

네트워크에 대한 학습 내용 정리

OSI 7계층

OSI 7계층은 네트워크 통신이 일어나는 과정을 7단계로 나눈 것

OSI 7계층을 나눈 이유는?

흐름을 한눈에 알아보기 쉽고 7단계 중 특정한 곳에 이상이 생기면 다른 단계의 장비 및 소프트웨어를 건들이지 않고도 이상이 생긴 단계만 고칠 수 있음

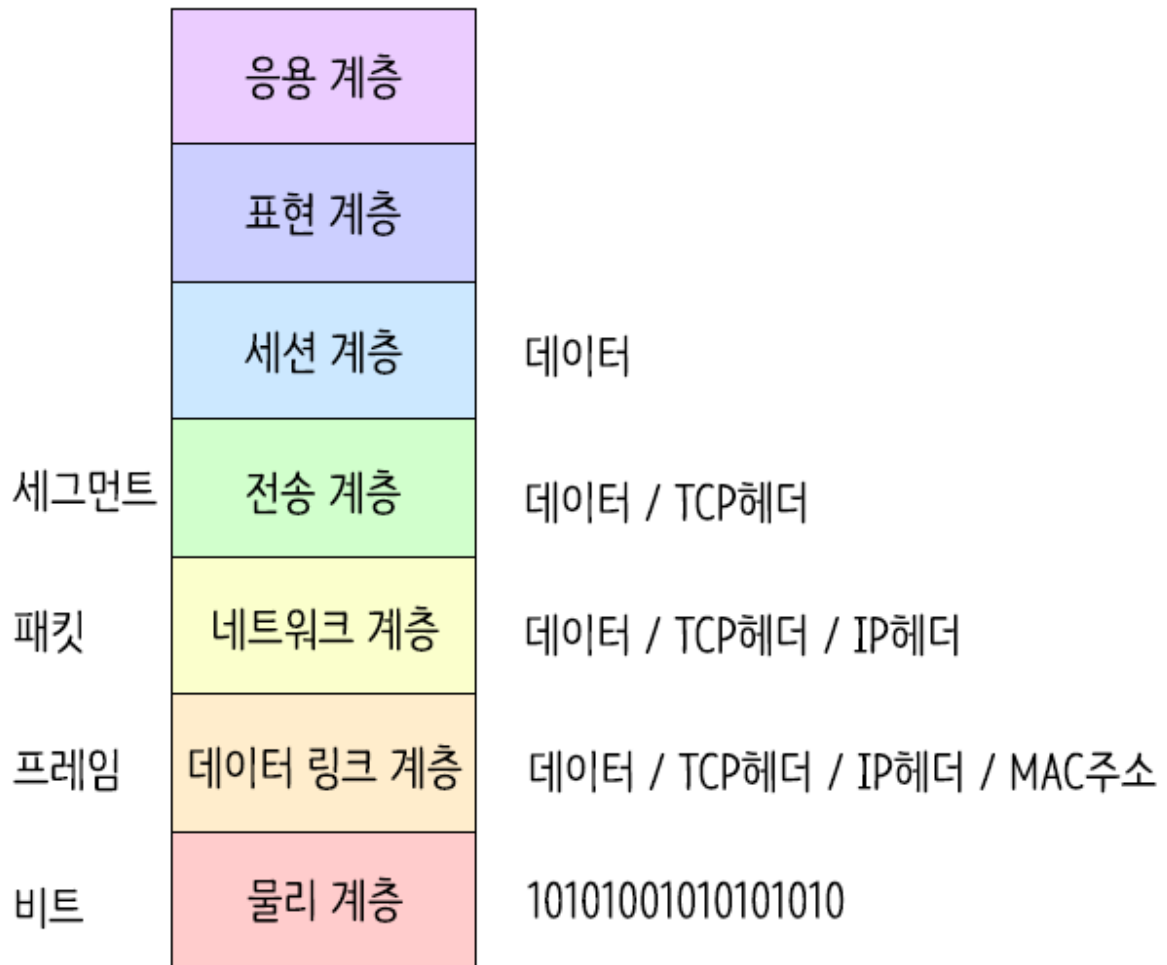
Tip

pc방에서 롤을 하고 있는데 연결이 끊겼다.

어디에 문제가 있는지 확인을 하기 위해서는?

- 모든 PC에 문제가 있다면 라우터의 문제(3계층 네트워크 계층)이거나 광랜을 제공하는 회사의 회선 문제(1계층 물리계층) 일 것이다.
- 한 PC만 문제가 있고 롤 소프트웨어에 문제가 있다면 → 7계층 어플리케이션 계층 문제
- 롤 소프트웨어에 문제가 없고 스위치에 문제가 있다면 → 2계층 데이터링크 계층 문제

OSI 7계층 종류



1. 물리 계층(Physical layer)

- 전기적 신호가 나가는 물리적인 장비
- 이 계층에서는 단지 데이터를 전달할뿐, 전송하려는(또는 받으려는) 데이터가 무엇인지, 어떤 에러가 있는지 등에 대해서는 신경쓰지 않는다.
- 단지 데이터를 전기적인 신호로 변환해서 주고받는 기능만 있을 뿐이다.

Tip

전송단위 : **비트**(Bit)장비 : 케이블, 허브



허브



리피터

2. 데이터 링크 계층(Data link layer)

- 물리계층을 통해 송수신되는 정보의 오류와 흐름을 관리하여 안전한 정보의 전달을 수행할 수 있도록 도와주는 역할. 따라서 통신에서의 오류도 찾아주고 재전송도 하는 기능을 가지고 있다.
- 이 계층에서는 맥 주소를 가지고 통신하게 된다.
- 데이터 링크 계층은 포인트 투 포인트(Point to Point)간 신뢰성 있는 전송을 보장하기 위한 계층으로 CRC기반의 오류 제어와 흐름 제어가 필요하다.

흐름 제어 : 송신 측과 수신 측의 속도 차이를 조정

오류 제어 : 오류 검출과 회복

순서 제어 : 프레임의 순서적 전송

프레임 동기화 : 프레임의 시작과 끝을 구별하기 위한 동기화

단위 : Frame

Tip

전송단위 : **프레임**(Frame)장비 : 브릿지, 스위치, 이더넷



스위치

3. 네트워크 계층(Network layer)

- 경로(Route)와 주소(IP)를 정하고 패킷을 전달해주는 것이 이 계층의 역할이다.
- 즉, 목적지까지 가장 안전하고 빠르게 데이터를 보내는 기능을 말한다. 따라서 **최적의 경로를 설정**해야 한다.
- 이런 라우팅 기능을 맡고 있는 계층이 네트워크 계층이다.

Tip

전송단위 : **패킷**(Packet/Datagram)

4. 전송 계층(Transport layer)

- 양 끝단의 사용자들 간의 **신뢰성있는 데이터**를 주고 받게 해주는 역할을 한다.
- 송신자와 수신자 간의 신뢰성있고 효율적인 데이터를 전송하기 위하여 **오류검출 및 복구, 흐름제어와 중복검사** 등을 수행한다.
- 데이터 전송을 위해서 **Port 번호**가 사용이 된다. 대표적인 프로토콜로는 **TCP와 UDP**.

Tip

전송단위 : **세그먼트**(Segment)

5. 세션 계층(Session layer)

- 응용 프로세스가 통신을 관리하기 위한 방법을 정의한다.
- 이 계층은 **TCP/IP 세션을 만들고 없애는 역할**을 하고 있다.
- API, SOCKET

소켓 통신



소켓 통신

• 단방향 통신인 Http 통신

Http 통신은 Client의 요청(Request)이 있을 때만 서버가 응답(Response)하여 해당 정보를 전송하고 곧바로 연결을 종료하는 방식입니다. Client가 요청을 보내는 경우에만 Server가 응답하는 단방향 통신으로 반대로 Server가 Client에게 요청을 보낼 수는 없습니다.

• 양방향 통신인 Socket 통신

Server와 Client가 특정 Port를 통해 실시간으로 양방향 통신을 하는 방식입니다. Http 통신과는 다르게 Server와 Client가 특정 Port를 통해 연결되어 있어서 실시간으로 양방향 통신을 할 수 있습니다.

Streaming 중계나 실시간 채팅, 게임 등과 같이 즉각적으로 정보를 주고받는 경우에 사용됩니다.

- 소켓은 응용프로그램에서 TCP/IP를 이용하는 창구 역할을 하며, 두 프로그램이 네트워크를 통해 서로 통신을 수행할 수 있도록 양쪽에서 생성되는 링크의 단자입니다. 두 소켓이 연결되면 서로 다른 프로그램이 서로 데이터를 전달할 수 있게 됩니다.

이러한 Socket 통신은 일련의 규칙이 정해져 있는데요.

1. 먼저 기다리는 측을 Server라고 하며, Server에서는 Port를 열고 Client의 접속을 기다립니다.
2. 그리고 접속하는 측을 Client라고 하며, Server의 IP와 Port에 접속하여 통신이 연결됩니다.
3. Server와 Client 간의 통신은 Send, Receive의 형태로 주고받습니다.
4. 그리고 통신이 끝나면 close()로 접속을 끊습니다.

▼ 채팅 서버 구현 코드

1. 사용자가 메시지를 입력하면 서버가 이를 이벤트로 받음. io.on으로 출력 (client)

```
form.addEventListener('submit', function(e) {
  e.preventDefault();
  if (input.value) {
    socket.emit('chat message', input.value);
    input.value = '';
  }
});
```

2. 모든 사람에게 이벤트를 보내기 위해 io.emit() 사용 (server)

```
io.on('connection', (socket) => {
  console.log('a user connected...')
  socket.on('chat message', msg => {
    io.emit('chat message', msg);
  });
});
```

3. 클라이언트 측에서 chat message 이벤트를 보여줌 (client)

```
socket.on('chat message', function(msg) {
  var item = document.createElement('li');
  item.textContent = msg;
  messages.appendChild(item);
  window.scrollTo(0, document.body.scrollHeight);
});
```

6. 표현 계층(Presentation layer)

- 전송하는 **데이터의 표현방식**을 결정한다. (예: 데이터변환, 압축, 암호화 등)
- GIF, JPEG, ASCII 등

표현계층은 3가지 기능을 가지고 있다.

1. 송신자에서 온 데이터를 해석하기 위한 응용계층 데이터 부호화, 변화
2. 수신자에서 데이터의 압축을 풀수 있는 방식으로 된 데이터 압축
3. 데이터의 **암호화와 복호화**

인코딩이나 암호화 등의 동작이 표현계층에서 이루어진다.

7. 응용 계층(Application layer)

- 사용자와 가장 가까운 계층이 바로 응용 계층이다.
- 우리가 사용하는 응용 서비스나 프로세스가 바로 응용계층에서 동작.
- **HTTP, FTP 등의 프로토콜**이 응용 계층에 속한다.