

컴네 기말 정리 Chapter 6

Chapter 6 The Link Layer and LANs

1. Introduction, services

a. Link layer: introduction

- i. link layer 란?: 물리적으로 연결된 노드들 사이에 datagram을 전송하는 기능을 가짐.
- ii. host와 router를 합쳐서 node라고 함.
- iii. link 의 종류
 - 1. wired links
 - 2. wireless links
 - 3. LANs
- iv. layer-2 packet을 frame이라고 함. (datagram을 encapsulate함, 즉 포함하고 있음)

b. Link layer: context

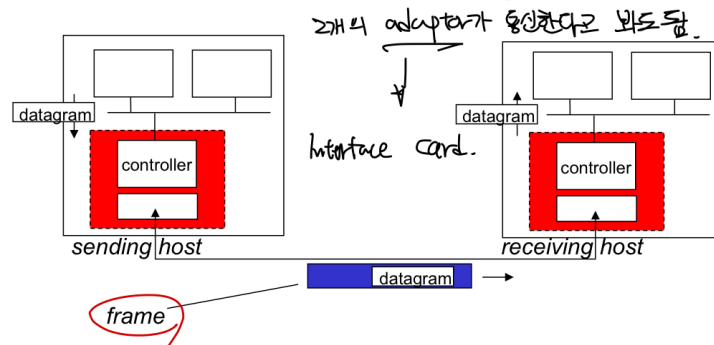
- i. 각각의 layer별로 서로다른 link protocol을 사용 할 수 있음.
- ii. 각각이 link protocol은 서로 다른 service를 제공함.
- iii. 예시)
 - 1. tourist = datagram
 - 2. transport segment = communication link
 - 3. transportation mode = link layer protocol
 - 4. travel agent = routing algorithm

c. Link layer services

- i. framing, link access
 - 1. 상위 계층에서 보내준 datagram을 fram화 하고, header를 추가하고 뒤에 trailer를 추가해 캡슐화함

2. channel에 접속하는 기능을 제공.
 3. Mutiple Access control(MAC) address는 frame header에서 사용, IP address와 다름
- ii. reliable delivery between adjacent nodes
1. error가 낮은 low bit error link(광섬유, 꼬아진 선)에선 잘 사용하지 않음.
 2. 선이 없는 links에선 error비율이 높기때문에 사용
 3. link level에서 error를 복구하면 좀 더 빨리 복구 할 수 있다는 장점이 있음.
- iii. flow control
1. 2개의 노드 사이에 전송 속도를 조절함
- iv. error detection
1. error를 감지함.
 2. receiver는 error를 detect할 수 있어야함. ex) transport layer의 checksum
- v. error 를 correction(수정)
- vi. half-duplex and full-duplex
1. 한개만 frame을 전송하거나 두개 다 동시에 frame을 전송함
- d. **어디에서 link layer는 실행 되는가?**
- i. link layer는 adaptor에서 실행됨.
- e. **Adaptors communication**

Adaptors communicating



- sending side: *datagram을 캡슐화함.*
 - encapsulates datagram in frame
 - adds error checking bits, rdt, flow control, etc.
- receiving side
 - looks for errors, rdt, flow control, etc.
 - extracts datagram, passes to upper layer at receiving side

i.

ii. 2개의 adapter(interface card) 가 통신한다고 봐도 됨.

iii. sending 측에선 datagram을 캡슐화함, error checking bit와 rdt, flow control 를 추가함.

iv. receiving 측에선 error를 탐지하고, datagram을 추출하여 상위 layer에 올림

2. error detection, correction

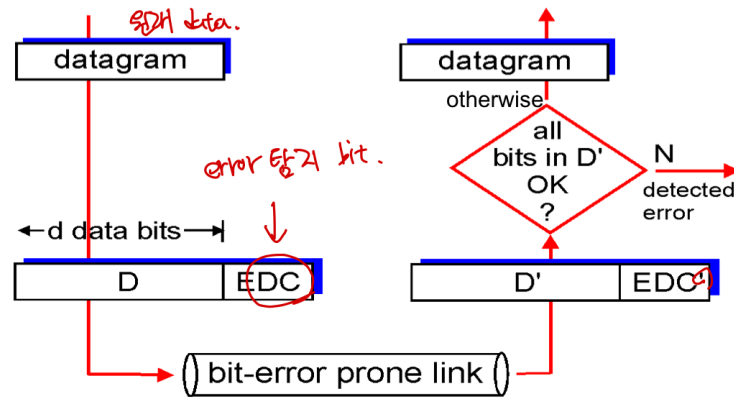
a. Error detection

EDC= Error Detection and Correction bits (redundancy)

D = Data protected by error checking, may include header fields

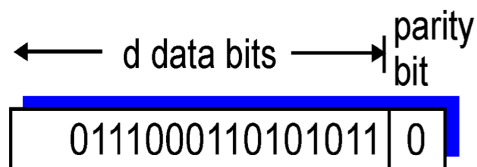
• Error detection not 100% reliable!

- protocol may miss some errors, but rarely
- larger EDC field yields better detection and correction



Link Layer and LAL

1. 원래 datagram에 EDC를 추가하여 탐지함.
2. EDC가 무엇이 있을까?
 - a. Parity checking
 - i. single bit parity

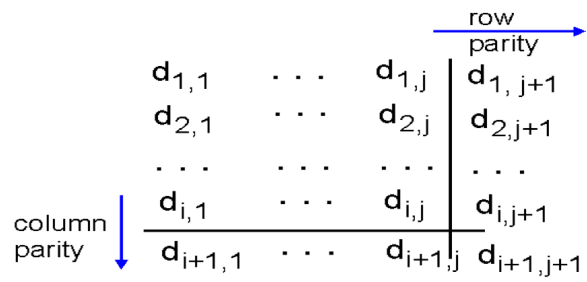


예) 1의 갯수를 홀수로 보내겠다?
parity bit 0으로 깨닫는다

1. detect single bit errors

- a. 1의 갯수를 짝수로 보낼지, 홀수로 보낼지 결정하여 parity bit의 값을 0또는 1로 설정하여 진행함.

ii. two-dimensional bit parity



```

1 0 1 0 1 | 1
1 1 1 1 0 | 0
0 1 1 1 0 | 1
-----
0 0 1 0 1 | 0

```

no errors

more
correctable

```

1 0 1 0 1 | 1
1 0 1 1 0 | 0
0 1 1 1 0 | 1
-----
0 0 1 0 1 | 0

```

parity
error

*correctable
single bit error*

1. detect and correct single bit errors

2. 2차원으로 parity error를 추가함. 이후 비교를 하여 개수가 맞지 않으면 error를 detect함.

b. Internet checksum

i. sender는 16bit 정수를 일렬로 segment를 처리함.

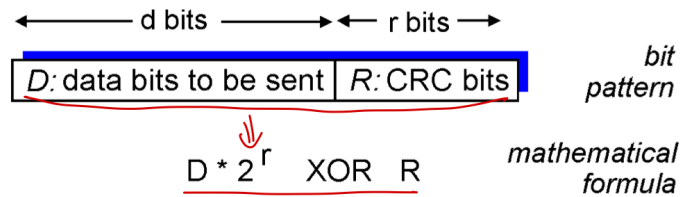
c. Cyclic redundancy check(CRC)

i.

Cyclic Redundancy Check

- more powerful error-detection coding
- view data bits, D , as a binary number
- choose $r+1$ bit pattern (generator), G
 - $G = r+1$ ex) $r=31\text{ bit}$
 $G=32\text{ bit}$
- goal: choose r CRC bits (R), such that
 - $\langle D, R \rangle$ exactly divisible by G (modulo 2)
 - D 와 G 를 이용하여 R 을 만들.
 - receiver knows G , divides $\langle D, R \rangle$ by G . If non-zero remainder: error detected!
 - $r+1\text{ bit}$ 보다 작은 error는 다 detect 나머지 값이 0이 아니면 error임.
 - can detect all burst errors less than $r+1$ bits 가능.
- widely used in practice (Ethernet, 802.11 WiFi, ATM)

$D+R$ 이
 G 로 나눴을 때
정확히 나뉘
떨어져야 함



Link Layer and LANs 6-15

- 아주 강력한 error를 탐지 할수 있는 coding이다.
- 긴 binary number인 D 를 사용. (보내는 쪽에서 알고 있음)
- choose $r+1$ bit pattern (generator) G (보내는 쪽과 받는 쪽 둘다 알고 있음)
- 목표는 CRC bits R , D 와 G 를 이용하여 R 을 만들. 즉 , $D+R$ 이 G 로 나눴을 때 정확히 나뉘 떨어져야함.
- $r+1$ bit보다 작은 error는 다 detect 가능. 나머지가 0이 아니면 error임.

d. CRC example

- 빼기가 아닌 XOR연산을 진행함.

want:

6월 4일 정호희 나뉘 떨어진다스

equivalently:

equivalently:

나머지가 R이 된다.

[illegible]

$$D \times 2^r \text{ XOR } R = nG$$
$$= D \times 2^r \boxed{\text{XOR } R \text{ XOR } R} = nG \text{ XOR } A$$

같은 비트면 0
다른 비트면 1

$$\begin{array}{r}
 10101 \\
 \hline
 1001 \overline{) 10110000} \\
 \underline{1001} \\
 0101 \\
 \underline{0000} \\
 1010 \\
 \underline{1001} \\
 0110 \\
 \underline{0000} \\
 1100 \\
 \underline{1001} \\
 1010 \\
 \underline{1001} \\
 011 \rightarrow R
 \end{array}$$

a. link 는 기본적으로 2가지 타입으로 존재한다.

- i. point-to-point : link가 있으면 양쪽에 2개의 node가 있음

컴네 기말 정리 Chapter 6

2. Ethernet switch사이에서도 사용함.

ii. broadcast(shared wire or medium)

1. 옛날 스타일의 방식

2. Hybrid Five cable (HFC) 집에서 사용했던 방식

b. Multiple access protocols

i. shared broadcast channel이 한개가 있을 때 여러개의 node가 있어서 각각 노드들이 붙어있지만 항상 데이터를 보내거나 하지 않음.

ii. 같은 시간에 2개이상의 transmission이 있으면 간섭이 일어남.

iii. multiple access protocol (분산 알고리즘(어떤 노드가 언제 전송할지 결정하는 algorithm))

1. 채널 공유를 어떻게 해야할지도 채널을 사용함. → 데이터를 주고 받는 채널을 상관없는 채널을 사용해야함.

c. An ideal multiple access protocol (이상적인 multiple access protocol)

i. R bps(bandwidth)의 channel이 주어짐.

ii. 바람직한 상황 4가지

1. 1개의 node가 전송을 원할 때, R 비율만큼 보낼 수 있어야함.

2. M개의 node가 transmit을 원하면, R/M 만큼만 사용해야함.

3. 간단해야함

4. 중앙에 하나의 노드가 있어서 분산된 결정이 이뤄져야함.

d. MAC protocol : 분류

i. 3가지의 class로 분류가 가능함.

1. channel partitioning

a. 한개의 channel을 작은 pieces로 나눠서 rbps 를 나눔

b. 각각의 piece를 node에 할당해주고 할당 된것만 사용함.

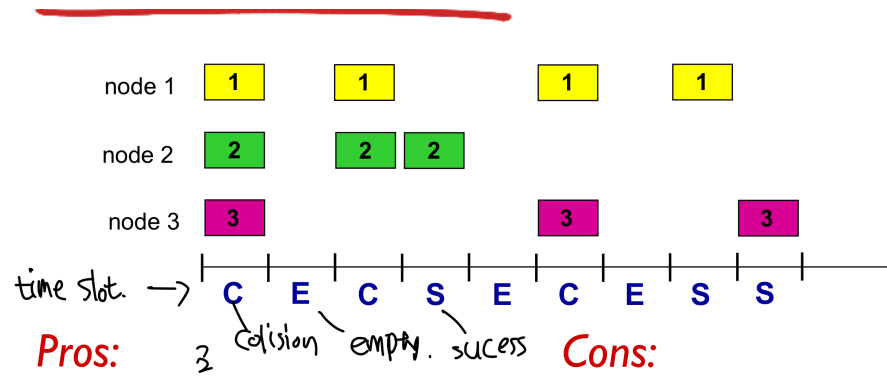
2. random access

a. 간섭이 발생 했을 때 recover가 일어남

b. 간섭을 탐지도 할줄 알아야함.

3. "taking turns"

- a. 돌아가면서 함. 각각의 노드가 순서를 정해놓고 한번씩 진행함.
- b. 나의 turn이 올 때까지 많은 시간이 걸릴 수도 있음.
- c. 어떤 node가 보낼 packet이 있으면 full rate인 R bps로 전송함.
- d. 미리 node들 사이에 조정하는 기능이 없음
- e. **Channel partitioning MAC protocols: TDMA**
 - i. TDMA(time division multiple access)
 - ii. channel을 쪼개서 round별로 access함.
 - iii. 각각의 round에서 고정된 slot을 할당 받은 후 node들은 전송이 가능함.
- f. **Channel partitioning MAC protocols: FDMA**
 - i. FDMA(frequency division multiple access)
 - 1. 전자파에 전송하는거라 주파수가 있음
 - 2. channel의 스펙트럼을 frequency band로 나눔
 - 3. 고정된 frequency band를 사용함.
 - 4. hz가 높을 수록 bit를 많이 보냄.
 - 5. channel을 frequency band로 잘라 node에 할당해줌.
- g. **Slotted ALOHA (Random access protocol)**
 - i. assumptions:
 - 1. 모든 frame이 같은 size (link layer는 network layer에서 받은 packet을 frame으로 만듦)
 - 2. 전체 시간을 frame 한개를 보낼 수 있는 시간으로 쪼갬
 - 3. 각 노드는 slot이 시작하는 시간에 전송을 함
 - 4. 모든 노드는 synchronize된 상태여야 함
 - 5. 2개 이상의 node가 slot 안에서 전송을 하면 모든 node들은 간섭을 감지함
 - ii. operation
 - 1. 노드가 fresh frame을 얻게 되면 다음 slot에 전송함.
 - 2. 간섭이 발생하면 그 다음 slot에서는 probability P 를 정하여 P 를 갖고 재전송함, random number를 생성하여 P 보다 작으면 전송, 아니면 넘어감.



iii.

iv. 3개의 node를 보내는데 9개의 slot이 사용됨.

v. 장점:

1. 하나의 노드가 들어오면 계속해서 full rate으로 보낼 수 있음. (이상적인 조건 1번 만족)
2. 좋은 분산, synchronized가 되면 알아서 노드들을 보낼 수 있음
3. 간단함.

vi. 단점:

1. 간섭, slot 을 낭비함.
2. 못쓰는 slot이 나올 수 있음
3. 모든 노드가 정확히 시간을 맞춰야함.
4. 효율성이 좋지 않음 . 0.37

h. Pure(unslotted) ALOHA

- i. slot도 필요없고 synchronziation 을 하지않아 간단함.
- ii. frame이 도달하면 바로 보냄
- iii. 간섭 가능성이 증가함
- iv. 2개의 slot동안 2개의 frame이 간섭이 일어나면 안됨
- v. 효율성이 약 0.18 정도 나옴 ALOHA보다 더 안 좋음.

i. CSMA(carrier sense multiple access)

- i. 전송을 하기 전 들음 → 전송하기전 누가 보낸것이 있는지 확인을 함.
- ii. 채널이 idle이라 확인됐다면 frame을 전송함.
- iii. channel이 바쁘면, 전송을 미룸.

- iv. 간섭은 계속 일어남: 2개의 node가 다른 노드들의 전송을 듣지 못 할 수도 있음
- v. 노드들에는 거리가 있고, propagation delay가 존재하기 때문에 간섭이 일어남.

j. **CSMA/CD (collision detection)**

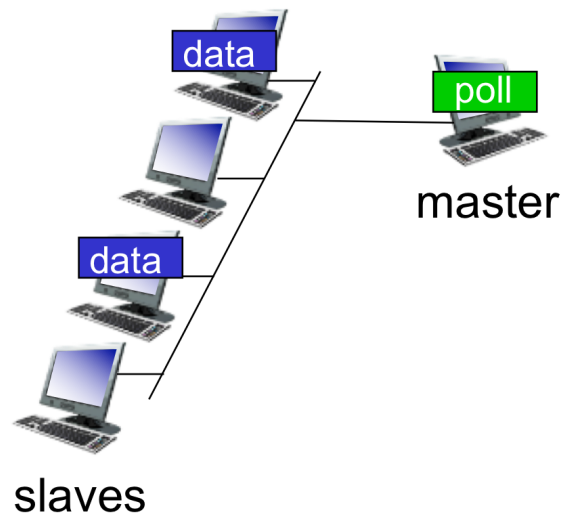
- i. 채널 감지를 한후, CSMA 처럼 미룸.
- ii. 간섭이 짧은 시간에 탐지되면 channel 낭비를 줄임

k. **Ethernet이 CSMA/CD 알고리즘을 사용함**

- i. NIC(network interface card)가 network layer에서 datagram을 받으면 frame을 만듦
- ii. 만약 NIC가 채널을 감지하면, frame을 전송함. 바쁘면 미뤘다가 재전송함
- iii. 만약 NIC가 다른 전송을 감지하지않고 다 전송한다면 NIC는 할일을 다함.
- iv. NIC가 다른 전송을 감지한다면, 곧바로 전송을 중단하고 jam signal을 보낸다.
- v. 취소를 한후, NIC는 binary backoff방식을 통해 재 전송함.

l. **“Taking turns” MAC protocols**

- i. channel partitioning과 random access의 장점을 합쳐서 만듦
- ii. channel partitioning MAC protocol
 - 1. high load일때 공평하고 효율적으로 channel공유가 가능함, 그러나 low load일 경우 channel 접근이 delay가 생기고, 1개의 노드를 위해 $1/N$ bandwidth를 할당 해줘야함.
- iii. random access MAC protocols
 - 1. high load일 경우 간섭이 발생, low load일 경우 효율적으로 single node를 완전히 쓸 수 있음
- iv. polling Algorithm
 - 1.



1. master node가 존재함. 이는 polling message를 통해 slaves node 들에게 보낼게 있는지 없는지 물어봄
2. polling하는 과정에 overhead가 발생할수 있음.
3. single point of failure → master노드가 죽을수도 있음. polling이 불가능해짐

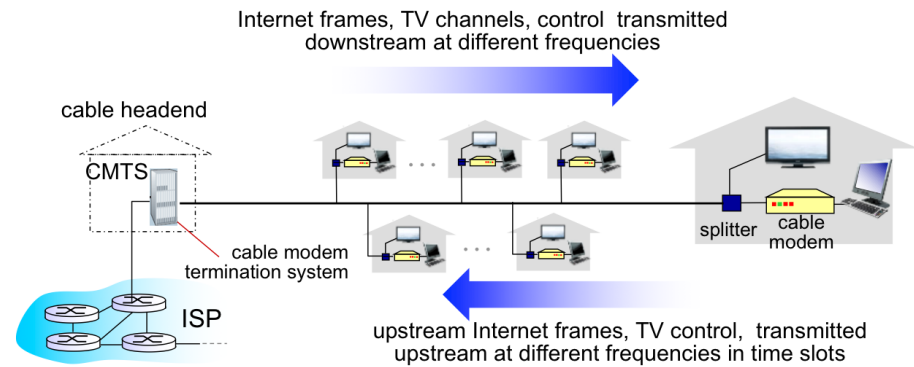
v. token passing

1. token packet을 이용하여 token 을 가지고 있어야만 node들이 보낼 수 있도록함
2. token message
3. token overhead가 발생할 수 있음
4. single point of failure → token packet이 없어지면 일을 하지 못함

vi. Cable access network

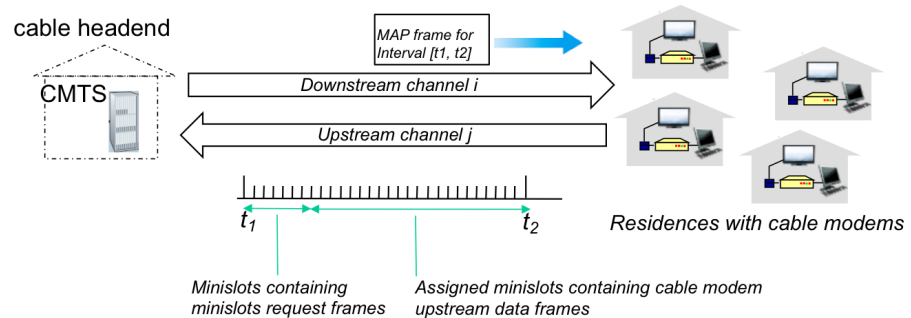
- 1.

Cable access network



- **multiple** 40Mbps downstream (broadcast) channels
 - single CMTS transmits into channels
- **multiple** 30 Mbps upstream channels → upstream에서 간헐적 전송이 있음,
 - **multiple access**: all users contend for certain upstream channel time slots (others assigned)

Link Layer and LANs 6-39



DOCSIS: data over cable service interface spec

- FDM over upstream, downstream frequency channels
- TDM upstream: some slots assigned, some have contention
 - downstream MAP frame: assigns upstream slots
 - request for upstream slots (and data) transmitted random access (binary backoff) in selected slots

Link Layer and LANs 6-4

1. cable network에서 upstream을 공유하고자 DOCSIS protocol을 사용함.
2. cable headend인 CMTS에 upstream 할 내용을 알려줌 (upstream 할 내용을 요청할 땐 충돌이 발생 할 수 있음)
3. CMTS는 그에 따른 정보를 집어다가 제공함. 그러면 충돌이 일어나지 않음

m. **Summary**

i.

Summary of MAC protocols

- **channel partitioning**, by time, frequency or code
 - Time Division, Frequency Division
- **random access** (dynamic),
 - ALOHA, S-ALOHA, CSMA, CSMA/CD
 - carrier sensing: easy in some technologies (wire), hard in others (wireless)
 - CSMA/CD used in Ethernet
 - CSMA/CA used in 802.11
- **taking turns**
 - polling from central site, token passing
 - Bluetooth, FDDI, token ring

n.

4. **LANs**

a. **addressing, ARP**

i. MAC addresses and ARP

1. 기존 IP address 는 32bit
2. MAC(or LAN or physical or Ethernet) address는 48bit.
3. MAC address 는 물리적으로 연결된 Interface들 사이에서 local(지역) 내에서만 frame을 전달하기 위한 Address

ii. LAN addresses

1. MAC address는 개개인 마다 고유한 Number가 있음, 계층구조가 없음, 이식성이 뛰어남.
2. IP addresses는 바뀔 수 있음. IP는 계층적인 구조를 가짐.

iii. ARP protocol: address resolution protocol

1. IP address를 알고 있을 때 Interface의 MAC address를 어떻게 알 수 있을까?
 - a. 각각의 host가 ARP table을 가지고 있음
 - b. IP address와 MAC address는 LAN node와 mapping을 함.

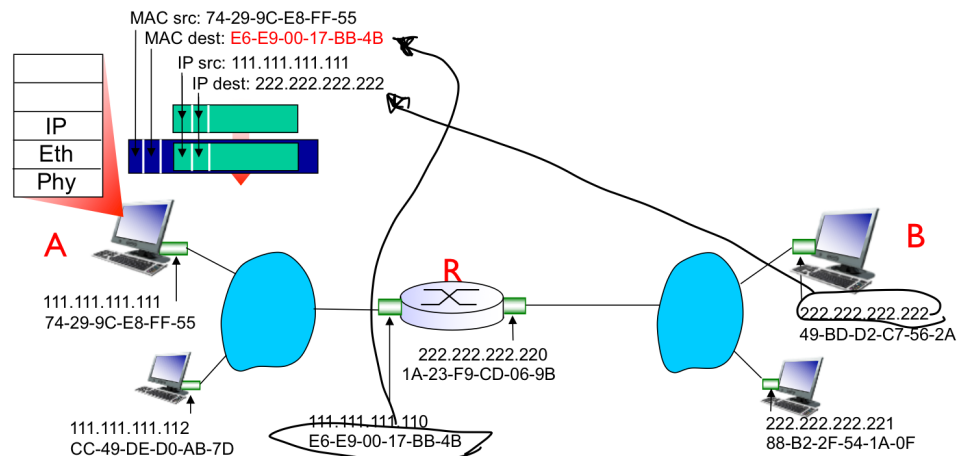
iv. ARP protocol: same LAN (ARP 작동방식)

1. B의 MAC주소가 A의 ARP table에 존재하지않음.
2. 호스트 A가 ARP쿼리를 broadcast함, LAN에 있는 모든 노드들이 ARP 쿼리를 받음
3. B는 쿼리 msg안에 자기의 IP주소가 있는 것을 확인한 후 reply를 함. 이를 A에게 보냄
4. A는 쿼리를 받으면 이를 ARP table에 집어넣음

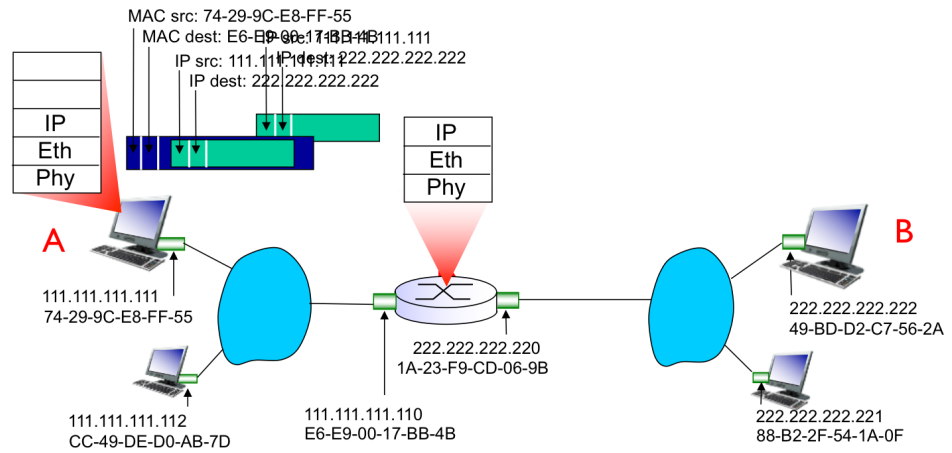
v. Addressing: routing to another LAN (subnet이 다른 host끼리 datagram을 보낼때

1.

- A creates IP datagram with IP source A, destination B
- A creates link-layer frame with R's MAC address as destination address, frame contains A-to-B IP datagram

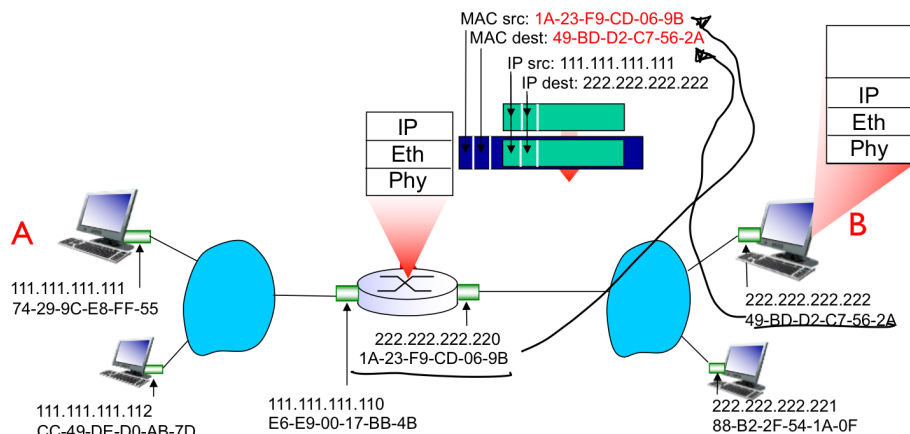


- frame sent from A to R
- frame received at R, datagram removed, passed up to IP



Link Layer and LANs 6-50

- R forwards datagram with IP source A, destination B
- R creates link-layer frame with B's MAC address as destination address, frame contains A-to-B IP datagram



Link Layer and LANs 6-51

b. Ethernet

i. Ethernet 특징

1. 유선상의 LAN을 구성할 때 제일 많이 사용됨
2. 한개의 chip과 다양한 speed가 있음
3. 맨처음 광범위하게 사용됨
4. 단순하고 싸

ii. Ethernet: 물리적인 topology

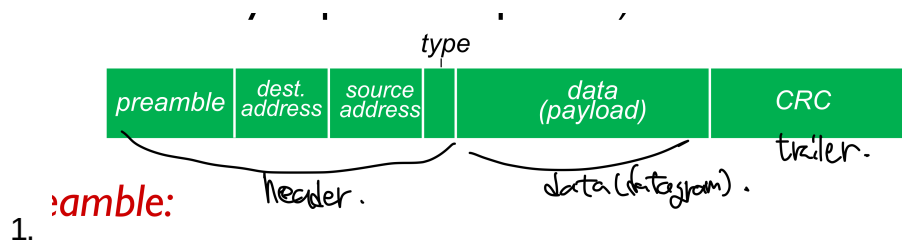
1. bus

- a. 모든 node가 충돌하는 영역이 있음
- b. 한개가 보낼 때 다른것은 사용을 못함

2. start

- a. signal을 한개가 아닌 여러개가 보낼 수 있음
- b. 중간에 hub나 switch를 뒤야함.

iii. Ethernet frame 구조



2. preamble

- a. 7 bytes의 패턴을 사용함, 10101010으로 시작하고 끝은 10101011로 끝남
- b. receiver와 sender를 동기화 하기 위해서 사용함.

3. addresses:

- a. 6 byte씩 dest MAC address와 source MAC address로 들어감

4. type:

- a. 상위 layer protocol을 가리킴

5. CRC: cyclic redundancy check

- a. frame이 drop되는 것을 감지함 → 에러 탐지

iv. Ethernet: unreliable, connectionless

- 1. connectionless: 보낼때 미리 신호를 주거나 그럴 필요가 없음
- 2. unreliable: seq num을 사용하거나 ack 등을 사용하지 않음, 보내면 끝임
- 3. Ethernet 's MAC protocol: unslotted CSMA/CD with binary backoff

c. switches

i. Ethernet switch

1. link-layer device: takes an active role (link-layer 기기)
 - a. 저장하고 Ethernet frame의 dest를 보고 전달함
 - b. hub는 해당 기능이 없음
 - c. frame의 MAC address가 들어오면 이를 destination을 본 후 선택적으로 전송함
2. transparent (투명함)
 - a. host들은 중간에 Ethernet switch가 있는지 없는지 알수 없음
3. plug-and-play, self-learning
 - a. switch는 power가 연결되면 바로 동작함.

ii. Switch: multiple simultaneous transmissions(여러개를 동시에 보내는 전송이 가능함)

1. switch는 packet을 store하고 보냄
2. 각각의 link에 대해서 Ethernet protocol이 사용됨
3. Collision domain이 분리되어 있음
4. 동시에 보낼 수 있음 → 간섭이 일어나지 않음

iii. Switch: self-learning (자기 스스로 학습을 함)

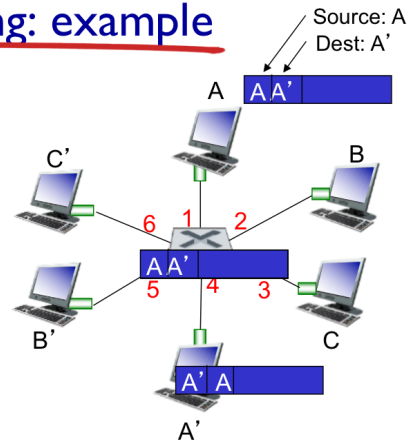
1. 어떻게 switch는 해당 번호에 누가 있는지 알까? → self learning을 통해 앎
2. frame을 받으면 해당 frame을 보낸 interface와 MAC addr를 switch table에 작성함

iv. Switch: frame filtering/forwarding

1. Switch가 frame을 받음
2. 들어오는 Link, MAC address를 기록함
3. 만약 entry에 목적지가 있는데 이미 그 frame이 도달했던 적이 있으면 frame을 드랍하고 그렇지 않으면 fram을 목적지 까지 보냄
4. entry에 목적지가 없으면 flood처리를 함(즉 다 보내버림)

Self-learning, forwarding: example

- frame destination, A', location unknown: **flood**
- destination A location known: **selectively send on just one link**



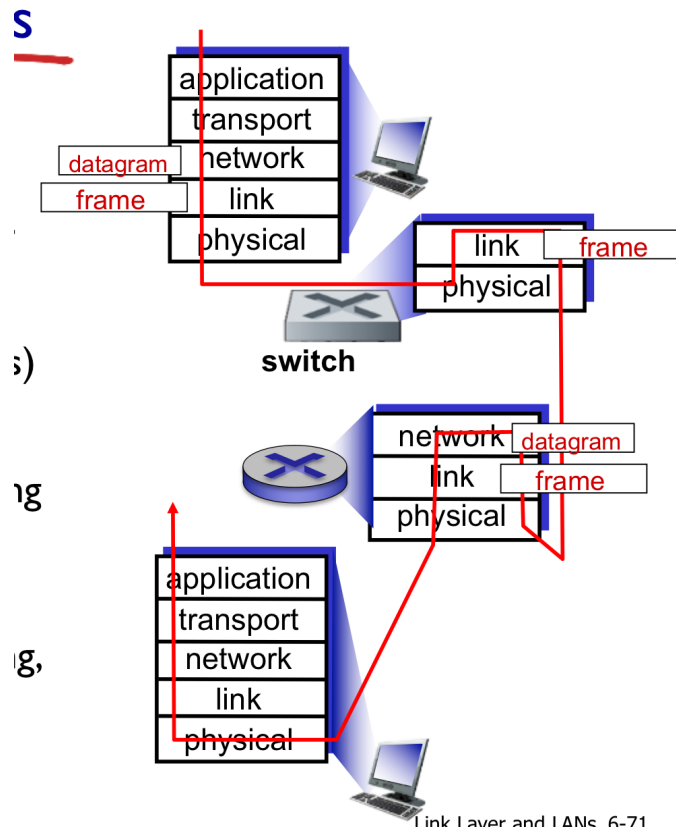
MAC addr	interface	TTL
A	1	60
A'	4	60

switch table
(initially empty)

5.

Link Layer and LANs 6-67

v. Switches vs routers

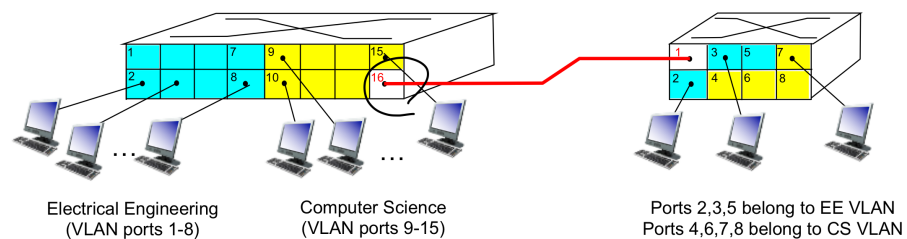


Link Layer and LANs 6-71

1. both are store-and-forward (둘다 store와 forward 기능 제공)
 - a. routers : network layer 장치, network layer header를 읽음

- b. switches: link-layer 장치, link layer header를 읽음
 - 2. both have forwarding tables
 - a. router: routing algorithm을 이용해 계산
 - b. switches: self-learning을 통해 계산
- d. **VLANs**
 - i. VLANs
 - 1. Switch에 VLAN 기능을 구성함 → 여러개의 가상의 LAN을 만들 수 있음
 - ii. Port-based VLAN
 - 1. traffic이 서로 분리됨
 - 2. VLAN을 MAC 주소 기반으로도 만들 수 있음
 - 3. VLAN Port를 유동적으로 바꿀 수 있음
 - 4. VLAN이 다른 것에서 통신을 할 때 routing을 통해 진행됨
 - iii. VLANs spanning multiple switches

VLANs spanning multiple switches



- 1. trunk port를 만들어 Link를 연결함
 - 2. 서로 다른 physical switch에 전달을 할 수 있음
 - 3. trunk port를 지날 때 802.1 Q protocol을 사용해야함(즉, VLAN ID를 포함해야함)
- 5. **link virtualization: MPLS**
 - a. **Multiprotocol label switching (MPLS)**

- i. IP forwarding을 좀더 빠르게 해보겠다가 목표임. 이때 고정된 label(MPLS header)를 사용함
 - ii. 1:1 매칭을 함
 - iii. IP datagram은 계속 IP address를 가지고 있음
- b. MPLS capable routers
 - i. 20bit label value만을 이용하여 나가는 Interface를 결정
 - ii. MPLS에서 내보내는 결정을 할 때 유연하게 결정을 할 수 있음
 - iii. link가 고장나면 재빨리 다른 경로로 바뀌줌
 - iv. MPLS forwarding table은 IP forwarding table과 다르게 존재해야함.
- c. MPLS versus IP paths
 - i. IP routing: 목적지까지의 경로가 목적지 주소에 의해 결정됨
 - ii. MPLS routing : source와 목적지 경로에 의해 결정되며 다른 경로를 만들어 줄수도 있음 , 이때 packet label을 사용함. (ip address를 보지 않고 label만 확인함) MPLS signaling을 통해 경로를 설정해줄수 있음

6. data center networking

7. a day in the life of a web request (진행상황을 쭉 봄)

- a. 가정) 한 학생이 www.google.com에 접속하려함
- b. laptop이 browser로 무엇을 보려함.
- c. DHCP를 이용하여 자기의 IP address를 얻어옴
- d. 첫번째 router의 IP address를 얻어옴
- e. DNS server의 IP address를 얻어옴. 이때 DHCP를 이용함
- f. IP 를 할당받고 ACK를 보냄, 이때 ACK를 보내는 와중에 switch는 self learning을 통해 자기의 table을 채움
- g. laptop이 DHCP를 통해 자기의 IP address하고 첫번째 router, DNS server의 IP address를 얻으면 브라우저를 실행시킴
- h. 해당 브라우저는 google.com에 대한 도메인을 찾아옴.
- i. DNS query msg를 UDP로 만듦 이후, network layer로 보내면 Ip protocol이 IP를 추가함
- j. 해당 datagram을 frame으로 만듦

- k. MAC address를 알고자 ARP을 사용함
- l. ARP query msg를 router에 보냄, 받은 router는 reply를 함.
- m. 이후 frame을 만든후 보내게 됨.
- n. router는 frame의 dest를 보고 링크를 통해 dest(DNS server)로 보냄
- o. dns server는 쿼리에 대한 reply를 해줌
- p. 해당 정보가 client에 도달함. → www.google.com에 대한 IP address가 포함하게 됨
- q. client는 구글닷컴에 대한 IP address를 알게 되어 web server에 대한 TCP connection을 만듦
- r. 이후 TCP SYN segment를 보내게 됨, 도달하면 이를 TCP SYNACK를 보내 다시 reply함
- s. 이러면 HTTP request를 만들어 web server에 다시 보냄
- t. HTTP는 response msg를 보내줌.
- u. 그러면 google의 web page를 볼 수 있음