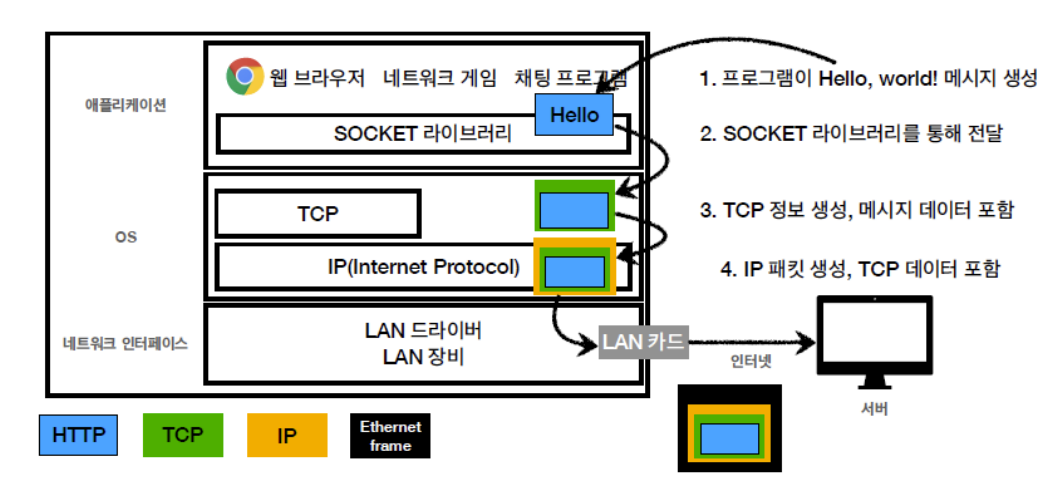
인터넷 프로토콜 슈트 중 TCP와 IP가 가장 많이 쓰이기 때문에 TCP/IP 프로토콜 슈트라고도 불린다.

인터넷 프로토콜 4계층

테이블이(가) 표시된 사진

자동 생성된 설명

TCP/IP

TCP/IP는 패킷 통신 방식의 인터넷 프로토콜인 IP (인터넷 프로토콜)와 전송조절 프로토콜인 TCP (전송 제어 프로토콜)로 이루어져 있다. IP는 패킷 전달 여부를 보증하지 않고,

패킷을 보낸 순서와 받는 순서가 다를 수 있다.(unreliable datagram services) TCP 는 IP 위에서 동작하는 프로토콜로, 데이터의 전달을 보증하고

보낸 순서대로 받게 해준다. HTTP, FTP, SMTP등 TCP를 기반으로 한 많은 수의 애플리케이션 프로토콜들이 IP 위에서 동작하기 때문에, 묶어서 TCP/IP라고 부른다.

TCP / UDP

User Datagram Protocol의 축약어로 컴퓨터가 다른 컴퓨터와 데이터통긴을 하기 위한 규약(프로토콜)의 일종이다. UDP는 세계 통신표준으로 개발된 OSI 모형에서

4번째 계층인 전송 계층(Transport Layer)에서 사용하는 규약이다. 동일 계층에서 사용하는 또 다른 프르토콜로 TCP가 존재한다.

연결방식

TCP : 연결기반/ 연결 후 통신(전화기) / 1 : 1 통신방식

UDP : 비연결기반 / 연결없이 통신(소포) / 1 : 1 , 1 : n , n : n 통신방식

특징

TCP : 데이터의 경계를 구분 안함, 신뢰성 있는 데이터 전송

* 데이터의 전송 여부가 보장됨
* 데이터의 수신여부를 확인함(데이터가 손실되면 재전송됨)
* 패킷을 관리할 필요가 없다.
* UDP보다 전송속도가 느리다.

UDP : 데이터의 경계를 구분함(datagram), 신뢰성 없는 데이터 전송

* 데이터의 전송순서가 바뀔 수 있음
* 데이터의 수신여부를 확인 안함(데이터가 손실되어도 알 수 없음)
* 패킷을 관리해주어야 함
* TCP 보다 전송속도가 빠름

관련 클래스

TCP : Socket, ServerSocket

UDP : DatagramSocket, DatagramPacket, MulticastSocket

활용

TCP : 비교적 안정적인 통신이 필요한 파일 교환

UDP : 게임이나 동영상의 데이터를 전송하는 경우 중간에 손실되어 좀 끊기더라도 속도만 빠르면 되므로 TCP보다 빠른UDP가 적합하다.

내가 생각하는 UDP

손상된 데이터가 있는지 확인하면서 보내는 TCP와는 다르게 다소 데이터 손실이 있어도 상관없이 빠르게 도착만 하면 된다는 생각으로 나온 방식 주로 게임, 동영상 스트리밍 같은 곳에 주로 쓰인다.

**TCP의 특징**

1. 연결 지향 – TCP 3 way handshake(가상 연결)
2. 데이터 전달 보증
3. 순서 보장
4. 신뢰할 수 있는 프로토콜
5. 현재는 대부분 TCP를 사용한다.

**UDP의 특징**

1. 하얀 도화지에 비유(기능이 거의 없다)
2. 연결지향 – TCP 3 way handshake X
3. 데이터 전달 보증 X
4. 순서 보장 X
5. 데이터 전달 및 순서가 보장되지 않지만, 단순하고 빠르다
6. 정리

* IP와 거의 같다 => (PORT + 체크섬(CheckSum)정도만 추가된 형태)
* 애플리케이션에서 추가 작업이 필요하다
* 튜닝 및 최적화로 요즘 뜨는 통신방식

**3. Statelful / Stateless**

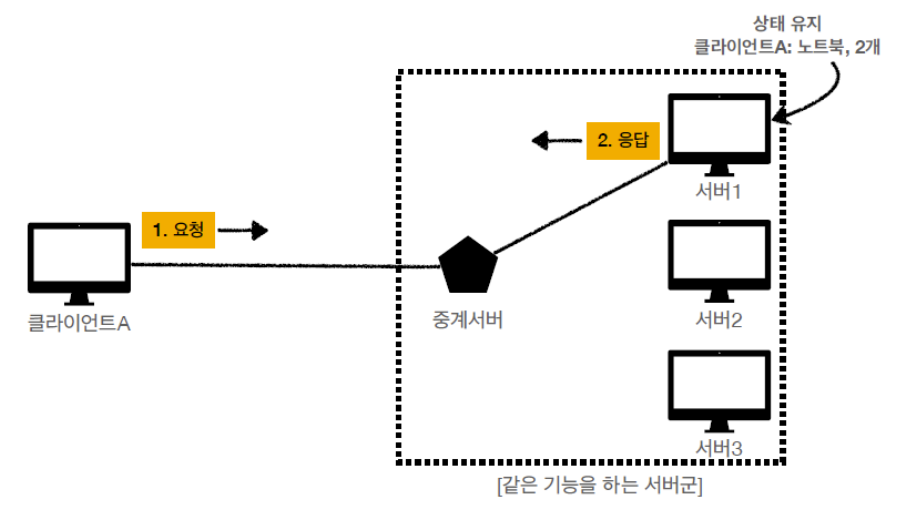
- 프로토콜에서 상태 처리에 대한 두 개념

- 클라이언트와 서버간의 네트워크 통신이 어떻게 이루어지는지에 대한 개념

(네트워크 프로토콜)

**상태 유지(Stateful)** : 중간에 다른 점원으로 바뀌면 안된다.

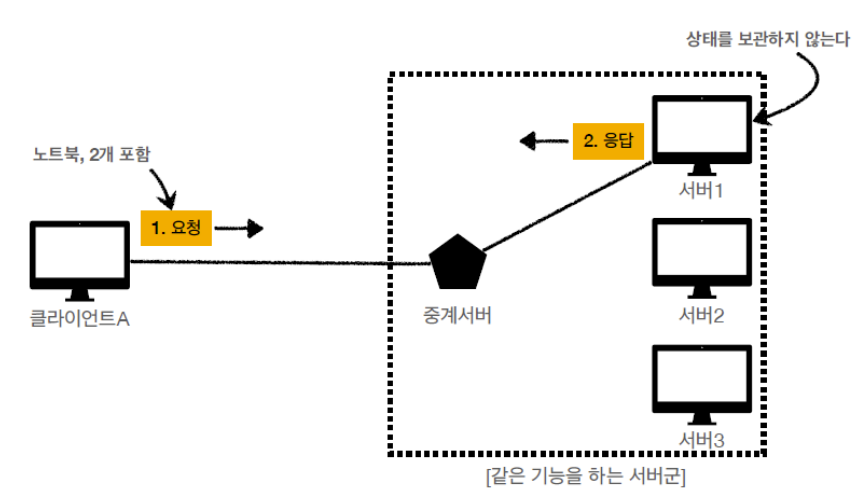
* 중간에 다른 점원으로 바뀔 때 상태 정보를 다른 점원에게 미리 알려줘야 한다.



서버 장애가 발생하면 상태 값이 소멸해서 처음부터 다시 요청해야 한다.

무상태(Stateless) : 중간에 다른 점원으로 바뀌어도 된다.

* 갑자기 고객이 증가해도 점원을 대거 투입할 수 있다.
* 갑자기 클라이언트 요청이 증가해도 서버를 대거 투입할 수 있다.
* 무상태는 응답 서버를 쉽게 바꿀 수 있다. -> 무한한 서버 증설 가능(L4 스위치, 쿠버네티스)



**서버 장애가 발생해도 클라이언트가 상태 값을 가지고 있기 때문에 다른 서버에서 요청하면 된다.**

무상태(Stateless) 프로토콜

1. 서버가 클라이언트의 상태를 보존 X
2. 장점 : 서버 확장성 높음(스케일 아웃)
3. 단점 : 클라이언트가 추가 데이터 전송

세션 상태

* 클라이언트가 서버간 통신 인증이 된 상태를 의미
* 인증된 상태에서 데이터 송수신이 가능

세션 정보

* 한 세션 내에서, 클라이언트가 서버에 전송할 데이터 정보를 의미합니다.
* 서버는 세션 유지 시간이 지나거나, 클라이언트가 전송하려했던 데이터를 모두 수신할 때까지 클라이언트와의 세션 상태를 유지합니다.

Proxy Server : 클라이언트가 자신을 통해서 다른 네트워크 서비스에 간접적으로 접속할 수 있게 해주는 컴퓨터 시스템이나 응용 프로그램, Server 와 Client 사이에 중계기로서 대리로 통신을 수행하는 것을 가르켜 ‘Proxy’, 그 중계 기능을 하는 것을 Proxy Server라고 부른다.

프록시 서버의 종류

1. 포워드 프록시(Forward Proxy)

Client -> Forward Proxy Server -> Internet -> Server

1. 리버스 프록시(Reverse Proxy)

Client -> internet -> Reverse Proxy Server -> Server

한줄 요약

1. TCP/IP **컴퓨터 사이의 통신 표준 및 네트워크의 라우팅 및 상호연결에 대한 자세한 규칙을 지정하는 프로토콜 스위트**

2. TCP/UDP 비연결형 서비스를 지원하는 전송계층 프로토콜로써, 인터넷상에서 서로 정보를 주고받을 때 정보를 보낸다는 신호나 받는다는 신호 절차를 거치지 않고,보내는 쪽에서 일방적으로 데이터를 전달하는 통신 프로토콜입니다.

3. Stateful/Stateless : 클라이언트와 서버간의 네트워크 통신이 어떻게 이루어지는지에 대한 개념 (네트워크 프로토콜)

Stateful: **세션이 종료될 때까지, 클라이언트의 세션 정보를 저장하는 네트워크 프로토콜.**

**Stateless : 서버가 클라이언트의 세션 상태 및 세션 정보를 저장하지 않는 네트워크 프로토콜**

4. proxy : 클라이언트가 자신을 통해서 다른 네트워크 서비스에 간접적으로 접속할 수 있게 해주는 컴퓨터 시스템이나 응용 프로그램