네트워크 특강 (7주차 과제관련 - 네트워크 개론, 클라이언트/서버)

네트워크 상호 연결되어 있는 사람들이나 사물들로 이루어진 그룹이나 시스템

점 간의 관계를 선으로 연결한 것

Internetwork

IP

네트워크를 연결하는 프로토콜

계층화의 장점

각 계층은 자기가 제공할 기능만 생각하면 된다.

각 계층은 자기 바로 아래 계층을 어떻게 쓰는지만 알면 된다.

문제 해결의 편의성

문제가 있는 계층만 디버깅하면 된다.

진화의 편의성

계층화의 단점

각 계층을 넘나드는 것이 비효율적일 수 있다.

계층화에서의 황금률

하위 계층에 뭘 넣지 마라.

그게 성능상 절대적으로 필요한 것이 아니라면…

OSI7 계층 모델

영어 이름에서 기준 모델에 주목

즉, 실제 구현으로 존재하는 것x

TCP/IP 역시 osi model 적용

Nbs resolution 이름으로부터 주소를 알아낸다.

DNS 캐싱

일반적인 네트워크에서 통신 하는법

Gateway: 관문

네트워크 세상에서는 네트워크끼리의 입출입 관문에 해당하는 친구

라우터도 gateway

라우팅과 포워딩

라우팅은 길찾기

포워딩 길찾은대로 물건 전달하기

Inter-domain 간의 통신은 빠르다고 선택x 주어진 정책에 가장 부합하는 경로

레이턴시(딜레이) : 송신자가 보낸 데이터가 얼마만에 수신자에게 도달하는가

Jitter(레이턴시의 변화 정도)

Osi layer 각 레이어별로 주소가 있다.

Switch 네트워크에서 이쪽으로 가라 저쪽으로 가라

판단정보는 헤더에 있음

Round-robin 방식

서버를 바꿔주는 것 http는 h7 switch

과부하방지

스위치 : 트래픽의 방향을 바꿔준다.

AWS ELB

레이어별로 통신 규약이 있다.

헤더를 통해 할 수 있는것과 할 수 없는 것이 있다.

헤더안에 있는 crc를 통해 오류 검출

보내는 데이터 양이 많을때는 상위 계층을 쪼개서 보냄

맵핑

Ip packet

UDP TCP

UDP

100바이트보내면 100바이트 받음

TCP

데이터가 연속적인 것 같은 황상

받았는지 안 받았는 체크

수신자의 여력을 체크

공학에서 트레이드오프개념이 중요

학부생 네트워크 과정에서 알아야할 것

TCP의 중요한 속성

시퀀스넘버와 어디까지 받았는지에 대한 정보

UDP stateless

TCP stateful

TCP는 특정 상대방과만 커넥션

TCP

네트워크 경로상의 장비나 회선이 과부하 걸려서 traffic을 drop하는 것

조금씩 데이터를 보내다가 문제가

UDP 보낼 데이터의 바운더리가 명확

하지만 TCP 는 조금씩 보냄 바운더리가 불명확

UDP TCP 모두 L4프로토콜

UDP는 L3인 IP를 L4에서 쓸수 있게하는 껍데기

UDP vs TCP 전송속도? UDP가 더 빠름

반면 TCP는 서서히 속도를 올림, 이 로직이 다 헤더에 들어있음

TCP는 누락되는 것을 기다려 재전송한다.

이 때문에 속도는 더 느릴 수 있다.

UDP vs TCP

누락되어도 상관없고 가급적 빨리 전송해야 되는 것 -> UDP

일반적으로 interactive한 audio, video 등

누락되는 것을 대비해서 각각의 diagram에 충분한 데이터를 넣는다.

누락되면 안되는 데이터를 안정적으로 전송해야 되는 것 -> TCP

Ex) file 전송

넷플릭스는 UDP방식 이용

통신사와의 대립 why???? 망사용료 개념

헤더구조를 보면 제공하는것과 안하는 것을 알 수 있다.

인터넷은 네트워크를 연결하는 방식

인터넷에서 사용되는 TCP/IP 모델은 OSI7 모델의 단순한 형태이다.

이처럼 layering은 단순화와 각 layer 의 독립적 발전을 제공한다.

인터넷의 레이어 특성은 프로토콜 헤더로 결정된다

절대반지 같은 실버 불릿 프로토콜은 없다

그 속성을 잘 이해하고 있다가 상황과 목적에 따라 선택해야 된다.

선배개발자로서의 조언

개발한지가 20년이 넘음

학부때 배운걸로 밥벌어먹고 살까요? 아니에요 20년동안 굉장히 많은 새로운 프로그래밍언어도 나오고 툴도 나오고 그럼

니가 뭘 하든 하나의 강력한 스트롱을 갖고 있으면 다른거를 할 때 디딤판이 된다.

어떤 프로그래밍언어든 어떠한 기술이든 그 논리가 있다. 그 논리를 캐치하면 다른 기술도

프로그래밍에서 인생을 배운다. 비슷한 논리의 조합이다. 단순히 안다 모른다 x 이게 왜 이런 논리를 가지고 있는지 왜 이렇게 만들어졌는지 파악한다면 바뀐기술에 대한 이해도 높아짐 부트캠프의 강사로 올 정도로 실력 발전 스스로 공부하는 것에 대한 연습이 되 있어야 한다.

블록체인이란것도 p2p방식 크립토 방식 위에서 동작 새로운 기술인것처럼 얘기하지만 아니다.