Computer Networks I

Local Area Networks

Amitangshu Pal
Computer Science and Engineering
IIT Kanpur

MAC addresses

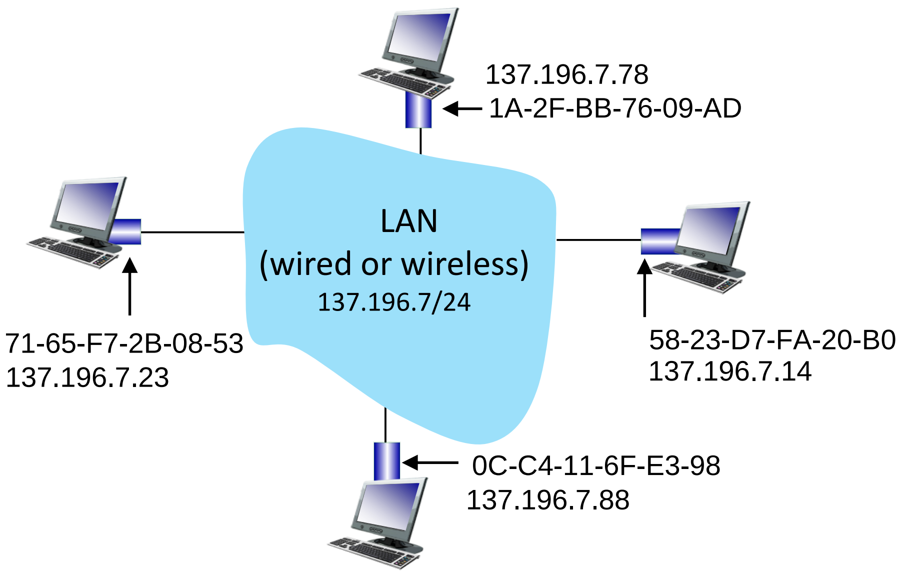
- 32-bit IP address:
  - *network-layer* address for interface
  - used for layer 3 (network layer) forwarding
  - e.g.: 128.119.40.136
- MAC (or LAN or physical or Ethernet) address:
  - function: used "locally" to get frame from one interface to another physically-connected interface (same subnet, in IP-addressing sense)
  - 48-bit MAC address (for most LANs) burned in NIC ROM, also sometimes software settable
  - e.g.: 1A-2F-BB-76-09-AD

hexadecimal (base 16) notation
(each "numeral" represents 4 bits)

MAC addresses

Each interface on LAN

- Has unique 48-bit MAC address
- Has a locally unique 32-bit IP address



MAC addresses

- MAC address allocation administered by IEEE
- Manufacturer buys portion of MAC address space (to assure uniqueness)
- Analogy:
  - MAC address: like Social Security Number
  - IP address: like postal address
- MAC flat address: portability
  - Can move interface from one LAN to another
  - IP address *not* portable: depends on IP subnet to which node is attached

Ethernet

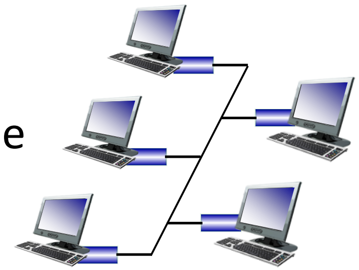
"Dominant" wired LAN technology:

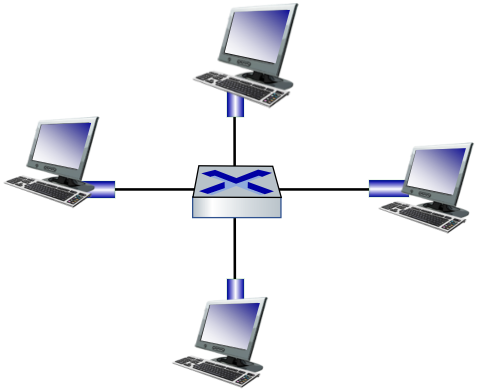
- First widely used LAN technology
- Simpler, cheap
- Kept up with speed race: 10 Mbps – 400 Gbps

Ethernet: physical topology

- Bus: popular through mid 90s
  - All nodes in same collision domain (can collide with each other)
- Switched: prevails today
  - Active link-layer 2 *switch* in center
  - Each “spoke” runs a (separate) Ethernet protocol (nodes do not collide with each other)

Bus: coaxial cable





Switched

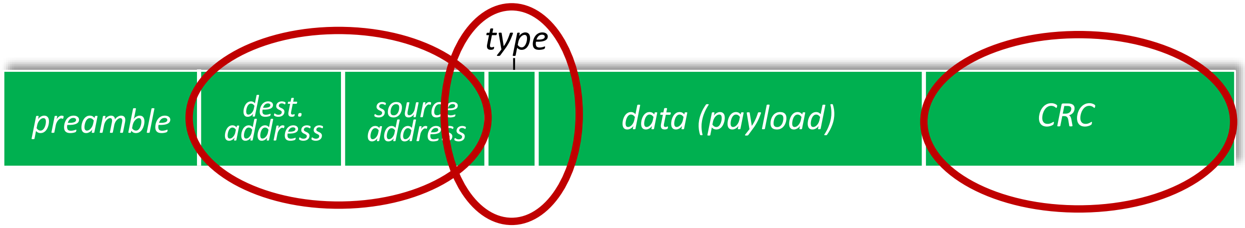
Ethernet frame structure

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

Preamble:

- Used to synchronize receiver, sender clock rates
- 7 bytes of 10101010 followed by one byte of 10101011

Ethernet frame structure (more)



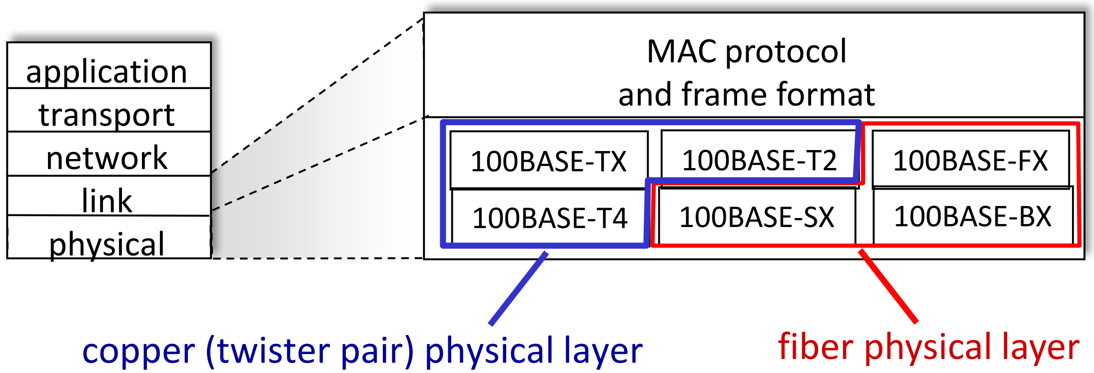
- Addresses: 6 byte source, destination MAC addresses
  - If adapter receives frame with matching destination address, or with broadcast address (e.g., ARP packet), it passes data in frame to network layer protocol
  - Otherwise, adapter discards frame
- Type: indicates higher layer protocol
  - Mostly IP but others possible, e.g., Novell IPX, AppleTalk
  - Used to demultiplex up at receiver
- CRC: Cyclic redundancy check at receiver
  - Error detected: frame is dropped

Ethernet: unreliable, connectionless

- Connectionless: No handshaking between sending and receiving NICs
- Unreliable: Receiving NIC doesn't send ACKs or NAKs to sending NIC
  - Data in dropped frames recovered only if initial sender uses higher layer rdt (e.g., TCP), otherwise dropped data lost
- Ethernet's MAC protocol: Unslotted CSMA/CD with binary backoff

802.3 Ethernet standards: link & physical layers

- *Many* different Ethernet standards
  - 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

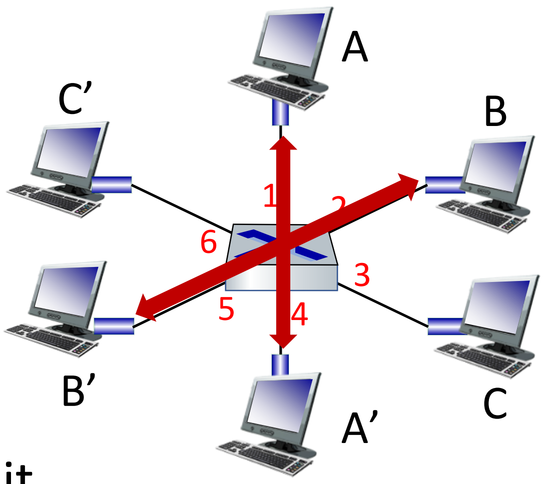


Ethernet switch

- Switch is a link-layer device: takes an *active* role
  - Store, forward Ethernet frames
  - Examine incoming frame's MAC address, *selectively* forward frame to one-or-more outgoing links when frame is to be forwarded on segment, uses CSMA/CD to access segment
- Transparent: hosts *unaware* of presence of switches
- Plug-and-play, self-learning
  - Switches do not need to be configured

Switch: multiple simultaneous transmissions

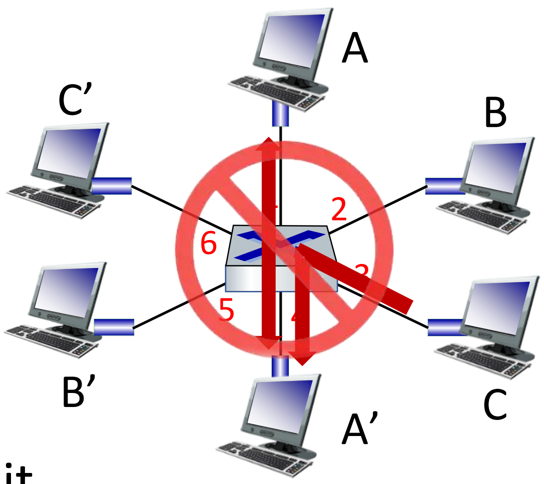
- Hosts have dedicated, direct connection to switch
- Switches buffer packets
- Ethernet protocol used on *each* incoming link, so:
  - No collisions; full duplex
  - Each link is its own collision domain
- Switching: A-to-A' and B-to-B' can transmit simultaneously, without collisions



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

Switch: multiple simultaneous transmissions

- Hosts have dedicated, direct connection to switch
- Switches buffer packets
- Ethernet protocol used on *each* incoming link, so:
  - No collisions; full duplex
  - Each link is its own collision domain
- Switching: A-to-A' and B-to-B' can transmit simultaneously, without collisions
  - But A-to-A' and C to A' can *not* happen simultaneously



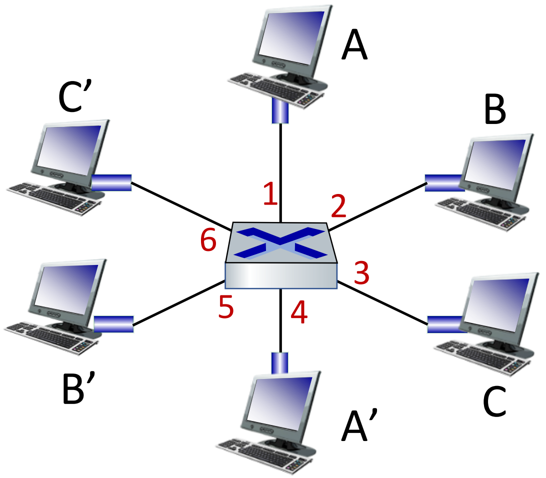
switch with six interfaces (1,2,3,4,5,6)

Switch forwarding table

Q: How does switch know A' reachable via interface 4, B' reachable via interface 5?

A: Each switch has a switch table, each entry:

- (MAC address of host, interface to reach host, time stamp)
- Looks like a routing table!

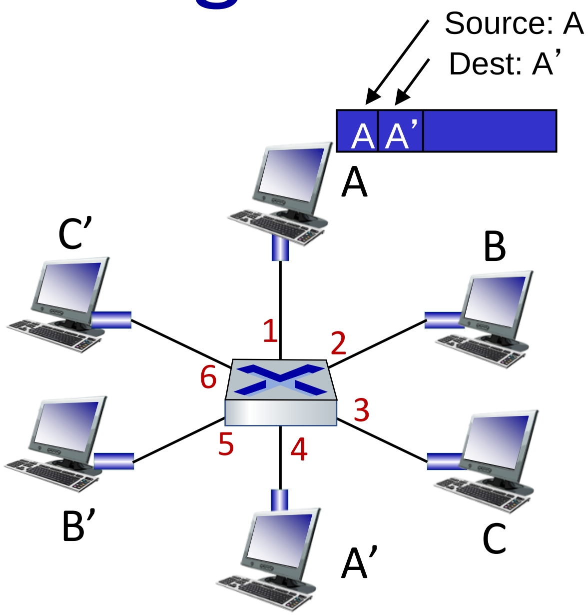


Q: How are entries created, maintained in switch table?

- Something like a routing protocol?

Switch: self-learning

- Switch *learns* which hosts can be reached through which interfaces
  - When frame received, switch "learns" location of sender: incoming LAN segment
  - Records sender/location pair in switch table



MAC addr	interface	TTL
A	*1*	*60*

Switch table (initially empty)

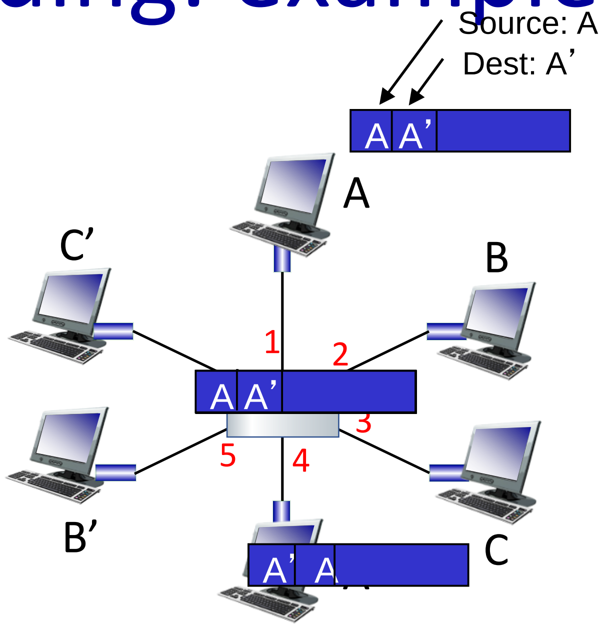
Switch: frame filtering/forwarding

When frame received at switch:

1. Record incoming link, MAC address of sending host
2. Index switch table using MAC destination address
3. If entry found for destination
   then {
   if destination on segment from which frame arrived
      then drop frame
      else forward frame on interface indicated by entry
   }
   else flood /* forward on all interfaces except arriving interface */

Self-learning, forwarding: example

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

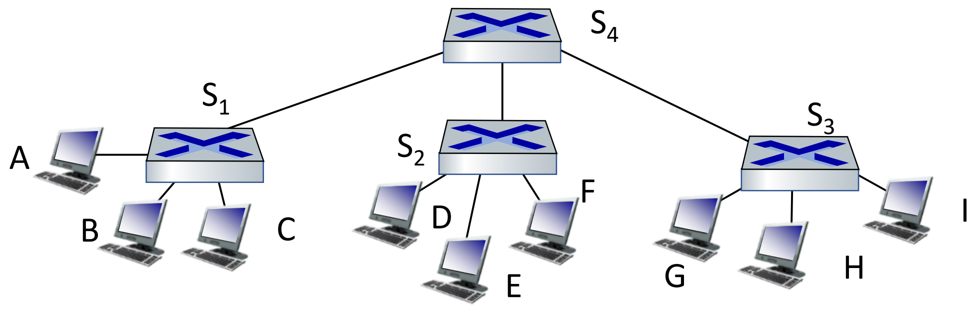


MAC addr	interface	TTL
A	*1*	*60*
A'	*4*	*60*

switch table (initially empty)

Interconnecting switches

Self-learning switches can be connected together:



Q: Sending from A to G - how does S_1 know to forward frame destined to G via S_4 and S_3?

- *A:* Self learning! (works exactly the same as in single-switch case!)

2 Observations on CSMA/CD

- Transmitter can send/listen concurrently
  - If (Transmitted - Sensed = null)? Then success

- The signal is identical at Tx and Rx
  - Non-dispersive

The TRANSMITTER can detect if and when collision occurs

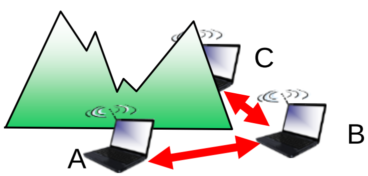
IEEE 802.11 Wireless LAN

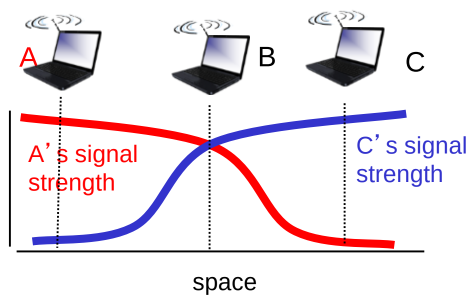
IEEE 802.11 standard	Year	Max data rate	Range	Frequency
802.11b	1999	11 Mbps	30 m	2.4 Ghz
802.11g	2003	54 Mbps	30m	2.4 Ghz
802.11n (WiFi 4)	2009	600	70m	2.4, 5 Ghz
802.11ac (WiFi 5)	2013	3.47Gpbs	70m	5 Ghz
802.11ax (WiFi 6)	2020 (exp.)	14 Gbps	70m	2.4, 5 Ghz
802.11af	2014	35 – 560 Mbps	1 Km	unused TV bands (54-790 MHz)
802.11ah	2017	347Mbps	1 Km	900 Mhz

- All use CSMA/CA for multiple access

IEEE 802.11: multiple access

- Avoid collisions: 2^+ nodes transmitting at same time
- 802.11: CSMA - sense before transmitting
  - Don't collide with detected ongoing transmission by another node
- 802.11: *no* collision detection!
  - Difficult to sense collisions: high transmitting signal, weak received signal due to fading
  - Can't sense all collisions in any case: hidden terminal, fading
  - Goal: *avoid collisions:* CSMA/CollisionAvoidance



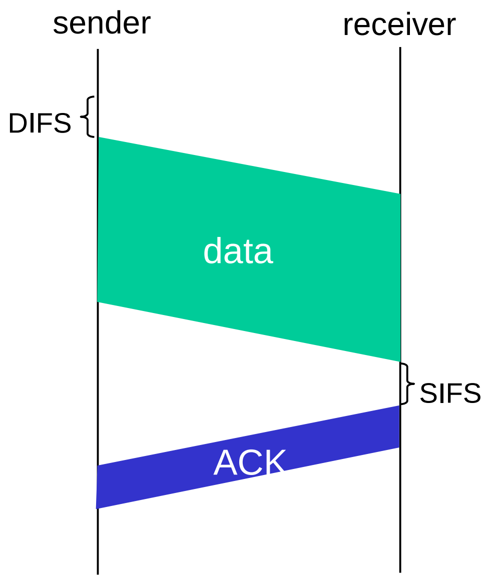


IEEE 802.11 MAC Protocol: CSMA/CA

802.11 sender

1 If sense channel idle for **DIFS** then
    transmit entire frame (no CD)

2 If sense channel busy then
    start random backoff time
    timer counts down while channel idle
    transmit when timer expires
    if no ACK, increase random backoff interval, repeat 2



802.11 receiver

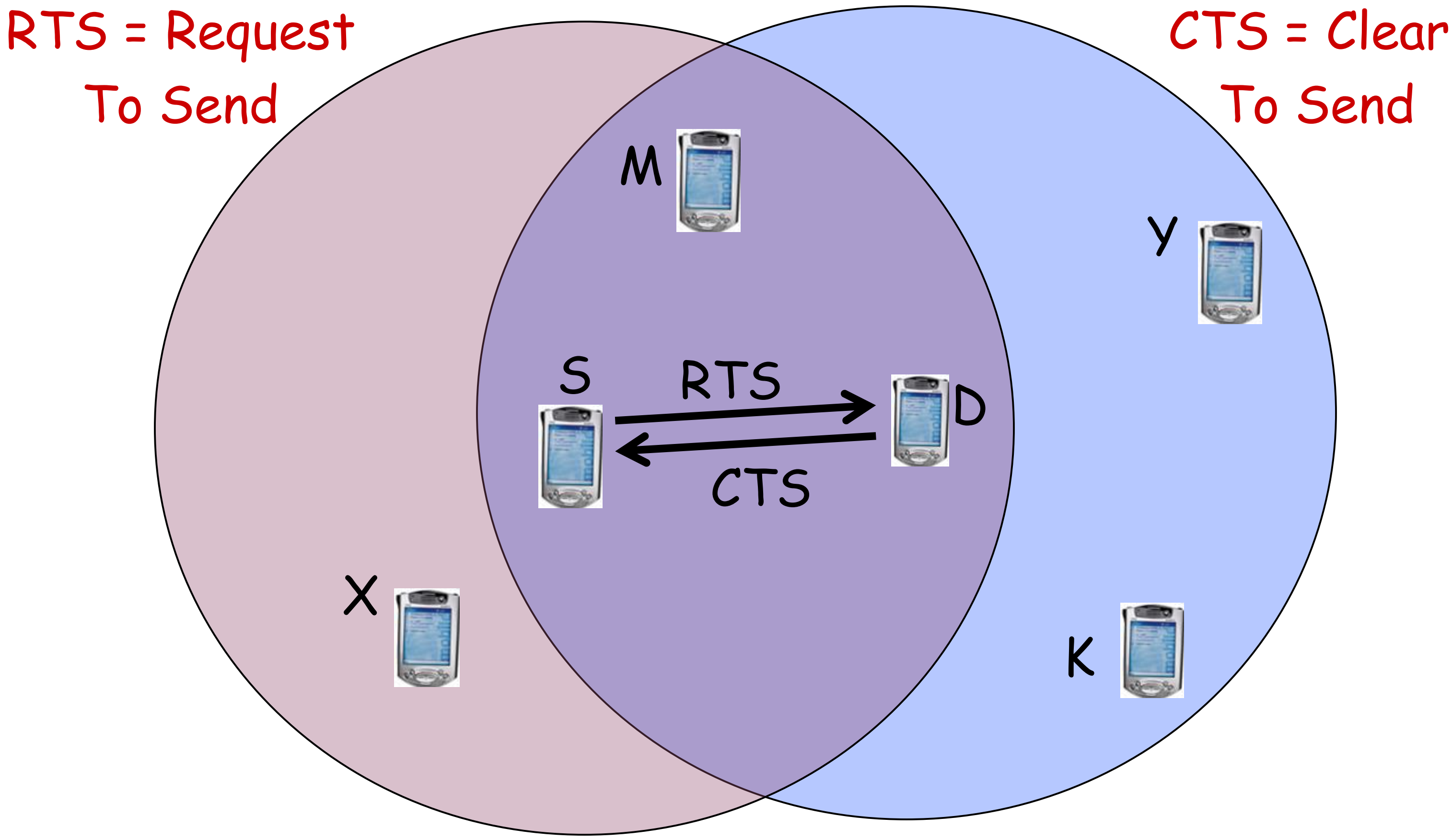
If frame received OK
  return ACK after **SIFS** (ACK needed due to hidden terminal problem)

Avoiding collisions (more)

Idea: sender "reserves" channel use for data frames using small reservation packets

- Sender first transmits *small* request-to-send (RTS) packet to BS using CSMA
  - RTSs may still collide with each other (but they're short)
- BS broadcasts clear-to-send CTS in response to RTS
- CTS heard by all nodes
  - Sender transmits data frame
  - Other stations defer transmissions

IEEE 802.11



IEEE 802.11