링크 계층 1, 2, 3

Link layer

Introduction. Link layer

패킷은 gateway router한테 보냄.

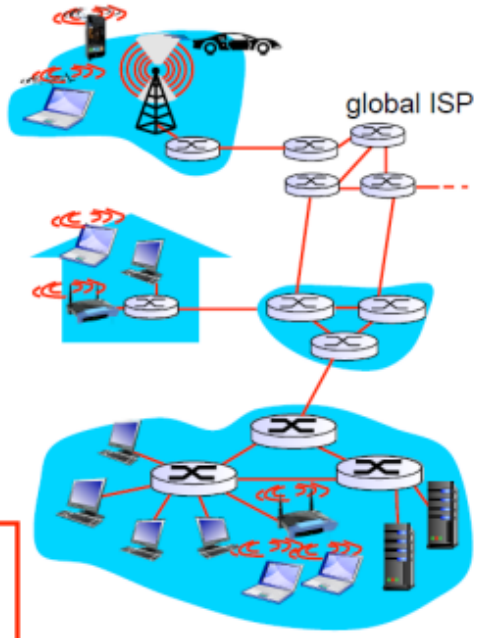
패킷을 보내면 physical layer에서 전파신호로 변환되어 갈 수 있는 곳으로 전파됨.

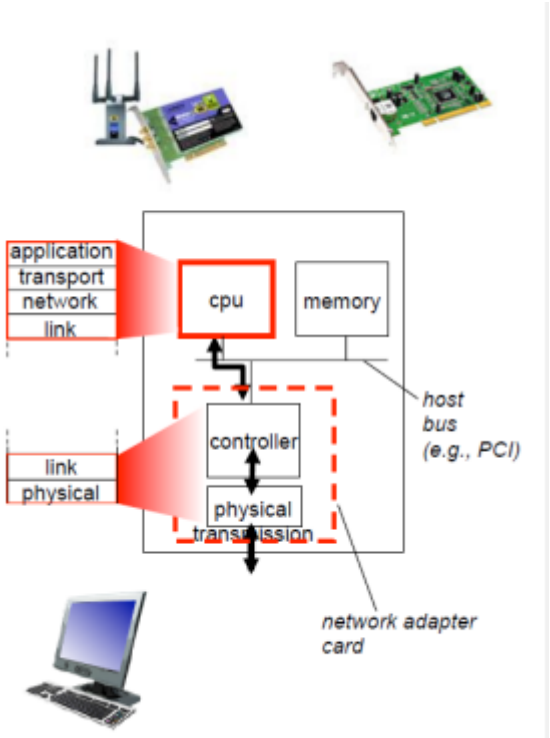
충돌이 발생하지 않아야, 발생해도 해결해야 함 → link later에서 하는 일

terminology:

- hosts and routers: nodes
- communication channels that connect adjacent nodes along communication path: links
  - wired links
  - wireless links
  - LANs
- layer-2 packet: frame, encapsulates datagram

data-link layer has responsibility of transferring datagram from one node to *physically adjacent* node over a link

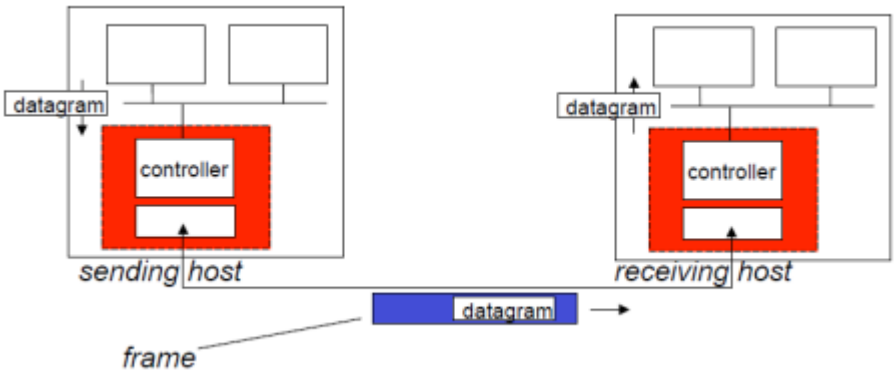




link layer는 어디에 존재하는가?

→ 어댑터 (aka network interface card NIC) or on a chip

Adaptors communicating



- sending side:
  - 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

Multiple access links, protocols

brodcast medium (shared wire on medium) .

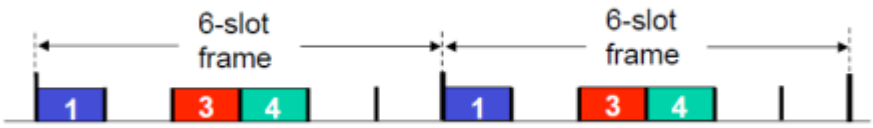
같은 매체를 사용하면서 발생하는 충돌을 최소화 하기 위한 기술 → **“Multiple Access Control (MAC)”**

MAC protocols: taxonomy

1. channel partitioning
2. random access
3. taking turns

1. Channel partitioning MAC protocols : TDMA

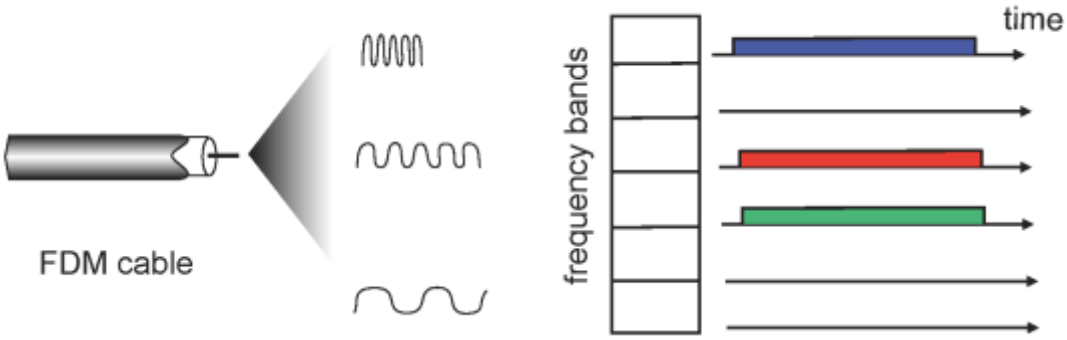
TDMA : time division multiple access



주어진 시간에만 전송할 수 있음 → 낭비됨

1. Channel partitioning MAC protocols : FDMA

FDMA : frequency division multiple access



정해진 주파수만 사용해서 전송 가능

Random access protocols

충돌이 일어 날 수 있음.

충돌을 어떻게 개선할 수 있는가가 핵심.

ex.

- slotted ALOHA
- ALOHA

- CSMA, CSMA/CD, CSMA/CA

CSMA (carrier sense multiple access)

CSMA : listen before transmit

if channel sensed idle : transmit entire frame

if channel sensed busy, defer transmission

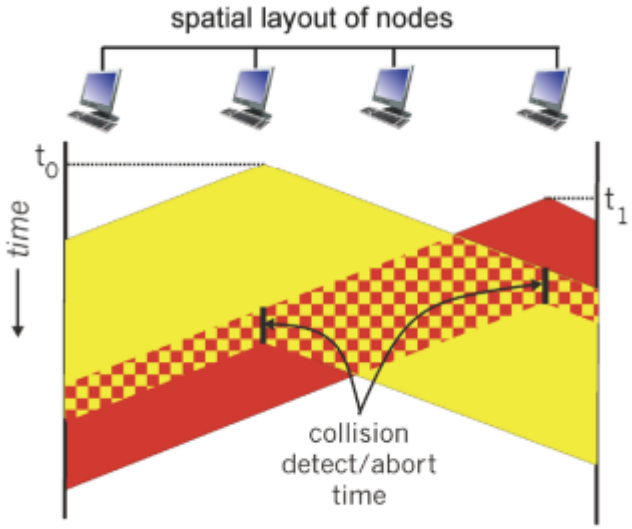
CSMA collisions

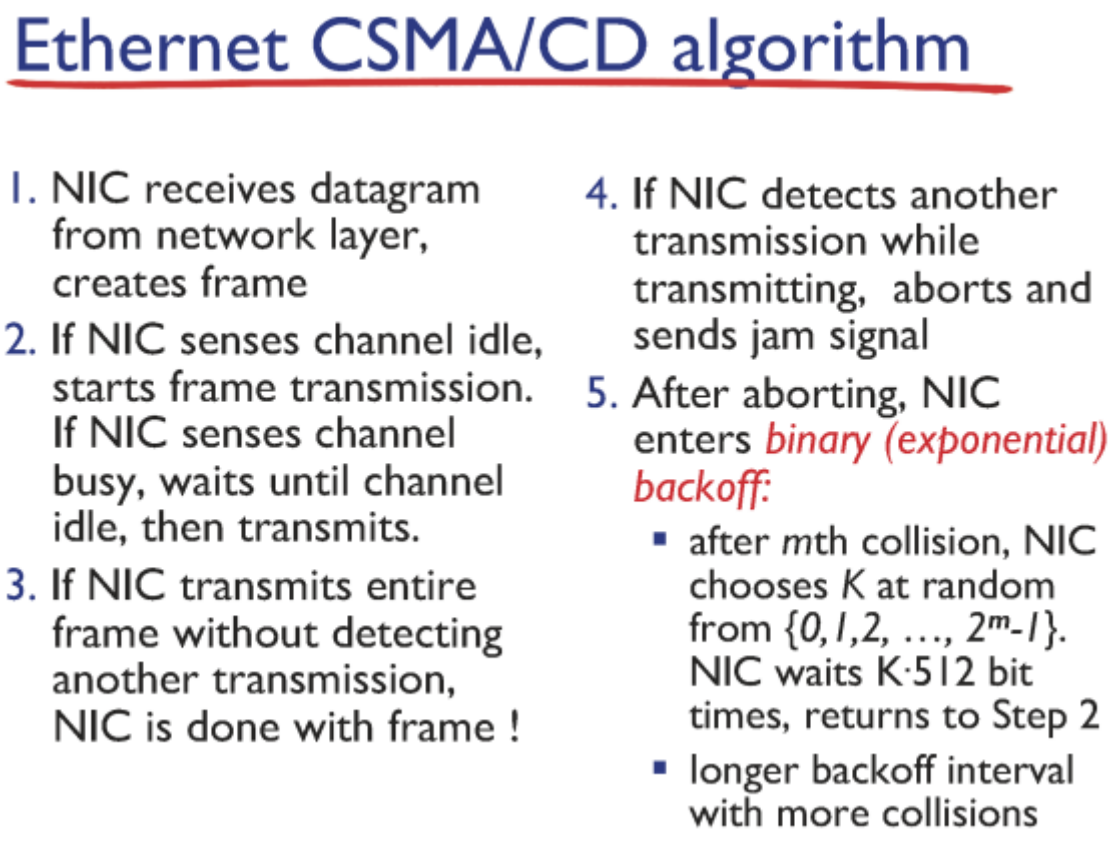
충돌이 발생하면 packet전송시간 낭비하게 됨

propagation delay가 발생하므로 충돌이 발생하게 됨

CSMA/CD (collision detection)

충돌을 감지하면 전송을 멈춤





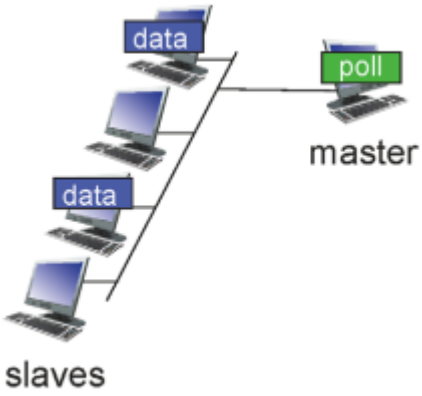
충돌을 감지하고 전송을 멈춘 후에는 binary(exponential) backoff방식을 사용해 기다림.

→ 충돌이 처음 일어났을 때는 조금 기다리지만 여러번 발생하면 오래 기다림 ⇒ backoff이 여러번 발생해서 사용자가 체감하는 대기시간이 길어짐.

서로 다른 숫자만큼 기다려야 충돌을 피할 수 있으니까.

“Taking turns” MAC protocols

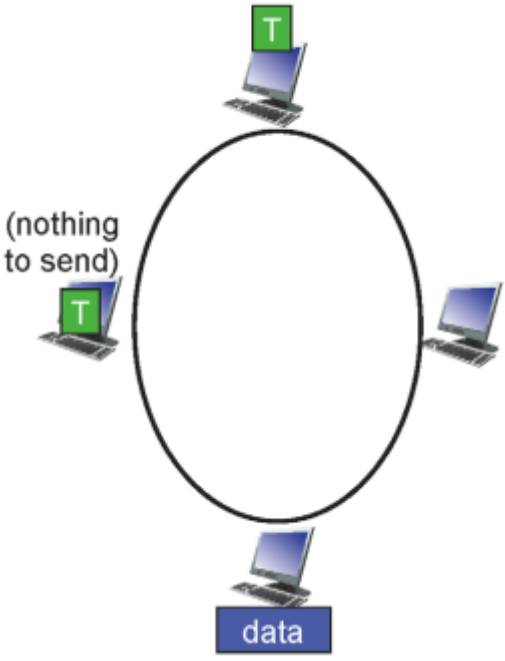
polling



master controller가 전송 순서를 정해줌

master가 작동을 멈추면 다같이 멈춤

token passing



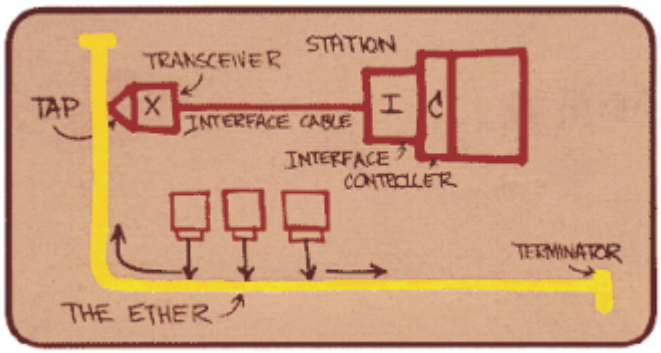
token을 가지고 있는 host만 전송할 수 있음

token을 잃어버리면 큰 문제 발생..

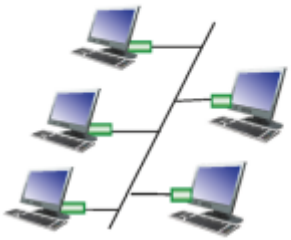
LANs

: router를 거치지 않고 연결할 수 있는 네트워크 집합

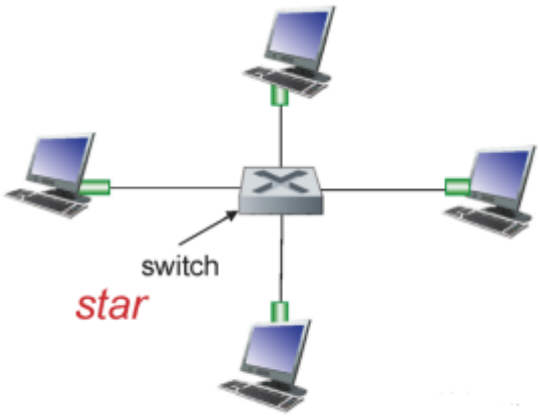
Ethernet



Metcalfe's Ethernet sketch



bus: coaxial cable



star

현재는 switch를 사용해 star방식을 사용함

Ethernet frame structure

CRC - error checking

data - 어떤 type의 protocol인지

Ethernet uses CSMA/CD

이더넷은 MAC protocol로 CSMA/CD 방식을 사용

- No slots
- adapter doesn't transmit if it senses that some other adapter is transmitting, that is, carrier sense
- transmitting adapter aborts when it senses that another adapter is transmitting, that is, collision detection
- Before attempting a retransmission, adapter waits a random time, that is, random access

collision이 발생하면 재전송, 아니면 재전송을 하지 않음

collision detection 을 못해서 충돌이 발생한 것을 모른다면 → 문제 발생..!

이더넷에서 충돌이 발생했는데, collision detecte를 하지 못한다면?

저 멀리서 일어난 collision detection 을 찾기 위해서 최소한의 비트길이 64bit가 정해짐

MAC addresses and ARP

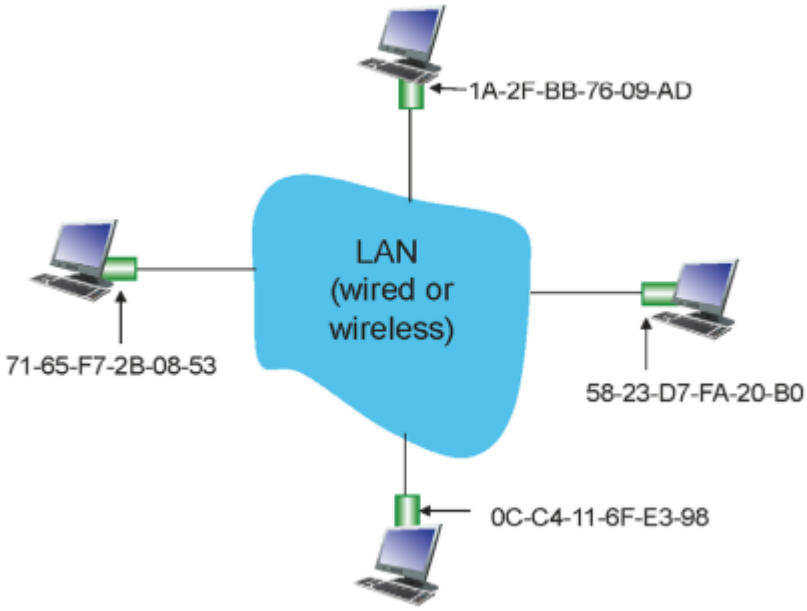
MAC address : 48bit MAC address, IP Address와 다름

이름 - host 이름

주소 - IP Address (변경 가능)

주민번호 - MAC address (변경 불가능)

LAN에 존재하는 각각의 adaptor는 고유의 LAN addresses를 가짐



frame에 Gateway의 MAC address를 써야하는데 모름.

→ IP Address는 알고 있으니까 그걸로 찾아내야함

ARP table

: IP 주소와 MAC 주소를 mapping 시켜둠

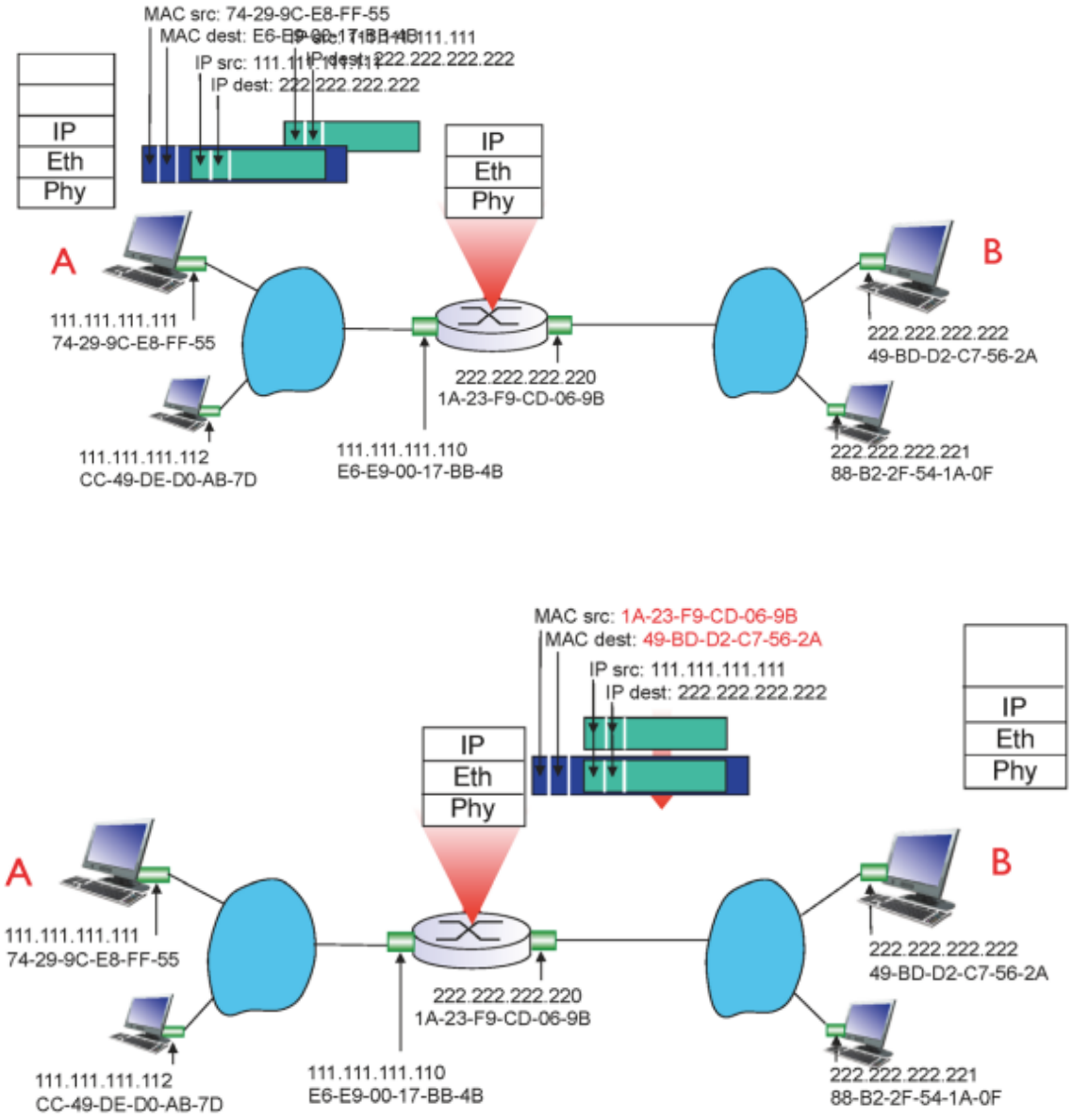
⇒ 캐시 테이블, 2시간 뒤면 사라짐

사라지면 ARP request를 통해 주소 받음

하나의 gateway여도 interface가 다르니까. 다른 MAC address를 가짐

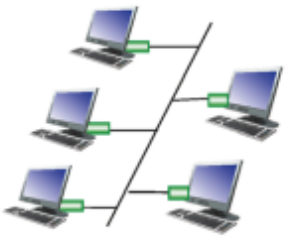
Addressing:routing to another LAN

walkthrough : send datagram from A to B via R

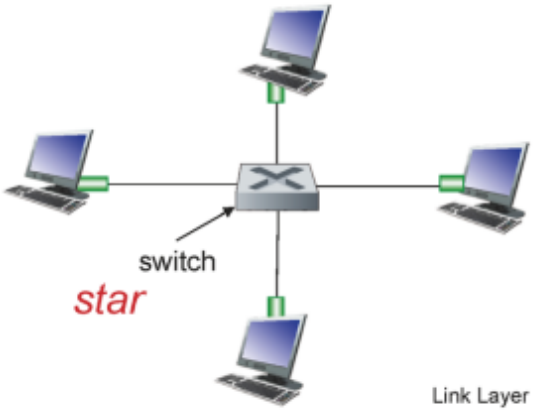


Ethernet : physical topology

- *bus:* popular through mid 90s
  - all nodes in same collision domain (can collide with each other)
- *star:* prevails today
  - active *switch* in center
  - each "spoke" runs a (separate) Ethernet protocol (nodes do not collide with each other)



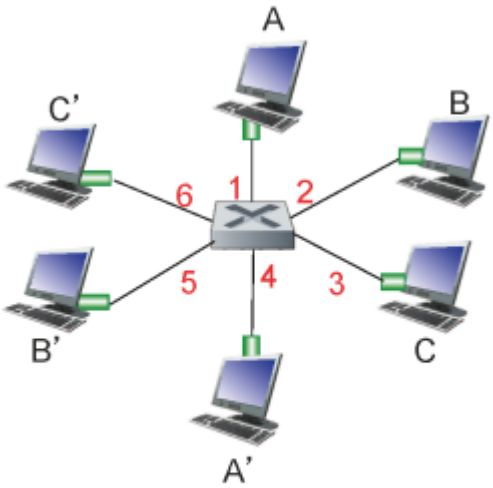
bus: coaxial cable



busd : 충돌이 많이 발생함

star : switch가 domain을 분리 시켜 충돌을 줄여줌

Switch: multiple simultaneous transmissions



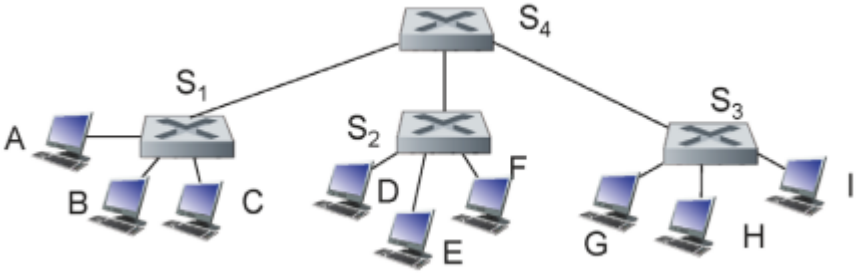
switch with six interfaces (1,2,3,4,5,6)

switch는 어떤 interface가 어떤 node와 연결되는지 어떻게 아는가? → switch table

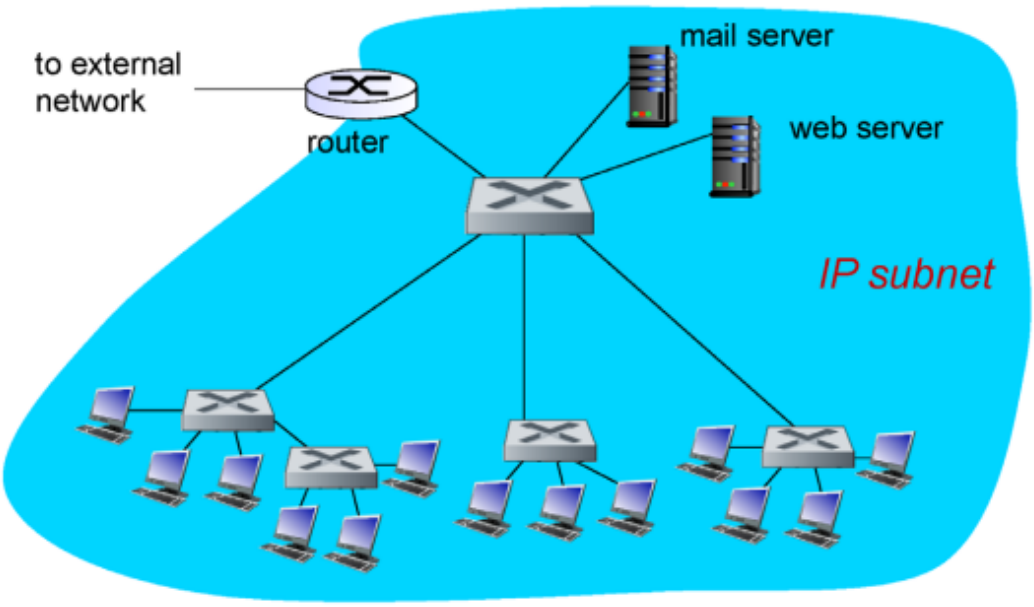
switching table은 self-learning을 통해 만들어감

(다 전송하고.. 그걸로 기록해 나가는..)

Self-learning multi-switch example



처음에 위치 찾을 때만 잉크 퍼져나가듯이 모두 탐색하고, 한번 탐색한 후에 전송할때는 table을 사용해서 바로바로 전송함



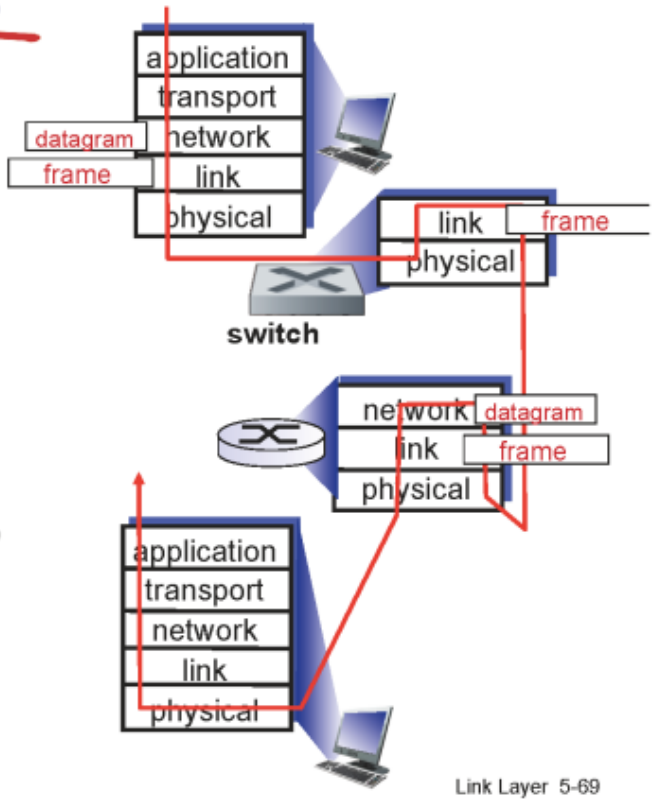
Switches vs. routers

both are store-and-forward:

- *routers:* network-layer devices (examine network-layer headers)
- *switches:* link-layer devices (examine link-layer headers)

both have forwarding tables:

- *routers:* compute tables using routing algorithms, IP addresses
- *switches:* learn forwarding table using flooding, learning, MAC addresses

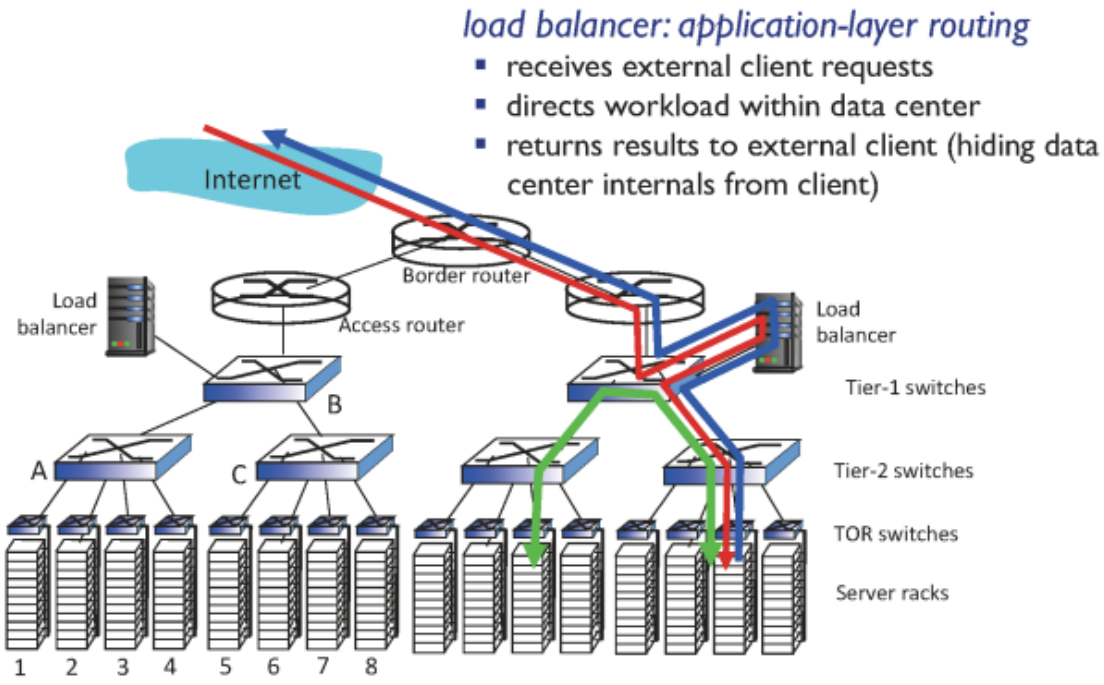


Link Layer 5-69

Data center networks

- 10's to 100's of thousands of hosts, often closely coupled, in close proximity:
  - e-business (e.g. Amazon)
  - content-servers (e.g., YouTube, Akamai, Apple, Microsoft)
  - search engines, data mining (e.g., Google)
- challenges:
  - multiple applications, each serving massive numbers of clients
  - managing/balancing load, avoiding processing, networking, data bottlenecks

Inside a 40-ft Microsoft container, Chicago data center



사용자가 많아질수록 계층화를 시키고, 그 계층화의 이음새 역할은 switch가 함