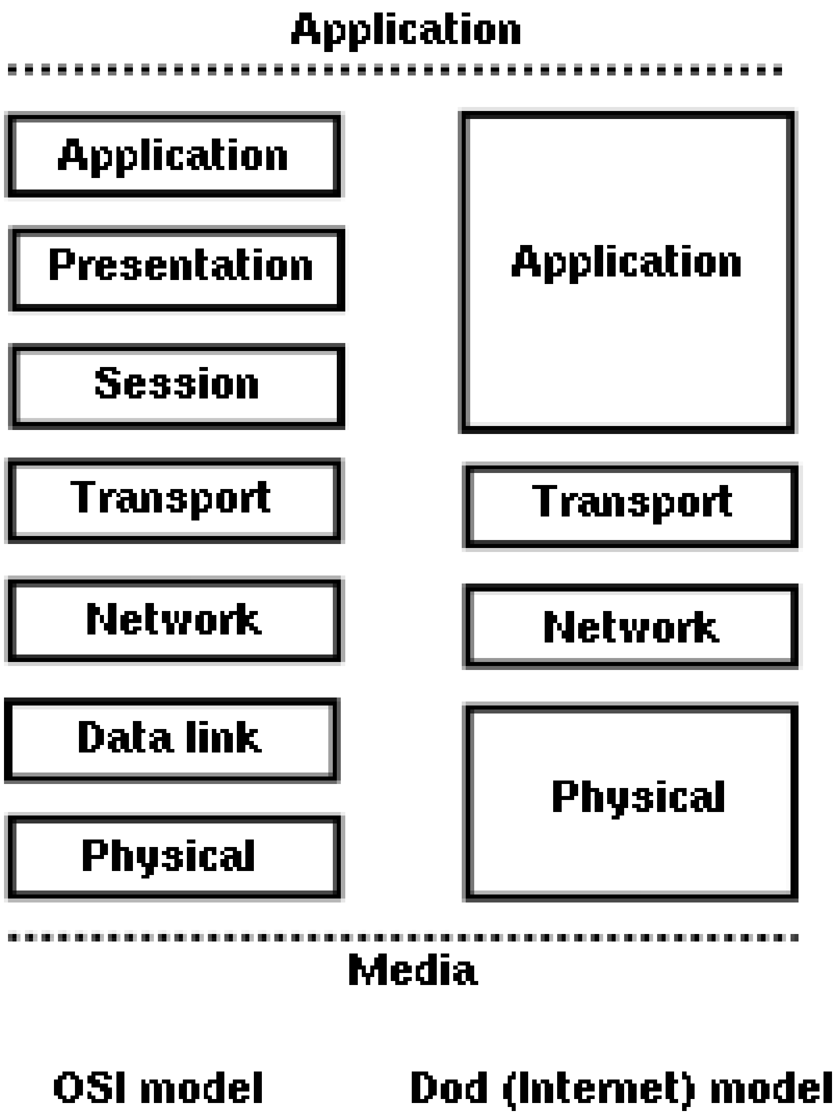
Data Link Layer

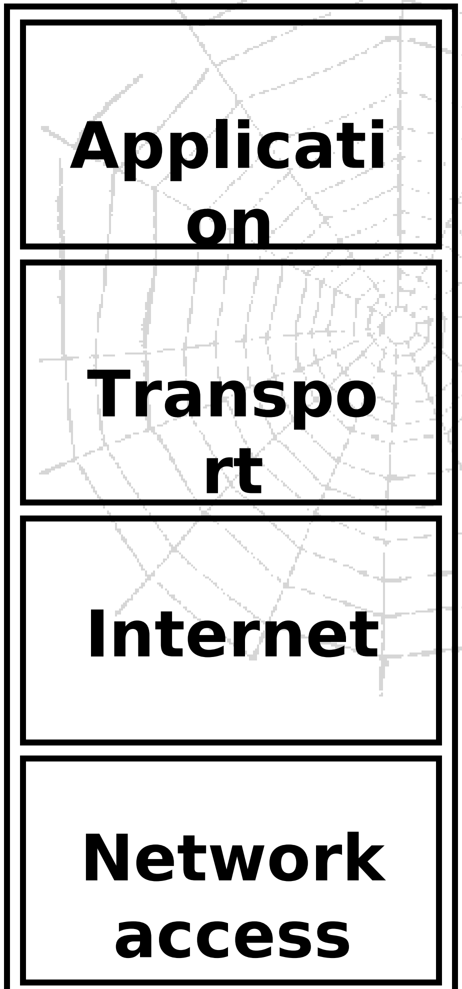
What is a layered model?



2 most common models for network communications

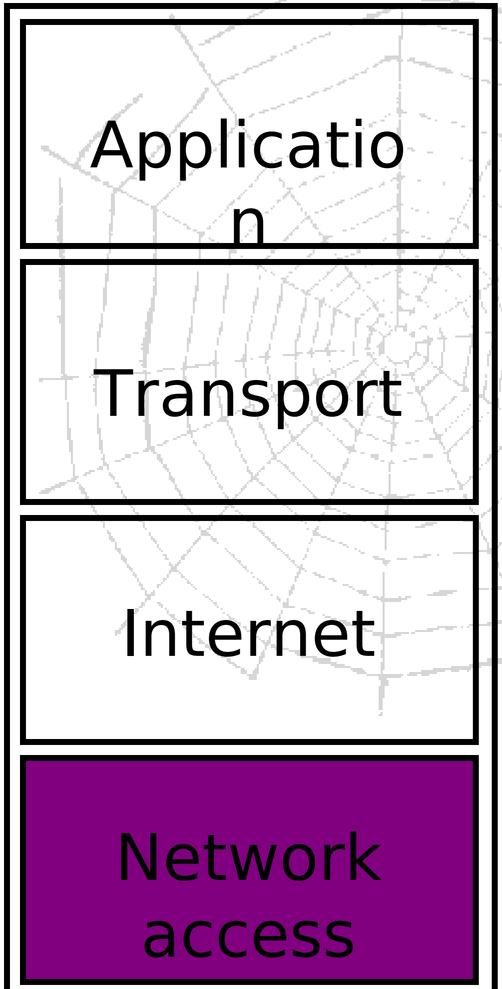
- ISO-OSI 7-Layer Model
  - International Standards Organization's - Open Systems Interconnection model
- TCP/IP 4-Layer Model
  - Developed by the Department of Defense

TCP/IP layered network model



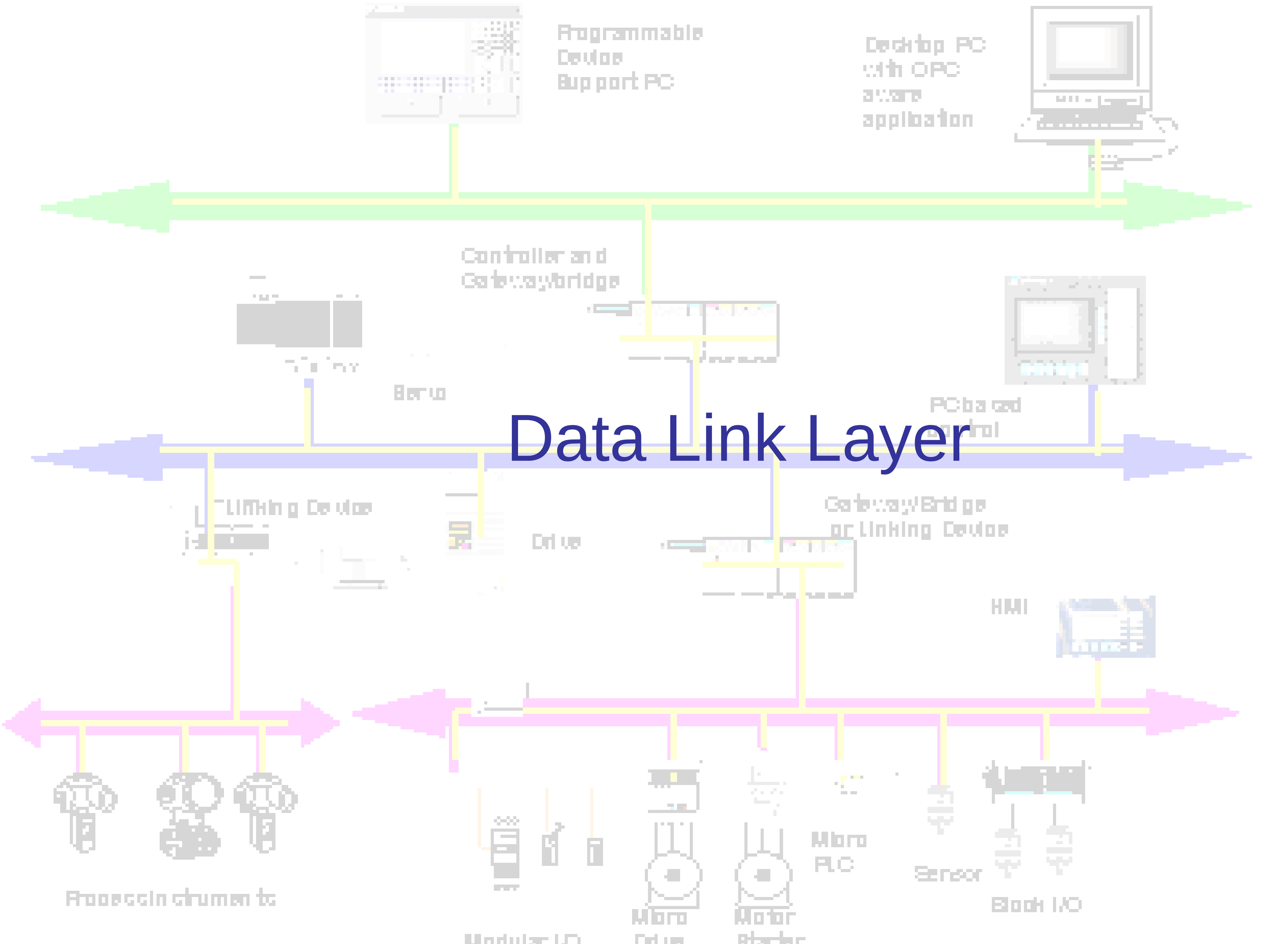
- Transmission Control Protocol and Internet Protocol
- TCP/IP is a suite of protocols, also known as the Internet Protocol Suite
- It was originally developed for the US Department of Defense Advanced Research Project Agency (DARPA) network, but it is now the basis for the Internet

TCP/IP Layers - What does each layer do?



- **Network Access**
- The combination of datalink and physical layers deals with pure hardware (wires, satellite links, network interface cards, etc.)
- Access methods such as **CSMA/CD** (Carrier Sensed Multiple Access with Collision Detection)
- Ethernet exists at the network access layer - its hardware operates at the physical layer and its medium access control method (CSMA/CD) operates at the datalink layer

Data Link Layer



The Data Link Layer

Our goals:

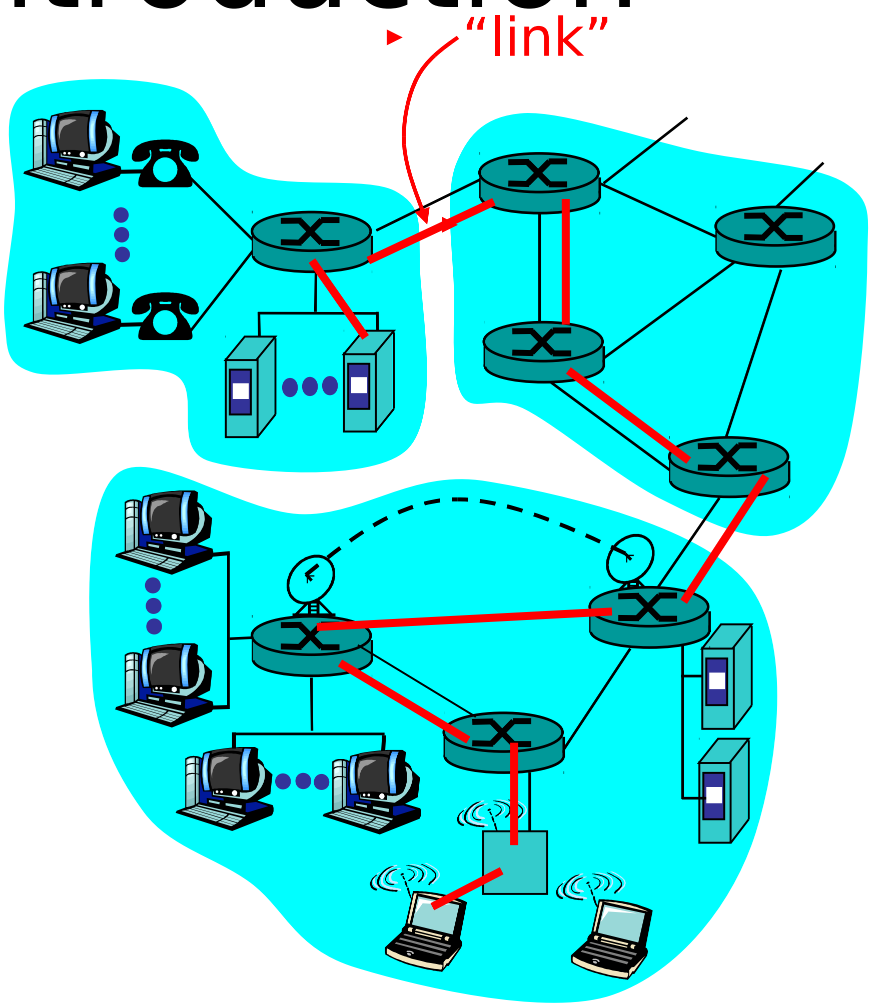
- understand principles behind data link layer services:
  - error detection, correction
  - sharing a broadcast channel: multiple access
  - link layer addressing
  - reliable data transfer, flow control
  - instantiation and implementation of various link layer technologies

Introduction and services

Link Layer: Introduction

Some terminology:

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



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

Link layer: context

- Datagram transferred by different link protocols over different links:
  - e.g., Ethernet on first link, frame relay on intermediate links, 802.11 on last link
- Each link protocol provides different services

transportation analogy

- trip from x to A
  - plane : x to y
  - bus : y to z
  - train: z to A
- tourist = datagram
- transport segment = communication link
- transportation mode = link layer protocol
- travel agent = routing algorithm

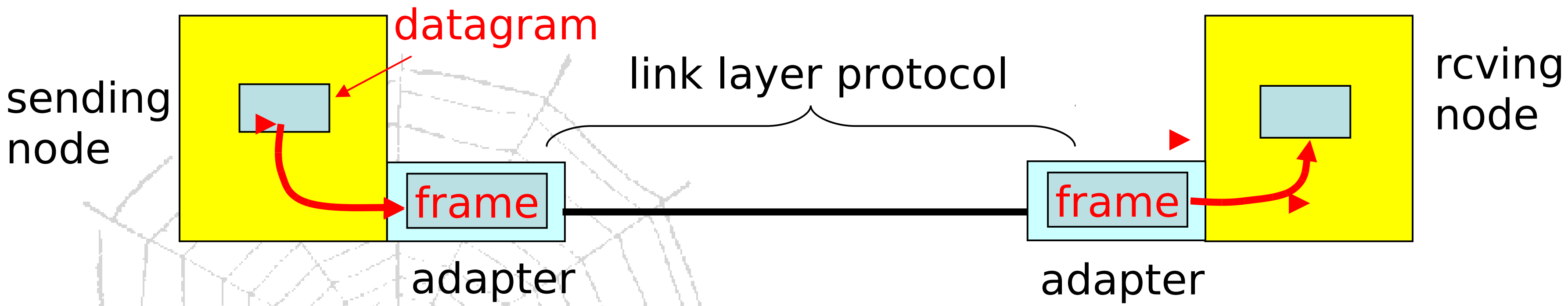
Link Layer Services

- Framing, link access:
  - encapsulate datagram into frame, adding header, trailer
  - channel access if shared medium
  - “MAC” addresses used in frame headers to identify source, dest
    - different from IP address!
- Reliable delivery between adjacent nodes
  - we learned how to do this already
  - seldom used on low bit error link (fiber, some twisted pair)
  - wireless links: high error rates

Link Layer Services (more)

- *Flow Control:*
  - pacing between adjacent sending and receiving nodes
- *Error Detection*:
  - errors caused by signal attenuation, noise.
  - receiver detects presence of errors:
    - signals sender for retransmission or drops frame
- Error Correction:
  - receiver identifies *and corrects* bit error(s) without retransmission
- *Half-duplex and full-duplex*
  - with half duplex, nodes at both ends of link can transmit, but not at same time

Adaptors Communicating



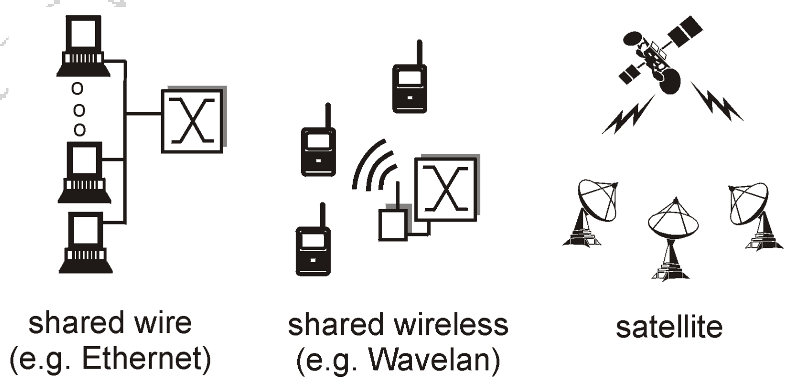
- link layer implemented in "adaptor" (NIC)
  - Ethernet card, 802.11 card
- sending side:
  - encapsulates datagram in a frame
  - adds error checking bits, flow control, etc.
- receiving side
  - looks for errors, flow control, etc
  - extracts datagram, passes to receiving node
- link & physical layers

Multiple access protocols

Multiple Access Links and Protocols

Two types of “links”:

- point-to-point
  - PPP for dial-up access
  - point-to-point link between Ethernet switch and host
- broadcast (shared wire or medium)
  - traditional Ethernet
  - 802.11 wireless LAN



Multiple Access protocols

- Single shared broadcast channel
- Two or more simultaneous transmissions by nodes: interference
  - collision if node receives two or more signals at the same time

multiple access protocol

- Algorithm that determines how nodes share channel, i.e., determine when node can transmit
- communication about channel sharing must use channel itself!

Ideal Mulitple Access Protocol

Broadcast channel of rate R bps

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

MAC Protocols: a taxonomy

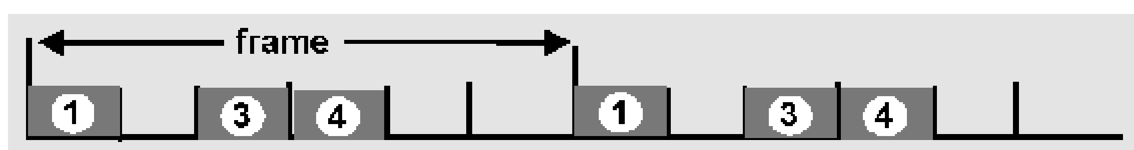
Three broad classes:

- Channel Partitioning
  - divide channel into smaller “pieces” (time slots, frequency, code)
  - allocate piece to node for exclusive use
- Random Access
  - channel not divided, allow collisions
  - “recover” from collisions
- “Taking turns”
  - Nodes take turns, but nodes with more to send can take longer turns

Channel Partitioning MAC protocols: TDMA

TDMA: time division multiple access

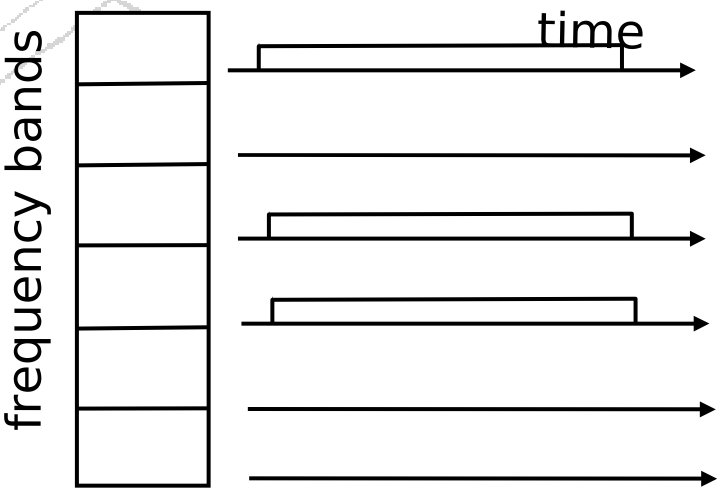
- access to channel in "rounds"
- each station gets fixed length slot (length = pkt trans time) in each round
- unused slots go idle
- example: 6-station LAN, 1,3,4 have pkt, slots 2,5,6 idle



Channel Partitioning MAC protocols: FDMA

FDMA: frequency division multiple access

- channel spectrum divided into frequency bands
- each station assigned fixed frequency band
- unused transmission time in frequency bands go idle
- example: 6-station LAN, 1,3,4 have pkt, frequency bands 2,5,6 idle



Random Access Protocols

- When node has packet to send
  - transmit at full channel data rate R.
  - no coordination among nodes
- two or more transmitting nodes ➜ “collision”,
- random access MAC protocol specifies:
  - how to detect collisions
  - how to recover from collisions (e.g., via delayed retransmissions)
- Examples of random access MAC protocols:
  - slotted ALOHA
  - ALOHA
  - CSMA, CSMA/CD, CSMA/CA

Slotted ALOHA

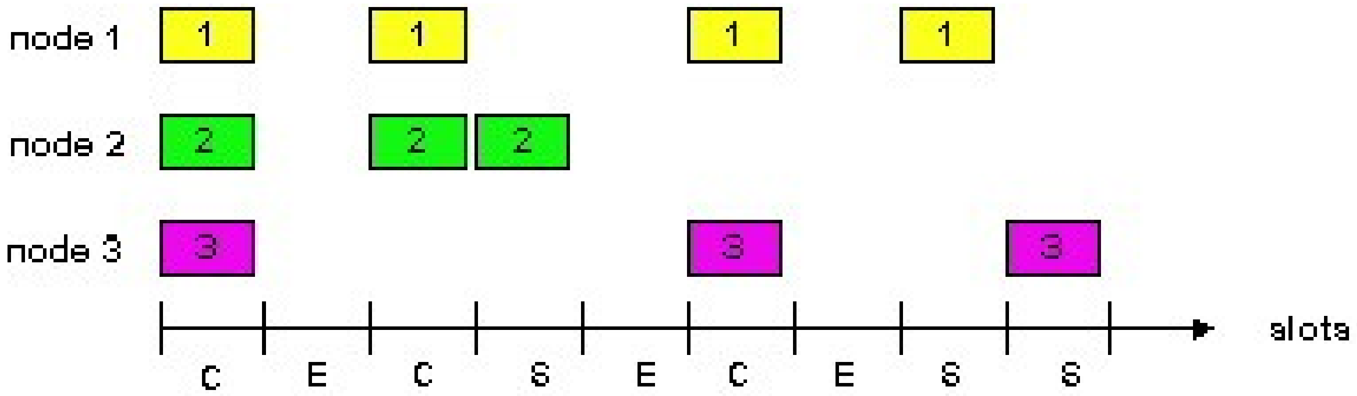
Assumptions

- all frames same size
- time is divided into equal size slots, time to transmit 1 frame
- nodes start to transmit frames only at beginning of slots
- if 2 or more nodes transmit in slot, all nodes detect collision

Operation

- when node obtains fresh frame, it transmits in next slot
- no collision, node can send new frame in next slot
- if collision, node retransmits frame in each subsequent slot with prob. p until success

Slotted ALOHA



Pros

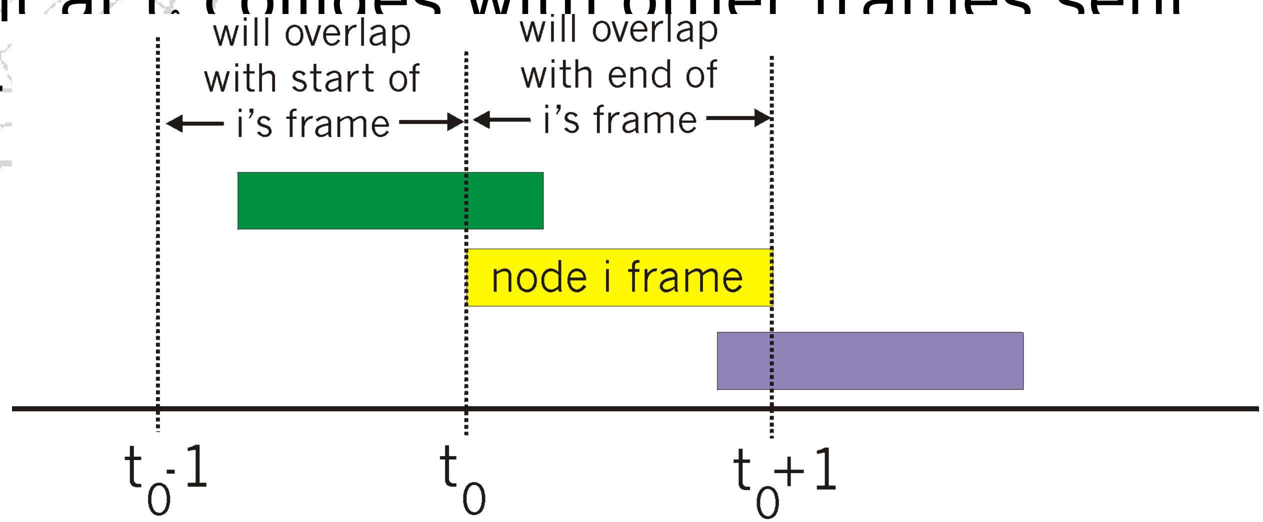
- single active node can continuously transmit at full rate of channel
- highly decentralized: only slots in nodes need to be in sync
- simple

Cons

- collisions, wasting slots
- idle slots
- nodes may be able to detect collision in less than time to transmit packet
- clock synchronization

Pure (unslotted) ALOHA

- unslotted Aloha: simpler, no synchronization
- when frame first arrives
  - transmit immediately
- collision probability increases:
  - frame sent at t_0 collides with other frames sent in $[t_0-1,t_0+$



CSMA (Carrier Sense Multiple Access)

Listen Before Speaking

Carrier Sensing

If someone else beings talking at the same time, Stop talking

Collision Detection

listen before transmit:

If channel sensed idle: transmit entire frame

- If channel sensed busy, postpone transmission
- Human analogy: don’t interrupt others!

CSMA collisions

collisions *can* still occur:

propagation delay means two nodes may not hear each other's transmission

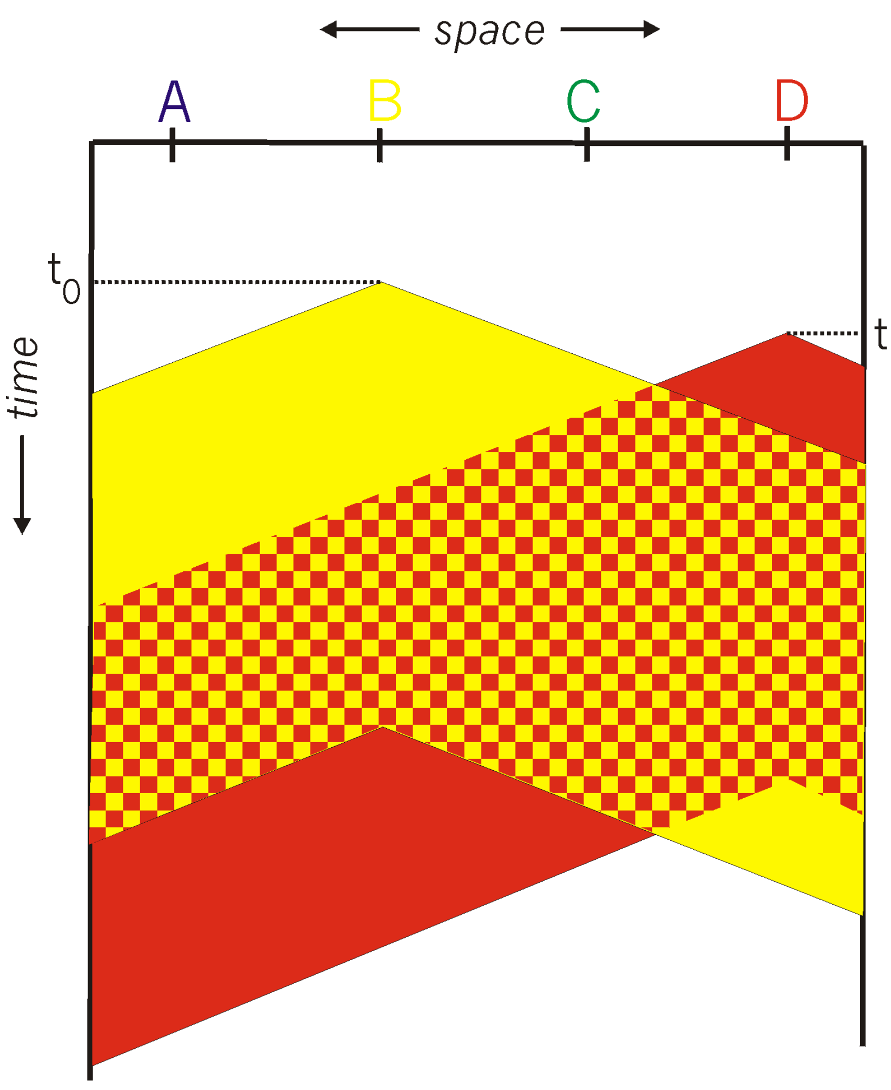
collision:

entire packet transmission time wasted

note:

role of distance & propagation delay in determining collision probability

spatial layout of nodes



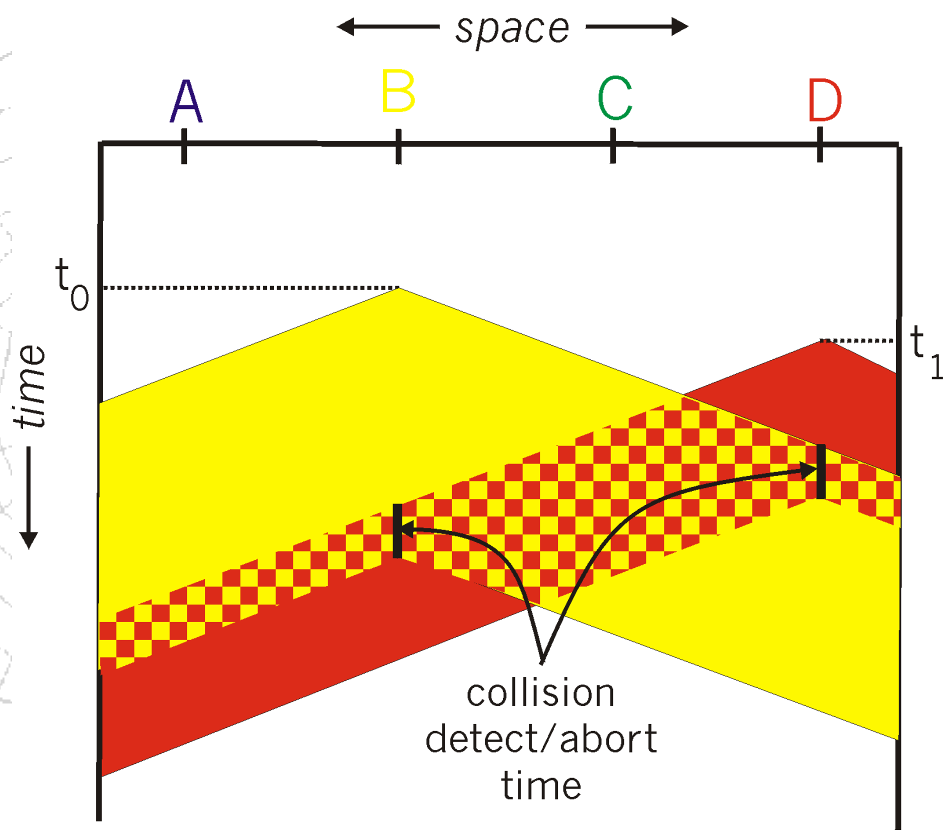
CSMA/CD (Collision Detection)

CSMA/CD: carrier sensing, deferral as in CSMA

- collisions *detected* within short time
- colliding transmissions aborted, reducing channel wastage

- collision detection:
  - easy in wired LANs: measure signal strengths, compare transmitted, received signals
  - difficult in wireless LANs: receiver shut off while transmitting
- human analogy: the polite conversationalist

CSMA/CD collision detection



“Taking Turns” MAC protocols

channel partitioning MAC protocols:

- share channel efficiently and fairly at high load
- inefficient at low load:

Random access MAC protocols

- efficient at low load: single node can fully utilize channel
- high load: collision overhead

“taking turns” protocols

look for best of both worlds!

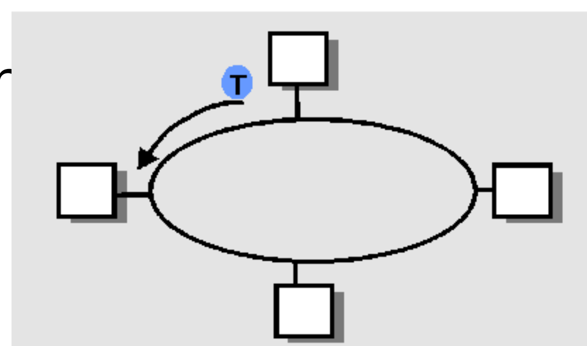
“Taking Turns” MAC protocols

Polling:

- master node “invites” slave nodes to transmit in turn
- concerns:
  - polling overhead
  - latency
  - single point of failure (master

Token passing:

- control **token** passed from one node to next sequentially.
- token message
- concerns:
  - token overhead
  - single point of failure (token)



Summary of MAC protocols

- What do you do with a shared media?
  - Channel Partitioning, by time, frequency or code
    - Time Division, Frequency Division
  - Random partitioning (dynamic),
    - ALOHA, S-ALOHA, CSMA, CSMA/CD
  - Taking Turns
    - polling from a central site, token passing

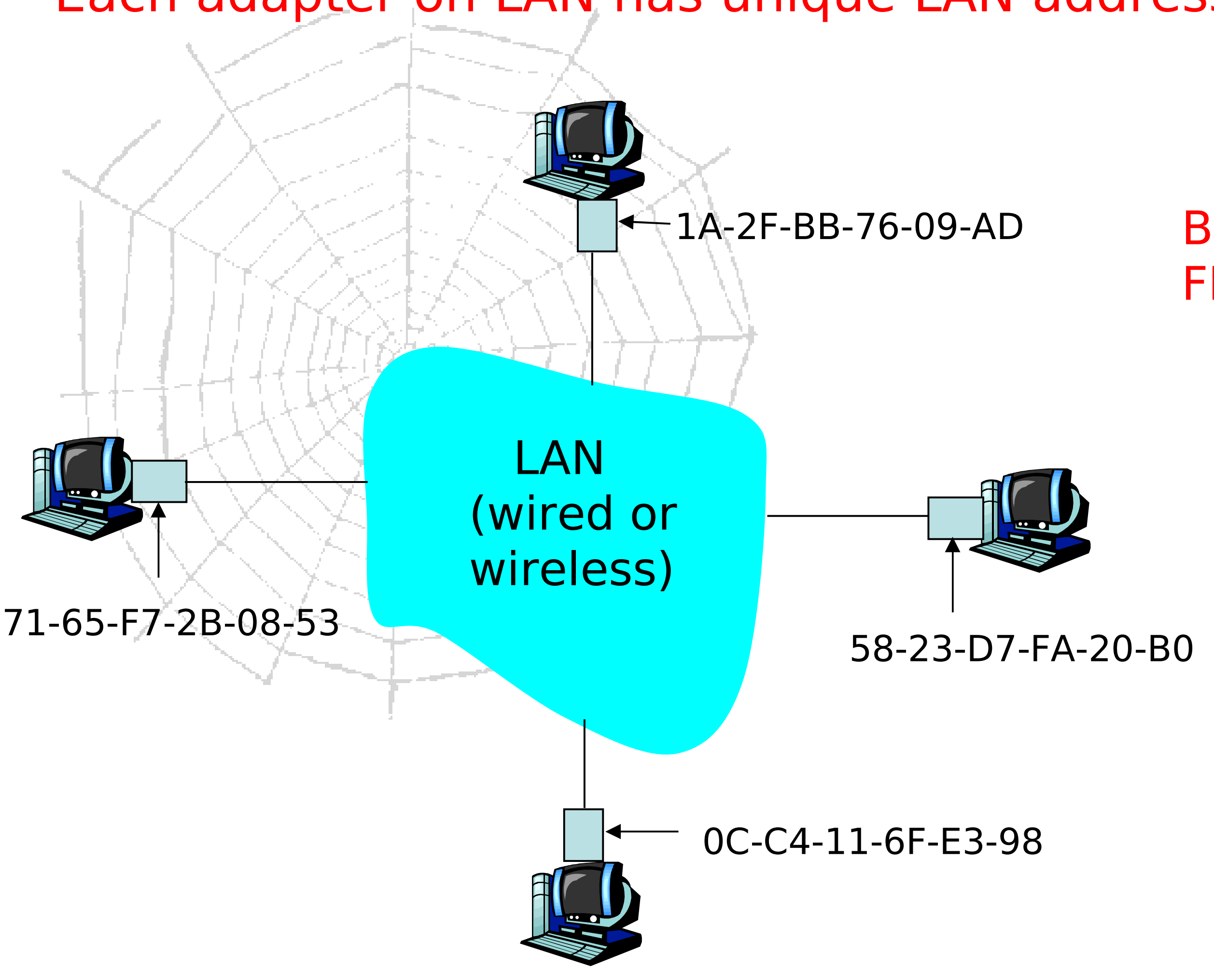
Address Resolution Protocol

MAC Addresses and ARP

- 32-bit IP address:
  - *network-layer* address
  - used to get datagram to destination IP subnet
- MAC (or LAN or physical or Ethernet) address:
  - used to get datagram from one interface to another physically-connected interface (same network)
  - 48 bit MAC address (for most LANs) burned in the adapter ROM

LAN Addresses and ARP

Each adapter on LAN has unique LAN address



Broadcast address =
FF-FF-FF-FF-FF-FF

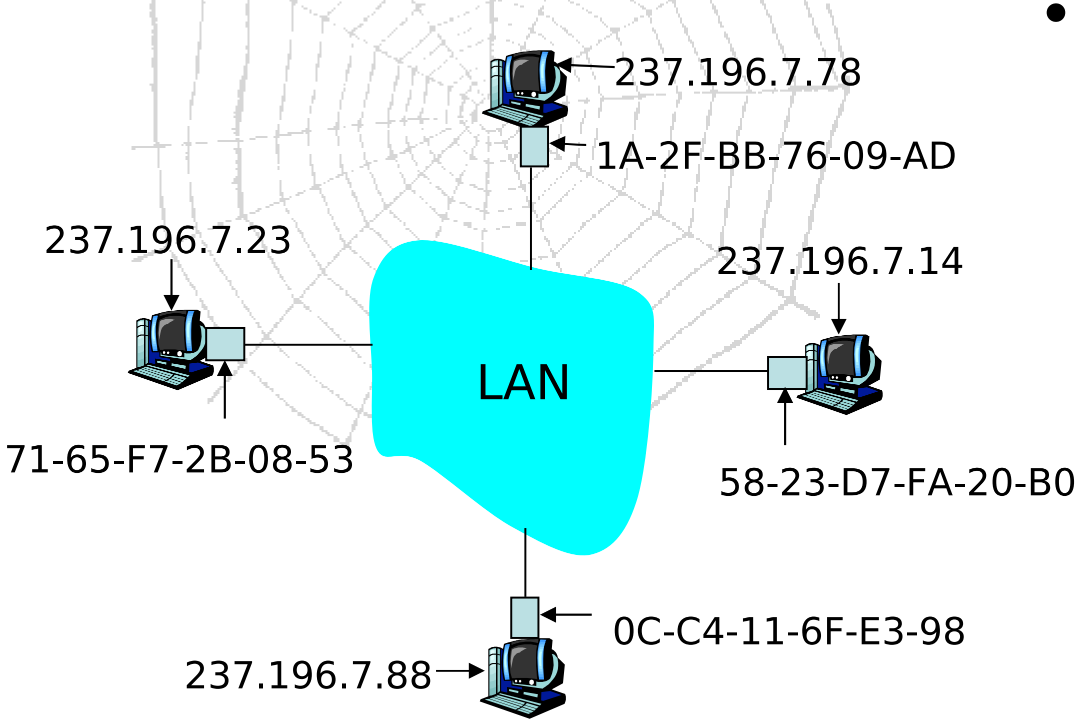
= adapter

LAN Address (more)

- MAC address allocation administered by IEEE
- manufacturer buys portion of MAC address space (to assure uniqueness)
- Analogy:
  - (a) MAC address: like Social Security Number
  - (b) IP address: like postal address
- MAC flat address → portability
  - can move LAN card from one LAN to another
- IP hierarchical address NOT portable
  - depends on IP subnet to which node is attached

ARP: Address Resolution Protocol

Question: how to determine MAC address of B knowing B's IP address?



- Each IP node (Host, Router) on LAN has ARP table
- ARP Table: IP/MAC address mappings for some LAN nodes

  < IP address; MAC address; TTL>

  - TTL (Time To Live): time after which address mapping will be forgotten (typically 20 min)

ARP is defined in RFC 826

ARP protocol: Same LAN (network)

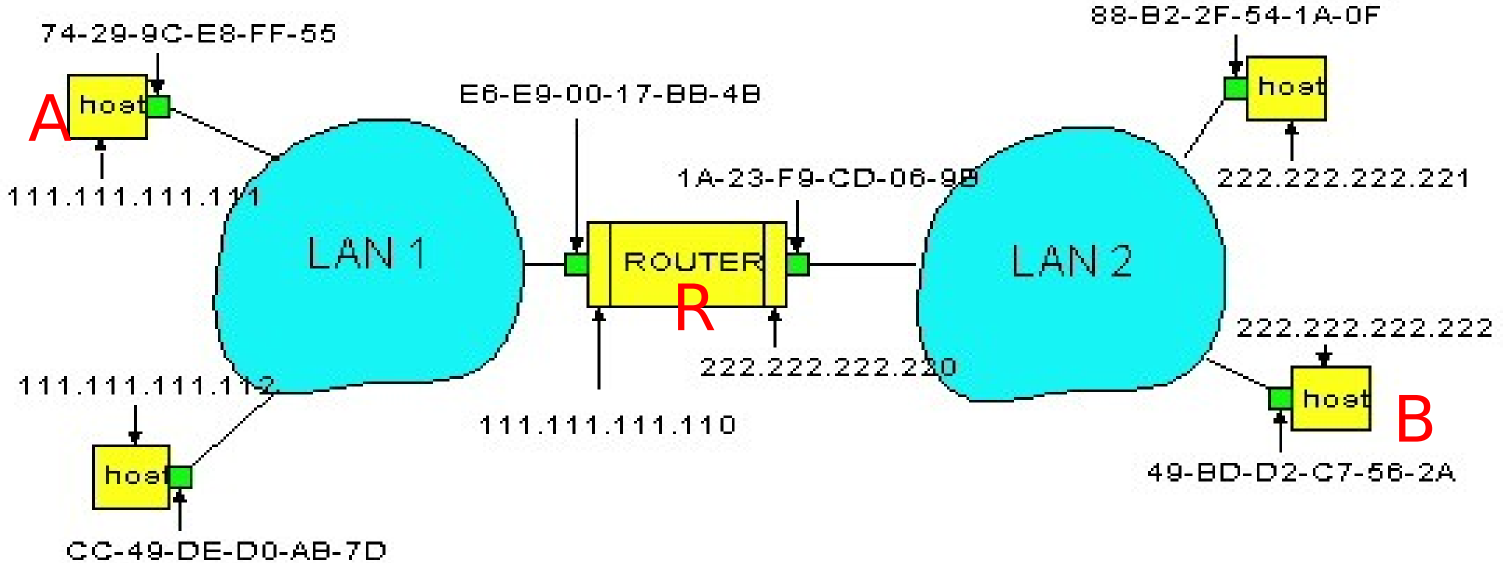
- A wants to send datagram to B, and B’s MAC address not in A’s ARP table.
- A broadcasts ARP query packet, containing B's IP address
  - Dest MAC address = FF-FF-FF-FF-FF-FF
  - all machines on LAN receive ARP query
- B receives ARP packet, replies to A with its (B's) MAC address
  - frame sent to A’s MAC address (unicast)

- A caches (saves) IP-to-MAC address pair in its ARP table until information becomes old (times out)
  - soft state: information that times out (goes away) unless refreshed
- ARP is “plug-and-play”:
  - nodes create their ARP tables without intervention from net administrator

Routing to another LAN

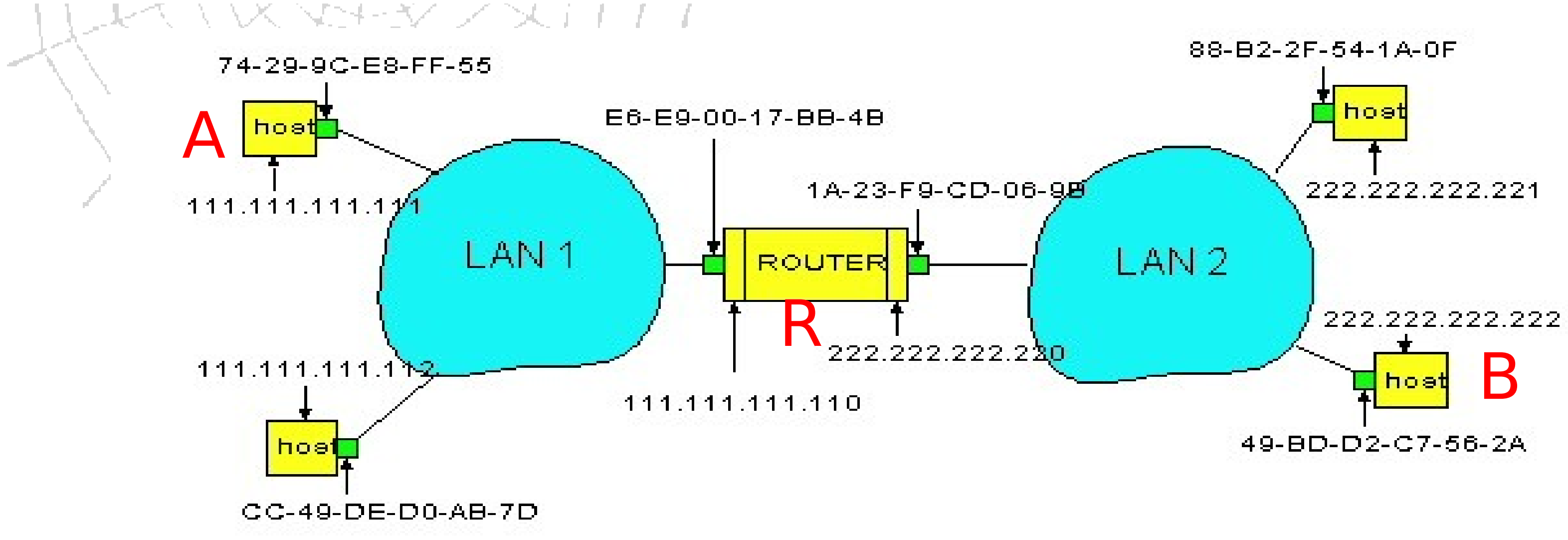
walkthrough: send datagram from A to B via R

assume A know's B IP address



- Two ARP tables in router R, one for each IP network (LAN)
- In routing table at source Host, find router 111.111.111.110
- In ARP table at source, find MAC address E6-E9-00-17-BB-4B, etc

- A creates datagram with source A, destination B
- A uses ARP to get R’s MAC address for 111.111.111.110
- A creates link-layer frame with R's MAC address as dest, frame contains A-to-B IP datagram
- A’s adapter sends frame
- R’s adapter receives frame
- R removes IP datagram from Ethernet frame, sees its destined to B
- R uses ARP to get B’s MAC address
- R creates frame containing A-to-B IP datagram sends to B



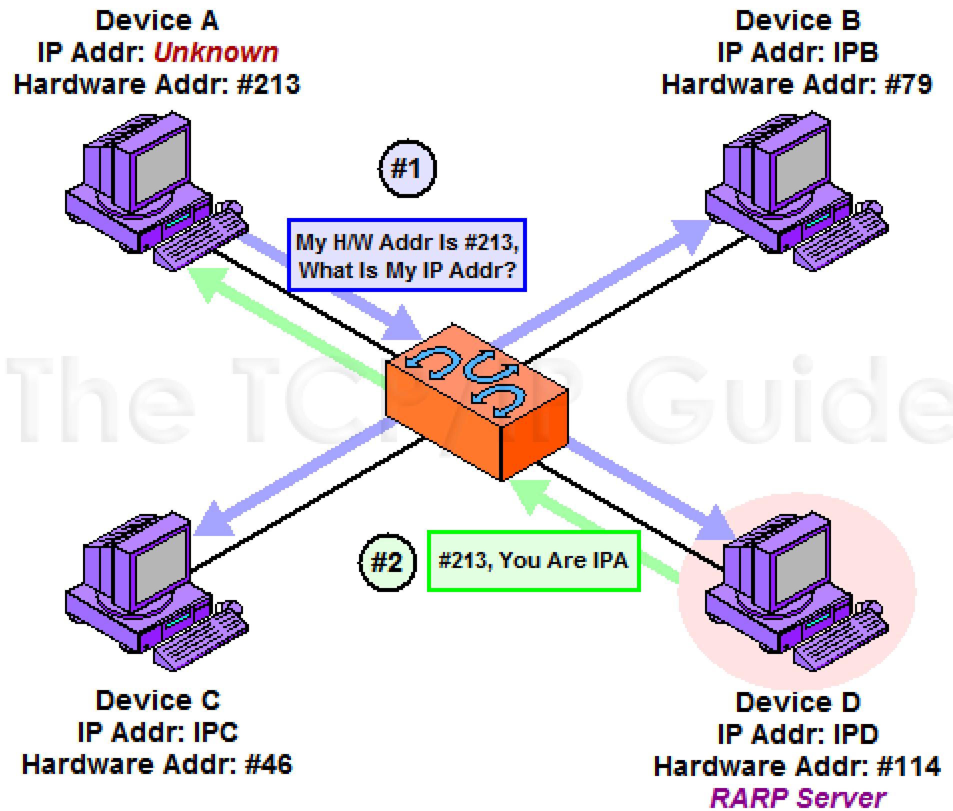
Reverse ARP

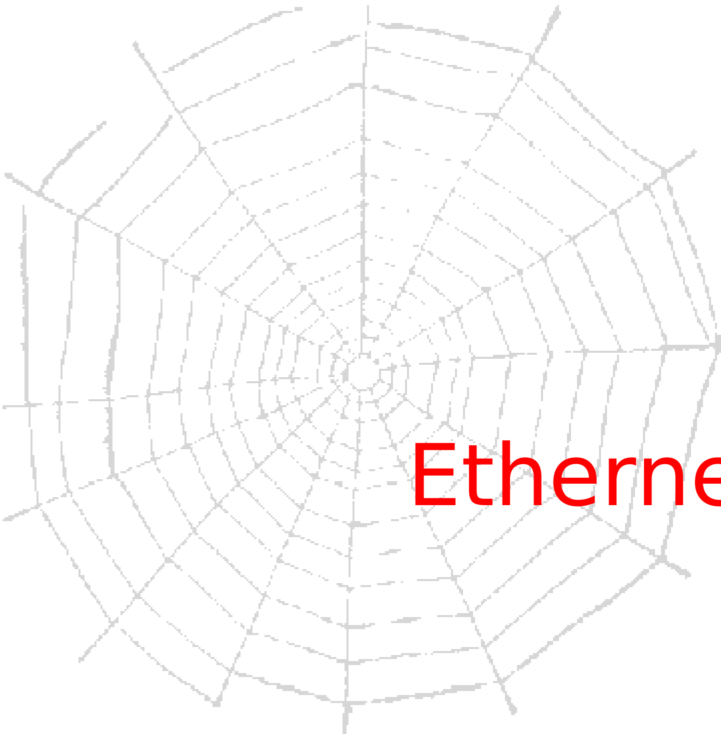
RARP is a network layer protocol used to obtain an IP address for a given hardware address

RARP is the complement of ARP

<u>**RARP SERVER**</u>
BOOTP
DHCP

RARP is described in RFC 903



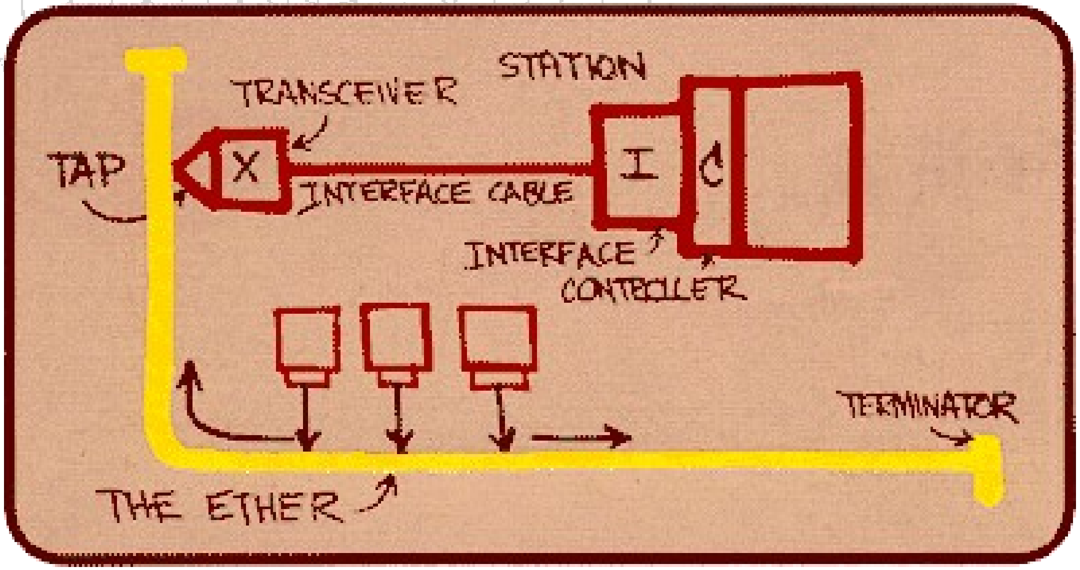


Ethernet

Ethernet

"dominant" wired LAN technology:

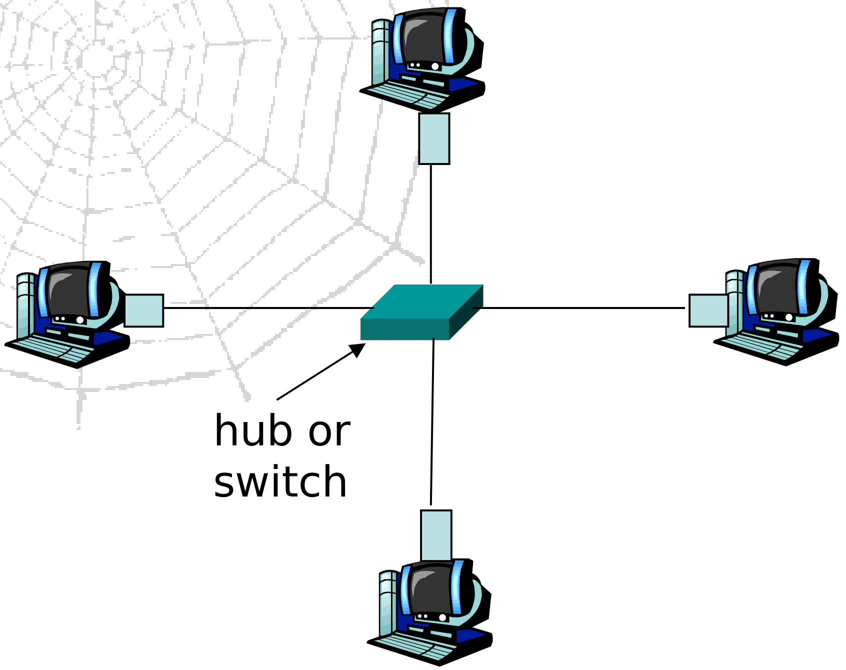
- first widely used LAN technology
- Simpler, cheaper than token LANs and ATM
- Kept up with speed rate: 10 Mbps – 10 Gbps



Ethernet sketch

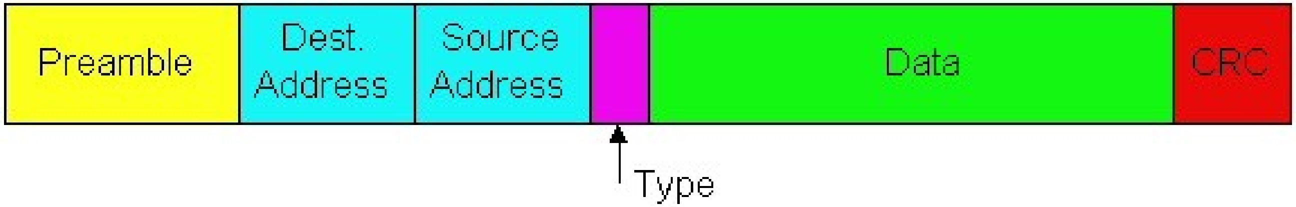
Star topology

- Bus topology popular through mid 90s
- Now star topology prevails
- Connection choices: hub or switch



Ethernet Frame Structure

Sending adapter encapsulates IP datagram (or other network layer protocol packet) in Ethernet frame

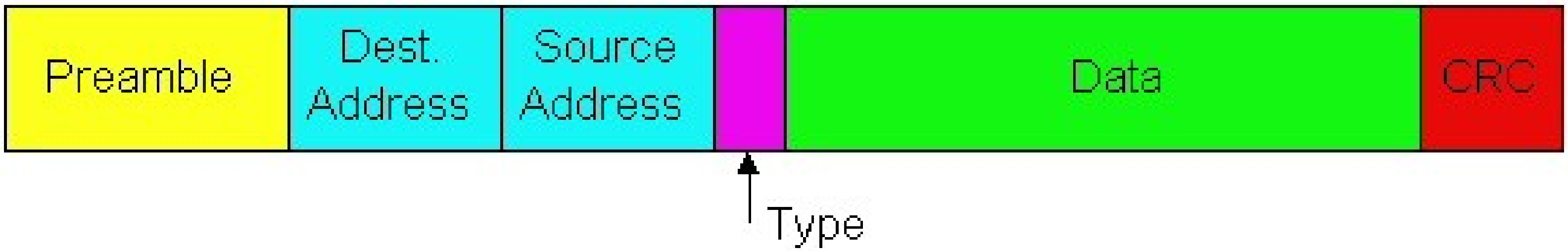


Preamble:

- 7 bytes with pattern 10101010 followed by one byte with pattern 10101011
- used to synchronize receiver, sender clock rates

Ethernet Frame Structure (more)

- Addresses: 6 bytes
  - if adapter receives frame with matching destination address, or with broadcast address (eg ARP packet), it passes data in frame to net-layer protocol
  - otherwise, adapter discards frame
- Type: indicates the higher layer protocol - mostly IP
- CRC: checked at receiver, if error is detected, the frame is simply dropped



Unreliable, connectionless service

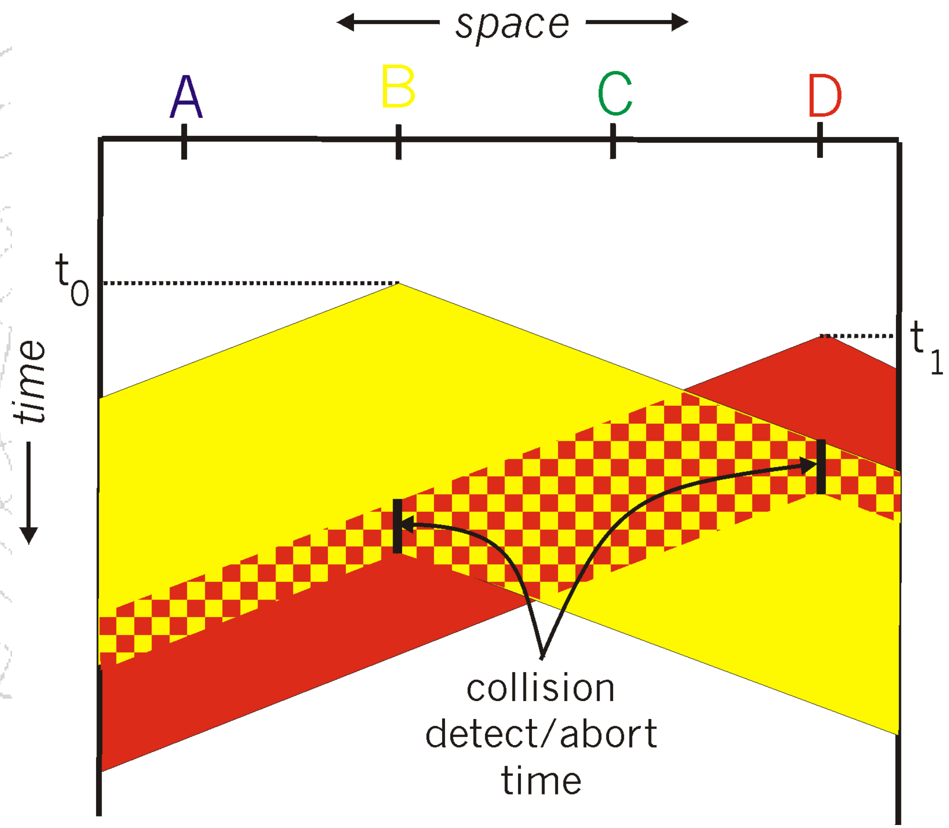
- Connectionless: No handshaking between sending and receiving adapter.
- Unreliable: receiving adapter doesn't send acks or nacks to sending adapter
  - stream of datagrams passed to network layer can have gaps
  - gaps will be filled if app is using TCP
  - otherwise, app will see the gaps

Ethernet uses CSMA/CD

- 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

CSMA/CD collision detection



Ethernet CSMA/CD algorithm

1. Adaptor receives datagram from net layer & creates frame
2. If adapter senses channel idle, it starts to transmit frame. If it senses channel busy, waits until channel idle and then transmits
3. If adapter detects another transmission while transmitting, aborts and sends jam signal
4. After aborting, adapter enters **exponential backoff**: after the mth collision, adapter chooses a K at random from $\{0,1,2,\ldots,2^m-1\}$. Adapter waits K·512 bit times and returns to Step 2

Ethernet's CSMA/CD (more)

Jam Signal: make sure all other transmitters are aware of collision; 48 bits

Exponential Backoff:

- *Goal*: adapt retransmission attempts to estimated current load
  - heavy load: random wait will be longer

- first collision: choose K from {0,1}; delay is $K \cdot 512$ bit transmission times
- after second collision: choose K from {0,1,2,3}…
- after ten collisions, choose K from {0,1,2,3,4,…,1023}