

2. 네트워크

▼ 목차

2.1 네트워크의 기초

2.1.1 처리량과 지연 시간

2.1.2 네트워크 토폴로지와 병목 현상

2.1.3 네트워크 분류

2.1.4 네트워크 성능 분석 명령어

2.1.5 네트워크 프로토콜 표준화

2.2 TCP/IP 4계층 모델

2.2.1 OSI 7계층 모델과 TCP/IP 4계층 모델

2.2.2 TCP/IP 4계층 모델의 계층 구조

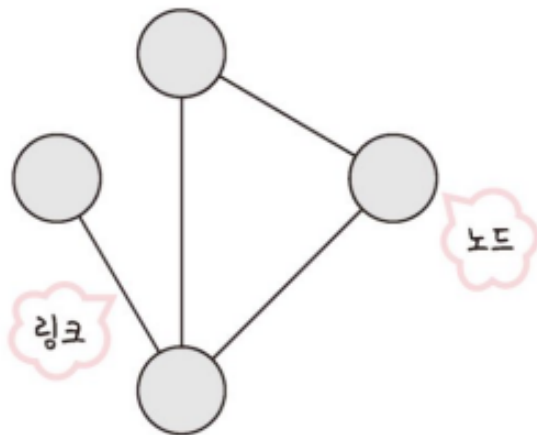
2.2.3 PDU

▼ memo

가상 회선 패킷 교환 방식, 데이터그램 패킷 교환 방식

전송계층 → 캡슐화 비캡슐화

2.1 네트워크의 기초



네트워크란 노드와 링크가 서로 연결되어 있거나 연결되어 있지 않은 집합체를 의미

노드 - 서버, 라우터, 스위치 등 네트워크 장치

링크 - 유선, 무선

2.1.1 처리량과 지연 시간

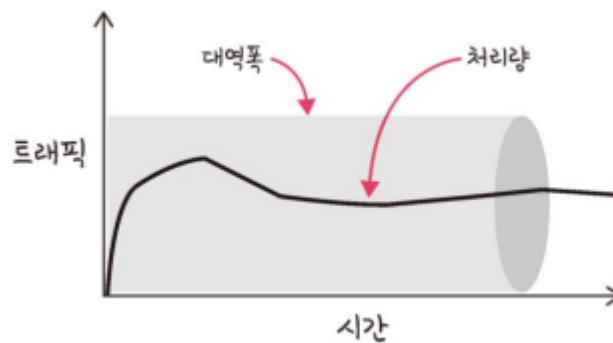
네트워크를 구축할 때에는 좋은 네트워크로 만드는 것이 중요

→ 좋은 네트워크란

많은 처리량, 지연 시간이 짧고, 장애 빈도가 적고, 좋은 보안

<처리량>

링크를 통해 전달되는 단위 시간당 데이터의 양

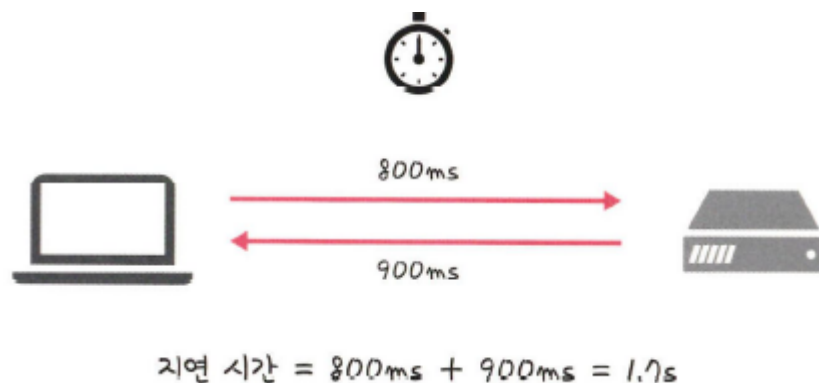


단위 : bps(bits per second)_초당 전송 또는 수신되는 비트 수

처리량은 사용자들이 많이 접속할 때마다 커지는 트래픽, 네트워크 장치 간의 대역폭, 네트워크 중간에 발생하는 에러, 장치의 하드웨어 스펙에 영향을 받는다

- 대역폭 : 주어진 시간 동안 네트워크 연결을 통해 흐를 수 있는 최대 비트 수

<지연시간>



요청이 처리되는 시간, 어떤 메시지가 두 장치 사이를 왕복하는데 걸리는 시간

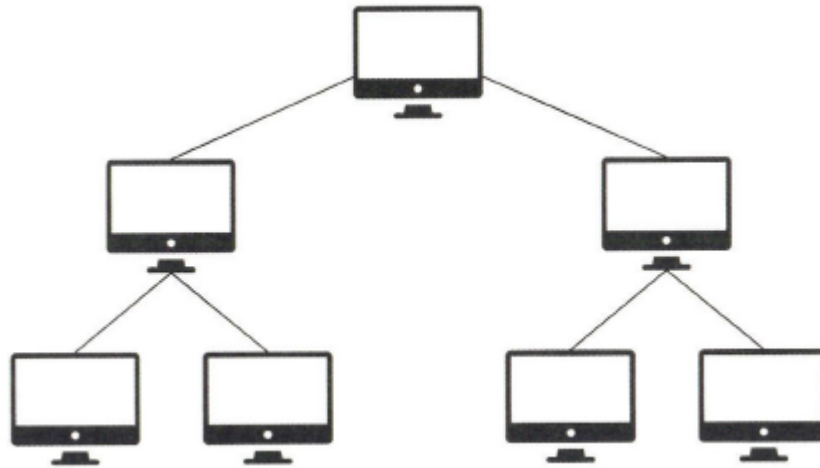
지연 시간은 매체 타입(무선, 유선) 패킷 크기, 라우터의 패킷 처리 시간에 영향을 받는다

2.1.2 네트워크 토폴로지와 병목 현상

<네트워크 토폴로지>

네트워크 토폴로지는 노드와 링크가 어떻게 배치되어 있는지에 대한 방식이자 연결 형태를 의미

<트리 토폴로지>

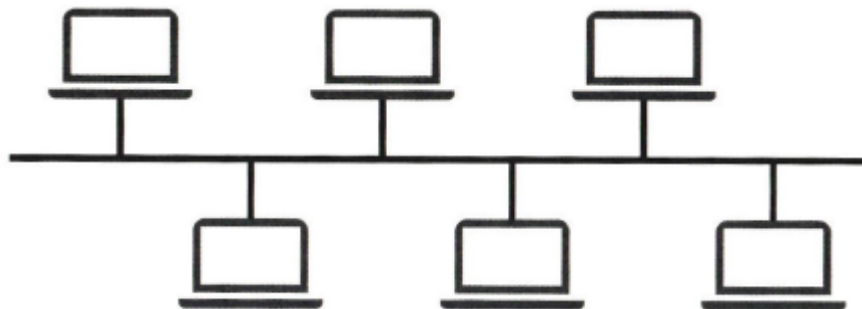


계층형 토폴로지라고 하며 트리 형태로 배치한 네트워크 구성을 말한다

노드 추가 삭제가 쉽다

특정 노드에 트래픽이 집중될 때 하위 노드에 영향을 끼칠 수 있다

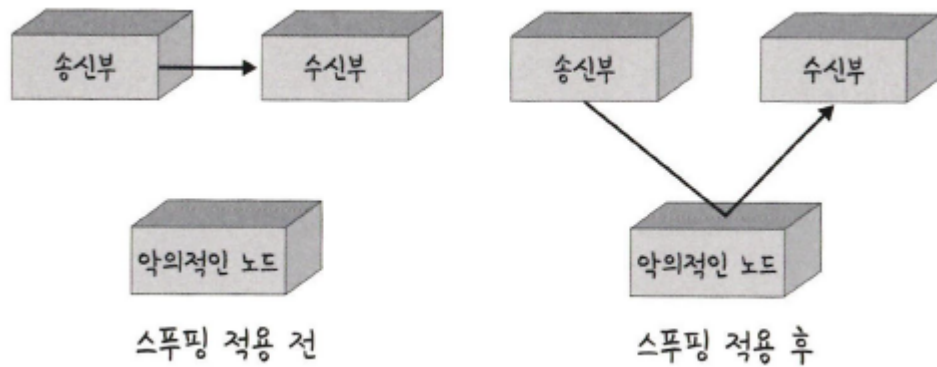
<버스 토폴로지>



버스 토폴로지는 **중앙 통신 회선 하나에 여러 개의 노드가 연결**되어 공유하는 네트워크 구성
근거리 통신망 LAN에서 사용한다

설치 비용이 적고 신뢰성이 우수하며, 중앙 통신 회선에 노드를 추가하거나 삭제하기 쉽다
그러나 스푸핑이 가능한 문제가 있다

⇒ 스푸핑



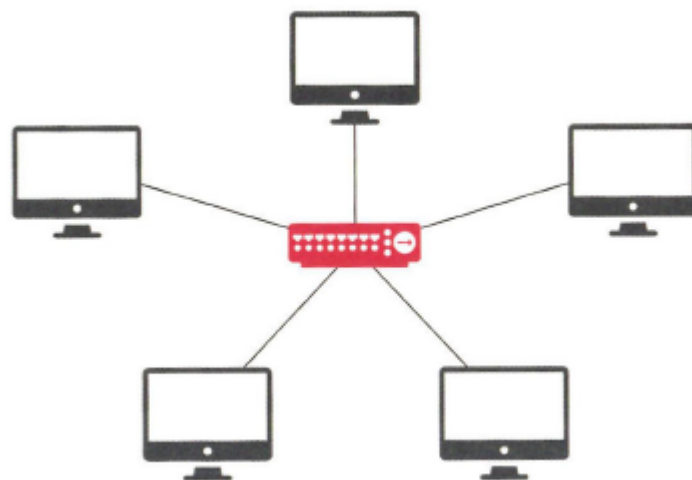
스푸핑은 LAN 상에서 송신부의 패킷을 송신과 관련 없는 다른 호스트에 가지 않도록 하는 스위칭 기능을 마비시키거나 속여서 특정 노드에 해당 패킷이 오도록 처리하는 것

버스 토폴로지는 여러 대의 컴퓨터가 한 개의 케이블에 연결되어 있는 형태의 네트워크 구성 방식입니다. 이 경우, 통신은 브로드캐스트 방식으로 이루어지기 때문에 컴퓨터마다 자신의 주소가 필요하지 않습니다. 하지만 이러한 구성 방식 때문에 모든 컴퓨터가 같은 케이블을 사용하기 때문에, 공격자는 이를 이용하여 스푸핑을 할 수 있습니다.

버스 토폴로지 스푸핑은 주로 이더넷(Ethernet)에서 발생합니다. 공격자는 이더넷 케이블을 가로채서 다른 컴퓨터의 패킷을 가로챈 다음, 이를 위조하여 다른 컴퓨터로 보내거나, 중간에서 변조하여 정보를 탈취하거나 변조할 수 있습니다.

버스 토폴로지 스푸핑을 방지하기 위해서는 네트워크에 보안 장비를 설치하거나, 가상 개인 네트워크(VPN)를 사용하여 통신을 암호화하거나, 다른 네트워크 구성 방식을 사용하는 것이 좋습니다.

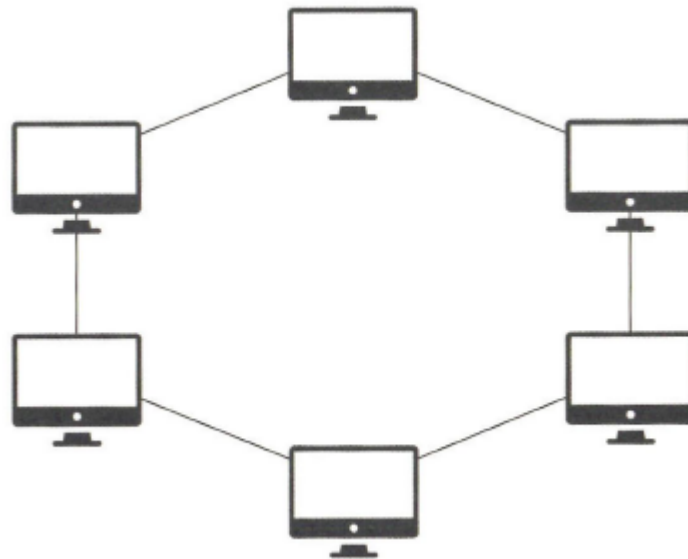
<스타 토폴로지>



스타 토폴로지는 중앙에 있는 노드에 모두 연결된 네트워크 구성을 말한다

노드를 추가하거나 에러를 탐지하기 쉽고, **패킷의 충돌 발생 가능성이 적다**. 또 어떠한 노드에 장애가 발생해도 쉽게 에러를 발견할 수 있으며, 장애 노드가 중앙 노드가 아닐 경우 다른 노드에 영향을 끼치는 것이 적다. 하지만 중앙 노드에 장애가 발생하면 전체 네트워크를 사용할 수 없고 설치 비용이 크다

<링형 토폴로지>

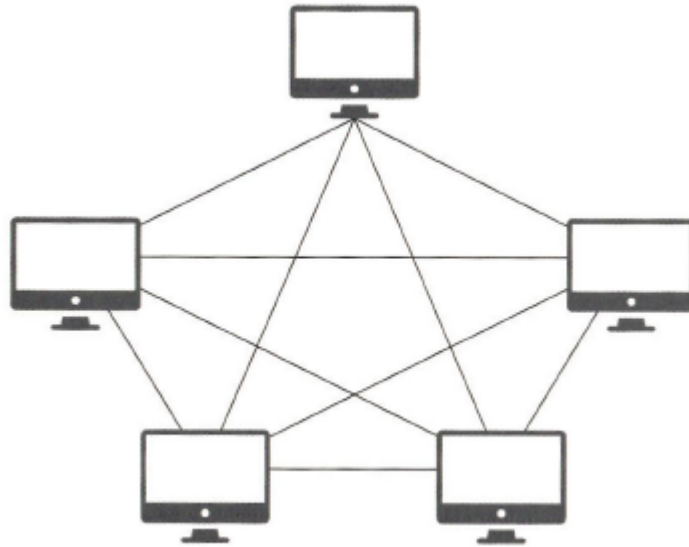


링형 토폴로지는 각각의 노드가 양 옆의 두 노드와 연결하여 전체적으로 고리처럼 하나의 연속된 길을 통해 통신을 하는 망 구성 방식

데이터는 노드로 이동을 하게 되며, 각각의 노드는 고리 모양의 길을 통해 패킷을 처리

노드 수가 증가되어도 네트워크 상의 손실이 거의 없고 충돌이 발생하는 가능성이 적고 노드의 고장 발견을 쉽게 찾을 수 있다. 하지만 네트워크 구성 변경이 어렵고 회선에 장애가 발생하면 전체 네트워크에 영향을 크게 끼치는 단점이 있다

<메시 토폴로지>



메시 토폴로지는 망형 토폴로지라고도 하며 그물망처럼 연결되어 있는 구조

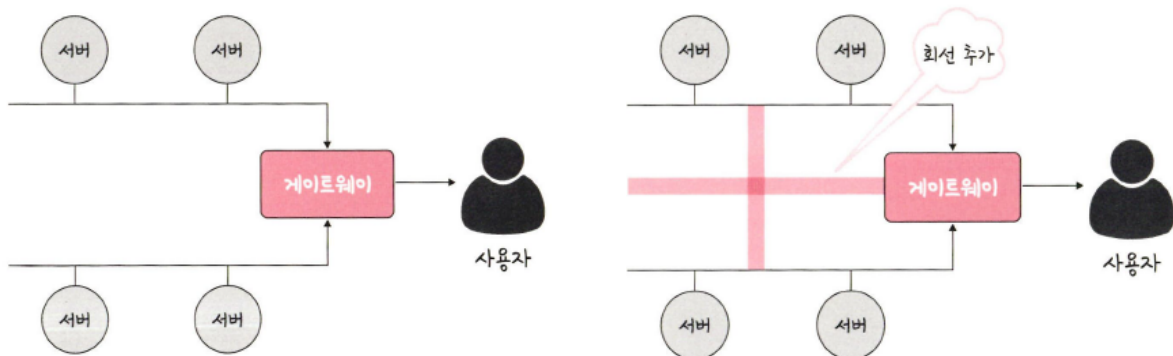
한 단말 장치에 장애가 발생해도 여러 개의 경로가 존재하므로 네트워크를 계속 사용할 수 있고 트래픽도 분산 처리가 가능하다. 하지만 노드의 추가가 어렵고 구축 비용과 운용 비용이 고가인 단점

<네트워크 병목현상>

네트워크 토폴로지가 중요한 이유는 병목 현상을 찾을 때, 중요한 기준이 되기 때문

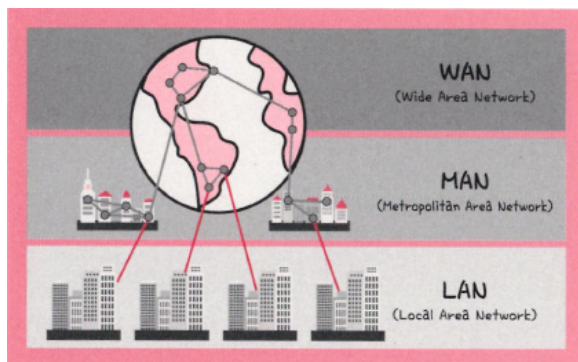
네트워크 병목 현상(Network bottleneck)은 **네트워크에서 데이터 전송 속도가 느려지거나 중단되는 현상**을 말합니다. 이러한 현상은 네트워크에서 전송할 수 있는 대역폭(bandwidth)이 한정되어 있거나, 데이터 전송 요청이 너무 많아서 네트워크 장비가 처리할 수 없는 경우 등 다양한 이유로 발생할 수 있습니다.

이러한 병목 현상이 발생하면 데이터 전송이 지연되거나 속도가 느려져서 전송 시간이 오래 걸리게 되고, 더 많은 데이터를 전송하기 위해서는 대역폭을 늘리거나 네트워크 구성을 최적화해야 합니다. 따라서 네트워크 관리자는 병목 현상을 예방하고 해결하기 위한 다양한 기술과 방법을 사용하여 네트워크의 성능을 최적화하고 안정적으로 운영해야 합니다.



이처럼 네트워크의 토폴로지를 알고 있어야 네트워크 병목현상을 해결할 수 있다.

2.1.3 네트워크 분류



LAN - 사무실, 개인적 소유 가능한 규모
MAN - 서울시 등 시 정도의 규모
WAN - 세계 규모

<LAN>

근거리 통신망
건물이나 캠퍼스 같은 좁은
공간에서 운영
전송 속도가 빠르고 혼잡하
지 않다

<MAN>

대도시 지역 네트워크
전송 속도는 평균이며
LAN보다는 더 혼잡

<WAN>

광역 네트워크
국가 또는 대륙 같은 넓은 지
역에서 운영
전송 속도는 낮으며 가장 혼
잡

2.1.4 네트워크 성능 분석 명령어

애플리케이션 코드상에는 전혀 문제가 없는데 사용자가 서비스로부터 데이터를 가져오지 못하는 상황이 발생되기도 하며, 이는 네트워크 병목 현상일 가능성이 있다.

<네트워크 병목 현상의 주된 원인 4가지>

- 네트워크 대역폭
- 네트워크 토폴로지
- 서버 CPU, 메모리 사용량
- 비효율적인 네트워크 구성

이때는 네트워크 관련 테스트와 네트워크와 무관한 테스트를 통해 네트워크로 부터 발생한 문제인지 확인하고 네트워크 성능 분석을 해봐야 한다. 이때 사용하는 명령어들에 대해 소개

<ping>

ping_(packet INternet Groper)은 네트워크 상태를 확인하려는 대상 노드를 향해 일정 크기의 패킷을 전송하는 명령어

이를 통해 해당 노드의 패킷 수신 상태와 도달하기까지 시간 등을 알 수 있으며 해당 노드까지 네트워크가 잘 연결되어 있는지 확인할 수 있습니다.

```
C:\Users\jhc>ping www.google.com -n 12

Ping www.google.com [172.217.26.228] 32바이트 데이터 사용:
172.217.26.228의 : 바이트=32 시간=56ms TTL=117
172.217.26.228의 : 바이트=32 시간=56ms TTL=117
172.217.26.228의 : 바이트=32 시간=57ms TTL=117
172.217.26.228의 : 바이트=32 시간=56ms TTL=117
172.217.26.228의 : 바이트=32 시간=56ms TTL=117
172.217.26.228의 : 바이트=32 시간=56ms TTL=117
172.217.26.228의 : 바이트=32 시간=56ms TTL=117
172.217.26.228의 : 바이트=32 시간=56ms TTL=117
172.217.26.228의 : 바이트=32 시간=57ms TTL=117
172.217.26.228의 : 바이트=32 시간=57ms TTL=117
172.217.26.228의 : 바이트=32 시간=56ms TTL=117
172.217.26.228의 : 바이트=32 시간=56ms TTL=117

172.217.26.228에 대한 Ping 통계:
   패킷: 보냄 = 12, 받음 = 12, 손실 = 0 (0% 손실),
   왕복 시간(밀리초):
     최소 = 56ms, 최대 = 57ms, 평균 = 56ms
```

ping www.google.com -n 12 라는 명령어를 구동한 것

-n 12 옵션을 넣어서 12번의 패킷을 보내고 12번의 패킷을 받는 모습

<netstat>

접속되어 있는 서비스들의 네트워크 상태를 표시하는데 사용되며 네트워크 접속, 라우팅 테이블, 네트워크 프로토콜 등 리스트를 보여준다. 주로 **서비스의 포트가 열려 있는지 확인할 때 사용**

```
C:\Users\jhc>netstat

활성 연결

프로토콜  로컬 주소          외부 주소          상태
TCP       121.165.224.223:6881  220.118.188.195:41519 TIME_WAIT
TCP       121.165.224.223:49245 211.115.106.72:http CLOSE_WAIT
TCP       121.165.224.223:50124 nrt12s51-in-f19:https ESTABLISHED
TCP       121.165.224.223:50278 118.223.101.233:56517 ESTABLISHED
TCP       121.165.224.223:52025 211.115.106.207:http CLOSE_WAIT
TCP       121.165.224.223:52042 211.115.106.207:http CLOSE_WAIT
TCP       121.165.224.223:52043 211.115.106.207:http CLOSE_WAIT
TCP       121.165.224.223:52220 211.249.220.83:https ESTABLISHED
TCP       121.165.224.223:52221 104.21.37.168:http ESTABLISHED
TCP       121.165.224.223:52243 a104-74-192-17:http TIME_WAIT
```

현재 접속하고 있는 사이트 등에 관한 네트워크 상태 리스트를 볼 수 있다

<nslookup>

DNS(Domain Name System)에 관련된 내용을 확인하기 위해 쓰는 명령어. 특정 도메인에 매핑된 IP를 확인하기 위해 사용

google의 DNS를 확인하는 것


```
C:\Users\jhc>nslookup
기본 서버:  kns.kornet.net
Address:  168.126.63.1

> google.com
서버:  kns.kornet.net
Address:  168.126.63.1

권한 없는 응답:
이름:  google.com
Addresses:  2404:6800:4004:820::200e
           172.217.31.174
```

<tracert>

목적지 노드까지 네트워크 경로를 확인할 때 사용하는 명령어. 목적지 노드까지 구간들 중 어느 구간에서 응답 시간이 느려지는지 등을 확인할 수 있다

```
C:\Users\jhc>tracert www.google.com

최대 30홉 이상의
www.google.com [142.250.199.100](으)로 가는 경로 추적:

  1  1 ms  *      *      121.165.224.254
  2  1 ms  1 ms   1 ms   61.78.42.172
  3  2 ms  2 ms   1 ms   112.189.31.209
  4  *      *      *      요청 시간이 만료되었습니다.
  5  1 ms  2 ms   1 ms   112.174.47.102
  6  41 ms 40 ms  41 ms   72.14.209.102
  7  36 ms 36 ms  37 ms   108.170.241.80
  8  43 ms 43 ms  41 ms   216.239.62.240
  9  53 ms 53 ms  53 ms   172.253.50.221
 10  56 ms 56 ms  56 ms   216.239.49.194
```

구글 사이트에 도달하기까지의 경로 추적하는 것

2.1.5 네트워크 프로토콜 표준화

네트워크 프로토콜이란 다른 장치들끼리 데이터를 주고 받기 위해 설정된 공통된 인터페이스를 말한다. 이러한 프로토콜은 IEEE 또는 IETF라는 표준화 단체가 결정한다

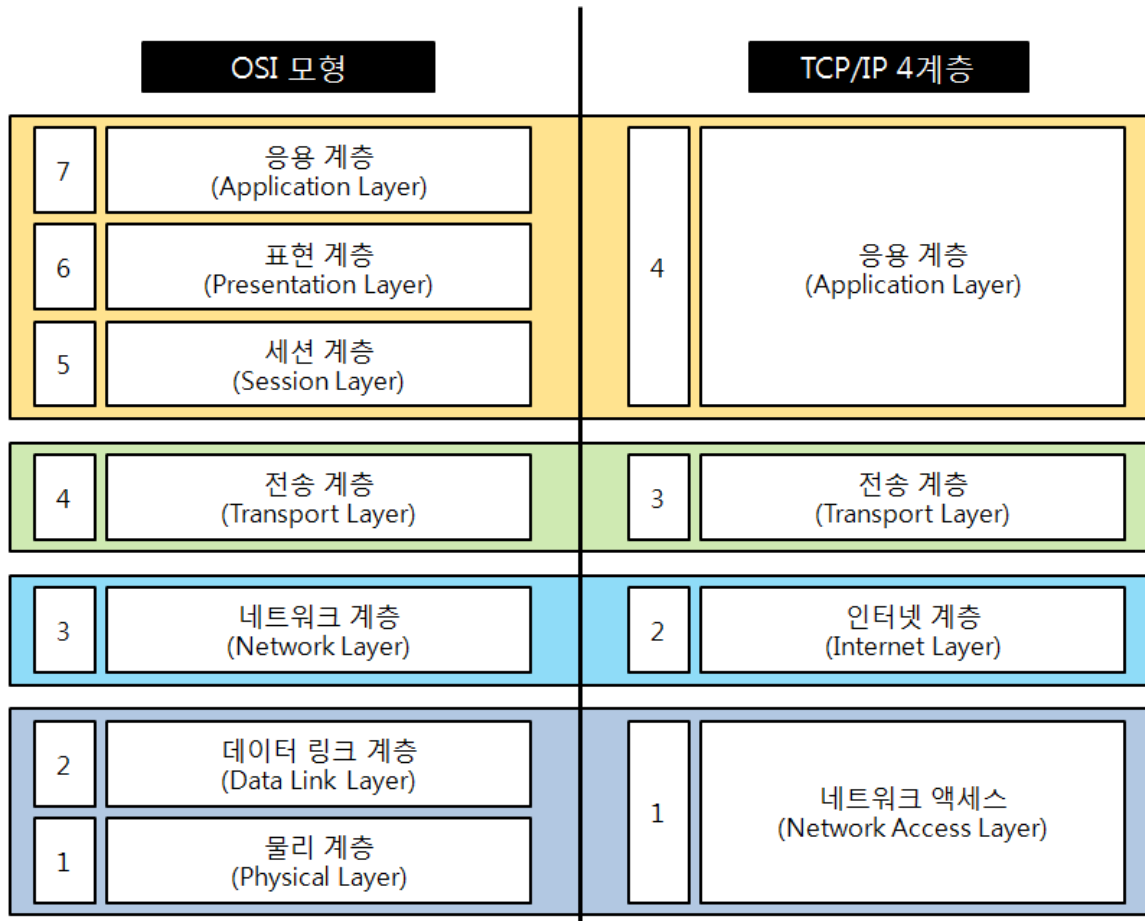
2.2 TCP/IP 4계층 모델

인터넷 프로토콜 스위트 _ (internet protocol suite)

- 인터넷에서 컴퓨터들이 서로 정보를 주고 받는 데 쓰이는 프로토콜의 집합
- 이를 OSI 7계층 모델이나, TCP/IP 4계층 모델로 설명한다.

⇒ 이렇게 계층을 나눈 이유는, 통신이 일어나는 과정을 단계별로 확인할 수 있으며, 특정 계층에서 문제가 발생할 경우 해당 계층만 핸들하면 되기 때문이다.

2.2.1 OSI 7계층 모델과 TCP/IP 4계층 모델



우선 **OSI 7계층 모델**은 국제 표준화기구에서 개발한 모델이다

네트워크 프로토콜 디자인과 데이터 통신 계층으로 나누어 표준화 했다

TCP/IP 4계층 모델은 OSI 7계층 모델을 좀 더 논리적으로 병합하여 축약한 것이다.

- **TCP(상위계층)** : 메세지나 파일을 작은 패킷으로 나누거나 재조립하여 송수신에 반영하는 담당
- **IP(하위계층)** : 각 패킷의 주소 부분들을 처리하여 패킷들이 목적지로 정확히 송수신되도록 기능함

보통 상위 계층은 소프트웨어로 구성되고, 하위 계층은 하드웨어로 구성된다.

2.2.2 TCP/IP 4계층 모델의 계층 구조

TCP/IP 4계층	역할	데이터 단위	전송 주소	예시	장비
응용 계층 (Application)	응용프로그램 간의 데이터 송수신	Data/M essage	-	파일 전송, 이메일, FTP, HTTP, SSH, Telnet, DNS, SMTP 등	-
전송 계층 (Transport)	호스트 간의 자료 송수신	Segmen t	Port	TCP, UDP, RTP, RTCP 등	게이트 웨이
인터넷 계층 (Internet)	데이터 전송을 위한 논리적 주소 지정 및 경로 지정	Packet	IP	IP, ARP, ICMP, RARP, OSPF	라우터
네트워크 연결 계층 (Network Access)	실제 데이터인 프레임을 송수신	Frame	MAC	Ethernet, PPP, Token Ring 등	브리지, 스위치

1. <애플리케이션 계층>

데이터 단위: Data/Message

- 사용자와 가장 가까운 계층으로 **사용자가 소프트웨어 application과 소통할 수 있게 해준다**
- 응용프로그램(application)들이 데이터를 교환하기 위해 사용되는 프로토콜
- 사용자 응용프로그램 인터페이스를 담당

예시

파일 전송, 이메일, FTP, HTTP, SSH, Telnet, DNS, SMTP 등

<사용자가 직접 사용하는 프로토콜>

프로토콜	동작 방식
HTTP	웹 클라이언트와 웹 서버 사이에서 웹 페이지 데이터를 주고 받는다.
POP, SMTP, IMAP	메일을 송수신하고 보관한다.
SMB/AFP	LAN 안에서 파일을 공유한다.

<사용자가 간접적으로 사용하는 프로토콜>

프로토콜	동작 방식
DNS	도메인명과 IP어드레스의 정보를 서로 변환할때 사용한다
DHCP	LAN 내의 컴퓨터에게 IP 어드레스를 할당할 때 사용한다
SSL/TLS	통신 데이터를 암호화하여 주요 정보를 안전하게 주고받을 때 사용한다

2. <전송 계층>

데이터 단위: Segment 전송 주소: Port

- 송신자와 수신자를 연결하는 통신 서비스를 제공
연결 지향 데이터 스트림 지원, 신뢰성, 흐름 제어 제공
애플리케이션과 인터넷 계층 사이의 데이터가 전달 될 때의 중계 역할을 한다
- 통신 노드 간의 연결 제어 및 자료 송수신을 담당
- 세그먼트 (Segment)단위의 데이터 구성
 - 실질적인 데이터 전송을 위해 데이터를 일정 크기로 나눈 것. 발신, 수신, 포트주소, 오류검출코드가 붙게된다

<전송 계층의 두 가지 역할>

- ① 데이터가 제대로 도착했는지 확인
- ② 전송된 데이터의 목적지가 어떤 애플리케이션인지 식별

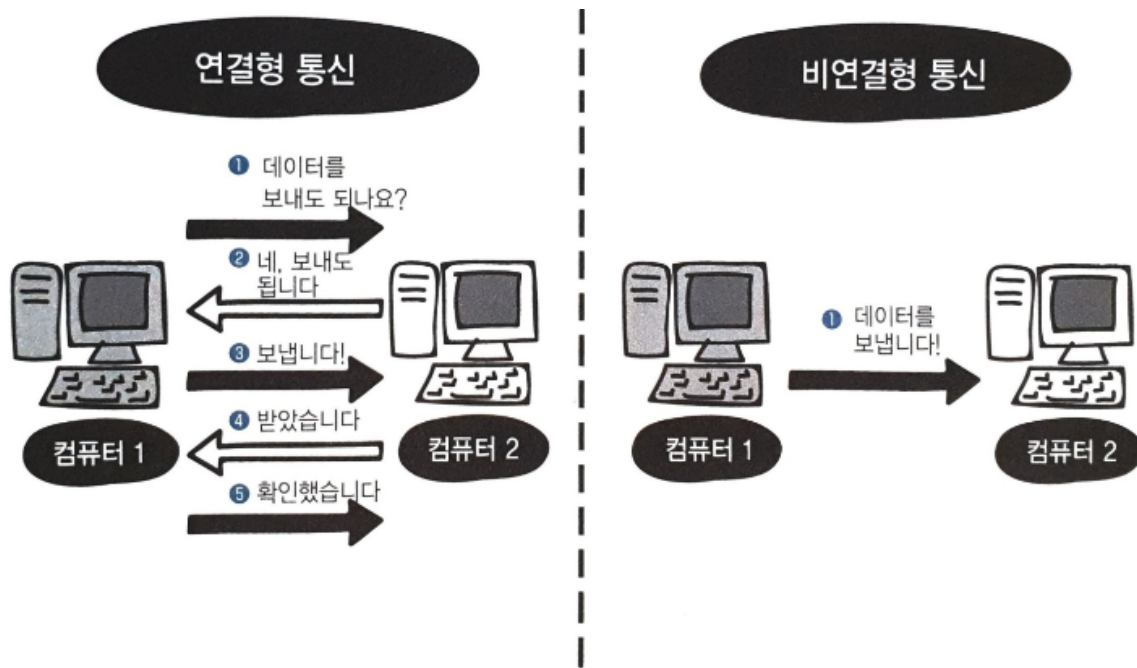
예시

TCP, UDP, RTP, RTCP 등

<연결형 통신과 비연결형 통신>

전송 계층의 특징을 간단히 요약하자면 신뢰성, 효율성으로 구분 가능하다.

- 신뢰성 : 데이터를 목적지에 문제 없이 전달 → 연결형 통신 (TCP)
 - 여러 번 확인하고 보내는, 상대방과 확인해가며 통신하는 방식
- 효율성 : 데이터를 빠르고 효율적으로 전달 → 비연결형 통신 (UDP)
 - 효율성이 우선인 통신이므로 확인 절차 없이 일방적으로 보내는, 상대방을 확인하지 않고 일방적으로 데이터를 전송하는 방식
 - 예시) 동영상: 데이터가 늦게 도착해서 화면이 버벅거리는 것 보다, 데이터가 약간 유실되더라도 원활하게 보는 것이 좋으니까!

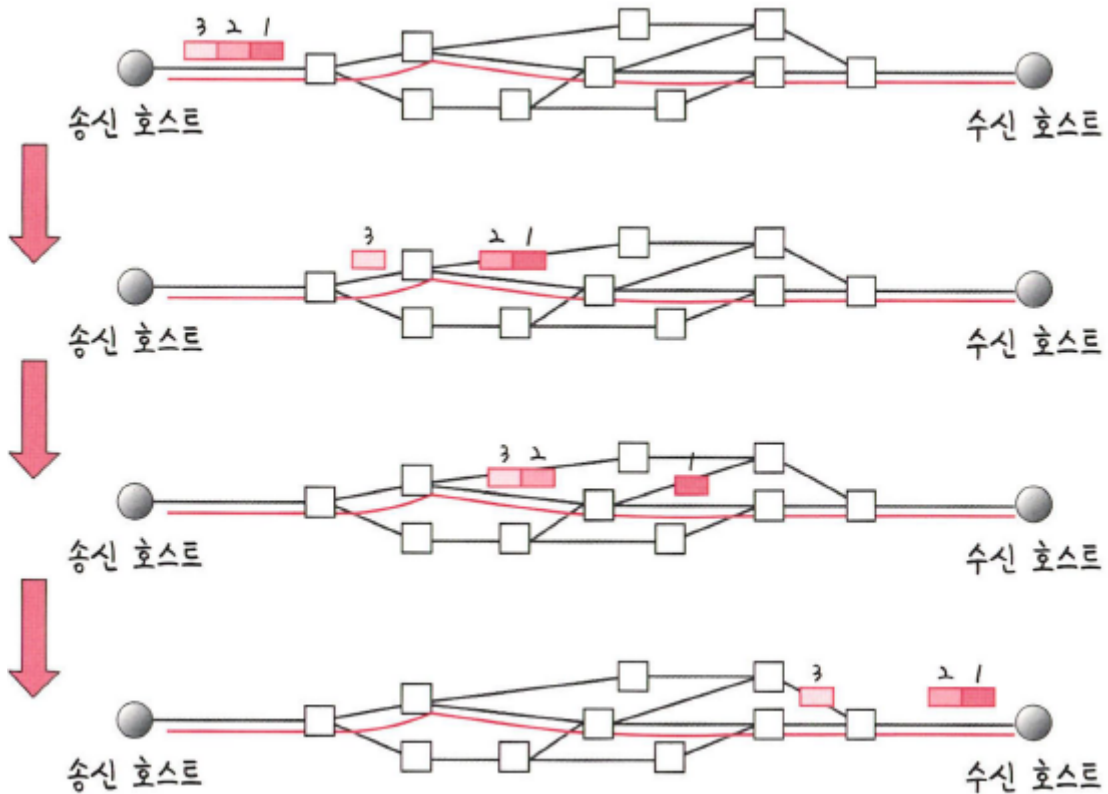


<TCP>

- 패킷 사이의 순서를 보장
- 연결 지향 프로토콜을 사용해서 연결을 하여 **신뢰성을 구축**해서 수신 여부를 확인
- '가상 회선 패킷 교환 방식' 사용

<TCP> _ 가상 회선 패킷 교환 방식

각 패킷에는 가상회선 식별자가 포함되며 모든 패킷을 전송하면 가상 회선이 해제되고 패킷들은 전송되는 순서대로 도착하는 방식



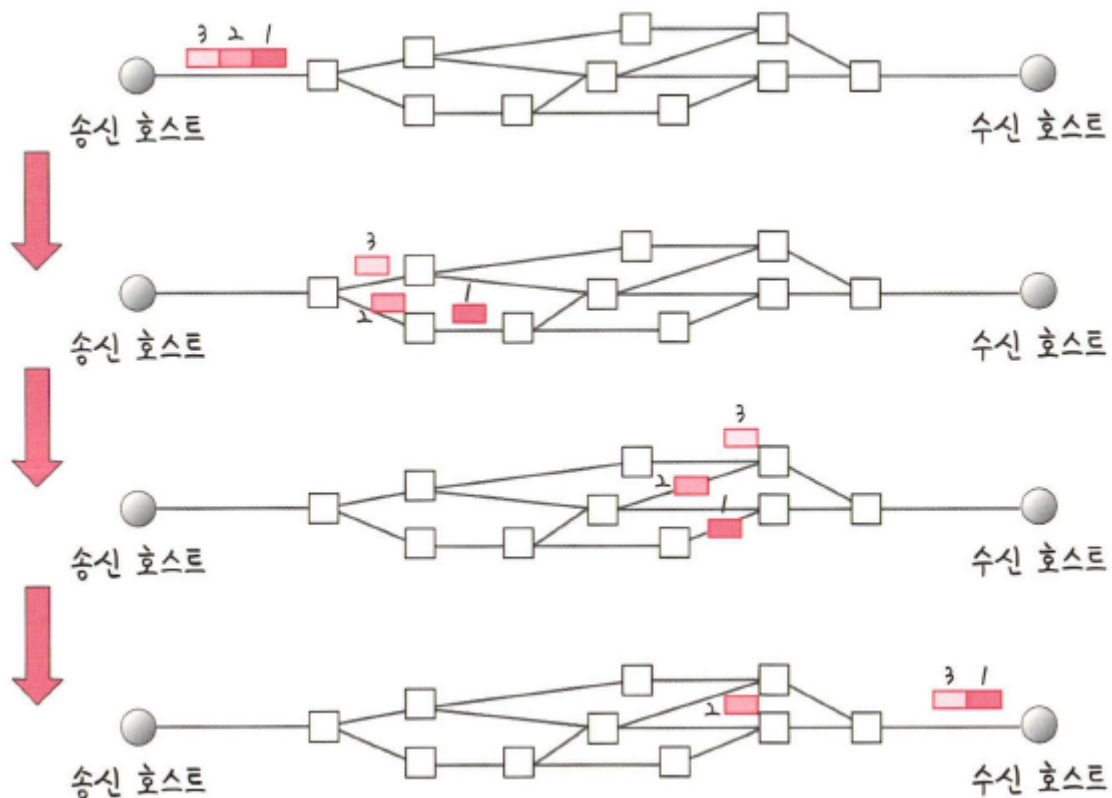
3,2,1 로 이루어진 패킷이 어떠한 회선을 따라 순서대로 도착하는 것인지 알 수 있다

<UDP>

- 순서를 보장하지 않음
- 수신 여부를 확인하지 않음
- **단순히 데이터만 주는** '데이터그램 패킷 교환 방식'을 사용

<UDP> _ 데이터그램 패킷 교환 방식

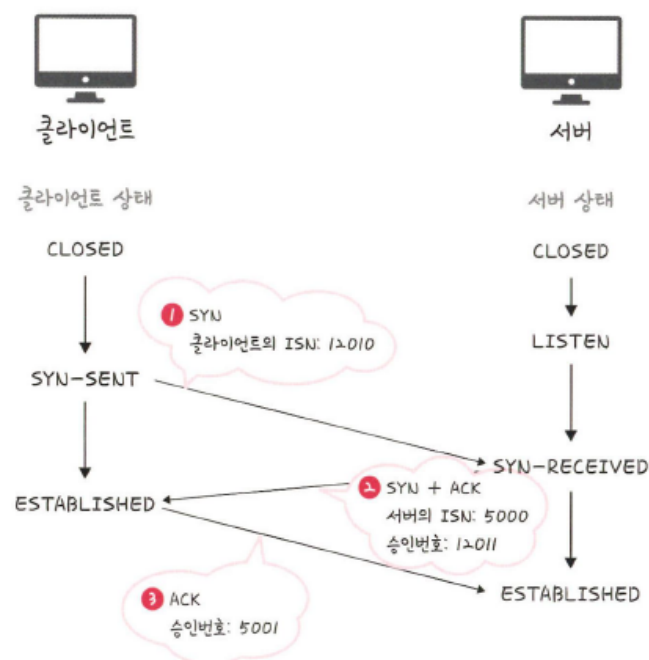
패킷이 독립적으로 이동하며 최적의 경로를 선택하여 가는데, 하나의 메시지에서 분할된 여러 패킷은 서로 다른 경로로 전송될 수 있으며 도착한 순서가 다를 수 있는 방식



3,2,1 패킷이 순서도 다르고 어떠한 회선을 중심으로 가는 것이 아니라 따로따로 이동하며 순서도 다르게 도착

<TCP 연결 성립 과정> _ “3 - 웨이 핸드셰이크”

TCP는 신뢰성을 확보할 때 ‘3 - 웨이 핸드셰이크’라는 작업을 진행



3 - 웨이 핸드셰이크 과정 이후 신뢰성이 구축되고 데이터 전송을 시작한다.

TCP는 이 과정이 있다 → 신뢰성이 있음

UDP는 이 과정이 없다 → 신뢰성이 없음

<클라이언트와 서버가 통신할 때 세 단계의 과정을 거친다>

1. SYN 단계 : 클라이언트가 서버에 연결 요청

클라이언트는 서버에 클라이언트의 ISN을 담아 SYN을 보낸다. ISN은 새로운 TCP 연결의 첫 번째 패킷에 할당된 임의의 시퀀스 번호를 말하며 이는 장치마다 다를 수 있다. (ISN_예시 12010)

2. SYN + ACK 단계 : 서버에서 클라이언트에게 요청 수락 패킷 전송

서버는 클라이언트의 SYN을 수신하고 서버의 ISN을 보내며 승인번호로 클라이언트의 ISN + 1 을 보낸다

3. ACK 단계 : 클라이언트는 서버에 응답 플래그(ACK) 보내고, 연결이 이루어지며 데이터가 오간다

클라이언트는 서버의 ISN + 1한 값인 승인번호를 담아 ACK를 서버에 보낸다

— SYN

SYNchronization의 약자, 연결 요청 플래그

— ACK

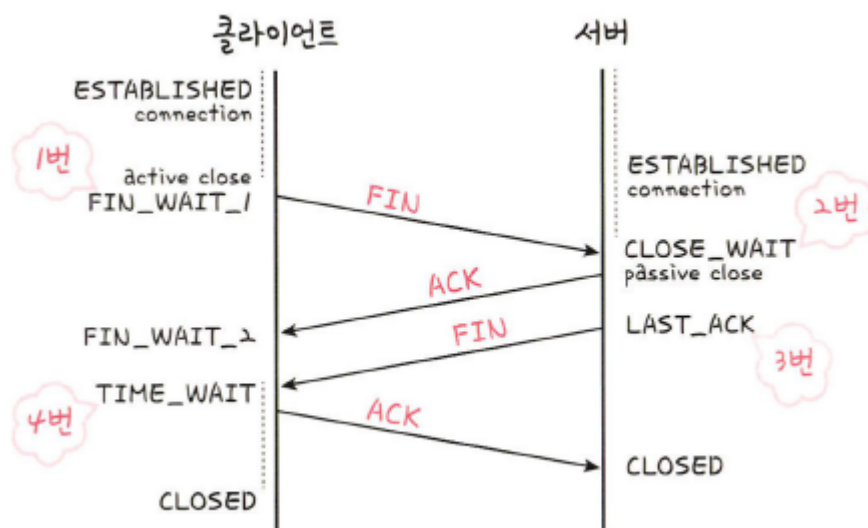
ACKnowledgement의 약자, 응답 플래그

— ISN

Initial Sequence Numbers의 약어, 초기 네트워크 연결을 할 때 할당된 32비트 고유 시퀀스 번호이다.

<TCP 연결 해제 과정>

TCP가 연결을 해제할 때는 4 - 웨이 핸드셰이크 과정이 발생



1. 클라이언트가 연결을 종료하겠다는 FIN 플래그를 전송
2. 서버는 일단 확인 메시지를 보내고 자신의 통신이 끝날 때까지 기다리는데 이 상태가 TIME_WAIT
3. 서버가 통신이 끝났으면 연결이 종료되었다고 클라이언트에게 FIN 플래그 전송
4. 클라이언트는 확인했다는 메시지를 보낸다

2의 과정에서 TIME_WAIT을 진행하는 이유

→ 소켓이 바로 소멸 되지 않고 일정 시간 유지되는 상태를 말한다. CentOS6 우분투에는 60초로 설정되어 있으며 윈도우는 4분으로 설정되어 있음

1. 지연 패킷이 발생할 경우를 대비하기 위함. 패킷이 뒤늦게 도달하고 이를 처리하지 못한다면 데이터 무결성 문제 발생
2. 두 장치가 연결이 닫혔는지 확인하기 위함. 만약 LAST_ACK 상태에서 닫히게 된다면 다시 새로운 연결을 하려고 할 때 장치는 줄곧 LAST_ACK로 되어 있기 때문에 접속 오류가 나타난다.
→ LAST_ACK란 연결은 종료되었고 승인을 기다리는 상태.

이러한 이유들 때문에 TIME_WAIT이라는 시간이 필요

3. <인터넷 계층>

데이터 단위: 패킷전송 주소: IP

- 네트워크상 최종 목적지까지 정확하게 연결되도록 연결성을 제공
- 단말을 구분하기 위해 논리적인 주소(Logical Address) IP를 할당
 - 출발지와 목적지의 논리적 주소가 담겨있는 IP datagram이라는 패킷으로 데이터를 변경
 - 데이터 전송을 위한 주소 지정
- 라우팅(Routing) 기능을 처리
 - 경로 설정
- 최종 목적지까지 정확하게 연결되도록 연결성 제공
- 패킷 단위의 데이터 구성
 - 세그먼트를 목적지까지 전송하기 위해 시작 주소와 목적지의 논리적 주소를 붙인 단위. 데이터 + IP Header

예시

IP, ARP, ICMP, RARP, OSPF

장치로부터 받은 네트워크 패킷을 IP주소로 **지정된 목적지로 전송하기 위해 사용되는 계층** 패킷을 수신해야 할 상대의 주소를 지정하여 데이터를 전달 상대방이 제대로 받았는지에 대해 보장하지 않는 **비연결형적인 특징**을 가진다

4. <링크 계층>

데이터 단위: 프레임 전송 주소: MAC

- 물리적으로 데이터가 네트워크를 통해 어떻게 전송되는지를 정의
 - 논리주소(IP주소 등)이 아닌 물리주소(예. MAC주소(Media Access Control Address))를 참조해 장비간 전송
 - MAC주소란 컴퓨터의 하드웨어 주소
- 기본적으로 에러검출/패킷의 프레임화 담당
- 프레임(Frame)단위의 데이터 구성
 - 최종적으로 데이터 전송을 하기 전 패킷헤더에 MAC주소와 오류 검출을 위한 부분을 첨부

예시

MAC, LAN, 패킷망 등에 사용되는 것에) Ethernet, PPP, Token Ring 등

전선, 광섬유, 무선 등으로 **실질적으로 데이터를 전달하며 장치 간에 신호를 주고 받는 규칙을 정하는 계층**

물리 계층과 데이터 링크 계층으로 나누기도 하는데,
물리 계층은 무선 LAN과 유선 LAN을 통해 0과 1로 이루어진 데이터를 보내는 계층을 말하며,
데이터 링크 계층은 이더넷 프레임을 통해 에러 확인 흐름 제어 접근 제어를 담당하는 계층

<유선 LAN>

유선 LAN을 이루는 **이더넷**은 IEEE802.3이라는 프로토콜을 따르며, **전이중화 통신을 사용한다**

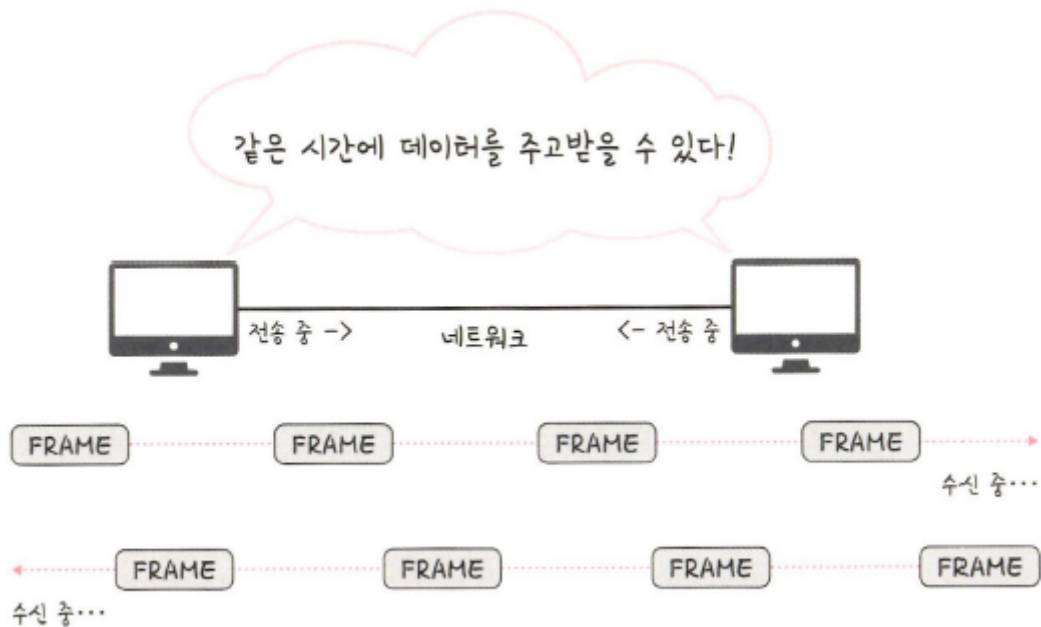
<이더넷>

이더넷(Ethernet)은 컴퓨터 네트워크 기술 중 하나로, 컴퓨터나 기타 네트워크 장치들이 서로 통신하기 위한 규칙과 방식을 정의하는 통신 프로토콜입니다. 이더넷은 주로 유선 랜(물리적인 케이블을 사용한 로컬 네트워크)에서 사용되며, 케이블을 통해 데이터를 전송합니다. 이더넷은 대부분의 컴퓨터에서 기본으로 지원되는 표준 기술로, 인터넷이나 기업 내부 네트워크 등 다양한 환경에서 사용됩니다.

<전이중화 통신>

양쪽 장치가 동시에 송수신할 수 있는 방식

송신로와 수신로로 나눠서 데이터를 주고 받으며 현대의 고속 이더넷은 이를 기반으로 통신



<CSMA/CD>

이전에는 유선 LAN에 '반이중화 통신' 중 하나인 CSMA/CD 방식을 사용했다.

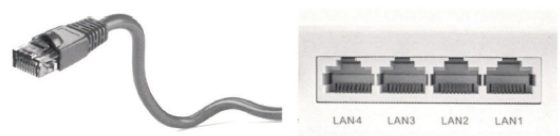
이 방식은 데이터를 보낸 이후 충돌이 발생한다면 일정 시간 이후 재전송하는 방식이다.

수신로와 송신로를 각각 둔 것이 아니고 **한 경로를 기반으로 데이터를 보내기** 때문에 데이터를 보낼 때 충돌에 대비해야 했다

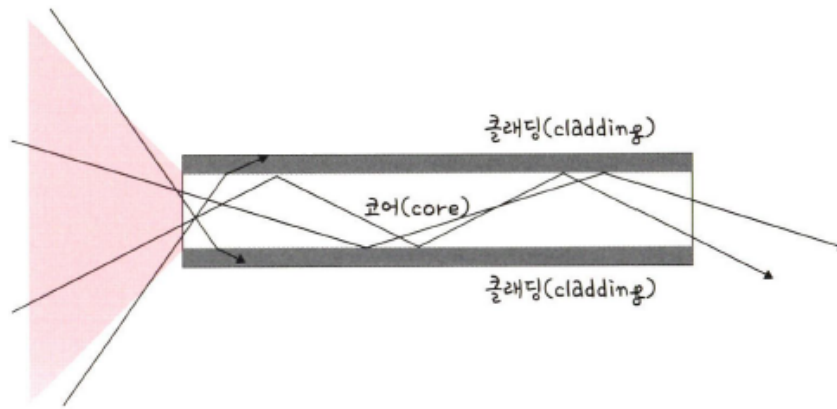
<유선 LAN을 이루는 케이블>

• TP 케이블 _ 트위스트 페어 케이블

하나의 케이블처럼 보이지만 실제로는 8개의 구리선을 꼬아 묶은 케이블



• 광섬유 케이블



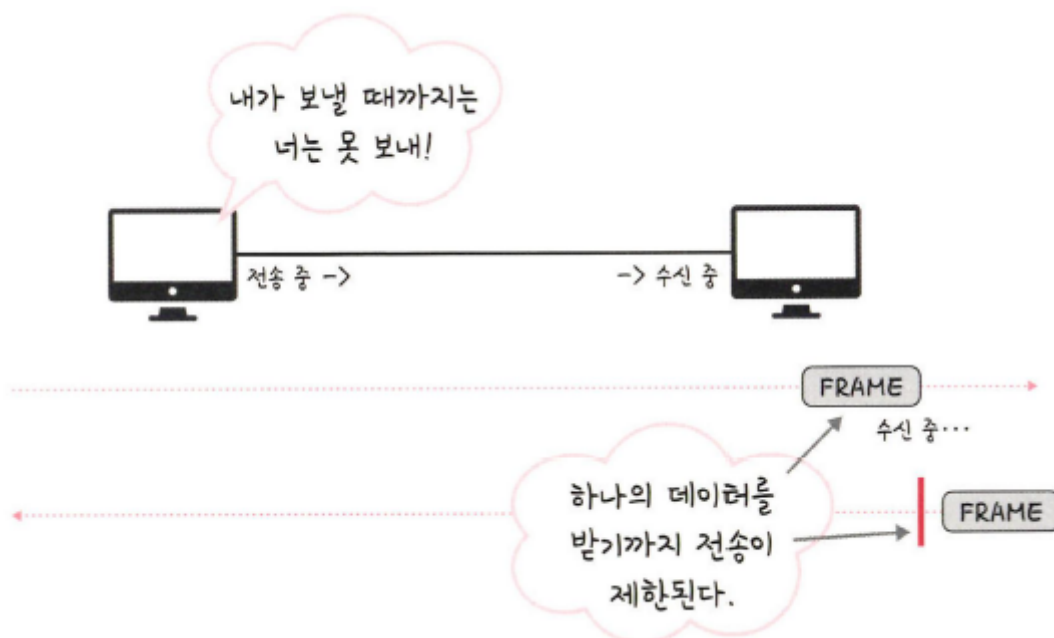
광섬유로 만든 케이블

레이저를 이용해서 통신하기 때문에 구리선과 비교할 수 없을 만큼의 장거리 및 고속 통신이 가능

광섬유 내부와 외부를 다른 밀도를 가진 유리나 플라스틱 섬유로 제작해서 한 번 들어간 빛이 내부에서 계속적으로 반사하며 전진해 반대편 끝까지 가는 원리

<무선 LAN>

무선 LAN 장치는 수신과 송신에 같은 채널을 사용하기 때문에 **반이중화 통신을 사용**
반이중화 통신은 동시에 통신할 수 없으며, 한 번에 한 방향으로만 통신할 수 있는 방식



<CSMA/CD>

CSMA/CD는 반이중화 통신 중 하나로 장치에서 데이터를 보내기 전에 캐리어 감지 등으로 사전에 가능한 한 충돌을 방지하는 방식을 사용

1. 데이터를 송신하기 전에 무선 매체 살핌
2. 캐리어 감지 : 회선이 비어 있는지 판단
3. IFS : 랜덤 값을 기반으로 정해진 시간 만큼 기다리며, 만약 무선 매체가 사용 중이면 점차 그 간격을 늘려가며 기다린다.
4. 이후에 데이터를 송신

<무선 LAN을 이루는 주파수>

무선 신호 전달 방식을 이용해 2대 이상의 장치를 연결하는 기술

2.4GHz 대역

장애물에 강한 특성을 가지고 있지만, 전자레인지, 무선 등 전파 간섭이 일어나는 경우가 많다

5GHz 대역

사용할 수 있는 채널 수도 많고 동시에 사용할 수 있기 때문에 상대적으로 깨끗한 전파 환경 구축

<와이파이>

전자기기들이 무선 LAN 신호에 연결할 수 있게 하는 기술

이를 활용하기 위해서는 무선 접속 장치인 공유기가 필요 → 유선 LAN에 흐르는 신호를 무선 LAN 신호로 바꿔주어 신호가 닿는 범위 내에서 무선 인터넷 사용 가능

<BSS>

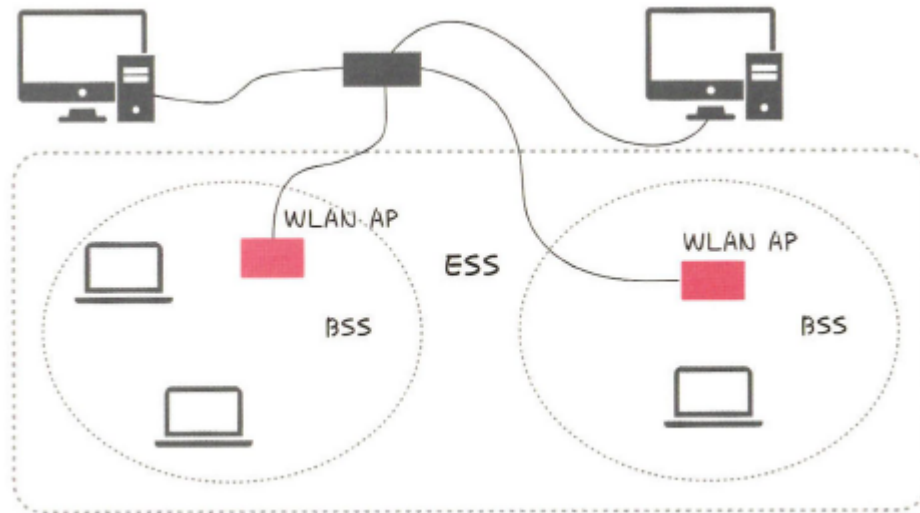
단순 공유기를 통해 네트워크에 접속하는 것이 아닌 동일 BSS 내에 있는 AP들과 장치들이 서로 통신이 가능한 구조를 말한다. 근거리 무선 통신을 제공하고, 하나의 AP만을 기반으로 구축이 되어 있어 사용자가 한 곳에서 다른 곳으로 자유롭게 이동하며 네트워크에 접속하는 것은 불가능

<ESS>

하나 이상의 연결된 BSS 그룹

장거리 무선 통신을 제공하며 BSS보다 더 많은 가용성과 이동성을 지원

사용자는 한 장소에서 다른 장소로 이동하며 중단 없이 네트워크에 계속 연결할 수 있다



<이더넷 프레임>

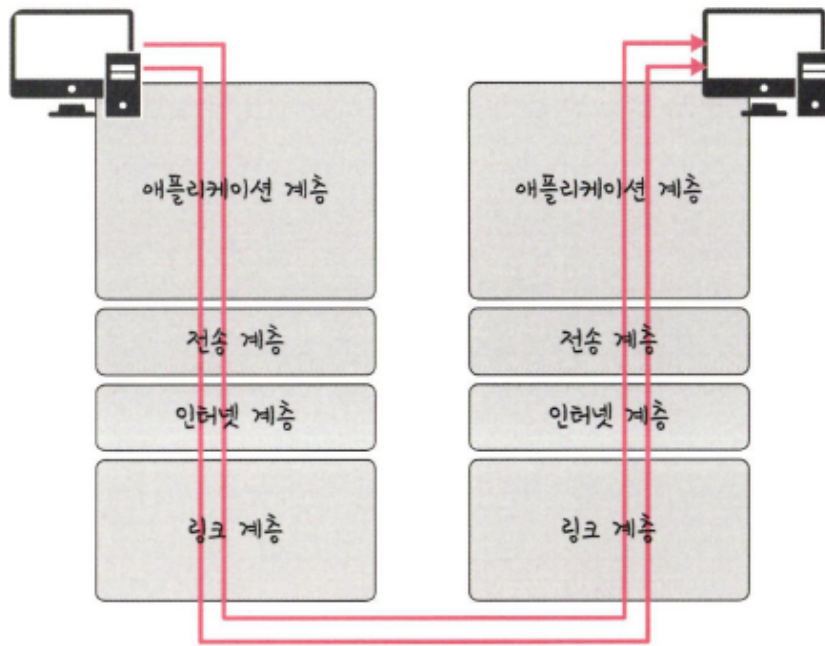
데이터 링크 계층은 이더넷 프레임을 통해 전달 받은 데이터의 에러를 검출하고 캡슐화한다



- **Preamble:** 이더넷 프레임이 시작임을 알립니다.
- **SFD(Start Frame Delimiter):** 다음 바이트부터 MAC 주소 필드가 시작됨을 알립니다.
- **DMAC, SMAC:** 수신, 송신 MAC 주소를 말합니다.
- **EtherType:** 데이터 계층 위의 계층인 IP 프로토콜을 정의합니다. 예를 들어 IPv4 또는 IPv6가 됩니다.
- **Payload:** 전달받은 데이터
- **CRC:** 에러 확인 비트

5. <계층간 데이터 송수신 과정>

ex) HTTP를 통해 웹 서버에 있는 데이터를 요청한다면 다음과 같은 과정이 발생



1. 애플리케이션 계층에서 전송 계층으로 사용자가 보내는 요청(request) 값들이 캡슐화 과정을 거쳐 전달
2. 다시 링크 계층을 통해 해당 서버와 통신을 하고, 해당 서버의 링크 계층으로부터 애플리케이션까지 비캡슐화 과정을 거쳐 데이터 전송

캡슐화 과정

송신 측에서 각 계층에서 페이로드에 헤더를 붙여 PDU로 만들어 아래 계층으로 전달한다
여기서 **헤더를 추가하는 과정, 필요한 데이터를 추가해 나가는 과정을 캡슐화라고 한다**

캡슐화 과정은 상위 계층의 헤더와 데이터를 하위 계층의 데이터 부분에 포함 시키고
해당 계층의 헤더를 삽입하는 과정

▼ 그림 2-32 캡슐화 과정



비캡슐화 과정

비캡슐화 과정은 하위 계층에서 상위 계층으로 가며 각 계층의 헤더 부분을 제거하는 과정

▼ 그림 2-33 비캡슐화 과정



2.2.3 PDU

프로토콜 데이터 단위 → 계층에서 처리하는 한 덩어리의 데이터 단위

네트워크의 어떠한 계층에서 계층으로 **데이터가 전달될 때, 한 덩어리의 단위를 PDU**(Protocol Data Unit)라고 한다

모든 계층에서, 우리가 전송하는 데이터를 '데이터'라고 부를까?

물론 데이터 자체는 동일하지만 각 레이어를 거치면서 헤더 정보가 추가되면서 이름이 달라진다.

계층	PDU 이름
애플리케이션 계층	메시지
트랜스포트 계층	세그먼트(TCP), 데이터그램(UDP)
네트워크 계층	패킷
데이터링크 계층	프레임
물리 계층	비트

PDU는 **제어 관련 정보들이 포함된 '헤더' + 데이터 자체를 의미하는 '페이로드'**로 구성

헤더(Header)

데이터를 전송하기 위한 정보를 담고 있으며, 일반적으로 **프로토콜, 송신자 및 수신자의 주소, 데이터의 길이** 등의 정보가 포함됩니다. 헤더는 일종의 소포에 비유할 수 있습니다. 소포

를 보낼 때는 소포에 대한 정보(받는 사람, 보내는 사람, 우편번호 등)를 적어서 붙인 봉투와 같은 역할을 합니다.

페이로드(Payload)

전송하고자 하는 실제 데이터를 의미합니다. 예를 들어, 인터넷에서 이메일을 보낼 때, 헤더에는 보내는 사람, 받는 사람, 제목 등의 정보가 포함되고, 페이로드에는 실제 이메일 내용이 포함됩니다. 따라서 페이로드는 일종의 소포 내부에 들어있는 실제 내용에 해당합니다.

이렇게 구성된 데이터는 전송 과정에서 여러 가지 프로토콜에 의해 처리되며, 이를 통해 데이터가 안전하게 전송됩니다. 이 과정에서 암호화 등의 보안 기술이 사용될 수도 있습니다.

• HTTP Body

- 메시지 본문을 통해 표현 데이터를 전달
- 메시지 본문 = 페이로드(payload)
- 표현은 요청이나 응답에서 전달할 실제 데이터
- 표현 헤더는 표현 데이터를 해석할 수 있는 정보 제공 (데이터 유형, 데이터 길이, 압축 정보 등)

