OSI 7 계층

OSI 7 계층이란?

다수의 시스템을 서로 연결해서 통신하려면 선행적으로 전체 시스템 구조를 표준화해야 합니다. 그래서 국제 표준화 단체인 ISO(International Standard Organization)에서는 **OSI(Open Systems Interconnection) 7 계층** 모델을 제한하여, 네트워크에 연결된 시스템이 갖추어야 할 기본 구조와 기능을 **응용 계층**, **표현 계층**, **세션 계층**, **전송 계층**, **네트워크 계층**, **데이터 링크 계층**, **물리 계층**의 계층적인 구조로 세분화하였습니다.

일반 사용자는 **응용 계층을 통해 데이터 전송을 요청**하며, 이 요청은 물리 계층까지 순차적으로 전달되어 상대 호스트에 전송됩니다. 그리고 상대 호스트에게 전송된 데이터는 물리 계층에서 순차적으로 응용 계층까지 다시 전달됩니다.

데이터가 하위 계층으로 내려갈 때는 각 계층의 프로토콜에서 정의한 헤더 정보가 추가됩니다. 물리 계층을 제외한 모든 계층에서 헤더 정보가 추가되고, 물리 계층은 단순히 데이터 링크 계층에서 수신한 데이터를 수신 호스트의 물리 계층에 전달합니다. 데이터를 수신하는 호스트에서는 반대로 상위 계층으로 올라가며 순차적으로 헤더 정보를 제거하고 해석하면서 프로토콜 기능을 수행합니다.

1 계층 - 물리 계층

네트워크에서 데이터를 전송하려면 반드시 물리적인 전송 매체로 연결되어 있어야 합니다. 데이터를 전기적, 기계적, 물리적인 신호로 변환하고 전송하는 역할을 하며, 크게 유선 매체와 무선 매체로 구분됩니다. 물리 계층에서 사용되는 신호는 데이터를 전송하는 매체에 따라 달라질 수 있습니다. 예를 들어, 유선 매체에서는 전기 신호가, 무선 매체에서는 라디오 파장이 사용됩니다.

이때 전송 매체에서는 디지털 형태인 0과 1의 비트 형태로 통과 시킬 수 없습니다. 따라서 **디지털 신호를 전송 가능한 아날로그 신호(전기 신호, 빛, 라디오 파장 등)로 변환하는 과정이 필요**합니다. 이것이 물리 계층이 하는 역할이라고 할 수 있습니다.

물리 계층에서 주로 사용되는 장비로는 케이블, 허브, 리피터, 모뎀 등이 있습니다. 케이블은 데이터 전송 매체로 사용되며, 허브는 여러 대의 컴퓨터를 연결하여 네트워크를 형성하는 역할을 합니다.

2 계층 - 데이터 링크 계층

물리 계층으로 데이터를 전송하는 과정에서는 잡음 등과 같은 여러 외부 요인에 의하여 물리적인 오류가 발생할 수 있습니다. 데이터 링크 계층은 이와 같은 **물리 계층의 오류에 관한 오류 제어 기능을 수행**하며, 이를 위해서는 오류의 발생 사실을 인지하는 기능과 오류 복구 기능이 필요합니다. 물리 계층은 일차적으로 물리적 전송 오류를 감지하는 기능을 제공해 상위 계층인 데이터 링크 계층에서 오류를 인지할 수 있도록 해주지만, 그렇지 않은 경우는 데이터 링크 계층 스스로 별도의 기능을 수행하여 오류를 인지해야 합니다.

대표적인 물리적 오류로는 데이터가 도착하지 못하는 **데이터 분실**과 내용이 깨져서 도착하는 **데이터 변형**이 있습니다. 일반적으로 컴퓨터 네트워크에서 오류 복구는 송신자가 원래의 데이터를 **재전송**하는 방식으로 처리합니다.

> ❓ **재전송 기법이란?**
>
> 재전송 기법은 송신 호스트가 전송한 **프레임**에 오류가 발생하면 송신 호스트가 전송 데이터를 재전송하여 오류를 복구하는 것입니다.

> ❓ **프레임이란?**
>
> 데이터 링크 계층에서는 **데이터를 작은 단위로 분할**하여 프레임이라는 형태로 전송합니다. 이렇게 데이터를 프레임으로 나누는 이유는 전송 중 오류가 발생했을 때 오류가 발생한 프레임만 재전송하며 전송 시간과 데이터 양을 줄이기 위해서입니다.

3 계층 - 네트워크 계층

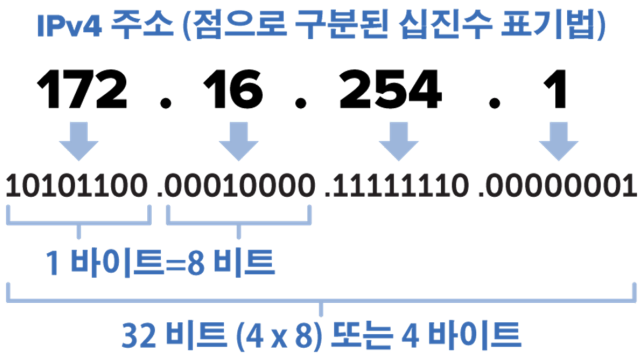
네트워크 계층은 OSI 7 계층 중 3번째에 해당하는 계층으로, **패킷의 라우팅**을 담당합니다. 참고로 패킷은 **네트워크 계층에서 데이터를 작은 단위로 나눈 것**을 말합니다. 패킷은 최종 목적지까지 전달되는 과정에서 라우터들을 거치는데, 이때 라우터가 패킷의 목적지까지 경로를 찾아주는 역할을 합니다.

또한, 패킷을 전송할 때는 인터넷상에서 데이터를 주고받기 위한 통신 규약인 **IP 프로토콜(Internet Protocol)**을 따르는데 IP 프로토콜은 **IP 주소**를 사용하여 출발지와 목적지의 주소를 부여합니다.

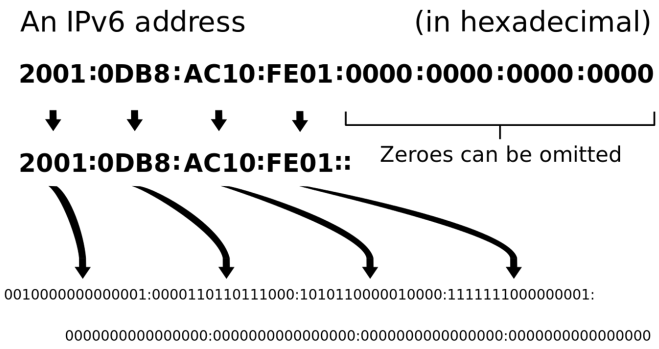
> ❓ **IP 주소란?**
>
> IP 주소는 데이터를 주고받기 위해 사용되는 주소로 IP 주소를 사용하여 각 기기를 식별할 수 있습니다. IP 주소에는 **IPv4**와 **IPv6** 주소 체계가 있는데 IPv4만으로 표현하기에

는 주소가 부족해지는 문제가 발생하여 IPv6 주소 체계가 등장하게 되었습니다.

IPv4는 사진과 같이 32비트로 이루어져 있습니다. 그래서, 총 2^32 = 4,294,967,296 (약 43억 개)의 고유한 주소를 가질 수 있습니다.



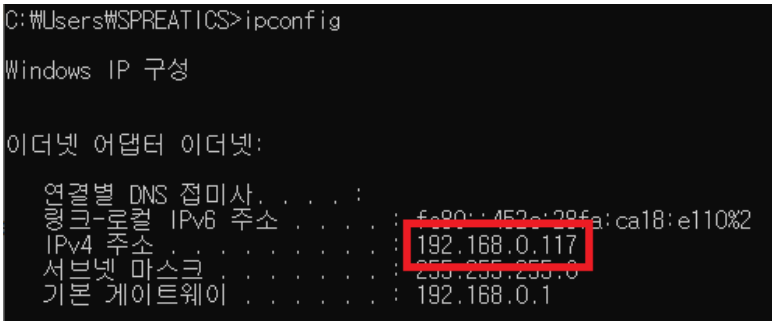
IPv6는 아래와 같이 128비트로 이루어져 있으며, 총 2^128개의 고유한 주소를 가질 수 있습니다. IPv4와 다른 점은 IPv6는 점(.) 대신 콜론(:)을 이용해서 표현하며, 마지막 64비트는 현재는 아직 사용하고 있지 않아서 생략이 가능합니다.



👩‍💻 내 IP를 한번 확인해볼까요!?

Window : 명령 프롬프트(CMD)에서 `ipconfig` 입력

Mac : 터미널에서 `ifconfig` 입력



하지만 IP 주소는 저희가 외우기엔 너무 복잡합니다. 그래서 웹 브라우저를 통해 특정 사이트를 진입할 때, 저희는 **도메인**이라는 것을 IP 주소 대신 사용하여 한눈에 파악하기 힘든 IP 주소를 보다 분명하게 나타냅니다.

ex) google.com, naver.com

IP 주소 대신 도메인을 사용하기 위해서는 도메인이 어떤 IP 주소에 매핑이 되어 있는지 알아야 합니다. 그래서 IP 주소와 특정 도메인이 같다는 것을 지정해두고, 인터넷 사용자들이 도메인 주소를 검색했을 때 IP 주소로 연결되도록 해주는 것을 **DNS (Domain Name System)**라고 합니다.

4 계층 - 전송 계층

전송 계층은 OSI 7 계층 모델에서 가장 중요한 계층으로 **송신 호스트와 수신 호스트를 직접 연결하는 단대단(end-to-end) 통신** 기능을 제공합니다.

전송 계층은 송수신자 간의 신뢰성 있고 효율적인 데이터를 전송하며, **오류 검출** 및 **복구**와 **흐름 제어**, **중복 검사** 등을 수행합니다. 이를 통해 전송 과정에서 데이터의 손실이나 오류를 최소화하고, 신뢰성 있는 데이터 전송을 보장할 수 있습니다.

프로토콜 종류

전송 계층의 기능을 제공하는 프로토콜에는 대표적으로 **TCP** 와 **UDP**가 있습니다.

[TCP 프로토콜]

TCP는 Transmission Control Protocol의 약자로, 연결 지향형 프로토콜로 신뢰성 있는 데이터 전송을 보장합니다. 데이터를 보내기 전에 먼저 연결을 설정하고, 데이터를 전송한 후에는 연결을 해제합니다. 이 연결 설정과 해제를 통해 패킷의 분실, 중복, 순서 변경 등의 문제를 예방할 수 있습니다. 따라서, 파일 전송, 이메일 전송, 웹 페이지 요청 등 신뢰성이 중요한 데이터 전송에 주로 사용됩니다.

[UDP 프로토콜]

UDP는 User Datagram Protocol의 약자로, 비연결형 프로토콜로 연결 설정과 해제 과정이 없습니다. 따라서, TCP보다 더 빠른 전송 속도를 가지고 있습니다. 그러나, 데이터 전송에 대한 신뢰성을 보장하지 않습니다. 즉, 패킷이 분실되거나 중복되어도 재전송을 하지 않기 때문에 데이터의 일부분이 유실되어도 전체 데이터를 재전송하는 방법을 제공하지 않습니다. 그렇기 때문에 DNS, 스트리밍, 실시간 게임 등 데이터 손실이 큰 영향을 주지 않는 데이터 전송에 사용됩니다.

쉽게 설명하면, **비연결형 통신**은 편지를 보내는 것과 비슷합니다. 우리는 편지를 보낼 때, 먼저 수신자에게 연락하여 “지금부터 편지를 보낼 거야”라고 말하지 않습니다. 단순히 편지를 써서 우편함에 넣으면, 그 편지는 어디든지 보내질 수 있습니다. 이와 마찬가지로, 비연결형 통신에서는 데이터 패킷을 보낼 때, 먼저 연결을 설정하는 과정 없이 단순히 데이터 패킷을 보내고 받는 것입니다.

반면에, **연결형 통신**은 통화를 하는 것과 비슷합니다. 통화를 하기 위해선 먼저 연결이 되어야 하는 것처럼 연결형 통신에서는 먼저 연결을 설정하고, 그 후에 데이터를 주고받습니다.

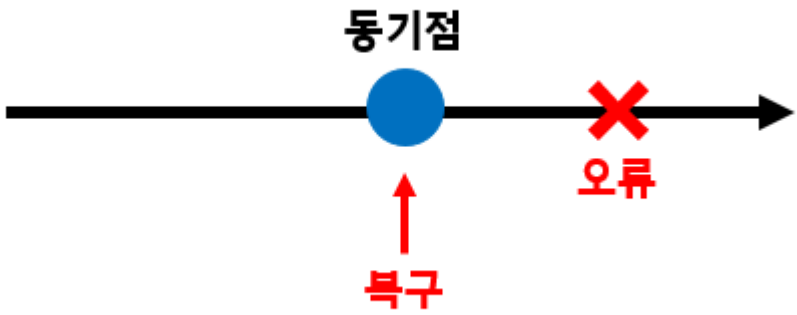
5 계층 - 세션 계층

세션 계층은 전송 계층에서 제공하는 연결 지향성과 신뢰성을 확장하여, 응용 프로세스 간의 논리적인 연결을 만들고 유지합니다. 이를 통해 데이터의 흐름이 제대로 이루어지고, 오류가 발생하면 재전송이 가능하도록 합니다.

예를 들어 인터넷에서 파일을 다운로드하는 과정에서 TCP 연결이 끊기면 파일 전송이 중단됩니다. 이때 세션 기능이 없으면 서버와 다시 연결하여 처음부터 파일을 다운로드해야 하지만, 세션 기능은 전송 계층의 연결이 끊겼을 때, 이를 복구하여 이전에 다운로드한 내용 이후부터 연속해서 전송받을 수 있는 기능을 제공합니다.

세션 계층에서 제공하는 가장 중요한 기능은 **동기(Synchronization)** 문제를 처리하는 것입니다. 동기 기능은 통신 양단에서 서로 동의하는 논리적인 공통 처리 지점, 즉 동기점을 지정하기 위해 사용합니다. 동기점을 설정하는 이유는 메시지 전송 과정에서 발생하는 오류를 복구하기 위해서인데 **동기점이 설정된 시점 이전까지는 통신 양단이 서로 완벽하게 처리했**

음을 합의했다는 의미입니다. 따라서 오류가 발생했을 때 동기점 이전 과정은 복구할 필요가 없습니다.



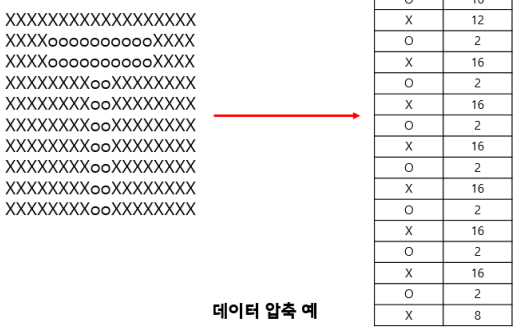
6 계층 - 표현 계층

표현 계층은 **데이터를 표현하고 변환**하는데 사용되는 계층입니다. 이 계층은 데이터의 형식, 암호화 및 인코딩, 데이터 압축, 그리고 파일 형식 변환 등을 다룹니다. 이를 통해 송신 측에서 생성된 데이터를 수신 측에서 이해할 수 있는 형식으로 변환하는 역할을 합니다.

예를 들어, 표현 계층은 송신 측과 수신 측의 컴퓨터가 서로 다른 운영 체제를 사용하고 있을 경우, 데이터를 운영 체제 간에 호환 가능한 형식으로 변환하며, 데이터가 전송될 때 암호화되어 전송되어야 하는 경우, 이를 담당하기도 합니다. 또한, 이미지나 비디오와 같은 미디어 파일을 전송할 때는 데이터를 압축하거나 인코딩하여 전송 효율을 높이기도 합니다.

0	00	NUL	25	19	EM	51	33	3	77	4D	M	103	67	g
1	01	SOH	26	1A	SUB	52	34	4	78	4E	N	104	68	h
2	02	STX	27	1B	ESC	53	35	5	79	4F	O	105	69	i
3	03	ETX	28	1C	FS	54	36	6	80	50	P	106	6A	j
4	04	EOT	29	1D	GS	55	37	7	81	51	Q	107	6B	k
5	05	ENQ	30	1E	RS	56	38	8	82	52	R	108	6C	l
6	06	ACK	31	1F	US	57	39	9	83	53	S	109	6D	m
7	07	BEL	32	20	space	58	3A	:	84	54	T	110	6E	n
8	08	BS	33	21	!	59	3B	;	85	55	U	111	6F	o
9	09	HT	34	22	"	60	3C	<	86	56	V	112	70	p
10	0A	LF	35	23	#	61	3D	=	87	57	W	113	71	q
11	0B	VT	36	24	$	62	3E	>	88	58	X	114	72	r
12	0C	FF	37	25	%	63	3F	?	89	59	Y	115	73	s
13	0D	CR	38	26	&	64	40	@	90	5A	Z	116	74	t
14	0E	SO	39	27	'	65	41	A	91	5B	[	117	75	u
15	0F	SI	40	28	(	66	42	B	92	5C	\	118	76	v
16	10	DLE	41	29	)	67	43	C	93	5D	]	119	77	w
17	11	DC1	42	2A	*	68	44	D	94	5E	^	120	78	x
18	12	DC2	43	2B	+	69	45	E	95	5F	_	121	79	y
19	13	DC3	44	2C	,	70	46	F	96	60	`	122	7A	z
20	14	DC4	45	2D	-	71	47	G	97	61	a	123	7B	{
21	15	NAK	46	2E	.	72	48	H	98	62	b	124	7C	|
22	16	SYN	47	2F	/	73	49	I	99	63	c	125	7D	}
23	17	ETB	48	30	0	74	4A	J	100	64	d	126	7E	~
24	18	CAN	49	31	1	75	4B	K	101	65	e	127	7F	DEL
			50	32	2	76	4C	L	102	66	f			

ASCII Table



7 계층 - 응용 계층

최종 사용자의 애플리케이션과 네트워크 사이의 인터페이스 역할을 합니다. 이 계층은 **사용자가 사용하는 소프트웨어 애플리케이션에 가장 가까운 계층**으로, 사용자가 인터넷이나 다른 네트워크에 연결하여 데이터를 전송할 때 이용됩니다.

응용 계층은 **이메일, 파일 전송, 웹 브라우저 등**과 같은 애플리케이션들과 통신합니다. 이 계층에서는 애플리케이션에 필요한 데이터 형식을 정의하며, 애플리케이션 간에 데이터를 교환할 수 있도록 합니다.

즉, 사용자가 볼 수 있는 유일한 계층으로서 인터페이스 및 네트워크 자원에 접근하는 방법을 제공합니다. ex) Chrome, 이메일, DBMS

따라서, 응용 계층의 가장 큰 특징은 계층에서 발생하는 문제는 사용자가 직접 인지하게 됩니다. 그러므로 응용 계층에서 발생하는 문제를 해결하기 위해서는 사용자가 문제를 파악하고, 해당 애플리케이션의 설정을 조정하거나, 새로운 프로그램을 설치해야 할 수도 있습니다.

프로토콜 종류

응용 계층은 다른 OSI 계층들과는 달리 프로토콜의 종류가 다양합니다. 예를 들어 HTTP, SMTP, FTP, Telnet, SSH, DNS, modbus, SIP, AFP, APPC, MAP 등의 프로토콜들이 있습니다. 이러한 프로토콜들은 각각의 목적에 맞게 데이터를 전송하는 규칙들을 정의하고 있습니다. 따라서 응용 계층에서는 이러한 프로토콜들 중에서 사용할 적절한 프로토콜을 선택하여 데이터를 전송합니다.

[HTTP 프로토콜]

HTTP 는 Hypertext Transfer Protocol의 약자로 인터넷 상에서 데이터를 주고받는 프로토콜 중 하나입니다. HTTP는 웹 브라우저와 웹 서버 간의 통신에서 사용되는 프로토콜로 HTML 문서, 이미지, 동영상 등의 데이터를 전송하기 위해 사용됩니다.

HTTP는 클라이언트-서버 모델을 따릅니다. 클라이언트(웹 브라우저)는 HTTP 요청 메시지를 만들어 웹 서버로 전송하고, 서버는 이에 대한 응답 메시지를 만들어 클라이언트로 보냅니다. HTTP는 기본적으로 TCP를 사용하며, 클라이언트와 서버 간의 연결을 유지합니다.

[SSH 프로토콜]

네트워크 상에서 안전하게 원격으로 컴퓨터를 제어하기 위한 프로토콜입니다. SSH는 Telnet이나 FTP와 같은 원격 접속 프로토콜의 보안 취약점을 보완하고, 암호화를 통해 안전한 데이터 전송을 지원합니다. SSH 또한 기본적으로 TCP를 사용하며, 공개키 암호화 방식을 사용하여 서버와 클라이언트 간의 인증 및 세션 암호화를 수행합니다.

[FTP 프로토콜]

FTP는 File Transfer Protocol의 약자로 인터넷 상에서 파일을 전송하기 위한 프로토콜입니다. FTP를 사용하면 클라이언트가 서버에 파일을 업로드하거나 서버에서 파일을 다운로드할 수 있습니다. 기본적으로 TCP/IP 프로토콜을 사용합니다.

FTP는 다른 프로토콜과 달리, 보안성이나 데이터 암호화를 제공하지 않습니다. 따라서 FTP를 사용하여 파일을 전송할 때는, 중요한 파일을 전송하기 전에 암호화나 보안 기술을 이용하여 파일을 보호해야 합니다.

현재는 보안성과 성능면에서 더 발전된 SFTP(Secure File Transfer Protocol)나 FTPS(FTP over SSL) 등의 프로토콜을 사용하고 있습니다. 이러한 프로토콜은 FTP와 유사한 방식으로 작동하지만, 데이터 전송 과정에서 암호화를 제공하여 더 안전하고 신뢰성 높은 파일 전송을 가능하게 합니다.

[Telnet 프로토콜]

Telnet은 인터넷 상에서 원격 호스트에 로그인하고, 명령어나 데이터를 전송하는 프로토콜입니다. 이를 통해 사용자는 다른 컴퓨터에 있는 프로그램이나 파일을 제어하거나, 다른 컴퓨터와 통신할 수 있습니다.

Telnet 클라이언트는 호스트와의 연결을 위해 TCP/IP 네트워크를 통해 서버에 접속하고, 서버에 로그인하여 사용자 인증을 수행합니다. 이후에 사용자는 키보드로 명령어나 데이터를 입력하고, 이를 서버로 전송합니다. 서버는 입력된 명령어나 데이터를 처리하고, 결과를 클라이언트에게 반환합니다.

Telnet은 암호화나 보안 기술을 제공하지 않으며, 전송되는 모든 데이터가 평문으로 전송됩니다. 따라서 Telnet을 사용하여 로그인 정보나 기밀 정보를 전송하는 것은 보안상 매우 위험합니다.