

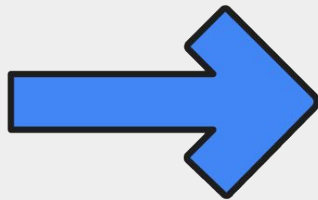


Google Developer Group
Incheon National University

TCP/IP Protocol

GDGoC INU Backend Part Seminar

Member 전유진



목차

TCP/IP 개요

- TCP/IP 개념 및 역할
- OSI 7계층과 TCP/IP 4계층 비교

IP와 TCP

IP

- IP 주소 체계 (IPv4 vs IPv6)
- 공인 IP vs 사설 IP, NAT
- IP 관련 기술, DNS

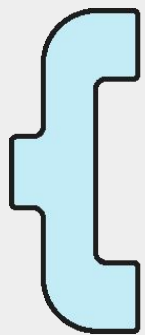
TCP

- TCP와 UDP의 차이
- TCP 소켓 프로그래밍 코드
: TCP 에코 서버 프로그램

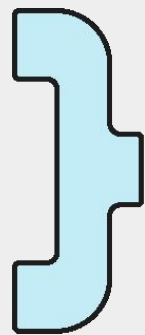
TCP/IP 기반 서버 통신

- Web Server와 TCP/IP
통신 과정
: HTTP의 연결 요청

참고 문헌



TCP/IP 개요



TCP/IP 개념 및 역할

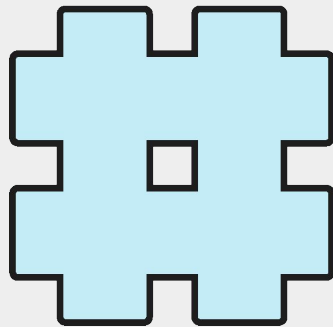
IP로 통신할 때에 필요한 많은 프로토콜군을 의미한다.

‘Internet Protocol Suite’라고도 부르며 ‘인터넷 구축에 필요한 프로토콜의 집합’이라는 뜻이다.

TCP = 포트 번호에 의해 상위층인 애플리케이션의 처리를 수행할 프로그램 식별

IP = 추상화를 통해 서로 다른 데이터 링크 사이의 통신이 가능하게 해주는 프로토콜

TCP/IP = 두 호스트 간에 데이터 흐름의 정확성을 보장하는 프로그램, 서로 다른 프로토콜 간의 통신을 가능하게 한다.



OSI 7계층과 TCP/IP 4계층 비교

애플리케이션층 (세션층 이상의 상위 계층)

- 클라이언트/서버 모델로 이루어져 있다.

트랜스포트층

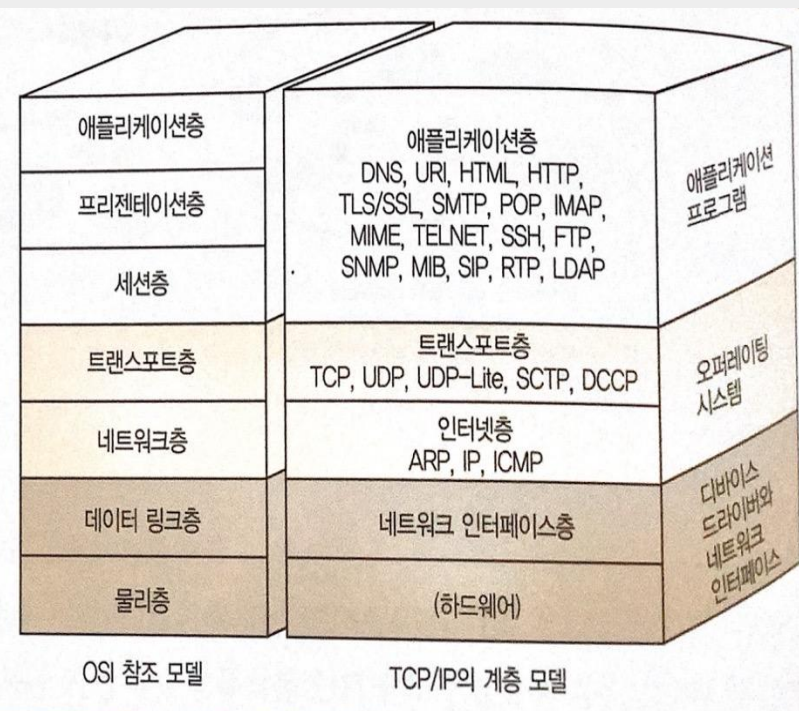
- 애플리케이션 프로그램 간의 통신을 구현한다.

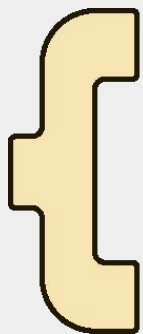
인터넷층 (네트워크층)

- 주로 IP 프로토콜을 사용한다.

네트워크 인터페이스층 (데이터 링크층)

- 네트워크 장비 간의 데이터 링크 통신을 위한 인터페이스이다.
- OS와 하드웨어를 이어준다.





IP와 TCP



IP

IP 주소 체계 IPv4 vs IPv6

항목	IPv4	IPv6
주소 길이	32bit	128bit
주소 공간	2^{32} = 약 43억개	2^{128} = 약 340억개
주소 표현	192.168.0.1(10진수)	2001:0db8:85a3::8a2e:0370:7334(16진수)
특징	NAT 사용	NAT 사용X

IP

공인 IP vs 사설 IP, NAT

Public IP

전체 인터넷에서 고유한
주소 할당

ISP가 제공

인터넷에 공개된 서버

8.8.8.8 (Google DNS),
52.78.32.1 (AWS EC2)



Private IP

동일 조직 내에서
고유한 주소 할당

라우터가 개별 호스트에
제공

인터넷에 직접 연결
불가

192.168.0.1, 10.0.0.1

IP

IP 관련 기술, DNS

사용자가 도메인명을 입력하면 자동으로 호스트명과 IP주소가 등록되어 있는 데이터베이스 서버를 검색하여 IP 주소의 정보를 얻는다.

DNS 레코드는 DNS 서버 내에서 도메인 이름을 인터넷 프로토콜(IP) 주소와 연결하는데 사용되는 명령어 집합이다.

호스트명과 IP 주소의 대응 = A 레코드

IP 주소로부터 호스트명을 검색 = PTR 레코드

상위나 하위 네임 서버의 IP 주소 대응 = NS 레코드

호스트의 별명에 대한 정식명 = CNAME

TCP

TCP와 UDP 차이

TCP(Transmission Control Protocol)

연결 지향

스트림형 프로토콜

신뢰성 높다(순서 제어, 재전송 제어)

1:1 통신

속도 느림

UDP(User Datagram Protocol)

비연결 지향

데이터그램형 프로토콜

신뢰성이 낮다

1:1, 1:n, n:n 통신

속도 빠름

TCP

TCP 기반 소켓 프로그래밍

```
if ((serv_sock = socket (PF_INET, SOCK_STREAM,  
                        IPPROTO_TCP)) < 0)  
    exit_error ("socket() failed");
```

1. socket() : 소켓 생성, 소켓은 클라이언트의 접속 요청을 접수

TCP

TCP 기반 소켓 프로그래밍

```
if ((clnt_sock = accept(serv_sock, (struct sockaddr *)  
                        &clnt_addr, &caddr_len)) < 0)  
    exit_error("accept() failed");
```

2. accept() : 데이터 소켓 생성, 클라이언트와 메시지를 교환

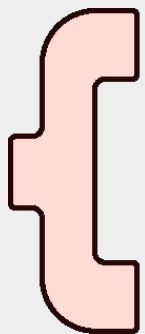
TCP

TCP 기반 소켓 프로그래밍

```
if ((msg_size = recv(clnt_sock, msg_buf, MSGBUFSIZE, 0)) < 0)
    exit_error("recv() failed");
```

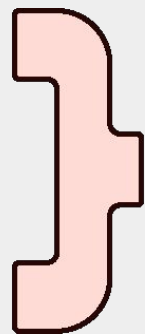
```
if (send(clnt_sock, msg_buf, msg_size, 0) != msg_size)
    exit_error("send() failed");
```

- 3. `recv()` : 메시지 수신, 소켓에 도착한 데이터를 버퍼에 저장하는 함수
- `send()` : 메시지 전달, 서버의 버퍼에 있는 데이터를 소켓이 전달하여 `client`로 전송

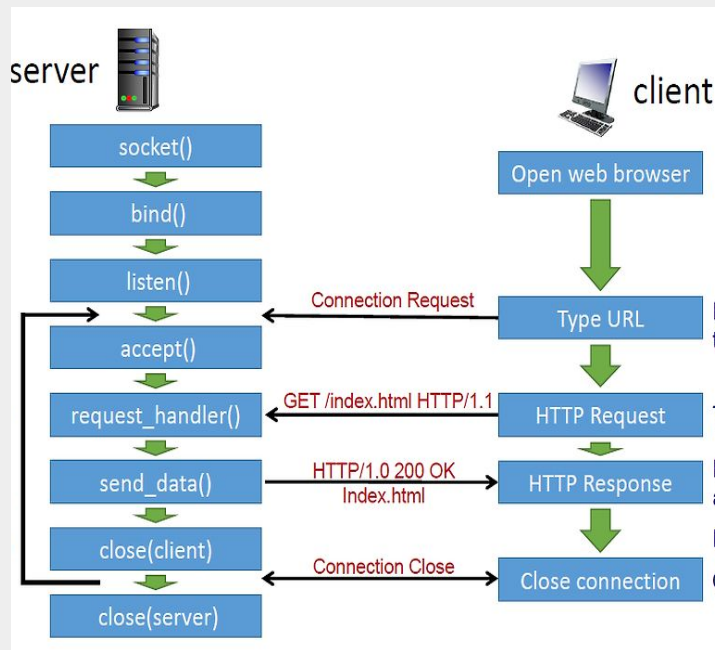
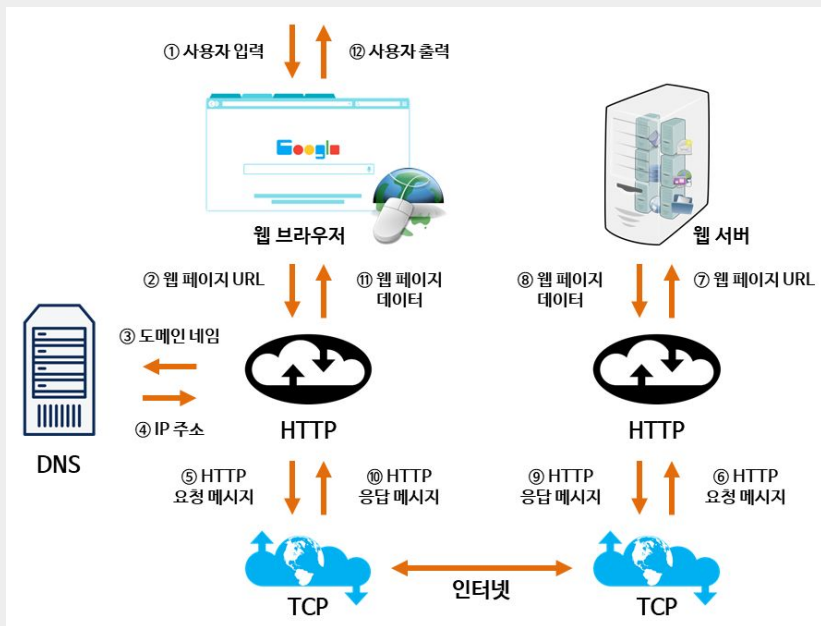


TCP/IP

기반 서버 통신



Web Server와 TCP/IP 통신 과정



참고 문헌

- 다케시타 다카후미, 마스터링 TCP/IP 입문편, 성안당, 2021.
- 주기호, 소켓 API를 이용한 네트워크 프로그래밍, 배재대학교출판부.
- 김선우, TCP/IP 소켓 프로그래밍 2판, 한빛아카데미, 2022
- 시스코시스템즈 G.K., 네트워크 엔지니어의 교과서, 로드북, 2016