네트워크 개론 Part 12
: TCP와 UDP

글. 조인준 KBS 미디어기술연구소 차장

방송제작을 위한 비디오/오디오 신호 전송 방식이 SDI에서 IP로 전환되는 시점에서 IP 방식의 실시간 비디오/오디오 전송 프로토콜인 RTP(Real-time Transfer Protocol)를 방송제작에 적합한 프로파일로 표준화한 SMPTE ST.2022와 ST.2110 표준에 관한 내용을 전해드리면서, 단순히 비디오/오디오 신호의 IP 전송 프로토콜을 설명하는 것으로는 부족함을 느꼈습니다. 이에 RTP에 기반한 SMPTE ST.2022와 ST.2110을 통해 비디오/오디오를 실어 보내는 IP 네트워크에 관한 이해를 위해 네트워크 개론 연재를 시작하였고, 이번 연재에서 RTP 기반의 SMPTE ST.2022와 ST.2110에 관한 지난 설명들과 네트워크 개론 연재가 만나는 지점인 TCP와 UDP에 관한 내용에 이르렀습니다.

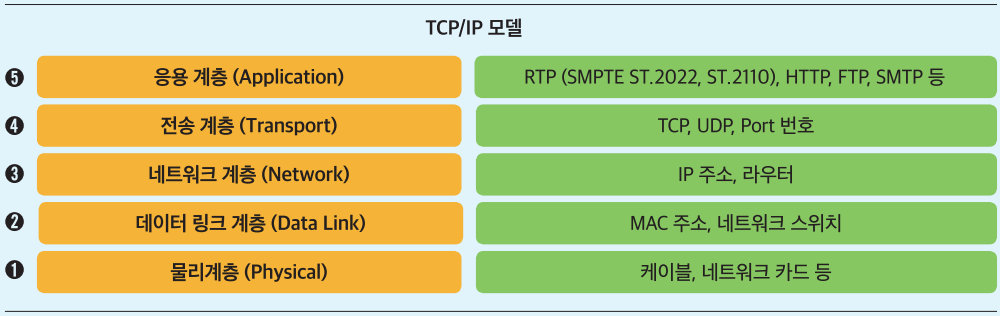

그림 1. TCP/IP 네트워크 통신 모델

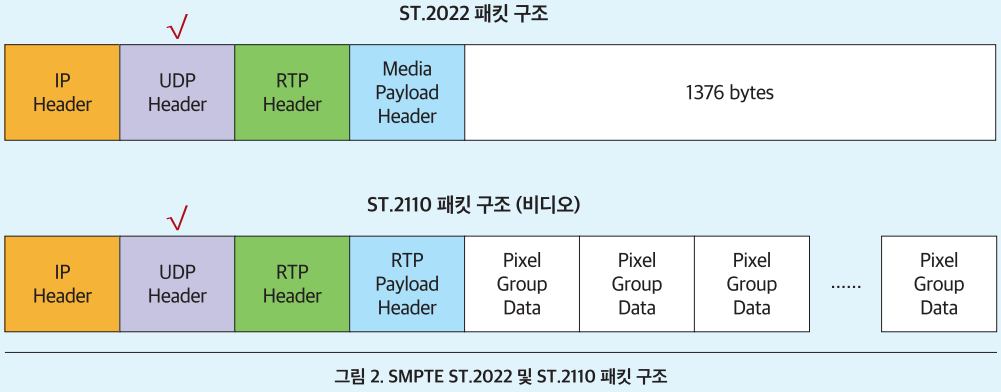

그림 2. SMPTE ST.2022 및 ST.2110 패킷 구조

[그림 1]에서 RTP에 기반한 SMPTE ST.2022와 ST.2110은 응용계층에 속합니다. 이 응용계층의 프로토콜은 [그림 2]와 같이 전송계층의 프로토콜로 감싸고, 다시 네트워크 계층 및 데이터링크 계층의 프로토콜로 감싸서([그림 2]에서는 네트워크 계층의 헤더까지만 표현) 호스트와 호스트 간에 전송되며, 응용

계층의 RTP를 감싸는 전송계층 프로토콜이 UDP(User Datagram Protocol)입니다. 또한, UDP와 같이 전송계층의 프로토콜로는 TCP(Transmission Control Protocol)가 있으며 이 두 프로토콜이 이번 연재의 주제입니다.

TCP와 UDP

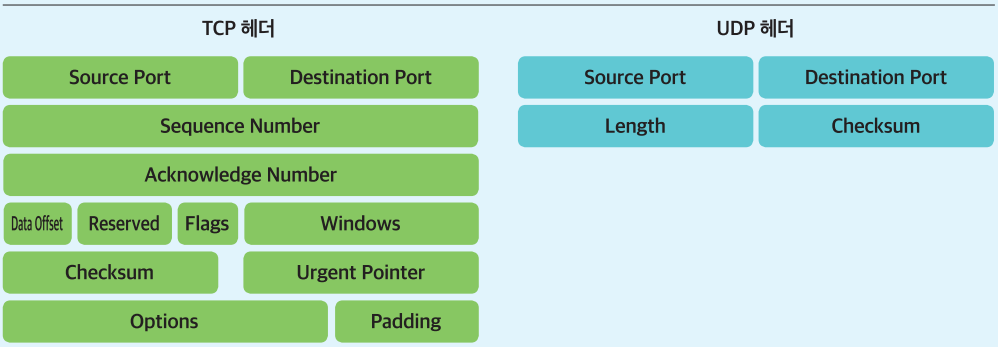

그림 3. TCP 헤더와 UDP 헤더

[그림 3]은 TCP와 UDP의 헤더입니다. 둘 다 Source Port와 Destination Port를 가지고 있다는 공통점이 있는 반면, TCP 헤더가 UDP 헤더에 비해 복잡한 구조를 가지고 있는 것을 볼 수 있습니다. 앞선 연재에서 몇 차례 포트 번호를 간략히 설명하며 패킷을 받아서 처리하는 애플리케이션을 구분하기 위한 번호라고 소개드렸습니다. 포트 번호는 0 ~ 65535 사이의 번호를 네트워크 통신을 이용하는 애플리케이션에 할당하여, 하나의 IP 주소로 여러 개의 네트워크 통신을 하고 있는 호스트에서 들어오는 패킷을 구분하여 해당 애플리케이션에 연결하기 위한 용도로 사용하는 번호입니다. 포트 번호에는 Well-Known Port, Registered Port, Dynamic Port의 세 종류가 있으며 그 특징은 다음과 같습니다.

Well-Known Port (잘 알려진 포트 번호)

Well-Known Port는 인터넷을 위한 HTTP(Hyper Text Transfer Protocol) 포트 번호 80, 이메일을 위한 SMTP(Simple Mail Transfer Protocol) 포트 번호 25, 파일 전송을 위한 FTP(File Transfer Protocol) 포트 번호 21 등 우리가 일상적으로 사용하는 특정 애플리케이션을 위해 IANA(Internet Assigned Numbers Authority, 인터넷 할당 번호 관리기관)에서 할당한 0번부터 1023번까지의 범위에 속하는 포트 번호입니다. 강제성은 없어서 지정되어 있는 포트 번호를 다른 용도로 사용할 수도 있고, 반대로 지정되어 있는 포트 번호 대신 다른 포트 번호를 사용할 수도 있습니다.

Registered Port (등록된 포트 번호)

Registered Port는 1024번부터 49151번까지의 포트 번호이며, 특정 용도로 사용하기 위해 IANA가 할당한 포트 번호이지만 Well-Known Port와 마찬가지로 강제성은 없습니다. HTTP 포트 번호 80을 대체하는 8080이나, 원격접속을 위한 포트 번호 3389 등이 Registered Port입니다.

Dynamic Port (동적 포트 번호)

Dynamic Port는 49152부터 65535까지의 포트 번호이며 Well-Known Port, Registered Port와 달리 임의로 필요에 따라 사용이 가능한 포트 번호입니다.

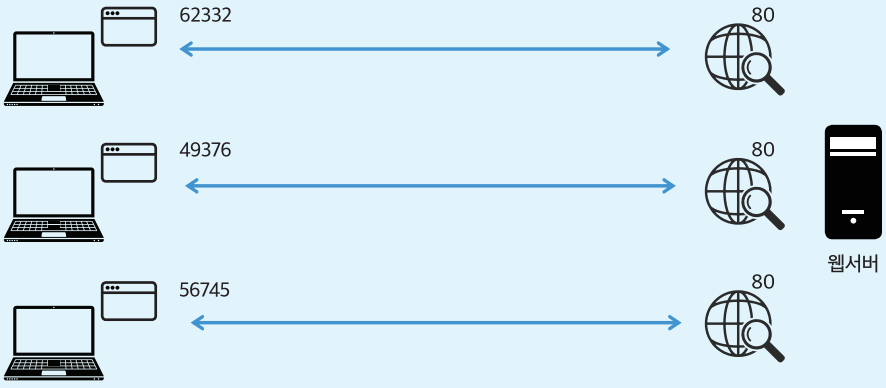

그림 4. 서버 애플리케이션 포트와 클라이언트 애플리케이션 포트

웹서버의 애플리케이션은 80번 포트로 클라이언트 애플리케이션(웹브라우저)들의 요청을 받아 처리하기 때문에 각 클라이언트는 자신의 IP 주소와 브라우저에 동적으로 할당된 포트 번호를 이용하여 웹서버의 IP 주소와 80번 포트로 요청을 보내며, 웹서버는 클라이언트의 IP 주소와 포트 번호로 각 클라이언트에서 동작하는 애플리케이션들을 특정하여 회신합니다. 이런 이유로 동일한 클라이언트에서 여러 개의 브라우저를 띄워 같은 웹사이트에 접속해도 각 브라우저에 할당된 포트 번호가 다르기 때문에, 서버는 각 브라우저에 개별적으로 반응할 수 있어서 각 브라우저가 독립적으로 동작할 수 있습니다.
참고로 웹서버는 80번 포트 이외의 포트로도 요청을 받을 수 있으며, 만약 8000번 포트를 사용한다고 하면 www.ooo.com같은 웹사이트 주소 뒤에 www.ooo.com:8000 같이 명시적으로 포트 번호를 붙여주면 됩니다. 우리가 www.ooo.com같이 포트 번호 없이 인터넷 주소를 사용하여 웹사이트에 접근할 수 있는 것은 80번을 HTTP 서버의 기본 포트로 가정하여 브라우저가 www.ooo.com:80같이 자동으로 포트 번호를 붙여 요청을 보내기 때문입니다.

조금 더 세부적으로 들어가면 네트워크 통신에서 서버 애플리케이션과 클라이언트의 애플리케이션은 소켓을 통해 통신합니다. 소켓은 [표 1]과 같이 로컬 IP, 로컬 포트 번호 및 프로토콜 정보를 통해 각 로컬로 들어오는 데이터를 해당 애플리케이션으로 연결해줍니다. 애플리케이션마다 포트 번호가 하나씩이 할당되니 포트 번호로 구분되는 소켓을 통해 들어오는 데이터를 애플리케이션으로 보내는 것에 문제가 발생할 여지가 없습니다.

로컬 IP	로컬 포트	프로토콜
121.53.200.215	80	TCP

표 1. 로컬의 소켓 관련 정보

그러나 웹서버같이 포트 번호 하나로 다양한 클라이언트들에게 서비스를 제공하는 경우 각 클라이언트에 대응되어 생성된 소켓은 [표 1]과 같은 로컬 정보만을 보면 모두가 같은 정보를 가지고 있기 때문에 식별할 수 없습니다. 이 경우 서버는 자신의 어떤 소켓이 어떤 클라이언트와 연결된 줄 알고 요청을 보낸 클라이언트에게 회신 데이터를 보낼 수 있을까요? 이를 위해 서버 측에서는 서버와 클라이언트 양쪽 소켓의 정보를 [표 2]와 같이 묶어서 5-tuple(5개의 요소로 된 집합)로 구분하면 각각의 클라이언트

애플리케이션에 연결된 서버의 소켓을 구분하는 것이 가능하고, 이를 통해 특정 데이터를 특정 클라이언트에게 보낼 수 있습니다. 서버 측에서 보면 [표 2]의 로컬 IP와 포트 번호는 서버 측의 IP와 포트 번호이고, 리모트 IP와 포트 번호는 클라이언트 측의 IP와 포트 번호입니다.

로컬 IP	로컬 포트	리모트 IP	리모트 포트	프로토콜
121.53.200.215	80	172.19.235.3	56745	TCP

표 2. 클라이언트 식별을 위한 소켓의 5-tuple 정보

지금까지 TCP와 UDP의 이해를 위해 필수적인 포트 번호에 대해서 알아보았습니다. 지금까지의 네트워크 개론 연재를 종합하면 데이터 링크 계층의 헤더에는 MAC 주소, 네트워크 계층 헤더에는 IP 주소, 전송 계층 헤더에는 포트 번호 정보 있으며, MAC 주소, IP 주소 및 포트 번호를 이용하여 어떻게 호스트 간에 데이터가 전송되고, 전송된 데이터가 호스트 내부의 애플리케이션을 어떻게 찾아가는지 큰 그림을 알 수 있습니다.

그렇다면 마지막으로 전송계층에는 왜 TCP와 UDP 두 가지 타입의 프로토콜이 존재하는 걸까요? 이 질문에 대한 대답을 [표 3]에 간략히 정리했습니다. 결론부터 말하면 안정성이 중요한 경우를 위해 TCP 프로토콜을 이용하며, 속도가 중요한 경우를 위해 UDP 프로토콜을 이용합니다. TCP는 연결 상태를 보장하기 위한 핸드-셰이킹, 네트워크의 혼잡 및 수신 측의 버퍼 오버플로우 방지를 위한 기능 등을 제공합니다. 하지만 UDP는 수신 측이 데이터를 받고 있는지 전송에 문제가 있는 지 등을 확인할 수 있는 기능을 제공하지 않습니다. 이는 [그림 3]에서 TCP 헤더가 UDP 헤더보다 훨씬 복잡한 구조를 갖고 있는 것을 보아도 쉽게 짐작할 수 있는 부분입니다. 하지만 UDP는 TCP 대비 속도가 빠르고 네트워크 부하가 적어서 신뢰성을 위해 전송에 실패한 데이터를 다시 보내줄 필요가 없고 무엇보다 속도가 중요한 실시간 비디오 스트리밍 등에 자주 사용됩니다. SMPTE ST.2022와 ST.2110이 TCP가 아닌 UDP를 사용하는 것도 바로 이 이유입니다.

구분	TCP	UDP
기능	안정적 송신을 위한 기능 다수	단순한 전송 관련 기능만 제공
연결	연결형 프로토콜	비연결형 프로토콜
오류복원	지원	지원하지 않음
흐름제어	지원	지원하지 않음
전송순서	데이터 전송순서 보장	데이터 전송순서 보장하지 않음
통신방식	1:1	1:1, 1:N, N:N
용도	신뢰성이 필요한 통신	속도가 중요한 실시간 통신 등

표 3. TCP와 UDP 비교

다음 연재에서는 TCP와 UDP의 상세 내용을 통해 좀 더 자세한 메커니즘을 설명 드리도록 하겠습니다.