1. TCP vs UDP

TCP와 UDP는 전송 계층 프로토콜

둘 다 데이터를 전송하지만 방식이 다르다

- TCP는 연결지향형 프로토콜

  통신하기 전에 3-way 핸드셰이크로 연결을 먼저 설정하고

  이후 데이터가 순서대로 잘 도착했는지 확인하면서 전송

  이 과정에서 수신자가 감당 가능한 속도로 전송 하여 흐름제어

  연결을 종료할 때는 4-way 핸드셰이크로 종료

  네트워크 상태에 따라 조절하는 혼잡제어 기능도 포함돼 있어서

  신뢰성이 매우 높은 프로토콜

- 반대로 UDP는 비연결지향형

  연결 설정 없이 그냥 바로 보내기 때문에 빠르고 단순

  다만 순서 보장, 재전송 같은 기능이 없어서 신뢰성은 떨어진다

  대신 실시간성이 중요한 곳에서 많이 사용

- TCP는 웹 브라우징(HTTP), 파일 전송(FTP), 이메일(SMTP) 같은

  정확성과 신뢰성이 중요한 경우에 사용되고

- UDP는 줌, 유튜브 스트리밍, 온라인 게임, DNS 질의 등

  약간의 손실이 있을 수 있어도 빠른 응답이 중요한 서비스에서 사용

  버퍼링으로 손실을 어느 정도 보완할 수 있다

흐름제어

수신 측에 비해 송신 측의 속도가 빠를 경우 문제가 생긴다.

수신 측에서 제한된 저장 용량을 초과한 이후에 도착하는 패킷은 손실될 수 있다.

따라서 흐름 제어는 송신 측과 수신 측의 TCP 버퍼 크기 차이로 인해 생기는 데이터 처리 속도 차이를 해결하기 위한 기법

흐름 제어 종류

- **Stop and Wait** : 매번 전송한 패킷에 대해 확인 응답을 받아야만 그 다음 패킷을 전송하는 방법
- **Sliding Window** (Go Back N ARQ) : 수신측에서 설정한 윈도우 크기만큼 송신측에서 확인 응답 없이 세그먼트를 전송할 수 있게 하여 데이터 흐름을 동적으로 조절하는 제어기법

혼잡제어

- 네트워크 내에 패킷의 수가 과도하게 증가하는 현상을 혼잡이라 하며, 혼잡 현상을 방지하거나 제거하는 기능을 혼잡제어라고 한다.
- 흐름제어가 송신측과 수신측 사이의 전송속도를 다루는데 반해, 혼잡제어는 호스트와 라우터를포함한 보다 넓은 관점에서 전송 문제를 다루게 된다.

혼잡 제어 종류

- AIMD(Additive Increase / Multiplicative Decrease) : 처음에 패킷을 하나씩 보내고 이것이 문제없이 도착하면 window 크기(단위 시간 내에 보내는 패킷의 수)를 1씩 증가시켜가며 전송하는 방법
- Slow Start : AIMD와 마찬가지로 패킷을 하나씩 보내면서 시작하고, 패킷이 문제없이 도착하면 각각의 ACK 패킷마다 window size를 1씩 늘려준다
  • AIMD 방식이 네트워크의 수용량 주변에서는 효율적으로 작동하지만, 처음에 전송 속도를 올리는데 시간이 오래 걸리는 단점이 존재했다.
- Fast Recovery (빠른 회복): 혼잡한 상태가 되면 window size를 1로 줄이지 않고 반으로 줄이고 선형증가시키는 방법 이 방법을 적용하면 혼잡 상황을 한번 겪고 나서부터는 `AIMD` 방식으로 동작한다.

---

2. DNS

DNS는 우리가 입력하는 도메인 주소를 실제 IP 주소로 바꿔주는 시스템

- 예로 `www.naver.com`을 브라우저에 입력하면

  DNS는 이걸 IP 주소 223.130.195.200 같은 숫자로 변환 해주는데

  그래야 브라우저가 서버에 실제로 접속할 수 있다

- DNS는 계층 구조로 되어 있다.

  루트 → TLD (.com, .kr) → 권한 있는 DNS 서버(Authoritative Server) 순으로

  도메인 정보를 찾아가는 구조

- 질의 방식은 재귀 질의랑 반복 질의가 있다

  브라우저가 DNS 서버에게 “얘 IP가 뭐야?”라고 물어보면,

  DNS 서버가 알아서 여러 서버를 대신 조회해서 답을 주는 게 재귀 질의고

  반복 질의는 “내가 모르니까 다른 서버에 물어봐”라고 넘겨주는 방식

- 또, DNS 응답은 캐시로 저장

  브라우저, OS, DNS 서버에 일정 시간 저장되기 때문에

  같은 요청이 다시 들어오면 빠르게 응답

  이 시간은 **TTL(Time To Live)**로 정해진다.

- 네가 웹사이트 들어가면 → 브라우저가 DNS 질의 → IP 주소 받음 → 서버 접속

DNS 레코드에는

→ A, AAAA, CNAME, MX, NS 같은 레코드가 있다

DNS 보안 문제

→ DNS 스푸핑, 캐시 포이즈닝 같은 공격이 있다

이를 막기 위해 DNSSEC이라는 보안 확장 기능을 사용

1. 재귀 질의 (Recursive Query):
   - 개념:

     클라이언트가 DNS 서버에 쿼리를 보내면, 해당 서버가 다른 DNS 서버들에게 쿼리를 반복적으로 보내면서 최종 결과를 찾아 클라이언트에게 반환

   - 과정:

     클라이언트 -> 로컬 DNS 서버 -> (재귀적으로 다른 서버에 쿼리) -> 최종 결과 반환

   - 특징:
     - 클라이언트는 최종 결과만 받으므로 간편
     - DNS 서버는 여러 단계를 거쳐 결과를 찾아야 하므로 부하가 발생할 수 있습니다.

    - 주로 일반 사용자의 웹 브라우저에서 도메인 이름을 입력할 때 사용됩니다.

2. 반복 질의 (Iterative Query):

- 개념:

  클라이언트가 DNS 서버에 쿼리를 보내면, 해당 서버는 자신이 알고 있는 최선의 정보를 제공하고, 더 자세한 정보는 다른 DNS 서버를 참고하라고 알려준다

- 과정:

  클라이언트 -> 로컬 DNS 서버 -> (로컬 DNS 서버는 다른 서버의 주소를 알려줌) -> 클라이언트 -> 다른 DNS 서버 -> (반복) -> 최종 결과 확인

- 특징:
    - 클라이언트가 여러 서버에 직접 쿼리를 보내야 하므로 복잡
    - 각 DNS 서버의 부담을 줄일 수 있다
    - 주로 DNS 서버 간의 정보 교환에 사용된다.

---

3. IP, MAC, ARP

IP랑 MAC은 네트워크에서 장치를 식별하는 주소인데,

IP는 논리적 주소고 MAC은 물리적 주소

- IP 주소는 네트워크상 위치를 나타내는 주소

  바뀔 수 있고, 라우팅이 가능해서 인터넷 전체를 연결할 수 있다

- MAC 주소는 네트워크 카드에 박혀 있는 고유한 식별자

  같은 네트워크(LAN) 내에서는 MAC 주소로 통신

- 그런데 이 둘을 이어주는 게 ARP

  ARP는 IP 주소를 기반으로 MAC 주소를 찾아주는 프로토콜인데

  해당 IP를 가진 장치가 자기 MAC 주소를 알려주는 방식

브로드캐스트

컴퓨터 네트워크에서 하나의 송신자가 네트워크에 연결된 모든 수신자에게 데이터를 동시에 전송하는 방식

유니캐스트

MAC 기반으로 상대측 IP주소를 목적지로하는 일대일 통신방식

현재 네트워크 상에서 가장 많이 사용되는 방식

IP랑 MAC 주소의 차이는

→ IP는 네트워크상의 주소고, MAC은 장치 고유 주소

IP는 바뀔 수 있지만 MAC은 보통 고정

ARP 스푸핑이란

→ ARP 응답을 위조해서 트래픽을 가로채는 공격.

가짜 MAC 주소로 속여서 다른 사람의 트래픽을 가로챔

스위치 허브 환경에선 특히 치명적이라서 보안 조치가 필요

- IPv6에도 ARP 있는가?

  → IPv6에서는 ARP 대신 **NDP(Neighbor Discovery Protocol)**를 사용