**네트워크 (1. 네트워크란 2.OSI계층 3.TCP/IP계층)**

**네트워크**란? 두 대 이상의 컴퓨터들을 연결하고 서로 통신할 수 있는 것

네트워크랑 네크워크를 연결하는 것을 **인터넷**이라고 하고, 연결하는 장비를 **라우터**라고 한다.

네트워크는 **데이터 전달 방식에 따라 두가지 네트워크로 나뉜다.**

**1. circuit switching 네트워크**

두명의 클라이언트가 하나의 링크를 공유하여 데이터를 주고받는다. 독점을 하고 있기 때문에 두 클라이언트가 통신이 끝나기 전까지 아무도 접근하지 못한다. 주로 전화하는 것과 같이 실시간 통신에 사용된다.

주로 사용방법이 두가지가 있는데 FDM이랑 TDM이 있다.

FDM (Frequency Division Multitasking)

할당된 대역폭을 여러 개의 대역폭으로 나누어 여러 단말기가 사용할 수 있게 하는 방식이다.

TDM (Time Division Multitasking)

TDM은 하나의 링크를 시간단위로 나누어 주어서 여러 명이 사용할 수 있게 만들어준다.

**Circuit switching 장점**

링크를 독점하고 있기 때문에 패킷 전송 방식에 비해 속도가 항상 일정하다. 그러므로 패킷의 단점이었던 데이터 loss와 delay가 없다.

**Circuit switching 단점**

결론적으로 보면 2 클라이언트가 하나의 링크를 독점하고 있기 때문에 다른 클라이언트들도 다른 클라이언트와 통신하려면 링크를 독점할 것이다. 수많은 링크를 만들어야 하는 단점이 있다.

**2.** **packet switching 네트워크**

네트워크 상에서 데이터를 일정한 크기인 패킷으로 나누어 주고받는 방법

(패킷은 보내는 데이터를 일정한 크기로 자른 뒤 목적지와 발신지의 주소를 추가한 데이터)

주로 라우팅 알고리즘을 이용하여 최단 경로를 설정하고 라우터들을 거쳐 도착하게 된다.

라우팅 알고리즘은 목적지까지 최적의 경로를 산출하고, 라우팅 테이블을 만들고 유지관리 하기위해 사용하는 알고리즘이다.

**packet switching 장점**

하나의 링크를 여러 컴퓨터가 공유하며 짧은시간 동안 packet을 보내서 효율적이다. 이는 circuit 전송방법 링크에 비해서 링크 가격이 저렴

Store and forward 방식을 사용. 그래서 일단 먼저 저장 후 crc, 에러 등의 확인을 진행한 후 전달하는 방식

**packet switching 단점**

라우터를 지날 때 마다 큐에 저장하여 차례대로 보내므로 queue의 메모리가 부족해지면 데이터 로스가 발생한다.

패킷이 많아지면 큐에서 대기하면 딜레이가 발생할 수 있다.

**OSI(Open systems innterconnection)**

- 개념 : 국제표준화기구(ISO)에서 개발한 모델, 일반적으로 **OSI 7계층 모형**이라고 한다.

컴퓨터 네트워크 프로토콜 디자인과 통신을 계층으로 나누어 설명한 것.

-> 즉, 네트워크에서 통신이 일어나는 과정을 7단계로 나눈 것.

- 계층 분류의 장점

1. 단순함: 공통된 기능을 하는 계층끼리 묶여있고 다른 기능을 하는 것들 끼리는 나뉘어져 있으므로 각각에 대해 설계하기 용이하고 유지 보수가 쉽고 간편

2. 유연성: 각 층에 대한 수정 및 향상이 필요할 때 다른 계층과 별개로 해당 계층에 대해서만 수정 및 향상을 진행할 수 있음

- 계층 분류의 단점

많은 계층들로 인해 유발되는 딜레이로 오버헤드 발생 가능

- 계층별 설명

**1. 물리 계층 :**

전기적, 기계적, 기능적인 특성을 이용해서 통신 케이블로 데이터를 전송

디지털 데이터를 아날로그적인 전기적 신호로 변환하여 물리적인 전송 가능하게 함

주소 개념이 없음

**신호로 변환하여 전송하는 계층**

통신 단위 : 비트

주요 프로토콜 : X.21, RS-232C 등

주요 장비 : 통신 케이블. 리피터(Repeater), 허브(Hub)

**2. 데이터 링크 계층**

인접한 노드간의 신뢰성 있는 데이터 전송을 제어

MAC 주소를 통해 목적지까지 찾아감

신뢰성 있는 전송을 위해 흐름제어(flow control), 오류제어(error control), 회선제어(line control) 수행

흐름 제어? 데이터 송.수신 속도에서 발생하는 프레임 유실을 막기 위한 기술

두 기기의 프레임 송,수신 속도가 다르면 문제 발생

따라서 기기 중 느린 네트워크 속도에 맞춰서 프레임 전송

오류 제어 ? 손실, 손상된 프레임에 대해 재전송 요청

(손실 : 수신 받은 노드에서 프레임이 도착했다는 확인 메시지 안옴

손상 : 프레임의 데이터가 변조된 상태, 손상된 상태 확인은 프레임의 트레일러를 사용해서 확인. 프레임의 bit개수(데이터 + 트레일러)가 짝수인지 확인하는 Parity bit check 방법을 이용))

**물리적인 매체에 데이터를 실어 보내는 계층**

통신 단위 : 프레임(frame)

주요 프로토콜 : Ethernet, TokenRing, FrameRelay 등

주요 장비 : 브리지(bridege), L2 스위치(swtich)

**3. 네트워크 계층 :**

종단 간 전송을 위한 **경로 설정**을 담당(End to End, Host to Host Delivery)

호스트로 도달하기 위한 최적의 경로를 라우팅 알고리즘을 통해 선택하고 제어

종단간 전송을 위한 주소로 IP주소를 사용

**네트워크를 논리적으로 구분하고 연결하는 계층**

통신 단위 : 데이터그램(datagram) or 패킷(packet)????????

주요 프로토콜 : **IP**, ARP, ICMP, IGMP, RIP, IGRP 등

주요 장치 : 라우터(router), L3 스위치(swtich)

**4. 전송 계층 :**

종단간 신뢰성 있는 데이터 전송을 담당(End to End Reliable Delibery)

주소로 Port Number을 이용

신뢰성 있는 데이터 전송을 위해 분할과 재조합, 연결제어, 흐름제어, 오류제어, 혼잡제어 수행

**데이터의 전송 방식을 담당하는 계층**

통신 단위 : 세그먼트(segment), 패킷(Packet), 유저데이터그램(user datagram) 등 이름이 다를 뿐 같은 말

주요 프로토콜 : **TCP, UDP**

주요 장비 : L4 스위치(swtich)

**\* data-link와 transport 차이점**

**오류제어**

데이터 링크 계층은 노드 대 노드 전송의 프레임에 대한 오류를 제어하나, 전송 계층은 전체 세그먼트 오류(손상, 유실 등) 없이 수신측 전송 계층에 도착하도록 오류 제어 기능을 수행.

**흐름제어**

데이터 링크에서는 송수신 양단 간에 송신율 및 수신율의 균형을 맞추는 것이고 트랜스포트에서는 단일 데이터 링크간이 아닌, 전송계층 종단-대-종단 간에 흐름제어 기능을 하는 것이다.

**5. 세션 계층 :**

응용 프로그램 간의 논리적인 연결(세션) 생성 및 제어 담당

세션 계층부터는 소프트웨어 개발자가 직접 코딩해야 하는 부분

**응용 프로그램(프로세스) 간의 연결을 지원해주는 계층**

통신 단위 : 데이터(data), 메시지(message)

주요 프로토콜 : 해당사항 없음

주요 장비 : 해당사항 없음

**6. 표현 계층 :**

데이터 표현 방식, 상이한 부호체계 간의 변화에 대해 규정

인코딩/디코딩, 압축/해제, 암호화/복호화 등의 역할 수행

**데이터의 변환 작업을 하는 계층**

통신 단위 : 데이터(data)

주요 프로토콜 : 해당사항 없음

주요 장비 : 해당사항 없음

**7. 어플리케이션 계층 :**

사용자가 네트워크에 접근할 수 있도록 해줌

**User Interface를 제공하는 계층**

통신 단위 : 데이터(data)

주요 프로토콜 : Telnet, FTP, SMTP, HTTP 등

주요 장비 : 해당 없음

각 계층의 주요 장비에 대한 간략한 설명

- 리피터(Repeater) : 전기적 신호를 받아 이진수로 바꿔서 다시 전기적 신호로 전해주는 기기

- 라우터(Router) : 두 개의 네트워크 사이에서 정보를 전달하도록 도와주는 기기, 보안 기능을 제공

- 허브(Herb) : 이더넷 케이블을 통해 연결하여 네트워크의 컴퓨터 간에 통신을 도와주는 기기, 스위치보다 느림

- 스위치(Switch) : 허브와 동일하게 작동하지만 대상을 식별 가능하며 정보를 동시에 주고 받을 수 있음

- 브릿지(Bridge) : 여러 네트워크를 연결하도록 스위치를 업그레이드 시킨 기기, 4G **↔** 3G **↔** Wifi 변환시켜주는 에그

- 게이트웨이(Gateway) : 네트워크 간의 통로 역할을 하는 장치, 다른 기종 간의 프로토콜을 적절히 변환시켜주는 역할

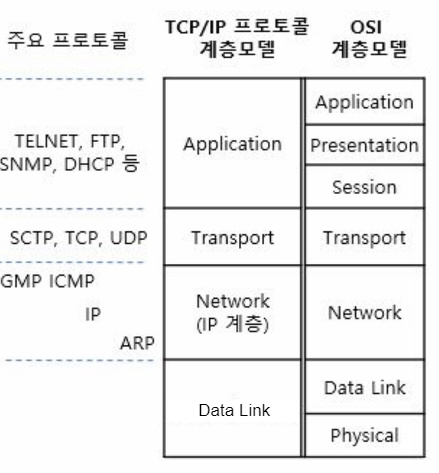
**TCP/IP protocol suite**

- 개념:

OSI 7계층보다 먼저 개발된 네트워크 모델, 4계층으로 구성

OSI 7계층은 네트워크의 표준이 되기는 하지만 네트워크가 OSI를 이용해 실제적으로 구현되는 예가 거의 없다.

**실질적인 통신은 TCP/IP프로토콜 모델을 기반으로 구현**된다.



- 계층별 설명(위에 설명한 OSI계층별 설명과 동일, 다른점은 몇 개의 계층이 합쳐있다는 점)

**1. 데이터링크계층 :**

OSI 7계층에서 물리계층과 데이터링크 계층에 해당

하드웨어적인 요소와 관련되는 모든 것을 지원하는 계층

송신측 컴퓨터의 경우 상위 계층으로부터 전달받은 패킷에 물리적인 주소은 MAC 주소 정보를 가지고 있는 헤더를 추가하여 프레임을 만들고, 프레임을 하위계층인 물리 계층으로 전달한다.

수신측 컴퓨터의 경우 데이터 링크 계층에서 추가된 헤더를 제거하여 상위 계층인 네트워크 계층으로 전달한다.

**2. 네트워크계층 :**

OSI 계층에서 네트워크 계층에 해당

라우팅 기능 + 상위 트랜스포트 계층으로부터 받은 데이터에 IP패킷 헤더를 부여 IP패킷을 만들고 이를 전송

**3. 전송계층 :**

OSI 계층에서 전송계층에 해당

네트워크 양단의 송수신 호스트 사이에서 신뢰성 있는 전송 기능 제공

**4. 응용계층 :**

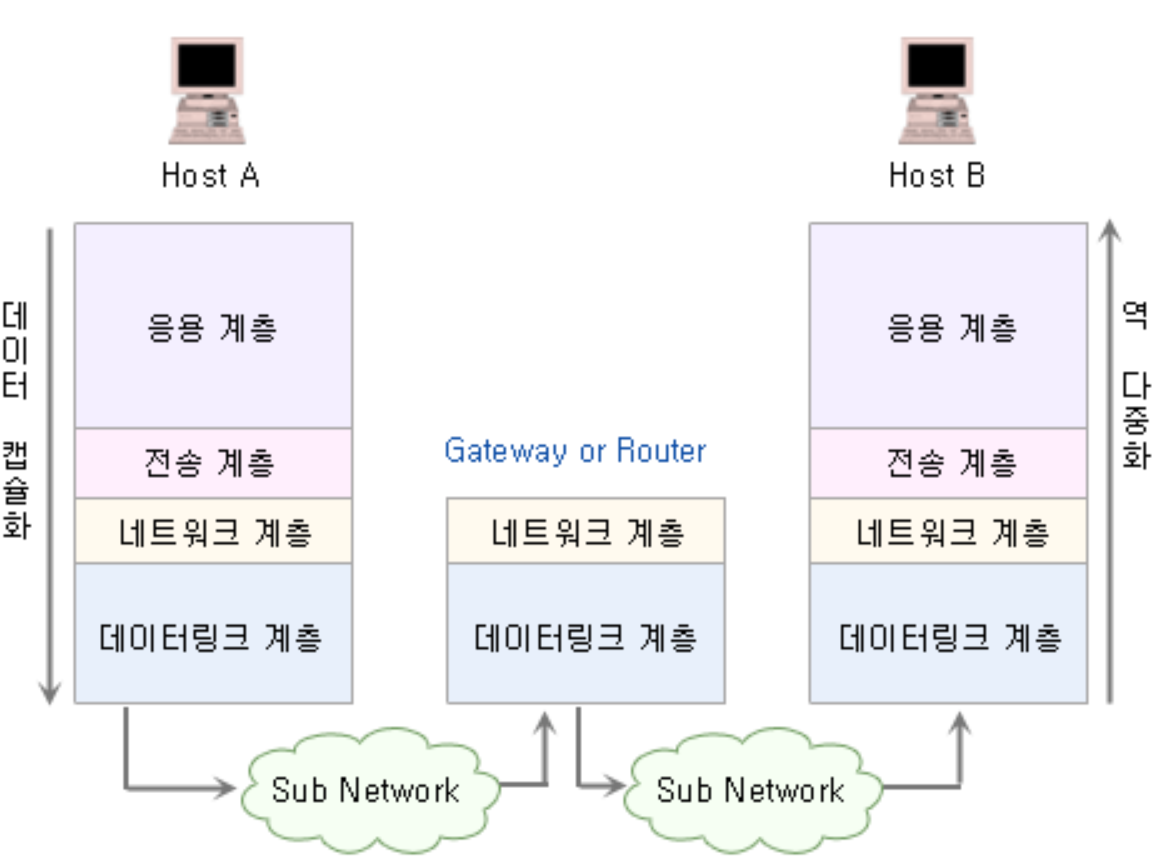
OSI 계층에서 세션계층, 프레젠테이션계층, 애플리케이션 계층에 해당

응용프로그램들이 네트워크서비스, 메일서비스, 웹서비스 등을 할 수 있도록 표준적인 인터페이스 제공

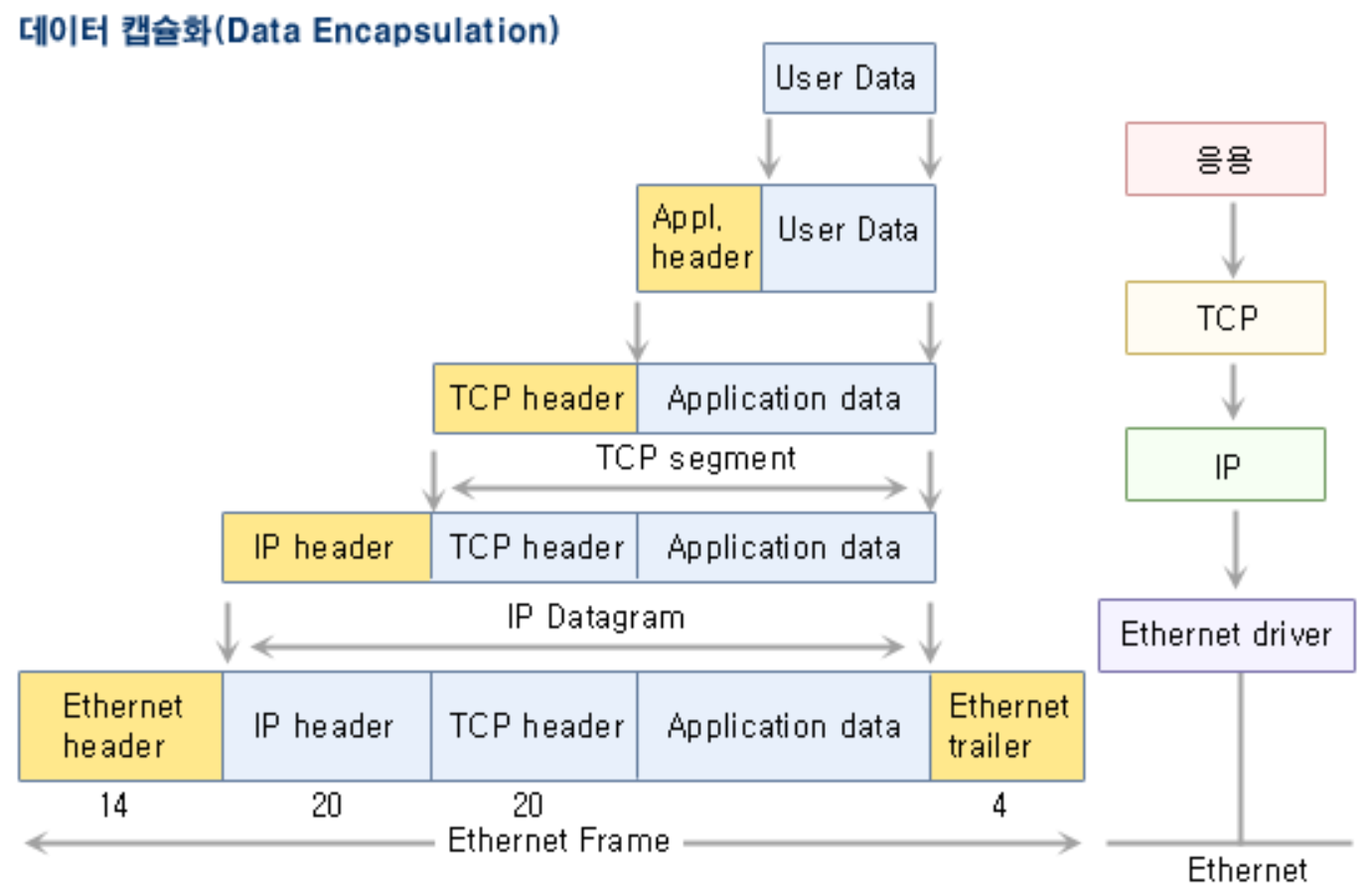
- 통신흐름

TCP/IP로 통신할 때 계층을 순서대로 거쳐 상대와 통신

송신하는 측은 애플리케이션 계층에서부터 내려가고, 수신하는 측은 네트워크 계층부터 올라감



**1. 데이터 캡술화(data encapsulaton)? 송신 측 입장에서 데이터 생성과정**



각 계층을 지나면서 각각의 **헤더(header)**를 붙이는 작업

헤더란? 각 계층에서 해야할 일들을 정의한 데이터.

**2. 역캡슐화? 수신 측 입장에서 데이터 생성과정**

상위 프로토콜 계층으로 올라가며 각 헤더 부분을 삭제하는 작업

**TCP/IP 모델의 각 계층별 프로토콜에 대해 자세히 설명해보자**

**프로토콜**

**1. Data-Link Layer : 패킷을 전달하는 인터페이스와 관련된 하드웨어를 제어하는 기능**

**① 이더넷**

- 회선상에서 데이터를 전송하는 여러 가지 방식들중 현재 전세계적으로 90%이상 사용되는 대표적인 네트워크 형태이다.

- 데이터를 전송하기 위해 CSMA/CD 라는 약속을 사용하며 전송속도는 현재 10~1000Mbps 까지 다양하게 존재한다.

- 이더넷에 접속되어 있는 장치들은 어느때라도 데이터를 전송할 수 있는데, 전송하기 전에 회선이 사용 중인지 감시하고 있다가 회선이 비어 있을 때 데이터를 전송한다. 만약 데이터를 전송하는 시점에 다른 장치가 동시에 전송을 개시하면 충돌이 발생하게 되며, 충돌한 데이터들은 버려지고 데이터를 전송한 장치들에게 재전송을 요구하게 된다.

**2. Network Layer : 네트워크 상의 패킷 이동의 제어 기능을 수행**

**① IP (Internet Protocol)**

- TCP/ IP 기반의 인터넷 망을 통하여 **데이터 전달을 담당**하는 프로토콜이다.

- 패킷의 완전한 전달을 보장하지 않는다. (비신뢰성, 비연결성)

- **IP패킷 헤더 내에 수신 및 발신 IP주소**를 포함한다.

**IP 주소 : 각각의 host를 구분하기 위해 사용되는 주소**

하나의 IP주소에는 network ID와 host ID가 존재한다.

(네트워크 id가 같으면 같은 network라는 뜻)

IP주소는 현재 2가지 체계가 존재

**1. IPv4** : 32비트 길이의 식별자.

일반적으로 8비트씩 끊어서 이를 0~255의 10진수로 나타내며, 각 숫자는 점(.)으로 구분한다.

예시)

127.0.0.1 : 자기 자신을 가리키기 위한 목적으로 쓰기 위해 예약된 IP주소

**2. IPv6 :** IPv4 주소가 고갈됨에 따라 기존의 IPv4주소체계를 128비트 크기로 확장한 차세대 인터넷 프로토콜 주소

IPv4와 달리 일반 숫자 표현을 10진수가 아닌 16진수로 표기

**IPv4에 대해 더 자세히 설명하자**

IPv4에서는 주소를A,B,C,D,E class로 나누어 network id와 host id 를 구분한다.

(인터넷상의 모든 host들을 관리하기 힘들므로 네트워크 범위 지정해서 관리 쉽게 하려고)

1. A class : 처음 8bit가 network id, 나머지 24bit는 host id

2. B class : 처음 16bit가 network id, 나머지 16bit는 host id

3. C class :

4. D class : 멀티캐스트를 위한 주소

네트워크 전송 방법

유니캐스트(unicast) : 1:1 -> server: cilent

브로드캐스트(broadcast): 1:all - >인하대 모든 동아리

멀티캐스트(multicast)? 1:N 특정 그룹에게만 -> 인하대 특정 동아리

5. E class : 미래에 사용하기 위해 남겨둔 주소

IP class구별 방법

IP 주소에서 쓸 수 있는 숫자의 범위는 0~255로 되어 있기에 첫 번째 Octet에서 0~255까지의 숫자를 5개로 나누어서 A, B, C, D, E Class로 구분

A Class : 0 ~ 127 (0.0.0.0 ~ 127.255.255.255)

B Class : 128 ~ 191 (128.0.0.0 ~ 191.255.255.255)

C Class : 192 ~ 223 (192.0.0.0 ~ 233.255.255.255)

D Class : 224 ~ 239 (224.0.0.0 ~ 239.255.255.255)

E Class : 240 ~ 255 (240.0.0.0. ~ 255.255.255.255)

**② ICMP (Internet Control Message Protocol)**

- 인터넷 계층 프로토콜 중 ICMP는 호스트 또는 라우터 사이에 오류정보, 제어 메세지를 전달하는데 사용되며 주로 IP가 이용하지만 ping과같은 응용프로그램이 직접 사용하는 경우도 있다.

**③ IGMP (Internet Group Message Protocol)**

- 로컬 네트워크상의 멀티캐스팅 그룹관리를 위한 프로토콜

- 하나의 라우터와 여러 호스트로 구성되는 서브네트워크 상에서, 호스트들이 어떤 멀티캐스트 그룹에 속하는 가를 라우터가 알도록 하기위한 일종의 그룹 관리용 프로토콜

- ICMP, IGMP등은 데이터 전송용 프로토콜이 아니고, 네트워크상에 이벤트 또는 변화를 알리는데 사용되는 제어용 프로토콜이다.

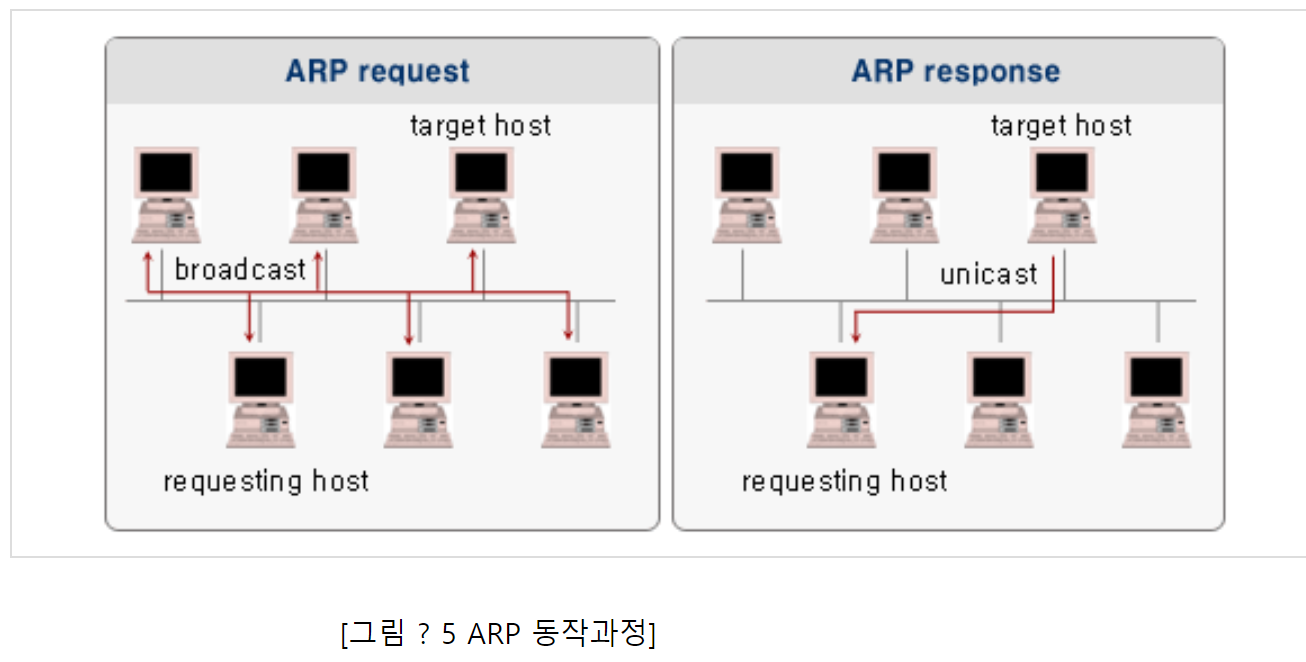
**④ ARP (Address Resolution Protocol)**

- 인터넷에 연결되는 모든 호스트는 네트워크 계층 주소에 해당하는 32bit의 IP주소를 가지고 있다.

- LAN에서 어떤 컴퓨터에 IP패킷을 물리적으로 전달하기 위해서는 먼저 그 장비의 MAC주소를 알아야 하는데 어떤 장비의 IP주소를 가지고 그 장비의 MAC 주소를 알아내기 위한 절차(프로토콜)를 ARP 라고 한다.

**MAC 주소**: 네트워크 세그먼트의 데이터 링크 계층에서 통신을 위한 네트워크 인터페이스에 할당된 **고유 식별자**

- ARP request(Broadcast)/ ARP response(unicast)



ARP request : 목적지 ip주소를 써서 모든 컴퓨터에게 보낸다. 즉 내가 목적지의 IP주소를 알고있는데 그 목적지의 MAC address를 알고 있으면 응답을 달라 요청함(brodadcast)

ARP response : 목적지의 컴퓨터가 요청한 컴퓨터에게 바로 직접적인 응답을 해준다(unicast)

- 동일한 목적지 호스트에 IP패킷을 연속하여 보낼 때 매번 ARP를 사용하면 ARP를 처리하기 위한 패킷들을 자주 전송하게 되어 대역 이용률이 떨어진다. 이와 같이 매번 ARP request 패킷을 발송하지 않도록 하기위하여, ARP로 얻은 최근의 정보를 캐시에 기록해 두는 것이 효율적이다.

**3. Transport Layer : 두 호스트 간에 종단 간 연결을 맺고 데이터를 전달할 수 있는 기능 수행**-

**① TCP (Transmission Control Protocol)**

- 인터넷상의 컴퓨터들 사이에서 데이터를 메시지의 형태로 보내기 위해 IP와 함께 사용되는 프로토콜이다.

- IP가 실제로 데이터의 배달처리를 하는 동안, TCP는 데이터 패킷을 추적 관리한다.

- TCP는 IP가 처리할 수 있도록 메시지를 여러개의 패킷들로 확실히 나누고, 반대편에서는 완전한 메시지로 패킷들을 재조립할 책임이 있다.

- TCP 특징

**1. connection-oriented**

연결 지향 프로토콜이기 때문에 통신을 위해선, 미리 연결을 맺어야 한다.

 → **three way handshake** 를 통해 연결을 맺는다.

**three way handshake** : TCP/IP 프로토콜을 이용해서 통신을 하는 응용프로그램이 데이터를 전송하기 전에  정확한 전송을 보장하기 위해 상대방 컴퓨터와 세션을 수립하는 과정 (양쪽 모두 데이터를 전송할 준비가 되있음을 보장)

그림으로 보자면

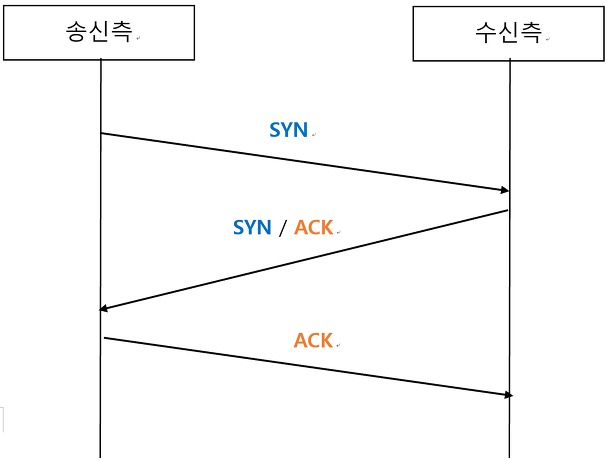


그림 . Three way handshake

STEP 1 : 송신측에 접속을 요청하는 SYN 패킷을 보낸다.

STEP 2 : 수신측은 SYN 요청을 받고 송신측에 요청을 수락한다는 ACK와 SYN 패킷을 발송해

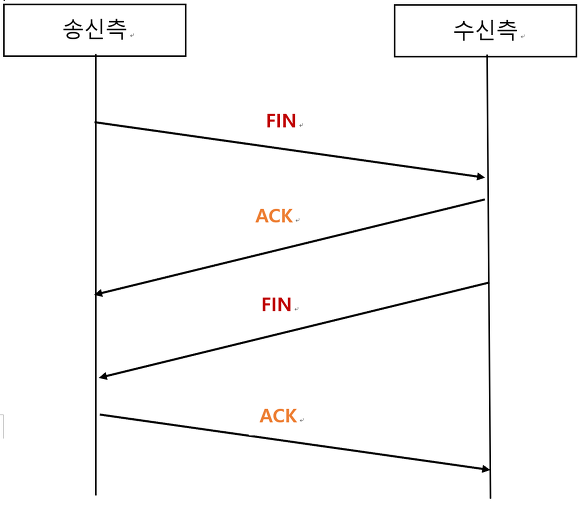
송신측이 다시 ACK으로 응답하기를 기다린다.

STEP 3 : 송신측이 수신측에 ACK을 보낸 이후로부터는 연결이 이루어지고 데이터가 오간다.

위와 같은 방식으로 통신하는 것이 신뢰성 있는 연결을 맺어 준다는 TCP의 3 way handshake 방식이다.

연결을 종료할 때는 **four way handshake**사용

그림으로 보자면



STEP 1 : 연결을 종료하고자 하는 송신측은 FIN 패킷을 전송한다.

STEP 2 : 수신측은 알겠다는 ACK 패킷을 전송하고, 자신의 통신이 끝날 때까지 기다림

(아직 전송할 데이터가 남아있다면 이어서 전송)

STEP 3 : 수신측의 통신이 완전히 끝났다면 연결 종료 요청에 합의한다는 의미로 송신측에

FIN 패킷 전송

STEP 4 : 송신측이 확인했다는 ACK 패킷 보냄

**2. Reliability**

신뢰성을 있게 Error detection & recovery를 수행

Error 종류

**1) Bit Level Error**

Channel/Physical Link Level에서 발생하는 Error로 Data의 값이 바뀐 경우.

**2) Packet Level Error**

Network/Data Link Level에서 발생하는 Error로 Data가 사라지는 경우.

Network의 router에서 입력 Data가 많으면 버퍼에 쌓이다가 없어지는 경우.

Loss, Out of Order, Duplicate

Error Detection 방법

**1) Bit Level Error** :CRC (Error Detection Code)

**2) Packet Level Error** : ARQ (Acknowledgement, Sequence Number)

Error recovery 방법 : **Retransmission**

**3. flow & congestion control**

**Flow Control: 송신측과 수신측의 데이터 처리 속도 차이를 해결하기 위한 기법**

**내부 방식으로는 stop and wait 방식과 sliding window 방식 존재**

**1) stop and wait:** 매번 전송한 패킷에 대해 확인 응답ACK을 받아야만 그 다은 패킷을 전송하는 방법, RTT(retransmission timer)안에 ACK가 오지 않으면 재전송

하지만 제대로 보내졌지만 ACK가 lost되거나 RTT를 넘기면 불필요한 재전송이 일어나게 된다는 단점이 있다.

**2) sliding window : 수신측에서 설정한 윈도우 크기만큼 송신측에서 확인응답 없이 세그먼트를 전송할 수 있게 하여 데이터 흐름을 동적으로 조절하는 제어기법**Stop and wait의 단점을 보완하는 방법

**윈도우(window)** : 단위 시간 내에 보내는 패킷의 수

**\*\* 참고 \*\***

**Silly Window Syndrome**

**body보다 header의 크기가 큰 패킷의 전송이 잦음으로 네트워크가 혼란스러운 상황**

**송신측 신드롬 ? application이 tcp버퍼안에다가 천천히 데이터 보내서 생긴 신드롬**

**작은 사이즈의 데이터를 잦은 전송을 해서**

**수신측 신드롬 ? 데이터의 처리가 느려질 경우 생긴 신드롬**

**작은 사이즈의 데이터를 여러번 받아야 해서**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Congestion Control: 송신측의 데이터 전달과 네트워크의 데이터 처리속도 차이를 해결하기 위한 기법**

송신측의 데이터는 지역망이나 인터넷으로 연결된 대형 네트워크를 통해 전달.

만약 한 라우터에 데이터가 몰릴 경우, 자신에게 온 데이터를 모두 처리 할 수 없게 된다.

이런 경우 호스트들은 또 다시 재전송을 하게 되고 결국 혼잡만 가중시켜 오버플로우나 데이터 손실을 발생시킨다.

따라서 이러한 네트워크의 혼잡을 피하기 위해 송신측에서 보내는 데이터의 전송속도를 강제로 줄이게 되는데, 이러한 작업을 **혼잡제어**라 한다.  
흐름제어가 송신측과 수신측 사이의 전송속도를 다루는데 반해 혼잡제어는 호스트와 라우터를 포함한 보다 넓은 관점에서의 전송 문제를 다룬다.

**혼잡제어 알고리즘 : AIMD(Additive Increase / Multiplicative Decrease)**

처음에 패킷을 하나씩 보내고 이것이 문제없이 도착하면 window크기(단위 시간 내에 보내는 패킷의 수)를 1씩 증가시켜가며 전송하는 방법. 만일 패킷전송을 실패하거나 일정 시간을 넘으면 패킷을 보내는 속도를 절반으로 줄인다. 이 방식을 사용하는 여러 호스트가 한 네트워크를 공유하고 있으면 나중에 진입하는 쪽이 처음에는 불리하지만 시간이 흐르면 평형 상태로 수렴하게 되는 특징이 있다. 문제점은 초기에 네트워크의 높은 대역폭을 사용하지 못하여 오랜 시간이 걸리게 되고, 네트워크가 혼잡해지는 상황을 미리 감지하지는 못한다는 점. 즉, 네트워크가 혼잡해지고 나서야 대역폭을 줄이는 방식이다.

**② UDP (User Datagram Protocol)**

- UDP는 IP를 사용하는 네트워크 내에서 컴퓨터들 간에 메시지들이 교환될 떄 제한된 서비스만을 제공하는 통신 프로토콜이다.

- 특징

**Connectionless :** 송수신측 연결 과정 없이 즉시 데이터 전송

**Unreliable :** datagram이 사라지거나 중복되어도 상관x

**No flow/congestion control**

- UDP 는 TCP와는 달리 메시지를 패킷으로 나누고, 반대편에서 재조립하는 등의 서비스는 제공하지 않는다.

- 교환해야할 데이터가 매우 적은 네트워크 응용프로그램들은 처리시간 단축을 위해 Tcp보다 UDP를 쓴다.

**4. Application Layer : TCP/IP프로토콜의 이용하는 서비스**

**① Telnet**

- 인터넷이나 로컬 영역 네트워크 연결에 쓰이는 프로토콜

- 원격 접속 서비스로서 특정 사용자가 네트워크를 통해 다른 컴퓨터에 연결하여 그 컴퓨터에서 제공하는 서비스를 받을 수 있게 한다.

**② FTP (File Transfer Protocol)**

- 파일 전송 프로토콜

**③SNMP (Simple Network Management Protocol)**

- SNMP는 TCP/IP 프로토콜 그룹을 이용하여 인터넷상에서 장치를 관리하기위한 기반구조이다. 이것은 인터넷을 감시하고 관리하기 위한 기본적인 운영을 제공한다.

**④SMTP (Simple Mail Transfer Protocol)**

- 전자 우편을 보내고 받는데 사용되는 TCP/IP 프로토콜이다.

**⑤HTTP (Hyper Text Transfer Protocol)**

- Web-Browser같은 응용프로그램을 통해 Web-Client와 Web-Server사이에 데이터를 전송하는 프로토콜 이다.

- 평문 (암호화 하지않은) 통신이기 때문에 도청이 가능하다.

**⑥HTTPS (Hyper Test Transfer Protocol Secure)**

- HTTP의 약점을 보안하기 위하여 HTTP 암호화나 인증 등의 구조를 더한 것 이다.

- 웹페이지의 로그인이나 쇼핑의 결제 화면 등에서 사용되고 있다.