Computer Networks
@CS.NYCU

Lecture 6: Link Layer

Instructor: Kate Ching-Ju Lin (林靖茹)

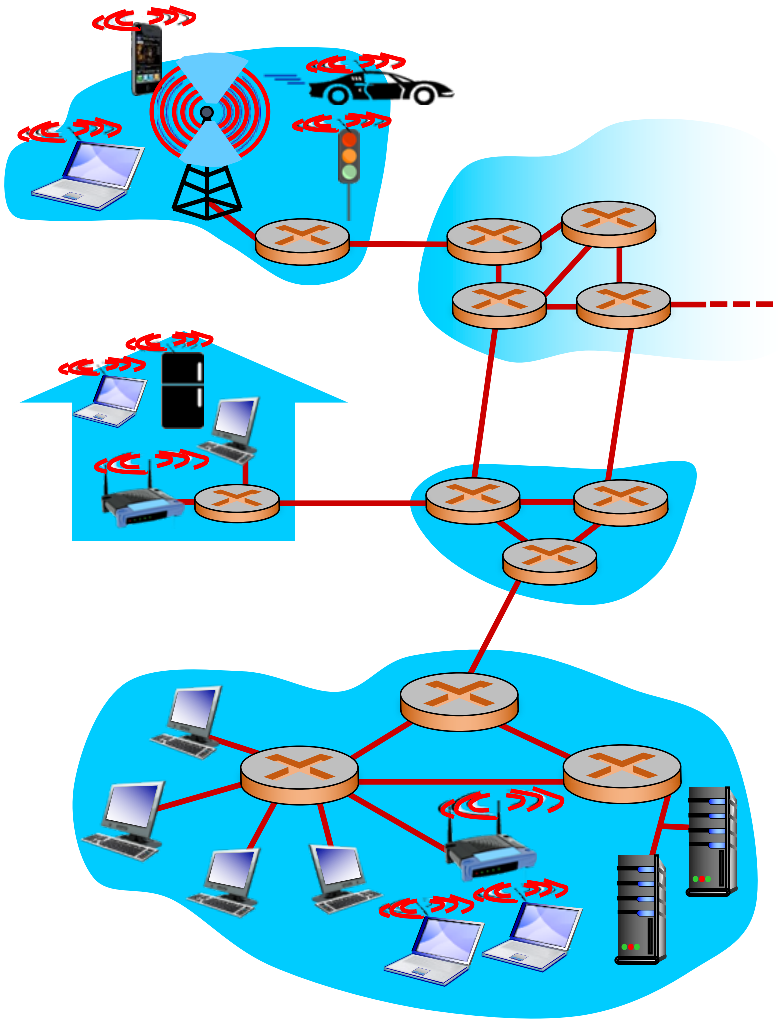
Slides modified from
"Computer Networking: A Top-Down Approach" 7th Edition

Outline

- Introduction to link layer
- Multiple access control
- LANs
  - ARP
  - Ethernet
  - Switches

Introduction to Link Layer

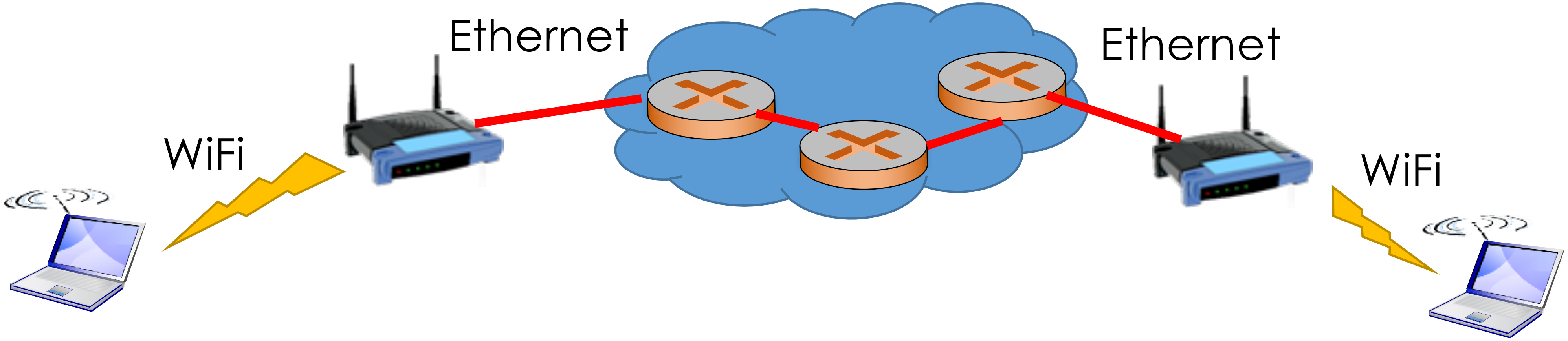
- **Nodes**
  - Hosts or routers
- **Links**
  - Communication channel that connects adjacent nodes
  - E.g., wires links, wireless links, LANs
- **Frames**
  - Layer-2 packet
  - Encapsulate datagram
- **Task**
  - Transfer datagram from one node to physically adjacent nodes over a link



Introduction to Link Layer

- Datagram transferred by different link protocols over different links
  - E.g., Ethernet links, 802.11 links

- Each link protocol provides different services
  - May or may not provide RDT (reliable data transfer)



Link Layer Services

- **Framing, link access**
  - Encapsulate datagram into frame
  - Channel access if the medium is shared by multiple nodes
  - MAC address used in frame headers

MAC addr	IP addr	TCP header	Data payload

- **Reliable delivery**
  - ACK and retransmission
  - Seldom used on low bit-error link (fiber, some twisted pair)
  - Used in high error links, e.g., wireless links
    - Q: why both link-level and end-end reliability?

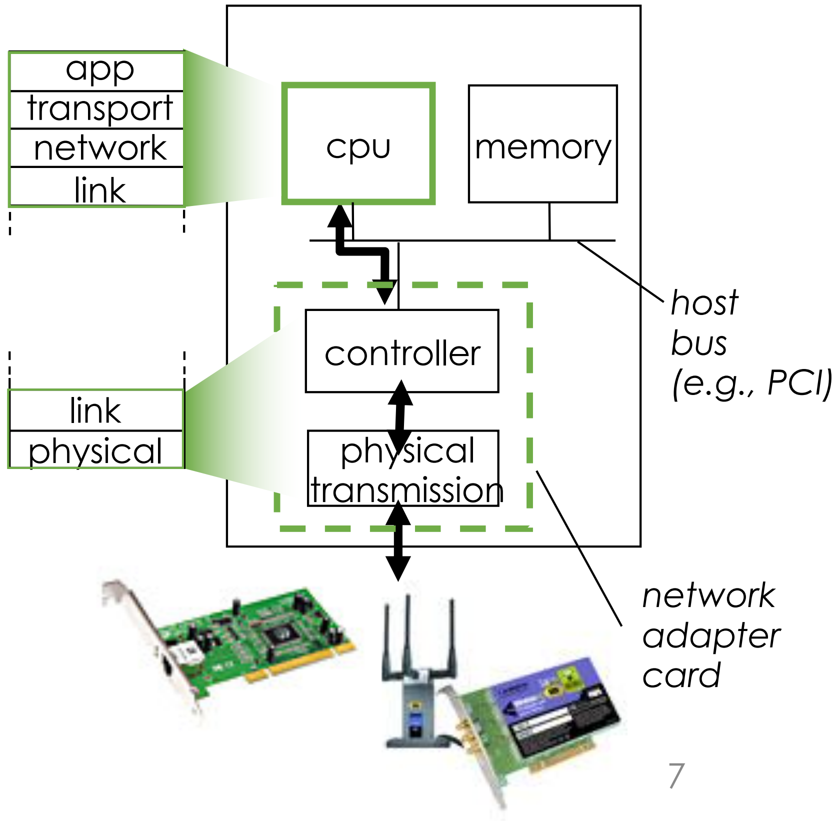
Link Layer Services

- **Flow control**
  - Between adjacent sending and receiving nodes
- **Error detection**
  - Errors caused by signal attenuation or noise
  - Detecting error at receiver
  - Trigger transmitter to retransmit or drop frames
- **Error correction**
  - Detecting and correct bit errors at receiver, without retransmission from sender
- **Half-duplex or full-duplex**
  - Half-duplex: node can either transmit or receive, not at the same time

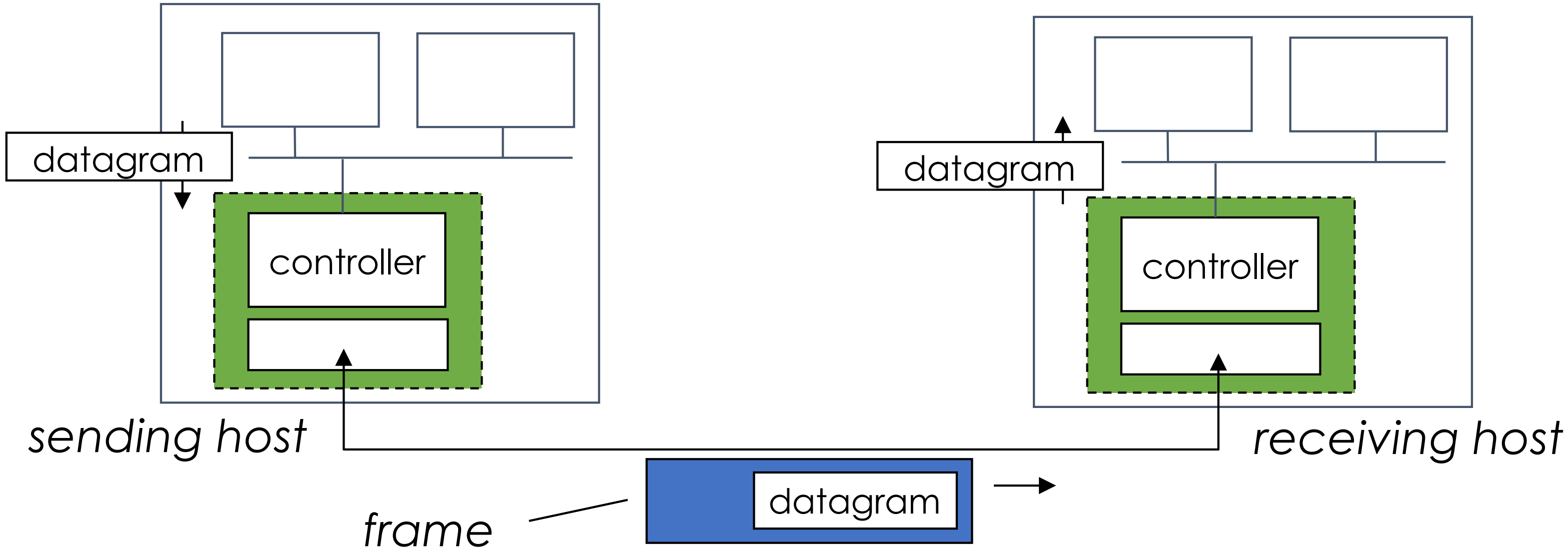
Where is Link Layer Implemented

- **Hardware**
  - Implemented on a chip or on adapter (network interface card, NIC)
  - E.g., Ethernet NIC, WiFi NIC
  - Attached into host's system buses

- **Software**
  - Implemented in drivers
  - Handel errors, push frames to upper layers
- combination of hardware, software, firmware



Adaptor Communication



- 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

Outline

- Introduction to link layer
- **Multiple access control**
- LANs
  - ARP
  - Ethernet
  - Switches

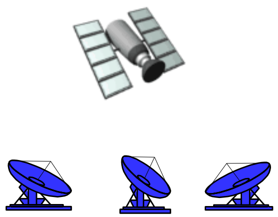
Multiple Access Links, Protocols

Two types of links

- **Point-to-point**
  - PPP for dial-up access
  - Point-to-point link between Ethernet switch, host
- **Broadcast (shared medium)**
  - Old-fashion Ethernet
  - Upstream HFC
  - 802.11 wireless LAN

shared wire (e.g., cabled Ethernet)

shared RF (e.g., 802.11 WiFi)



shared RF (satellite)

humans at a cocktail party

Broadcast Medium

Difference between TV broadcasting and LAN

- **TV broadcasting**
  - One-way broadcast: one-to-many (single sender)
  - Non-reliable
- **LAN**
  - Any two nodes would communicate with each other
  - Nodes may talk at the same time
  - How to negotiate?
    1. Give everyone a chance to talk
    2. Don't speak until you are spoken to
    3. Don't monopolize the conversation
    4. Raise your hand if you have a quest
    5. Don't interrupt when someone is talking

Multiple Access Protocols

- Single shared broadcast channel

- Two or more simultaneous transmissions by nodes: interference
  - Collision if a node receiver two or more frames the same time

- **Multiple access protocols**
  - Distributed (decentralized):
    → nodes make their decision by their own
  - No out-of-band channel for coordination:
    → data frames and control frames exchanged on the same channel

Ideal Multiple Access Protocols

- Given: broadcast channel of rate ***R*** bps
- Goal:
  1. When ***one*** node wants to transmit, it can send at rate ***R***
  2. When ***M*** nodes want to transmit, each can send at average rate *R/M*
  3. Fully decentralized:
     - no special node to coordinate transmissions
     - no synchronization of clocks, slots
  4. Simple

Types of MAC Protocols

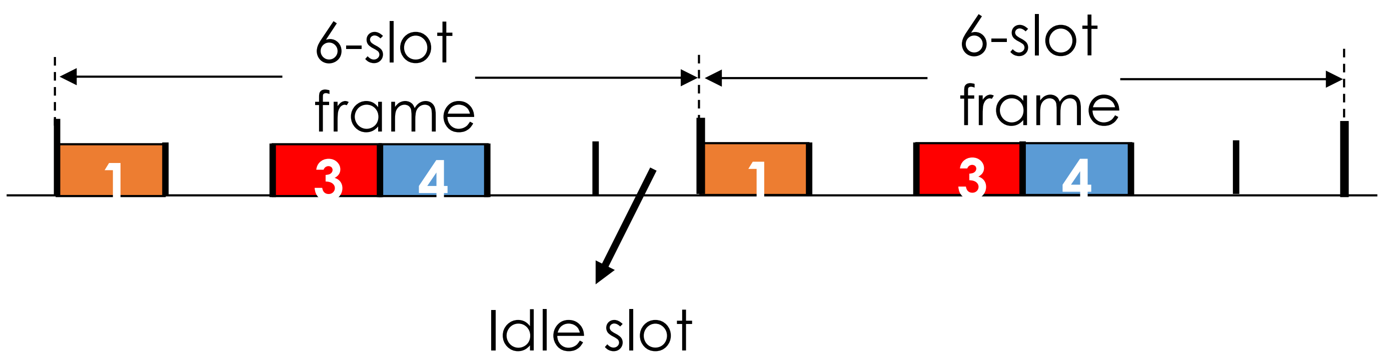
- **MAC: Medium Access Control**
- **Channel partitioning**
  - Divide channel into smaller "pieces"
  - Allocate piece to node for exclusive use
- **Random access**
  - Channel not divided → allow collisions
  - Should resolve collisions
- **Taking turns**
  - Nodes take turns
  - Nodes with more traffic take longer turns

Channel Partitioning

- Partition Channel by **time, frequency or codes**

TDMA: time-division multiple access

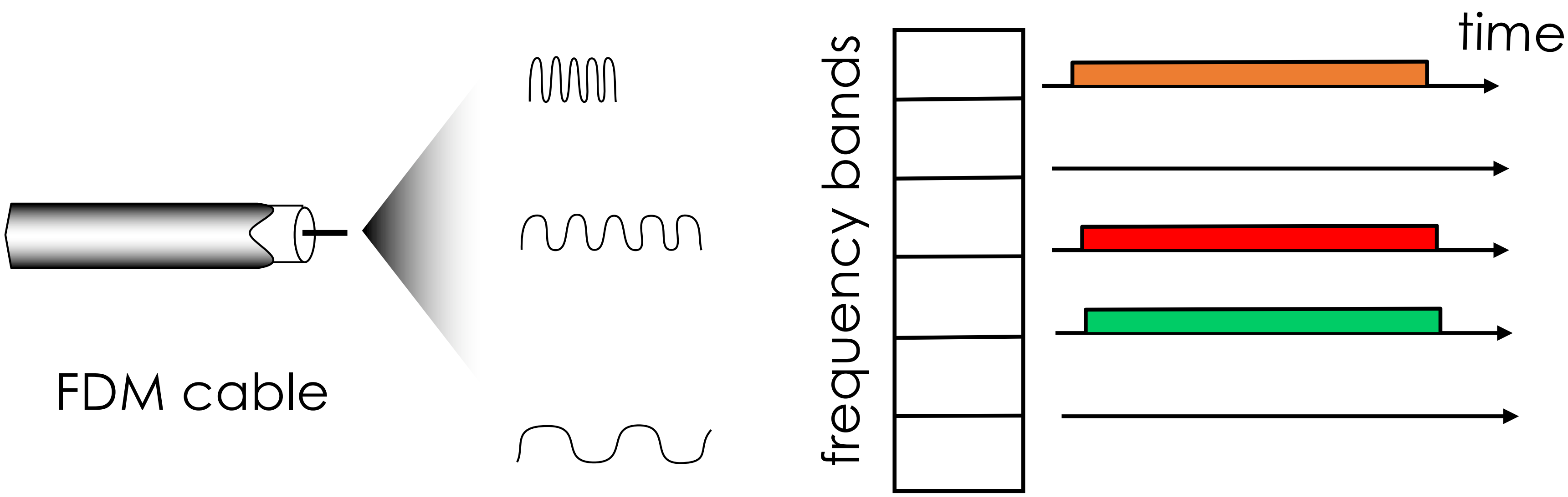
- Channel time divided into time slots
- Access to channel in "rounds"
- Each node gets fixed length slot in each round
- Unused slots go idle
- Example: 6 nodes, nodes 2, 5, 6 have no traffic



FDMA

Frequency-Division Multiple Access

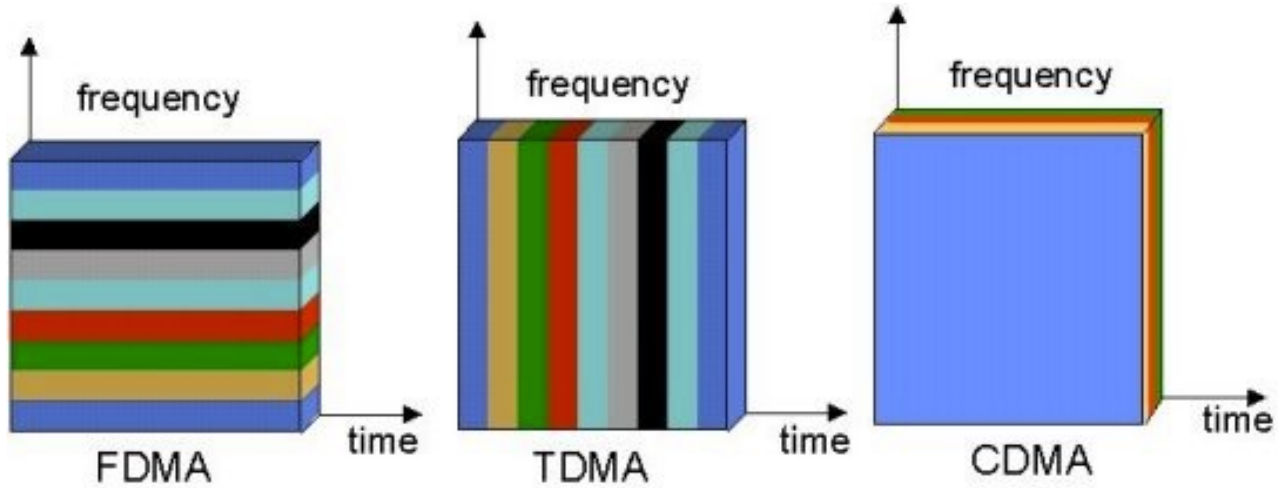
- channel spectrum divided into frequency bands
- Each node assigned a fixed band
- Unused frequency bands go idle
- Example: 6 nodes, nodes 2, 5, 6 have no traffic



CDMA

Code Division Multiple Access

- Find multiple codes that can be separated despite collisions
- Assign a difference code to each node
- Each sender uses its code to encode data bits
- Multiple senders transmit simultaneously
- Each receiver gets a collided frame, but can extract its own frame using the assigned code



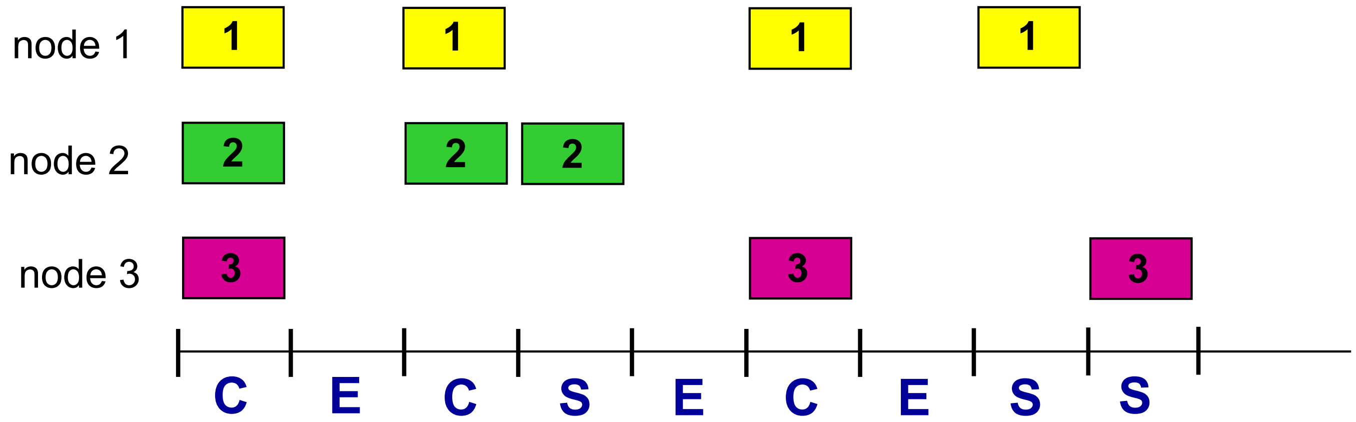
Types of MAC Protocols

- **MAC: Medium Access Control**
- **Channel partitioning**
  - Divide channel into smaller "pieces"
  - Allocate piece to node for exclusive use
- **Random access**
  - Channel not divided → allow collisions
  - Should resolve collisions
- **Taking turns**
  - Nodes take turns
  - Nodes with more traffic take longer turns

Random Access

- When node has packet to send
  - Transmit at full channel data rate R
  - no a priori coordination among nodes
- Two or more transmitting nodes → **Collisions**
- **Random access protocol** specifies
  - How to detect collisions
  - How to recover from collisions
- Examples of random access protocols
  - Slotted ALOHA
  - ALOHA
  - CSMA, CSMA/CD, CSMA/CA

Slotted ALOHA



- Assumptions
  - All frames are of equal size
  - Time divided into equal size slots (1 slot 1 frame)
  - Node start transmitting in the beginning of a slot
  - Nodes are synchronized
- Protocol
  - If a node has a frame, transmit in the next slot
  - If collisions, retransmit in the following slots with probability p until success

Slotted ALOHA

- **Pros**
  - Single active node can continuously transmit at full rate of channel
  - Highly decentralized (but need synchronization)
  - Simple
- **Cons**
  - Collisions → waste slot
  - Idle slots
  - Clock synchronization

Slotted ALOHA

- **Efficiency**
  - long-run fraction of successful slots
- Assumption
  - N nodes with frames to send, each transmits in a slot with probability p
- Probability that <u>a give node</u> succeeds in a slot

$$p(1-p)^{N-1}$$

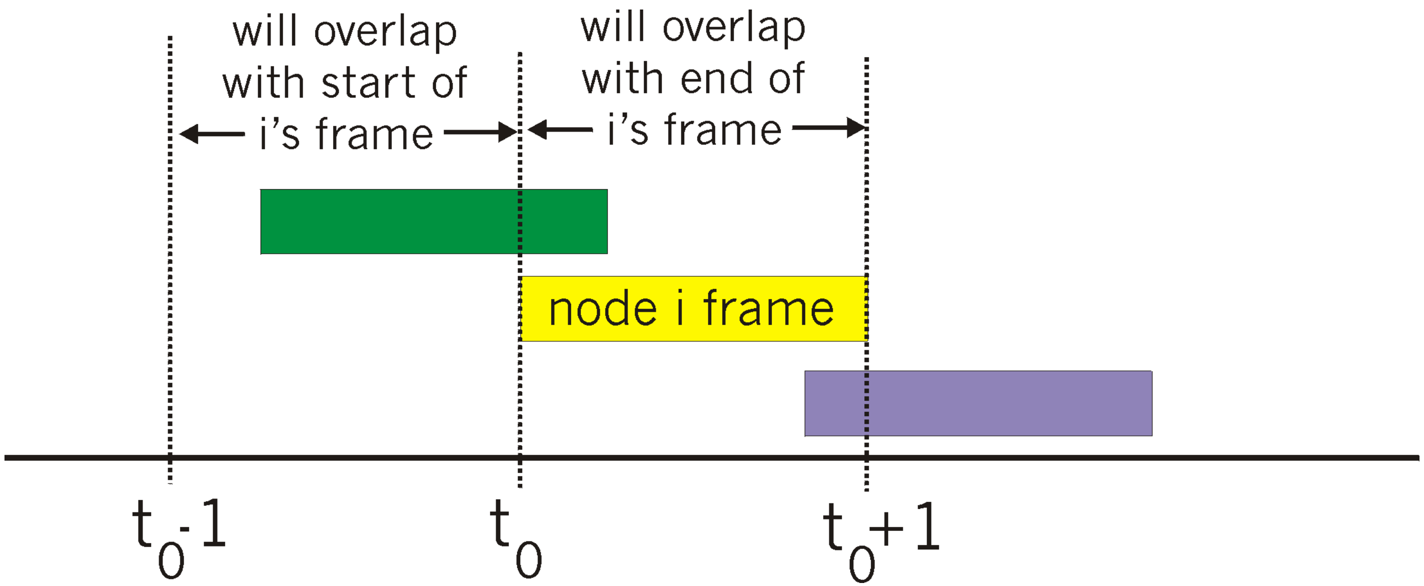
- Probability that <u>any node</u> succeeds in a slot

$$Np(1-p)^{N-1}$$

- If N → infinity, efficiency is only 1/e = 0.37
  - Channel utilization is only 37%!!

ALOHA

- Unslotted ALOHA
  - Simpler, no synchronization
- When a frame arrives, transmit immediately
- Collision probability increases (as compared to slotted ALOHA)
  - Any part of a frame can be collided



ALOHA Efficiency

- **Efficiency**
  - Assume that a collision can be in the first part or/and in the second part

P[success by a given node] =
    P[node transmit] *
    P[no other node sends during [t_0 – I] *
    P[no other node sends during [t_0 + I]
$= p * (1-p)^{N-1} * (1-p)^{N-1}$
$= p * (1-p)^{2(N-1)}$

If N → infinity, efficiency is about 1/2e = 18%

Worse than slotted ALOHA

CSMA

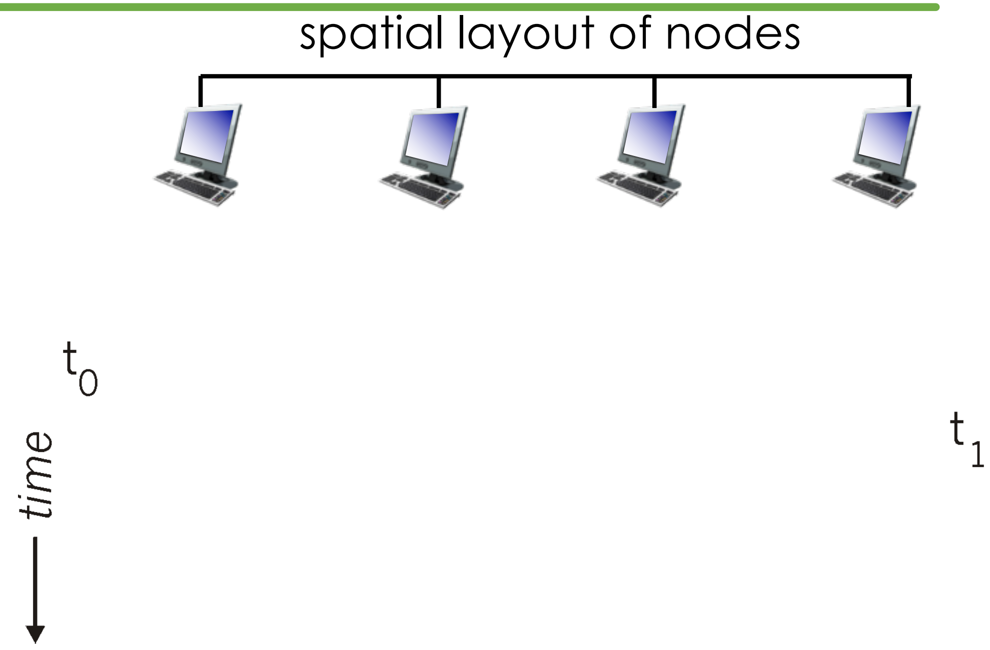
- **CSMA: Carrier Sense Multiple Access**

- **Listen before talk**!
  - If channel sensed idle, transmit an entire frame
  - If channel sensed busy, defer transmission
  - "Try to" avoid collisions (not guarantee!)

- Human analogy: do no interrupt others!
  - Avoid interfere/collide others

CSMA Collisions

- Collisions can still occur
  - Why? Propagation delay!
  - Sensed idle, but some other node just starts its transmission
- When a collision occurs
  - Entire frame transmission time wasted
  - Distance (propagation delay) determines the collision probability



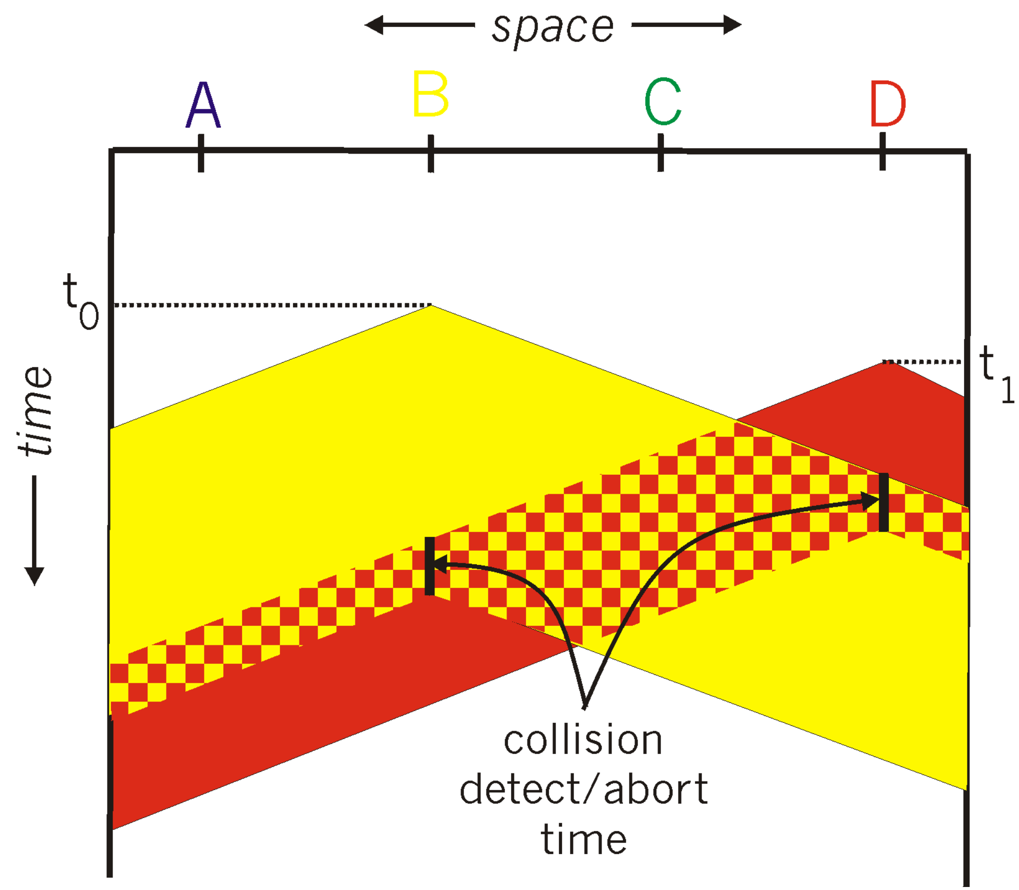
CSMA/CD (Collision Detection)

- **Carrier sensing + collision detection**
  - Detect collision as sending a frame
  - Abort transmission as a collision is detected
  - Reduce channel wastage!

- Properties
  - Easy to implement in wired LANs: measure signal strengths, compare transmitted, receive signals
    → Why? Need full-duplex capability

  - Difficult in wireless LANs
    → why: wireless is usually half-duplex. Cannot transmit and receive at the same time

CSMA/CD (Collision Detection)



Ethernet CSMA/CD Algorithm

1. If NIC senses channel idle, start frame transmission. Otherwise, wait until channel idle
2. If NIC transmits the entire frame without detecting any collisions, done!
3. If NIC detects any collision, abort immediately
4. If a collision detected, enter binary (exponential) backoff
   - After *m*-th collision, NIC chooses *K* at random from $\{0,1,2, \ldots, 2^{m-1}\}$ → more collisions, longer waiting time
   - Wait *K* * 512 bit time, go back to step 2

Types of MAC Protocols

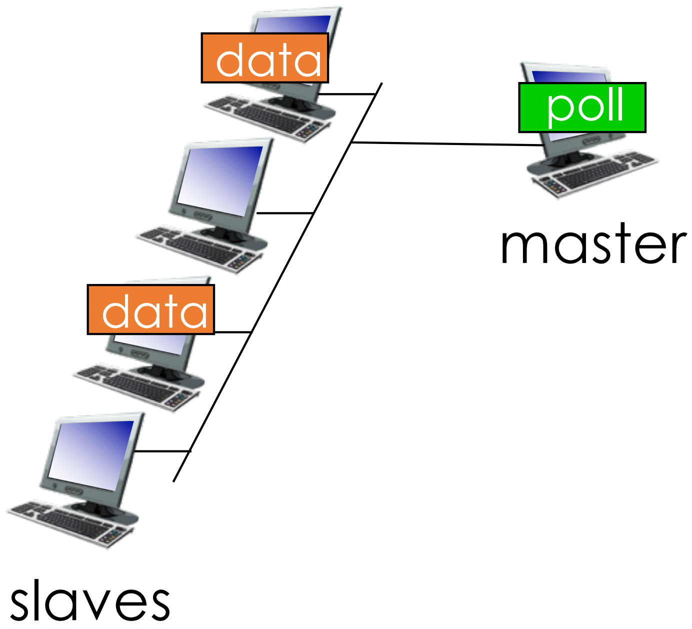
- MAC: Medium Access Control
- Channel partitioning
  - Divide channel into smaller "pieces"
  - Allocate piece to node for exclusive use
- Random access
  - Channel not divided → allow collisions
  - Should resolve collisions
- Taking turns
  - Nodes take turns
  - Nodes with more traffic take longer turns

Taking Turns MAC

- **Channel partitioning**
  - Share channel efficiently and fairly at high load
  - Inefficient at low load (if some nodes are assigned resource but not transmitting)
- **Random access**
  - Efficient at low load: single node fully utilized the whole channel
  - Collisions at high load
- **Taking turns**
  - Look for the best of both worlds!

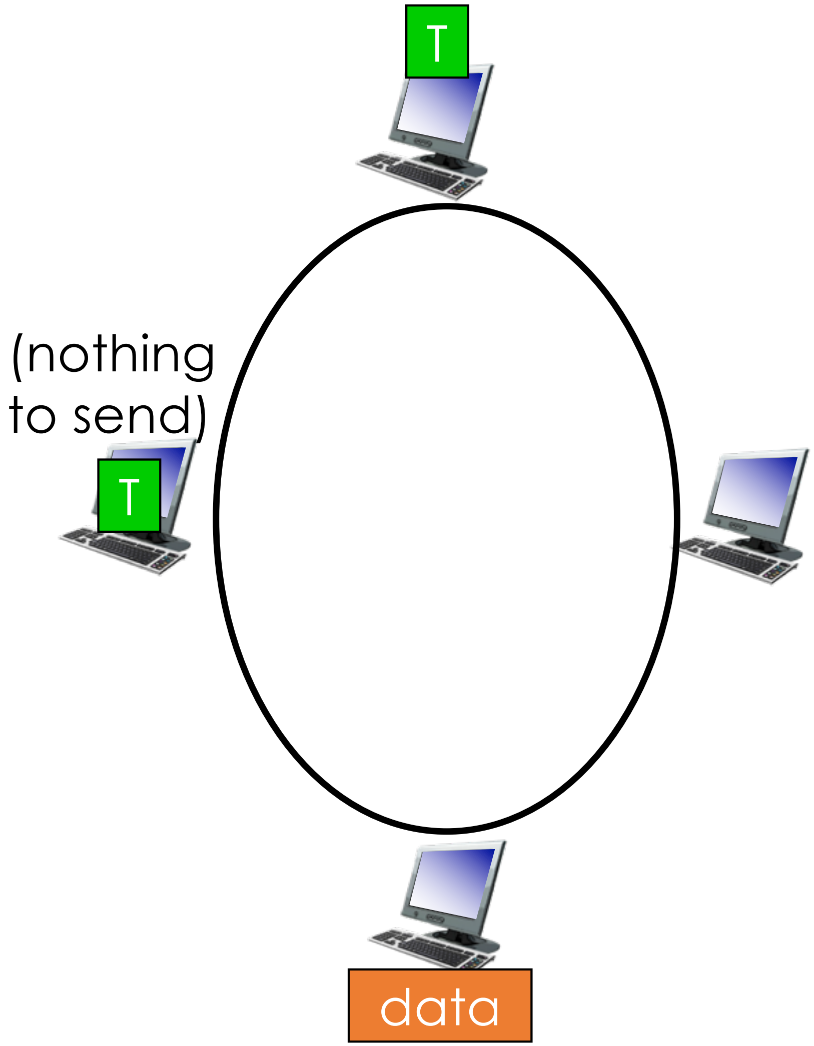
Taking Turns: Polling

- master node "invites" slave nodes to transmit in turn
- typically used with "dumb" slave devices
- concerns:
  - polling overhead
  - latency
  - single point of failure (master)



Taking Turns: Token Passing

- Control token passed from one node to next sequentially
- Token message
- Concerns:
  - token overhead
  - Latency
  - Single point of failure (token)



Summary of MAC Protocols

- **Channel partitioning**
  - 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

Outline

- Introduction to link layer
- Error detection/correction
- Multiple access control
- LANs
    - **ARP**
    - Ethernet
    - Switches

MAC Address

- 32-bit IP address
  - A.B.C.D
  - Network-layer (layer-3) address for each interface

Like your home address, hierarchical

- 48-bit MAC address
  - E.g., 1A-2F-BB-76-09-AD
    - Typically written in hexadecimal (base 16)
  - Burned in NIC (network interface)
    - Cannot be modified in old interfaces
    - Now can sometimes be modified by software

Like your personal ID, flat

MAC Address

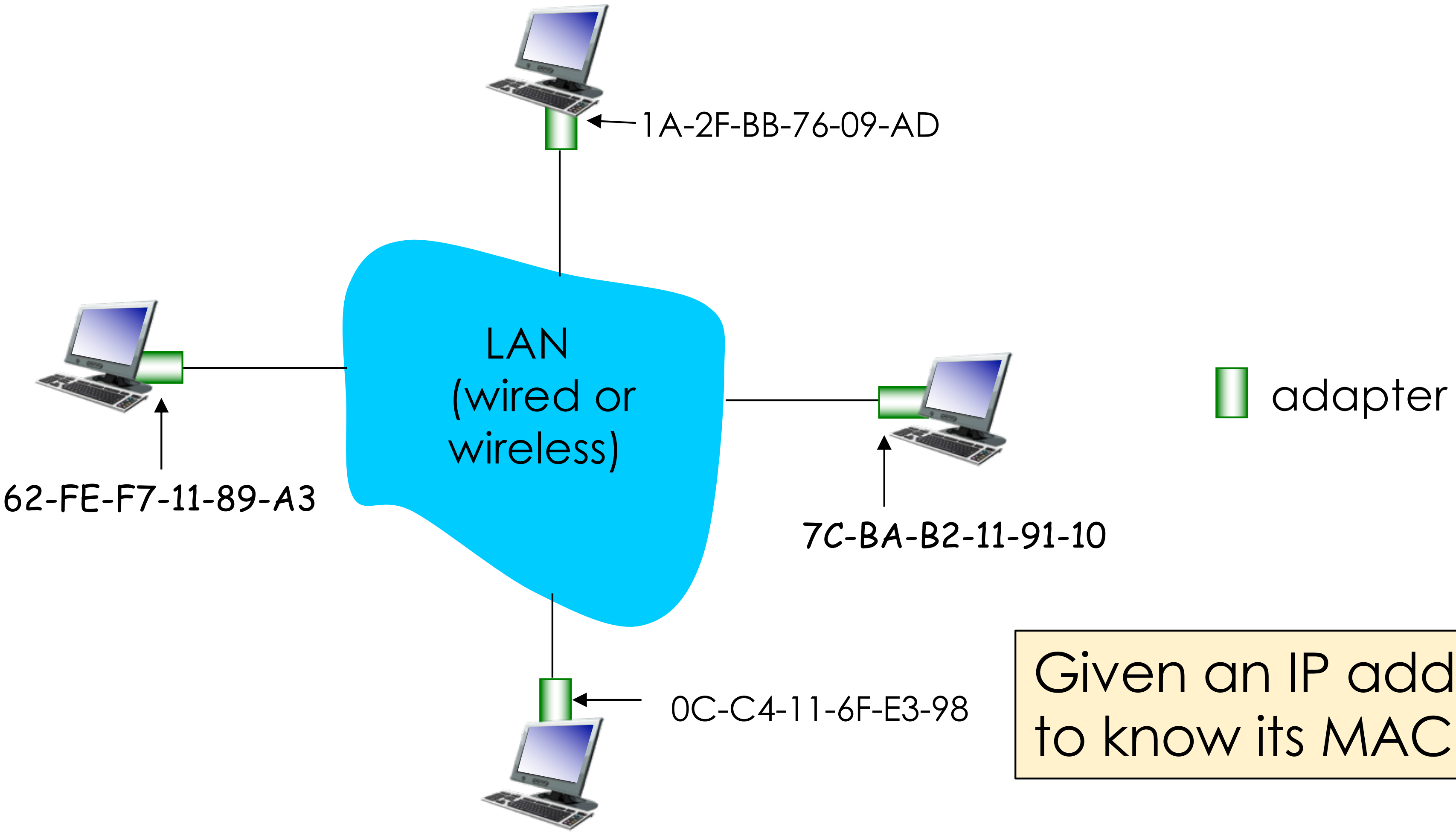
- Each interface with an unique MAC addr
    - Assigned by IEEE

- Used to locally get frames from one interface to another physically-connected interface (in the same sub-net)

- MAC broadcast address: FF-FF-FF-FF-FF-FF
    - All the hosts in a LAN receive the frame
    - IP broadcast address: 255.255.255.255

MAC Address and ARP

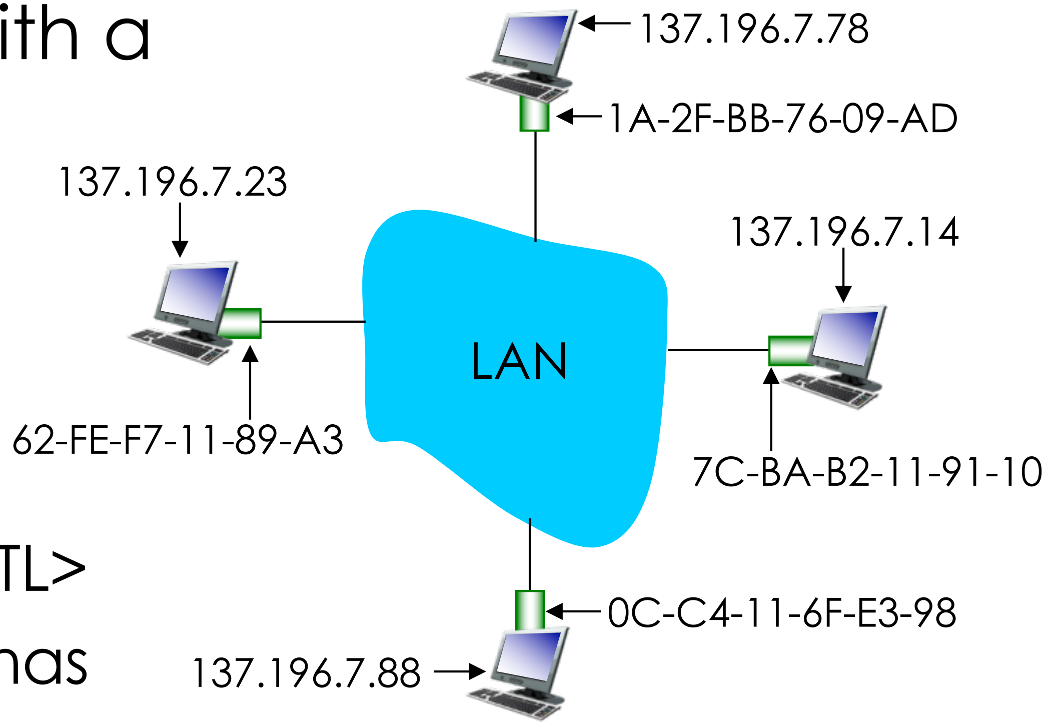
each adapter on LAN has unique *MAC (LAN)* address



Given an IP addr, how to know its MAC addr?

ARP: Address Resolution Protocol

- Discover the link layer address associated with a given IP address
  - RFC 826, 1982
- ARP table
  - Each IP in the same subnet has an entry <IP addr, MAC addr, TTL>
  - Each router interface has an ARP module
  - Similar to DNS

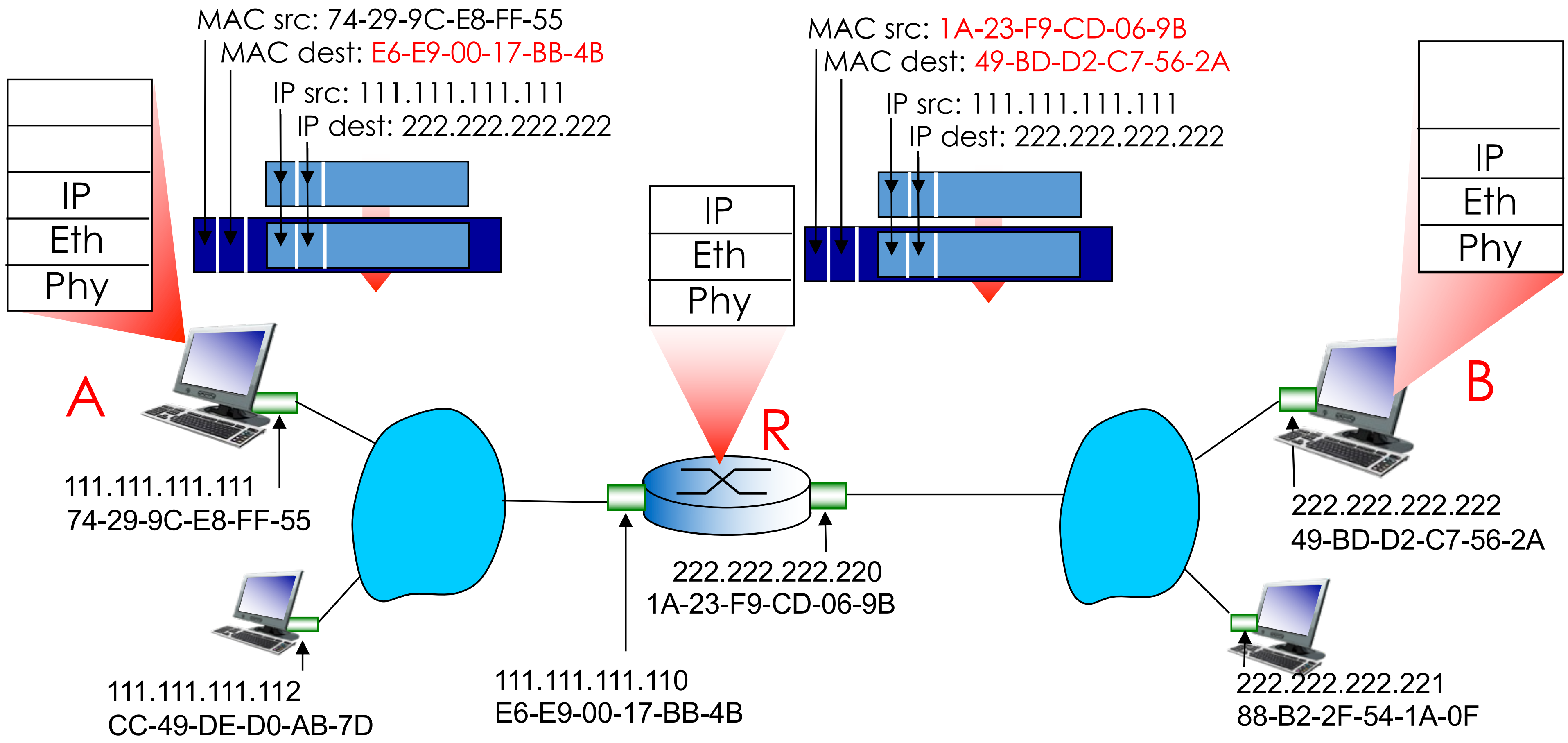


ARP: Same LAN

- A wants to send datagram to B
  - B's MAC addr not in A's ARP table.
- A broadcasts ARP query packet, containing B's IP address
  - Destination MAC addr = FF-FF-FF-FF-FF-FF
  - All nodes on LAN receive ARP query
- B receives ARP packet, replies to A with its (B's) MAC addr
  - Frame sent to A's MAC addr (unicast)
- A caches (saves) IP-to-MAC addr pair in its ARP table until information becomes times out
- "plug-and-play"
  - nodes create their ARP tables *without intervention from net administrator*

ARP: Different LANs

- Example: two subnets interconnected by a router
  → A knows B's IP, A wants to send to B

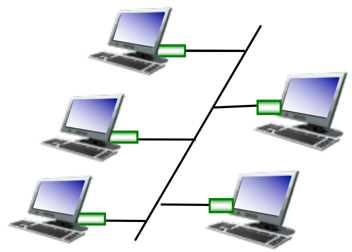


Outline

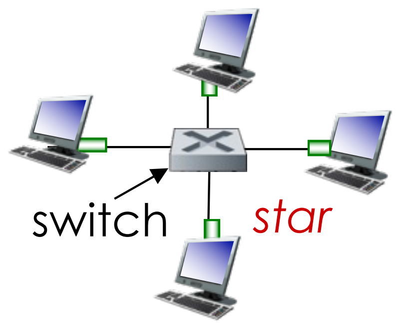
- Introduction to link layer
- Error detection/correction
- Multiple access control
- LANs
  - ARP
  - **Ethernet**
  - Switches

Ethernet

- Dominant wired LAN protocol
  - Single chip, multiple speeds
  - First widely used LAN technology
  - Simple, cheap
  - Increasing speed: 10Mbps to 10Gbps
  - Other LAN technologies: FFFI, ATM ☹
- Old Ethernet design
  - 1970: bus topology (coaxial bus)
    - All host connect to a broadcast LAN → can collide with each other
  - 1990: hub-based star topology
    - Also a broadcast LAN, two frame coming from different interfaces would collide



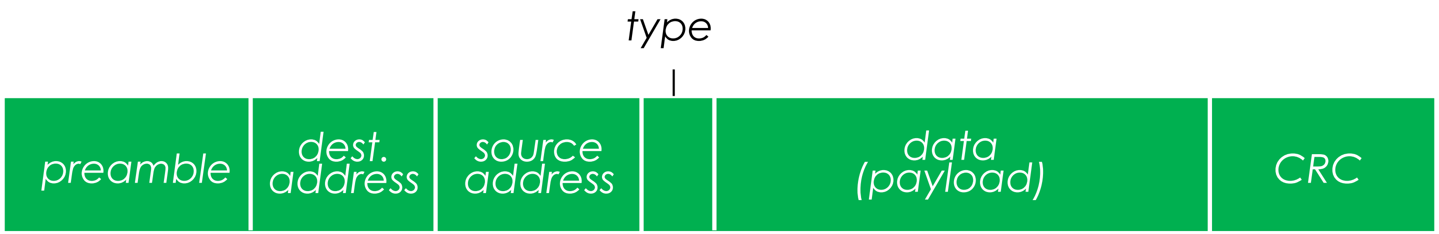
bus: coaxial cable



Now: switch!

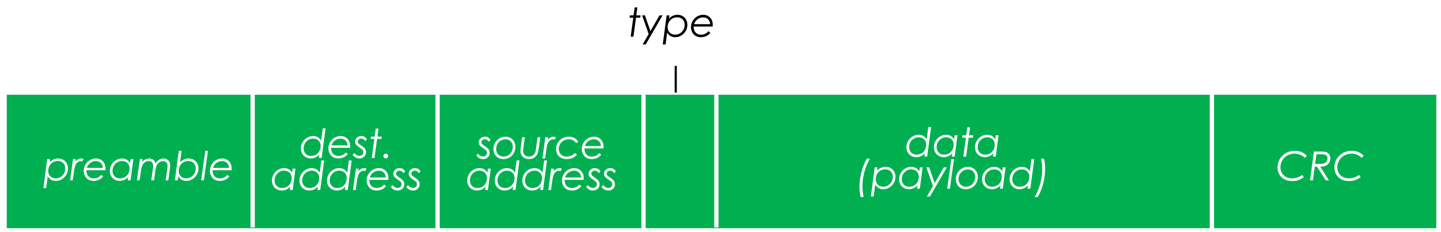
Ethernet Frame Structure

- Sending adapter encapsulates IP datagram in Ethernet frame



- **Preamble**:
  - 7 bytes with pattern 10101010 followed by one byte with pattern 10101011
  - Used to synchronize receiver and sender clock rates

Ethernet Frame Structure



- **Addresses**: 6 byte source, destination MAC addresses
  - If adapter receives frame with matching destination address (or broadcast address), pass frame to network layer
  - otherwise, adapter discards frame
- **Type**: indicates higher layer protocol
  - Mostly IP, or others, e.g., Novell IPX, AppleTalk
- **CRC**: cyclic redundancy check
  - error detected: frame is dropped

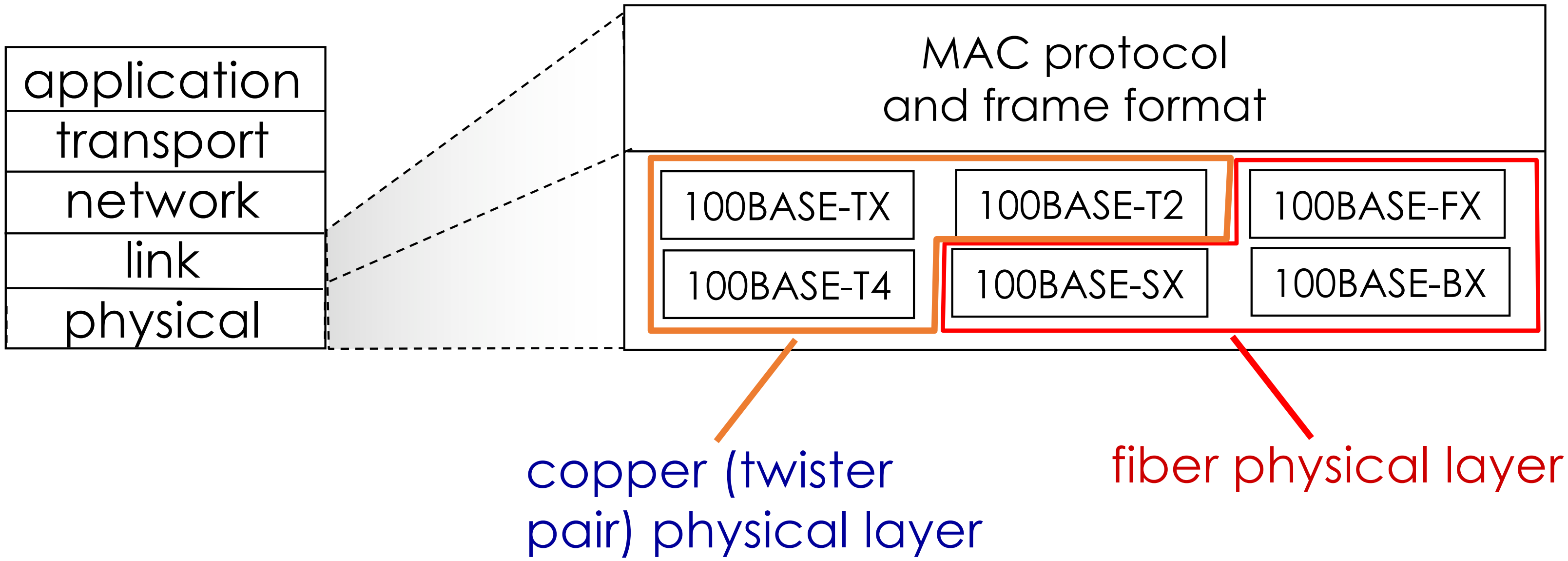
Ethernet Properties

- **Connectionless**
  - No handshaking between sender and receiver
- **Unreliable**
  - Receiving NIC does not feedback ACK or NACK
  - Dropped frames recovered only by higher layers (e.g., TCP)

- Ethernet's MAC protocol
  - Unslotted CSMA/CD with binary backoff

Ethernet Standard: 802.3

- Many different Ethernet standards
  - Define link and physical layers
  - Common MAC protocol and frame format
  - Different speeds: 2 Mbps, 10 Mbps, 100 Mbps, 1Gbps, 10 Gbps, 40 Gbps
  - Different physical layer media: fiber, cable



Outline

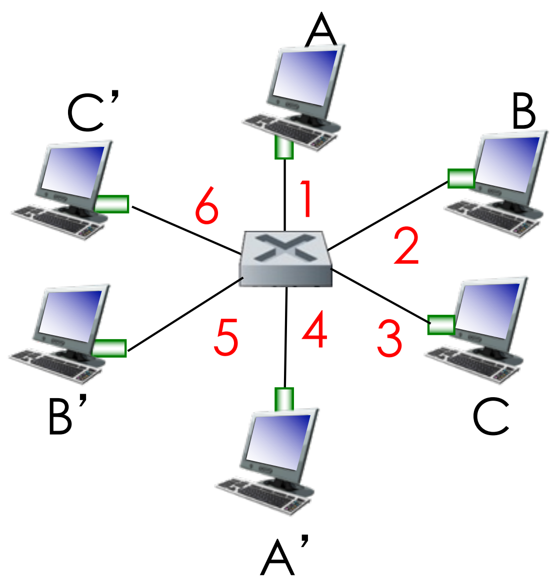
- Introduction to link layer
- Error detection/correction
- Multiple access control
- LANs
  - ARP
  - Ethernet
  - **Switches**

Ethernet Switch

- Link layer devices
  - Store and forward Ethernet frames
  - Examine incoming frames' MAC address
  - Forward frame to outgoing links
  - CSMA/CD access
- **Transparent**
  - Hosts are unaware of the presence of switches
  - Don't know whether a switch actually forward frame
- **Plug-and-play**
  - Self-learning! Table built automatically
  - Switches do not need to be configured

Switch: Multiple Access

- Hosts have direct connection to switch
- Switches buffer packets
- Ethernet used on each link, but not collisions
  - Each link is its own collision domain
- **Full-duplex**
  - Send and receive at the same time
- Collision-free
  - A-to-A' and B-to-B' can transmit simultaneously



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

Switch: Forwarding and Filtering

- **Filtering**
  - Switches determine whether a frame should be forwarded to some interface or dropped
- **Forwarding**
  - Switches determine the interfaces to which a frame should be forwarded
- **Switch table**
  - 1)MAC address, 2) switch interface connected to that MAC address, 3) time at which the entry is inserted

Address	Interface	Time
62-FE-F7-11-89-A3	1	9:32
7C-BA-B2-B4-91-10	3	9:36
....	....	....

Switch: Forwarding and Filtering

- Say frame coming from interface x to the destination address is DD-DD-DD-DD-DD-DD

- Three possible forwarding cases

  1. If no MAC address in the table, forward frame to all interfaces, except for x

  2. If there is an entry in the table of x (i.e., in = out), no need to forward

  3. If there is an entry in the table of *y* (i.e., in != out), forward frame to interface *y*

Switch: Self-Learning

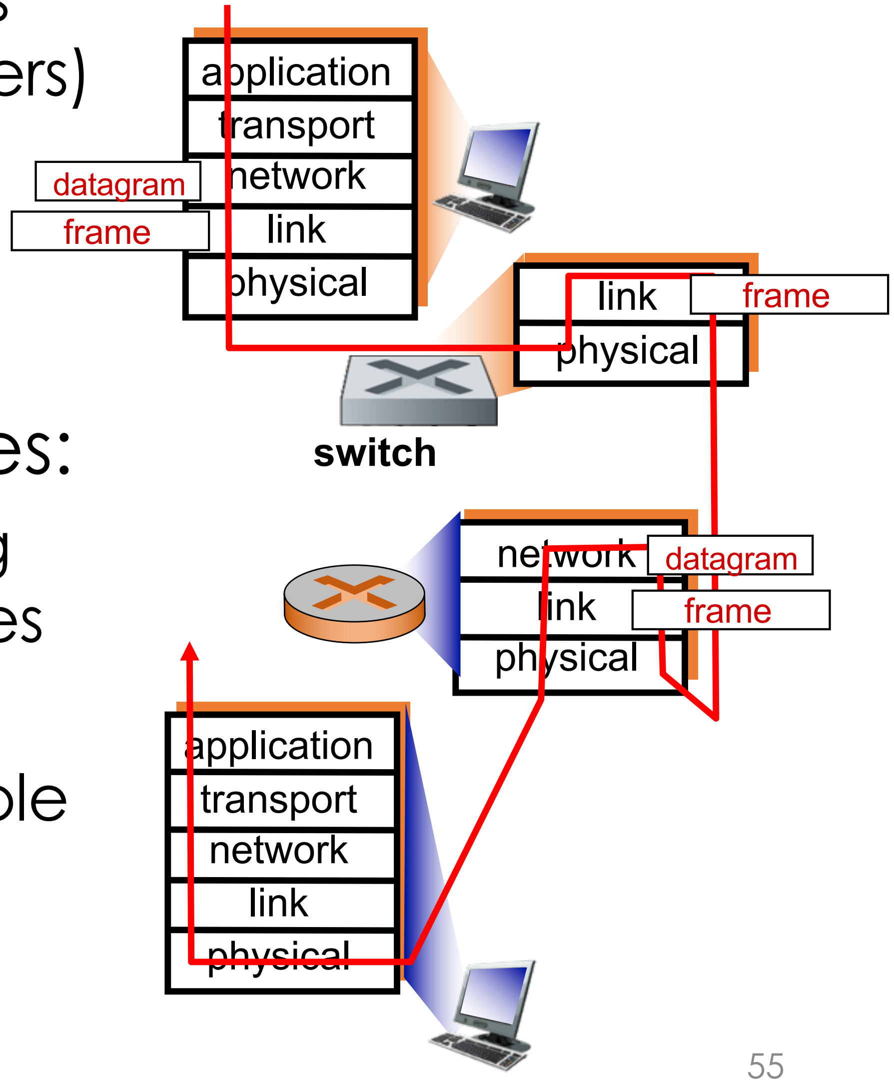
- Switching table is built automatically, dynamically and autonomously
- When frame received, record sender/location pair in switch table
  - Automatically "learn" MAC of sender
  - Interface where the frame arrives
  - Current time

Address	Interface	Time
62-FE-F7-11-89-A3	1	9:32
7C-BA-B2-B4-91-10	3	9:36
....	....	....

- Administrators don't need to configure the switch table

Switch vs. Router

- 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
    - Suitable for large networks
  - Switches: learn forwarding table via self-learning
    - Suitable for small networks