

백엔드/프론트 역할 및 네트워크

개발에 들어가기 앞서

네트워크

Network

프로토콜

OSI 7계층

1계층 - 물리계층

물리 계층

데이터 링크 계층

네트워크 계층

전송 계층

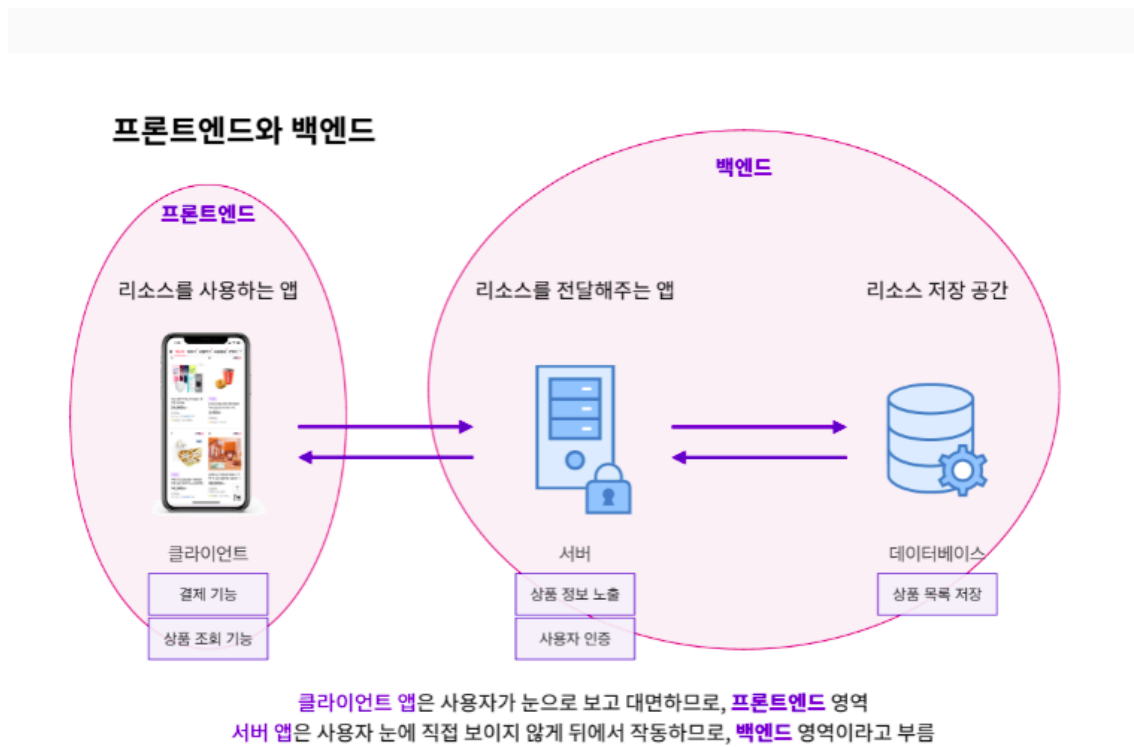
응용 계층

개발에 들어가기 앞서

우리가 매일 보고 사용하는 웹, 앱 서비스들은 **우리 눈에 보이는 영역을 개발하는 일**과 **눈에 보이지 않는 뒷단을 개발하는 일**로 나눌 수 있습니다.

프론트엔드는 우리가 늘 보고 있는 웹 사이트 내 화면처럼 사용자가 볼 수 있는 화면을, 그리고 백엔드는 사용자가 볼 수 없는 환경을 구현합니다. 사용자가 원하는 정보를 제공할 수 있게 데이터를 저장 및 관리하거나, 서버가 터지지 않게 운영하는 일 등을 합니다.

정리하면 **프론트엔드에 있는 사용자들이 원하는 행동(이벤트)을 처리하는 것이 백엔드**라고 할 수 있습니다.



그렇다면 프론트 영역에서의 사용자들의 행동을 어떻게 알고 백엔드에서 처리할 수 있을까요?

간단하게 레스토랑 예시를 들어서 설명하겠습니다.

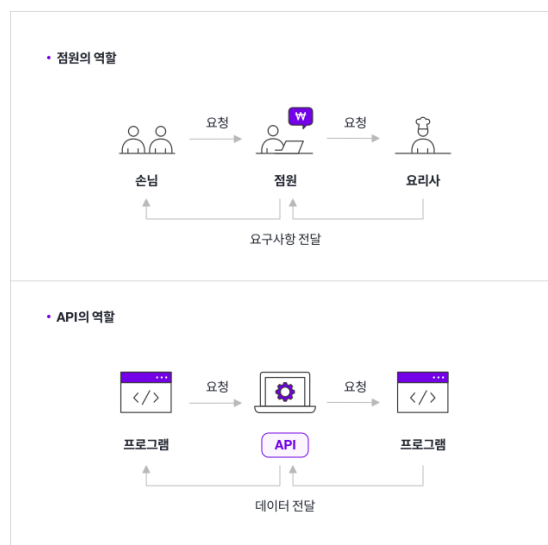
우리가 가게를 들어가게 되며 점원이 가져다준 메뉴판을 보며 메뉴를 고르게 되는데, 메뉴를 고르게 되면 점원이 해당 주문을 받고 요리사에게 요청을 하게 됩니다. 그러면 요리사는 해당 요청을 보고 맛있는 음식을 만들어서 점원에게 전달해주고, 우리는 그 음식을 점원에게 받아 맛있는 음식을 먹게 되는거죠.

여기서 손님은 프론트. 요리사는 백엔드이고, 점원은 바로 API 입니다.

API란? Application Programming Interface.

프로그램들이 서로 상호작용하는 것을 도와주는 매개체 역할을 해줌으로써,

손님(프론트)에서 주문할 수 있게 메뉴(명령 목록)를 정리하고, 주문(요청)을 보내면 요리사(백엔드)와 상호작용하여 요청된 메뉴(응답)를 전달하게 됩니다.



API는 정보전달의 역할 외에도 여러 역할을 하는데.

1. API는 서버와 데이터베이스에 대한 출입구 역할
2. 서비스(어플리케이션 등)과 기기가 원활하게 통신할 수 있게 하는 역할
3. 모든 접속을 표준화해주는 역할

등을 합니다.

API는 클라이언트와 서버 간의 정보전달 방법이기도 하지만, 앱과 앱, 프로그램과 프로그램 등에서의 정보전달도 모두 포함하는 인터페이스들을 한꺼번에 말하기도 합니다.



REST API, CRUD 등 다른 개념들이 존재하지만, 개인적인 학습 또는 2주차에 추가적으로 설명드리겠습니다!

자 그러면 우리가 사용하는 인터넷에서는 프론트와 백엔드 영역이 존재하고, 그 사이 정보전달에서는 API라는 매개체가 요청과 응답을 중계해준다는 것을 알게되었습니다.

우리가 폰에서 쿠팡에 들어가서 보고싶은 상품을 클릭했었을때(프론트에서의 클릭이라는 행위)
API가 해당 상품에 대한 요청을 백엔드로 전송하고, 백엔드에서는 해당 상품에 대한 정보를
다시 API에 응답을 보내게 되면 우리 핸드폰에 클릭했던 상품의 정보가 보이는거죠..!

하지만 풀리지 않는 의문이 있습니다.

그렇다면 어떻게 우리 집에서 쿠팡 앱에 접속했는데 저 멀리 있는 쿠팡 서버에 연결되서 데이터를 전달 받게 된 것일까요..?

연결은 그렇다 치고 컴퓨터는 단순히 0과 1을 계산하는 녀석으로 알고 있는데, 클릭 한번에 우리가 알아볼 수 있는 상품 데이터를 가져올 수 있던걸까요..?

이 데이터의 전달에 대해 알기 위해서는 **"네트워크"**에 대한 이해가 필요합니다.

네트워크

네트워크에 대해 다들 한번씩 들어보셨을꺼라 생각합니다.

어쩌면 인터넷이라는 단어가 좀 더 익숙하실 수도 있겠네요.

우리가 은행 업무를 처리하기 위해 은행에 직접 찾아가 이체를 하고, 등본을 떼기 위해 동사무소에 직접 찾아가고 친구들과 재밌는 게임하러 동네 오락실에 찾아가던 옛날 세상에서

집에서도 모든 업무를 처리하고, 재밌는 게임도 밖에 나가지 않고 친구와 각자 카카오톡이나 디스코드를 하면서 소통할 수 있도록 만들어준 것도 바로 인터넷. 즉, 네트워크입니다.

이제는 네트워크 없는 세상은 상상할 수도 없게 되어버렸죠.

지금 보고있는 이 노선의 글조차 네트워크가 존재하기에 볼 수 있는거죠.

그렇다면 네트워크란 정확히 무엇일까요?



Network

정보 공유를 목적으로 컴퓨터와 컴퓨터가 **네트워킹**하여 형성된 망을 의미합니다.

여기서 또 **네트워킹**이란 서로 연결된 장비끼리 통신(대화) 할 수 있도록 연결된 것을 의미합니다.

네트워크가 생기게 된 이유를 간단히 설명하자면

비싼 장비를 혼자 쓰는 게 아까웠던 것에서부터 시작되었습니다. 그래서 여러 대의 컴퓨터에서 장비를 공유해서 사용해보자 해서 호스트 컴퓨터에 여러 장비들을 연결하고 거기서 또 호스트 컴퓨터들을 공유해보자 라는 아이디어가 또 나오게 되고 점점 발전하여 현재의 네트워크까지 도달하게 되었습니다.

하지만 여러 장치들이 연결된 네트워크에선 정보 교환을 위해 몇 가지 규약이 필요했습니다.

이를 위해 **프로토콜**이라는 통신 규약이 등장했습니다.

프로토콜

네트워크 안에서 데이터 전송을 원활하게 하기 위해 필요한 규칙과 약속을 미리 정한 것입니다.

쉽게 말하면, 정보 공유를 위해 서로 약속을 하고 대화를 하는 거죠

우리가 서로 약속 시간이 달라지면 만나지 못하듯 프로토콜도 서로 다른 프로토콜일 경우 소통을 하지 못하는 문제점이 발생하기도 합니다.

네트워크가 서로를 연결한 것이라면

프로토콜이라는 통신 규약으로 전세계가 약속하여 서로서로가 연결되어 있는걸까요?

그렇다면 전 세계가 어떻게 연결되어 있는걸까요?

진짜 서로서로가 연결되어 있는걸까요?

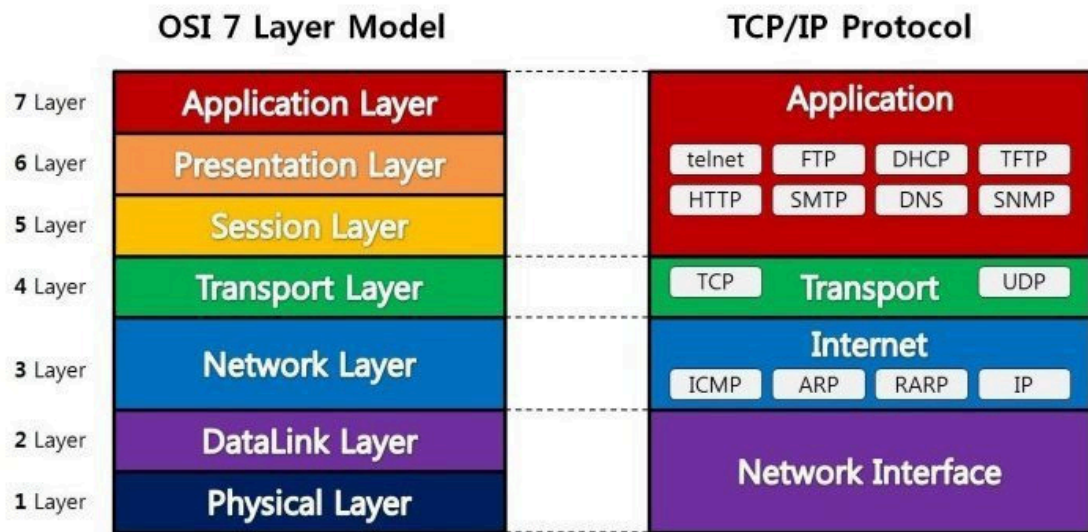
이 질문들을 쉽게 설명하기 위해서는 OSI 7계층에 대해서 알고 있어야합니다.

OSI 7계층

OSI 7계층이라는 네트워크에서 통신이 일어나는 과정을 총 7가지 단계로 나눈 것을 말한겁니다.

그렇다면 왜 나눈 것이냐?

통신이 일어나는 과정을 단계별로 한눈에 파악할 수 있기 때문입니다.



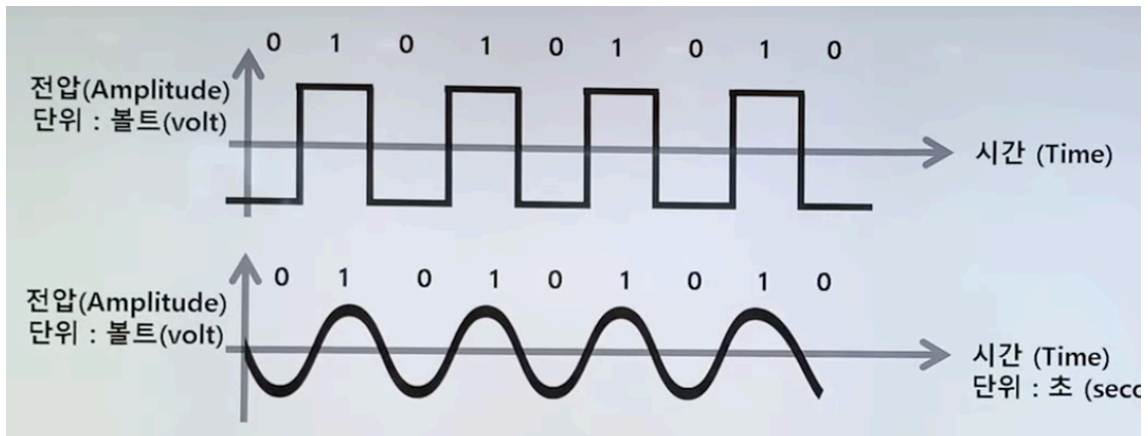
각 계층별로 어떻게 전 세계의 그 많은 컴퓨터와 스마트폰이 연결되어 있는지 차근차근 설명드리도록 하겠습니다.

1계층 - 물리계층

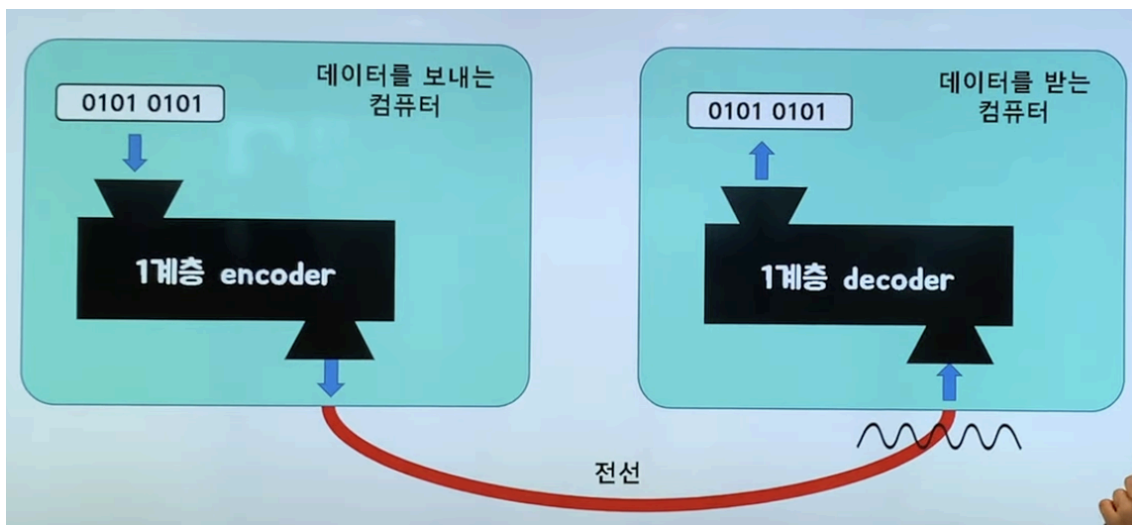
물리 계층

0과 1로 이루어진 데이터를 다른 컴퓨터로 전송하기 위해서는 "전선"을 이용하여 전송.

즉, 데이터의 전기적 신호를 전달하는 과정에서 디지털 신호를 아날로그 신호로 변경해서 전송해야 함.



1. 0과 1의 나열을 아날로그 신호로 바꾸어 전선으로 흘려보냄. (encoding)
2. 아날로그 신호가 들어오면 0과 1의 나열을 해석(decoding)
3. 물리적으로 연결된 두 대의 컴퓨터가 0과 1의 나열을 주고받을 수 있게 해주는 모듈(module)



▼ 이와 관련된 장치들



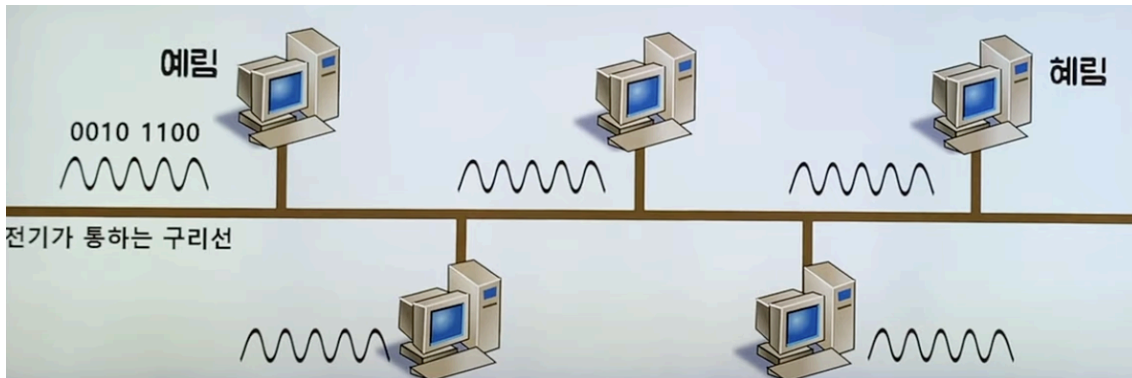
리피터



허브

위에서는 장치와 장치 사이의 통신 과정을 알게 되었다면, 여러 대의 장치들이 통신하기 위한 과정은?

데이터 링크 계층



단, 특정 사람에게 정보를 보내고 싶다면..?

각 장치마다 들고있는 고유한 주소 **"MAC 주소"** 활용

MAC 주소를 활용하여 원하는 장치에 정보를 전달하는 장치를 **"스위치"** 라고 함.



스위치

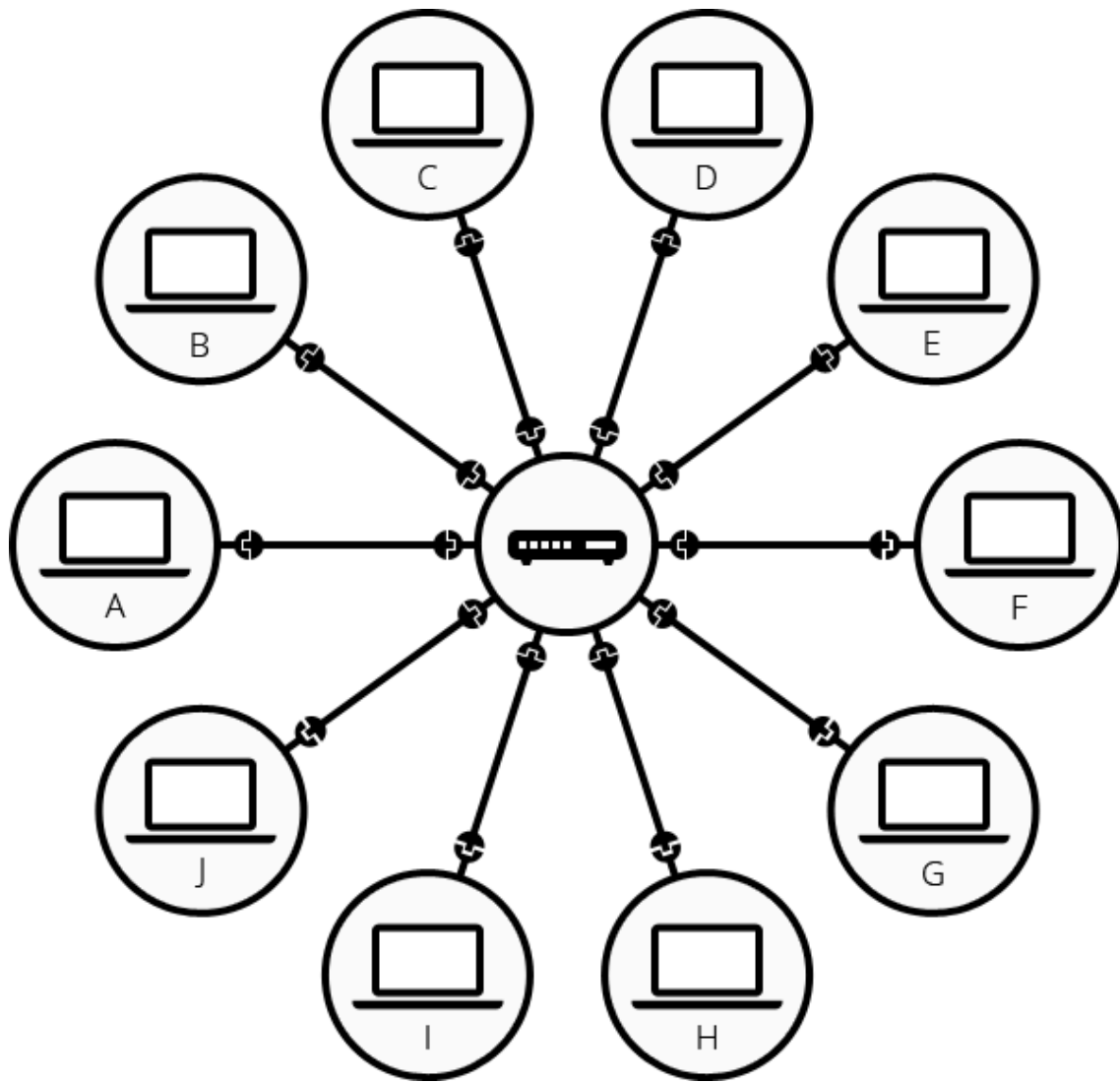
정보 단위는 **프레임**

Preamble	Start of Frame	목적지 Mac 주소	출발지 Mac 주소	length/type	data	FCS
7byte	1byte	6byte	6byte	2byte	46-1500 byte	4byte

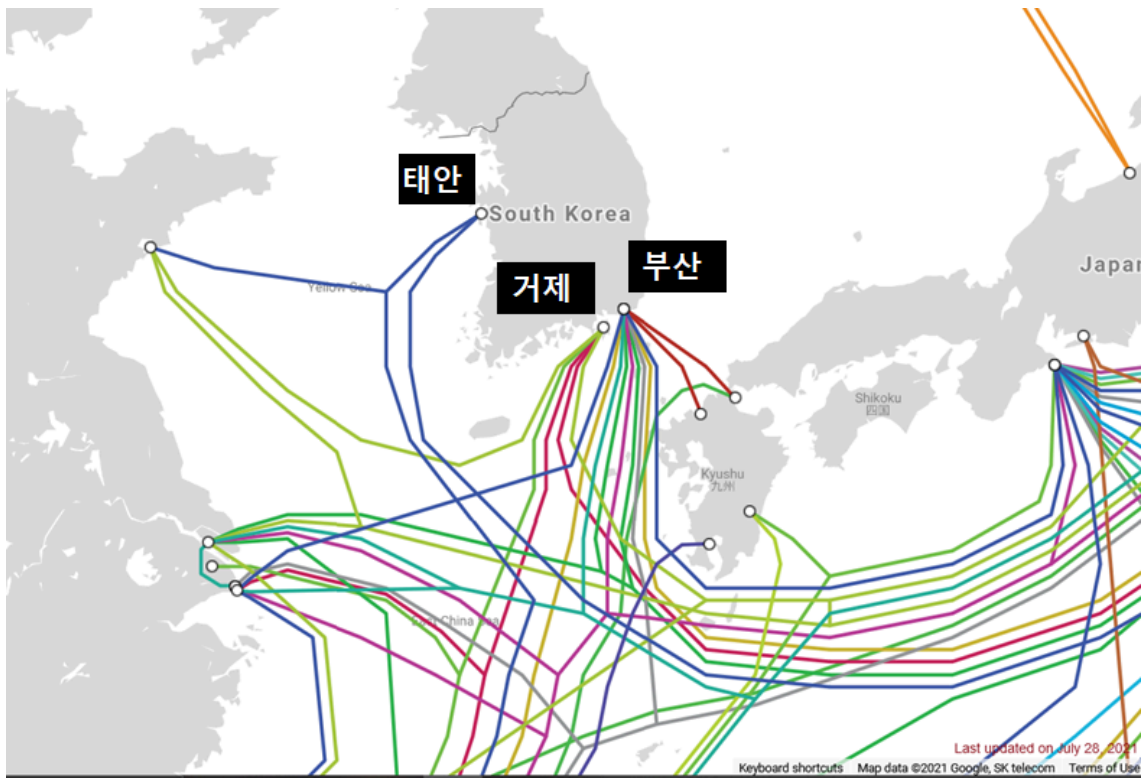
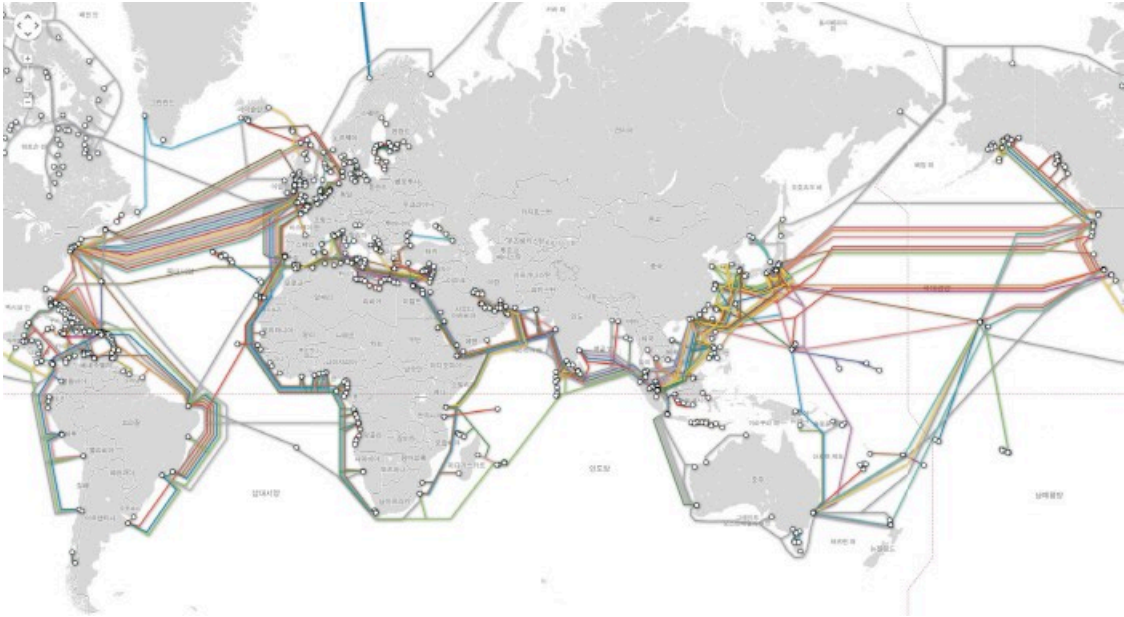
네트워크 계층

스위치 한 대에 수십 수백대의 PC를 연결하기란 불가능.

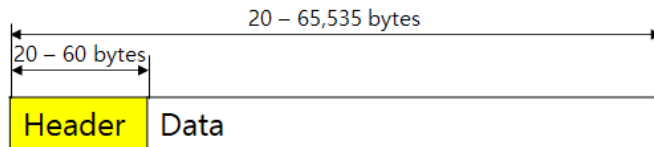
그래서 스위치와 스위치 간의 연결을 도와주는 장비가 필요해지고 네트워크가 방대해지면서 이 연결들을 **최적의 경로를 설계해주는 장비**가 필요해짐.



각 노드에 IP 주소를 부여하고, 여러 개의 노드를 거칠때마다 최적의 경로를 설정해주는 장비인 **"라우터"**를 사용.



정보 단위는 패킷.



VER 4 bits	HLEN 4 bits	Service type 8 bits	Total length 16 bits	
Identification 16 bits			Flags 3bits	Fragmentation offset 13 bits
Time to live 8 bits		Protocol 8 bits	Header checksum 16 bits	
Source IP address 32 bits				
Destination IP address 32 bits				
Option + padding 0 to 40 bytes				

수많은 네트워크들의 연결로 이루어지는 인터넷 환경 속에서

어딘가에 있는 목적지로의 데이터 전송을 위해

IP 주소를 이용해서 길을 찾고 (**Routing**)

자신 다음의 라우터에게 데이터를 넘겨주는 것(**forwarding**)

이제 모든 인터넷 상의 컴퓨터가 서로 통신 가능...!

그렇다면 어떤 데이터를 운영체제의 어떤 프로세스에게 값을 전달해줘야할지 어떻게 알 수 있을까?

전송 계층

데이터를 받고자 하는 프로세스들은 통해 **포트 번호**를 부여받고 이를 통해 원하는 정보를 주고 받음.



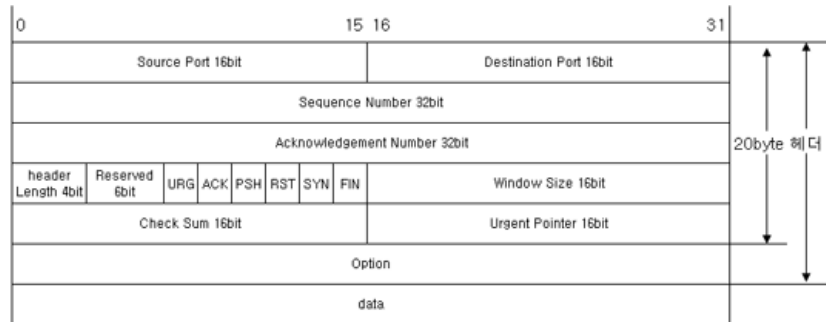
포트란?

하나의 컴퓨터에서 동시에 실행되고 있는 프로세스들이 서로 겹치지 않게 가져야하는 정수값

송신자는 데이터를 보낼 때 데이터를 받을 수신자 컴퓨터에 있는 프로세스의 포트 번호를 붙여서 보냄.

수신자는 데이터를 받을 때 해당 세그먼트의 포트 번호를 읽고 해당 프로세스에 데이터를 전달.

TCP 세그먼트 구조



(세션, 표현 계정은 운영체제가 모두 수행해줌.)

응용 계층

최종 목적지로서 HTTP, FTP, SMTP, POP3, IMAP, Telnet 등과 같은 프로토콜이 존재.

해당 통신 패킷들은 방금 나열한 프로토콜에 의해 모두 처리되며 우리가 사용하는 브라우저나, 메일 프로그램은 프로토콜을 보다 쉽게 사용하게 해주는 응용 프로그램.

한마디로 모든 통신의 양 끝단은 HTTP와 같은 프로토콜이지 응용프로그램이 아님...!!

OSI 7계층에 대해 좀 더 쉬운 설명이 필요하시다면 밑 영상을 참조하시면 좋습니다!

<https://www.youtube.com/watch?v=1pfTxp25MA8>

<https://www.youtube.com/watch?v=BEK354TRgZ8>