

# 링크계층 1

## 1. Link Layer가 하는 일

- 충돌이 일어나지 않게 하기
- 충돌이 일어난다면 해결하기
- 무선이든 유선이든 (단, 내부 디테일은 좀 다름)

## 2. Link Layer

- Network Interface Card 내부에 존재

## 3. Multiple Access Links, Protocols

- broadcast medium
  - 만약 데이터를 전달할 때에 하나의 전용선을 이용한다면 충돌이 일어날 일이 없을 것이다. 하지만, 여러개가 존재하면 같은 매체를 이용하는 과정에서 전달 시에 충돌이 발생한다.
    - 이를 컨트롤 하는 것이 Medium Access Control
    - 즉, MAC Protocol 이라고 한다.

## 4. 이상적인 MAC protocol을 이루기 위한 세가지 시도

### 1. channel partition 방식

- a. TDMA : time division multiple access : 시간을 나눠서 충돌이 없게 설정, 그러나 유저가 적을 경우에 낭비되는 부분이 있다는 한계가 있다.
- b. FDMA : frequency division multiple access

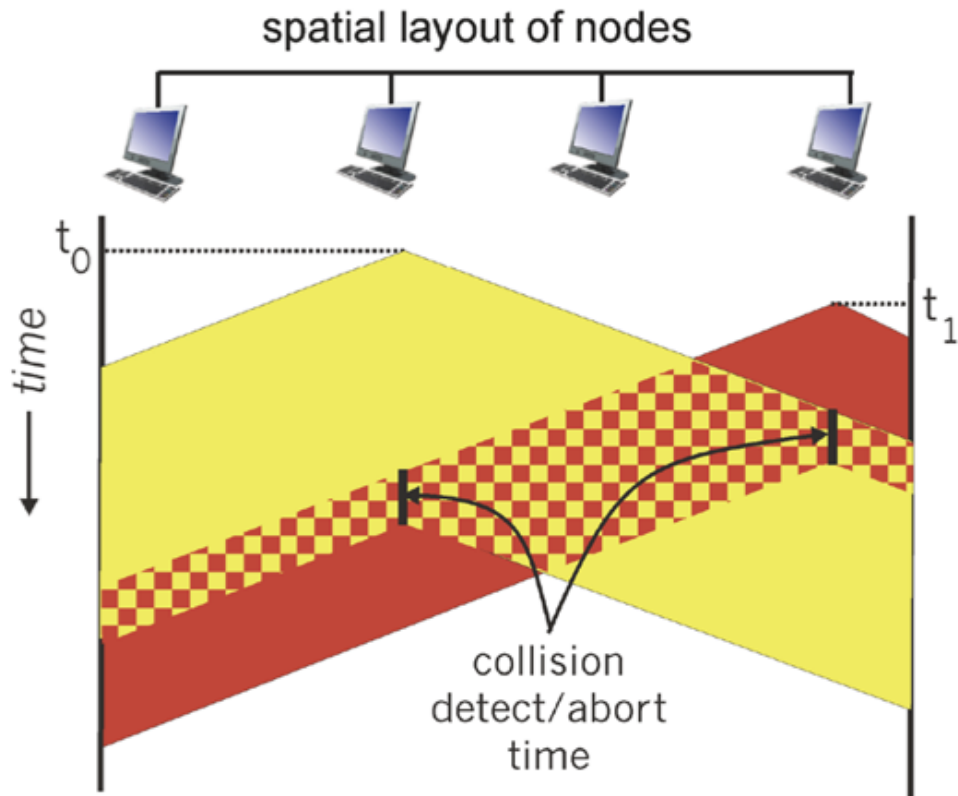
### 2. Random Access Model

- CSMA : carrier sense multiple access

1. broadcast medium 을 이용해서 음성 데이터를 전달한다고 할 때, 한번에 한명씩만 이야기를 해야 오디오가 안 겹쳐.

2. 노드가 여러개

3. 노랑 프레임과 빨강 프레임이 겹치는 부분에서 충돌이 발생



protocol delay 때문에 충돌이 발생 → delay 를 0으로 만들 수 있나? → 없다! 충돌은 나기 마련! → 그러면?

충돌이 나면, 멈추는 게 좋겠지? 충돌을 감지하고 바로 멈추는 것이

- CSMA/CD (CSMA에서 한층 개선된 방식)
- 충돌을 감지해 멈춘 후에 어떻게 행동할 것인가?
  - 연속적으로  $n$ 번의 충돌이 있었다면, binary backoff
  - $\{0, 1, 2, \dots, 2^n - 1\}$  중에 한가지 수를 골라서 그만큼 기다린다!
  - 저 숫자 집합의 범위는 (개수는) 충돌 횟수가 누적될 수록 더 많아진다.
  - 그러므로, 충돌이 많을수록 더 기다리는 시간이 확률적으로 길어지게 설정된 것이다.
  - 왜 그렇게 설정했을까?

- 나 말고도 누군가가 frame을 전송했구나, 하지만 몇명이 전송했는지 알 수가 없어.
- 집합 내에서, 각각 다른 숫자를 골라야지 (숫자=기다리는 시간) 충돌이 없으니까
- 충돌이 많을수록 딜레이도 길어짐
- backoff = delay
- backoff(delay) 시간은 사람들이 (frame transmission)이 많을수록 늘어남.
- 단점 : 사용자가 느끼는 지연 시간이 길다.

### 3. taking turns

토큰을 가지고 있는 호스트만이 데이터를 전송할 수 있는 개념

토큰을 가지고 있는데 전송할 데이터가 없으면 토큰을 다음 사람에게 넘김

문제점 : 누군가 토큰을 잃어버리면 망함 ( single control failure 과 동일한 단점 )

⇒ 이렇게 세가지 장단점 있는 방식

일반적으로는 인터넷 패킷 스위칭에는 csma/cd 이용함

와이파이는 csma/a 사용함 → 다음 수업