



UDP

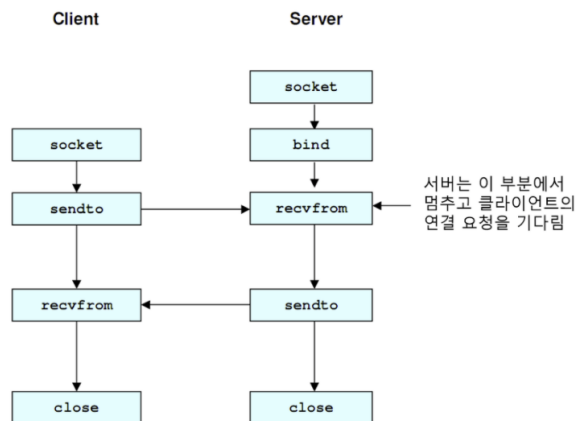
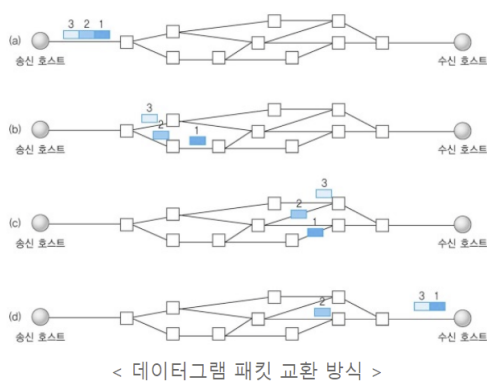
UDP (User Datagram Protocol)

| 비연결형, 신뢰성이 없는 전송 프로토콜!

데이터를 데이터그램 단위로 처리하는 프로토콜이다(독립적인 관계를 지니는 패킷)

UDP는 **비연결형 프로토콜**이다. 즉, 할당되는 논리적인 경로가 없고 각각의 패킷이 다른 경로로 전송되고 이 각각의 패킷은 독립적인 관계를 지니게 되는데, 이렇게 데이터를 서로 다른 경로로 독립 처리 하는 프로토콜을 UDP라고 한다.

UDP는 연결을 설정하고 해제하는 과정이 존재하지 않는다. 서로 다른 경로로 독립적으로 처리함에도 **패킷에 순서를 부여하여 재조립하거나 흐름제어 및 혼잡제어를 수행하지 않아** 속도가 빠르며 네트워크 부하가 적다는 장점이 있지만 데이터 전송의 신뢰성이 낮다. 연속성이 중요한 실시간 서비스(스트리밍)에 좋다.



UDP 특징

- 비연결형 서비스로 데이터그램 방식을 제공한다
- 정보를 주고 받을 때 정보를 보내거나 받는다는 신호절차를 거치지 않는다.
- UDP헤더의 CheckSum 필드를 통해 최소한의 오류만 검출한다.
- 신뢰성이 낮다
- TCP보다 속도가 빠르다

TCP, UDP 등장 배경

- IP의 역할은 Host to Host (장치 to 장치)만을 지원한다. 장치에서 장치로의 이동은 IP로 해결되지만 하나의 장비안에서 수많은 프로그램들이 통신할 경우에는 IP만으로 한계가 있다.

⇒ 포트 번호

- IP에서 오류가 발생하면 ICMP에서 알려준다. 하지만 ICMP는 알려주기만 할 뿐 대처를 하지 못하기 때문에 IP 보다 위에서 처리해야 한다.

ICMP : 인터넷 제어 메시지 프로토콜로 네트워크 컴퓨터 위에서 돌아가는 운영체제에서 오류 메시지를 전송 받는데 주로 쓰임

⇒ TCP, UDP

UDP와 TCP의 오류 해결 방식

- TCP

데이터의 분실, 중복, 순서가 뒤바뀔 등을 자동으로 보정해줘서 송수신 데이터의 정확한 전달을 할 수 있도록 해준다.

- UDP

IP가 제공하는 정도의 수준만을 제공하는 간단한 IP 상위 계층의 프로토콜이다. TCP와는 다르게 에러가 날 수도 있고 재전송이나 순서가 뒤바뀔 수도 있어서 이 경우 어플리케이션에서 처리하는 번거로움이 존재한다.

UDP를 사용하는 이유

신뢰성 보다 속도가 중요한 경우!

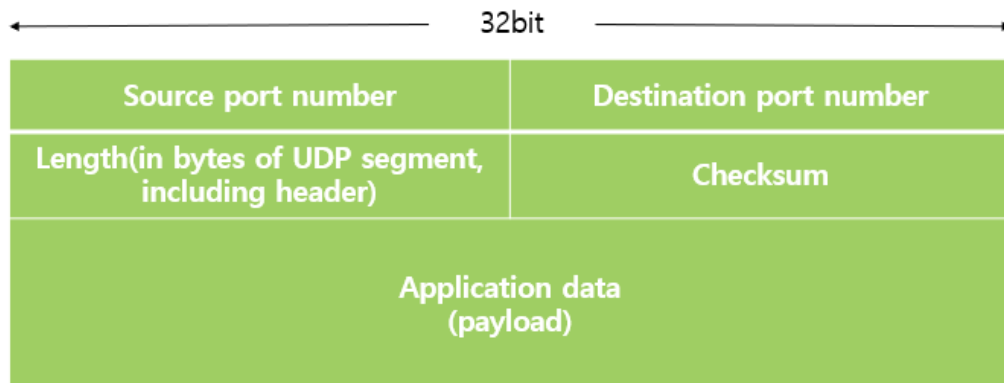
UDP는 TCP와 다르게 흐름제어나 오류제어 등이 없기 때문에 전송 속도를 최대한 빠르게 할 수 있다. 하지만 TCP처럼 신뢰성 있는 전송을 보장할 수 없다. 따라서 신뢰성 보다 속도가 중요한 부문에서 UDP를 사용한다.

예를 들어 유튜브 동영상 같은 스트리밍 어플리케이션은 신뢰성보다 속도가 중요하므로 UDP를 사용한다. 온라인 게임에서도 사용된다. 네트워크 환경이 안좋을 때 끊기는 현상을 생각하면 된다.

또한 DNS에도 사용된다.

UDP Checksum

Checksum : 중복 검사의 한 형태로, 오류 정정을 통해 공간이나 시간 속에서 송신된 자료의 무결성을 보호하는 단순한 방법이다.



Checksum : 전송된 segment의 에러를 탐지하기 위한 것

- 도착 IP 주소, 송신 포트번호, 수신 포트번호, 데이터 길이, payload 등의 데이터들을 16 비트 단위로 쪼개서 전부 더한다.
- 만약 더하는 도중 overflow 되서 carry-out된 값이 있다면 결과에 다시 더해서 sum 값을 만든다.
- 계산한 sum값을 1의 보수를 취하면 checksum값이 된다.
- 송신자
 - 전송 전 Checksum에 0이 저장되어 있다.
 - 송신 포트번호, 수신 포트번호, 길이, 페이로드를 모두 더한 뒤 1의 보수를 취한 값을 checksum에 저장하여 수신자에게 전송한다.
- 수신자
 - 수신된 segment의 checksum을 계산한다.
 - 계산된 checksum이 checksum 필드의 값과 같은지 확인한다.
 - 같지 않음

오류가 감지됨

- 같음

오류가 감지되지 않음.

데이터를 더한다		1	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
		1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
Carry-out →	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1
		다시 더한다															
sum		1	0	1	1	1	0	1	1	1	0	1	1	1	1	0	0
1의 보수를 취하면 -> checksum		0	1	0	0	0	1	0	0	0	1	0	0	0	0	1	1

하지만 이 방법에도 결함이 있다. 전송 도중 checksum 값이 바뀔 수도 있고 데이터가 변형 되었음에도 불구하고 checksum값이 동일한 경우도 발생할 수 있다.

DNS와 UDP

IP 네트워크 상에서 사람이 기억하기 쉽게 문자로 만들어진 도메인을 숫자로 된 IP 주소로 바꾸는 서버를 DNS라 한다.

DNS는 응용 계층 프로토콜이다. 모든 응용 계층은 TCP, UDP중 하나의 트랜스포트 계층을 사용해야 한다.

DNS가 UDP를 사용하는 이유

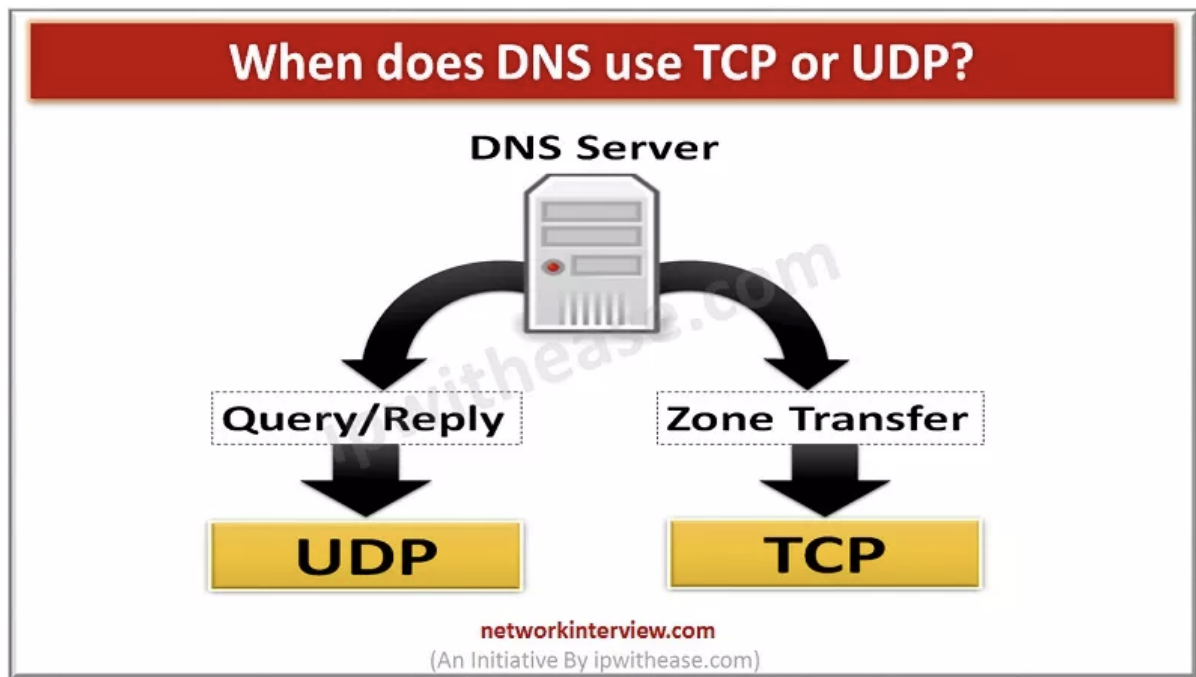
1. TCP는 데이터 전송을 시작하기 전에 3-way-handshake를 사용하는 반면에 UDP는 형식적인 예비 동작이 없다. 즉, **연결 설정에 드는 비용이 없다**. 또한 연결 상태를 유지할 필요도 없다. TCP는 end point에서 연결 상태를 유지하며 여러 정보를 기록하고 있지만 UDP는 어떠한 정보도 기록하지 않고 기록할 필요도 없다. 따라서 특정 애플리케이션에 할당된 서버는 애플리케이션이 TCP보다 UDP에서 동작할 때 **더 많은 클라이언트를 수용할 수 있다**. 도메인 네임을 IP로 변경함으로 항상 많은 클라이언트를 수용하는 dns 서버에게는 **연결 상태를 유지하지 않아 정보 기록을 최소화 하는 UDP가 알맞다**.
2. DNS Request는 UDP segment에 꼭 들어갈 정도로 작다.

DNS 쿼리는 single UDP request와 서버로부터의 single UDP reply로 구성되어 있음

DNS의 쿼리의 사이즈는 512바이트 제한이 있지만 UDP의 세그먼트 사이즈는 최대 65536바이트 정도 된다.

3. UDP가 신뢰성이 없지만 신뢰성은 응용 계층에서 추가될 수 있다. 예를 들어 타임아웃을 건다거나 재전송 작업을 통해 확보할 수 있다.

DNS는 신뢰성보다 속도가 더 중요하고, 많은 클라이언트를 수용하는 것을 필요로 한다. 따라서 속도가 빠르고, 연결 상태를 유지하지 않고 정보 기록을 최소화하여 많은 클라이언트 수용이 가능한 UDP를 사용한다.



하지만 TCP를 사용하는 경우도 있다. Zone Transfer(DNS 서버 간의 요청을 주고 받을 때 사용하는 Transfer)를 사용해야 하는 경우에는 TCP를 사용해야 한다. 또한 데이터가 512바이트를 넘거나 응답을 못 받은 경우도 TCP로 한다.