

링크계층 1,2,3



날짜

@2022년 12월 6일

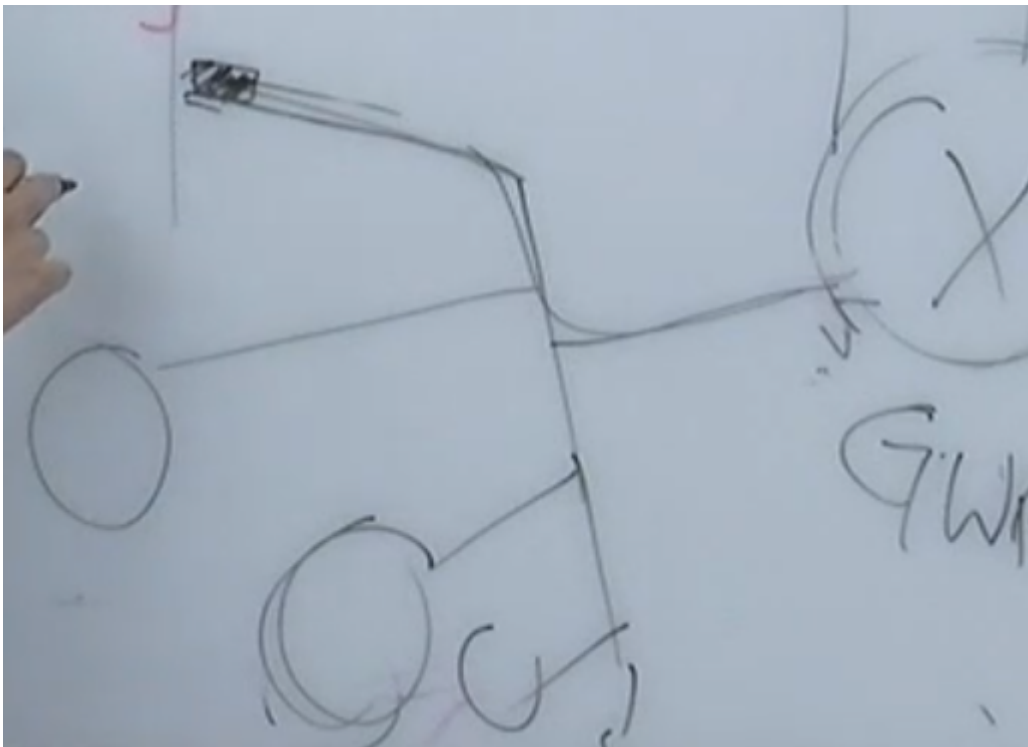
Link layer

실제로는 공유하는 채널이 존재하고 그 채널을 통해 패킷을 전달함.

모든 방향으로 신호가 퍼짐. 다른 연결된 컴퓨터에도 패킷이 전송이 됨.

= broadcast(전송) medium (매체)

그러니까 다른 컴퓨터가 보낸 신호와 내가 보낸 신호가 충돌이 일어날수 있음



Link layer에서는 신호들이 서로 충돌하지 않게 방지하는 역할!

Medium Access Control(MAC protocol)

매체를 컨트롤해서 충돌 문제를 해결하는 방안

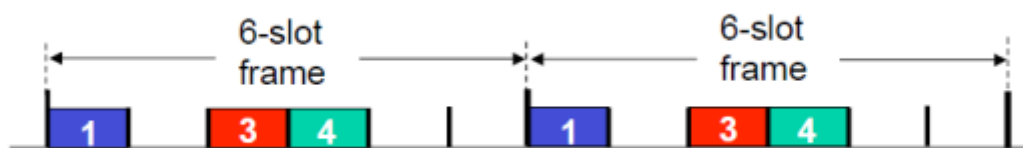
ex) Wifi

이상적인 MAC 조건

1. 한 노드가 전송을 원할 때 전체의 BW 사용
2. M개의 노드가 전송을 원할때 전체/M의 BW 사용
3. 분산처리 가능
4. 단순

<이를 처리위한 방식>

1. Chanel Partitoning



시간을 분할하여 특정 유저는 지정된 시간에만 전송되게 하는 역할

문제점 : 자원이 낭비됨 / 그림에서만 봐도 1과 3 사이에 빈공간이 있음

이를 보완하기 위해 FDMA를 사용

지정된 주파수를 사용함.

Random access protocol

위와 같은 방식의 문제점을 보완하기 위해 내가 보내고자 할 때 패킷을 전송하는 방식, 하지만 동시에 보낼경우 충돌이 발생함

충돌 처리 방법 : ALOHA, CSMA, CSMA/CD, CSMA/CA

CSMA(carrier sense multiple access)

주 특성 : listen before transmit = 이전에 패킷 전송 중인지 체크하고 보냄

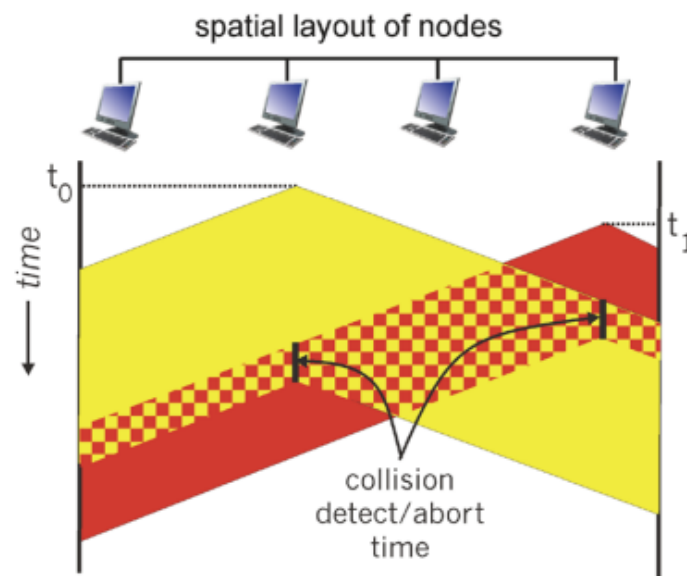
충돌 나는 경우 : 동시에 패킷을 전송하면 충돌이 일어나는 문제점이 발생

조금 더 자세하게 설명하면 propagation delay 때문에 없다고 판단하고 다른 매체가 패킷을 전송하여 충돌이 일어남

하지만 delay 0으로 못함 → delay는 빛의 속도로 존재하기 때문에... → 결론 : 충돌이 날 수 밖에 없다.

해결방안 : 전송을 멈추고 다시 재전송함 = (CSMA/CD)

CSMA/CD (collision detection)



문제 : CSMA 의 특징 및 문제점을 설명하고 해결책이 무엇인지 설명하시오

답 : listen before transmit 특성으로 이전 패킷이 전송 중인지 체크함 하지만 propagation delay로 동시에 패킷을 전송하는 경우가 생기고 충돌이 발생

따라서 CSMA/CD로 충돌 발생 시 데이터 전송을 멈추고 알고리즘을 통해 전송 순서를 정함 (행동방침)

알고리즘 : Binary exponential backoff

- $\{0, 1, 2, \dots, 2^m - 1\}$ 범위 내에서 랜덤으로 숫자를 선택하여 기다림.(작은범위)
- 충돌이 추가로 날 경우 숫자 범위를 증가시켜 waiting시간을 더 늘림
- 즉 충돌이 발생할 수록 점점 오래 기다림(delay)이 증가함.

왜 이런 방식을 채택했을까?

충돌 발생 : 충돌 발생은 눈치챈으나 몇 명이랑 충돌 한지 알 수 없음. 충돌 날 수록 점점 대기 하는 인원이 많다고 생각하고 선택의 범위를 늘려서 충돌 나지 않게 하기 위한 방안

하지만 충돌이 일어났지만 충돌을 발견할 수 없는 경우도 생김

이를 해결하기 위해 Minimum Frame Size(64byte)를 강제로 선정함

TCP 재전송 vs Link Layer MAC Protocol 재전송

TCP

source - server 와의 관계 ack 가 오지 않았을 때 재전송 (상위 레벨)

Link

collision detect 상태일 때만 재전송 (frame 단위)

MAC addresses and ARP

❖ 32-bit IP address:

- *network-layer* address for interface
- used for layer 3 (network layer) forwarding

❖ MAC (or LAN or physical or Ethernet) address:

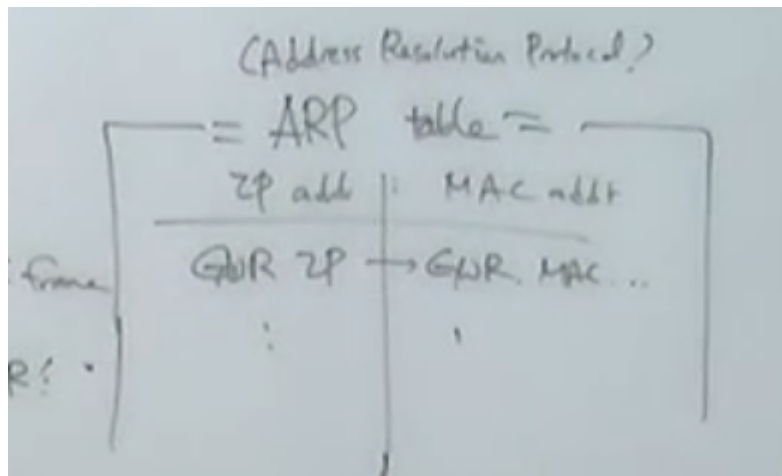
- function: *used "locally" to get frame from one interface to another physically-connected interface (same network, in IP-addressing sense)*
- 48 bit MAC address (for most LANs) burned in NIC ROM, also sometimes software settable
- e.g.: 1A-2F-BB-76-09-AD

hexadecimal (base 16) notation
(each "number" represents 4 bits)

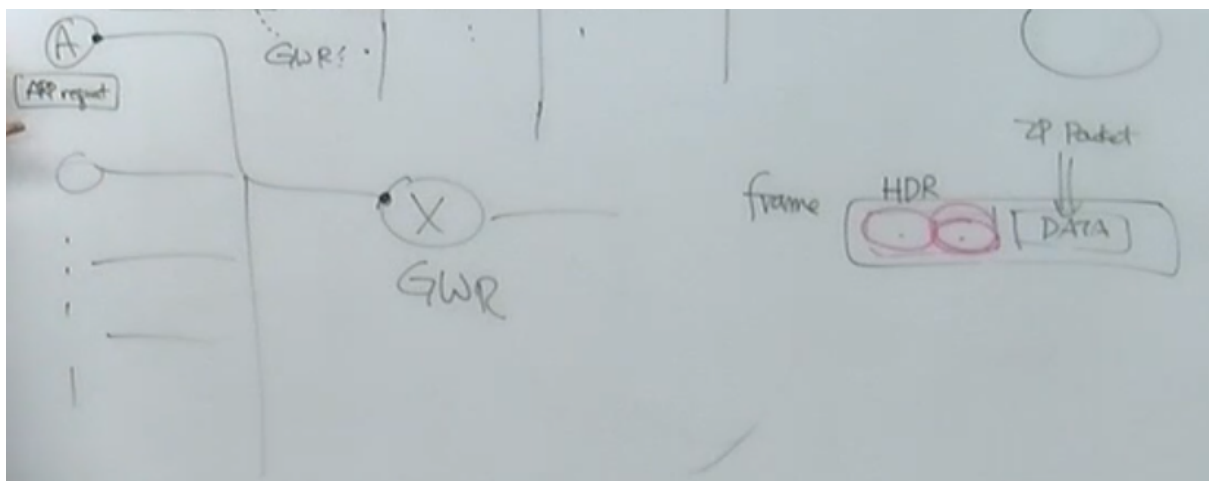
MAC address : 고유한 주소 (변경불가능 - 주민번호와 같은 역할)

MAC address 변경 = 실제 주소를 변경하는게 아니라 Source address만 변경하는 것

ARP(Address Resolution Protocol)



Mac address를 찾기 위한 용 = ARP table



ARP request 안에 Gateway IP 존재 → gateway는 받아서 ARP table 참고하여 IP에 매칭된 MAC address를 source에 전달

Gateway Router

frame의 Source 에 담겨야하는 정보 : GWR MAC address

frame의 destination 에 담겨야하는 정보 : 도착지의 MAC address

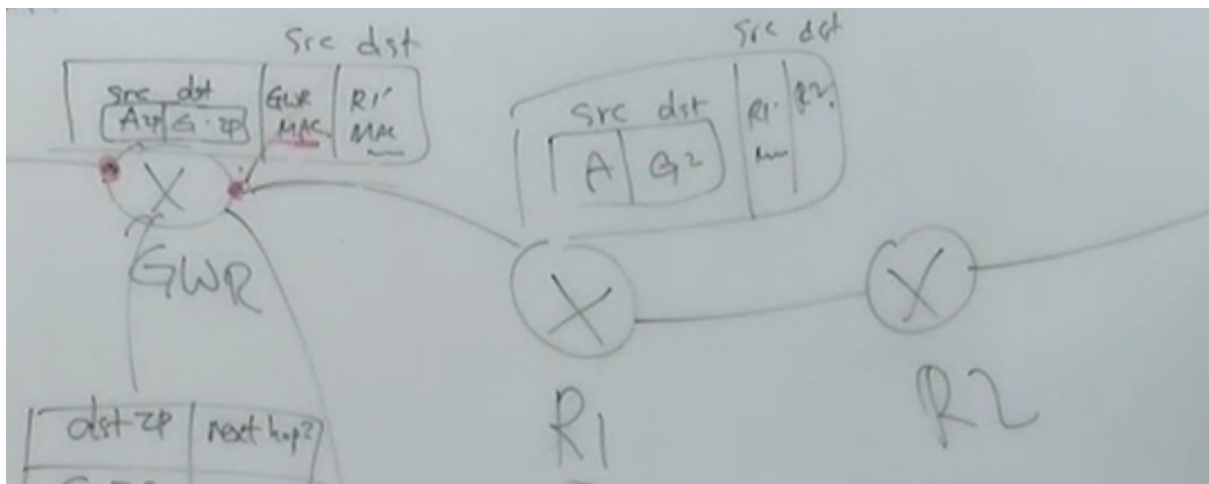
ARP table을 참고하여 가능.

Q) IP패킷을 전송하려고 한다. Link layer에서 frame을 만들어 전송할 때 _____ 을 참고하여 MAC address 를 얻는다.

A) ARP table

Q) GWR frame에서 R1의 MAC Address와 R1 source 에서 MAC Address는 동일하다 (O/X)

다름) Router내의 인터페이스가 다르기 때문에 다른 MAC Address를 가진다.



Switch(Device)

포트에다가 인터넷 선 꼽는다고 생각.



A → I 가는법

S1 → S4 → S2(I없음을 확인) , S3로 동시 확산 → S3에 I 있음을 확인

flowding (퍼지는것) 최초에 한번 사용하고 이후 Self-learning으로 탐색한 정보를 저장
 한번 간 곳은 되돌아올때 flowding없이 바로바로 정보를 받아 목적지 도달가능

S_1		S_2		S_3		S_4	
DST	Port	DST	Port	DST	Port	DST	Port
C	3	C	0	C	0	C	0
I	0			I	3	I	2