네트워크

1. HTML & Internet

1.1. HTML이란?

HTML은 Hyper Text Mark Language의 약자로써 월드와이드웹 문서를 작성하는 Markup Language이다. HTML은 여러 태그들로 구성되어 있으며 각 태그들을 사용하여 원하는 형태의 문서를 만들 수 있다.

1.2. HTML의 장점.

쉽다. 그리고 중요하다.(웹페이지는 우리가 매일 보는 것이고, 이를 만드는 언어가 HTML이기 때문에.)

1.3. HTML의 문법.

TAG(태그)와 데이터로 구성되어 있다. : 태그란 '<>' 두 괄호 안에 설정된 명령어로 HTML문서의 모양이나 행동양식을 정해주는 구성요소이다. 보통 다른 문장들과 구별되며 서로 쌍을 이루는 형식이다. 그 안에는 Element(엘리먼트 : 일반적으로 태그라 칭함) 라는 속성 및 속성 값이 들어간다.

- 태그는 대소문자의 구분이 없다.
- HTML파일의 확장자는 *.htm, *.html로 저장됨
- 기본적으로 공백, 탭, 엔터키가 적용되지 않는다.
- 공백이나 줄바꿈은 특수기호나 태그를 사용한다.
- 태그를 중첩해서 사용할 경우 열어준 순서와 반대로 가장 마지막에 입력한 태그부터 닫는다.
- **<태그명 속성="값"속성="값">내용</태그명>**
- <태그명>이 시작을, </태그명>이 끝을 알리며, 이 사이에 실제로 출력 될 값이 입력된다.
- HTML 의 기본구조

```
<!DOCTYPE html>
 <html>
      <head>
         <meta charset="utf-8">
           <title> 문서의 제목이 들어 갑니다.</title>
      </head>                                          HEAD 부분
      <body>
       html 문서의 본문 내용이 들어갑니다.
      </body>                                          BODY 부분
 </html>
```

> - <!doctype> : 현재 웹 문서가 어떤 HTML버전에 맞게 작성되었는지를 알려준다.
> - <html> ~ </html> : HTML문서의 시작부분과 마지막 부분에 태그 사용
> - <meta> : 웹 페이지를 만들 때 필요한 정보(메타 정보) 지정
> - <title> : 웹 페이지 제목 지정
> - <body>~ </body> : 웹 페이지의 내용

1.4. Internet

서로 통신할 수 있는 둘 또는 그 이상의 네트워크 집합을 internet(i로 시작)이라고 하며, 가장 대표적인 것이 Internet(I로 시작)이다. Internet은 수천개의 **상호 연결되어있는 네트워크들**로 이루어져 있다.

- 웹 브라우저/ 웹 서버 **웹 브라우저** : HTML 문서와 그림, 멀티미디어 파일등 월드 와이드 웹을 기반으로 한 인터넷의 컨텐츠를 검색 및 열람하기 위한 응용 프로그램의 총칭. **웹 서버** : 서버에 접속한 사용자에게 웹 서비스를 제공하기 위하여 사용되는 서버의 한 종류.

- WWW(World wide web) 인터넷에 연결된 컴퓨터를 이용해 사람들과 정보를 공유할 수 있는 거미줄처럼 얼기설기 엮인 공간을 뜻한다. HTTP프로토콜을 기반으로 HTML로 작성 된 하이퍼텍스트 페이지를 웹 브라우저라는 특정한 프로그램을 읽을 수 있게 하도록 구성되어 있다. WWW의 5가지 기능으로는

> - Universal Readership: 하나의 플랫폼으로 다양한 데이터베이스/환경에 접근하여 필요한 정보를 검색, 수집할 수 있는 기능
> - Hypertext: 하이퍼텍스트 링크를 통한 문서간의 연결 기능
> - Searching: 방대한 문서에서 필요한 단어 / 부분을 찾을 수 있는 기능
> - Client-Server Model: 중심에서 흐름을 관리하는 관리자나 관리기능이 존재하지 않으며 누구라도 문서를 제작하고 읽을 수 있는 기능
> - Format negotiation: 공용화할 수 있는 표시 언어. 즉 HTML. 이 있다.

1.5. Client/Server

클라이언트(client)는 클라이언트-서버(client-server) 구성에서 **사용자가 서버에 접속하기 위해 사용하는 프로그램 또는 서비스**를 말한다. 서버(server)는 **클라이언트에게 네트워크를 통해 서비스하는 컴퓨터**를 의미한다

- 호스트 서버-클라이언트에서 변형된 구성이며, 모든 서버는 호스트이지만 모든 호스트가 서버인 것은 아니다. 네트워크에 연결이 확립된 모든 장치는 호스트의 자격이 있는 반면, 다른 장치(클라이언트)로부터의 연결을 수락하는 호스트만 서버가 될 수 있다.

2. 컴퓨터 네트워크란?

각각의 컴퓨터들이 자원을 공유할 수 있게 하는 **디지털 전기통신망의 하나**이다. 즉, 분산되어 있는 컴퓨터들을 하나의 통신망으로 연결 한 것으로, 각각의 컴퓨터들 전화선, 동축 케이블, 위성통신 ,무선 등 다양한 통신 기술로 연결해 놓은 것이다. 이를 통해 각 컴퓨터 간에 데이터를 송신, 수신하는것이 가능해 진다. ###2.1. 활용분야

- 자원 공유 (Resource sharing)

> - 하드웨어(하드디스크, 프린터, 컴퓨팅)
> - 소프트웨어(어플리케이션 소프트웨어)

- 정보 공유 (Information Sharing)

> - 어느곳에서나 손쉬운 접근(파일, 데이터베이스)
> - 검색 및 정보 조회(웹 컨텐츠, 포털, SNS, 검색 엔진)

- 통신(Communication)

> - 전자우편
> - 메시지 브로드캐스팅
> - 원격 컴퓨팅
> - 네트워크를 통한 원격 컴퓨터 제어

- 분산 컴퓨팅(Distribute processing or GRID computing)

> - 대량의 데이터를 처리하기 위해 수많은 컴퓨터를 조합(예시: CETI 프로젝트)

2.2. 네트워크 구성방식

- 피어 투 피어(peer to peer, P2P) : 컴퓨터들이 서로 동등한 권한으로 연결되어 컴퓨터의 자원을 서로 공유하는 방식. 별도의 서버 없이 각 컴퓨터가 서버의 역할을 하기도 하며, 클라이언트 역할을 하기도 한다.
- 호스트 터미널 방식 : 정보처리가 호스트에서 이루어진다. 터미널에서는 독자적으로 데이터를 처리할 수 없고 단지 입출력만 할 수 있다.
- 클라이언트 - 서버방식 : 클라이언트는 서버에 작업을 요청, 서버는 요청된 작업을 처리하여 클라이언트에게 제공하는 방식.

2.3. 네트워크 토폴로지(망 구성방식)

컴퓨터 네트워크의 요소들(링크, 노드 등)을 물리적으로 연결해 놓은 것, 또는 그 연결 방식을 말한다

- 버스 토폴로지 : 버스라 불리는 공유 통신 경로를 통해 연결된 클라이언트의 집합을 가리키는 네트워크 구조

> - 모든 노드(node)들은 간선을 공유하며 버스 T자형(Tap) 등으로 연결
> - 간선과 각 단말 장치와의 접속은 간단한 접속장치를 붙이는 것으로 가능

- 망 토폴러지 : 단말기 또는 컴퓨터를 다른 모든 단말기와 서로 연결시킨 형태.

> - 네트워크상의 모든 노드를 상호 연결
> - 통신선로의 총길이가 가장 긴 네트워크 구조
> - 초기 데이타 통신 네트워크의 전형적인 형태

- 트리 토폴로지 : 최상위에 중안 컴퓨터를 중심으로 단계적으로 하위, 상위 개념을 계층적으로 적용한 형태

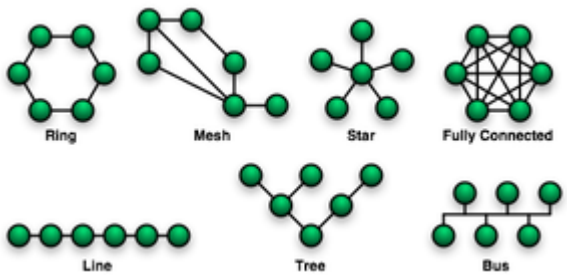
> - 지역과 거리에 따라 연결하므로 통신선로의 총경로가 가장 짧음
> - 접속되는 단말기의 숫자에 맞는 통신장비 이용이 가능

- 링 토폴로지 : 인접해잇는 양 옆의 두 노드를 연결하는 단방향 전송 형태

> - 각 링크가 단방향이어서 데이타는 한 방향으로만 전송()
> - 각 노드는 데이타의 송수신을 제어하는 엑세스 제어논리(토큰)을 보유

- 성형 토폴로지 : 중앙에 있는 노드(허브,스위치 등)에 각 노드들을 직접 연결시킨 형태

> - 중앙집중식 구조
> - 중앙의 교환장비가 데이타 경로를 개설하고 유도



3. LAN / WAN

LAN과 WAN은 네트워크의 규모 혹은 연결 범위에 따라 구분하는 방식이다.

3.1. LAN

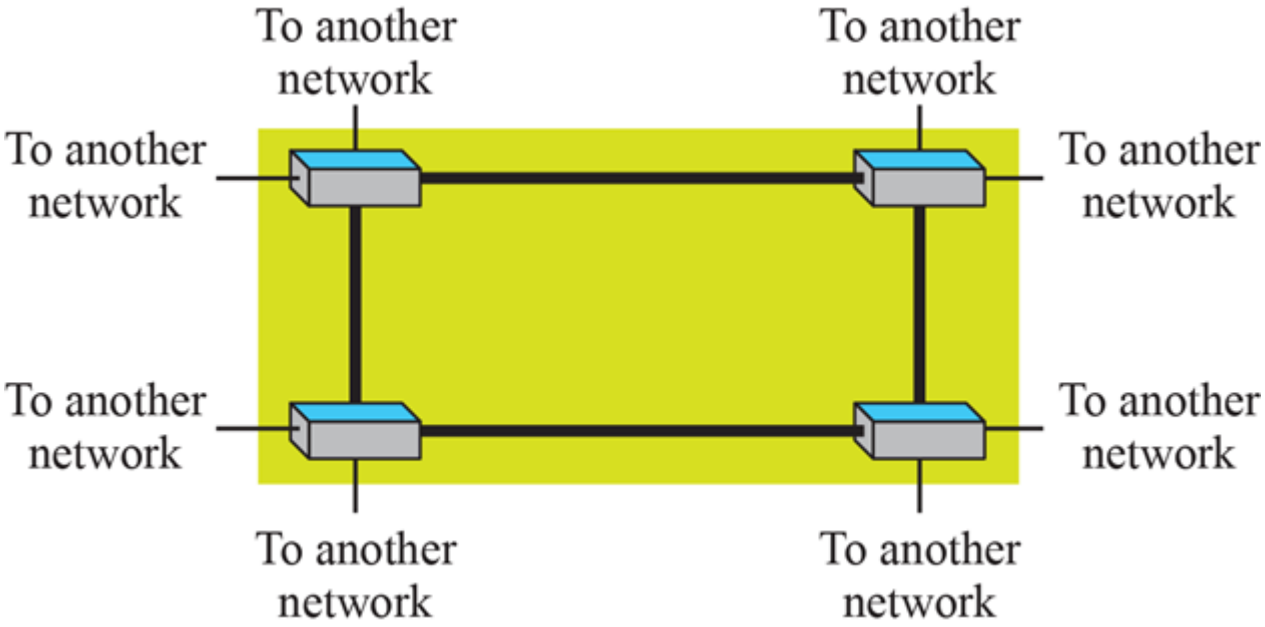
근거리통신망(LAN : local area network)은 개인 소유이거나 단일 사무실, 건물 혹은 학교 등에 있는 호스트들을 연결한다. 조직의 요구와 사용되는 기술의 종류에 따라 LAN은 개인 사무실에서 2대의 PC와 프린터를 연결하는 데 사용되거나 회사 전체로 확대될 수 있고, 음성, 음향, 비디오 장치를 포함 할 수 있다. 연결 방식으로는 공통 케이블, 연결 교환기 등을 사용한다.

3.2. WAN

광역통신망(WAN : wide area network)도 통신이 가능한 장치들의 상호 연결이다. 하지만 LAN과는 다르게 넓은 지리적은 크기를 갖는 도시나 주, 국가, 또는 세계에 사용된다. 교환기, 라우터 또는 모뎀과 같은 연결 장치를 사용하여 장비를 연결하고, 일반적으로 통신회사가 만들고 이를 임대하기 위한 목적으로 사용한다.

- 점-대-점 WAN 전송매체(케이블 또는 공기)를 통해 두 통신 장치를 연결하는 네트워크.

- 교환형 WAN 2개보다 더 많은 종단점을 가진 네트워크로, 오늘날 세계적인 통신 백본망에 사용된다.



3.3. 네트워크간 연결

요즘은 LAN이나 WAN이 독립적으로 분리되어 있는 것은 매우 드문 경우이며, 일반적으로 그들은 서로 연결되어 있다. 2개 이상의 네트워크가 연결 될 경우,internet 또는 네트워크간 연결을 구성하게 된다.

4. 회선교환방식 / 패킷교환방식

인터넷은 교환기가 적어도 2개의 링크를 연결하는 교환형 네트워크이다. 교환기는 필요할 때 한쪽 링크에서 다른 쪽 링크로 데이터를 포워딩한다.

4.1. 회선교환방식

교환기를 통해 통신회선을 설정하여 직접 데이터를 교환하는 방식이다. 특징으로는 회선이 이미 설정되어 있으므로 데이터는 항상 동일한 경로로 가게 된다. 이러한 것을 '독점'이라고 표현한다. 회선교환방식은 데이터 전송이 없어도 회선이 연결된 상태이기 때문에 직접 회선 해제 요청을 하지 않는 이상 계속 접속을 유지하게 된다. 이 방식은 **대용량의 데이터를 고속으로 전송**할 때나, **연속적인 전송을 할 때 이점**이 있으며, **고정적인 대역폭을 사용**한다. 하지만 데이터를 전송하지 않을 때에는 회선이 독점되어 회선이 이용되지 않기 때문에

- 회선이용률 면에서는 비효율적이다
- 연결된 두 장치가 반드시 같은 전송률을 필요로 한다
- 속도나 코드의 교환이 불가능하다
- 속도나 코드의 변환이 불가능하다
- 사용자가 직접 에러제어나 흐름제어를 수행하여야 한다

이로 인해, **실시간 전송보다 에러없는 데이터 전송이 요구되는 구조에서는 부적합**하다.

4.2. 패킷교환방식

송신측에서 모든 메시지를 일정한 크기의 패킷으로 분해/전송하는 방식으로 수신측에서는 메시지를 받아 원래의 메시지로 재조립하는 방식이다. 패킷교환방식을 이용하기 위해서 패킷 다중화, 논리채널, 경로선택제어, 순서제어, 트래픽제어, 오류제어를 필요로 한다.

- 순서 제어는 목적지에서 송신된 패킷의 순서와 수신된 패킷의 순서를 재정렬하는 것을 의미
- 트래픽제어는 흐름제어, 혼잡제어, 교착상태를 제어하는것을 의미
- 경로선택제어는 성능과 시간, 장소를 통해 경로를 배정하는 것을 의미
- 오류제어는 오류검사를 실시하여 오류 발생 시 재전송이 가능한 것을 의미

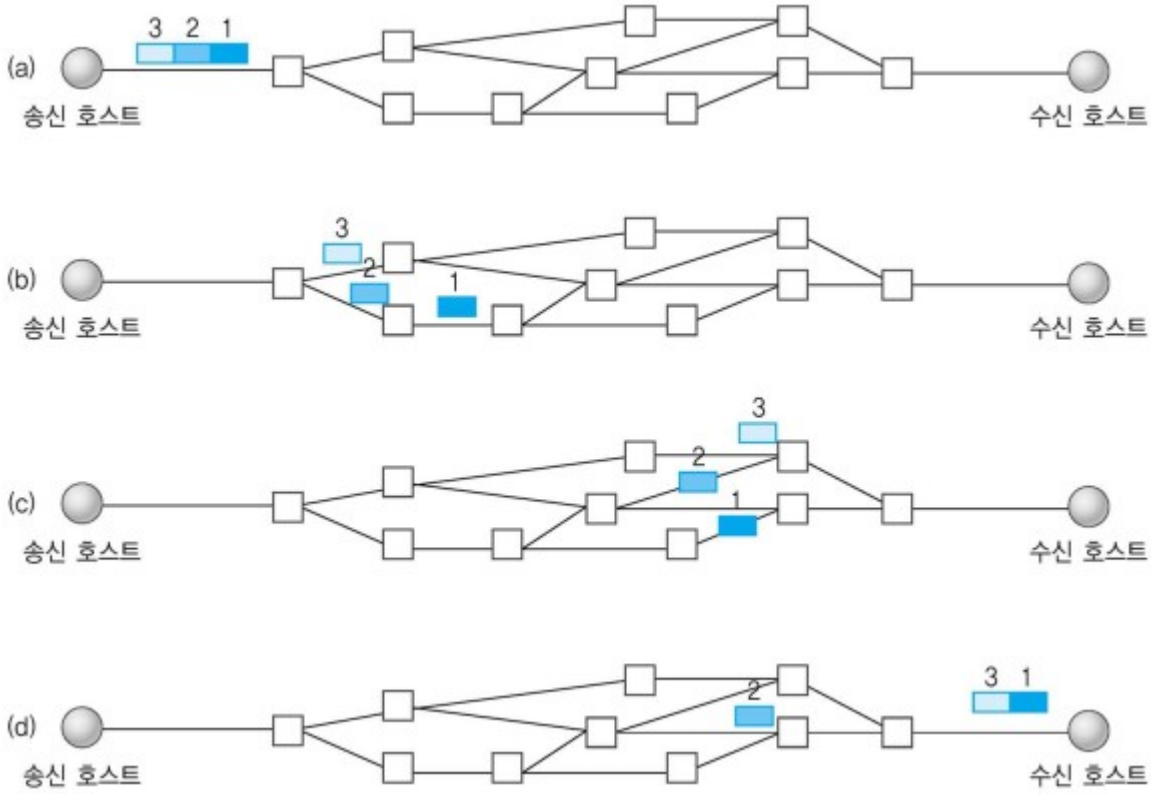
이런 점에서 패킷교환 방식은 **회선 이용률이 높고**, **속도와 프로토콜 변환이 자유롭다**. 또한 상황에 따라 **교환기 및 회선등의 장애가 발생해도 패킷의 우회전송이 가능**하여 전송의 신뢰성이 높고, 디지털 전송이므로 **오류검사를 실시하여 오류 발생 시 재전송이 가능**하다. 그리고 **다중화를 통해 사용 효율이 높고, 다른 방식의 단말장치여도 전송속도와 프로토콜을 교환망이 변환처리를 하여 송수신할 수 있게 합니다**.이런한 특징으로 인해 패킷교환 방식은 쉽게 **고신뢰성, 고품질, 고효율, 다른기종간 통신의 특성**을 가질 수 있습니다. 하지만 단점도 존재하는데, **교환기를 통해서 전송하므로 교환기마다 다소의 지연시간이 발생**하며, 패킷에 **경로와 에러검출을 위한 헤더가 추가되어 오버헤드 발생 가능성**이 존재합니다.

패킷교환방식에는 다음과 같은 2가지 방식이 존재한다.

4.2.1. 데이터그램 방식 : 비연결 서비스

각 전송 패킷을 **미리 정해진 경로가 없이 독립적으로 처리**하여 교환한다. 즉, 같은 목적지의 패킷도 같은 경로를 거치지 않고 서로 다른 경로를 통해서 목적지에 도달하게 된다. 이러한 점은 망의 한 부분이 혼잡할 때 전송 패

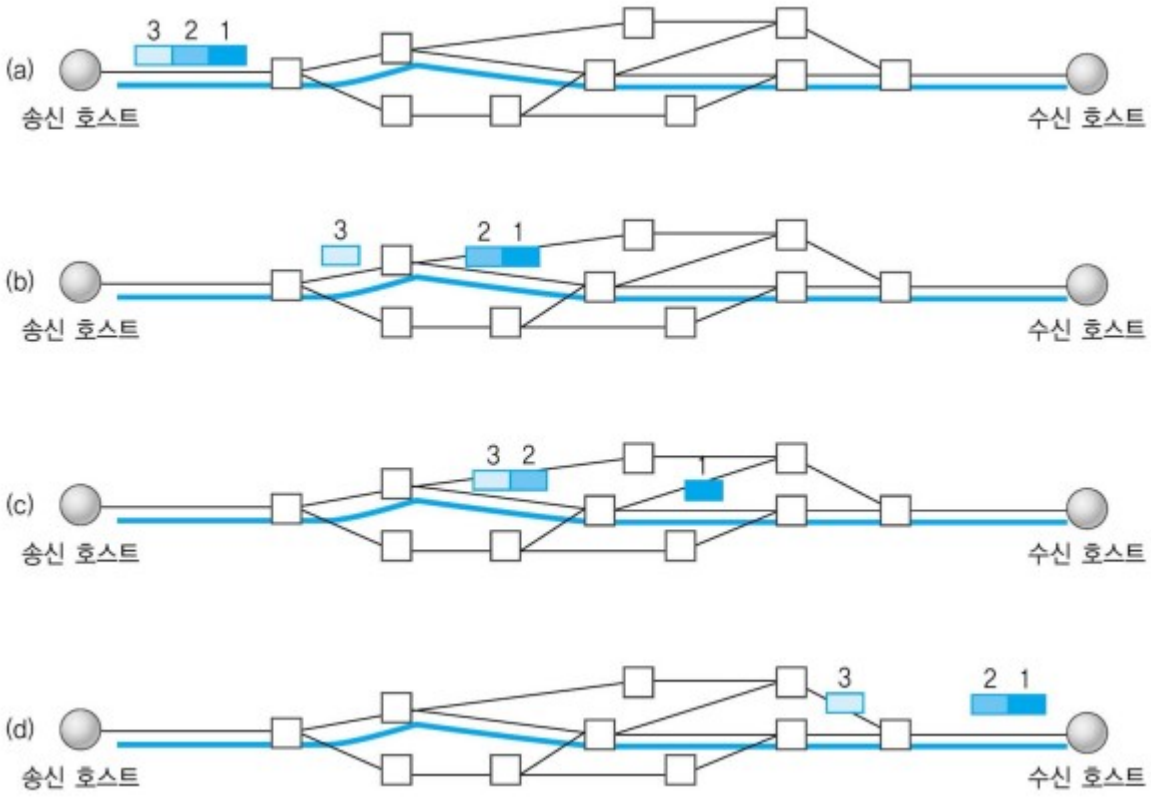
킷에 다른 경로를 배정할 수 있게 하며, **융통성 있는 경로를 설정**한다. 가상회선은 특정 교환기가 고장났을 시, 모든 패킷을 잃어버리게 되지만, 고장난 경로를 피해서 전송할 수 있으므로 더욱 신뢰할 수 있게 된다. 짧은 메시지의 패킷을 전송할 때 효과적이다.



4.2.2. 가상 회선 방식 : 연결지향형 서비스

패킷을 전송하기 전에 논리적인 연결을 먼저 수행한다. 가상회선의 논리적인 연결은 호출요구를 하고 호출 수신 패킷을 주고받는 방식이다. 이것을 X.21방식이라고 하며, 가상회선 방식을 **연결 지향형**이라고 한다. 이 과정을 보면 가상회선방식은 회선교환 방식처럼 사용되는 것을 알 수 있는데, 교환기에 패킷이 일시적으로 저장되어 **일정한 전송률을 보장**할 수 있다. 이러한 특징으로 비교적 긴 메시지의 전송 시 더 효과적이다. 이미 확립된

접속을 끝내기 위하여 Clear Request패킷을 이용한다.



	회선교환방식	패킷교환방식
패킷 교환 방식	데이터의 순서 보장	데이터의 순서가 어긋날 수 있다
오 류	오류가 발생 시 전송 실패	전송 우회 가능하므로 오류에 강함
전송시간	전파지연시간만 소비(빠름)	전파지연시간,큐잉지연(느림)

5. 프로토콜이란?

정보기기 사이 즉 컴퓨터끼리 또는 컴퓨터와 단말기 사이 등에서 정보교환이 필요한 경우, 이를 원활하게 하기 위하여 정한 여러 가지 통신규칙과 방법에 대한 약속 즉, 통신의 규약을 의미한다.

5.1. 프로토콜의 구성

물리적 측면과 논리적 측면으로 나뉜다.

- 물리적 측면 : 자료 전송에 쓰이는 전송 매체, 접속용 단자 및 전송 신호, 회선 규격 등.
- 논리적 측면 : 프레임 구성, 프레임 안에 있는 각 항목의 뜻과 기능, 자료 전송의 절차 등

> - 폐쇄적인 프로토콜 : 자사 장치들끼리 통신하기 위한 독자적인 통신 규약. 자세한 규격이 공개되지 않아서 크래킹 위협에 상대적으로 안전하다.
> - 공개된 범용 프로토콜 : 여러 장치들에 쓰이는 널리 알려진 규격이며, 규격이 널리 공개되어 있기 때문에 컴퓨터와 네트워크 크래킹에 취약한 편이다. ###5.2. 프로토콜의 종류

- HTTP : WWW 상에서 정보를 주고받을 수 있는 프로토콜
- HTTPS : WWW 통신 프로토콜인 HTTP의 보안이 강화된 버전
- FTP : TCP/IP 프로토콜을 가지고 서버와 클라이언트 사이의 파일 전송을 하기 위한 프로토콜

- SFTP : 신뢰할 수 있는 데이터 스트림을 통해 파일 접근, 파일 전송, 파일 관리를 제공하는 네트워크 프로토콜
- Telnet : 인터넷이나 로컬 영역 네트워크 연결에 쓰이는 네트워크 프로토콜
- POP3 : 응용 계층 인터넷 프로토콜 중 하나로, 원격 서버로부터 TCP/IP 연결을 통해 이메일을 가져오는데 사용
- SMTP : 인터넷에서 이메일을 보내기 위해 이용되는 프로토콜
- SSH : 네트워크 상의 다른 컴퓨터에 로그인하거나 원격 시스템에서 명령을 실행하고 다른 시스템으로 파일을 복사할 수 있도록 해 주는 응용 프로그램 또는 그 프로토콜
- SSL : 인터넷 같이 TCP/IP 네트워크를 사용하는 통신에 적용되며, 통신 과정에서 전송계층 종단간 보안과 데이터 무결성을 확보해주는 프로토콜
- SOAP : 일반적으로 널리 알려진 HTTP, HTTPS, SMTP 등을 통해 XML 기반의 메시지를 컴퓨터 네트워크 상에서 교환하는 프로토콜
- ARP : 네트워크 상에서 IP 주소를 물리적 네트워크 주소로 대응(bind)시키기 위해 사용되는 프로토콜

6. TCP/IP & OSI 7 모델

TCP/IP와 OSI 7계층을 알기 위해선 우선 **프로토콜 계층화**가 무엇인지 알아야 한다.

- 프로토콜 계층화 : 통신이 복잡할 때 각 계층마다 프로토콜이 필요할 때, 프로토콜 계층화로 서로 다른 계층 간에 임무를 나눌 수 있다.
- 프로토콜 계층화의 원칙

> 1. 양 방향 통신을 원한다면, 각 계층이 각 방향으로 한 가지씩, 상반되는 두 가지 작업을 수행할 수 있도록 만들어야 한다.
> 2. 양 측의 각 계층에 있는 객체는 서로 동일해야 한다.

6.1. TCP/IP 모델

TCP/IP는 현재의 인터넷에서 사용하는 프로토콜 그룹이다. 상호작용하는 모듈로 이루어진 계층적 프로토콜인데, 각 모듈은 특정한 기능을 제공한다. 계층적이라는 말은 각 상위 계층 프로토콜은 1개 이상의 하위 계층 프로토콜로부터 제공되는 서비스들의 자원을 받는다는 의미이다. 현재 TCP/IP모델은 4계층(5계층일시 : 네트워크 접근계층 -> 물리층, 데이터링크층) 모델로 간주된다. 각 계층의 설명은 아래 OSI 7 계층에서 설명한다.

6.2. OSI 7 모델

모든 유형의 컴퓨터 시스템 간의 통신을 허용하기 위해 정의된 7계층 표준 네트워크 모델.

- 계층 모델

> - 네트워크 지원 계층(물리, 데이터링크, 네트워크계층) : 한 장치에서 다른 장치로 데이터가 이동할 때 필요한 기능 처리
> - 트랜스포트 계층(전송계층) : 종단 대 존단간 신뢰성 있는 데이터 전송을 보장
> - 사용자 지원 계층(응용계층, 표현계층, 세션계층) : 사용자 서비스를 정의

- 물리층 : 물리적 매체를 통한 비트 스트림 전송에 요구되는 기능을 담당.
- 데이터링크층 : 한 노드에서 다른 노드로 프레임을 신뢰성 있게 전송하는 책임을 가짐.

> 기능
>
> - 프레임구성: 네트워크 계층으로부터 받은 비트 스트림을 프레임 단위로 나눔

> - 물리주소지정: 송신자와 수신자의 물리 주소를 헤더에 추가
> - 흐름제어: 수신자의수신데이터전송률을고려하여데이터전송하도록제어
> - 오류제어: 손상 또는 손실된 프레임을 발견/재전송, 트레일러를 통해 이루어짐
> - 접근제어: 주어진 어느 한순간에 하나의 장치만 동작하도록 제어

- 네트워크계층 : 발신지 호스트로부터 최종 목적지 호스트로 패킷을 전달하는 책임을 갖는다

> 기능
>
> - 발신지-대-목적지 전달
> - 논리 주소 지정 : 상위 계층에서 받은 패킷에 발신지와 목적지의 논리주소를 헤더에 추가
> - 라우팅 : 패킷이 최종 목적지에 전달될 수 있도록 경로를 지정하거나 교환

- 전송계층 : 메시지의 프로세스 대 프로세스 전달에 대한 책임을 가짐

> 기능
>
> - 포트 주소 지정 : 포트 주소를 포함 네트워크 계층은 데이터를 목적지 컴퓨터에, 전송 계층은 해당 컴퓨터의 정확한 프로세스에게 전달
> - 분할과 재조립 전달 가능한 세그먼트 단위로 나누며, 각 세그먼트는 순서번호를 가지며 이를 통해 재조립 또는 패킷의 손실여부를 판단한다
> - 연결제어
> - 흐름제어
> - 오류제어

- 세션계층 : 대화 제어와 동기화에 책임을 갖는다.

> 기능
>
> - 세션관리
> - 동기화
> - 대화 제어
> - 원활한 종료

- 표현계층 : 변환, 압축, 암호화에 책임을 갖는다.
- 기능

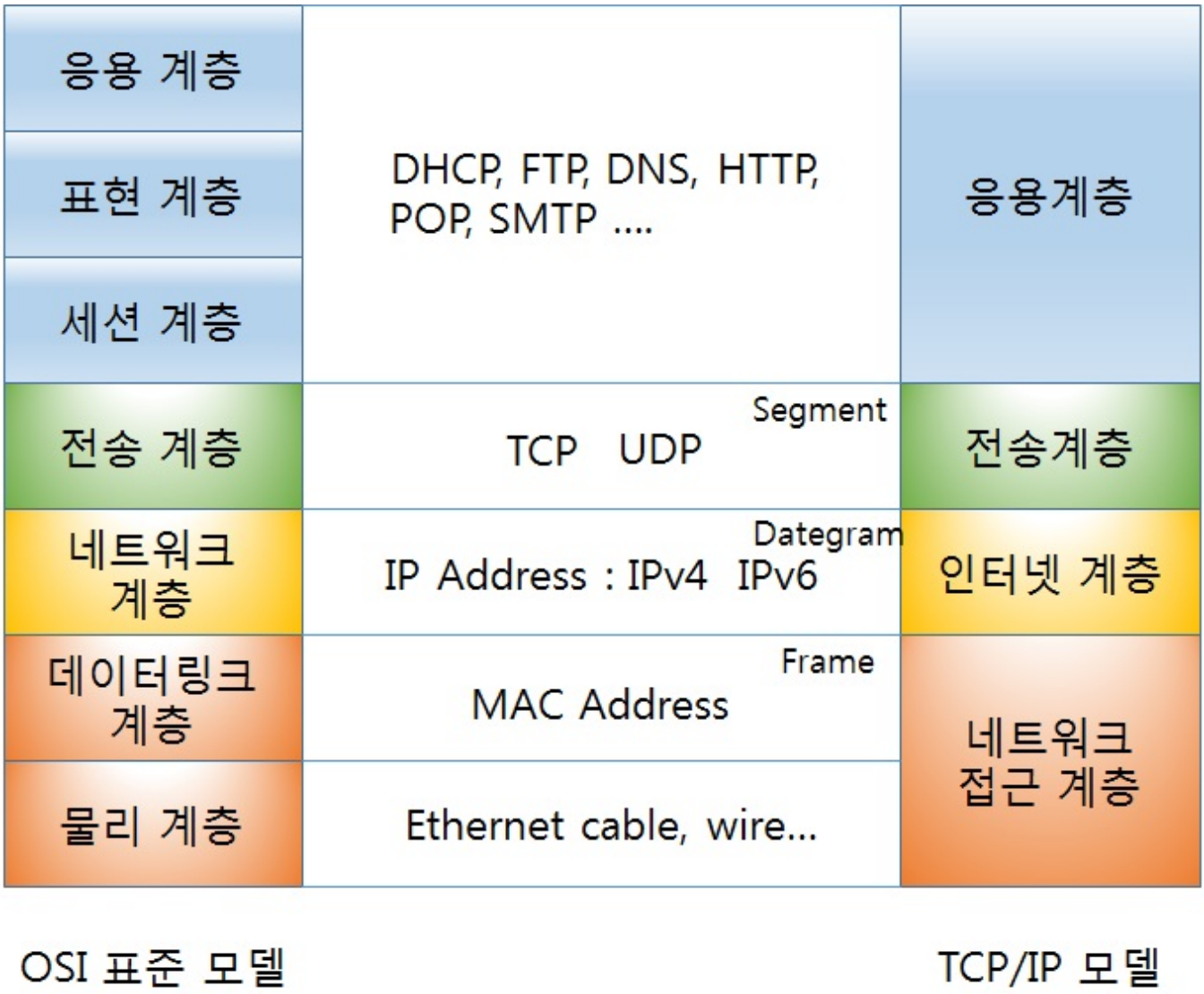
> - 변환
> - 암호화
> - 압축
> - 보안

- 응용계층 : 사용자에게 서비스를 제공하는 책임을 진다.(사용자 인터페이스 제공)

> 서비스
>
> - 원격 로그인
> - 파일 액세스, 전송, 관리
> - 우편 서비스 WWW 접근 : 웹 접근

6.3. OSI 7 vs TCP/IP

TCP/IP 계층은 OSI 7계층을 더 단순화 시켜서 4개의 계층(Layer)로 만들어서 사용한다



OSI 모델은 TCP/IP모델 이후에 나와 모든 TCP/IP모델을 대체할 것이라 여겨졌지만, TCP/IP모델이 완전히 자리잡고, OSI모델의 일부 계층은 완전히 정의되지 않았으며, 다른 응용의 협회에 의해 구현되었을 때 그것은 TCP/IP의 프로토콜로부터 충분히 높은 수준의 성능을 보여주지 못하였기 때문에 실패하게 되었다.

7. TCP / UDP

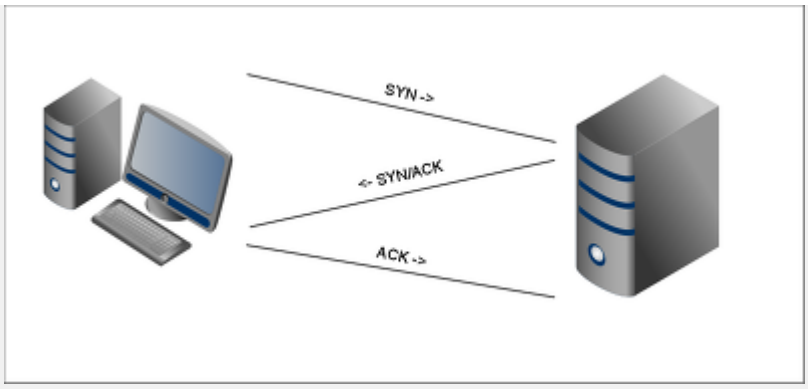
전송층에서 데이터를 전송하기 위한 프로토콜이라는 점에서 TCP와 UDP는 비슷하지만 특성에 있어서 약간의 차이가 있다.

7.1. TCP

연결지향형이며, 자체적으로 오류를 처리하며, 네트워크 전송중 순서가 뒤바뀐 메시지를 교정해주는 기능을 가지고 있다. 연결지향형이란 말은 데이터를 전송하는 측과 데이터를 전송받는 측에서 전용의 데이터 전송 선로(세션)을 만든다는 의미니다. 데이터의 신뢰도가 중요하다고 판단될 때 사용한다.

- 동작방식

> TCP는 패킷을 성공적으로 전송하면 Acknowledgement(ACK) 라는 신호를 보낸다. 만일 ACK 신호가 제시간에 도착하지 않으면 Timeout이 발생하여, 패킷 손실이 발생한 패킷에 대해 다시 전송하게 된다.



7.2. UDP

비연결지향형이며, 오류를 처리하거나 순서를 재조합시켜주는 기능을 가지고 있지 않다. 단순히 데이터를 받거나, 던져주기만 하는 프로토콜이다. UDP는 특히 실시간 멀티미디어 정보를 처리하기 위해서 주로 사용한다.

- 동작방식

> - TCP와 달리 연결 설정이 없으며, 혼잡제어를 하지 않기 때문에 TCP보다 빠르다는 장점이 있다. 하지만 데이터 전송에 대한 보장을 하지 않기 때문에 패킷 손실이 발생할 수 있다.
> - UDP는 헤더에 있는 Checksum 필드를 통해 최소한의 오류는 검출한다.
> - 최근에는 속도가 빠른 UDP에 신뢰성있는 데이터 전송을 추가하여 서버를 구현하기도 한다.

7.3. TCP vs UDP

TCP	UDP
Connection-oriented protocol (연결지향형 프로토콜)	Connection-less protocol (비 연결지향형 프로토콜)
connection by byte stream (바이트 스트림을 통한 연결)	connection by message stream (메시지 스트림을 통한 연결)
congestion control, flow control (혼잡제어, 흐름제어)	no congestion control, flow control (혼잡제어, 흐름제어 지원하지 않음)
ordered, lower speed (순서 보장, 상대적으로 느림)	not ordered, higher speed (순서 보장되지 않음, 상대적으로 빠름)
reliable data transmission (신뢰성 있는 데이터 전송, 안정적)	unreliable data transmission (데이터 전송을 보장하지 않음)
TCP packet : segment (TCP 패킷은 세그먼트)	UDP packet : datagram (UDP 패킷은 데이터 그램)
HTTP, Email, File transfer (HTTP, 전자우편, 파일전송에서 사용)	DNS, Broadcasting (도메인, 실시간 동영상 서비스에서 사용)

8. HTTP & HTTPS

8.1. HTTP란?

HTTP는 하이퍼텍스트를 빠르게 교환하기 위한 프로토콜의 일종으로, 서버와 클라이언트의 사이에서 어떻게 메시지를 교환할 지를 정해놓은 규칙이다. 1996년에 첫 상용화버전인 HTTP/1.0가 발표되었고, 1999년에 HTTP/1.1, 그리고 2015년 HTTP/2를 공식으로 발표하였다.

8.1.1. HTTP의 구조

HTTP의 구조는 **요청**(Request)과 **응답**(Response)으로 구성되어 있고. 클라이언트가 요청을 하면 서버가 응답을 하는 구조로 되어있다. FTP나 Telnet은 클라이언트가 서버에 정보를 요청해도 서버가 클라이언트와 연결을 끊지 않지만, HTTP는 클라이언트가 서버에 정보를 요청하면 응답 코드와 내용을 전송하고 클라이언트와 연결을 종료한다.

- 요청

> - GET : 클라이언트가 서버에게 URL에 해당하는 자료의 전송을 요청한다.
> - HEAD : GET 요청으로 반환될 데이터 중 헤더 부분에 해당하는 데이터만 요청한다.
> - POST : 클라이언트가 서버에서 처리할 수 있는 자료 보낸다. 예를 들어, 게시판에 글을 쓸 때 클라이언트의 문서가 서버로 전송되어야 한다.
> - PUT : 클라이언트가 서버에게 지정한 URL 에 지정한 데이터를 저장할 것을 요청한다.
> - DELETE : 클라이언트가 서버에게 지정한 URL 의 정보를 제거할 것을 요청한다.
> - TRACE : 클라이언트가 서버에게 송신한 요청의 내용을 반환해 줄 것을 요청한다.
> - CONNECT : 클라이언트가 특정 종류의 프록시 서버에게 연결을 요청한다.
> - OPTIONS : 해당 URL 에서 지원하는 요청 메세지의 목록을 요청한다.

이 중 GET 과 HEAD 요청은 원칙적으로 이를 호출한다고 해서 서버 측의 데이터에 변화가 있어서는 안 된다. 이를 **Safe Method** 라고 분류한다. 또한, GET, HEAD, PUT, DELETE 는 동일한 요청이 한 번 전송되었을 때와 여러 번 연속하여 전송되었을 때의 서버 측의 처리 결과가 동일해야 한다. 이를 **Idempotent Method** 라고 분류한다.

- 응답

1. 1XX : 정보전달. 요청을 받았고, 작업을 진행 중이라는 의미. HTTP/1.0 이후 정의되지 않았다. 서버들도 클라이언트에게 이 코드를 보내지는 않는다. 단 101의 경우 WebSocket등에서 쓰인다.

> - 100 Continue
> - 101 Switching Protocols
> - 102 Processing

2. 2XX : 성공. 이 작업을 성공적으로 받았고, 이해했으며, 받아들여졌다는 의미이다. 200과 206을 제외하고는 볼 일이 거의 없는 코드들이다.

> - 200 OK : 성공적으로 처리했을 때 쓰인다. 가장 일반적으로 볼 수 있는 HTTP 상태.
> - 201 Created : 요청이 성공적으로 처리되어서 리소스가 만들어졌음을 의미한다.
> - 202 Accepted : 요청이 받아들여졌지만 처리되지 않았음을 의미한다.
> - 203 Non-Authoritative Information
> - 204 No Content : 성공적으로 처리했지만 컨텐츠를 제공하지는 않는다.about:blank
> - 205 Reset Content : 서버가 요청을 성공적으로 처리했지만 콘텐츠를 표시하지 않는다. 204 응답과 달리 이 응답은 요청자가 문서 보기를 재설정할 것을 요구한다(예: 새 입력을 위한 양식 비우기).
> - 206 Partial Content : 컨텐츠의 일부 부분만 제공한다. 보통 클라이언트에서 시작 범위나 다운로드할 범위를 지정한 경우 자동으로 해당 부분만 제공할 때 사용하는 코드이다.

- 207 Multi-Status
- 208 Already Reported
- 226 IM Used

3. 3XX : 리다이렉션. 이 요청을 완료하기 위해서는 리다이렉션이 이루어져야 한다는 의미이다. 짧은 주소(단축 URL) 서비스의 경우 접속 시 301이나 302 코드를 보내고, 헤더의 location에 리다이렉션할 실제 URL을 적어 보낸다.

- 300 Multiple Choices
- 301 Moved Permanently : 영구적으로 컨텐츠가 이동했을 때 쓴다.
- 302 Foun d: 다른 페이지로 이동하지만, 나중에 바뀔 수 있음.
- 303 See Other
- 304 Not Modified : 200 다음으로 많이 볼 수 있는 HTTP 상태이다. 이 경우 보통 브라우저에 캐시되어 있는 버전을 쓴다.
- 305 Use Proxy : 요청자는 프록시를 사용하여 요청한 페이지만 액세스할 수 있다. 서버가 이 응답을 표시하면 요청자가 사용할 프록시를 가리키는 것이기도 하다.
- 306 Switch Proxy
- 307 Temporary Redirect : 일시 리다이렉트
- 308 Permanent Redirect : 영구 리다이렉트

4. 4XX : 클라이언트 오류. 이 요청은 올바르지 않다는 의미이다. 여기서부터 브라우저에 직접 표출된다. 굵게 강조된 것은 자주 보이는 오류들이다.

- 400 Bad Request : 요청 자체가 잘못되었을때 사용하는 코드이다.
- 401 Unauthorized : 인증이 필요한 리소스에 인증 없이 접근할 경우 발생한다. 이 응답 코드를 사용할 때에는 반드시 브라우저에게 어느 인증 방식을 사용할 것인지 보내 주어야 한다. 단순히 권한이 없는 경우 이 응답 코드 대신 아래 403 Forbidden을 사용해야 한다.
- 403 Forbidden : 서버가 요청을 거부할 때 발생한다. 관리자가 해당 사용자를 차단했거나 서버에 index.html 이 없는 경우에도 발생할 수 있다. 혹은 권한이 없을 때(로그인 여부와는 무관하다)에도 발생한다.
- 404 Not Found : 찾는 리소스가 없다는 뜻이다.
- 405 Method Not Allowed : PUT이나 DELETE 등 서버에서 허용되지 않은 메소드로 요청시 사용하는 코드이다.
- 406 Not Acceptable : 요청은 정상이나 서버에서 받아들일 수 없는 요청일시 사용하는 코드이다. 보통 웹 방화벽에 걸리는 경우 이 코드가 반환된다.
- 407 Proxy Authentication Required : 프록시 인증이 필요할 경우
- 408 Request Timeout : 요청 중 시간이 초과되었을때 사용하는 코드이다.
- 409 Conflict
- 410 Gone : 404와는 달리 찾는 리소스가 영원히 사라진 경우 사용하는 코드이다.
- 411 Length Required
- 412 Precondition Failed
- 413 Requested Entity Too Large : 요청 본문이 너무 긴 경우 발생한다. 서버 소프트웨어로 엔진엑스를 사용하는 경우 기본 설정 그대로 사용하면 큰 첨부파일을 올릴 때 이 오류 코드가 발생하게 된다.
- 414 Requested URL Too Long: URL이 너무 길 때 발생한다.
- 415 Unsupported Media Type
- 416 Requested Range Not Satisfiable : 요청 헤더의 Range로 지정한 범위가 잘못되었을 때 발생한다.

> - 417 Expectation Failed
> - 429 Too Many Requests: 일정 시간 동안 너무 많은 요청을 보냈을 때 이를 거부하기 위해 사용한다.
> - 451 Unavailable For Legal Reasons : 국가 검열 등의 이유로 차단되었을 경우 사용할 수 있도록 정의된 코드이다.

5. 5XX : 서버 오류. 올바른 요청에 대해 서버가 응답할 수 없다는 의미이다.500 Internal Server Error: 서버에 오류가 발생해 작업을 수행할 수 없을 때 뜬다. 보통 설정이나 퍼미션 문제. 아니면 HTTP 요청을 통해 호출한 문서가 실제 HTML 문서가 아니라 JSP, PHP, 서블릿 등의 프로그램일 경우 그 프로그램이 동작하다 세미콜론 빼먹는 등의 각종 에러로 비정상종료하는경우 이 응답코드를 보낸다.

> - 501 Not Implemented
> - 502 Bad Gateway : 게이트웨이가 잘못 되었을 때 나온다.
> - 503 Service Temporarily Unavailable : 서버를 현재 일시적으로 사용할 수 없을 때 뜬다. 유지보수 중이거나, 터졌거나 할 때 발생한다.
> - 504 Gateway Timeout
> - 505 HTTP Version Not Supported : HTTP 버전을 서버가 처리할 수 없다.
> - 509 Apache bw/limited extension : 대역폭 제한 초과,
> - 520 Unknown Error : 말 그대로 알 수 없는 오류.

8.2. HTTPS

HTTPS(HyperText Transfer Protocol over Secure Socket Layer, HTTP over TLS, HTTP over SSL, HTTP Secure)는 월드 와이드 웹 통신 프로토콜인 HTTP의 보안이 강화된 버전이다.

8.2.1. HTTPS가 중요한 이유

웹사이트의 무결성 보호 HTTPS는 침입자가 웹사이트와 사용자 브라우저 간 통신을 변조하는 것을 방지하는 데 도움을 준다. 침입자란 의도적인 악성 공격자와 합법적인 침입 회사(광고를 페이지에 삽입하는 ISP 또는 호텔)를 포함한다.

- 사용자의 개인정보 및 보안 보호 HTTPS는 침입자가 웹사이트와 사용자 간 통신을 몰래 수신하지 못하도록 방지한다.

9. DNS(Domain Name System)

IP 네트워크에서 사용하는 시스템(DNS는 도메인 이름과 IP 주소를 서로 변환하는 역할을 한다)이다. 우리가 인터넷을 편리하게 쓰게 해주는 것으로, 영문/한글 주소를 IP 네트워크에서 찾아갈 수 있는 IP로 변환해 준다. 이 DNS를 운영하는 서버를 네임서버(Name Server)라고 한다. 규모가 있는 사이트의 경우에는 네임서버를 자체 운영하는 경우가 많다.

9.1. IP adress & host

- IP adress : 각 장치를 나타내는 IP 주소를 가리키는 말

- host : 네트워크에 연결 된 장치들.

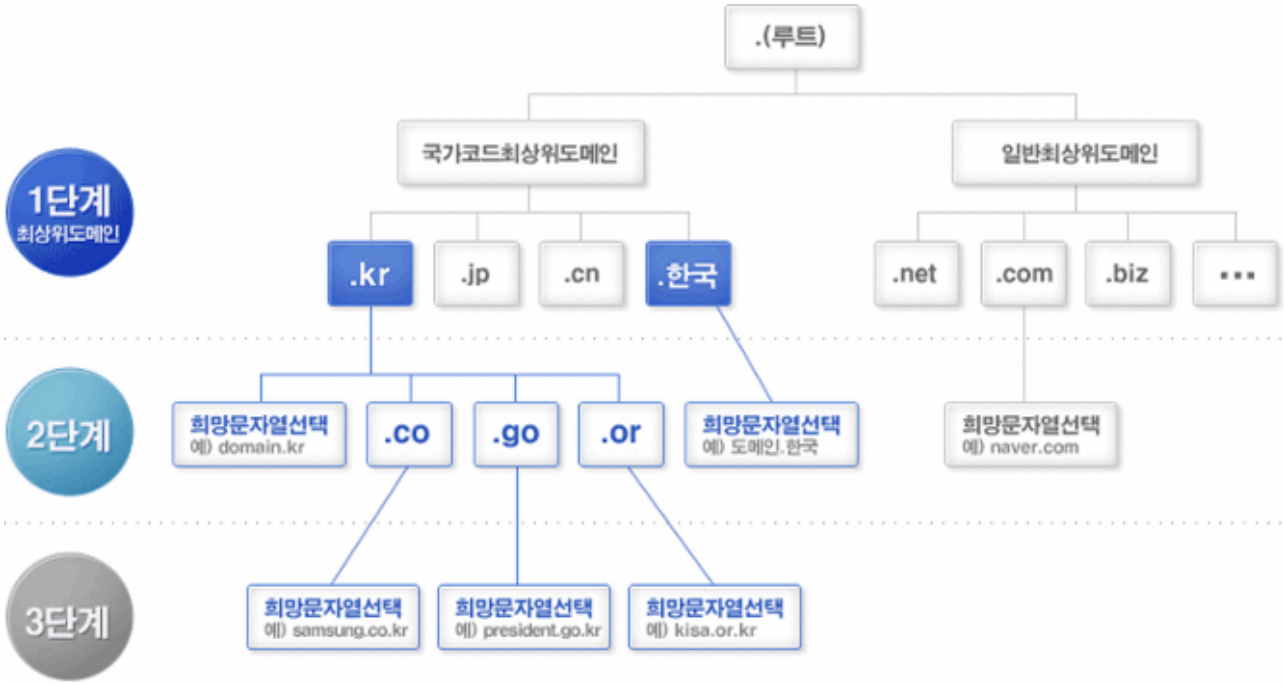
host 파일에 example.com에 접속했을때 93.184.216.34로 연결되도록 설정한다.

9.2. Domain Name Service

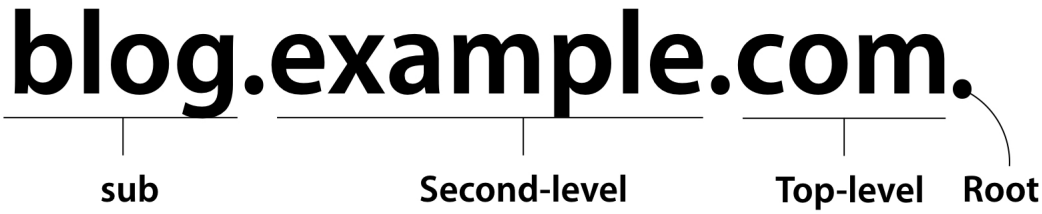
도메인 네임 서비스(Domain Name Service) 는 호스트의 도메인 이름을 호스트의 네트워크 주소로 바꾸거나 혹은 그 반대로 변환을 수행할 수 있도록 하기 위해 개발되었다. 예를 들어 TCP/IP 주소 체계에서 수많은 IP 주소를 사용자는 모두 기억할 수 없다. 조금 더 쉽게 기억하기 도메인을 설정하고 IP 주소 대신에 도메인을 기억함으로써 조금 더 쉽게 해당 시스템을 찾아갈 수 있는 서비스라 생각하면 된다. 즉 도메인 이름과 IP 주소를 매핑 시켜주는 거대한 분산시스템이라 보면 된다.

- DNS의 구성 최상의 루트 도메인이 존재하고 그 아래 com, net, org 등 top 도메인이 존재한다. top 도메인은 국가명을 나타내는 국가 최상위 도메인과 일반적으로 사용되는 일반 최상위 도메인으로 구분된다. 그다음 second 도메인이 구성되고 그다음 서브 도메인이 구성되는 트리 구조 형태로 되어 있다. 보통 .com 이면 기업체, .edu 이면 교육기관, .gov 는 정부기관, .kr 은 국가 도메인으로 체계적으로 분류되어 관

리되고 있다.

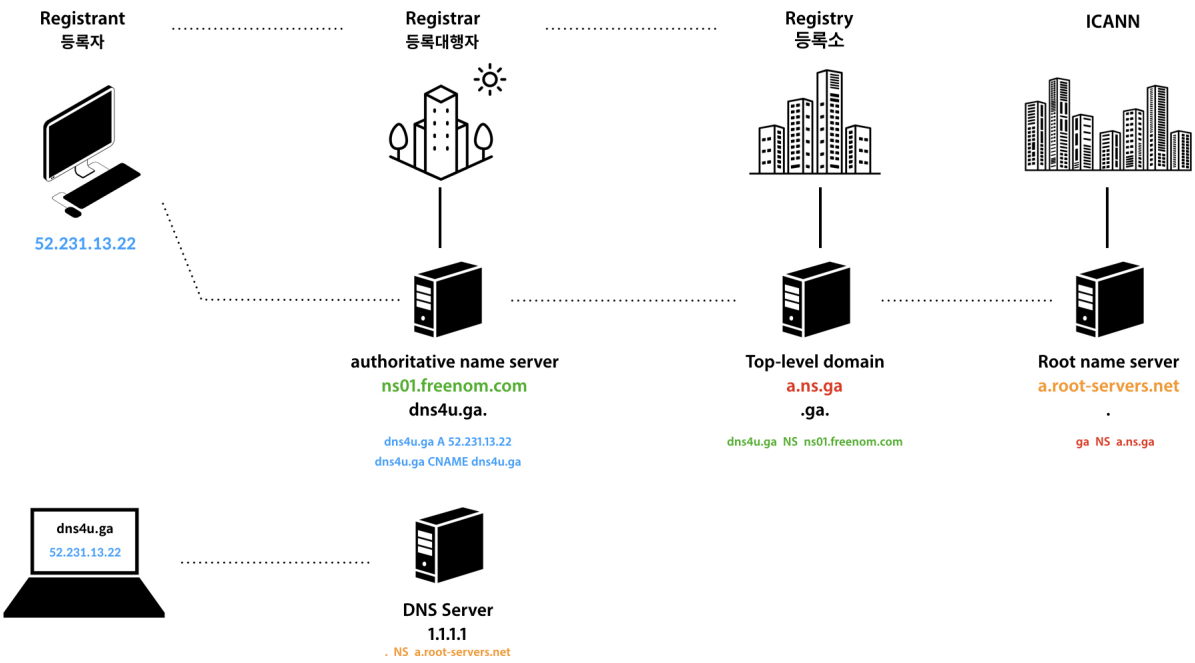


- DNS의 이름 구조 DNS Server는 IP 주소와 Domain 이름을 기억하는 기능과 Client가 이름을 물어보면 IP를 알려주는 기능을 갖고 있다. 수천대의 서버가 같이 협력하고 있다.



각각의 부분들은 부분들을 담당하는 독자적인 Server Computer가 존재한다. Root는 Top-level을 담당하는 Server의 목록과 IP를 알고 있으며, Top-level은 Second-level, Second-level은 sub의 목록과 IP를 알고 있다(상위 목록이 직속 하위 목록을 알고 있음) 최초 root 네임서버의 IP 주소에게 blog.example.com을 물어보면 .com을 담당하는 Top-level을 알려주고, Top-level은 example.com을 담당하는 Second-level을 알려주고, Second-level은 blog.example.com을 담당하는 sub DNS Server에게 물어보고, sub가 해당 IP 주소를 알려준다. 계층적인 구조를 가지고 있다

- DNS 이름 등록 과정

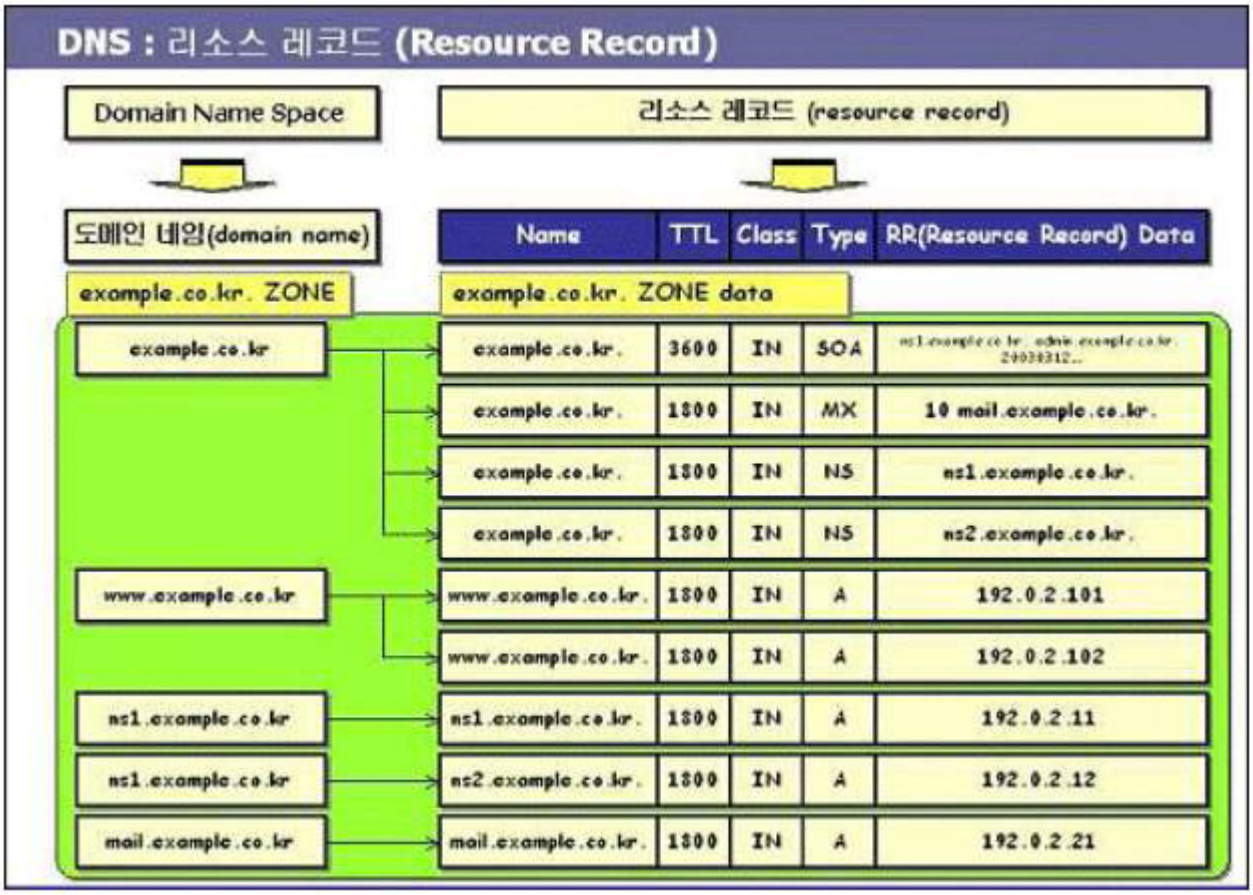


ICANN(전세계에 있는 ip주소를 관리하는 비영리단체) - Registry(등록소) - Registrar(등록대행자) - Registrant(등록자) 관계

9.3. DNS 레코드

도메인 네임 공간(Domain Name Space)에서 설정된 도메인 네임은 그 자체로서는 인터넷 자원과 관련된 아무런 정보를 가지고 있지 않다. 즉, 하나의 도메인 네임은 전체 네임체계 속에서 유일성을 갖는 네임으로서의 특성만을 가지고 있다. 도메인 네임은 인터넷 상에서 사용하는 각종 자원(resource) 정보와 연관시킬 때, 인터넷 네임체계로서의 의미를 지닐 수 있다. 리소스 레코드(RR : Resource Record)는 도메인 네임(domain name)과 인터

넷 자원(resource) 정보를 매핑하여 하나의 분산 데이터베이스를 구성하기 위한 수단이다.



Type	Code	의미	설명
A	1	A host address	32비트의 IPv4 주소를 의미한다.
AAAA	28	IPv6 address	128비트의 IPv6 주소를 의미한다
NS	2	An authoritative name server	DNS 구역을 위한 Authoritative DNS 서버의 네임을 나타낸다.
CNAME	5	The canonical name for an alias	노드의 실제 네임을 가리키도록 정의한 별칭을 위해 사용한다.
SOA	6	Mark the start of a zone of authority	DNS 구역의 시작을 표시하는 데 사용하며 DNS 구역에 대한 중요한 정보이다. 모든 구역은 정확히 하나의 SOA 레코드를 가져야 한다.
PTR	12	A domain name pointer	IP 주소를 기반으로 도메인 이름을 찾는 데 사용된다. A 레코드와 반대 개념이다.
MX	15	Mail exchange	도메인으로 오는 이메일을 처리하는 위치를 명시한다.
TXT	16	Text strings	도메인과 관련해 저장해야 할 임의의 문자를 나타낸다.
HINFO	13	Host information	호스트에 대한 일반 정보를 얻는 데 사용된다. CPU 및 OS 유형을 알려준다.

10. Home server

가정에서 간단하게 사용하기 위한 서버. 데이터 공유가 주목적이 되는 것은 NAS로 따로 분류한다.

10.1.Router

라우터(router) 혹은 라우팅 기능을 갖는 공유기는 패킷의 위치를 추출하여, 그 위치에 대한 최적의 경로를 지정하며, 이 경로를 따라 데이터 패킷을 다음 장치로 전향시키는 장치이다. 대표적으로 보통 가정에서 쓰는 공유가기 있다. ![라우터]

- Public IP Adress : 공유기가 갖게 되는 WAN IP.
- Private IP Address : 공유기를 통과한 LAN IP

  라우터는 LAN IP와 WAN IP 2가지 모두를 갖고 있다.

10.2.NAT

NAT(Network Address Translation)은 사설IP를 쓰고 있는 각각의 컴퓨터들이 외부의 인터넷에 들어갈 수 있게 해주는 기술이다.

예를 들어, A는 IP 192.168.0.4 라는 IP를 갖고 공유기에 외부 인터넷에 접속할 신호를 보낸다

> 1. 공유기는 이 A의 IP를 기억
> 2. NAT라는 기술을 통해 IP를 변환하여 외부의 인터넷 망에 요청
> 3. 외부의 인터넷 망이 요청을 받고 요청받은 IP로 답장
> 4. 공유기는 해당 IP를 기억한 A의 IP에 응답

이 과정을 통해 Private IP를 가지는 내부망이 외부의 인터넷 망에 접속을 할 수 있게 해 준다.

10.2.1. NAT의 종류

- 정적 NAT : 하나의 내부 IP주소와 외부 IP주소를 1:1로 매핑한다.

> - 동작방식
>
> 1. 하나의 내부 IP주소와 하나의 외부 IP주소를 1:1로 미리 지정
> 2. 외부주소로 들어온 요청을 미리지정된 내부주소로 변환하여 내부로 전달

- 동적 NAT : 여러개의 내부 IP주소와 여러개의 외부 IP주소를 동적으로 매핑한다.

> - 동작방식
>
> 1. 호스트가 외부로 보내는 트래픽을 라우터로 보냄
> 2. 라우터에 설정된 공인IP주소 풀에서 쉬고있는 IP중 하나로 변환하여 외부로 내보냄 (1:1로 사용하며, 사용중인 외부IP는 nat table에 기록해 놓고 중복으로 사용하지 않는다)
> 3. 외부에서 응답신호가 라우터로 돌아오면 라우터는 nat table 에 있는 정보를 확인하고 사설 IP로 변환
> 4. 내부망으로 전달

- 오버로딩 (Overloading) : 서로 다른 포트를 사용하여 등록되지 않은 여러 IP 주소를 등록 된 단일 IP 주소로 매핑하는 동적 NAT의 한 형태. PAT (포트 주소 변환), 단일 주소 NAT 또는 포트 수준 다중화 NAT라고도 한다.

10.3.Port & Port forwarding

10.3.1. Port

포트는 네트워크의 가장 마지막 종착점을 알려주는 것으로 각 PC마다 포트는 0번부터 65535번 까지 있다.

> - 22번은 SSH에 고정적으로 사용
> - 23번은 텔넷
> - 53번은 DNS
> - 80번은 Http에 고정적으로 사용
> - 0번부터 1023번 포트는 Well Known Port(예약된 포트)라고 해서 사용 용도가 정해져서 마음대로 쓰면 안되는 포트

10.3.2. Port forwarding

공유기 외부에서 공유기 내부의 컴퓨터에 접속하기 위해서는 공유기의 몇번 포트에 접속한 정보를 공유기 내의 어떤 아이피의 몇번 포트로 연결해줄 것인지를 공유기에게 알려줘야 한다. 이를 포트포워딩이라고 한다.

10.4.Dynamic IP & Static IP

- 유동 IP는 일반적으로 DHCP라는 서버로 부터 ip를 할당받아 사용하고 빌린 기간이 끝나면 IP adress를 반납후 다시 받아온다. 이런 방식은 컴퓨터를 껐다 켰을때 arp라는 신호를 보내어 DHCP server로부터 IP address를 받게 된다. ip가 모자라는 현실에서 상당히 중요한 방식 중 하나이다.
- 고정 IP는 하나의 IP address를 하나의 컴퓨터가 고정적으로 가지고 있음으로써 자신의 IP가 변하지 않는 것을 말한다. 자신의 컴퓨터의 주소를 항상 숙지했다가 인터넷이 연결된 어느곳에서든 자신의 컴퓨터에 접속해 작업할 수 있고 web서비스나 ftp서비스도 할 수 있다. 단점은 특정 Ip를 고정으로 사용하는 만큼 해킹의 목표가 되기 쉬울 것이다.

10.5. DHCP

DHCP(Dynamic Host Configuration Protocol)은 네트워크에 접속한 장치의 ip, subnet mask, gateway address, DNS와 같은 정보를 자동으로 설정해주는 기술이다.