2021-09-11

-질문-

HTTP3같은 경우 TCP가 아닌 UDP로 사용되었습니다. TCP가 가지는 문제인 HOLB(Head of line Blocking)와 Handshake 때문에 레이턴시가 생기고 이를 항상 줄이는 노력을 해왔지만 UDP같은 경우는 그 과정을 아예 없애버릴 수 있어서 HTTP3 같은 경우는 UDP로 썼다고 자료를 찾아보니 나와있습니다. 그렇다면 그 2가지 과정 때문에 생기는 레이턴시 문제 말고 다른 문제가 있는지 궁금합니다. ⇒ 위의 방법과 마찬가지로 획기적으로 레이턴시를 줄이지만 없어지지 않는 부분이 있는지, 또는 TCP만의 고질적인 문제는 무엇이며 이를 가장 최소화 하기 위한 방법으로는 어떤 것이 있는지?

-설명을 듣고 난 뒤 드는 생각은 TCP를 대부분 사용한다고 했지만 UDP만의 장점도 있기 때문에 UDP의 장점을 살리면 TCP처럼 사용할 수도 있지 않을까? 라는 생각

⇒ 찾아보니 HTTP3가 UDP를 사용

점유율은 [2020년 10월 기준으로, HTTP/3 프로토콜은 인터넷 드래프트(초안)이며 여러 구현체를 보유하고 있다. W3Techs에 따르면 10,000,000개 웹사이트 중 8%가 HTTP/3를 지원하고 있다.[2] 파이어폭스와 크롬 안정판은 현재 HTTP/3을 지원하지만 기본적으로 비활성화되어 있다. macOS 빅서에서 사파리 14는 HTTP/3를 기본으로 활성화한 최초의 브라우저이다.]

HTTP/2 점유율

HTTP/3 는 기존의 HTTP/1, HTTP/2와는 다르게 UDP 기반의 프로토콜인 QUIC을 사용하여 통신하는 프로토콜이다. [HTTP/1 과 HTTP/2는 UDP가 아닌 TCP로 사용

→ UDP기반의 QUIC는 무엇인가?

QUIC : TCP 같은 프로토콜

HTTP/3는 QUIC이라는 프로토콜에서 돌아가는데 QUIC은 UDP를 사용하기 때문에 HTTP/3가 UDP로 만들어졌다고 하는 것.

[제목 없음](https://www.notion.so/0e7439142e3241f48e00e93fbe26cd24)

여기서 나오는 신뢰성이란 얼마나 내가 전송하고 싶은 내용을 정확하게 전달할 수 있느냐? 즉 발신하는 측의 데이터가 수신 쪽에 완벽하게 나올 수 있는지.

TCP는 클라와 서버간의 통신이기 때문에 레이턴시는 생길 수 밖에 없다.

이런 레이턴시를 줄이기 위해 여러 방법을 사용해보았지만 레이턴시를 아예 줄일 수는 없다.

[데이터 양이 점점 더 많아지고 더 많은 요구가 필요하기 때문]

이러한 이유 때문에 QUIC(UDP)를 사용하여 프로토콜 자체를 수정하는 것.

[Handshake에 대해]

즉 TCP를 사용하는 이상 이러한 과정을 거쳐야 한다.

[HOLB에 대해]

또한 TCP의 현상 중 HOLB 라는 것이 있는데 그 과정은 다음과 같다.

하나의 TCP연결에서 3개의 이미지(a.png, b.png, c.png)를 얻을려고 하는경우 HTTP의 요청순서는 다음 그림과 같다.

| --- a.png --- | | --- b.png --- | | --- c.png --- |

순서대로 첫번째 이미지를 요청하고 응답받고 다음 이미지를 요청하게 되는데 만약 첫번째 이미지를 요청하고 응답이 지연되면 아래 그림과 같이 두,세번째 이미지는 당연히 첫번째 이미지의 응답처리가 완료되기 전까지 대기하게 되며 이와 같은 현상을 **HTTP의 Head of Line Blocking** 이라 부르며 파이프 라이닝의 큰 문제점 중 하나이다.

| ------------------------------- a.png --------------- --- | | -b.png- | | --c.png-- |

=⇒ QUIC는 처음부터 hankshake과정을 최적화하는데 초점, UDP를 이용하여 구현

결과적으로 이러한 문제들을 해결하기 위해 HTTP/3는 UDP를 사용하여 handshake를 없앴다.

🡺 네트워크 강의를 듣고 내 스스로에게 질문을 내려서 정보를 좀 찾았다.

+

같이 네트워크 스터디 하시는 분은 퀀트(투자) 쪽으로 가실 계획이다.

퀀트 투자(?)