예상문제 : 네트워크 편

1. osi 7계층에 대해 설명해주세요

1계층 : 물리계층

데이터를 전기 신호로 바꿔서 와이어에 실어주는 계층

리피터,허브

2계층:데이터링크 계층

p2p간 신뢰성있는 전송을 보장하기 위한 계층으로 crc 기반의 오류 제어와 흐름 제어가 필요

3계층 : 네트워크 계층

ip주소 기반으로 경로를 찾아주는 계층

라우터를 통해 이동할 경로를 선택하여 ip주소를 지정하고, 해당 경로에 따라 패킷을 전달

장비: router

단위: 패킷

프로토콜: ip, rip, arp, icmp

4계층: 전송계층

tcp,udp등의 프로토ㅗㄹ을 통해 통신을 활성화하고

포트를 열어두고 프로그램들이 전송할 수 있도록 제공

오류검출 복구, 흐름 제어 등 수행

포트기반 데이터 세그먼트 전송

장비:게이트웨이프로토콜:tcp/ip

5계층:세션 계층

TCP/IP세션을 만들고 없애는 책임을 지닌다. 세션 설정, 유지,종료 등의 기능이 있다

SOCKET,API

6 표현계층

데이터 표현에 대한 독립성을 제공하고 암호화하는 역할

JPEG,MPEG

7응용계층

일반적인 응용 서비스를 수행한다. 사용자 인터페이스,전자우편, 데이터베이스 등의 서비스를 제공한다.

HTTP,FTP,SMTP,IMAP등

2. TCP와 UDP의 특징과 장단점

TCP와 UDP는 전송계층에서 사용되는 전송 프로토콜이다.

TCP는 인터넷상에서 데이터를 메세지의 형태로 보내기 위해 IP와 함께 사ㅇ사용하는 프로토콜

신뢰적, 연결지향성 서비스 제공

신뢰적인 전송을 보장하기위해 HANDSHACKING하고 데이터의 흐름제어와 혼잡제얼르 수해 앟기 때문에 느리다.

신뢰성있는 데이터 통신,양방향 흐름제어,혼잡제어 역할

UDP는 데이터를 데이터그램 단위로 처리하는 비연결형 프로토콜

데이터를 서로 다른 경로로 독립적으로 처리하는 프로토콜이다. 패킷에 순서를 부여하여 재조립하거나 흐름제어 및 혼잡제어를 수행하지 않아 속도가 빠르며 네트워크 부하가 적지만 데이터 전송의 신뢰성이 낮다. 실시간 서비스에 좋음

3. HTTP와 HTTPS의 차이점

HTTP는 인터넷 상에서 클라이언트와 서버가 자원을 주고 받을 때 ㅆ느ㅡㄴ 통신규약이다. 텍스트 교환이므로, 누군가 네트워크에서 신호를 가로채면 내용이 노출되는 보안이슈가 존재한다. 이 보안 문제를 해결하기 위해 HTTPS

HTTP는 인터넷 상에서 정보를 암호화하는 SSL프로토콜을 사용해 클라이언트와 서버가 자원을 주고 받을 때 쓰느 통신 규약

HTTP는 암호화가 추가되지 않기 때문에 보안에 취약하지만 HTTPS는 안전하게 데이터를 주고받을 수 있다.

HTTPS는 암호화/복호화 과정이 필요하기 때문에 HTTP보다 속도고 느리다. 인증서를 발급하고 유지하기 위한 추가 비용도 발생한다.

4. 쿠키와 세션의 차이점

클라이언트가 매번 누구인지 확인해야하는 HTTP프로토콜의 특과 약점을 보완하기위해 쿠키와 세션을 사용한다.

쿠키는 그 사이트가 사용하고 있는 서버에서 사용자의 컴퓨터에 저장하는 작은 기록 정보 파일

아이디 비번 자동 입력

세션: 화면이 이동해도 로그인이 풀리지 않고 로그아웃하기 전까지 유지

일정시간(웹브라우저를 통해 웹서버에 접속한 시점으로부터 웹브라우저를 종료하여 연결을 끝내는 시점)동안 같은 사용자로부터 들어오는 일련의 요구를 하나의 상태로 보고 그 상태를 일정하게 유지시키는 기술

쿠키는 정보파일을 브라우저 로컬에 저장하고 유효시간을 정할 수 있다. 세션은 서버측에서 관리하기 때문에 서버에서는 클라이언트를 구분하기 위해 세션 ID를 부여하여 웹브라우저가 서버에 접속해서 브라우저를 종료할 떄까지 인증상태를 유지한다. 서버에 정보를 두기 떄문에 보안에 좋지만 사용자가 많아질수록 메모리를 많이 차지한다.

쿠키는 서버 요청시 빠르다.

5.GET과 POST메소드 설명

GET은 정볼르 조회하기 위한 메소드

POST는 리소스를 생성/변경하기 위해 설계되었다. GET과 달리 전송해야할 데이터를 HTTP 메시지의 BODY에 담아 전송한다. BODY는 길이의 제한없이 데이터를 전송할 수 있다. 따라서 GET과 달리 대용량 데이터를 전송할 수 있다.

6.로드밸런싱이란?

하나의 인터넷 서비스가 발생하는 트래픽이 많을 때 여러 대의 서버가서버의 로드율 등가, 부하량, 속도저하 등을 고려하여 적절히 분산 처리하여 해결해주는 서비스

7.CORS

한 도메인 또는 ORIGIN의 웹페이지가 다른 도메인을 가진 리소스에 액세스할 수있게 하는 보안 메커니즘

종류로 SIMPLE REQUEST, PREFLIGHT REQUEST, CREDENTIAL REQUST, NON-CREDENTIAL REQUEST가 있다.

CORS요청시 미리 OPTIONS주소로 서버가 CORS를 허용하는지 물어본다

이때 ACCESS-CONTROL-REQUEST-METHOD로 실제로 보내고자 하는 메서드를 알리고 access-control-request-headers로 실제로 보내고자 하는 헤더들을 알린다.

allow를 request에 대응되는 것으로 서버가 허용하는 메서드와 헤더를 응답하는데 사용된다. request와 aloow가 일치ㅏㅎ면 cors요청이 이루어진다.

8. PORT: PORT정보를 통해 어떤 프로세스가 메세지를 받아야 하는지 알 수 있다.

9. 127.0.0.1: 패킷을 외부로 전송하지 않고 그대로 자신이 수신한다.

10. 흐름제어와 혼잡제어

흐름제어: 각 상대측 노드의 데이터 처리 속도 차이를 해결하는 방법

혼잡제어:네트워크 상황에 맞게 데이터의 양을 제어하는 방법

11. HTTP1.1 과 HTTP 2.0의 다른점

HTTP1.1.은 기본적으로 연결 당 하나의 요청과 응답을 처리하기 때문에 동시전송 문제와 다수의 리소스를 처리하기에 속도와 성능 이유를 가진다.

RTT증가, HOL BLOCK발생, 헤더가 큼

HTTP2.0이 등장

MULTIPLEXED STREAMS: 한 연결에 여러개의 메시지를 동시에 주고 받을 수 있음

HOL BLOCK이 발생하지 않음

헤더 압축

HTTP1.1.과 높은 수준의 호환성 페이지 로딩 속도 향상

12. IPv4 IPV6

32 비트 128비트

헤더크기 가변 고정

PLUG*PLAY 불가 가능

브로드캐스트 주소 있음 없ㅅ음

13. REST API

REST아키텍처 스타일을 따르는 API로

특징은 확장성과 재사용성을 높여 유지보수가 용이하다.

클라이언트 서버 구조이다.

14. 주소창에 특정 url을입력하면 일어나는 일

브라우저

URL에 입력된 값을 브라우저 내부에서 결정된 규칙에 따라 그 의미를 조사한다. 조사된 의미에 따라 HTTP REQUEST메시지를 만든다. 만들어진 메시지를 웹 서버로 전송한다.

프로토콜 스택,LAN어댑터

프로토콜 스택이 브라우저로부터 메시지를 받는다. 브라우저로부터 받은 메시지를 패킷 속에 저장한다. 수신처 주소등의 제어정보를 덧붙인다. 그런 마등 패킷을 LAN어댑터에 넘긴다.

LAN어댑터는 다음 HOP의 MAC주소를 붙인 프레임을 전기신호로 변환시킨다.

신호를 LAN케이블로 송출시킨다.

허브,스위치,라우터

LAN어댑터가 송신한 프레임은 스위칭 허브를 경유하여 인터넷 접속용 라우터에 도착한다. 라우터는 패킷을 프로바이더(통신사)에 전달한다

인터넷으로 들어가게 된다

액세스 회선,프로바이더

패킷을 인터넷의 입구에 있는 액세스 회선에 의해 POP까지 운반된다

수많은 고속 라우터들 사이로 패킷이 목적지를 향해 흘러가게 된다

POP을 거쳐 인터넷의 핵심부로 들어가게 된다

방화벽,캐시서버

패킷은 인터넷 핵심부를 통과하여 웹 서버측의 LAN에 도착한다

기다리고 있던 방화벽이 도착한 패킷을 검사한다.

패킷이 웹서버까지 가야하는지 가지 않아도 되는지 판단하는 캐시서버가 존재한다.

웹서버

패킷이 물리적인 웹서버에 도착하면 웹 서버의 프로토콜 스택은 패킷을 추출하여 메시지를 복원하고 웹 서버 애플리케이션에 넘긴다.

메시지를 받은 웹 서버 애플리케이션은 요청 메시지에 따른 데이터를 응답 메시지에 넣어 클라이언트로 회송한다.

왔던 방식대로 응답 메시지가 클라이언트에게 전달된다.