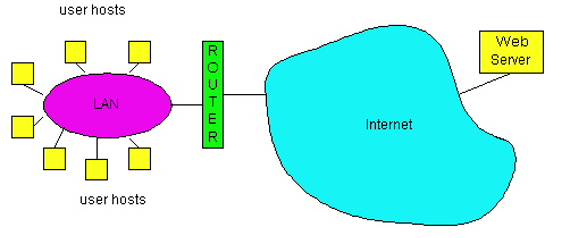
# 5.2 – Data link layer

## Introduction: LAN Technologies

* Data link layer so far:
  + services, error detection/correction, multiple access
* Next: LAN technologies
  + addressing
  + Ethernet, 802.11 WLAN
  + hubs, bridges, switches
  + PPP, ATM

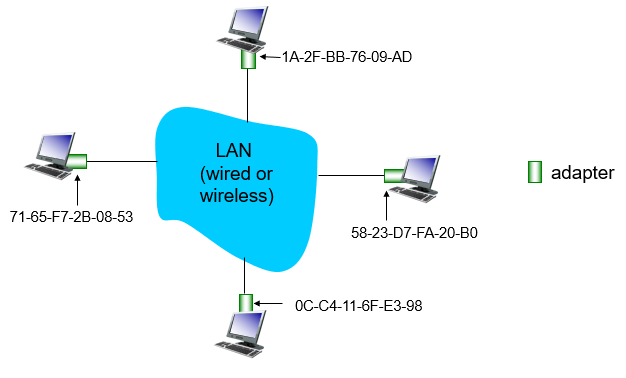


## MAC Addresses and ARP

* 32-bit IP address:
  + *network-layer* address for a host (more specifically, an interface)
  + used for layer 3 forwarding; i.e. to get datagram to destination network
* MAC (or LAN or physical or Ethernet) address:
  + used “locally” to get datagram from one interface to another physically-connected interface (same network, in IP-addressing sense)
  + 48 bit MAC address (for most LANs) burned in the adapter ROM, also sometimes software settable
  + e.g. 1A-2F-BB-76-09-AD
    - hexadecimal (base 16) notation; each “number”, e.g. the 1 of 1A, represents 4 bits

## LAN addresses and ARP

* each adapter on LAN has unique LAN address

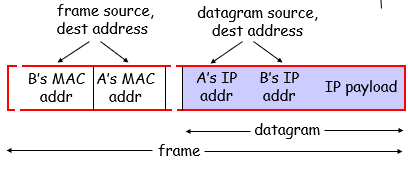
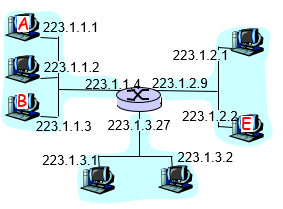


## LAN Address (more)

* MAC address allocation administered by IEEE
* manufacturer buys portion of MAC address space (to assure uniqueness)
* analogy:
  + MAC address: like social security number (absolutely unique)
  + IP address: like postal address (partially unique)
* MAC flat address ➜ portability: can move LAN card from one LAN to another
* IP hierarchical address *not* portable: address depends on IP subnet to which node is attached

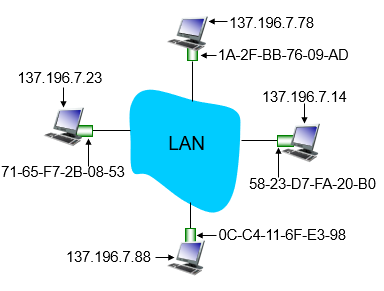
## Recall earlier routing discussions

* Starting at A, given IP datagram addressed to B:
  + look up net. address of B, find B on same net. as A
  + link layer sends datagram to B inside link-layer frame



## Address Resolution Protocol (ARP)

* Question: how to determine MAC address of an interface, given its IP address?

****

* **E**ach IP node (host, router, switch, etc) on LAN has ARP table
* **ARP Table:** IP/MAC address mappings for some LAN nodes

<IP address; MAC address; TTL>

<……………………………….>

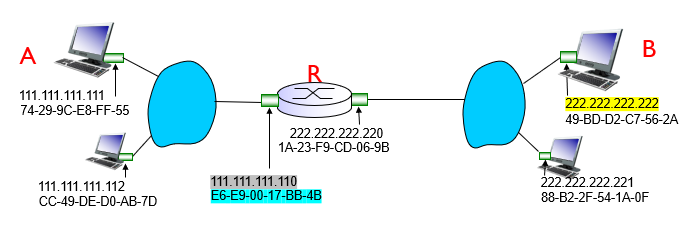
* + TTL (Time To Live): time after which address mapping will be forgotten
  + to see the ARP table on a computer, write [arp -a] in the command prompt

## ARP Protocol: same LAN

* Host A wants to send datagram to Host B
  + Host B’s MAC address not in Host A’s ARP table
  + Host A knows B's IP address, wants to learn physical address of Host B
* Host A *broadcasts* ARP query pkt, containing Host B's IP address
  + destination MAC address = FF-FF-FF-FF-FF-FF (probably to default router)
  + all machines on LAN receive ARP query
* Host B receives ARP packet, replies to Host A with its (Host B's) MAC address
  + frame sent to Host A’s MAC address (unicast)
* Host A caches (saves) IP-to-physical address pairs 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 admin

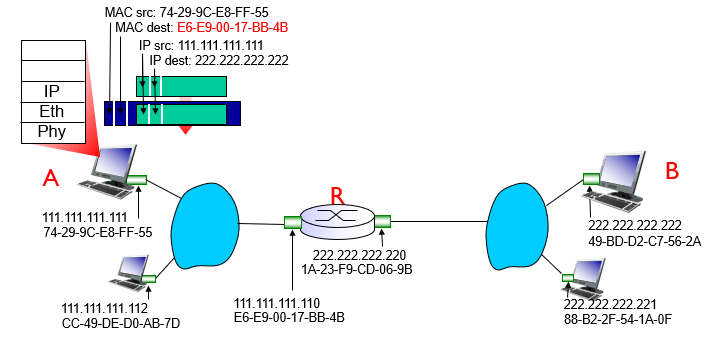
## Addressing: Routing to another LAN

* walkthrough: **send datagram from Host to Host B through Router R**
  + the focus on addressing – at IP layer (datagram) and MAC layer (frame)
  + assume Host A knows Host B’s IP address (through DNS server)
  + assume Host A knows IP address of first hop router, R (through DHCP)
  + assume Host A knows Router R’s MAC address (through ARP)



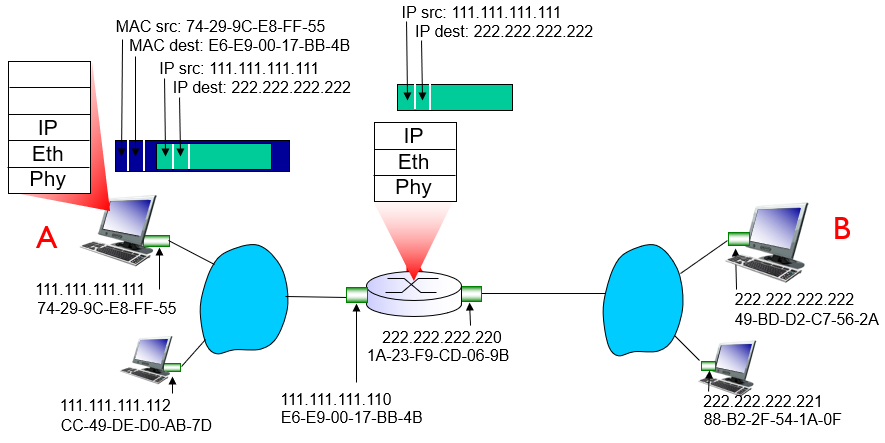
## Step 1: Creating the message at Host A

* Host A creates IP datagram with IP source A, destination B
* Host A uses ARP to get Router R’s MAC address
* Host A creates link-layer frame with R’s MAC address as destination (frame contains   
  A-to-B IP datagram)



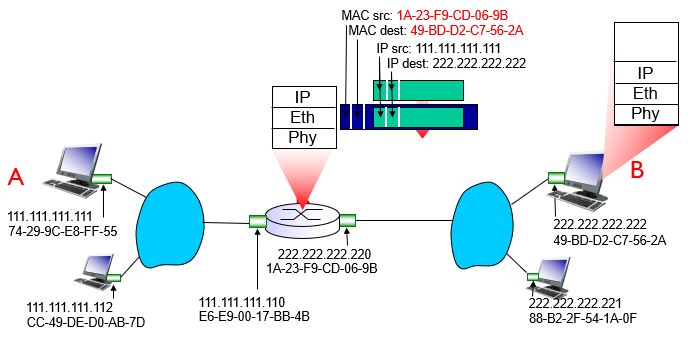
## Step 2: Sending the message through Router R

* frame sent from Host A to Router R
* frame received at Router R, datagram removed, passed up to IP



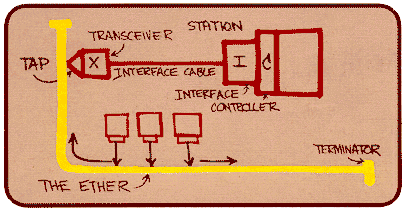
## Step 3: Forwarding the message to Host B

* Router R forwards datagram with IP source Host A, destination Host B
* Router R uses ARP to get Host B’s MAC address
* Router R creates link-layer frame with Host B’s MAC address as destination, frame contains A-to-B IP datagram



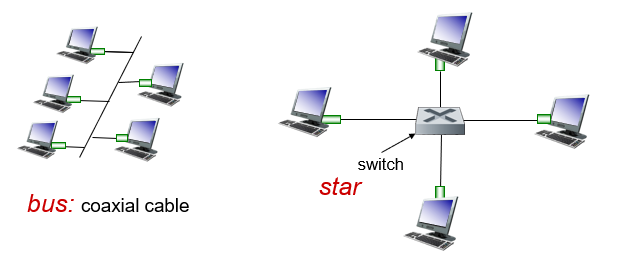
## Ethernet: Rise of Wired Networking

* a family of wired computer networking technologies
  + the “dominant” (and first widely used) LAN technology
  + cheap $20 for NIC
  + simpler, cheaper than token LANs and ATM
  + kept up with speed race: 10 Mbps – 10 Gbps

  
*Ethernet sketch by its inventor, Robert Metcalfe*

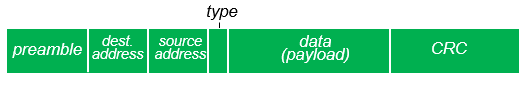
## Ethernet: physical topology

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



## 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 sync receiver, sender clock rates
* **addresses:** 6 byte source, destination MAC addresses
  + frame is received by all adapters on a LAN
  + dropped if address does not match
* **type:** indicates higher layer protocol (mostly IP but others possible, e.g., Novell IPX, AppleTalk)
* **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*

## Ethernet: uses CSMA/CD

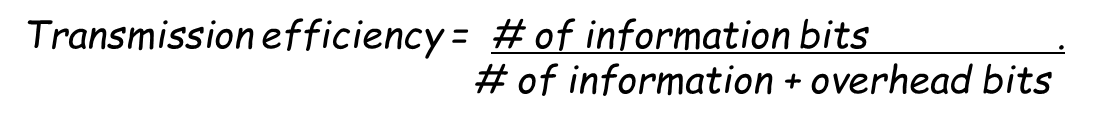
* **A**: sense channel, **if** idle
* **then** {
  + transmit and monitor the channel;
  + **If** detect another transmission
    - **then** {
      * abort and send jam signal;
      * update # collisions;
      * delay as required by exponential backoff algorithm;
      * go to A
    - }
  + **else** {done with the frame; set collisions to zero}
* }
* **else** {wait until ongoing transmission is over and go to A}

## Ethernet: uses 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 x 512 bit transmission times
* after second collision: choose K from {0,1,2,3}…
* after ten or more collisions, choose K from {0,1,2,3,4, …, 1023}

## Transmission Efficiency

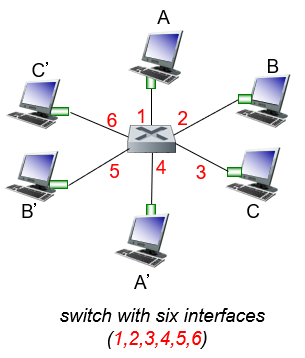
* Formula for transmission efficiency:



* Diagram of frame size effects on throughput

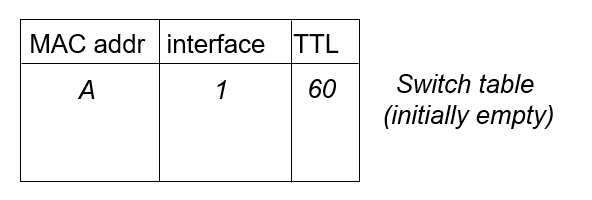
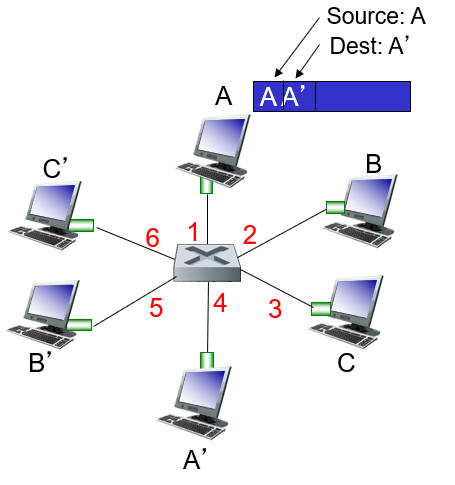
c04f012.eps

## Ethernet Switch

* link-layer (layer 2) device: takes an *active* role and does the following functions
  + filtering: examine incoming frame’s MAC address; decide if the frame should be forwarded or dropped
  + store and send frames to the right interfaces (forwarding)
  + store known MAC addresses in a switch table
* *switching*
  + A-to-B and A’-to-B’ simultaneously, no collisions
  + large number of interfaces
  + often individual host star-connected into switch (no collisions!)
* *transparent*
  + hosts are unaware of presence of switches
* *plug-and-play, self-learning*
  + switches do not need to be configured

## Switch forwarding table

* Question: how does switch know A’ reachable via interface 4, B’ reachable via interface 5?
* Answer: each switch has a **switch table**!
  + each entry contains MAC address of host, interface to reach host, timestamp
  + looks like a routing table!

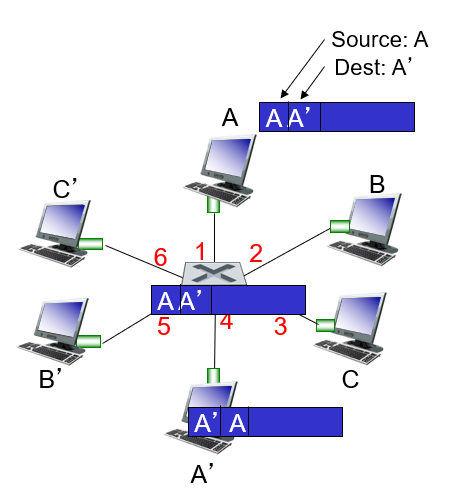


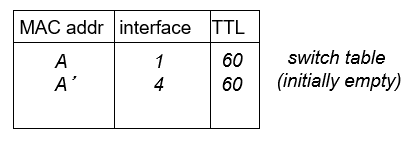
## Switch: self-learning

* A switch is self-learning: its table is built automatically, dynamically, and autonomously (without intervention from net admin or config protocol).
* This capability is accomplished as follows:
  1. The switch table is initially empty
  2. For each incoming frame received on an interface, the switch stores in its table (1) the MAC address in the frame’s source field, (2) interface from which frame arrived, and (3) current time; in effect, the LAN segment of the sender.
  3. The switch deletes an address in the table if no frames are received with that address as the source address after some period of time (the **aging time**).
  4. Thus, if a PC is replaced by another PC (with different adapter), the MAC address of original PC will eventually be purged from the switch table.

## Self-learning, forwarding: example

* A sends frame to destination A’
  + switch learns location of A
* A-frame arrives at switch
  + Location of A’ is unknown
  + flood LAN with ARP requests
* A-frame arrives at A’, which sends reply frame back to A
  + destination A’ location now known by switch
  + selectively send on just one link

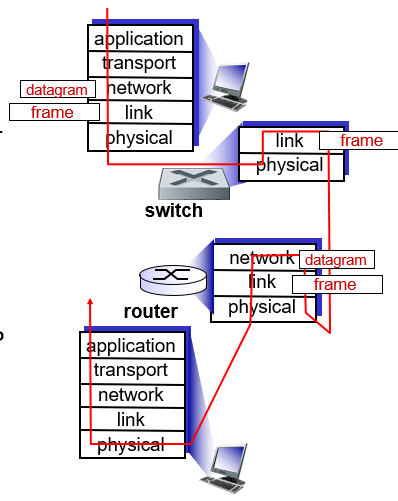




## Properties of Link-Layer Switching

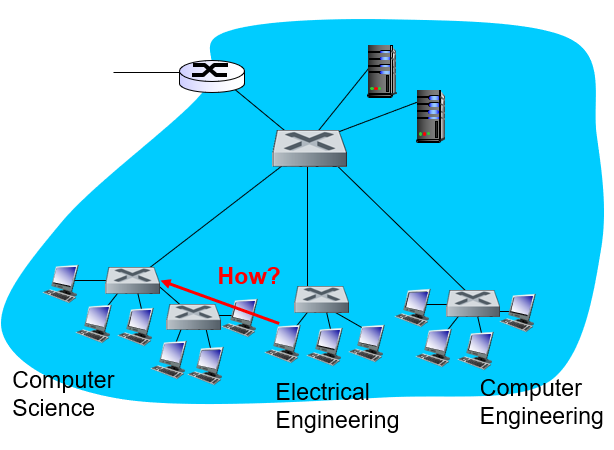
* There are several advantages of using switches, rather than broadcast links such as buses or hub-based star topologies:
  + **Elimination of collisions:** In a LAN built from switches, there is no wasted bandwidth due to collisions! The switches buffer frames and never transmit more than 1 frame on a segment at any 1 time.
  + **Heterogeneous links:** Because a switch isolates one link from another, the different links in the LAN can operate at different speeds and run over different media.
  + **Security:** A switch provides enhanced security; less vulnerable to sniffing.
  + **Management:** A switch also eases network management. For example if an adapter malfunctions and sends frames continually (i.e. a jabbering adapter), a switch can detect the problem and disconnect the jabbering adapter. Switches also gather stats on bandwidth usage, collision rates, etc.

## 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

## VLANs: motivation

* Modern institutional LANs often configured hierarchically, with each department/workgroup having its own switched LAN connected to other switched LANs via a switch hierarchy, which as numerous drawbacks:
  + **Lack of traffic isolation:** broadcast traffic will still traverse the entire institutional network, which limits LAN performance and raises security/privacy concerns.
  + **Inefficient use of switches:** The more groups there are, the more switches are required; if each group were small, a single 96-port switch should be large enough to handle everyone, but this would not provide traffic isolation.
  + **Managing users:** If an employee moves between groups, the physical cabling must be changed to connect the employee to a different switch; employees belong to two or more groups makes this problem even more complex.
* Fortunately, all these difficulties can be handled by a switch that supports virtual LAN capabilities.
* *Case study:* CS user moves office to EE, but wants to connect to CS switch.

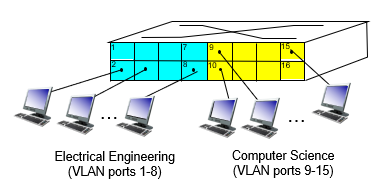
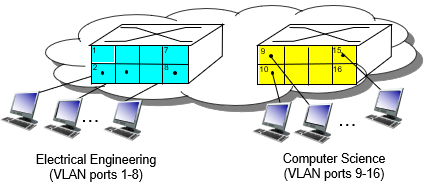
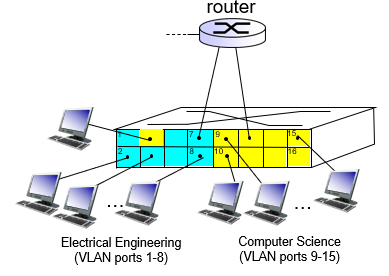


<https://zhuanlan.zhihu.com/p/393964865>

## Virtual LAN (VLAN)

* switch(es) supporting VLAN capabilities can be configured to define multiple virtual LANs over single physical LAN infrastructure.
* Hosts within a VLAN communicate with each other as if they (and no other hosts) were connected to the switch.

## Port-based VLAN

* How it works…
  + The switch’s ports/interfaces are divided into groups by net manager.
  + Each group constitutes a VLAN, with the ports in each VLAN forming a *single* broadcast domain (i.e. broadcast traffic restricted to that VLAN/group).
* *dynamic membership:* ports can be dynamically assigned among VLANs
* *traffic isolation:* frames to/from ports 1-8 can only reach ports 1-8
  + can also define VLAN based on MAC addresses of endpoints, rather than switch port (known as MAC-based VLAN)
* *forwarding between VLANs:* done via routing (just as with separate switches)
  + in practice, vendors sell combined switches plus routers