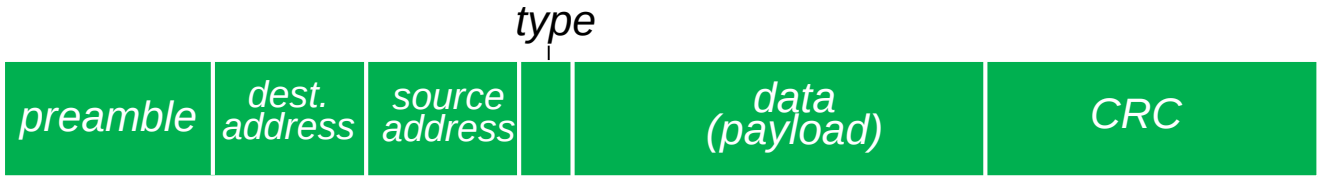
# Ethernet Switch

- 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 are unaware of presence of switches
- *plug-and-play, self-learning*
  - switches do not need to be configured

# Ethernet Frame Structure

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

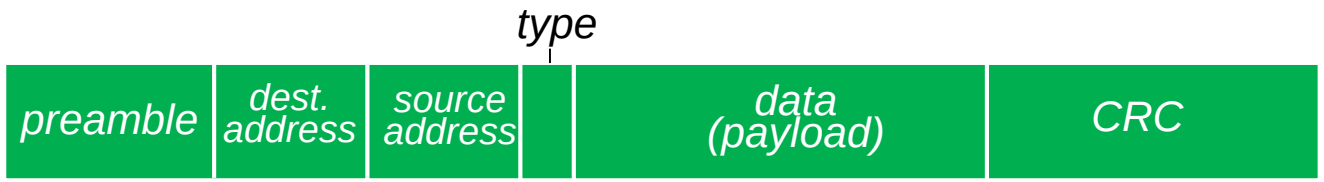



*preamble:*

- 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 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)
- *CRC:* cyclic redundancy check at receiver
  - error detected: frame is dropped



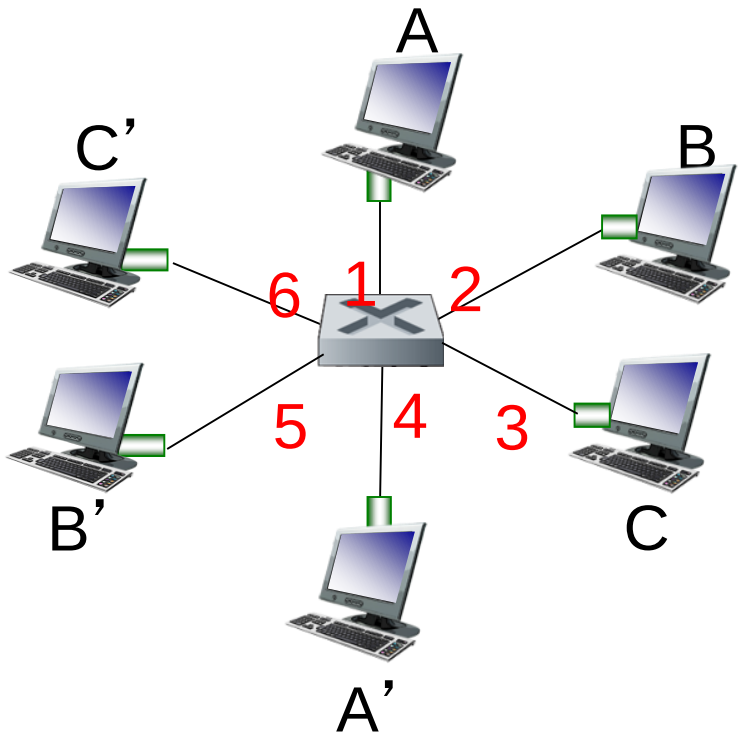
# Ethernet: Unreliable, Connectionless

- *connectionless:* no handshaking between sending and receiving NICs
- *unreliable:* receiving NIC doesn't send acks or nacks 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*

**Will discuss Ethernet MAC CSMA/CD protocol later!**

# Switch: *Multiple* Simultaneous Transmissions

- hosts have dedicated, direct connection to switch
- switches buffer packets
- Ethernet protocol used on *each* incoming link, but 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)*

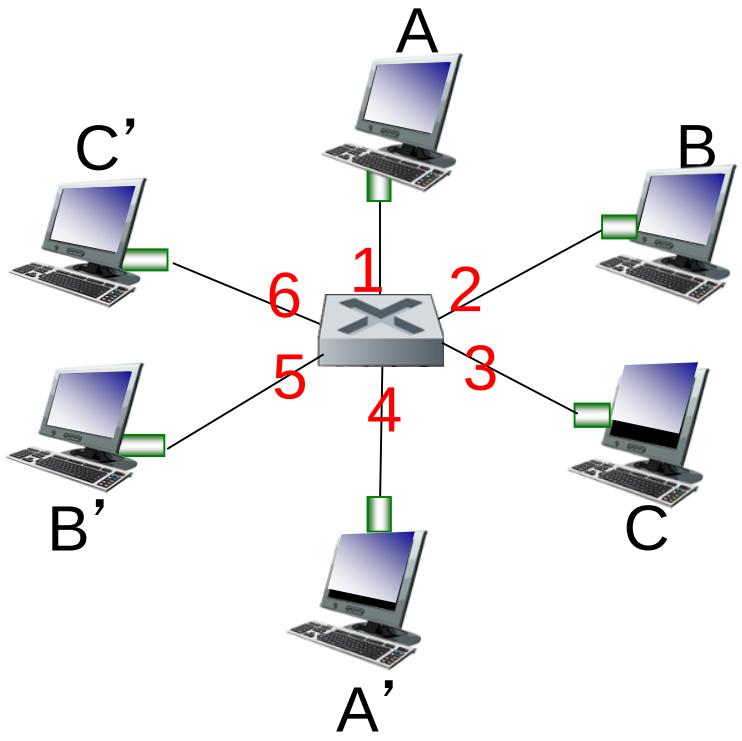
# Switch Forwarding Table

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?



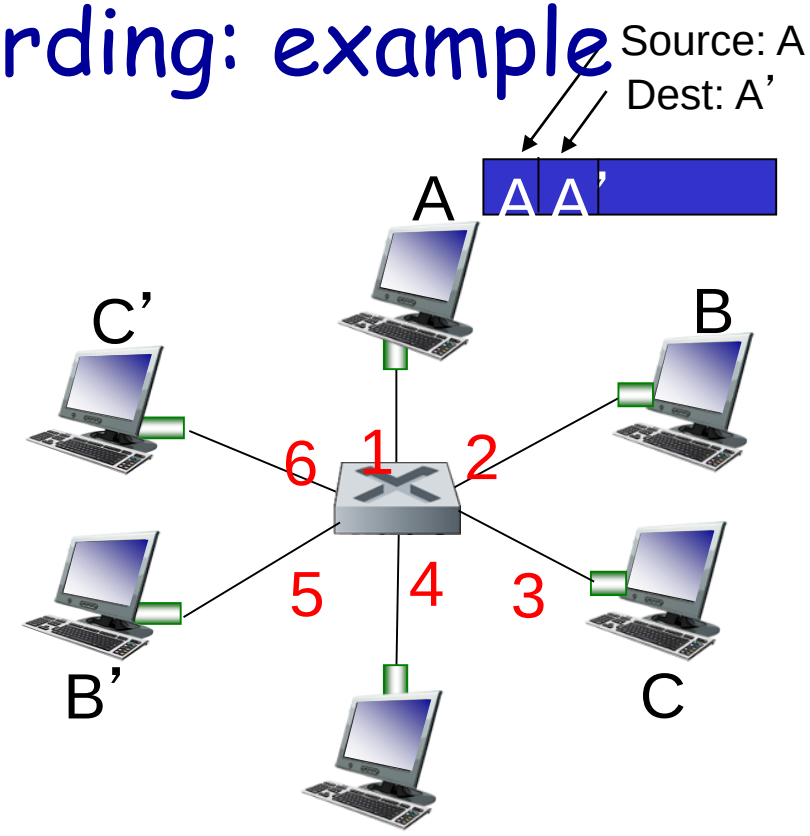

*switch with six interfaces (1,2,3,4,5,6)*

# Self Learning

- A bridge/switch has a forwarding (or switch) table
- entry in forwarding table:
  - (MAC Address, Interface, Time Stamp)
  - stale entries in table dropped (TTL can be 60 min)
- Bridge/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 forwarding table

# Self-learning, forwarding: example

- 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)*

# Filtering/Forwarding

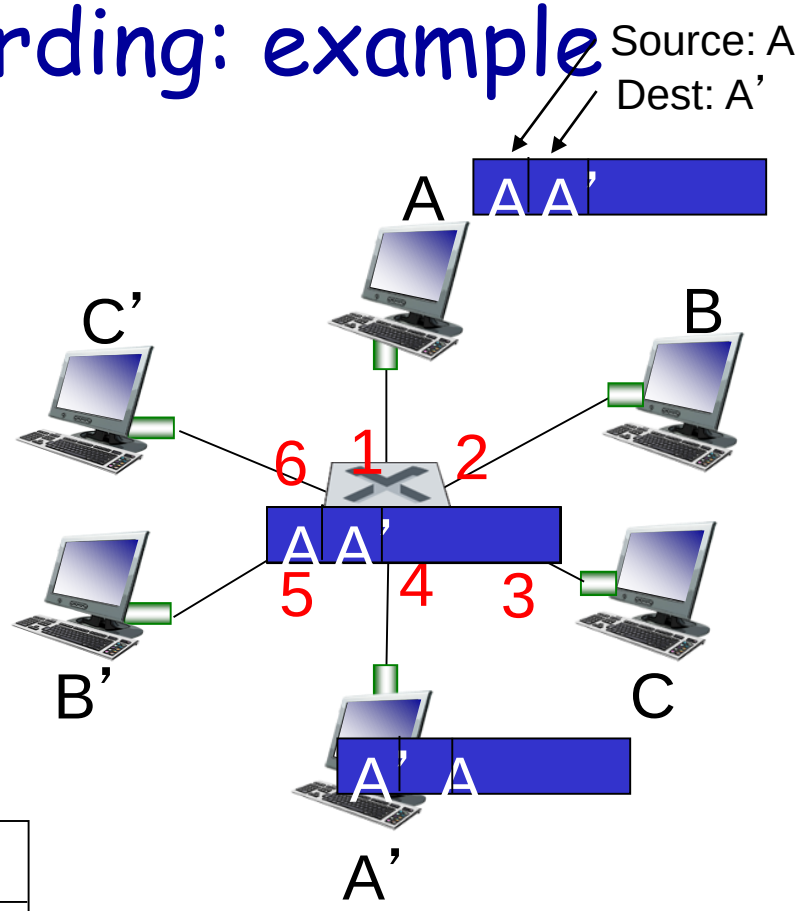
when frame received at switch:

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

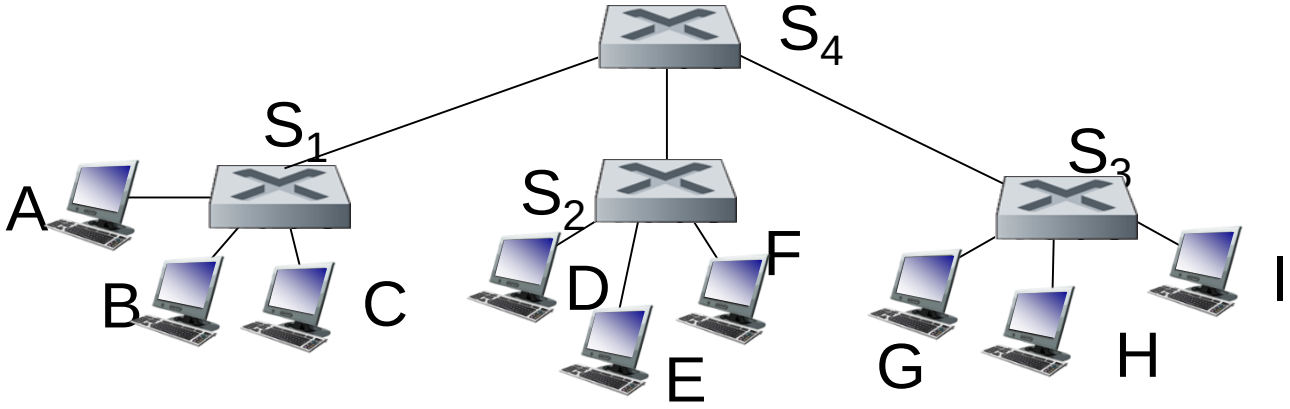



| MAC addr | interface | TTL |
|---|---|---|
| *A* | *1* | *60* |
| *A'* | *4* | *60* |

*switch table (initially empty)*

# Interconnecting switches

self-learning switches can be connected together:

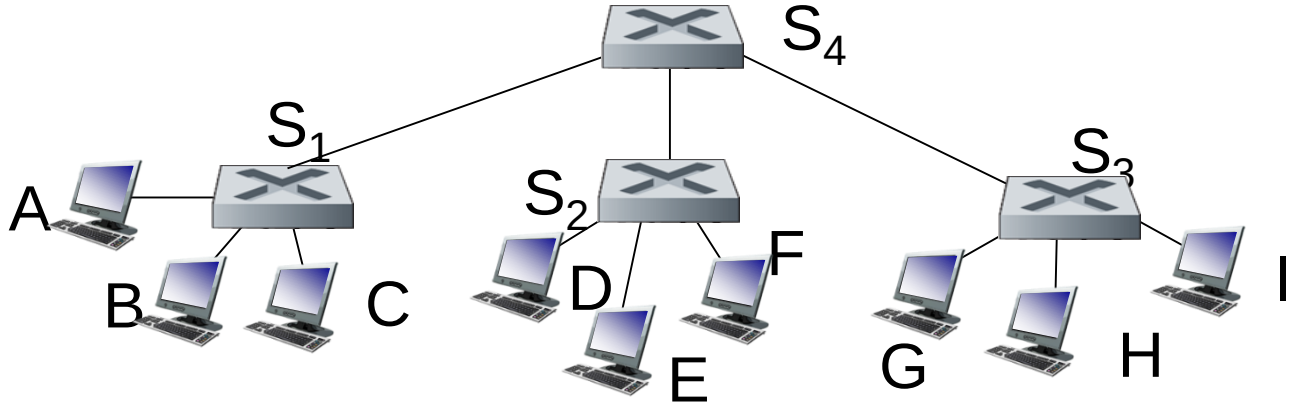



Q: sending from A to G - how does $S_1$ know to forward frame destined to G via $S_4$ and $S_3$?

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

# Self-learning multi-switch example

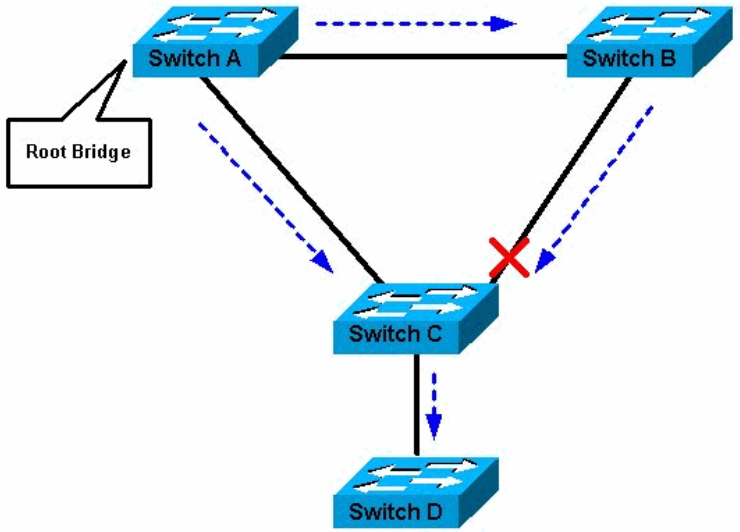
Suppose C sends frame to I, I responds to C




- Q: show switch tables and packet forwarding in $S_1$, $S_2$, $S_3$, $S_4$

# Spanning Tree Protocol

- for increased reliability, desirable to have redundant, alternative paths from source to destination
- with multiple paths, cycles result - switches may multiply and forward frame forever
- solution: organize switches in a spanning tree by disabling subset of interfaces



# Switch Spanning Tree Algorithm:
## Algorhyme

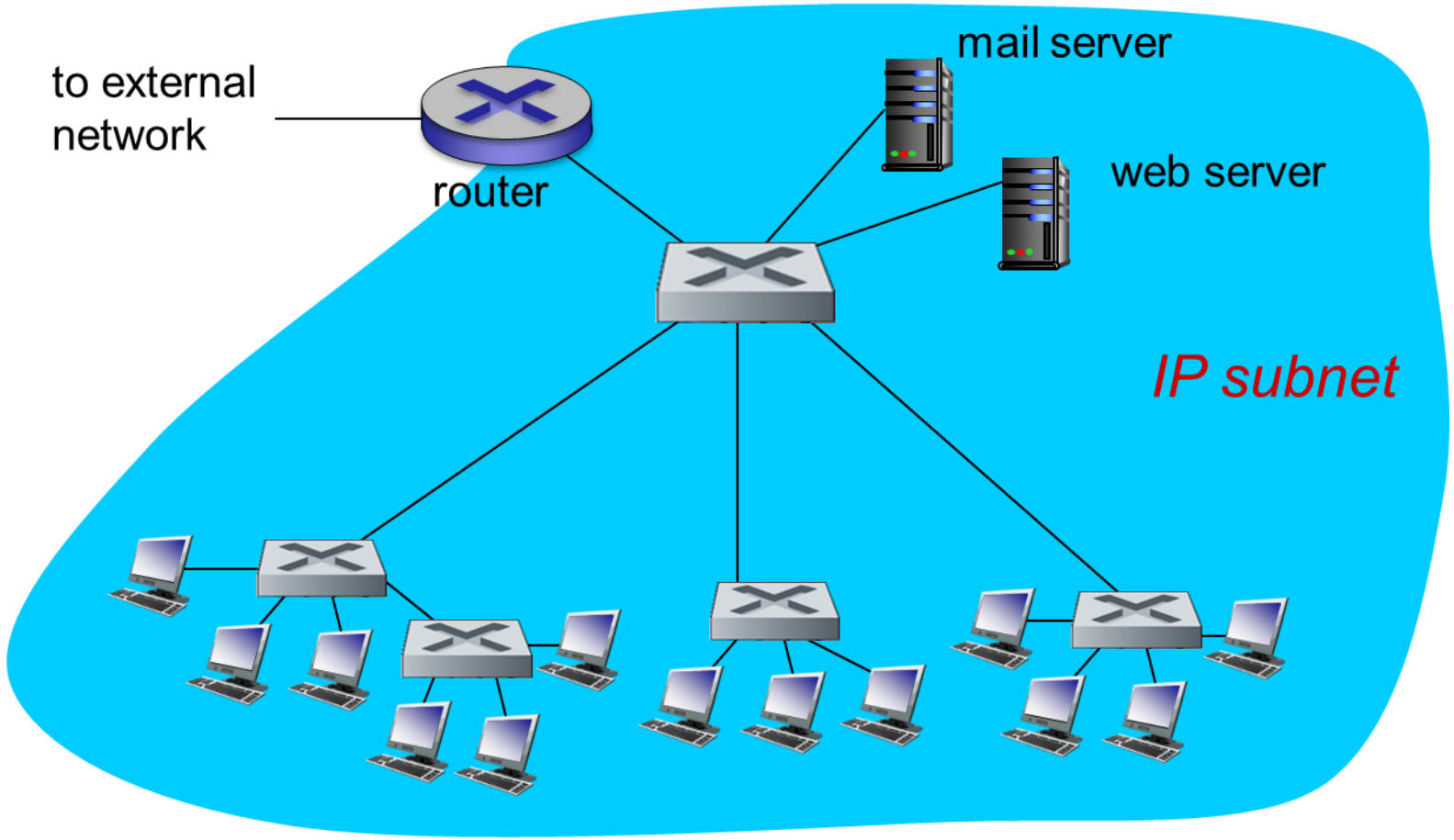
I think that I shall never see
A graph more lovely than a tree.
A tree whose crucial property
Is loop-free connectivity.
A tree that must be sure to span
So packets can reach every LAN.
First, the root must be selected.
By ID, it is elected.
Least cost paths from root are traced.
In the tree, these paths are placed.
A mesh is made by folks like me,
Then bridges find a spanning tree

-- Radia Perlman

# Some Switch Features

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

# Institutional Network



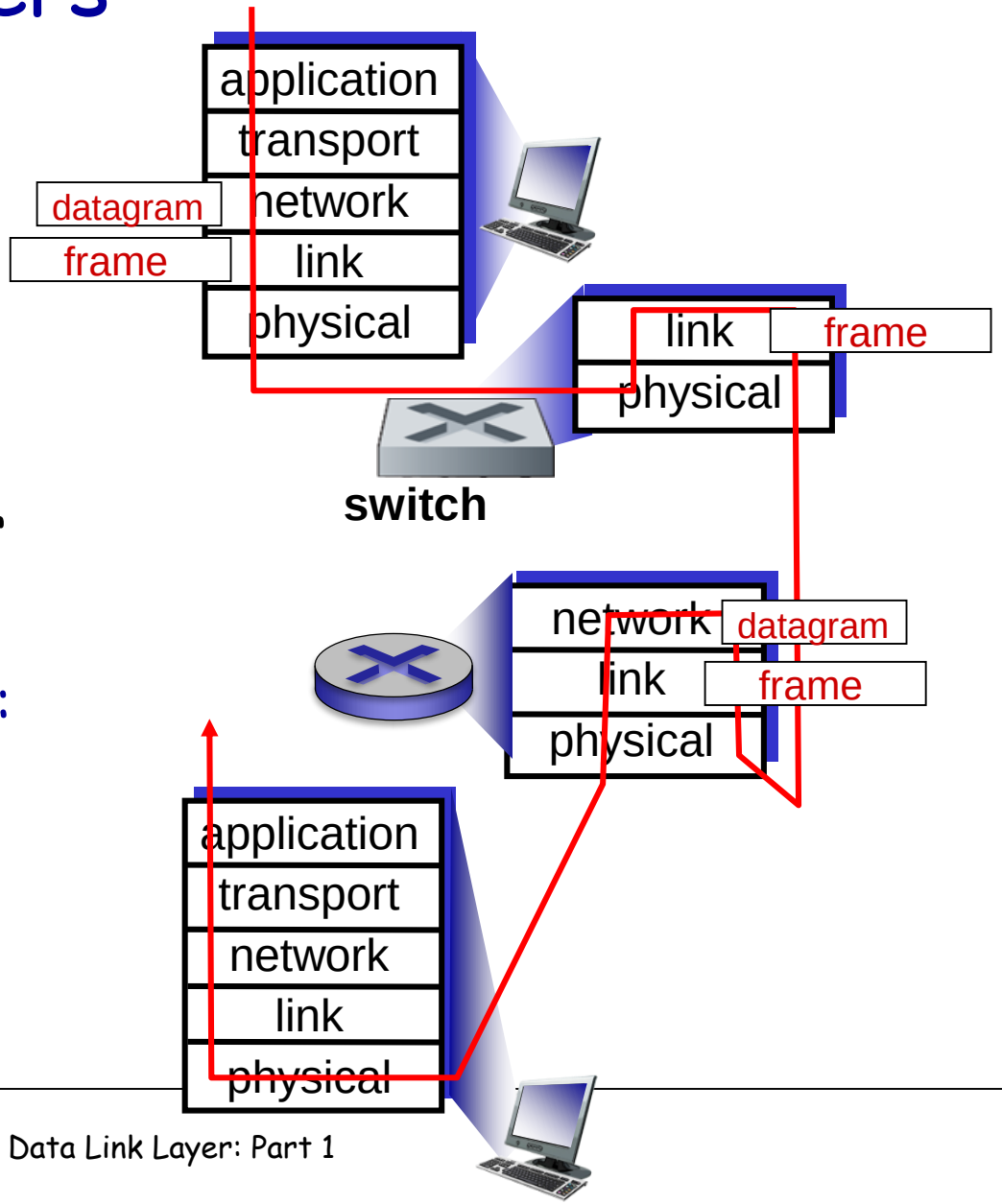
# Switches vs. Routers

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



# Routers vs. Switches

Switches+ and -

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

# Routers vs. Switches

Routers + and -

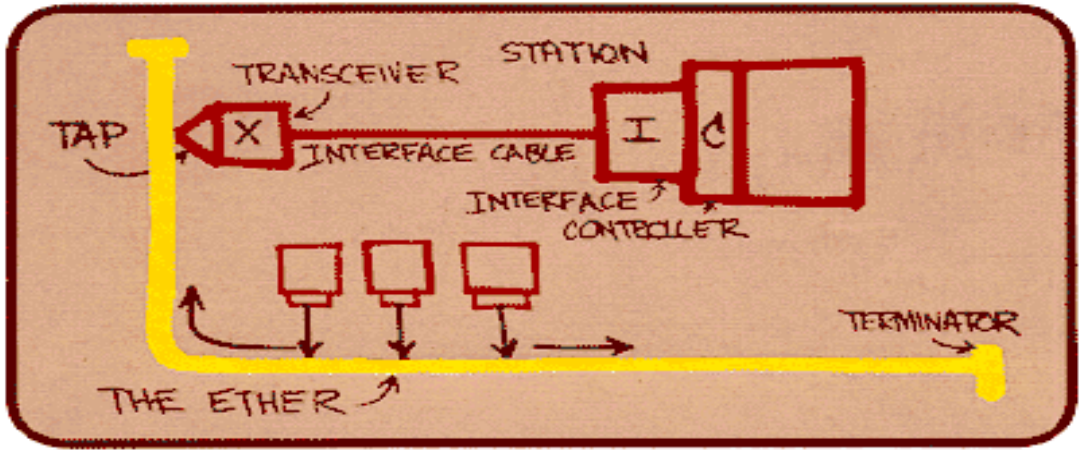
+ arbitrary topologies can be supported, cycling is limited by TTL counters (and good routing protocols)
+ provide protection against broadcast storms
- require IP address configuration (not plug and play)
- require higher packet processing

- switches do well in small (few hundred hosts) while routers used in large networks (thousands of hosts)

# Ethernet

"Dominant" LAN technology today:

- cheap $20 or less for 100 Mbps or even 1Gbps!
- first widely used LAN technology
- Simpler, cheaper than alternative technologies such as token ring LANs
- Kept up with speed race: 10, 100, 1 Gbps, 10 Gbps, 40 Gbps, and now 100 Gbps

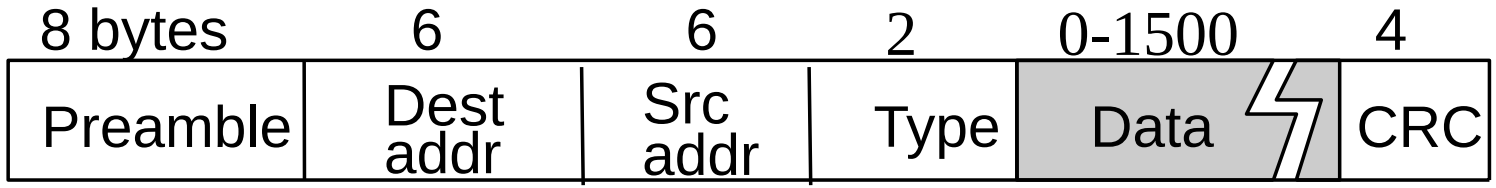



Metcalfe's Ethernet sketch

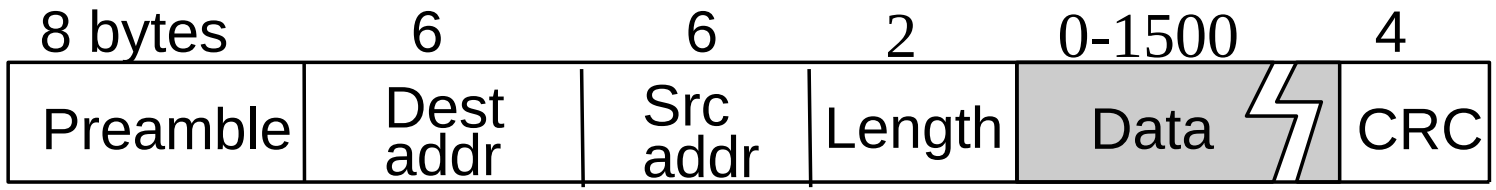
# Ethernet Frame Format

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

DIX frame format




IEEE 802.3 format




- Ethernet has a maximum frame size: data portion <=1500 bytes
- It has imposed a minimum frame size: 64 bytes (excluding preamble)
  If data portion <46 bytes, pad with "junk" to make it 46 bytes

Q: Why minimum frame size in Ethernet?

# Fields in Ethernet Frame Format

- Preamble:
  - 7 bytes with pattern 10101010 followed by one byte with pattern 10101011 (SoF: start-of-frame)
  - used to synchronize receiver, sender clock rates, and identify beginning of a frame
- 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 but others may be supported such as Novell IPX and AppleTalk)
  - 802.3: Length gives data size; "protocol type" included in data
- CRC: checked at receiver, if error is detected, the frame is simply dropped

# Ethernet and IEEE 802.3

1-persistent CSMA/CD

- Carrier sense: station listens to channel first
  - Listen before talking
- If idle, station may initiate transmission
  - Talk if quiet
- Collision detection: continuously monitor channel
  - Listen while talking
- If collision, stop transmission
  - One talker at a time

# Ethernet CSMA/CD Algorithm

1. Adaptor gets datagram from and 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 transmits entire frame without detecting another transmission, the adapter is done with frame ! Signal to network layer "transmit OK"
4. If adapter detects another transmission while transmitting, aborts and sends jam signal
5. 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
6. Quit after 16 attempts, signal to network layer "transmit error"

# Ethernet's CSMA/CD (more)

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

Bit time: .1 microsec for 10 Mbps Ethernet ;
for K=1023, wait time is about 50 msec

See/interact with Java applet on AWL Web site: highly recommended !

Exponential Backoff:

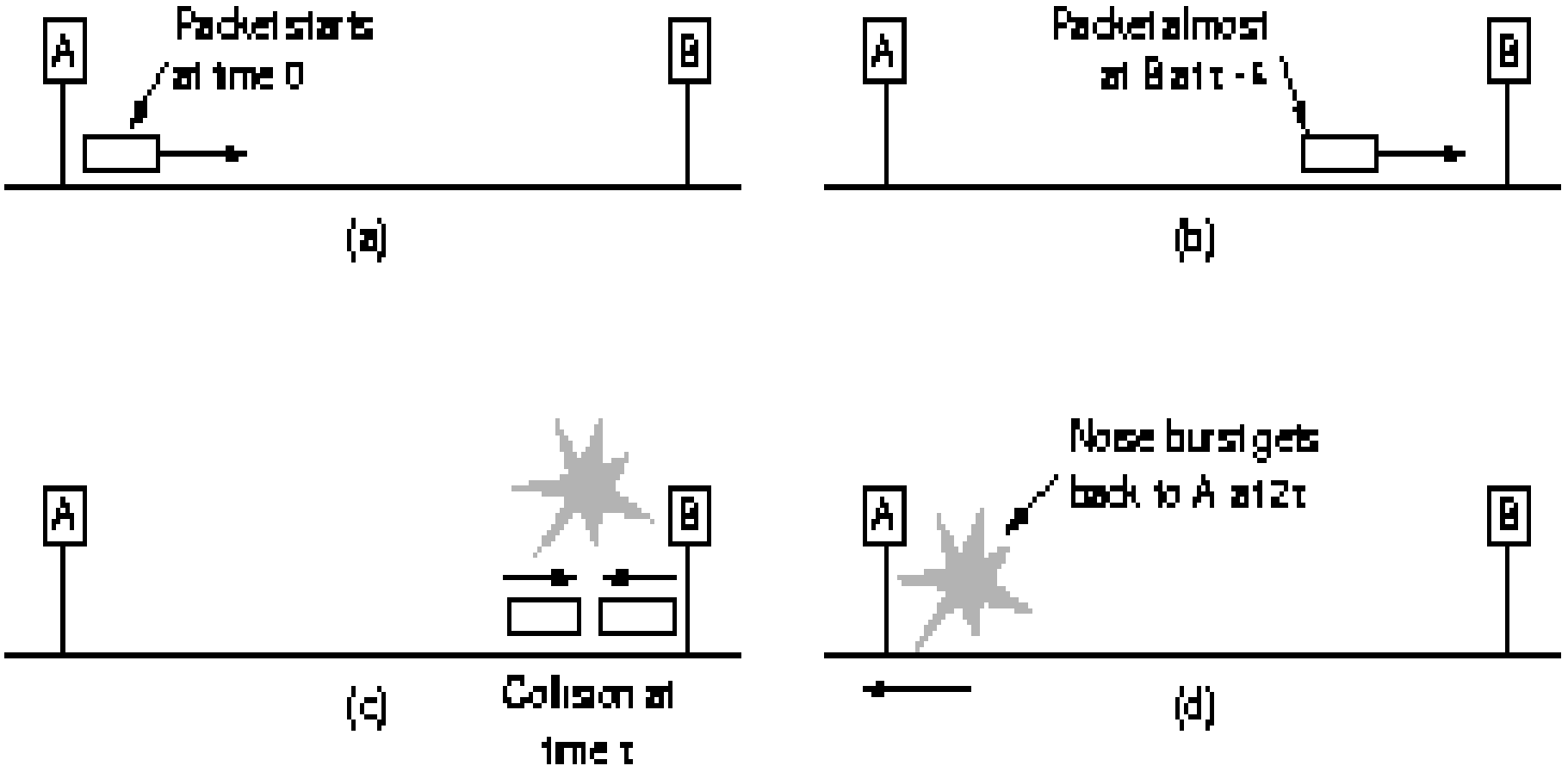
- *Goal*: adapt retransmission attempts to estimated current load
  - heavy load: random wait will be longer
- first collision: choose K from {0,1}; delay is K x 512 bit transmission times
- after second collision: choose K from {0,1,2,3}…
- after ten collisions, choose K from {0,1,2,3,4,…,1023}

# IEEE 802.3 Parameters

- 1 bit time = time to transmit one bit
  - 10 Mbps ➔ 1 bit time = 0.1 microseconds ($\mu s$)
- Maximum network diameter <= 2.5km
  - Maximum 4 repeaters
- "Collision Domain"
  - Distance within which collision can be detected
  - IEEE 802.3 specifies:

    worst case collision detection time: 51.2 $\mu s$
- Why minimum frame size?
  - 51.2 $\mu s$ => minimum # of bits can be transited at 10Mpbs is 512 bits => 64 bytes is required for collision detection

# Worst Case Collision Detection Time



# CSMA/CD Efficiency

Relevant parameters

- cable length, signal speed, frame size, bandwidth

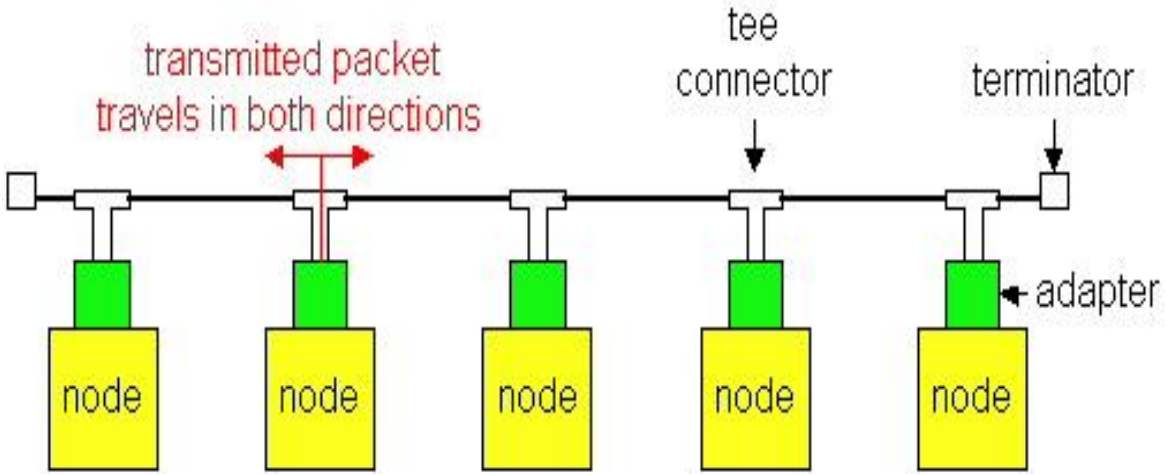
- $T_{prop}$ = max prop between 2 nodes in LAN
- $t_{trans}$ = time to transmit max-size frame

$$\text{efficiency} \approx \frac{1}{1 + 5t_{prop}/t_{trans}}$$

- Efficiency goes to 1 as $t_{prop}$ goes to 0
- Goes to 1 as $t_{trans}$ goes to infinity
- Much better than ALOHA, but still decentralized, simple, and cheap

# Ethernet Technologies: 10Base2

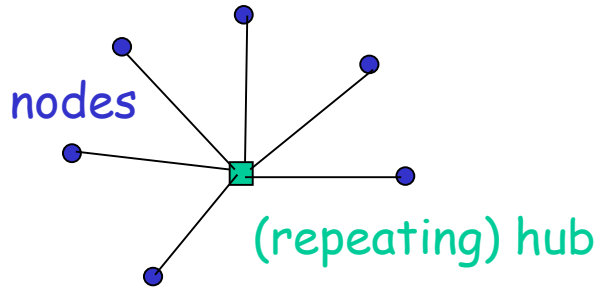
- 10: 10Mbps; 2: under 200 meters max cable length
- thin coaxial cable in a bus topology




- repeaters used to connect up to multiple segments
- repeater repeats bits it hears on one interface to its other interfaces: physical layer device only!
- has become a legacy technology

# 10BaseT and 100BaseT

- 10/100 Mbps rate; latter called "fast ethernet"
- T stands for Twisted Pair
- Nodes connect to a hub: "star topology"; 100 m max distance between nodes and hub




- Hubs are essentially physical-layer repeaters:
  - bits coming in one link go out all other links
  - no frame buffering
  - no CSMA/CD at hub: adapters detect collisions
  - provides net management functionality

# 100Base T (Fast) Ethernet: Issues

- 1 bit time = time to transmit one bit
  - 100 Mbps ➔ 1 bit time = 0.01 $\mu s$ (microseconds)
- If we keep the same "collision domain", i.e., worst case collision detection time kept at 51.2 (microseconds
  Q: What will be the minimum frame size?
  - 51.2 $\mu s$ => minimum # of bits can be transited at 100Mpbs is 5120 bits => 640 bytes is required for collision detection
  - This requires change of frame format and protocol!
- Or we can keep the same minimum frame size, but reduce "collision domain" or network diameter!
  - from 51.2 $\mu s$ to 5.12 $\mu s$ !
  - maximum network diameter $\pounds$ 100 m!

# Gigabit Ethernet & Beyond

## 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
  - also uses hubs, called "Buffered Distributors"
- Full-Duplex at 1 Gbps for point-to-point links

- Now: 10 & 40 Gbps are widely available
- And 100 Gbps is also here !
- All are used in "point-to-point" settings with Ethernet switches

# Ethernet Summary

- 1-persistent CSMA/CD
- 10Base Ethernet
  - 51.2 $\mu s$ to seize the channel
  - Collision not possible after 51.2 $\mu s$
  - Minimum frame size of 64 bytes
  - Binary exponential backoff
  - Works better under light load
  - Delivery time non-deterministic
- Evolution of Ethernet: Fast (100BaseT) and Gigabit Ethernet, and beyond