**CS 스터디 6주차 공부**

수정시간: 2023/11/25

수정자 명: 김도언

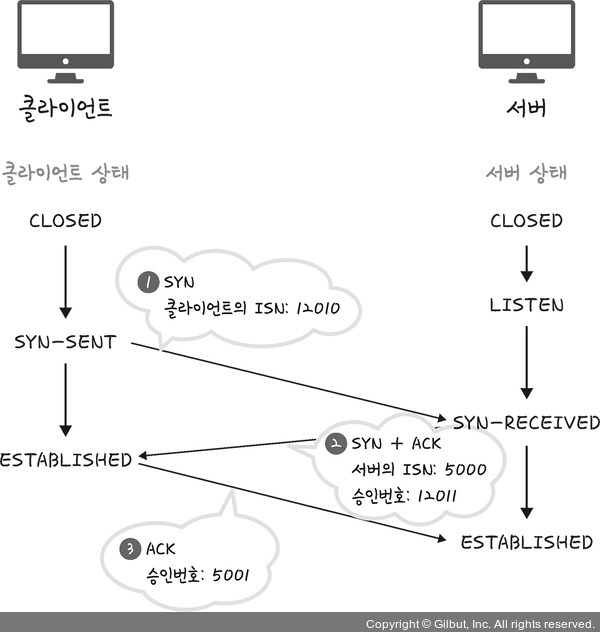
**- 공부 내용**

P. 85~ 105

P. 114~ 132

**TCP 연결 성립 과정**

TCP는 신뢰성을 확보할 때 ‘3-웨이 핸드셰이크(3-way handshake)’라는 작업을 진행합니다.



1) SYN 단계: 클라이언트는 서버에 클라이언트의 ISN을 담아 SYN을 보냄

ISN은 새로운 TCP 연결의 첫 번째 패킷에 할당된 임의의 시퀀스 번호를 말하며(예시로 12010을 들었습니다) 이는 장치마다 다를 수 있습니다.

2) SYN + ACK 단계: 서버는 클라이언트의 SYN을 수신하고 서버의 ISN을 보내며 승인번호로 클라이언트의 ISN + 1을 보냅니다.

3) ACK 단계: 클라이언트는 서버의 ISN + 1한 값인 승인번호를 담아 ACK를 서버에 보냅니다.

이렇게 3-웨이 핸드셰이크 과정 이후 신뢰성이 구축되고 데이터 전송을 시작합니다. 참고로 TCP는 이 과정이 있기 때문에 신뢰성이 있는 계층이라고 하며 UDP는 이 과정이 없기 때문에 신뢰성이 없는 계층이라고 합니다.

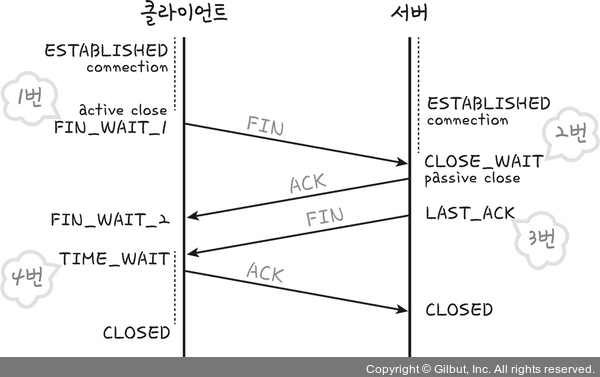
––– SYN : SYNchronization의 약자, 연결 요청 플래그

––– ACK : ACKnowledgement의 약자, 응답 플래그

––– ISN: Initial Sequence Numbers의 약어, 초기 네트워크 연결을 할 때 할당된 32비트 고유 시퀀스 번호이다

**TCP 연결 해제 과정**

TCP가 연결을 해제할 때는 4-웨이 핸드셰이크(4-way handshake) 과정이 발생합니다.



• 1번: 먼저 클라이언트가 연결을 닫으려고 할 때 FIN으로 설정된 세그먼트를 보냅니다. 그리고 클라이언트는 FIN\_WAIT\_1 상태로 들어가고 서버의 응답을 기다립니다.

• 2번: 서버는 클라이언트로 ACK라는 승인 세그먼트를 보냅니다. 그리고 CLOSE\_WAIT 상태에 들어갑니다. 클라이언트가 세그먼트를 받으면 FIN\_WAIT\_2 상태에 들어갑니다.

• 3번: 서버는 ACK를 보내고 일정 시간 이후에 클라이언트에 FIN이라는 세그먼트를 보냅니다.

• 4번: 클라이언트는 TIME\_WAIT 상태가 되고 다시 서버로 ACK를 보내서 서버는 CLOSED 상태가 됩니다. 이후 클라이언트는 어느 정도의 시간을 대기한 후 연결이 닫히고 클라이언트와 서버의 모든 자원의 연결이 해제됩니다.

**TIME\_WAIT사용이유**

첫 번째는 지연 패킷이 발생할 경우를 대비하기 위함입니다. 패킷이 뒤늦게 도달하고 이를 처리하지 못한다면 데이터 무결성 문제가 발생합니다.

두 번째는 두 장치가 연결이 닫혔는지 확인하기 위해서입니다. 만약 LAST\_ACK 상태에서 닫히게 되면 다시 새로운 연결을 하려고 할 때 장치는 줄곧 LAST\_ACK로 되어 있기 때문에 접속 오류가 나타나게 될 것입니다.

이러한 이유로 TIME\_WAIT라는 잠시 기다릴 시간이 필요합니다.

––– TIME\_WAIT : 소켓이 바로 소멸되지 않고 일정 시간 유지되는 상태를 말하며 지연 패킷 등의 문제점을 해결하는 데 쓰인다. CentOS6, 우분투에는 60초로 설정되어 있으며 윈도우는 4분으로 설정되어 있다. 즉, OS마다 조금씩 다를 수 있다.

––– 데이터 무결성(data integrity): 데이터의 정확성과 일관성을 유지하고 보증하는 것

**인터넷 계층**

인터넷(internet) 계층은 장치로부터 받은 네트워크 패킷을 IP 주소로 지정된 목적지로 전송하기 위해 사용되는 계층입니다. IP, ARP, ICMP 등이 있으며 패킷을 수신해야 할 상대의 주소를 지정하여 데이터를 전달합니다. 상대방이 제대로 받았는지에 대해 보장하지 않는 비연결형적인 특징을 가지고 있습니다.

**링크 계층**

링크 계층은 전선, 광섬유, 무선 등으로 실질적으로 데이터를 전달하며 장치 간에 신호를 주고받는 ‘규칙’을 정하는 계층입니다. 참고로 네트워크 접근 계층이라고도 합니다.

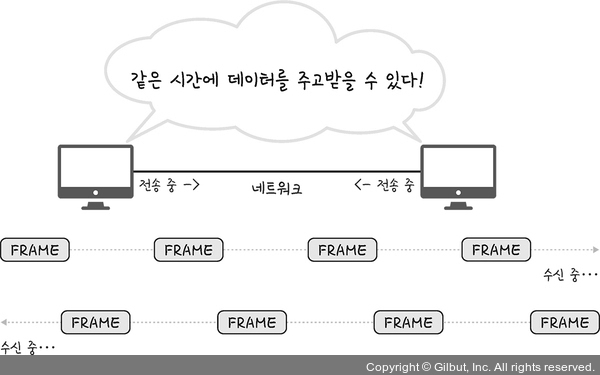
이를 물리 계층과 데이터 링크 계층으로 나누기도 하는데 물리 계층은 무선 LAN과 유선 LAN을 통해 0과 1로 이루어진 데이터를 보내는 계층을 말하며, 데이터 링크 계층은 ‘이더넷 프레임’을 통해 에러 확인, 흐름 제어, 접근 제어를 담당하는 계층을 말합니다.

**유선 LAN(IEEE802.3)**

유선 LAN을 이루는 이더넷은 IEEE802.3이라는 프로토콜을 따르며 전이중화 통신을 씁니다.

**전이중화 통신**

전이중화(full duplex) 통신은 양쪽 장치가 동시에 송수신할 수 있는 방식을 말합니다. 이는 송신로와 수신로로 나눠서 데이터를 주고받으며 현대의 고속 이더넷은 이 방식을 기반으로 통신하고 있습니다.



**CSMA/CD**

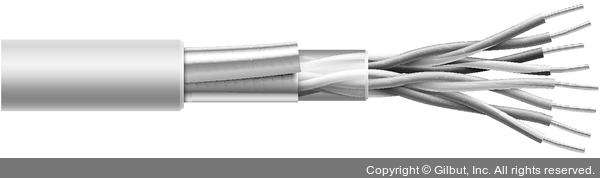
참고로 이전에는 유선 LAN에 ‘반이중화 통신’ 중 하나인 CSMA/CD(Carrier Sense Multiple Access with Collision Detection) 방식을 썼습니다. 이 방식은 데이터를 ‘보낸 이후’ 충돌이 발생한다면 일정 시간 이후 재전송하는 방식을 말합니다. 이는 수신로와 송신로를 각각 둔 것이 아니고 한 경로를 기반으로 데이터를 보내기 때문에 데이터를 보낼 때 충돌에 대해 대비해야 했기 때문입니다.

**유선 LAN을 이루는 케이블**

유선 LAN을 이루는 케이블로는 TP 케이블이라고 하는 트위스트 페어 케이블과 광섬유 케이블이 대표적입니다.

**트위스트 페어 케이블**

트위스트 페어 케이블(twisted pair cable)은 하나의 케이블처럼 보이지만 실제로는 여덟 개의 구리선을 두 개씩 꼬아서 묶은 케이블을 지칭합니다.



앞의 그림처럼 여러 개의 구리선으로 이루어져 있습니다.

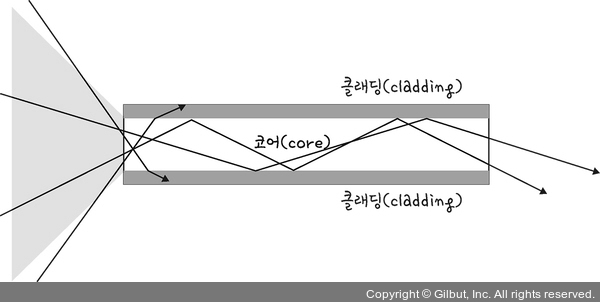
케이블은 구리선을 실드 처리하지 않고 덮은 UTP 케이블과 실드 처리하고 덮은 STP로 나눠집니다. 여기서 우리가 많이 볼 수 있는 케이블은 UTP 케이블로 흔히 LAN 케이블이라고 합니다.



참고로 이 LAN 케이블을 꽂을 수 있는 커넥터를 RJ-45 커넥터라고 합니다.

**광섬유 케이블**

광섬유 케이블은 광섬유로 만든 케이블입니다. 레이저를 이용해서 통신하기 때문에 구리선과는 비교할 수 없을 만큼의 장거리 및 고속 통신이 가능합니다. 보통 100Gbps의 데이터를 전송하며 다음 그림처럼 광섬유 내부와 외부를 다른 밀도를 가지는 유리나 플라스틱 섬유로 제작해서 한 번 들어간 빛이 내부에서 계속적으로 반사하며 전진하여 반대편 끝까지 가는 원리를 이용한 것입니다.



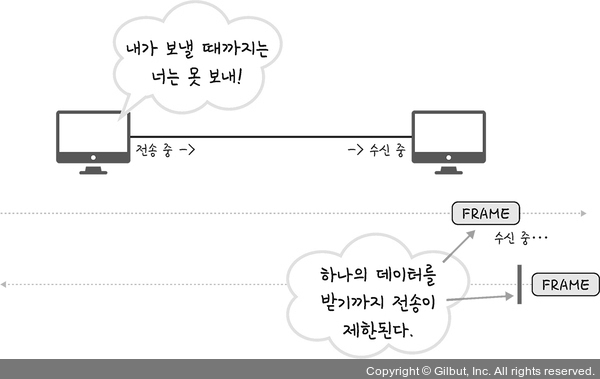
참고로 빛의 굴절률이 높은 부분을 코어(core)라고 하며 낮은 부분을 클래딩(cladding)이라고 합니다.

**무선 LAN(IEEE802.11)**

무선 LAN 장치는 수신과 송신에 같은 채널을 사용하기 때문에 반이중화 통신을 사용합니다.

**반이중화 통신**

반이중화 통신(half duplex)은 양쪽 장치는 서로 통신할 수 있지만, 동시에는 통신할 수 없으며 한 번에 한 방향만 통신할 수 있는 방식을 말합니다.



일반적으로 장치가 신호를 수신하기 시작하면 응답하기 전에 전송이 완료될 때까지 기다려야 합니다. 또한, 둘 이상의 장치가 동시에 전송하면 충돌이 발생하여 메시지가 손실되거나 왜곡될 수 있기 때문에 충돌 방지 시스템이 필요합니다.

**CSMA/CA**

CSMA/CA는 반이중화 통신 중 하나로 장치에서 데이터를 보내기 전에 캐리어 감지 등으로 사전에 가능한 한 충돌을 방지하는 방식을 사용하며 과정은 다음과 같이 이루어집니다.

1. 데이터를 송신하기 전에 무선 매체를 살핍니다.

2. 캐리어 감지: 회선이 비어 있는지를 판단합니다.

3. IFS(Inter FrameSpace): 랜덤 값을 기반으로 정해진 시간만큼 기다리며, 만약 무선 매체가 사용 중이면 점차 그 간격을 늘려가며 기다립니다.

4. 이후에 데이터를 송신합니다.

참고로 이와 반대되는 전이중화 통신은 양방향 통신이 가능하므로 충돌 가능성이 없기 때문에 충돌을 감지하거나 방지하는 메커니즘이 필요하지 않습니다.

**무선 LAN을 이루는 주파수**

무선 LAN(WLAN, Wireless Local Area Network)은 무선 신호 전달 방식을 이용하여 2대 이상의 장치를 연결하는 기술입니다.

비유도 매체인 공기에 주파수를 쏘아 무선 통신망을 구축하는데, 주파수 대역은 2.4GHz 대역 또는 5GHz 대역 중 하나를 써서 구축합니다. 2.4GHz는 장애물에 강한 특성을 가지고 있지만 전자레인지, 무선 등 전파 간섭이 일어나는 경우가 많고 5GHz 대역은 사용할 수 있는 채널 수도 많고 동시에 사용할 수 있기 때문에 상대적으로 깨끗한 전파 환경을 구축할 수 있습니다. 그렇기 때문에 보통은 5GHz 대역을 사용하는 것이 좋습니다.

**와이파이**

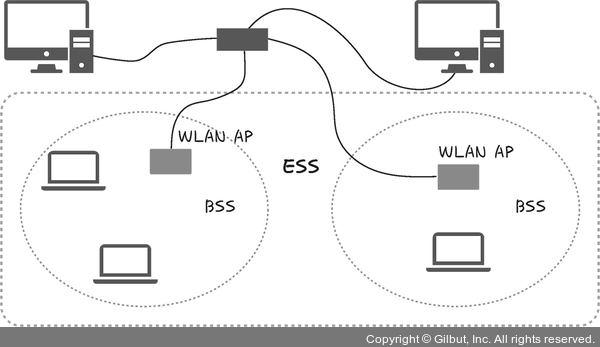
와이파이(wifi)는 전자기기들이 무선 LAN 신호에 연결할 수 있게 하는 기술로, 이를 사용하려면 무선 접속 장치(AP, Access Point)가 있어야 합니다. 흔히 이를 공유기라고 하며, 이를 통해 유선 LAN에 흐르는 신호를 무선 LAN 신호로 바꿔주어 신호가 닿는 범위 내에서 무선 인터넷을 사용할 수 있게 됩니다. 참고로 무선 LAN을 이용한 기술로는 와이파이만 있는 것이 아니고 지그비, 블루투스 등이 있습니다.

**BSS**

BSS(Basic Service Set)는 기본 서비스 집합을 의미하며, 단순 공유기를 통해 네트워크에 접속하는 것이 아닌 동일 BSS 내에 있는 AP들과 장치들이 서로 통신이 가능한 구조를 말합니다. 근거리 무선 통신을 제공하고, 하나의 AP만을 기반으로 구축이 되어 있어 사용자가 한 곳에서 다른 곳으로 자유롭게 이동하며 네트워크에 접속하는 것은 불가능합니다.

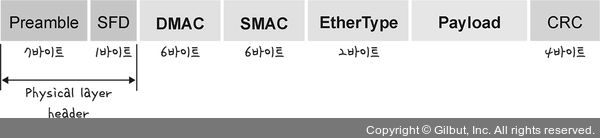
**ESS**

ESS(Extended Service Set)는 하나 이상의 연결된 BSS 그룹입니다. 장거리 무선 통신을 제공하며 BSS보다 더 많은 가용성과 이동성을 지원합니다. 즉, 사용자는 한 장소에서 다른 장소로 이동하며 중단 없이 네트워크에 계속 연결할 수 있습니다.

.

**이더넷 프레임**

참고로 데이터 링크 계층은 이더넷 프레임을 통해 전달받은 데이터의 에러를 검출하고 캡슐화하며 다음과 같은 구조를 가집니다.



• Preamble: 이더넷 프레임이 시작임을 알립니다

• SFD(Start Frame Delimiter): 다음 바이트부터 MAC 주소 필드가 시작됨을 알립니다.

• DMAC, SMAC: 수신, 송신 MAC 주소를 말합니다.

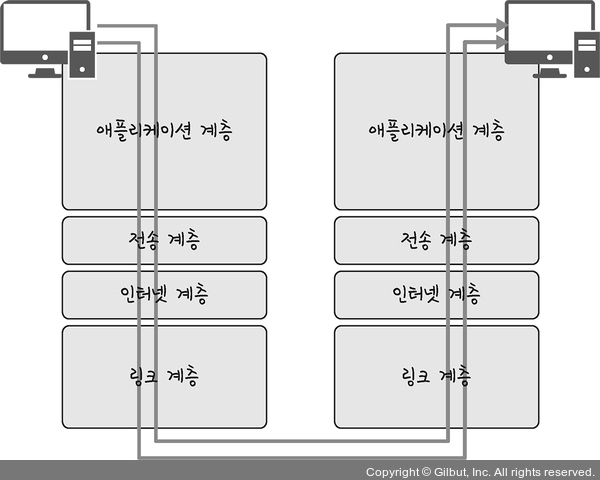
• EtherType: 데이터 계층 위의 계층인 IP 프로토콜을 정의합니다. 예를 들어 IPv4 또는 IPv6가 됩니다.

• Payload: 전달받은 데이터

• CRC: 에러 확인 비트

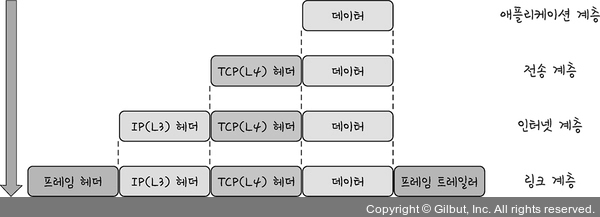
––– MAC 주소 : 컴퓨터나 노트북 등 각 장치에는 네트워크에 연결하기 위한 장치(LAN 카드)가 있는데, 이를 구별하기 위한 식별번호를 말한다. 6바이트(48비트)로 구성된다.

**계층 간 데이터 송수신 과정**

애플리케이션 계층에서 전송 계층으로 필자가 보내는 요청(request) 값들이 캡슐화 과정을 거쳐 전달되고, 다시 링크 계층을 통해 해당 서버와 통신을 하고, 해당 서버의 링크 계층으로부터 애플리케이션까지 비캡슐화 과정을 거쳐 데이터가 전송됩니다.

**캡슐화 과정**

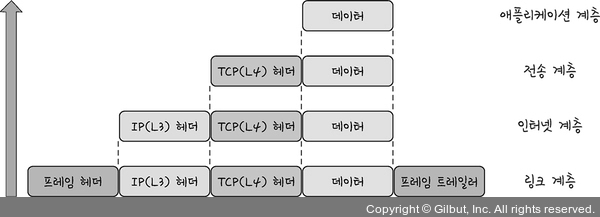
캡슐화 과정은 상위 계층의 헤더와 데이터를 하위 계층의 데이터 부분에 포함시키고 해당 계층의 헤더를 삽입하는 과정을 말합니다.



애플리케이션 계층의 데이터가 전송 계층으로 전달되면서 ‘세그먼트’ 또는 ‘데이터그램’화되며 TCP(L4) 헤더가 붙여지게 됩니다. 그리고 이후 인터넷 계층으로 가면서 IP(L3) 헤더가 붙여지게 되며 ‘패킷’화가 되고, 이후 링크 계층으로 전달되면서 프레임 헤더와 프레임 트레일러가 붙어 ‘프레임’화가 됩니다.

**비캡슐화 과정**

비캡슐화 과정은 하위 계층에서 상위 계층으로 가며 각 계층의 헤더 부분을 제거하는 과정을 말합니다.



이렇게 캡슐화된 데이터를 받게 되면 링크 계층에서부터 타고 올라오면서 프레임화된 데이터는 다시 패킷화를 거쳐 세그먼트, 데이터그램화를 거쳐 메시지화가 되는 비캡슐화 과정이 일어납니다. 그 이후 최종적으로 사용자에게 애플리케이션의 PDU인 메시지로 전달됩니다.

**2.2.2 PDU**

네트워크의 어떠한 계층에서 계층으로 데이터가 전달될 때 한 덩어리의 단위를 PDU (Protocol Data Unit)라고 합니다.

PDU는 제어 관련 정보들이 포함된 ‘헤더’, 데이터를 의미하는 ‘페이로드’로 구성되어 있으며 계층마다 부르는 명칭이 다릅니다.

• 애플리케이션 계층: 메시지

• 전송 계층: 세그먼트(TCP), 데이터그램(UDP)

• 인터넷 계층: 패킷

• 링크 계층: 프레임(데이터 링크 계층), 비트(물리 계층)

예를 들어 애플리케이션 계층은 ‘메시지’를 기반으로 데이터를 전달하는데, HTTP의 헤더가 문자열인 것을 예로 들 수 있습니다.

참고로 PDU 중 아래 계층인 비트로 송수신하는 것이 모든 PDU 중 가장 빠르고 효율성이 높습니다. 하지만 애플리케이션 계층에서는 문자열을 기반으로 송수신을 하는데, 그 이유는 헤더에 authorization 값 등 다른 값들을 넣는 확장이 쉽기 때문입니다.

**SECTION 2.3 네트워크 기기**

**2.3.1 네트워크 기기의 처리 범위**

네트워크 기기는 계층별로 처리 범위를 나눌 수 있습니다. 물리 계층을 처리할 수 있는 기기와 데이터 링크 계층을 처리할 수 있는 기기 등이 있습니다. 그리고 상위 계층을 처리하는 기기는 하위 계층을 처리할 수 있지만 그 반대는 불가합니다. 예를 들어 L7 스위치는 애플리케이션 계층을 처리하는 기기로, 그 밑의 모든 계층의 프로토콜을 처리할 수 있습니다. 하지만 AP는 물리 계층밖에 처리하지 못합니다.

• 애플리케이션 계층: L7 스위치

• 인터넷 계층: 라우터, L3 스위치

• 데이터 링크 계층: 브리지, L2 스위치

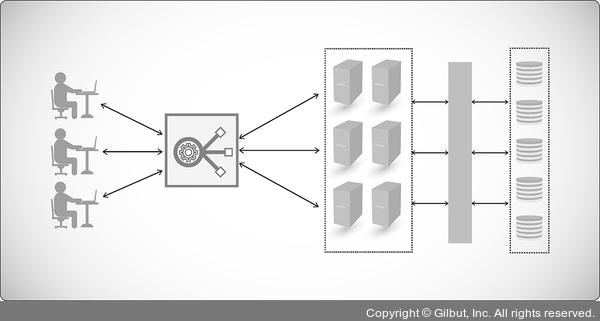
• 물리 계층: NIC, 리피터, AP

**2.3.2 애플리케이션 계층을 처리하는 기기**

애플리케이션 계층을 처리하는 기기로는 L7 스위치가 있습니다.

**L7 스위치**

스위치는 여러 장비를 연결하고 데이터 통신을 중재하며 목적지가 연결된 포트로만 전기 신호를 보내 데이터를 전송하는 통신 네트워크 장비입니다.



L7 스위치는 로드밸런서라고도 하며, 서버의 부하를 분산하는 기기입니다. 클라이언트로부터 오는 요청들을 뒤쪽의 여러 서버로 나누는 역할을 하며 시스템이 처리할 수 있는 트래픽 증가를 목표로 합니다. URL, 서버, 캐시, 쿠키들을 기반으로 트래픽을 분산합니다. 또한, 바이러스, 불필요한 외부 데이터 등을 걸러내는 필터링 기능 또한 가지고 있으며 응용 프로그램 수준의 트래픽 모니터링도 가능합니다.

만약 장애가 발생한 서버가 있다면 이를 트래픽 분산 대상에서 제외해야 하는데, 이는 정기적으로 헬스 체크(health check)를 이용하여 감시하면서 이루어집니다.

**L4 스위치와 L7 스위치 차이**

로드밸런서로는 L7 스위치뿐만 아니라 L4 스위치도 있습니다. L4 스위치는 인터넷 계층을 처리하는 기기로 스트리밍 관련 서비스에서는 사용할 수 없으며 메시지를 기반으로 인식하지 못하고 IP와 포트를 기반으로(특히 포트를 기반으로) 트래픽을 분산합니다. 반면 L7 로드밸런서는 IP, 포트 외에도 URL, HTTP 헤더, 쿠키 등을 기반으로 트래픽을 분산합니다.

참고로 클라우드 서비스(AWS 등)에서 L7 스위치를 이용한 로드밸런싱은 ALB (Application Load Balancer) 컴포넌트로 하며, L4 스위치를 이용한 로드밸런싱은 NLB (Network Load Balancer) 컴포넌트로 합니다.

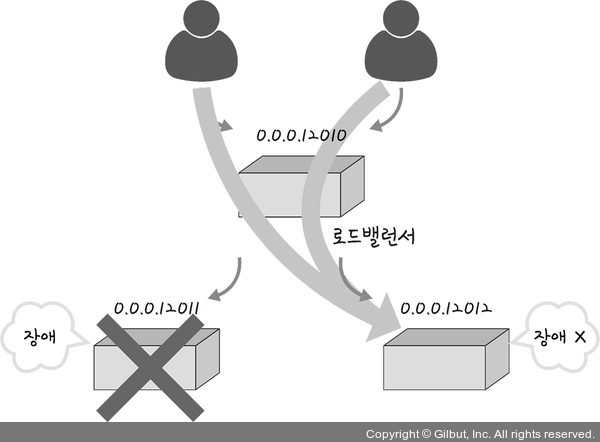
**헬스 체크**

L4 스위치 또는 L7 스위치 모두 헬스 체크를 통해 정상적인 서버 또는 비정상적인 서버를 판별하는데, 헬스 체크는 전송 주기와 재전송 횟수 등을 설정한 이후 반복적으로 서버에 요청을 보내는 것을 말합니다.

물론 이때 서버에 부하가 되지 않을 만큼 요청 횟수가 적절해야 합니다. TCP, HTTP 등 다양한 방법으로 요청을 보내며 이 요청이 정상적으로 이루어졌다면 정상적인 서버로 판별합니다. 예를 들어 TCP 요청을 보냈는데 3-웨이 핸드셰이크가 정상적으로 일어나지 않았다면 정상이 아닌 것이죠.

**로드밸런서를 이용한 서버 이중화**

로드밸런서는 대표적인 기능으로 서버 이중화를 들 수 있습니다. 서비스를 안정적으로 운용하기 위해서는 2대 이상의 서버는 필수적입니다. 에러가 발생하여 서버 1대가 종료되더라도 서비스는 안정적으로 운용되어야 하기 때문이죠. 로드밸런서는 2대 이상의 서버를 기반으로 가상 IP를 제공하고 이를 기반으로 안정적인 서비스를 제공합니다.



앞의 그림처럼 로드밸런서가 제공한 0.0.0.12010이란 가상 IP에 사용자들이 접근하고 뒷단에 사용 가능한 서버인 0.0.0.12011과 0.0.0.12012를 기반으로 서빙합니다. 이렇게 하면 0.0.0.12011이란 서버에 장애가 발생해도 그것과는 무방하게 0.0.0.12012 서버를 기반으로 안정적인 서비스를 운용할 수 있는 것이죠.

**2.3.3 인터넷 계층을 처리하는 기기**

인터넷 계층을 처리하는 기기로는 라우터, L3 스위치가 있습니다.

**라우터**

라우터(router)는 여러 개의 네트워크를 연결, 분할, 구분시켜주는 역할을 하며 “다른 네트워크에 존재하는 장치끼리 서로 데이터를 주고받을 때 패킷 소모를 최소화하고 경로를 최적화하여 최소 경로로 패킷을 포워딩”하는 라우팅을 하는 장비입니다.



**L3 스위치**

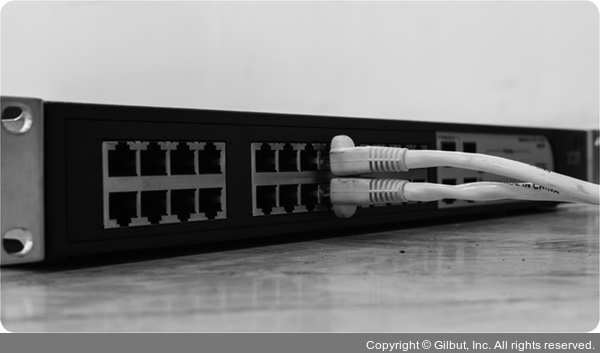
L3 스위치란 L2 스위치의 기능과 라우팅 기능을 갖춘 장비를 말합니다. L3 스위치를 라우터라고 해도 무방합니다. 라우터는 소프트웨어 기반의 라우팅과 하드웨어 기반의 라우팅을 하는 것으로 나눠지고 하드웨어 기반의 라우팅을 담당하는 장치를 L3 스위치라고 합니다.

|  |  |  |
| --- | --- | --- |
| 구분 | L2 스위치 | L3 스위치 |
| 참조 테이블 | MAC 주소 테이블 | 라우팅 테이블 |
| 참조 PDU | 이더넷 프레임 | IP 패킷 |
| 참조 주소 | MAC 주소 | IP 주소 |

**2.3.4 데이터 링크 계층을 처리하는 기기**

데이터 링크 계층을 처리하는 기기로는 L2 스위치와 브리지가 있습니다.

**L2 스위치**

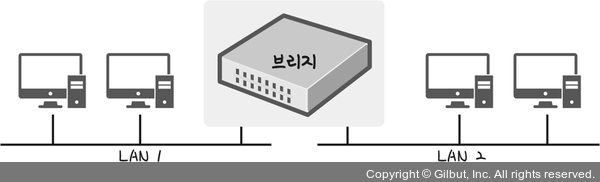


L2 스위치는 장치들의 MAC 주소를 MAC 주소 테이블을 통해 관리하며, 연결된 장치로부터 패킷이 왔을 때 패킷 전송을 담당합니다.

IP 주소를 이해하지 못해 IP 주소를 기반으로 라우팅은 불가능하며 단순히 패킷의 MAC 주소를 읽어 스위칭하는 역할을 합니다. 목적지가 MAC 주소 테이블에 없다면 전체 포트에 전달하고 MAC 주소 테이블의 주소는 일정 시간 이후 삭제하는 기능도 있습니다.

**브리지**

브리지(bridge)는 두 개의 근거리 통신망(LAN)을 상호 접속할 수 있도록 하는 통신망 연결 장치로, 포트와 포트 사이의 다리 역할을 하며 장치에서 받아온 MAC 주소를 MAC 주소 테이블로 관리합니다.



브리지는 통신망 범위를 확장하고 서로 다른 LAN 등으로 이루어진 ‘하나의’ 통신망을 구축할 때 쓰입니다.

**2.3.5 물리 계층을 처리하는 기기**

물리 계층을 처리하는 기기는 NIC, 리피터, AP가 있습니다.

**NIC**

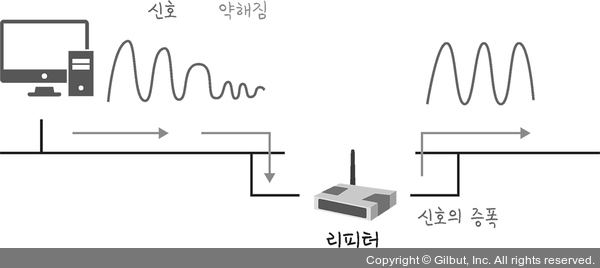
LAN 카드라고 하는 네트워크 인터페이스 카드(NIC, Network Interface Card)는 2대 이상의 컴퓨터 네트워크를 구성하는 데 사용하며, 네트워크와 빠른 속도로 데이터를 송수신할 수 있도록 컴퓨터 내에 설치하는 확장 카드입니다

.

각 LAN 카드에는 주민등록번호처럼 각각을 구분하기 위한 고유의 식별번호인 MAC 주소가 있습니다.

**리피터**

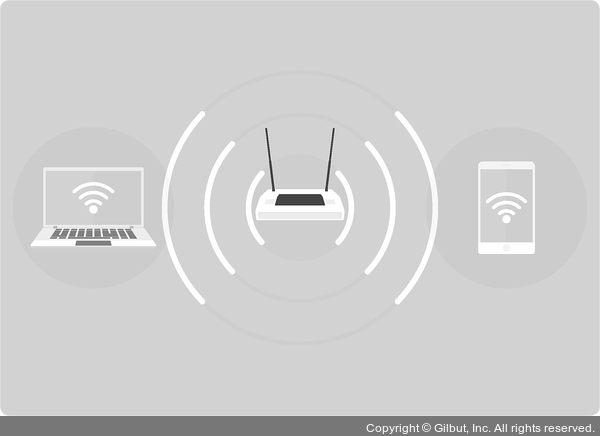
리피터(repeater)는 들어오는 약해진 신호 정도를 증폭하여 다른 쪽으로 전달하는 장치를 말합니다. 이를 통해 패킷이 더 멀리 갈 수 있습니다. 하지만 이는 광케이블이 보급됨에 따라 현재는 잘 쓰이지 않는 장치입니다.



**AP**

AP(Access Point)는 패킷을 복사하는 기기입니다.

AP에 유선 LAN을 연결한 후 다른 장치에서 무선 LAN 기술(와이파이 등)을 사용하여 무선 네트워크 연결을 할 수 있습니다.



**/////P.85~P104 정리**

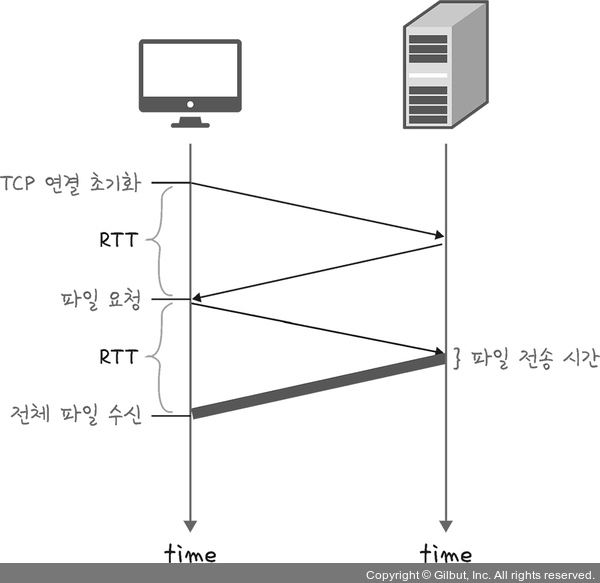
**///P114~P132 정리**

**SECTION 2.5 HTTP**

기본적으로 HTTP는 앞서 설명한 전송 계층 위에 있는 애플리케이션 계층으로서 웹 서비스 통신에 사용됩니다. HTTP/1.0부터 시작해서 발전을 거듭하여 지금은 HTTP/3이며 HTTP/1.0부터 HTTP/3까지 차근 차근 알아보겠습니다.

**2.5.1 HTTP/1.0**

HTTP/1.0은 기본적으로 한 연결당 하나의 요청을 처리하도록 설계되었습니다. 이는 RTT 증가를 불러오게 되었습니다.



서버로부터 파일을 가져올 때마다 TCP의 3-웨이 핸드셰이크를 계속해서 열어야 하기 때문에 RTT가 증가하는 단점이 있었습니다.

––– RTT : 패킷이 목적지에 도달하고 나서 다시 출발지로 돌아오기까지 걸리는 시간이며 패킷 왕복 시간

**RTT의 증가를 해결하기 위한 방법**

매번 연결할 때마다 RTT가 증가하니 서버에 부담이 많이 가고 사용자 응답 시간이 길어졌습니다. 이를 해결하기 위해 이미지 스플리팅, 코드 압축, 이미지 Base64 인코딩을 사용하곤 했습니다.

**이미지 스플리팅**

많은 이미지를 다운로드받게 되면 과부하가 걸리기 때문에 많은 이미지가 합쳐 있는 하나의 이미지를 다운로드받고, 이를 기반으로 background-image의 position을 이용하여 이미지를 표기하는 방법입니다.

**코드 압축**

코드 압축은 코드를 압축해서 개행 문자, 빈칸을 없애서 코드의 크기를 최소화하는 방법입니다.

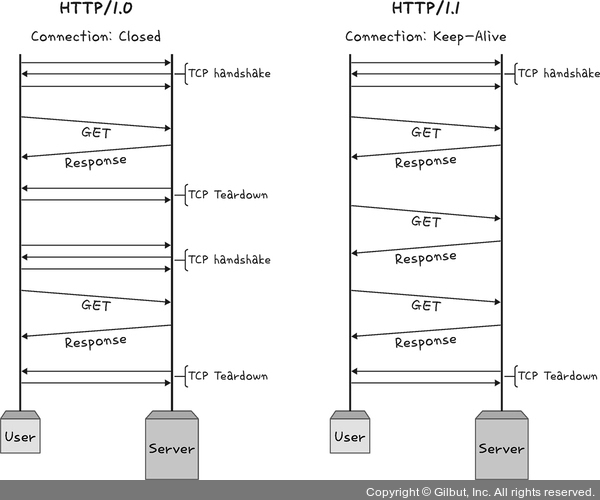
**이미지 Base64 인코딩**

이미지 파일을 64진법으로 이루어진 문자열로 인코딩하는 방법입니다. 이 방법을 사용하면 서버와의 연결을 열고 이미지에 대해 서버에 HTTP 요청을 할 필요가 없다는 장점이 있습니다. 하지만 Base64 문자열로 변환할 경우 37% 정도 크기가 더 커지는 단점이 있습니다.

**––– 인코딩 :** 정보의 형태나 형식을 표준화, 보안, 처리 속도 향상, 저장 공간 절약 등을 위해 다른 형태나 형식으로 변환하는 처리 방식

**2.5.2 HTTP/1.1**

HTTP/1.0에서 발전한 것이 바로 HTTP/1.1입니다. 매번 TCP 연결을 하는 것이 아니라 한 번 TCP 초기화를 한 이후에 keep-alive라는 옵션으로 여러 개의 파일을 송수신할 수 있게 바뀌었습니다. 참고로 HTTP/1.0에서도 keep-alive가 있었지만 표준화가 되어 있지 않았고 HTTP/1.1부터 표준화가 되어 기본 옵션으로 설정되었습니다.



다음 그림처럼 한 번 TCP 3-웨이 핸드셰이크가 발생하면 그다음부터 발생하지 않는 것을 볼 수 있습니다. 하지만 문서 안에 포함된 다수의 리소스(이미지, css 파일, script 파일)를 처리하려면 요청할 리소스 개수에 비례해서 대기 시간이 길어지는 단점이 있습니다.