

Transport Layer Part.1

Data Link Layer : Node-to-Node delivery(link를 통해 인접한 두 노드 간에 frame을 전달하는 역할을 한다.)

Network Layer : Host-to-host delivery(두 호스트 사이의 데이터그램 전달을 담당)

Transport Layer : Process-to-Process delivery(프로세스간의 패킷 전달)

Client : 로컬 호스트의 process

Server : 원격 호스트에서 서비스를 제공하는 process , 접속을 받는 애. 미리 port number를 지정해서 다른 것들에게 알려줘야 함.

Client와 Server는 같은 이름을 가진다

Local host, Local process, Remote host, Remote process

전세계에서 유니크한 컴퓨터인데 그 컴퓨터 안에서 그 프로그램만 가지는 고유한 주소를 또 다시 부여 받음. Local host + Local process, Remote host + Remote process

Connectionless Service 은 에러 검출 및 복구 작업을 하지 않는다. 패킷은 번호가 매겨지지 않으며 지연되거나 연결이 해제될 수 있다. 대표적으로 UDP protocol

Connection-Oriented Service은 reliable하고 에러 검출 및 복구 작업을 한다. 대표적으로 TCP와 SCTP가 있다. SCTP는 통신사업자들이 주로 사용하고 멀티 라인을 사용한다. TCP는 그정도까진 아니라서 단일라인사용하고 끊어져도 별로 중요치 않음.

UDP사용 : Daytime은 운영체제 클라이언트에서 시간을 가져옴

실시간 streaming / youtube는 UDP TCP 어느것을 사용할까?

7:3 혹은 8:2로 TCP를 사용한다

5초 광고 동안 영화를 미리 다운받고 에러 검출 및 복구를 해줌으로서 영화를 깨지지 않고 잘보이게 함

윈도우 플레이어에서 UDP, TCP, SCTP, HTTP 설정 가능. 버퍼링 시간도 설정 가능하다.

Reliable Service

어플리케이션 계층 프로그램에 신뢰성이 필요한 경우, TCP 나 SCTP와 같은 전송 프로토콜을 사용한다. 하지만 에러검출과 복구작업으로 인해 더 느리고 더 복잡해진다.

Unreliable Service

어플리케이션 계층 프로그램에서 자체 흐름 및 오류 제어를 할 때 신뢰성이 불필요 한경우, 빠른 서비스가 필요할 때 UDP와 같은 프로토콜을 사용함.

Boot protocol

장치가 살아나면 네트워크로 요구를 해서 어플리케이션을 가져온다.(크기가 작아야함) 크기가 작은 것을 가져와서 부팅을 함.

장점 : 서버에다가 업데이트된 어플을 올려놓으면 로봇들이 정해진 시간에 가서 운영체제를 가져오면 된다. 개별로 로봇에다 업데이트를 할 수고가 사라짐.

UDP : connectionless service(전송된 데이터그램은 독립적임), Flow and Error control(오류 제어x), 캡슐화 및 디캡슐화.

UDP를 사용하는 이유 : 만약 어떤 프로세스가 크기가 작은 메시지를 보내고 싶어하고 신뢰성이 필요없을때 사용한다. TCP를 사용하는 것보다 송신자와 수신자 사이의 상호작용을 훨씬 덜 취하게 됨.

TCP : connection-oriented protocol. stream-oriented protocol(가상의 'stream of bytes'라는 튜브에 의해 연결된 것처럼 보이는 환경을 만듦)

Byte numbers : 세그먼트 번호 값에 대한 필드가 없지만 Sequence number, Acknowledgment number의 필드가 존재함. 이 것은 byte number를 가리킨다. 번호는 무작위로 생성되고 번호 지정은 흐름 및 오류 제어를 위해 사용됨.

Sequence number : 바이트 번호가 부여된 후, TCP는 전송되는 각 세그먼트에 시퀀스 번호를 할당. 각 세그먼트에 대한 번호는 해당 세그먼트에 포함된 첫번째 바이트 수이다.

Acknowledgment number : 당사자가 수신할 다음 바이트 수를 정의.

TCP에서 저런식으로 속도가 좌지우지될 때 예측할 수 있는 문제 : 사람마다 받는 시간이 달라서. 어떤사람은 최고점에 있고 어떤사람은 최저점에 있어서 게임을 할때 시간이 안맞는 문제. 절대시간에 가야하는 서비스는 방금 말한 거와 같이 안갈수도있음

TCP는 전송의 품질따위의 개념은 없음

해결하기 위한 아이디어 : 네개의 TCP 세션의 스트림을 뚫으면됨. 코어가 4개라면 a컴퓨터 b컴퓨터 사이에 cpu코어가 4개이므로 파일을 4개로 쪼개 보냄.<멀티코어

결과적으로 해결책은 TCP의 개수를 늘리면 됨

ex) 토렌트 p2p

스포티파이는 한번 누르는순간 바로 나옴 반응속도가 빠름

장점이 스포티파이가 빠지면 개네들끼리 피어투피어로 구축함

전송속도 말고 다른 속도를 올려보려면? 수신버퍼의 사이즈를 키우면 속도가 올라갈수 있음

Transport Layer part.2

QUIC : 2012년 구글이 만들었으며 실험적인 transport layer network protocol이다. 표준화를 위해서 IETF에 3년뒤 QUIC 규격의 인터넷 초안이 제출되었다. 2018년 IETF의 HTTP와 QUIC Working Group은 프로토콜 HTTP/3의 이름을 전세계 표준으로 만들기 전에 공식적으로 요청하였다. QUIC의 목표는 TCP기반의 연결 지향 웹 애플리케이션의 perceived performance를 개선하는 것이다. QUIC의 가장 큰 특징은 UDP를 사용하여 multiplexed 연결을 설정함. (TCP의 slow strat를 제거하여 속도를 개선.) Forward Error Correcition를 통해 자체적으로 에러 검출및 복구를 함.

QUIC은 먼저 만들어진 것은 먼저 도착하므로 지연이 적고 동시다발적인 반면에 HTTP는 순차적으로 요청하면 순차적으로 도착해야하므로 지연이 많이 생긴다.

ZeroMQ(Message Queue)

zero의 의미 : ZeroMQ는 필요한 애들끼리 통신, 항상 살아있는 애들끼리 통신. 이미 동작하고 있는 어플리케이션에서 사이에서 정보 통신을 할때.

RabbitMQ

Centralized 방식이어서 A가 B에게 정보를 전달하고 싶을때는 RabbitMQ서버를 통해서 전달해야함. (마치 카카오톡처럼) 상대방에게 무언가를 보내는 점에선 zeroMQ와 같지만 zeroMQ는 직

접 통신 선이 연결되어 있어야 하고 RabbitMQ는 상대방이 잠들어 있어도 서버로 보내기 때문에 상대방이 살아있다면 전달을 해줌, 수시로 들락날락하는 애에게 방식으로 많이 쓰인다

Network Layer part.1

Datagram Approach : 앞뒤의 관계를 고려하지않고 독립적으로 패킷을 처리하고, 기본적으로 선에대한 개념이 없는 네트워크(connectionless) 각 패킷이 독립적으로 처리되므로 목적지에 순서에 무관하게 뒤에 것이 더 빨리 도착 할 수 있음.TCP에서 방법사용
패킷은 매번 통신 할 때마다 어느 경로로 가는게 빠른가 판단하게 되는데 이때 경로를 설정해주는 장치를 라우터나 스위치라고 함. 테이블을 바탕으로 경로 설정

Virtual-Circuit Approach : 가상의 줄이 있고 그 줄을 따라 메세지들이 지나다님. 연결과 해제과정 포함됨. 패킷이 전송될 때 순서대로 패킷이 목적지에 도착.

network layer는 완벽하지 않다. 네트워크의 퍼포먼스는 delay, throughput, packet loss로 판단됨. congestion control은 성능 향상.

DHCP : IP를 동적, 자동으로 할당하여 사용할 수 있도록 해주고, 사용하지 않을 시에는 반환받아 다른 컴퓨터에 할당.

DHCP Discover->DHCP Offer->DHCP Request->DHCP ACK의 과정을 통해 아이피 주소를 할당하게 됨.

NAT : 개인 주소와 범용 주소간의 mapping을 제공하고 동시에 가상 사설 네트워크를 지원할 수 있는 기술. 사이트가 내부 통신을 위해 일련의 개인 주소와 나머지 다른 통신을 위해 일련의 글로벌 인터넷 주소를 사용할 수 있게 해준다.

IPv4 Address : TCP/IP 프로토콜 IP 계층에서 각 장치의 인터넷 연결을 식별하는데 사용되는 식별자를 인터넷 주소 또는 IP address라고 한다. 호스트나 라우터의 인터넷 연결을 보편적으로 정의하는 32비트 연결 주소.

ICMPv4 :

Network Layer part.2

어떤 기술이 오래갈까?(1:34:00)

기술영업, 기획 등

SDN : 물리적 또는 가상 네트워크 환경에서 트래픽의 유연한 전달 및 제어

NFV : 네트워크 및 클라우드 전반에 걸친 유연한 배치 기능

SDN&NFV는 완전한 네트워크 프로그래밍 가능성을 달성하기 위한 보완 툴이다.