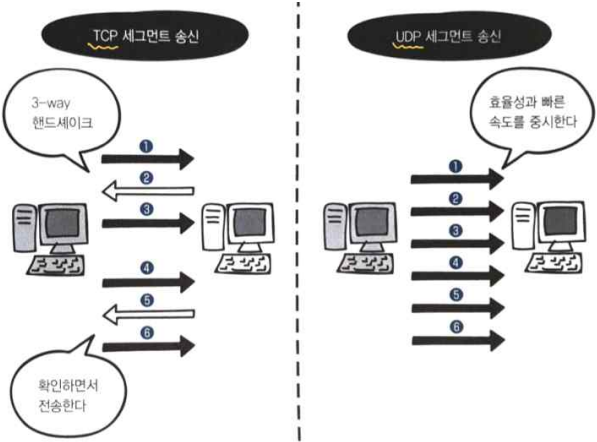
UDP에서는 UDP헤더가 붙은 데이터를 **UDP 데이터 그램**이라고 합니다.



글, 그림 참조:
https://velog.io/@april_5/%EB%84%A4%ED%8A%B8%EC%9B%8C%ED%81%AC-%EA%B5%AC%EC%A1%B0-%EC%9D%B4%ED%95%B4-%EC%A0%84%EC%B6%A1-%EA%B3%84%EC%B8%B5-%EC%8B%A0%EB%A2%B0%ED%95%A0-%EC%88%98-%EC%9E%88%EB%8A%94-%EB%8D%B0%EC%9D%B4%ED%84%B0-%EC%A0%84%EC%86%A1%ED%95%98%EA%B8%B0

TCP & UDP

정리



글, 그림 참조:
https://velog.io/@april_5/%EB%84%A4%ED%8A%B8%EC%9B%8C%ED%81%AC-%EA%B5%AC%EC%A1%B0-%EC%9D%84%ED%95%B4-%EC%A0%84%EC%B6%A1-%EA%B3%84%EC%B8%B5-%EC%8B%A0%EB%A2%B0%ED%95%A0-%EC%88%98-%EC%9E%88%EB%8A%94-%EB%8D%B0%EC%9D%B4%ED%84%B0-%EC%A0%84%EC%86%A1%ED%95%98%EA%B8%B0

감사합니다