Week 9.

Link Layer Services

차례

- **Link Layer Services**
  - Link layer 기초
  - Error Detection & Correction
  - Multiple Access Protocols
  - MAC Protocols without Collision
    - Channel Partitioning
    - Taking turns
  - MAC Protocols with Collision
  - Address Resolution Protocol
  - Ethernet 이더넷
  - Web Request in Real Life

Link Layer Services

Link Layer Basics

- Node : 네트워크의 호스트나 라우터 같은 것들
- Link : 노드를 직접 연결해주는, 물리적인 연결 등. Wired, wireless 두가지가 있음.
- Frame : 링크 레이어(2계층) 에 생성되는 패킷. 다른 계층의 패킷과의 측별한 차이는 없음.
  (트랜스포트 레이어(4계층)에서 만드는 패킷을 세그먼트, 네트워크 레이어(3계층)에서 만드는 패킷을 데이터 그램.)
- 링크 레이어의 역할
  서로 연결 되어 있는 노드 사이에서 데이터 그램을 전달해주는 역할.
- 링크 레이어의 기능
  - Packet collision handling
  - Bit error handling

Link Layer Basics

- 링크레이어의 서비스
  - Framing, link access
    - 위의 계층에 있는 데이터 그램을 전달 받아서 앞에는 헤더, 뒤에는 트레일러를 붙여서 프레임을 만든다.
    - Channel access : 노드간 다 대 다 연결을 할 때 채널을 공유하게 되는데, 누가 전송을 하고 누가 기다릴 것인지를 결정하는 기술.
    - MAC 주소를 부여 받고 이 주소를 가지고 데이터를 주고 받음.
  - Flow control
    - 링크 레이어에서의 flow control은 직접적인 연결을 하고 있는 두 노드 사이에서 데이터를 너무 빨리 보내지 않도록 조절하는 것.
  - Error detection
    - 링크를 통해서 데이터를 보낼 때, 비트에러가 발생할 수 있는데, 이를 검사함.
  - Error correction
    - 에러가 난 상황, 에러 패킷의 형태를 보고 원데이터를 추정해서 복구, 복원함. 모든 링크 레이어에서 지원하는 것은 아니고, 특정 기술에 한 함.

- Sending side 에서는 framing, flow control을 제공함. Error checking bit를 추가함.
- Receiving side에서는 에러 검사, flow control를 제공함.

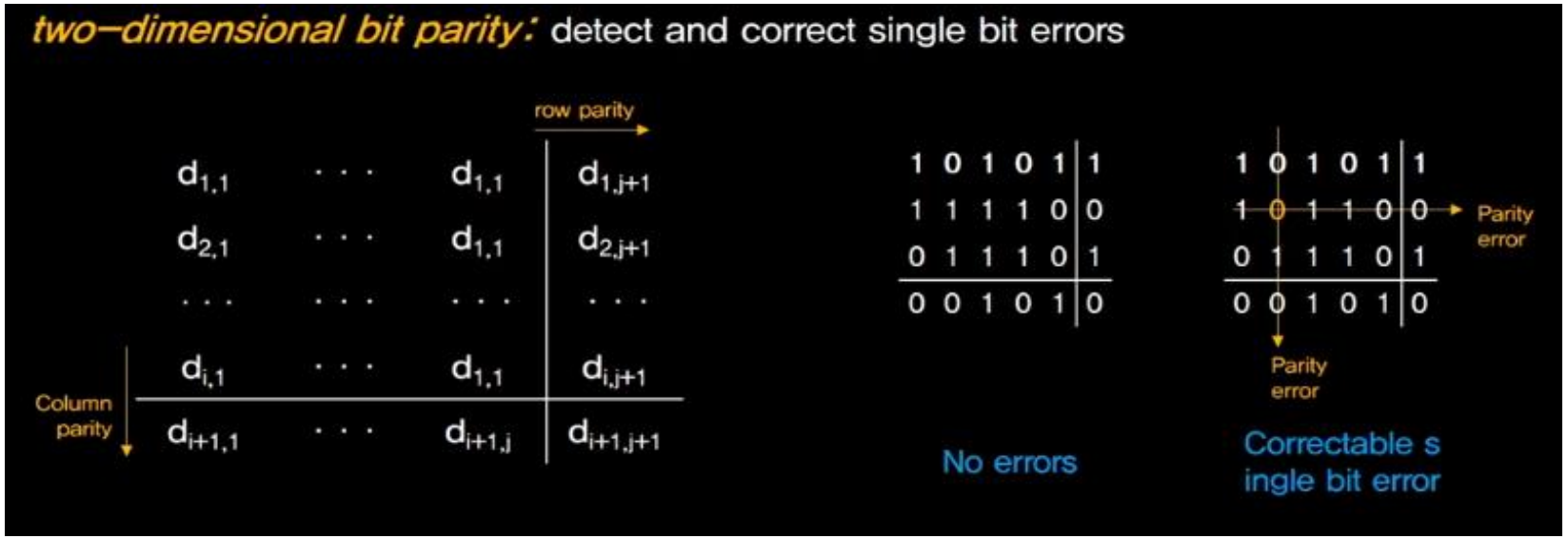
9-2) Error Detection & Correction

- 에러 감지 및 수정하는 데에 있어 대부분의 알고리즘이 Redundancy를 이용.
- 즉, redundant한 부가적인 정보를 붙여가지고 전송 함. 그러나 완벽하게 감지하고 수정할 수는 없음.
- 링크 레이어 에서는 Datagram에다가 원 데이터를 담은 뒤에 EDC (Error Detection and Correction bit)를 덧붙여서 전송.
- 정확성을 높이려면 EDC를 더 많이 붙이면 되지만, 데이터 전송에서 오버헤드가 크게 발생.
- 그러므로 trade-off를 생각해서 잘 결정해야 함.

Error Detection & Correction

Parity check

- 보내는 쪽과 받는 쪽에서 약속을 해서 1의 개수가 짝수가 or 홀수가 되도록 함.
  홀수면 odd parity, 짝수면 even parity
- Ex) 만약 odd parity를 사용한다고 가정했을 때, 원데이터 에 1이 짝수개만큼 들어있으면
  그 데이터에 parity bit로 1을 추가해서 보내고, 원데이터에 1이 홀수개만큼 들어있으면
  parity bit로 0을 붙여서 보낸다
- 그러나 에러가 두 비트 이상이 동시에 발생했을 때는 감지하지 못해서 2차원 parity bit 방식도 제안되었다



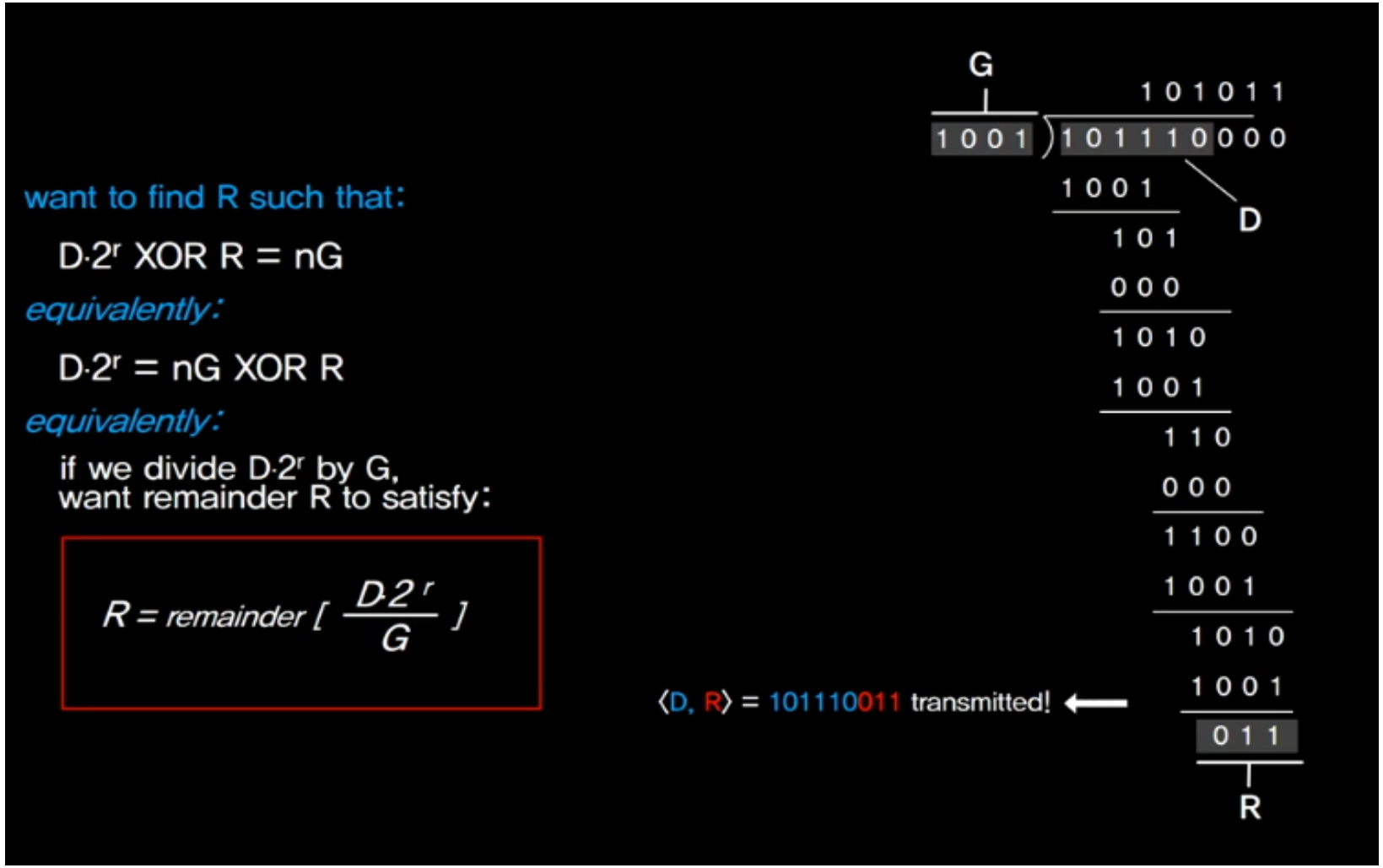
Error Detection & Correction

CRC (Cyclic Redundancy Check)

- 가장 널리 쓰이고 있음. 실제로 이더넷에서 사용하는 기술.
- 이것도 역시 원데이터 뒤에 CRC 비트를 붙이는 방식이다.
- Sender 와 Receiver 사이에 G라는 약속된 코드가 있다. CRC비트의 길이가 r비트라고 했을 때, G는 r+1비트의 크기를 가진다.
- 그리고 데이터를 보내거나 받을 때, CRC비트를 포함한 데이터의 전체길이를 G 로 나누어서 딱 나누어 떨어진다면 에러가 없다고 판단한다.
- 이더넷이나 와이파이에서도 사용 된다.

Error Detection & Correction

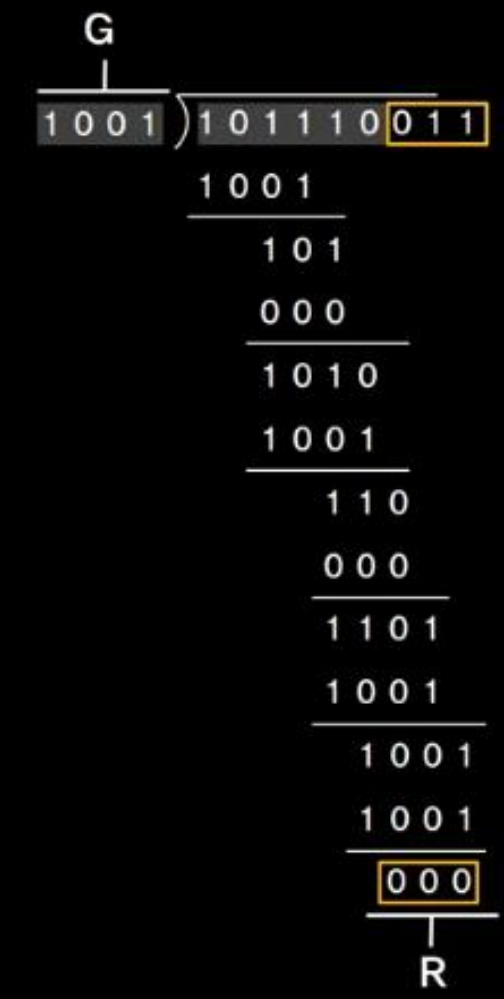
CRC (Cyclic Redundancy Check) – sender



Error Detection & Correction

CRC (Cyclic Redundancy Check) – Receiver

- Receiving host receives packet, and checks the CRC:
  - see if the remainder is 0 by dividing ⟨D, R⟩ with G



Error Detection & Correction

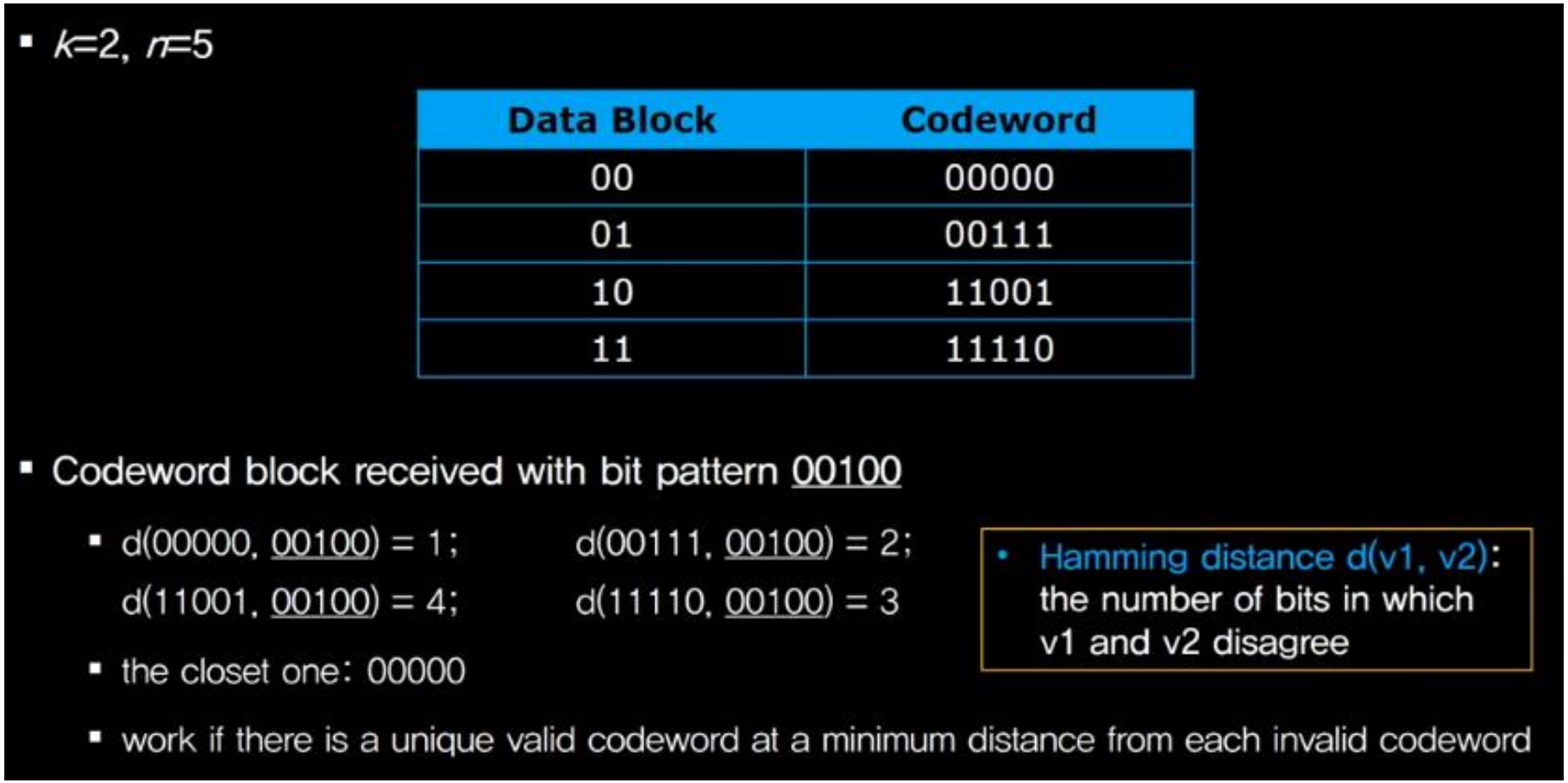
FEC (Forward Error Correction) or channel coding

- 에러를 실제로 수정할 수 있었으면 좋겠다 싶어서 나온 기술.
  데이터를 전달 받은 Receiver가 수정을 함.
- FEC에서 전송하는 데이터의 형태를 코드워드(codeword)라고 한다.
- 데이터의 길이를 k, 코드워드의 길이를 n이라고 할 때, (n-k)만큼의 parity bit가 데이터 뒤에 붙는 형태이다.
- N은 k보다 크거나 같다.
  redundancy : (n-k)/k, code rate : k/n 으로 구할 수 있다.

Error Detection & Correction

FEC (Forward Error Correction) – Hamming code

- FEC에서 가장 간단하고 많이 쓰이는 기술이 hamming code.



- k=2, n=5

Data Block	Codeword
00	00000
01	00111
10	11001
11	11110

- Codeword block received with bit pattern 00100
  - d(00000, 00100) = 1; d(00111, 00100) = 2;
    d(11001, 00100) = 4; d(11110, 00100) = 3
  - the closet one: 00000
  - work if there is a unique valid codeword at a minimum distance from each invalid codeword

- Hamming distance d(v1, v2): the number of bits in which v1 and v2 disagree

Error Detection & Correction

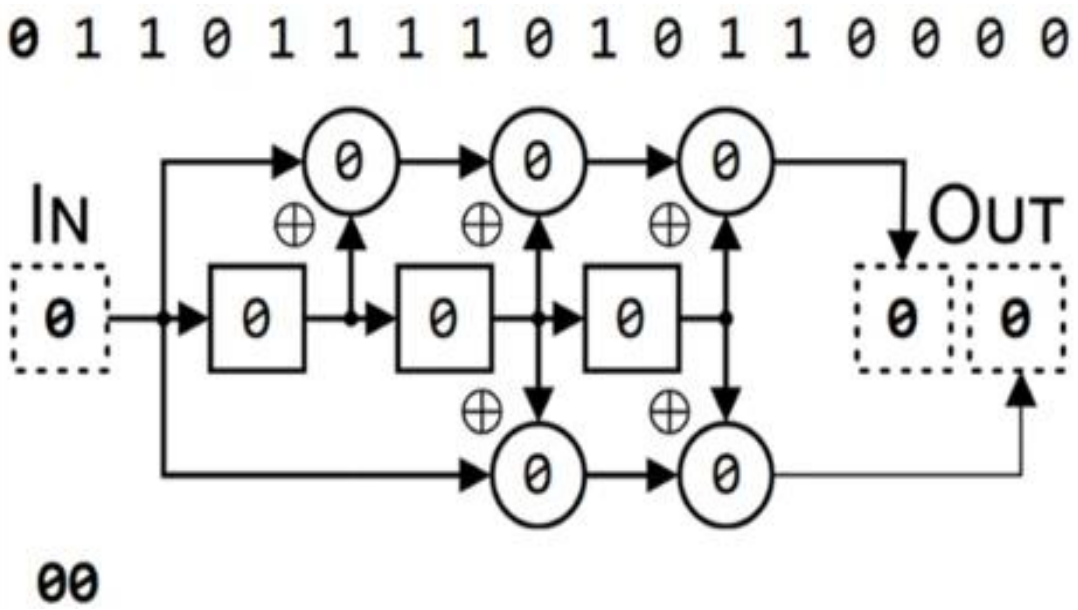
FEC (Forward Error Correction) – Hamming code

- 5bit 의 2진수면 32가지의 경우의 수가 있는데 왜 00000, 00111, 11001, 11110만 사용하는지?
  - 두 개의 코드워드 사이의 비트 차이를 Hamming distance라고 한다.
  - 각 코드워드 간 distance는 최소 3(최대 4)이다. 코드워드간 Distance가 서로 짧다면, 한 비트짜리 에러가 발생했을 때, 이를 감지하지 못할 수 있다.
  - 한 비트만 에러가 난다면 수정이 가능해지고, 두 비트만 에러가 난다면 감지가 가능해지고, 세 비트짜리 에러는 감지하지 못한다.
  - 더 큰 비트 차이의 에러를 감지하고 싶다면 그만큼 codeword를 늘리면 된다.

Error Detection & Correction

FEC (Forward Error Correction) – Convolutional code

- 통화 같은 연속적인 데이터는 모아놨다가 블록단위로 코드워드를 만들어 보내기 어렵다. 그래서 고안된 것.
- Convolutional code는 매 비트 마다 전송하는 코드워드가 생긴다.
- 이전 데이터의 히스토리가 현재 비트를 전송하려는 코드로 변경시키는데 영향을 준다.



0이 들어갔을 때는 00

01까지 들어갔을 때는 00 11
011까지 들어갔을 때는 00 11 01
0110까지 들어갔을 때는 00 11 01 01
01101까지 들어갔을 때는 00 11 01 01 11

→ 에러를 감지하고 수정하는 기능이 강해짐.
요새 블루투스에서 이런 식으로 사용함.

9-3) Multi Access Protocol

- 링크는 크게 두가지가 있다.

- Point-to-point
  - ■ 하나의 호스트와 다른 호스트가 1:1로 연결 되어있음.
- Broadcast(shared medium)
  - ■ 채널을 공유함. 버스나 RF(와이파이, 위성 등)

- 여러 노드가 동시에 공유하면 충돌이 발생하게 되는데 이런 문제를 해결하는 것을 Multiple Access Protocol 이라고 부른다.
- 줄여서 MAC Protocol이다.

Multi Access Protocol

Ideal MAC Protocol

1. 노드 중에서 하나만 동작 할 경우에는 그 노드가 모든 capacity를 다 사용할 수 있으면 좋겠다.
2. 만약 N개의 노드가 전송을 하려고 하는 경우 모든 노드들이 R/N이라는 똑같은 capacity(=data rate)를 할당 받을 수 있으면 좋겠다.
3. 누가 전송하고, 전송하지 않고를 결정하는 것을 개별 노드들이 알아서 결정을 하되 충돌을 최소화 할 수 있으면 좋겠다. (=완전히 분산적으로 동작할 수 있으면 좋겠다.)
4. 간단했으면 좋겠다.

Multi Access Protocol

MAC Protocol 분류

- Channel partitioning : 전체 채널을 분할을 해서 나눠준다. Time slot이나 주파수 또는 코드로 나눠서 하나의 piece를 개별적인 유저가 갖게끔.
- Taking turns : 채널을 나누는 것이 아니라 순서를 정해 줌.

MAC protocols without collision

- Random access : 위의 두 방식은 중앙 집중형적인 개념이 들어가지만, 이 방법은 개별 노드들이 알아서 눈치껏 채널을 사용하는 방식.

MAC protocols with collision

→ Random Access는 위의 두 방법에 비해서 완전히 분산적이라는 장점이 있으나, collision이 발생할 수 있고, 그 collision을 해결하는 방식도 같이 제공이 되어야 한다.

9-4) MAC protocols without collision

Channel Partitioning

- Channel Partitioning : FDMA (Frequency Division Multiple Access)
  - 개별 노드마다 서로 다른 주파수를 할당하기 때문에 충돌이 일어날 일이 없음.
  - 그러나 보낼 데이터가 없을 때는 해당 채널이 놀게 됨. = 자원낭비
- Channel Partitioning : TDMA (Time Division Multiple Access)
  - 프레임을 타임슬롯으로 나누고 각각의 타임 슬롯을 특정 사용자에게 할당하는 방식.
  - 전화망에서 실제로 이런 방식으로 나누어서 사용중.
  - 그러나 보낼 데이터가 없을 때는 타임 슬롯이 낭비가 되는 단점이 있음.
- Channel Partitioning : CDMA (Code Division Multiple Access)
  - 과거에 휴대전화에서 많이 썼던 방식. 휴대전화 사용자들이 기지국으로부터 코드를 받아서 자기가 보낼 데이터를 이 코드로 인코딩해서 보냄.
  - 기지국에서는 적용한 코드를 거꾸로 다시 적용시켜서 해당되는 데이터를 얻을 수 있음. FDMA, TDMA보다 효율이 훨씬 좋음.
  - 전체 주파수나 시간으로 나누지 않고 다 쓰다보니 그럼.

MAC protocols without collision

Taking turns

- Taking turns : polling
  - Master가 모든 slave에게 돌아가면서 순서를 물어본다.
  - 충돌은 없으나 오버헤드가 있을 수 있다.
  - 마스터 노드에 문제가 생기면 전체가 통신을 못한다. (single point of failure)
- Taking turns : Token passing
  - 마스터가 있는게 아니라 Token을 가진 노드만 데이터를 전송할 수 있음.
  - 데이터를 보내고, 다 보내면 Token을 다음 노드에게 보냄.
  - Token을 관리하는 오버헤드 발생, Token이 데이터가 없는 노드를 다 거치고 오므로 latency 발생
  - Token을 가진 노드가 갑자기 고장나면 전체가 통신을 못한다. (single point of failure)
  - 그래서 FDDI (Fiber Distributed Dual Interface) 라는 것이 고안된다.
    - Primary ring으로 데이터를 보내다가 primary ring이 끊어지면 secondary ring으로 보낸다.
    - 안정성이 더 확보 되어짐.

9-5) MAC protocols with collision

Random Access Protocols

- Random Access Protocols : 채널 파티셔닝이나 순서를 정해주는 마스터의 기능이 없음.
- 그래서 이상적인. MAC 프로토콜이 갖는 특성인 '하나의 노드만 전송할 게 있다면 채널의 전체 capacity를 이용함'과 '개별적으로 알아서 수행'이라는 특성을 가짐.
- 그러나 단점은 눈치 보면서 알아서 전송하기 때문에 collision이 발생할 수 밖에 없음.

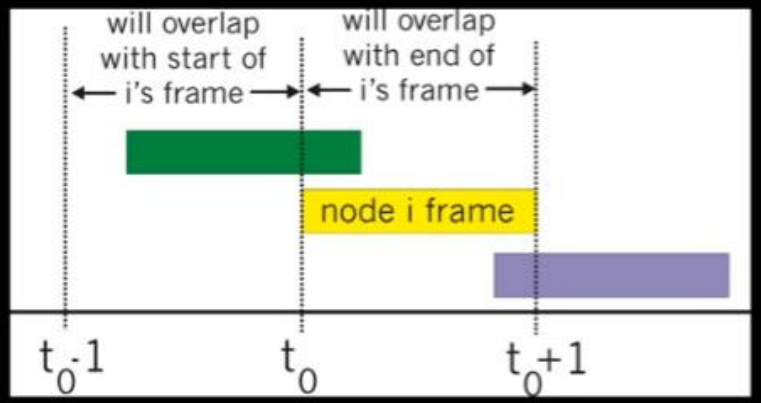
- 대표적인 Random Access Protocols
  - ALOHA
  - CSMA

MAC protocols with collision

ALOHA

- 71년도에 만들어짐. 모든 프레임들이 같은dd 길이를 갖게끔 함.
- 모든 디바이스들의 전송하는 rate이 같다.
- Collision이 발생하면 랜덤한 시간이 흐른 뒤에 재전송을 성공할 때까지 함.
- 프레임의 타임슬롯 내에 전송을 시도하는 노드의 평균 수를 G라고 가정하자.

- Collision analysis
  - frame sent at t_0 collides with other frames sent in $[t_0-1, t_0+1]$



Max. prob. of successful trans.: 0.184

- Assuming number of stations → ∞
  - G: average number of transmission–attempt nodes per timeslot

$$P(k \text{ nodes transmit in a timeslot}) = \frac{G^k e^{-G}}{k!}$$

$$P(k \text{ nodes transmit in 2 timeslots}) = \frac{(2G)^k e^{-2G}}{k!}$$

$$\therefore P(\text{no transmission in 2 timeslots}) = e^{-2G}$$

Then, frame delivery prob. in a timeslot $S(G) = Ge^{-2G}$

If differentiated, $S'(G) = (1-2G)e^{-2G}$

$\therefore$ When $G = 0.5$, $S(G)$ is the maximum : $S(0.5) \approx 0.184$

MAC protocols with collision

Slotted ALOHA

- 모든 노드들의 시간이 synchronized 여야 함. 전체 air time도 똑같은 사이즈로 나눔.
- 기존 ALOHA는 보낼 데이터가 있으면 아무 때나 전송했지만 slotted ALOHA는 준비가 되었어도 타임슬롯의 시작 시각에 맞춰 전송을 시작해야 함.
- 기존 ALOHA는 자기가 선택한 타임슬롯 외에도 직전 타임슬롯에도 누군가가 전송하면 안됐음. 그런데 slotted ALOHA는 타임슬롯의 시작 시각에 맞춰 전송을 시작하기 때문에 충돌의 확률이 반으로 줄어듦.
- 그러므로 max efficiency 는 37%로 증가함.

- 근데 이렇게 노력해도 37%밖에 안돼?

➔ 각 노드들이 다른 노드를 전혀 케어하지 않기 때문. 자기가 보낼 데이터가 있으면 그냥 보내버림.

➔ 그러면 좀 다른 노드들을 신경쓰는 프로토콜을 만들어 보면 어떨까?

MAC protocols with collision

CSMA (Carrier Sense Multiple Access)

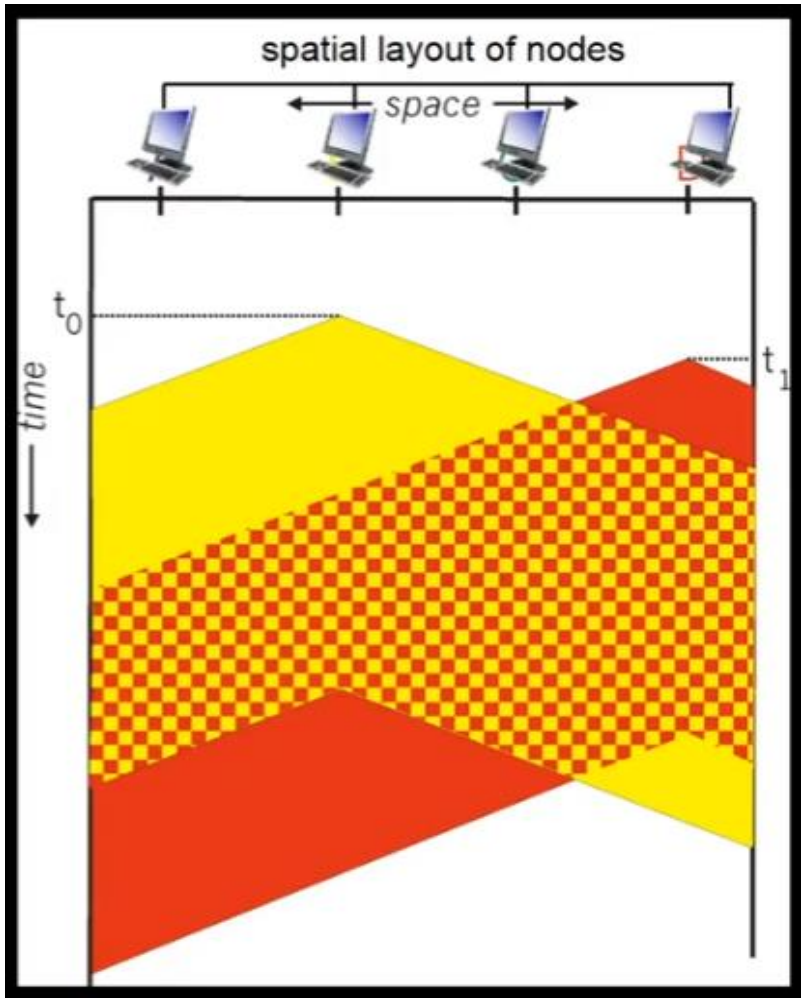
- 전송 전에 반송 주파수(carrier frequency)를 먼저 감지한다.

  Listen before transmit

- 그래서 채널로 아무도 보내고 있지 않으면 자기가 전송하고,

  채널에 누가 보내고 있으면 자기의 전송을 딜레이 함.

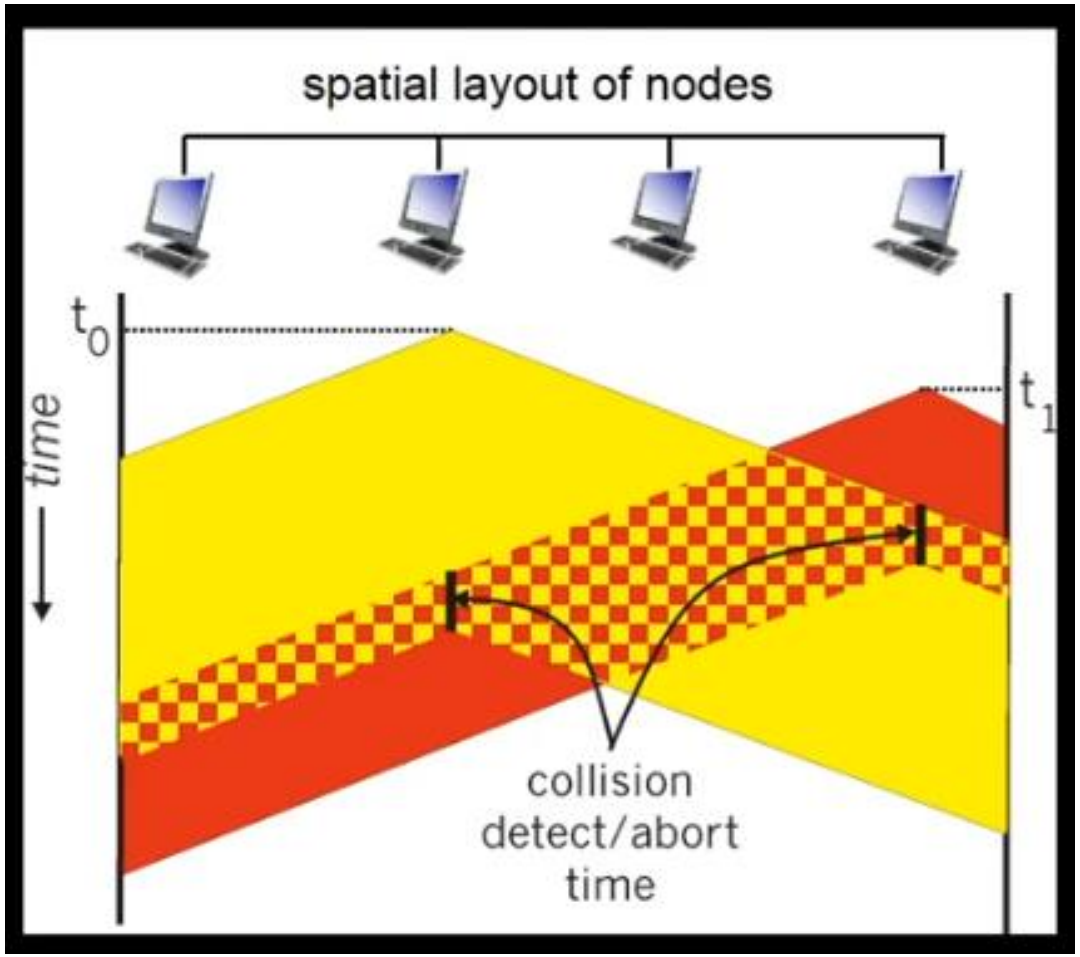
- 그래도 충돌은 발생함



MAC protocols with collision

CSMA / CD (Collision Detection)

- 기존과 똑같은데, 충돌을 감지하는 순간 전송을 중단하도록 되어있음.



MAC protocols summary

Types	Pros and Cons
Channel partitioning	• share channel efficiently and fairly at high load • inefficient at low load: delay in channel access, 1/N bandwidth allocated even if only 1 active node!
Random access	• efficient at low load: single node can fully utilize channel • high load: collision overhead
Taking turns	• look for best of both worlds! • control overhead, single point of failure

- 전체 트래픽 로드가 높을 때는 Channel partitioning이 좋음. 하지만 낮을 때는 bandwidth를 낭비함.
- Random access는 로드가 낮을 때는 효율이 높으나, 모든 노드들이 전송할 게 많은 경우에는 계속 collision이 발생하여 안 좋음.
- Taking turns는 두가지를 다 해결할 수 있음.

9-6) Address Resolution Protocol

- Basics

  - Address 는 MAC address를 뜻함. IP 주소는 논리적인 주소 (Logical Address)
    비교하자면 집주소.
  - 그러나 MAC주소는 링크 레이어에서 부여되는 주소이며 물리적 주소다. (Physical Address)
    비교하자면 주민등록 번호.
  - 그래서 IP주소는 portable하지 않지만 MAC 주소는 portable함.

IP주소가 있는데 MAC은 왜 있는 걸까?

- MAC이 원래 있었던 거고, IP가 나중에 생긴 것.
- 각각의 레이어 기술을 모듈화하여 독립적으로 발전할 수 있게끔(네트워크의 계층화) 하기 위해 상위 계층에서는 가상의 주소 체계인 IP를 지원하게끔 만들어 놓은 것이다.
- IP 라우팅을 통해서 목적지까지 데이터가 오게 되면 그 다음에는 MAC 주소, 링크 레이어 기술을 통해서 해당 호스트를 찾아가야 하기 때문.

9-6) Address Resolution Protocol

- Introductions

  - 각각 랜안의 컴퓨터들은 ARP 테이블을 가짐.
    테이블에는 컴퓨터가 사용하는 IP주소와 그 컴퓨터의 실제 MAC 주소, TTL, 매핑의 유효시간에 대한 정보가 담겨 있음.

  - 그래서 라우터가 데이터 그램을 받으면 목적지 주소에 해당하는 entry를 찾고 MAC 주소를 찾음.

  - 그리고 그 MAC 주소를 담아서 이더넷 프레임을 만들어서 전송함.

  - ARP는 범위에 따라 두 가지로 나뉜다.
    - 같은 랜 안에서의 ARP
    - 다른 랜으로 라우팅 할 때의 ARP

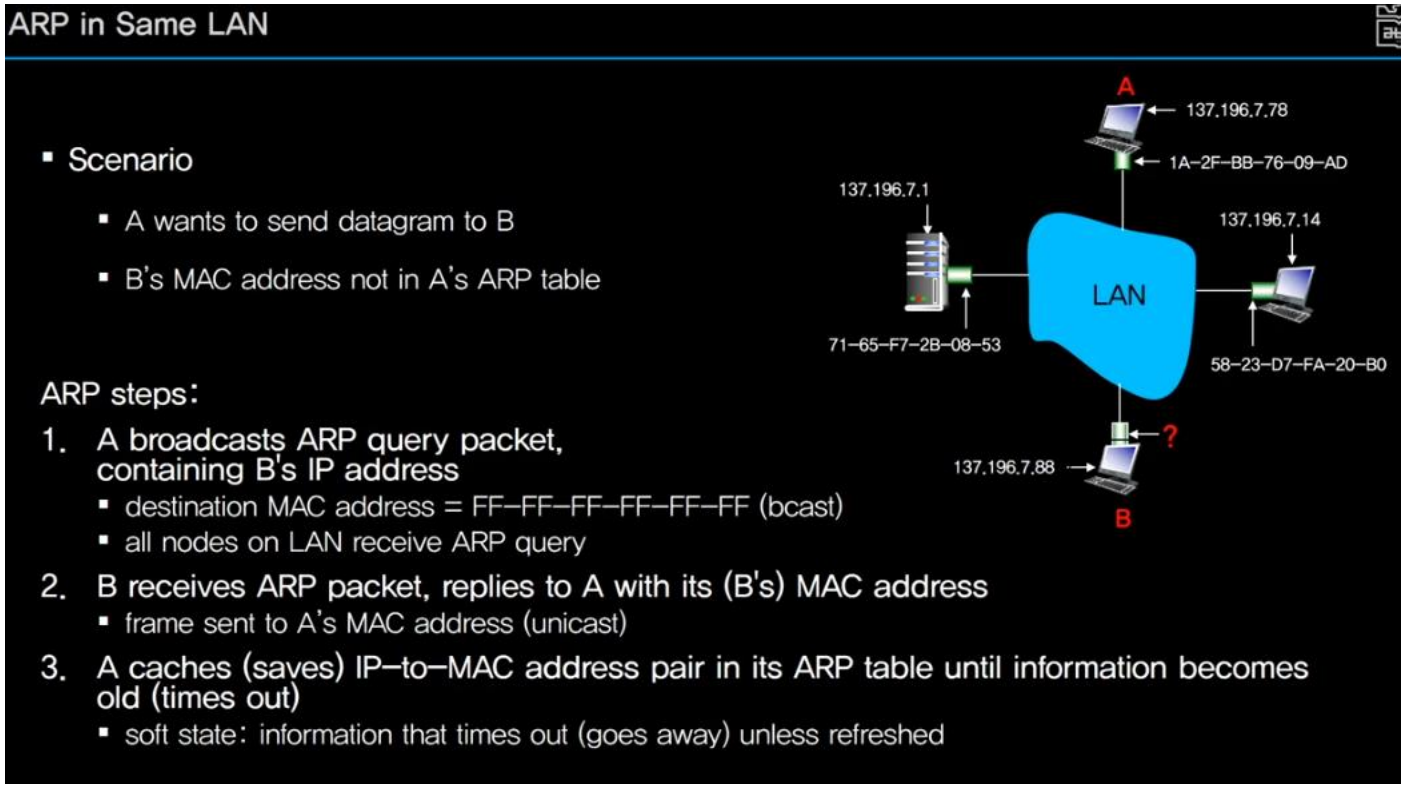
9-6) Address Resolution Protocol

- 같은 랜 안에서의 ARP

1. A가 ARP 쿼리 메시지를 broadcasting 한다. 쿼리 메시지에는 B의 IP 주소가 담겨있다.

2. 모든 컴퓨터가 메시지를 보게되고, B 컴퓨터는 자신의 MAC 주소를 담아서 reply를 한다.

3. A가 주소를 알게 되면 보내려고 했던 데이터 프레임을 만들어서 전송하고, 해당 MAC 주소를 자기 ARP 테이블에 담는다.



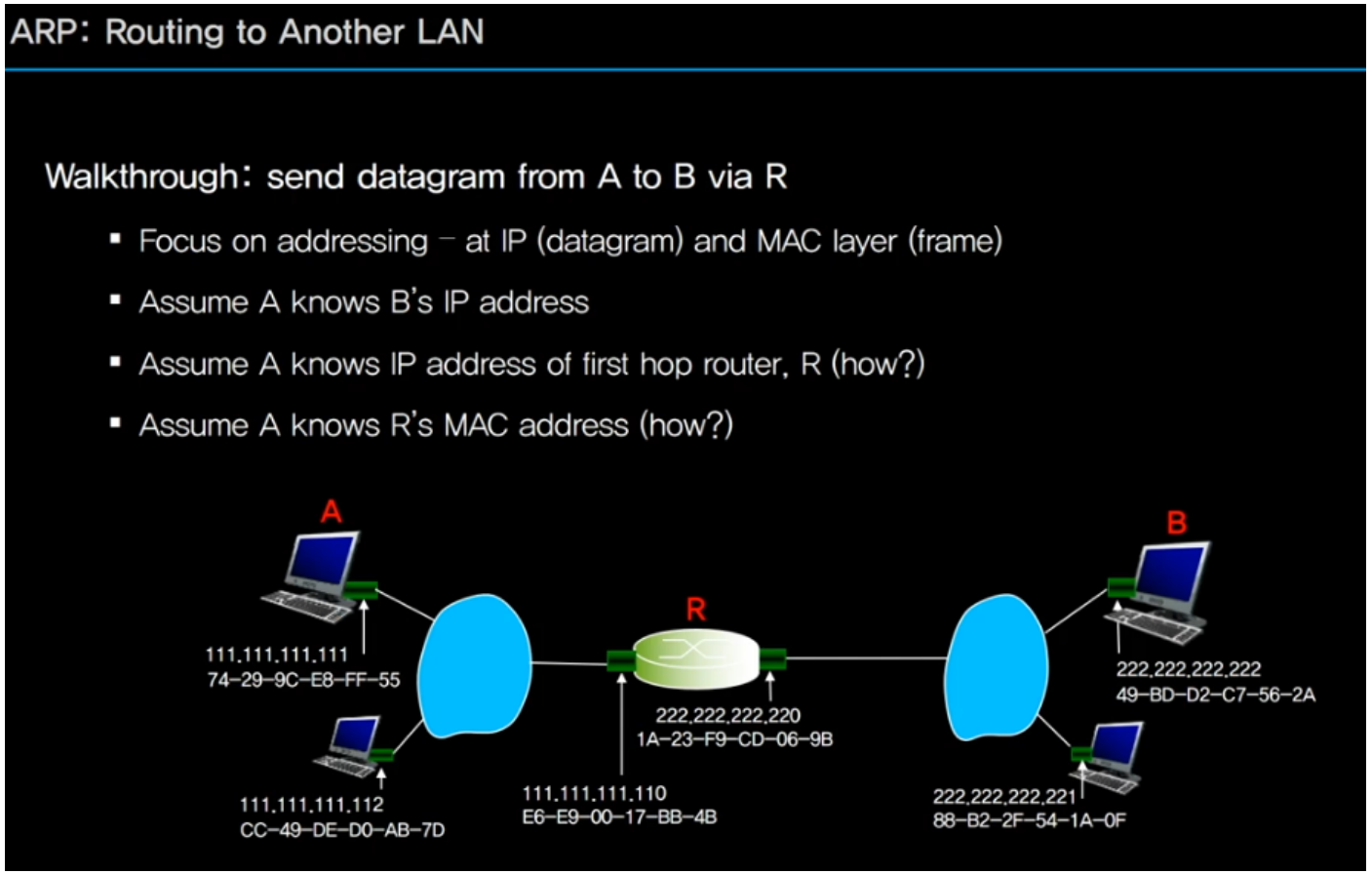
9-6) Address Resolution Protocol

- 다른 랜으로 라우팅 할 때의 ARP

1. A가 라우터 R의 MAC 주소를 담아서 전송

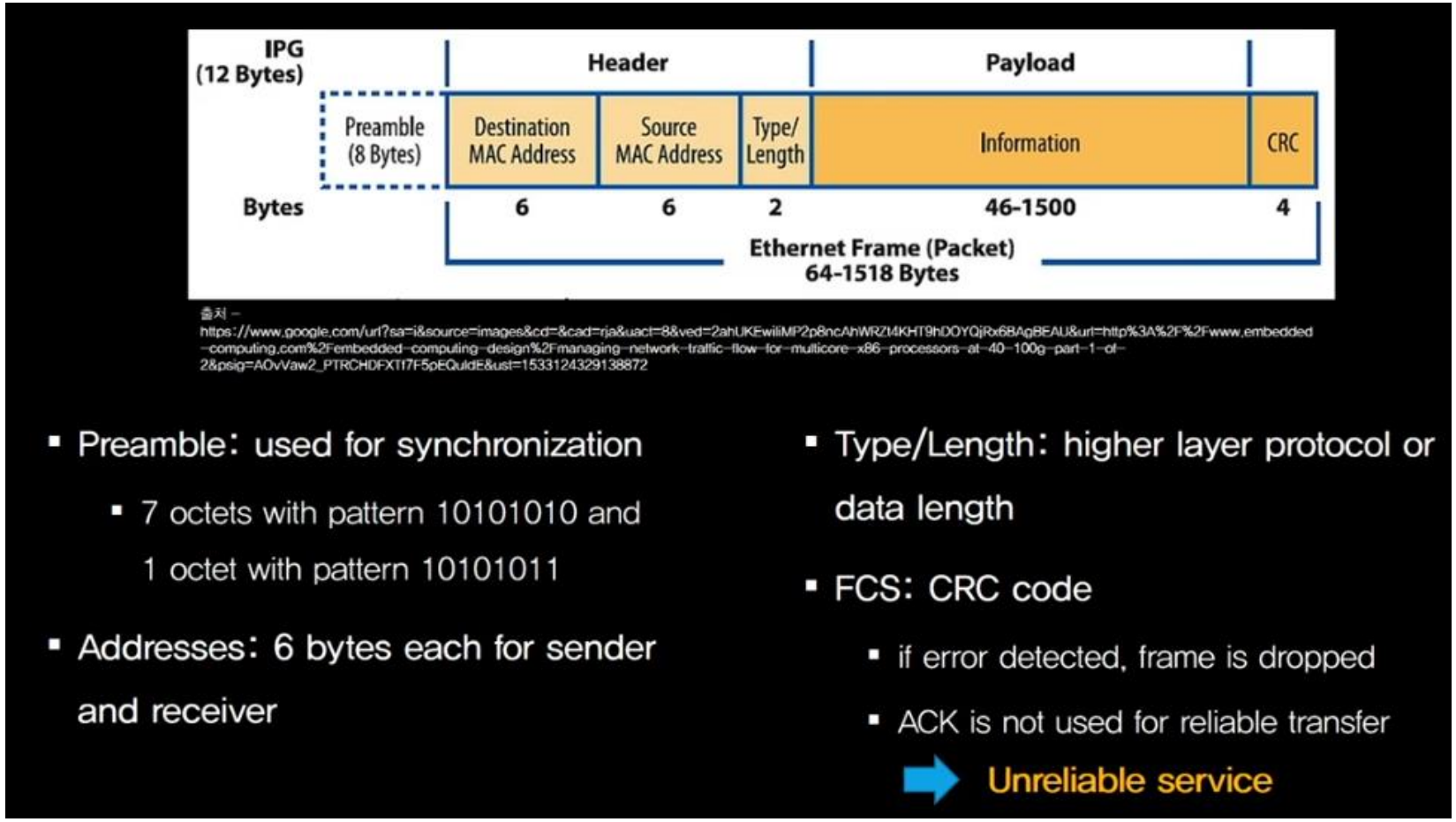
2. 라우터 R은 해당 패킷을 열어보고 IP주소를 확인함으로써 이 메시지가 자신한테 온 것이 아니라 A가 대신 전달해달라고 보낸 것임을 알게 됨.

3. 자신의 ARP 테이블에 있는 MAC 주소 중에서 해당 IP주소를 갖는 컴퓨터를 확인하고 B의 MAC 주소를 담아서 B에게 보낸다.



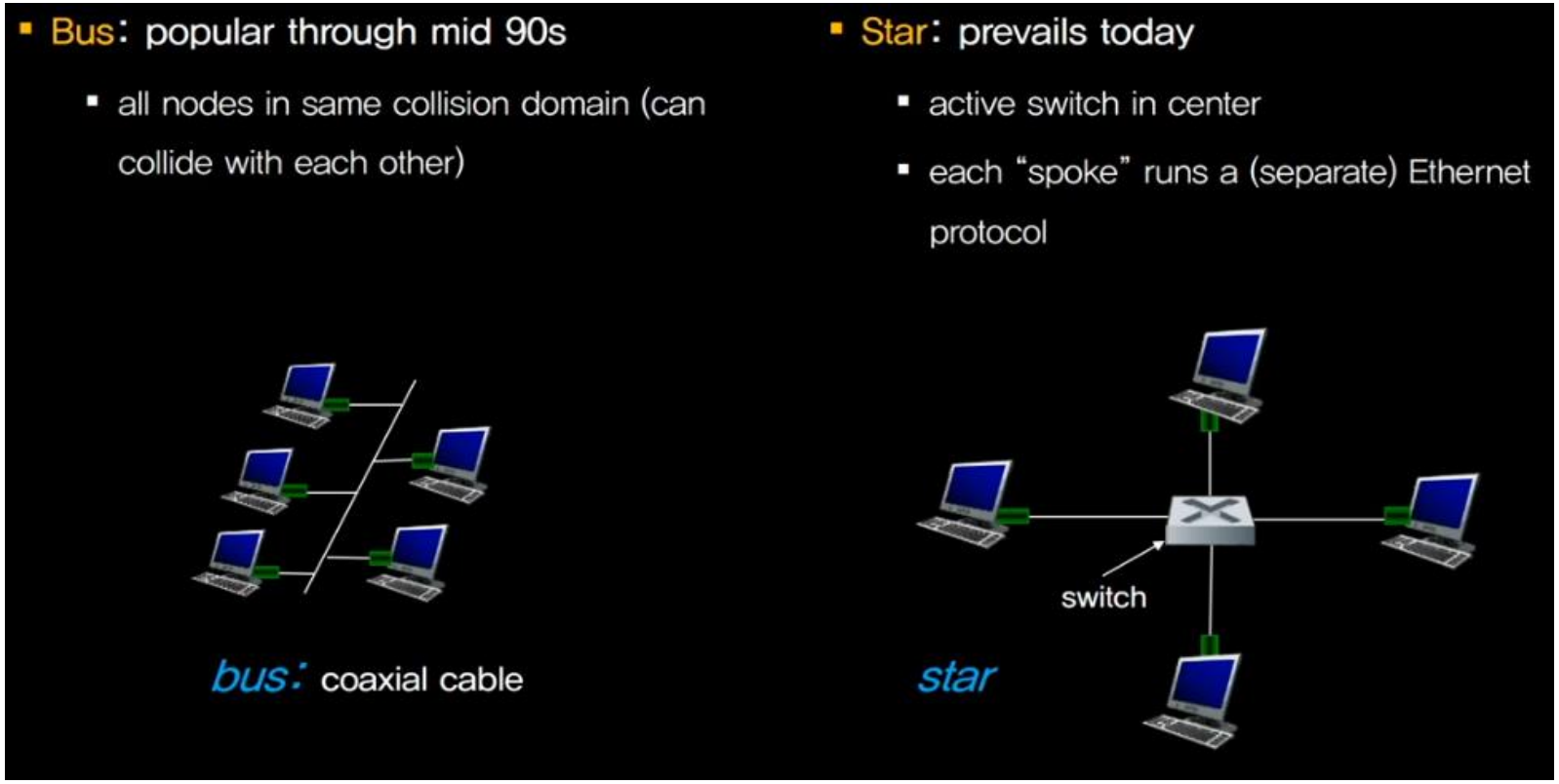
9-7) 이더넷 (Ethernet)

- Ethernet frame



9-7) 이더넷 (Ethernet)

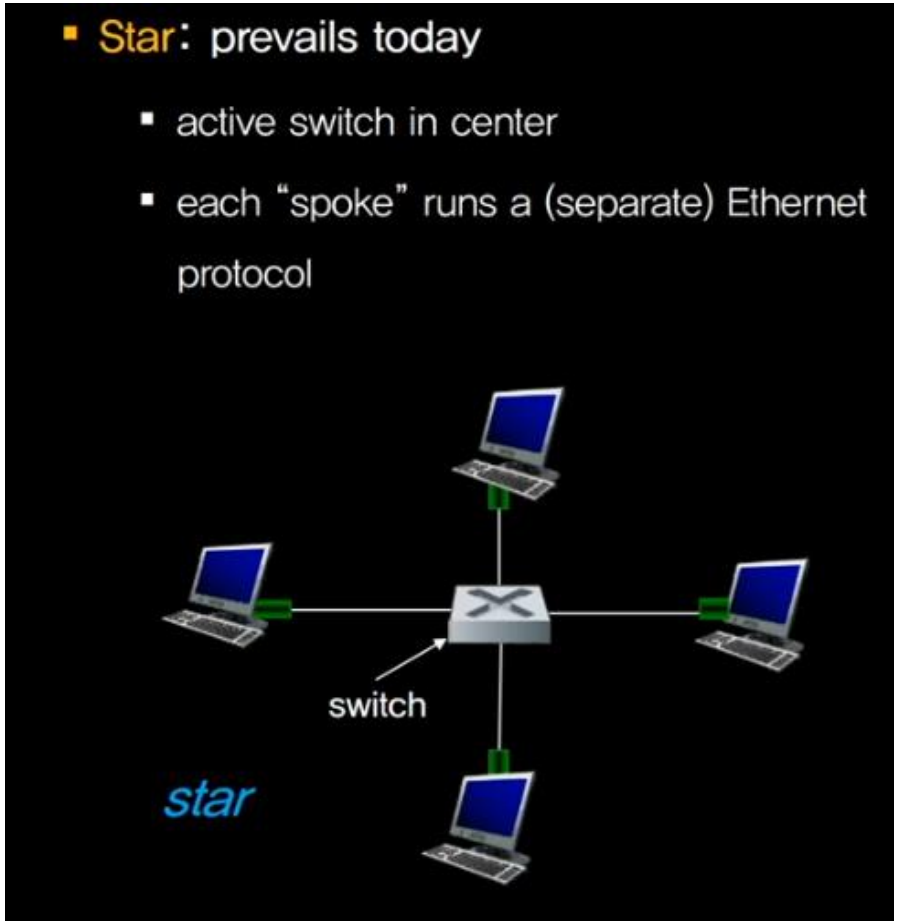
- Physical topology



- 처음에 Metclafe가 이더넷을 개발할 때는 bus 형태의 여러 노드들이 이더넷 카드로 물려져 있는 형태의 topology를 생각함.
- 최근에는 하나의 스위치 허브에 노드들이 연결되는 star형식의 topology가 사용되고 있음. 기존 bus 형식과는 다르게 충돌을 피할 수 있음.

9-7) 이더넷 (Ethernet)

- Star 방식에서 Switch의 역할.
  - 우선 하나의 링크에는 나와 스위치만 있기 때문에 다른 노드와 링크를 공유하지 않음. 그래서 충돌을 피할 수 있음.
  - 데이터를 버퍼링 할 수 있음. 또한 스위치 포워딩 테이블이 있어서 버퍼링한 것을 어디로 보내야하는 지도 앎
  - 3계층의 라우터와 비슷한 역할. 둘 다 포워딩 테이블을 가지고 있는 것도 그렇고
- Switch table은 어떻게 만들어지나
  - 호스트가 먼저 데이터를 보내오면 호스트의 MAC주소를 테이블에 저장함.
  - 만약 A가 A`에게 보내려고 하는데 스위치가 A`의 MAC주소를 모른다면 모든 호스트에게 플러딩(flooding)을 해버린다. 그러면 A`만 reply가 온다. → Self-learning



9-8) Web Request in Real Life

1. 학생이 학교에 도착해서 컴퓨터를 키면 IP 주소를 할당 받아야 함. → DHCP를 이용한다.
   추가로 학교 서브넷을 담당하는 라우터의 IP주소를 알아야함.
2. DNS 서버의 주소를 알아야 함.
   A. DHCP 리퀘스트가 만들어지고 UDP, IP이더넷을 거쳐서 전송이 됨.
   B. 근데 어디로 전송되어야 할 지 모르기 때문에 broadcast가 됨.
   C. DHCP를 지원하는 DHCP서버에서 응답을 하게 됨
   D. 노트북이 사용할 주소와 1st hub router의 주소, DNS 서버의 주소를 알려줌

BOJ 2606 바이러스

3. 구글 서버가 사용하는 IP 주소를 가져오기 위해 DNS 쿼리를 날림
   A. DNS 시스템의 IP주소는 알고 있는 상태기 때문에 MAC 주소만 알면 됨 그러므로 ARP가 필요함.
   B. ARP 쿼리를 날려서 1st hub router의 MAC 주소를 알아옴.
   C. 이제 DNS 쿼리가 바깥으로 날아갈 수 있음.
4. 구글 서버에 접속
   A. HTTP는 반드시 TCP 위에서 동작하기 때문에 TCP 3-way handshaking을 해야함.
      그래서 TCP에 SYN Flag를 세팅해서 SYN segment를 구글 서버에 보냄.
   B. 구글에서는 ACK를 보냄
   C. HTTP request 메시지를 보냄.

summary

- 순환 중복 검사(Cyclic Redundancy Check, CRC): 송신부에서 데이터를 체크값으로 나누어 도출한 나머지를 데이터와 함께 전송하면 수신부에서는 받은 데이터를 체크값으로 나눈 결과의 나머지를 비교함으로써 데이터의 오류 유무를 확인하는 오류 검출 방법(https://en.wikipedia.org/wiki/Cyclic_redundancy_check)

- 순방향 오류 정정법(Forward Error Correction, FEC): 송신측에서 추가한 중복코드를 이용하여 재전송 없이 수신측에서 발견된 에러를 정정하는 기술

- 주파수 분할 다중접속(Frequency Division Multiple Access, FDMA): 가용 스펙트럼을 정해진 대역폭 내에서 여러 무선 채널로 분할하여 사용하는 다중접속 방식

- 시분할 다중접속(Time Division Multiple Access, TDMA): 시간을 여러 슬롯으로 분할하여 사용하는 다중접속 방식

- 코드분할 다중접속(Code Division Multiple Access, CDMA): 보내고자 하는 신호를 그 신호의 주파수 대역보다 아주 넓은 주파수 대역으로 확산시켜 전송하는 방식. 서로 다른 코드를 사용하여 동시에 여러 사용자가 채널을 공유하는 다중접속 방식 (https://www.youtube.com/watch?v=oYRMYSIVj1o)

- 반송파 감지 다중 접속(Carrier Sense Multiple Access, CSMA): 충돌을 회피하기 위해 전송 전 다른 장치의 전송유무를 확인 후 전송하는 다중접속 방식 (https://www.youtube.com/watch?v=IAKncL67Pp4)

- 주소 결정 프로토콜(Address Resolution Protocol, ARP): 네트워크 상에서 IP 주소를 물리적 네트워크 주소로 대응시키기 위해 사용하는 프로토콜

감사합니다