Network Systems 문답

# Index	5
CreatedAt	
Person	A Ally Hyeseong Kim
Status	In progress
Tags	Computer Science Network Systems
UpdatedAt	

References

Interview_Question_for_Beginner/Network at master · JaeYeopHan/Interview_Question_for_Beginner

:boy: :girl: Technical-Interview guidelines written for those who started studying programming. I wish you all the best. :space_invader: - Interview_Question_for_Beginner/Network at master · JaeYeo...

https://github.com/JaeYeopHan/Interview_Question_for_Beginner/tree/master/Network

Tech

tech_interview.zip/Network.md at main · 4z7l/tech_interview.zip

✅ 취준하면서 모았던 면접 질문 모음집 ✅. Contribute to 4z7l/tech_interview.zip development by creating an account on GitHub.

https://github.com/4z7l/tech_interview.zip/blob/main/직무/Network.md

4z7l/
tech_interview.zip

✅ 취준하면서 모았던 면접 질문 모음집 ✅

1 Contributor 1 Issue 1k Stars 132 Forks

면접 시리즈4 - 네트워크

Java, JPA, Spring을 주로 다루고 공유합니다.

https://backtony.github.io/interview/2021-12-12-interview-19/#전송-계층

https://github.com/AllyHyeseongKim/backend-interview-question

References

1. OSI 7계층에 대해 설명해주세요.

OSI 7계층은 컴퓨터 네트워크에서 통신이 이루어지는 과정을 7단계로 나눈 모델입니다.

계층화를 통해 각 단계별로 독립적으로 개발이 가능해 문제 발생시 해당 단계만 수정할 수 있도록 하였습니다.

각 계층을 대략적으로 설명하자면,

1계층, 물리계층은 데이터를 전기신호로 변환하여 전송하기 위한 계층으로 허브와 리피터 장비를 사용합니다.

2계층, 데이터링크계층은 물리계층에서의 데이터의 오류와 흐름을 제어하기 위한 계층으로 스위치를 사용합니다.

3계층, 네트워크계층은 데이터를 목적지로 전송하기 위한 라우팅을 하는 계층으로 라우터를 사용합니다.

4계층, 전송계층은 데이터를 종단 프로세스간 전송하기 위한 계층입니다.

5계층, 세션계층은 세션으로 논리적 연결을 관리하기 위한 계층입니다.

6계층, 표현계층은 데이터 표현 방식을 관리하고 암호화하기 위한 계층입니다.

7계층, 응용계층은 최종 목적지로, 애플리케이션과 관련된 서비스를 하는 계층입니다.

1~4 계층에 대해 좀 더 자세히 설명하면,

1계층 장비인 허브는 브로드캐스팅으로 데이터를 전송하고, 리피터는 신호를 증폭해주는 역할을 합니다.

2계층은 프레임 단위로 데이터를 전송하는데, 물리주소인 MAC 주소를 기반으로 통신합니다. 데이터를 출발 도착지 MAC 주소가 담긴 헤더와 CRC 알고리즘으로 추출한 checksum으로 감싸 프레임을 만듭니다. 2계층 장비인 스위치를 이용해 MAC 주소 테이블에 있는 목적지는 해당 목적지로, 없는 목적지면 모든 포트로 전송하게 합니다.

3계층은 패킷 단위로 데이터를 전송하는데, 논리주소인 IP 주소를 기반으로 통신합니다. 프레임을 출발지 도착지 IP 주소가 담긴 IP 헤더로 감싸 패킷을 만듭니다. 3계층 장비인 라우터를 이용해 라우팅합니다.

4계층은 세그먼트 단위로 데이터를 전송하는데, 포트를 기반으로 통신합니다. 패킷을 출발지 도착지 포트가 담긴 TCP 혹은 UDP 헤더로 감싸 세그먼트를 만듭니다.

1.1. 브로드캐스트, 멀티캐스트, 유니캐스트에 대해 설명해주세요.

타겟 통신 대상에 따라 나누는데, 브로드캐스트는 네트워크에 있는 모든 대상, 유니캐스트는 특정 대상과 1:1 통신, 멀티캐스트는 특정 다수 그룹과 1:N 통신을 말합니다.

여기서 브로드캐스트는 네트워크 내에서 가장 마지막 IP를 브로드캐스트 주소로 사용합니다. 또한 모든 네트워크 내 모든 로컬 호스트로 데이터를 전송하므로 성능 저하가 발생할 수 있습니다.

1.2. 브로드캐스트로 인한 성능 저하를 어떻게 해결할 수 있나요?

브로드캐스트 트래픽이 증가하면 네트워크 대역폭을 차지해 다른 데이터의 전송 속도를 늦추거나 일시적으로 마비시킬 수 있습니다. 이에 대한 해결방법으로 서브넷을 분할하는 서브네팅을 사용하여 해결할 수 있습니다.

서브네팅은 IP Class에서 Network ID와 Host ID를 분리하는 서브넷 마스크의 비트 수를 증가시켜 원본 네트워크를 여러 개의 네트워크로 분리하는 방법입니다.

예를 들어 C 클래스인 IP가 192.168.32.0인 네트워크를 가지고 있다고 할 때, 할당 가능한 host의 수는 2^8-2(네트워크 주소, 브로드캐스팅 주소)개 입니다. 이때, 서브넷 마스크의 bit 수를 하나 증가시키면, 네트워크의 수는 두개가 되고 할당 가능한 host의 수는 2^7 - 2개가 됩니다.

따라서, 서브네팅을 이용하면 네트워크를 분리해 브로드캐스트 범위를 줄일 수 있습니다.

1.3. CRC 알고리즘으로 checksum으로 검증하는 과정에 대해 설명해주세요.

CRC 알고리즘은 이진변환된 CRC-n+1 다항식으로 데이터를 XOR 연산으로 나눗셈하여 checksum을 추출하는 방법입니다.

송신 측에서는, 이진수로 이루어진 데이터에 n개의 0을 붙이고, 이진 변환된 다항식으로 XOR 연산으로 나눗셈한 나머지를 checksum으로 전송합니다.

수신 측에서는, 이진수로 이루어진 데이터에 checksum을 붙이고, 이진 변환된 다항식으로 XOR 연산합니다. 이때 나머지가 0일 경우 손상되지 않고 올바르게 도착한 데이터라고 판단합니다.

1.4. 라우터가 라우팅하는 방법에 대해 설명해주세요.

3계층 장비에 해당하는 라우터는 패킷의 목적지 IP 주소를 확인하고 해당 주소까지의 경로를 결정하는 라우팅 역할을 합니다.

이때, 라우터는 라우팅 테이블을 이용해 현재 라우터에서 다음 라우터 경로를 결정하는데, 이 라우팅 테이블을 업데이트하는 알고리즘으로 주로 두 가지가 사용됩니다.

첫번째는 모든 라우터의 연결 상태와 링크 비용을 알고 있으며 이벤트가 있을 경우 갱신하는 중앙 집중형 방식입니다. 이 방법은 주로 다익스트라 알고리즘을 사용합니다.

두번째는 각 라우터가 연결된 인접 노드에 대한 링크 비용을 알고있으며 주기적으로 라우터들에 의해 갱신되는 분산 라우팅 방식입니다. 이 방법은 주로 벨만 포드 알고리즘을 사용합니다.

1.5. 중앙 집중형 방식과 분산 라우터 방식은 각각 언제 사용하면 좋을까요?

중앙 집중형 방식은 모든 라우터의 정보를 알고있어야하므로 네트워크가 작고 간단한 경우에 적합합니다. 따라서 학교나 작은 기업의 경우 사용될 수 있습니다.

반면, 분산 라우터 방식은 모든 라우터의 정보 없이 인접 노드만 알고 있으므로 상대적으로 큰 네트워크에서 사용할 수 있습니다.

1.6. 다익스트라 알고리즘과 벨만 포드 알고리즘에 대해 설명해주세요.

1.7. TCP와 UDP에 대해 설명해주세요.

TCP와 UDP는 4계층인 전송 계층의 프로토콜로, 포트를 기반으로 프로세스 사이를 연결합니다.

먼저, UDP는 비연결형 프로토콜로 헤더에는 출발지, 도착지 포트 정보를 포함하고 TCP 헤더와 달리 연결에 필요한 다른 정보를 포함하고 있지 않습니다. 따라서 신뢰성이 낮은 대신 속도가 빨라 스트리밍 서비스에 사용됩니다.

두번째로, TCP는 연결형 프로토콜로 데이터 전송 전과 후에 handshake를 통해 연결합니다. 헤더에는 handshake를 위해 출발지, 도착지 포트 정보뿐만 아니라, SYN, ACK, FIN, sequence number(송신 데이터 순서 번호) 등을 포함합니다. 따라서 신뢰성이 높아 클라이언트와 서버 간의 데이터 신뢰성이 중요한 HTTP에 사용됩니다.

TCP의 handshake 과정에 대해 더 자세히 설명하자면,

우선, 연결을 위해 3-way handshake를 진행합니다. 먼저, 클라이언트에서 연결 요청을 위해 서버에 SYN을 보내고, 서버에서 응답과 함께 연결이 가능하다는 의미로 SYN과 ACK를 보냅니다. 마지막으로 클라이언트에서 응답으로 ACK를 보내면 클라이언트와 서버가 연결됩니다.

연결을 종료할 때는 4-way handshake를 진행합니다. 먼저, 클라이언트에서 종료 요청을 위해 서버에 FIN을 보내면, 서버에서 응답으로 ACK를 보냅니다. 이후 서버가 클라이언트의 이전 요청을 다 끝내고 클라이언트에 FIN을 보내면 클라이언트가 응답으로 ACK를 보내고 연결이 종료됩니다.

1.8. 3-way와 4-way 단계 차이가 나는 이유에 대해 설명해주세요.

TCP 연결 시에 3-way, 종료 시에 4-way 단계를 거치게 됩니다. 종료 시에 단계가 더 많은 이유는 다음과 같습니다.

이때, 연결 종료 시에는 클라이언트가 데이터 전송이 끝났다고 하더라도 서버는 아직 처리해서 응답해야할 요청들이 남아 있을 수 있습니다. 따라서 서버는 우선 FIN에 대한 응답 신호인 ACK를 먼저 보낸 후, 요청을 모두 처리한 후 FIN 메세지를 보내게 됩니다.

1.9. 서버에서 FIN flag를 전송하기 전에 전송한 패킷이 라우팅 지연이나 패킷 유실로 인한 재전송 등으로 FIN 패킷보다 늦게 도착하면 어떻게 되나요?

서버에서 전송한 패킷이 FIN 패킷보다 늦게 도착할 상황을 대비해 클라이언트는 서버로부터 FIN flag를 수신해도 time wait 동안 세션을 남겨 놓고 잉여 패킷을 기다린다고 알고있습니다.

1.10. 초기 sequence number를 0부터 시작하지 않고 난수를 생성해서 설정하는 이유를 설명해주세요.

TCP 연결 시 사용하는 포트는 시간이 지남에 따라 재사용될 수 있습니다. 따라서 과거에 사용된 포트 번호를 사용한다면, 순차적 숫자를 사용할 경우 이전 연결에서 전송된 패킷으로 인식할 수 있습니다.

1.11. TCP 프로토콜에서 흐름제어와 혼잡제어 방법에 대해 설명해주세요.

TCP 통신에서 네트워크 혼잡을 피하고 송신 측과 수신 측의 데이터 속도 차이를 제어하기 위해 흐름 제어와 혼잡 제어를 합니다.

우선, 흐름 제어는 흐름 제어 신호를 통해 송신 측의 데이터 전송 속도를 조절해 수신 측 버퍼 오버플로우와 데이터 유실을 방지하는 방법입니다. 흐름 제어를 위한 알고리즘은 Stop-and-wait 방식과 sliding window 방식이 있는데, stop-and-wait 방식은 데이터 전송 후 ACK 신호를 기다리므로 전송 지연이 발생할 수 있어 sliding window 방식을 사용합니다.

두번째로, 혼잡 제어는 송신 측에서 보내는 데이터가 라우터가 처리할 수 있는 양을 초과하지 않도록 제어하는 방법입니다. 혼잡 제어를 위한 알고리즘으로는 합 증가/곱 감소, 느린 시작, 빠른 재전송, 빠른 회복 방법이 있습니다. 이러한 알고리즘을 조합해 TCP Tahoe, TCP Reno 정책을 사용합니다.

1.12. TCP 흐름 제어 시 오류는 어떻게 제어할 수 있나요?

패킷이 손상되었거나 유실될 경우 재전송을 통해 오류를 제어합니다. 이때, 오류 제어를 구현하는 방법은 두가지가 있습니다.

첫번째는 Go-Back-N 방식입니다. 손상된 패킷부터 재전송하는 방법으로 너무 많은 데이터가 손실되면 TCP HOL 블로킹 문제가 생겨 데이터가 손실될 확률이 낮은 통신망에서 사용할 수 있습니다.

두번째는 Selective Repeat 방식입니다. 손상된 패킷만 재전송하는 방법으로 데이터가 손실될 확률이 높은 무선 통신 환경에서도 사용할 수 있습니다.

이때, TCP는 혼합형 방식을 통해 오류를 제어한다고 볼 수 있습니다. 순차적으로 일련번호를 부여해 패킷을 전송하고, 패킷 손상시 selective ACK 패킷을 보내 손상된 패킷만 재전송 받을 수 있습니다.

2. TCP/IP 4계층에 대해 설명해주세요.

실무에서는 OSI 7계층 대신 TCP/IP 4계층이 많이 사용되고, 현재는 TCP/IP 5계층으로 불린다고 알고 있습니다.

TCP/IP 4계층에서는, 물리계층과 데이터링크계층을 합쳐 인터넷 연결 계층, 세션계층, 표현계층, 응용계층을 합쳐 응용계층으로 정의합니다.

그리고 TCP/IP 5계층에서는, 세션계층, 표현계층, 응용계층을 합쳐 응용계층으로 정의합니다.

3. www.naver.com을 접속 했을 때에 대해 설명해주세요.

사용자가 브라우저에 www.naver.com 도메인을 입력하면 필요한 리소스를 서버와 통신해 출력하는 과정이 일어납니다. 이를 자세히 설명하면 다음과 같습니다.

첫번째로, 브라우저가 URL로 HSTS 목록을 조회해 해당 요청을 HTTPS로 보낼지 판단합니다. 이때, HSTS 목록에 URL이 존재하면 명시적으로 HTTP로 요청해도 브라우저가 HTTPS로 요청합니다.

두번째로, DNS Lookup을 통해 도메인에 대응되는 IP 주소를 찾습니다. DNS Lookup 과정에 대해 설명하자면, 브라우저, OS, 라우터, ISP 캐시를 순서대로 확인해 대응되는 IP 주소가 없으면 ISP의 DNS 서버에서 root DNS 서버에 query를 날리고, 순차적으로 다음 level의 DNS 서버 IP 주소를 반환받아 최종적으로 도메인에 해당하는 IP 주소를 반환받고, 이를 ISP DNS 서버에 캐싱한 후 반환합니다.

두번째로, 브라우저는 필요한 리소스를 요청하는 HTTP 요청 메세지를 생성합니다.

세번째로, 생성된 데이터를 하위 계층에 헤더들을 붙어 라우터를 통해 IP주소에 해당하는 서버로 전송합니다.

네번째로, 브라우저는 IP 주소에 해당하는 서버와 3-way handshake를 통해 TCP 연결을 합니다.

다섯번째로, TCP 연결이 완료된 후 SSL handshake를 통해 인증합니다.

여섯번째로, 서버에 HTTP 메세지를 요청하고 서버에서 HTTP 응답 메세지를 생성해 브라우저로 다시 전송합니다.

마지막으로, 브라우저에서 응답 메세지를 해석해 사용자에게 출력합니다.

3.1. SSL handshake에 대해 설명해주세요.

HTTPS 프로토콜은 기존의 HTTP 프로토콜에 SSL 프로토콜을 사용해 암호화한 프로토콜입니다. 이때, SSL은 대칭키와 공개키 방식을 사용해 SSL handshake를 통해 인증합니다. SSL handshake 동작 과정은 다음과 같이 설명할 수 있습니다.

첫번째로, 클라이언트가 서버로 클라이언트가 생성한 랜덤 데이터, 클라이언트가 지원하는 암호화 방식, 세션ID, SSL 프로토콜 버전을 포함해 Client Hello 메세지를 보냅니다.

두번째로, 서버가 클라이언트에 서버가 생성한 랜덤 데이터, 서버가 선택한 암호화 방식, SSL 프로토콜 버전을 담아 Server Hello 메세지를 보냅니다.

세번째로, 서버가 클라이언트에 CA와 서버의 공개키가 담긴 인증서를 담아 Certificate 메세지를 보냅니다. 이때, 클라이언트는 CA 리스트를 확인해 CA의 공개키로 인증서를 복호화하여 확인합니다.

네번째로, 서버가 클라이언트에 Server Hello Done 메세지를 보냅니다.

다섯번째로, 클라이언트가 서버에 대칭키를 서버 공개키로 암호화해서 담아 Client Key Exchange 메세지를 보냅니다. 이때, 클라이언트는 클라이언트의 랜덤 데이터와 서버의 랜덤 데이터를 조합해 대칭키를 생성합니다. 이후 서버는 클라이언트가 전송한 암호화된 대칭키를 서버 비공개키로 복호화해 세션 키를 얻습니다.

마지막으로, 서버가 클라이언트에 Change Cipher Spec 메세지와 Finished 메세지를 보냅니다.

SSL handshake로 생성된 세션 키를 이용해 서버와 클라이언트가 통신하고 세션이 종료될 경우 해당 세션키를 폐기합니다.