# Review (230703 ~ 230720)

## Network Layer

### TCP/IP 4 Layer

IP : 패킷 통신 방식의 프로토콜 / TCP : 전송 조절 프로토콜
IP는 패킷 전달 여부를 보증하지 않고, 패킷을 보낸 순서와 받는 순서가 다를 수 있다 (unreliable datagram service).
TCP는 IP 위에서 동작하는 프로토콜로, 데이터의 전달을 보증하고 보낸 순서대로 받게 해준다.

#### Link Layer

Link Layer는 호스트가 물리적으로 연결되는 링크 상에서만 운용된다. LAN이나 WAN의 인접한 네트워크 노드 간에만 운용된다. 물리적 주소인 MAC주소를 사용한다.
(프로토콜 : Ethernet, PPP, Token Ring, etc)

#### Internet Layer

Internet Layer는 네트워크 패킷을 IP 주소로 지정된 정확한 목적지로 전송하기 위해 사용된다.
(프로토콜 : IP, ARP, ICMP, RARP, OSPF, etc)

#### Transport Layer

Transport layer는 송신자와 수신자를 연결하는 통신 서비스를 제공한다. 연결 지향 데이터 스트림 지원, 신뢰성, 흐름 제어, 그리고 다중화와 같은 편리한 서비스를 제공한다.
(프로토콜 : TCP, UDP, DCCP, SCTP, etc)

#### Application Layer

특정 서비스를 제공하기 위해 애플리케이션끼리 정보를 주고 받을 수 있다.
(프로토콜 : HTTP, FTP, SSH, Telnet, DNS, SMTP, etc)

### OSI 7 layer

Main source : https://youtu.be/1pfTxp25MA8

#### Physical Layer

인코딩 : 이진수 데이터를 아날로그 신호로 변환
디코딩 : 아날로그 신호를 이진수 데이터로 변환
Physical Layer를 통해 연결된 2대의 컴퓨터가 0과 1의 나열을 주고받을 수 있다.

#### Data Link Layer

하나의 LAN 내에서 임의의 서로 다른 2개의 네트워킹 장치 간의 데이터 전송을 담당하는 계층이다. 이 계층에서 전송되는 데이터를 frame이라고 한다. IP 주소가 아닌 MAC 주소를 사용한다.

## Network Layer

IP 주소가 주어졌을 때 routing, forwarding으로 출발지에서 목적지까지 패킷을 보내는 역할을 담당하는 계층.

ICMP 같은 Network Layer 프로토콜은 Transport Layer 프로토콜과 연결되지 않는다.
(출처 : ICMP는 어떻게 작동할까요?)

### Routing vs Forwarding

Routing : 출발지에서 목적지까지 경로를 선택하는 것.
Forwarding : Routing을 통해 선택한 경로를 바탕으로 router의 입력 포트에서 출력 포트로 패킷을 전송하는 것.

## Transport Layer

Transport Layer는 크게 2가지 역할을 수행한다:

1. 3계층인 Network Layer가 데이터를 전달할 때 4계층인 Transport Layer는 데이터가 제대로 도착했는지 확인한다.
2. 데이터가 최종적으로 도착할 애플리케이션이 무엇인지 식별한다. 애플리케이션의 포트 번호에 따라 해당하는 애플리케이션에 데이터를 보낸다.

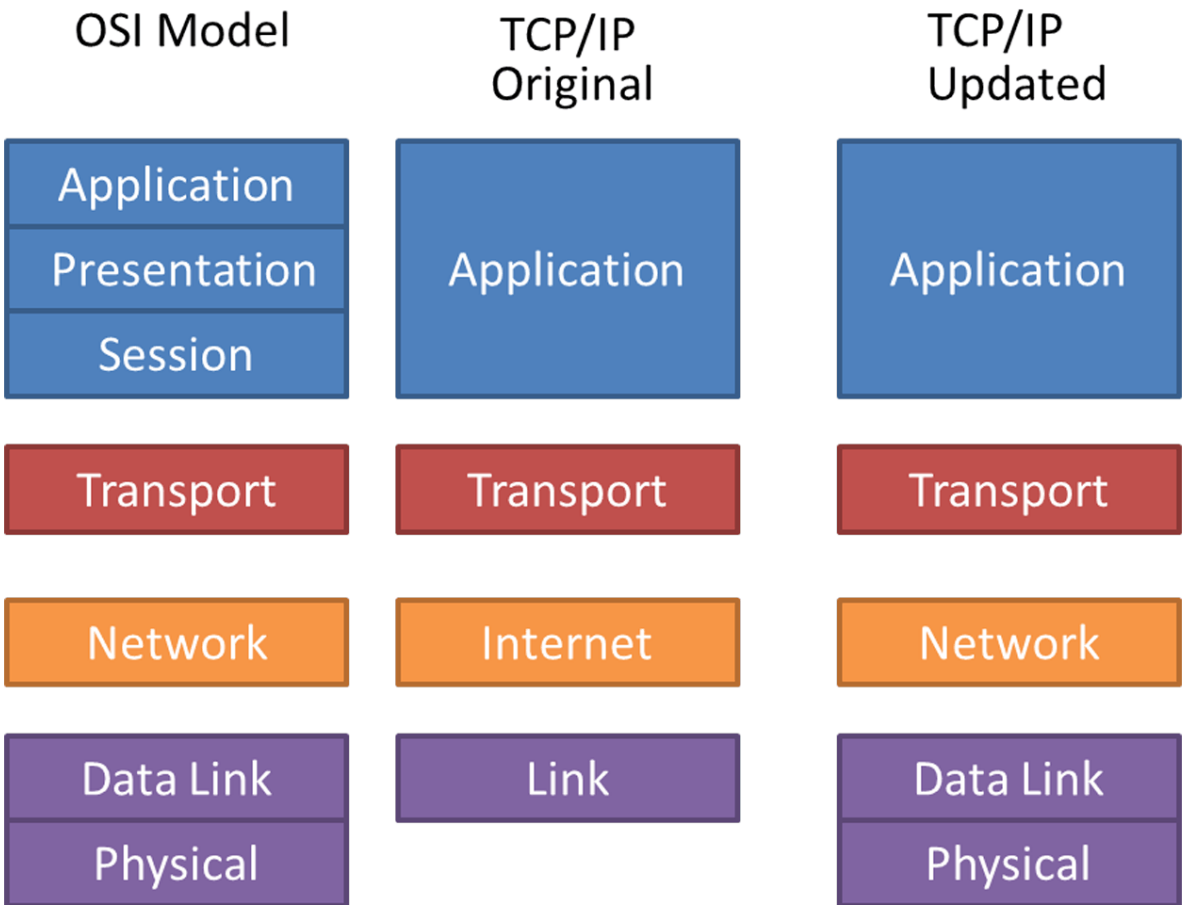
## Session Layer - 생략

## Presentation Layer - 생략

## Application Layer

특정 서비스를 제공하기 위해 애플리케이션끼리 정보를 주고 받을 수 있다.
(프로토콜 : HTTP, FTP, SSH, Telnet, DNS, SMTP, etc)

## TCP/IP Layer and OSI Layer




TCP/IP Layer와 OSI Layer는 위 그림과 같이 대응된다.
TCP/IP Updated에서는 Link가 OSI Layer와 같이 Data Link와 Physical로 구분된다.
TCP/IP Updated에서는 TCP/IP Original의 Internet Layer가 Network Layer로 바뀌었다.

### Layered Architecture

Layered Architecture라는 소프트웨어 아키텍처가 있다. (소프트웨어 아키텍처란? 예 : MVC 패턴)

TCP/IP Layer, OSI Layer는 Layered Architecture를 따르는 거대한 소프트웨어다. 인터넷을 하나의 거대한 소프트웨어로 생각할 수 있다.

# Why are there so many routers instead of a single router?

세 가지 문제가 있다:

- Load Balancing : 하나의 router만을 사용하면, 인터넷에 연결되는 모든 device를 하나의 router에 연결해야 하고, 이렇게 많은 device를 연결하는 router를 설계하는 것은 불가능하다.
- Single Point of a Failure : 모든 load를 single point에 부과해서는 안된다. 이를 Single Point of Failure라 한다. 모든 load가 부과된 하나의 router가 고장난 상황을 생각해보자. 하나의 router가 고장난 결과로 전체 internet을 사용할 수 없게 된다.
- 하나의 거대한 router에 모든 LAN을 연결하기 위해서는 어마어마하게 긴 케이블이 필요하다.

따라서 하나의 거대한 router를 사용하는 대신, 많은 router를 distributed structure로 사용해야 한다.

# HTTP

TCP/IP Layer를 따름. Application Layer 프로토콜. 클라이언트와 서버가 통신하는 방법을 표준화함.

# Browser

Browser가 어떤 웹 페이지를 띄울 때 무슨 일이 일어나는가?

1. DNS Lookup
2. TCP Handshake : TCP/IP Layer를 이용해서 통신을 하는 애플리케이션이, 데이터를 전송하기 전에, 정확한 전송을 보장하기 위해 수행하는 상대방 컴퓨터와의 사전 작업
3. TLS Negotiation : HTTPS를 통할 때 필요한 또 다른 handshake
4. HTTP Request & Response
5. Parsing : 수신한 데이터를 DOM (HTML, XML 문서를 읽고 수정할 수 있는 인터페이스) 및 CSSOM (문서의 스타일 관련 정보를 읽고 수정할 수 있는 API 세트)으로 변환.
6. Rendering : 요청한 콘텐츠를 display

# DNS

Domain Name System. DNS는 Domain Name가 주어질 때 IP Address를 찾는 기능을 수행한다.

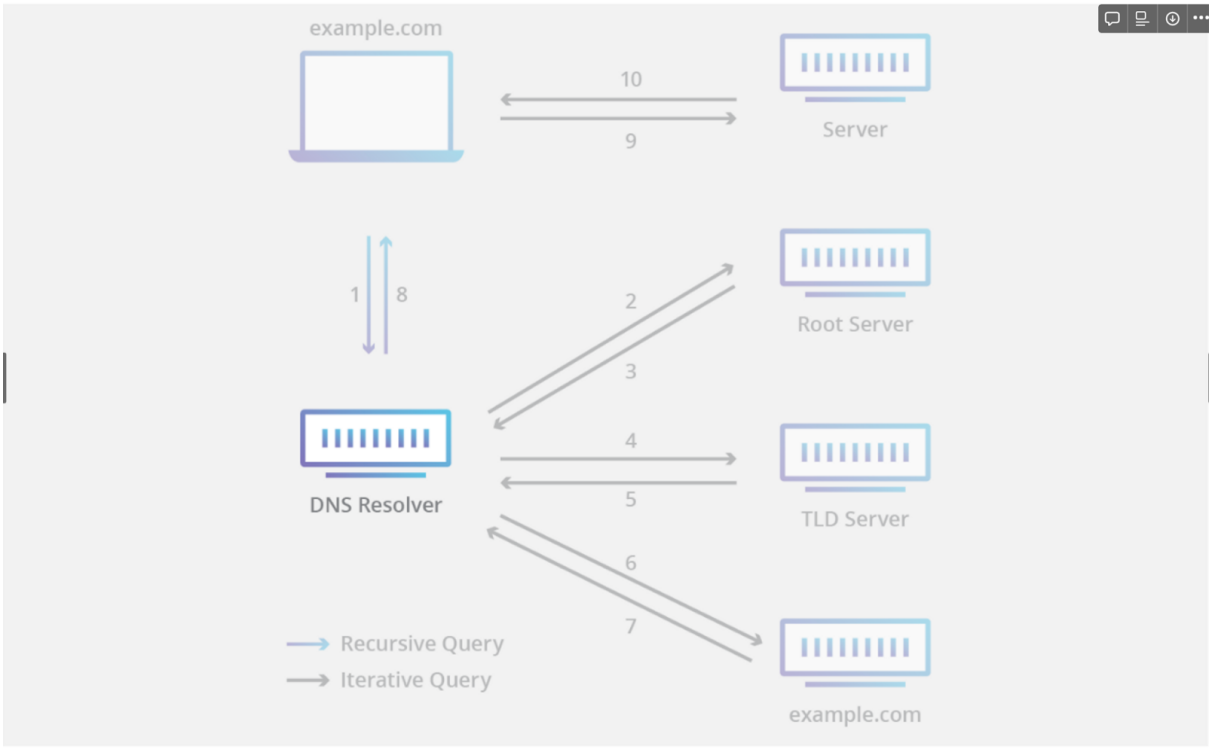
아래 내용에서 caching은 고려하지 않는다.

4 DNS Servers:

- DNS recursive resolver
- Root nameserver : DNS 조회를 시작하는 영역.
- TLD (Top-Level Domain) nameserver : URL의 마지막 점 뒤에 오는 확장자(예 : com, org, me)를 공유하는 domain name에 대한 정보 유지.
- Authoritative nameserver : Domain name의 특정한 정보가 포함됨.

사용자가 어떤 domain name에 대한 IP 주소를 알고 싶을 때:

1. 사용자가 browser에 ‘example.com’을 입력하면, query가 인터넷으로 이동하여 DNS recursive resolver에게 전달된다.
2. Resolver는 DNS root nameserver에 요청한다.
3. Root server가 TLD DNS server의 주소로 응답한다.
4. Resolver는 TLD DNS server에 요청한다.
5. TLD server는 domain’s nameserver인 example.com의 IP 주소로 응답한다.
6. 마지막으로, recursive resolver가 domain’s nameserver에 query를 보낸다.
7. example.com의 IP 주소가 nameserver에서 resolver로 전달된다.
8. DNS resolver가 web browser에게 IP 주소를 응답한다.


example.com
10
9
Server
1
8
2
3
Root Server
DNS Resolver
4
5
TLD Server
6
7
Recursive Query
Iterative Query
example.com