

Our goals:

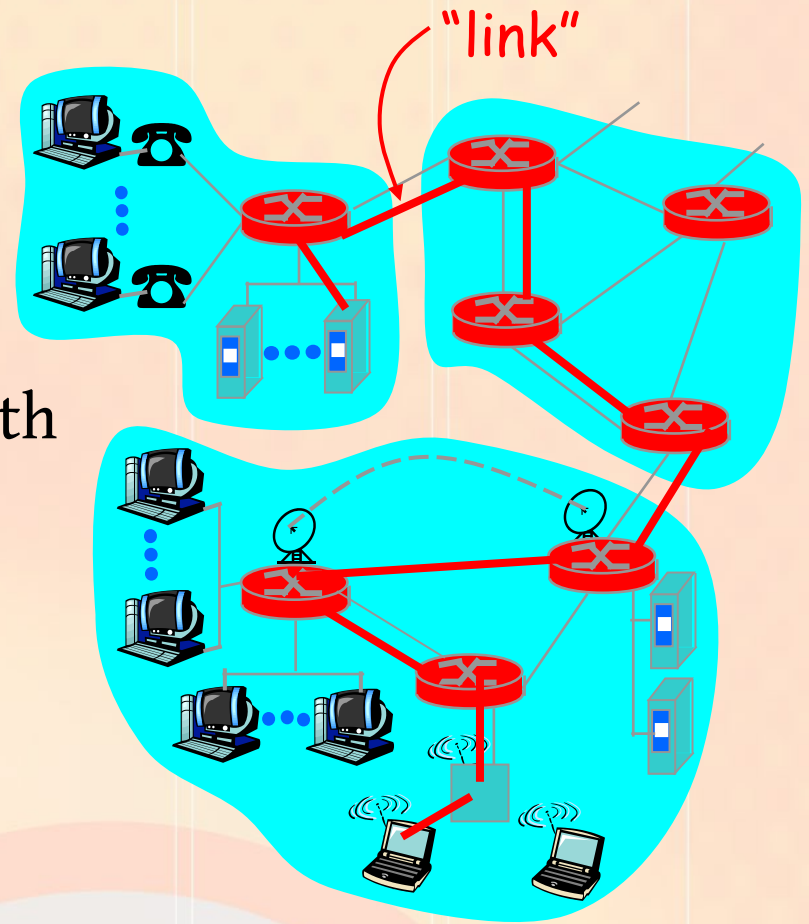
- understand principles behind data link layer services:
 - link layer addressing
- instantiation and implementation of various link layer technologies

- **Introduction and services**
- LAN addresses and ARP
- Ethernet
- Hubs, bridges, and switches

Link Layer: Introduction

Terminology:

- hosts and routers are **nodes** (bridges and switches too)
- communication channels that connect adjacent nodes along communication path are **links**
 - wired links
 - wireless links
- Layer 2-PDU is a **frame**, encapsulating a datagram



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

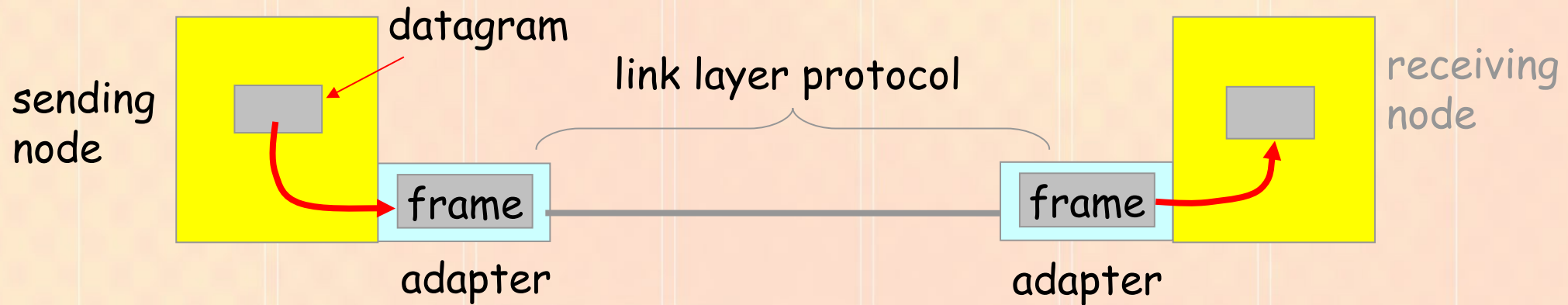
Link layer: context

- 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
 - e.g., may or may not provide reliable data transfer over link

transportation analogy

- trip from Princeton to Lausanne
 - limo: Princeton to JFK
 - plane: JFK to Geneva
 - train: Geneva to Lausanne
- tourist = **datagram**
- transport segment = **communication link**
- transportation mode = **link layer protocol**
- travel agent = **routing algorithm**

- **Framing, link access:**
 - encapsulate datagram into frame, adding header, trailer
 - channel access if shared medium
 - **'physical addresses'** used in frame headers to identify source and destination
 - different from IP address!
- **Reliable delivery between adjacent nodes**
 - seldom used on low bit error link (fibre, twisted pair)
 - wireless links: high error rates
 - Q: why both link-level and end-end reliability?



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

32-bit IP address: (Logical)

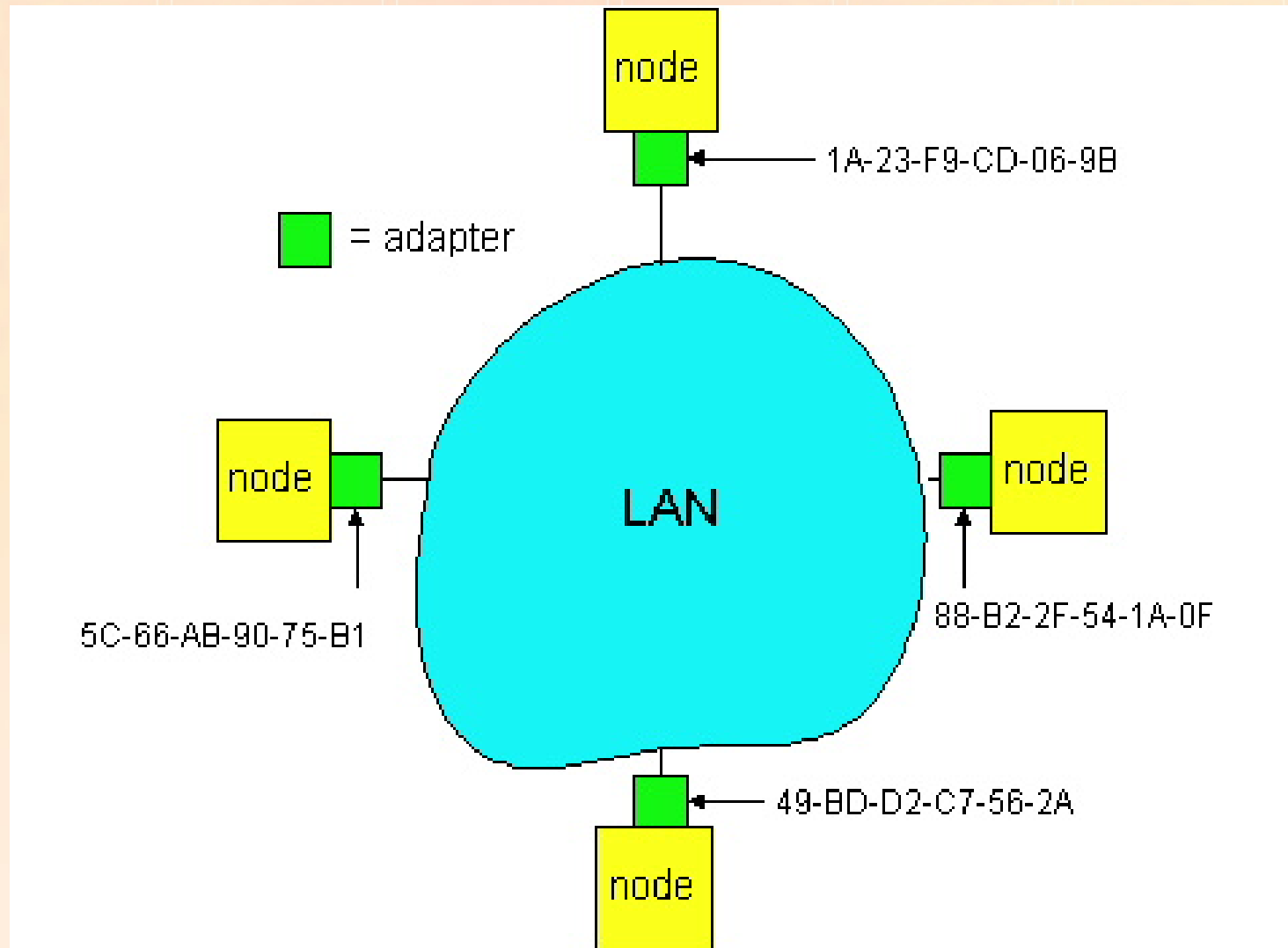
- *network-layer* address
- used to get datagram to destination IP network (recall IP network definition)

48-bit LAN (or MAC 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 hence cannot be changed.

LAN Addresses and ARP

Each adapter on LAN has unique LAN address

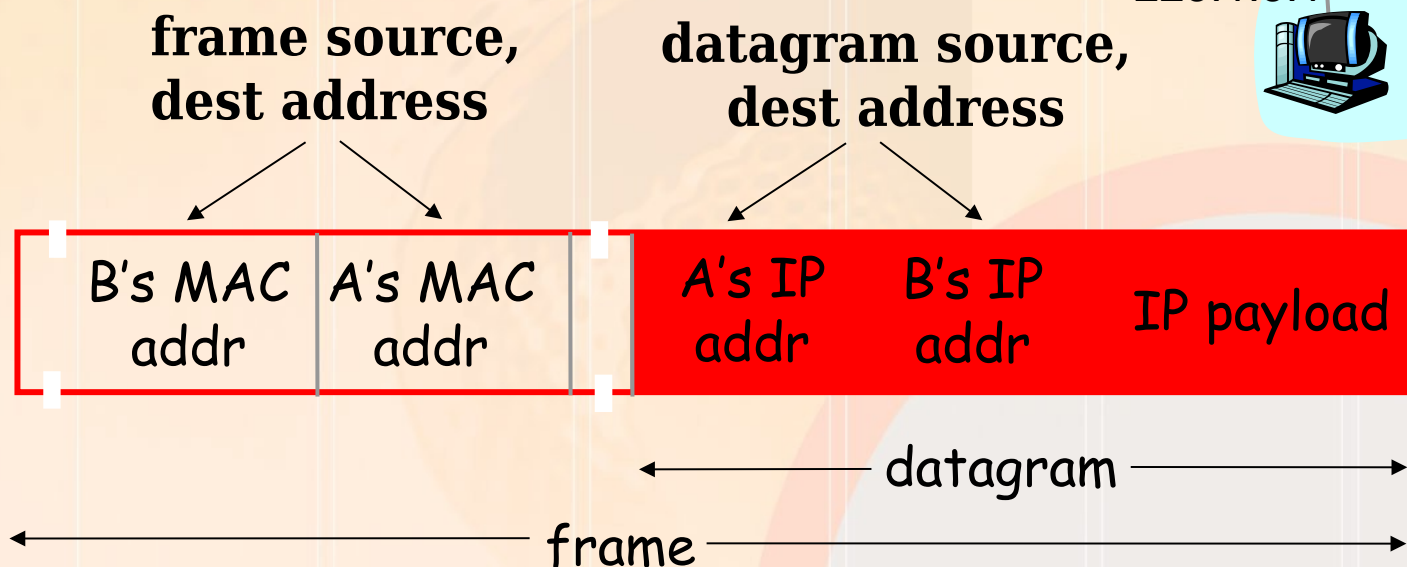
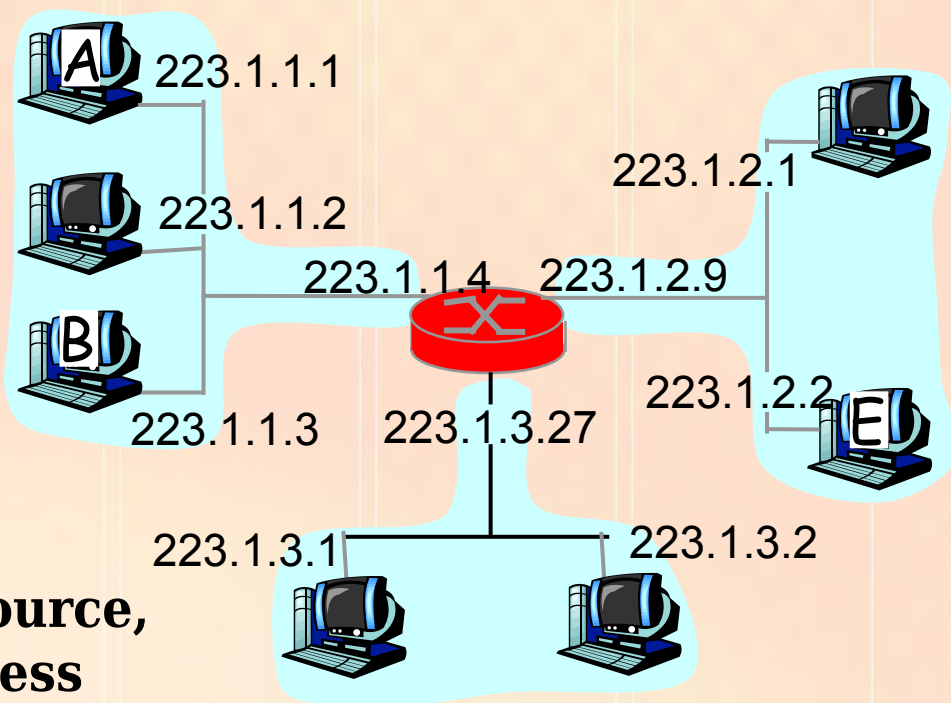


- MAC address allocation administered by IEEE
- manufacturer buys portion of MAC address space (to assure uniqueness)
- Analogy:
 - (a) MAC address: like NIC Number
 - (b) IP address: like postal address
- MAC flat address => portability
 - can move LAN card from one LAN to another
- IP hierarchical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

Recall earlier routing discussion

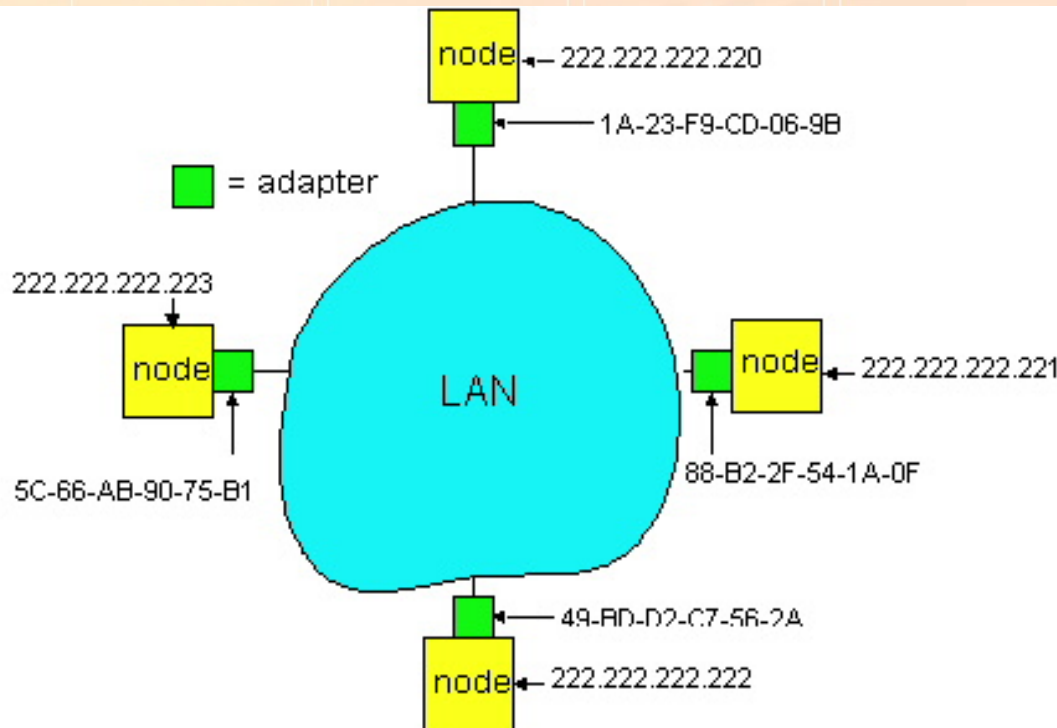
Starting at A, given IP datagram addressed to B:

- look up net. address of B, find B on same net. as A
- link layer send datagram to B inside link-layer frame



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)

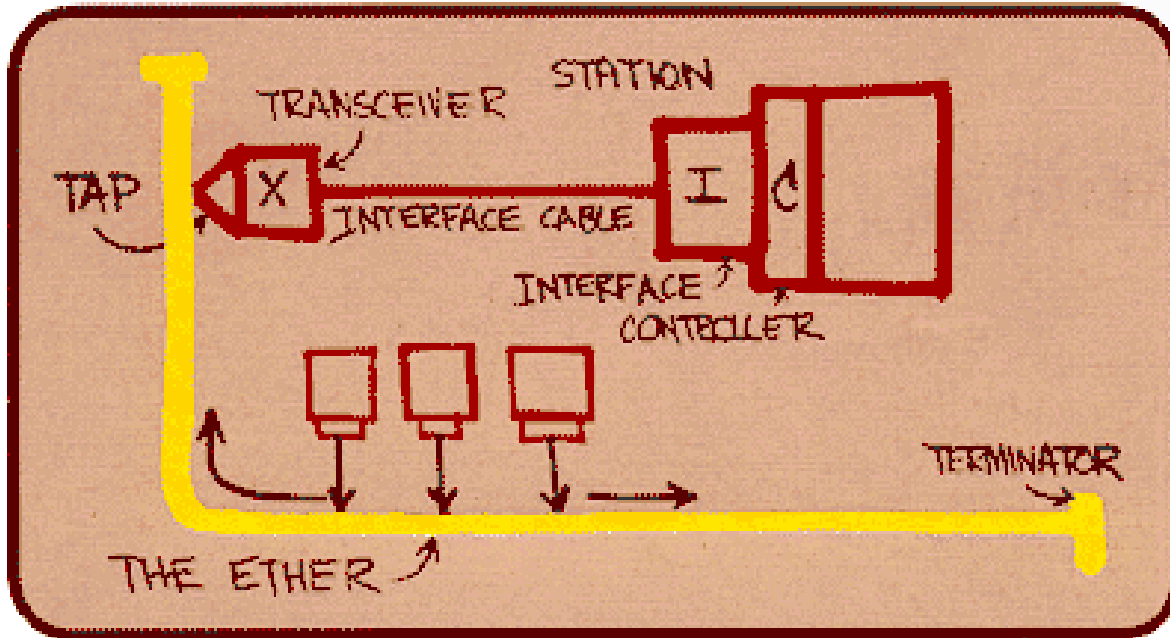
- A wants to send datagram to B, and A knows B's IP address.
- Suppose B's MAC address is not in A's ARP table.
- A **broadcasts** ARP query packet, containing B's IP address
 - all machines on LAN receive ARP query
- B receives ARP packet, replies to A with its (B's) MAC address
 - frame sent to A's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



Ethernet

“dominant” LAN technology:

- cheap \$20 for 100Mbps !
- first widely used LAN technology
- Simpler, cheaper than token LANs and ATM
- Kept up with speed race: 10,100,1000 Mbps, 10 Gbps

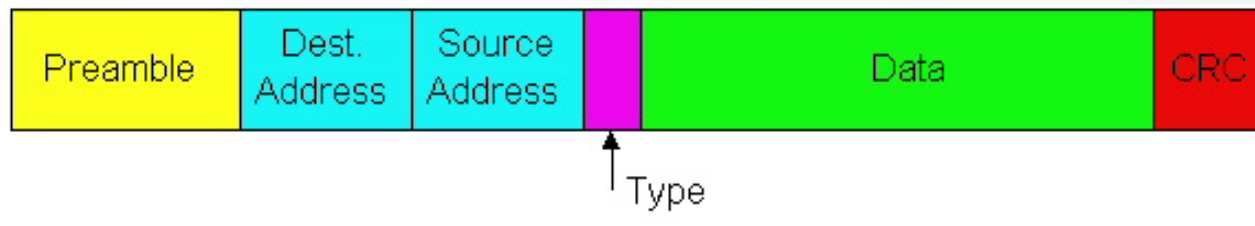


Metcalfe's Ethernet sketch



Ethernet Frame Structure

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



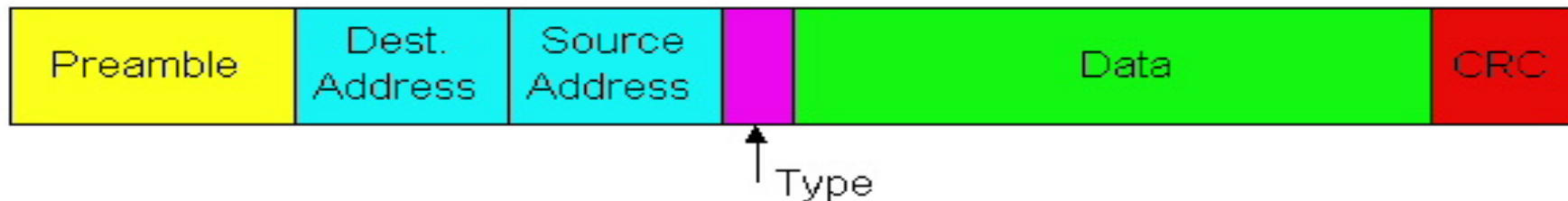
Preamble:

- 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 (e.g. ARP packet), it passes data in frame to network layer protocol
 - otherwise, adapter discards frame
- **Type:** indicates the higher layer protocol, mostly IP but others may be supported such as Novell IPX and AppleTalk)
- **CRC:** checked at receiver, if error is detected, the frame is simply dropped



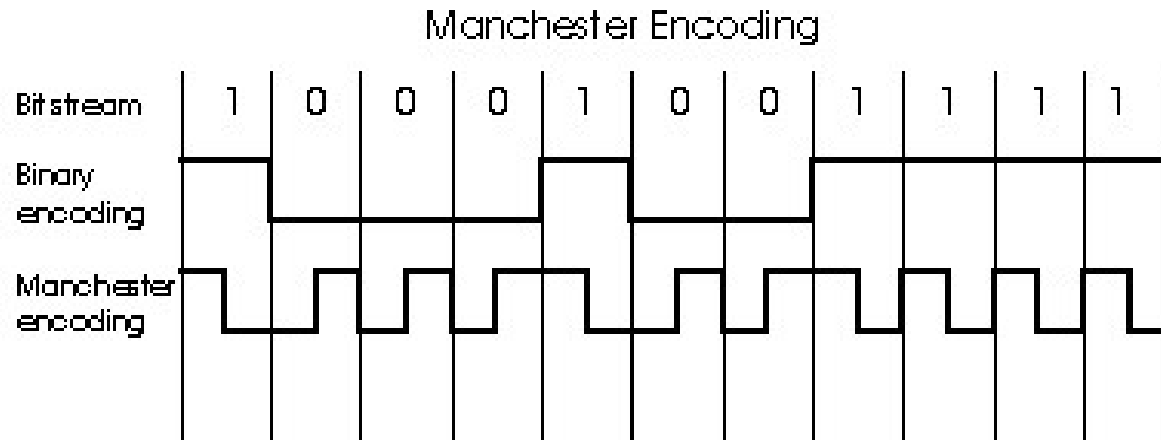


Unreliable, connectionless service

- **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



Manchester Encoding



- Used in 10BaseT, 100BaseT, 1000 BaseT
- Each bit has a transition
- Allows clocks in sending and receiving nodes to synchronize to each other
 - no need for a centralized, global clock among nodes!
- This is physical-layer stuff !



Gigabit Ethernet

- use standard Ethernet frame format
- allows for point-to-point links and shared broadcast channels
- in shared mode, CSMA/CD is used; short distances between nodes to be efficient
- uses hubs, called here “Buffered Distributors”
- Full-Duplex at 1 Gbps for point-to-point links
- 10, 40, 100 Gbps now !!!



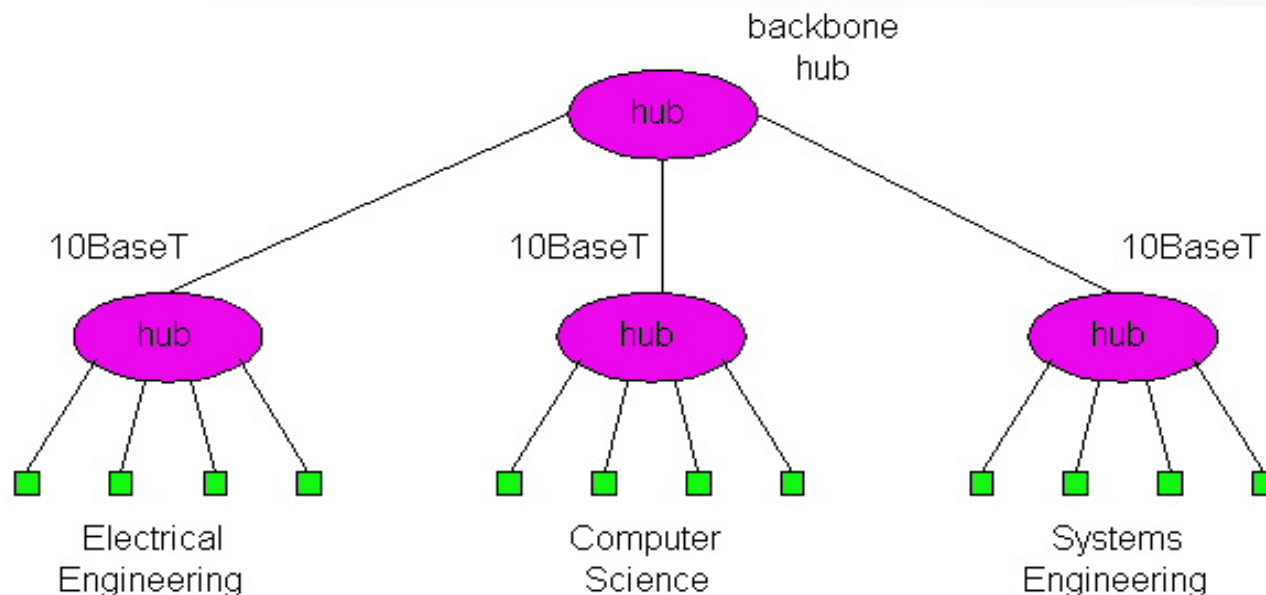
Interconnecting LAN segments

- Hubs
- Bridges
- Switches
 - Remark: switches are essentially multi-port bridges.
 - What we say about **bridges** also holds for **switches!**



Interconnecting with hubs

- Backbone hub interconnects LAN segments
- Extends max distance between nodes
- But individual segment collision domains become one large collision domain
 - if a node in CS and a node EE transmit at same time: collision
- Can't interconnect 10BaseT & 100BaseT





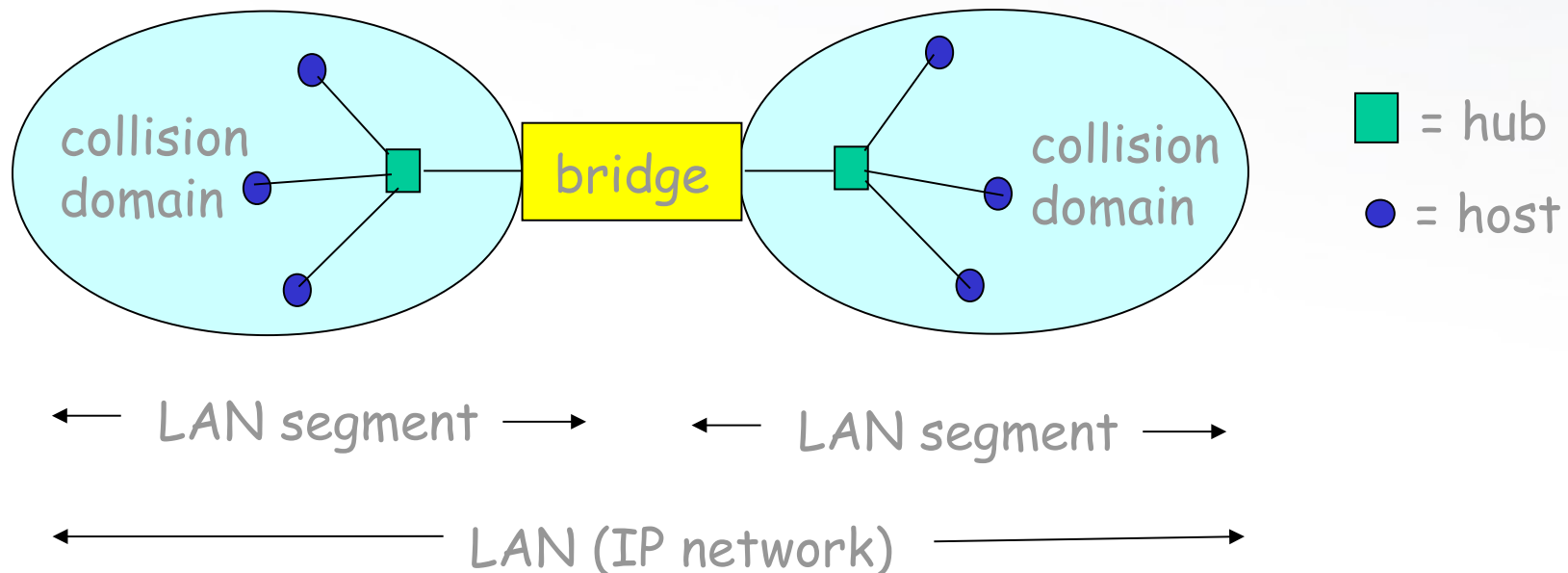
Bridges

- **Link layer device**
 - stores and forwards Ethernet frames
 - examines frame header and **selectively** forwards frame based on MAC destination address
 - when frame is to be forwarded on segment, uses CSMA/CD to access segment
- transparent
 - hosts are unaware of presence of bridges
- plug-and-play, self-learning
 - bridges do not need to be configured



Bridges: traffic isolation

- Bridge installation breaks LAN into LAN segments
- bridges **filter** packets:
 - same-LAN-segment frames not usually forwarded onto other LAN segments
 - segments become separate **collision domains**





Self learning

- A bridge has a **bridge or forwarding table**
- entry in bridge table:
 - (*Node LAN Address, Bridge Interface, Time Stamp*)
 - stale entries in table dropped (TTL can be 60 min)
- bridges **learn** which hosts can be reached through which interfaces
 - when frame received, bridge “learns” location of sender: incoming LAN segment
 - records sender/location pair in forwarding table



Filtering / Forwarding

When bridge receives a frame:

index bridge table using MAC dest address

if entry found for destination

then{

if dest on segment from which frame arrived

then drop the frame

else forward the frame on interface indicated

}

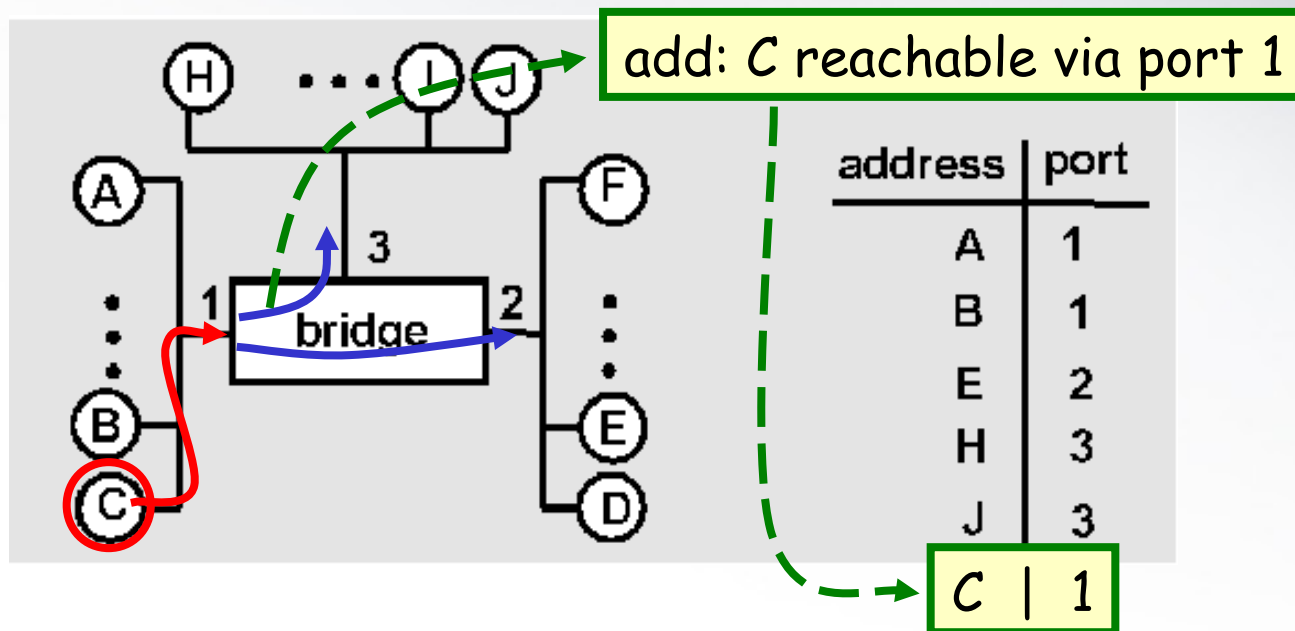
else flood

*forward on all but the interface
on which the frame arrived*



Bridge Learning: example

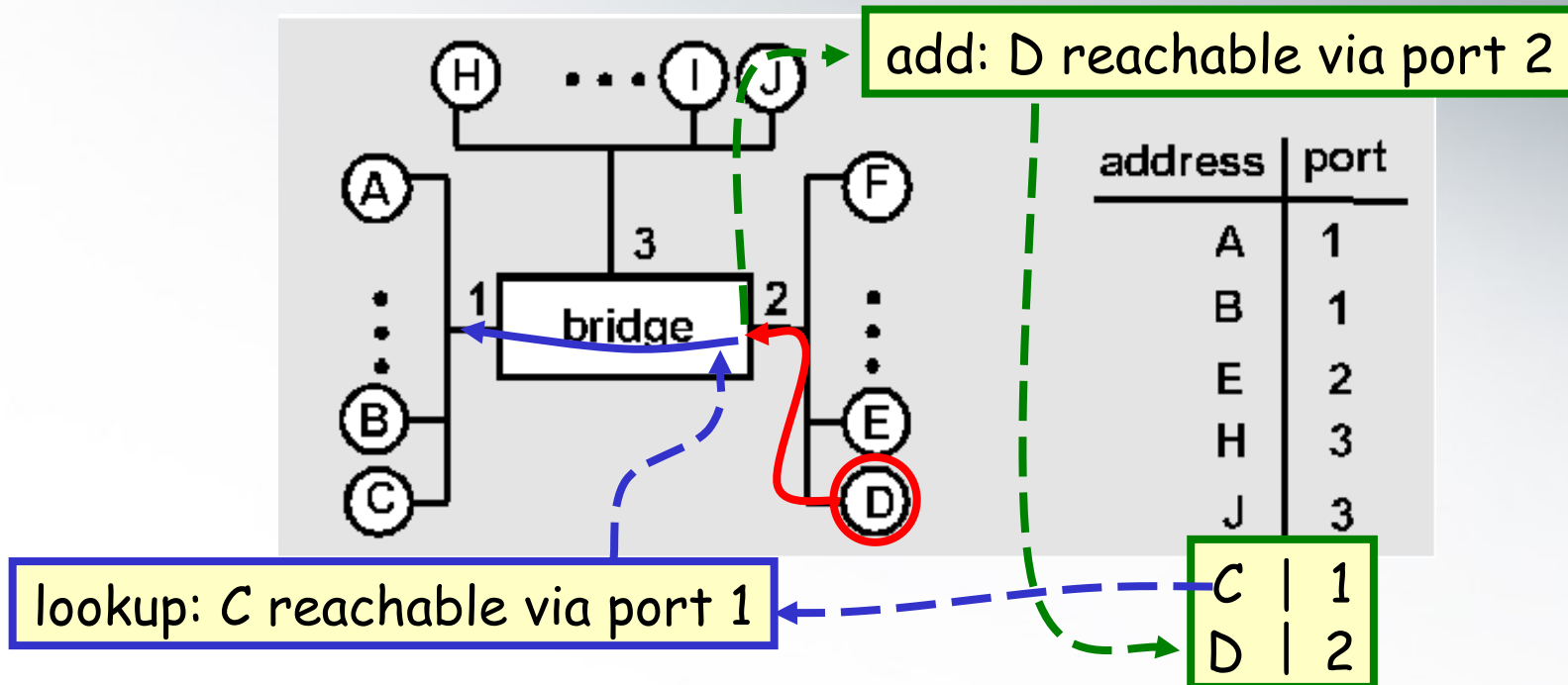
Suppose C sends frame to D and D replies back with frame to C.



- Bridge receives frame from C
 - Insert entry in bridge table that C is on interface 1
 - because D is not in table, bridge sends frame into interfaces 2 and 3 (flooding)
- frame received by D



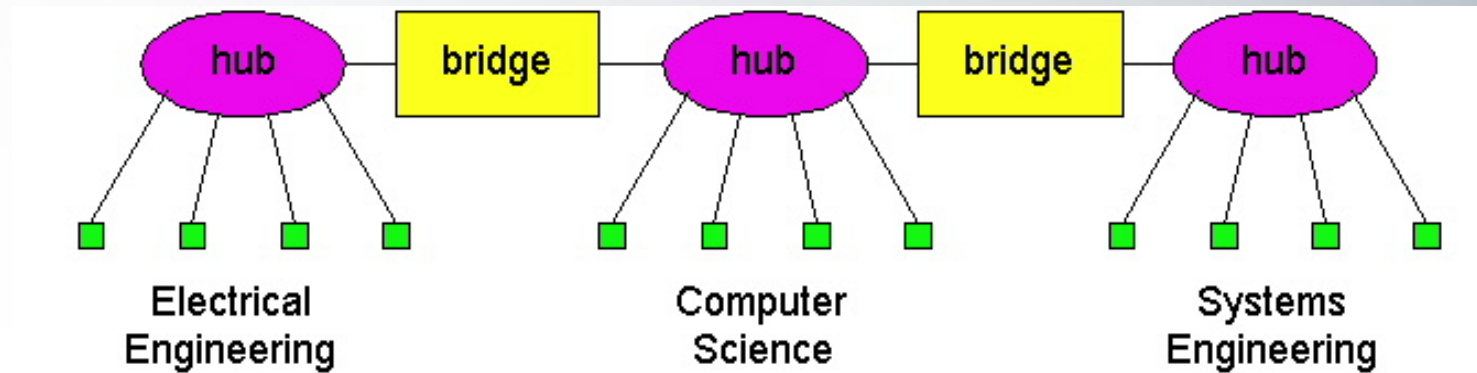
Bridge Learning: example



- D generates frame for C, sends
- bridge receives frame
 - Insert entry in bridge table that D is on interface 2
 - bridge knows C is on interface 1, so *selectively* forwards frame to interface 1



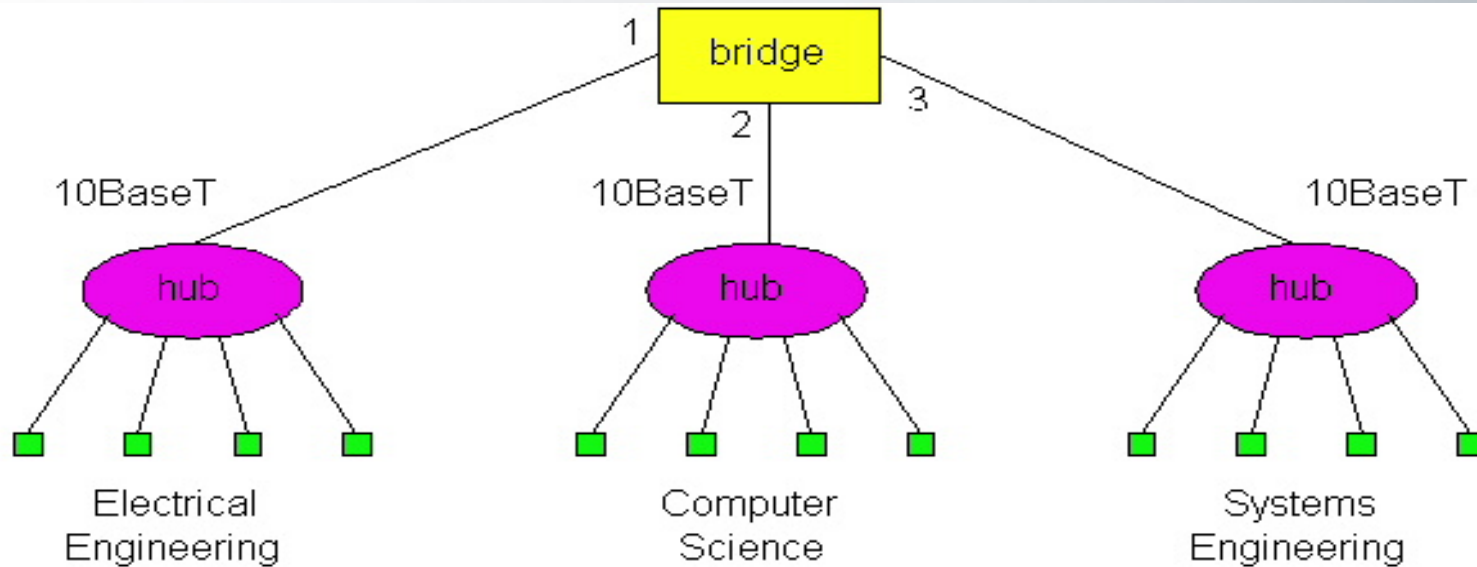
Interconnection without backbone



- Not recommended for two reasons:
 - single point of failure at Computer Science hub
 - all traffic between EE and SE must pass through the CS segment



Backbone configuration



Recommended !



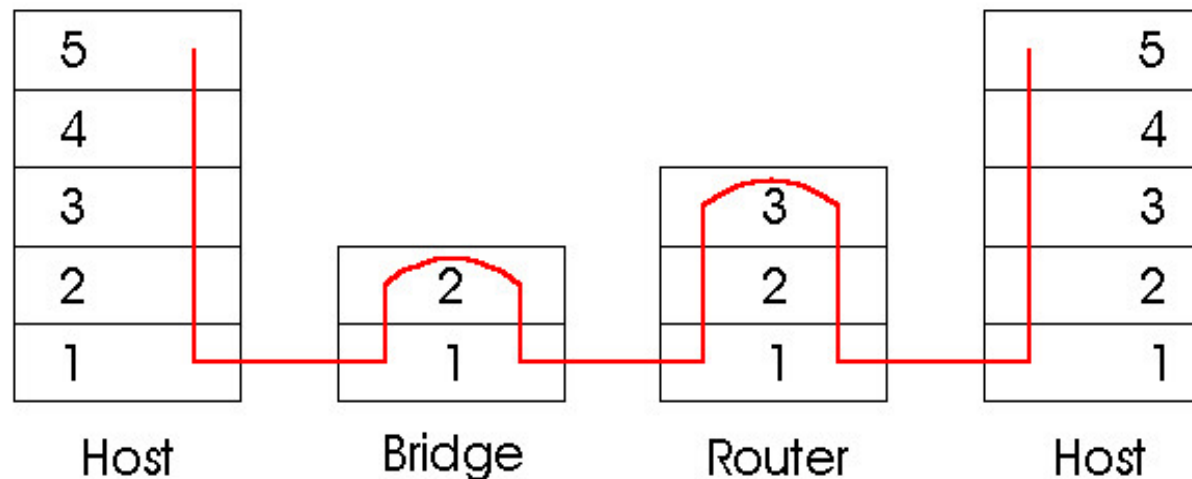
Some Bridge features

- Isolates collision domains resulting in higher total maximum throughput
- limitless number of nodes and geographical coverage
- Can connect different Ethernet types
- Transparent (“plug-and-play”): no configuration necessary



Bridges vs. Routers

- **Both** store-and-forward devices
 - routers: network layer devices (examine network layer headers)
 - bridges are link layer devices
- **Routers** maintain routing tables, implement routing algorithms
- **Bridges** maintain bridge tables, implement filtering, learning and spanning tree algorithms





Routers vs. Bridges

Bridges + and -

- + Bridge operation is simpler requiring less packet processing
- + Bridge tables are self learning
- All traffic confined to spanning tree, even when alternative bandwidth is available
- Bridges do not offer protection from *broadcast storms*



Routers vs. Bridges

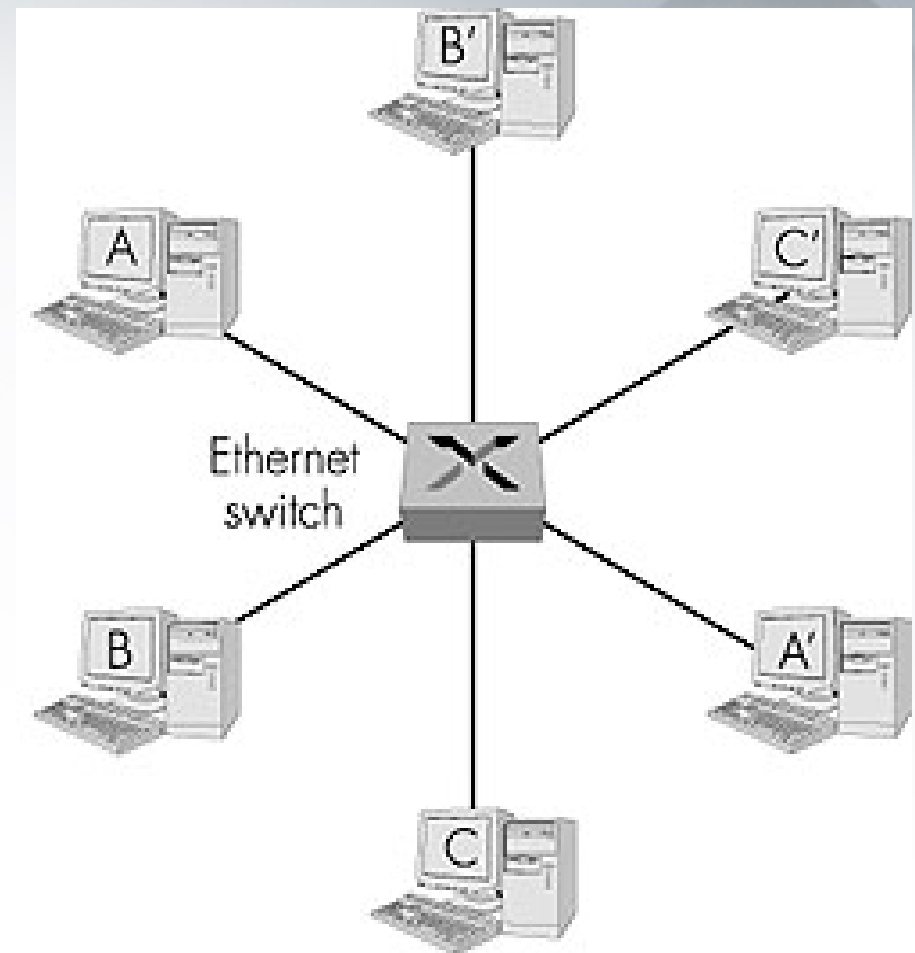
Routers + and -

- + arbitrary topologies can be supported, cycling is limited by TTL counters (and good routing protocols)
 - + provide protection against broadcast storms
 - require IP address configuration (not plug and play)
 - require higher packet processing
-
- bridges do well in small (few hundred hosts) while routers used in large networks (thousands of hosts)



Ethernet Switches

- Essentially a multi-interface bridge
- layer 2 (frame) forwarding, filtering using LAN addresses
- **Switching:** A-to-A' and B-to-B' simultaneously, no collisions
- large number of interfaces
- often: individual hosts, star-connected into switch
 - Ethernet, but no collisions!



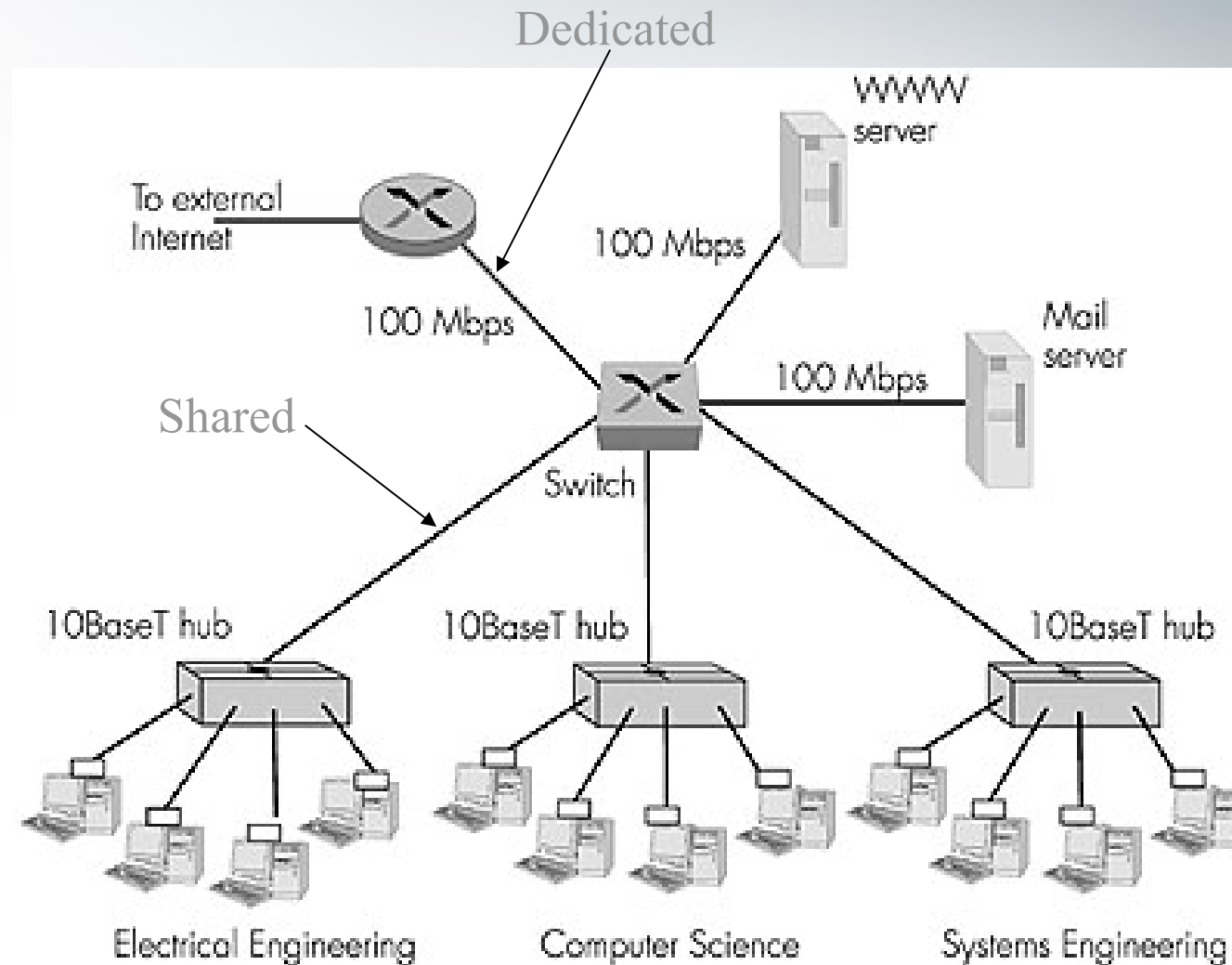


Ethernet Switches

- **cut-through switching:** frame forwarded from input to output port without awaiting for assembly of entire frame
 - slight reduction in latency
- **Store-N-Forward switching:** frame buffered completely and error-checked (via CRC) before being forwarded.



A typical LAN (IP network)





Summary comparison

hubs

bridges

routers

switches

traffic

no

yes

yes

yes

isolation

plug & play

yes

yes

no

yes

optimal

no

no

yes

no

routing

cut

yes

no

no

yes

through