Transmission over links and local area networks

Outline

- Context/overview
- Introduction and services
- Link types and overview of multiple access protocols
- Link-layer addressing
- Ethernet
- Link-layer switches
- PPP
- A day in the life of a web request

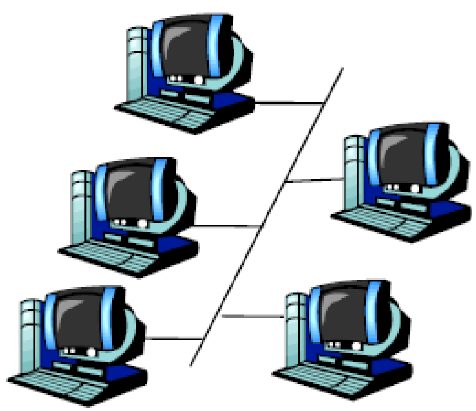
Ethernet

"dominant" wired LAN technology:

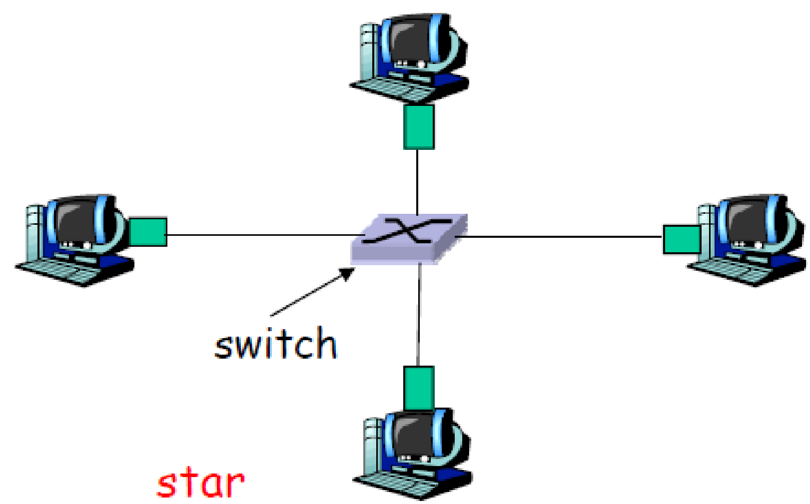
- First widely used LAN technology
- simpler, cheaper than competitors
  - token-ring (16 Mbps), FDDI (100 Mbps), and ATM (155 Mbps)
- kept up with speed race: 10 Mbps – 10 Gbps

Star topology

- bus topology popular through mid 90s, and later star topology with hub at center
  - all nodes in same collision domain (their transmissions can collide with each other) * broadcast-link/channel
- today: star topology with active switch (layer 2) at center
  - no collision Today's Ethernet is also called 'Switched Ethernet'



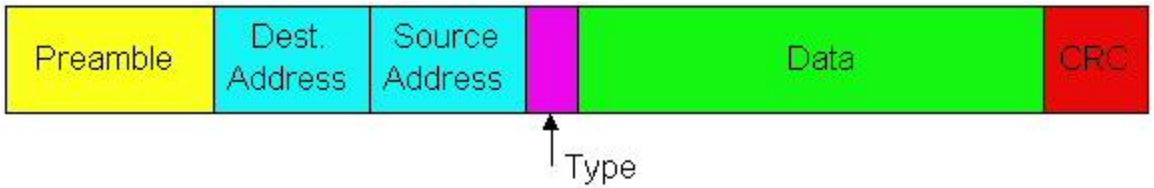
bus: coaxial cable



star

Ethernet Frame Structure

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



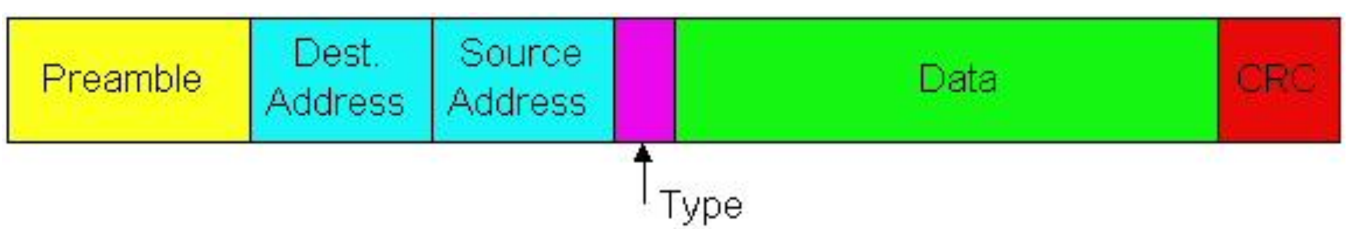
Preamble:

- 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 network layer protocol
  - otherwise, adapter discards frame
- Type: indicates higher layer protocol (mostly IP but others possible, e.g., Novell IPX, AppleTalk) or ARP
- CRC: checked at receiver, if error is detected, frame is dropped

Data field: 46 to 1,500 bytes



Ethernet: Unreliable, connectionless

- connectionless: No handshaking between sending and receiving NICs
- unreliable: receiving NIC doesn't send acks or nacks to sending NIC
  - stream of datagrams passed to network layer can have gaps (missing datagrams)
  - gaps will be filled if app is using TCP
  - otherwise, app will see gaps
- Ethernet's MAC protocol: unslotted CSMA/CD

Ethernet CSMA/CD algorithm

1. NIC receives datagram from network layer, creates frame
2. If NIC senses channel idle, starts frame transmission If NIC senses channel busy, waits until channel idle, then transmits
3. If NIC transmits entire frame without detecting another transmission, NIC is done with frame !
4. If NIC detects another transmission while transmitting, aborts and sends jam signal
5. After aborting, NIC enters **exponential backoff**: after *m*th collision, NIC chooses *K* at random from $\{0,1,2,\ldots,2^m-1\}$. NIC waits $K \cdot 512$ bit times, returns to Step 2

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 !

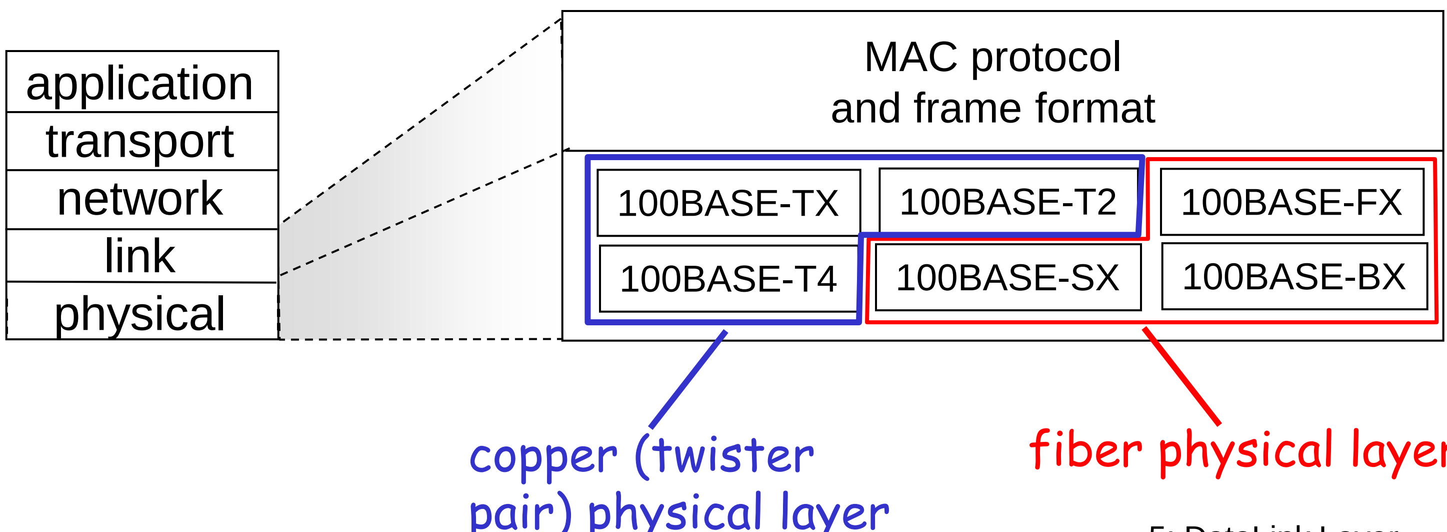
Exponential Backoff:

- *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· 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}

802.3 Ethernet Standards: Link & Physical Layers

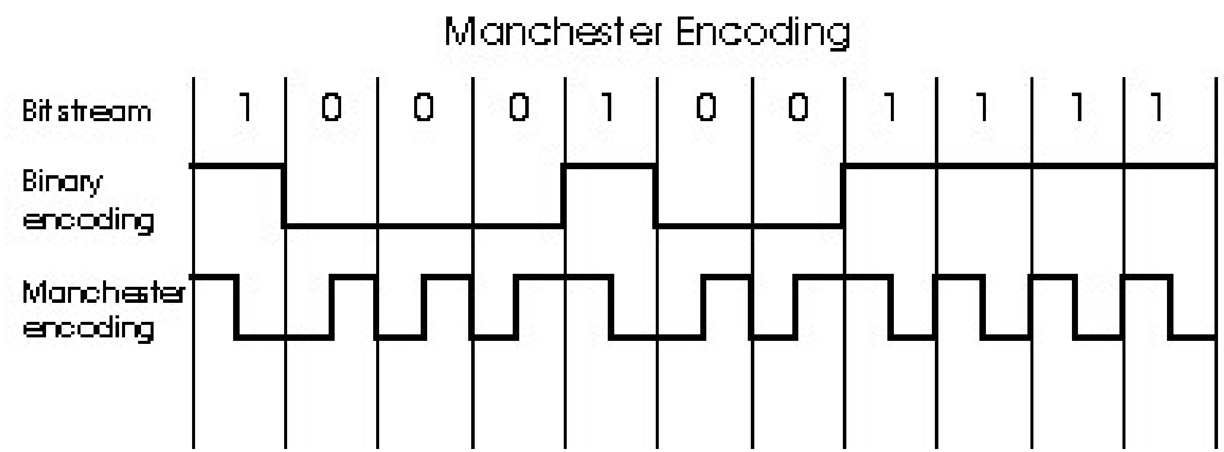
- *many* different Ethernet standards
  - common MAC protocol and frame format
  - different speeds: 2 Mbps, 10 Mbps, 100 Mbps, 1Gbps, 10G bps
  - different physical layer media: fiber, cable

These and many other Ethernet technologies have been standardized by the IEEE 802.3 CSMA/CD (Ethernet) working group



Other e.g., 10BASE-T, 10BASE-2, 100BASE-T, 1000BASE-LX, and 10GBASE-T

Manchester encoding



- used in 10BaseT
- 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!
- Hey, this is physical-layer stuff!

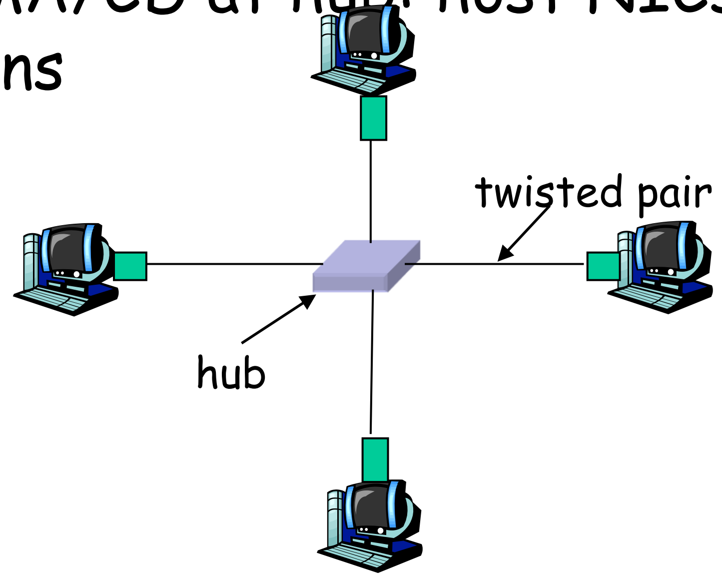
Transmission over links and local area networks
Outline

- Context/overview
- Introduction and services
- Link types and overview of multiple access protocols
- Link-layer addressing
- Ethernet
- Link-layer switches
- PPP
- A day in the life of a web request

Hubs

… physical-layer ("dumb") repeaters:

- bits coming in one link go out *all* other links at same rate
- all nodes connected to hub can collide with one another
- no frame buffering
- no CSMA/CD at hub: host NICs detect collisions



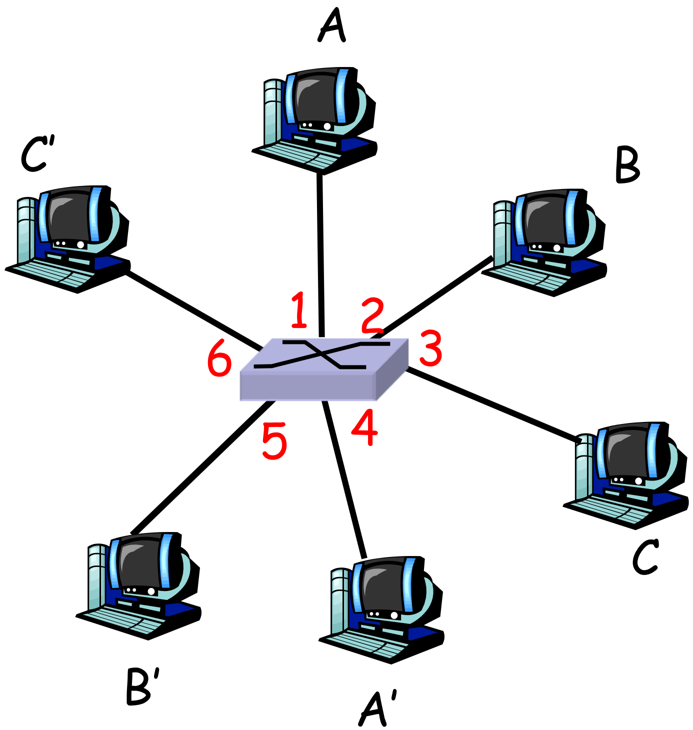
Switch (Layer 2)

- link-layer device: smarter than hubs, take *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

'Filtering' is the switch function that determines whether a frame should be forwarded to some interface or should just be dropped.

Switch: allows *multiple* simultaneous transmissions

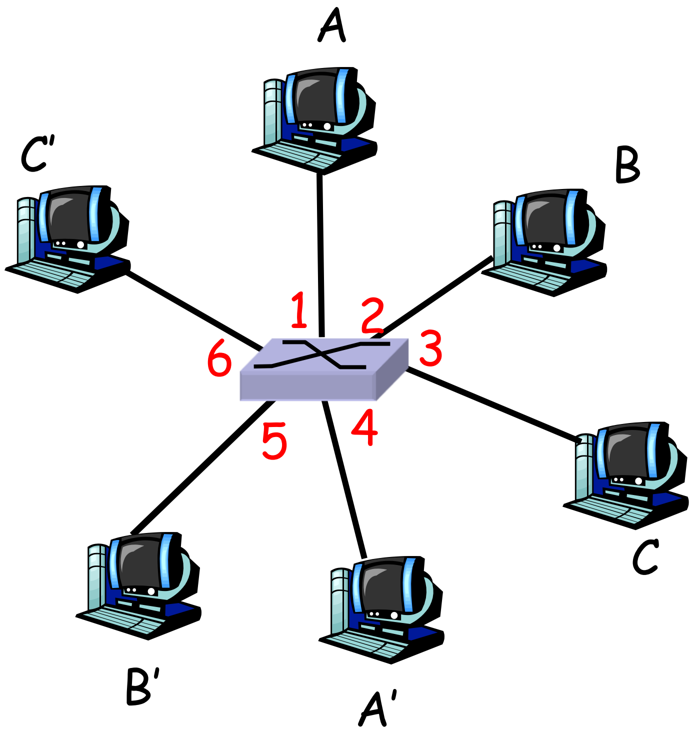
- hosts have dedicated, direct connection to switch
- switches buffer packets
- Ethernet protocol used on *each* incoming link, but no collisions; full duplex
  - each link is its own collision domain
- *switching:* A-to-A' and B-to-B' simultaneously, without collisions
  - not possible with dumb hub



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

Switch Table

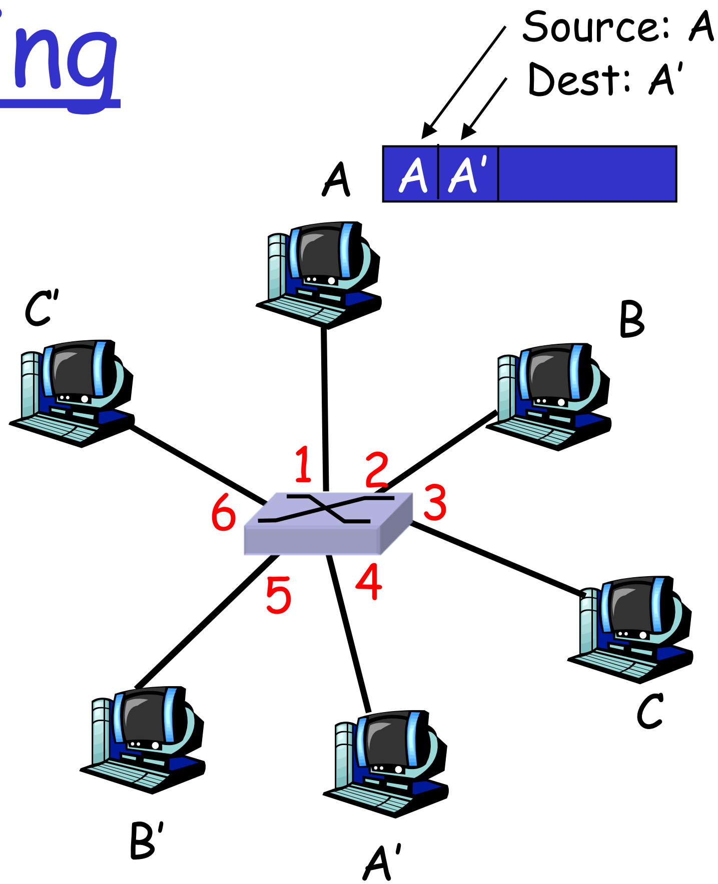
- Q: how does switch know that A' reachable via interface 4, B' reachable via interface 5?
- A: each switch has a switch table, each entry:
  - (MAC address of host, interface to reach host, time stamp)
- looks like a routing table!
- Q: how are entries created, maintained in switch table?
  - something like a routing protocol?



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

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

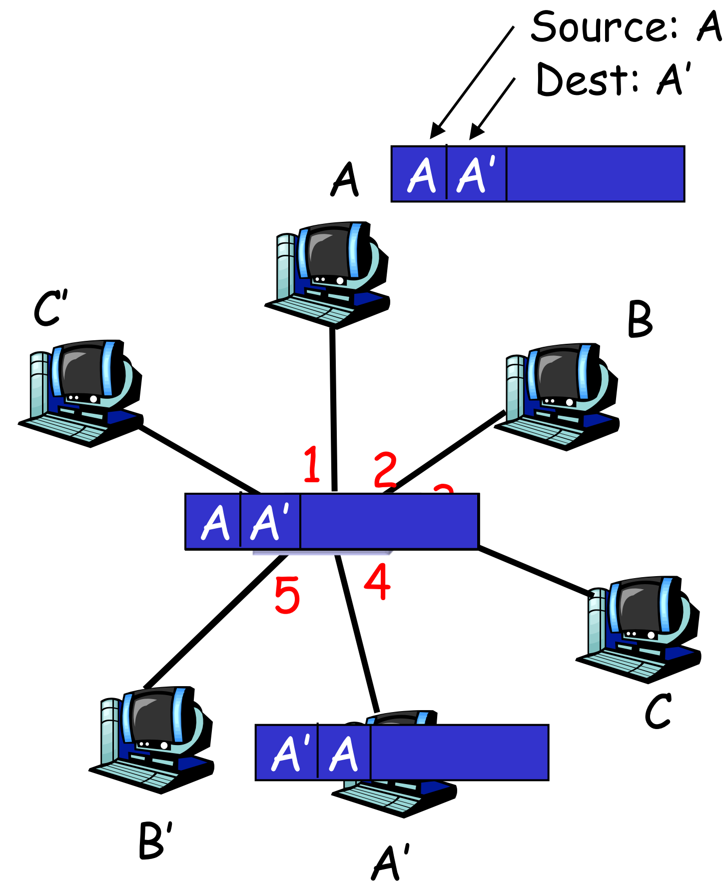
When frame received:

1. record link associated with sending host
2. index switch table using MAC dest address
3. **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

Self-learning, forwarding: example

- frame destination unknown: *flood*
- destination A location known: *selective send*

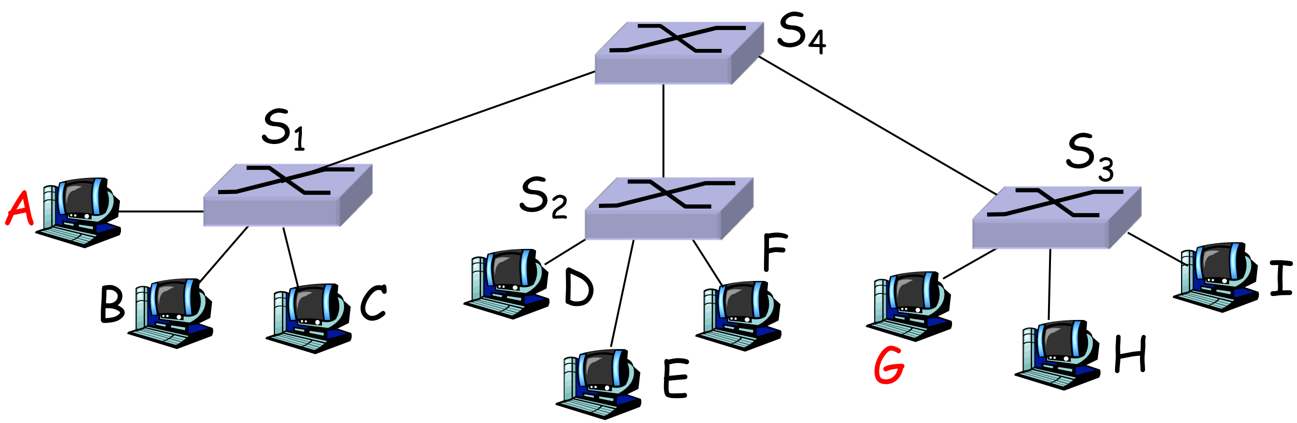


MAC addr	interface	TTL
A	*1*	*60*
A'	*4*	*60*

Switch table (initially empty)

Interconnecting switches

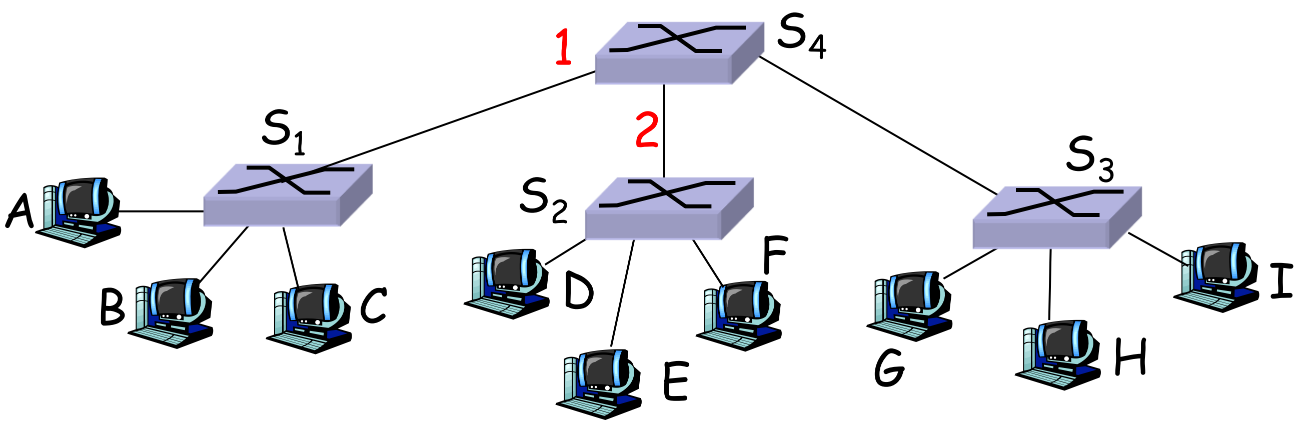
- switches can be connected together



- 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!)

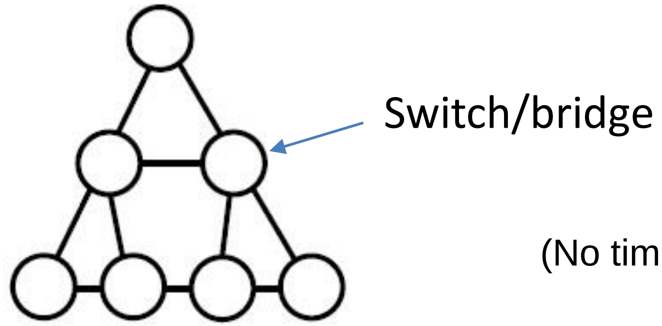
Self-learning multi-switch example

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

Flooding/broadcasting (in self-learning also): What if physical topology has loops?

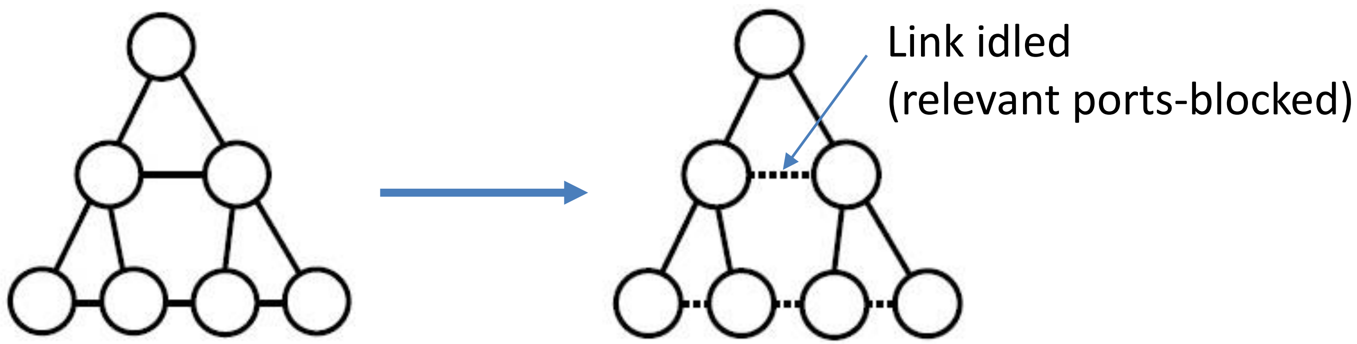


Broadcast storm!

(No time-to-live (TTL) in layer2 header!)

Solutions:

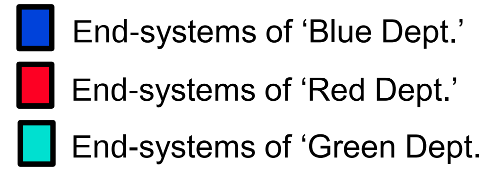
- Spanning Tree Protocol (STP): Helps form a tree out of a mesh topology

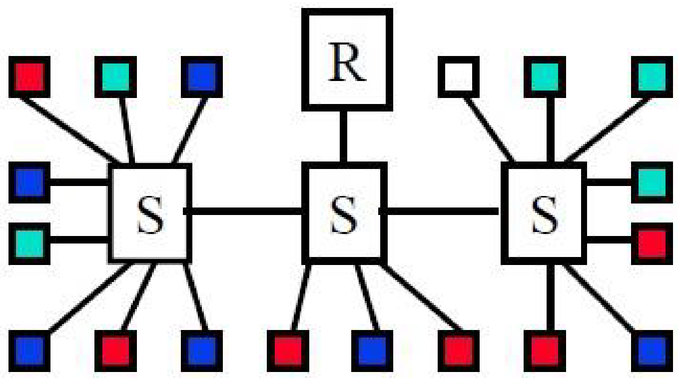


- Rapid STP (RSTP, improvements to STP), Multiple STP (extension to RSTP for VLANs)

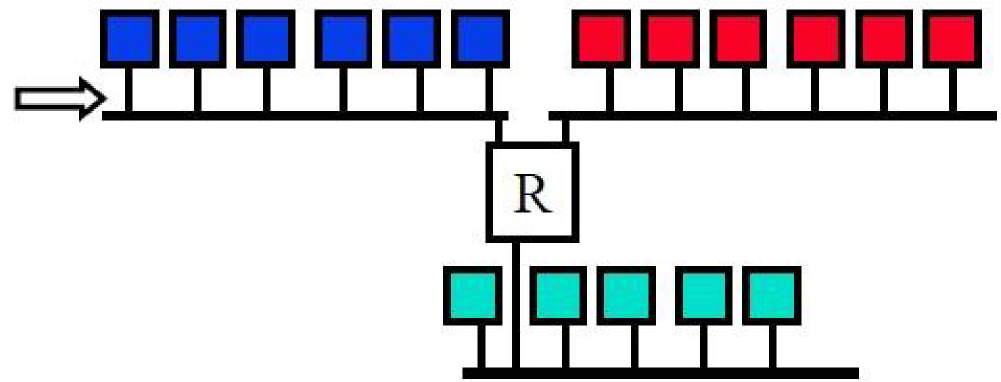
- Shortest Path Bridging (SPB, 2012) (for replacing STP/RSTP/MSTP)
  - Spanning tree protocol is wasteful of resources (STP blocks certain links!) and slow.
  - SPB: Routing (IS-IS routing protocol, similar to OSPF) is used to populate switch-tables instead of self-learning!
  - Routing @ Ethernet! (Layers are being broken?)
  - Allows all links to be used.
  - Equal-cost multi-path (ECMP) used to distribute load

Virtual Local Area Networks (VLANs)





(Physical network)



3 VLANs: Blue, Red, Green

(Logical view after configuring VLANs on switches)

- VLANs allow different non-trusting entities to share an Ethernet network.
- Broadcasts and multicast goes only to the nodes in the same VLAN.
- VLAN membership- defined by the administrator. It could be based on: physical ports (typical), MAC addresses, IP subnets.
- VLAN Tag (includes fields: VLAN-ID, Priority) added into a frame on ‘trunk’ links (e.g., link between switches) to identify the VLAN to which the frame belongs.
- Switching based on VLAN ID + Destination MAC
- Multiple STP (MSTP, extension of RSTP to VLANs)

Evolution of Ethernet: From Traditional Ethernet to Carrier Ethernet

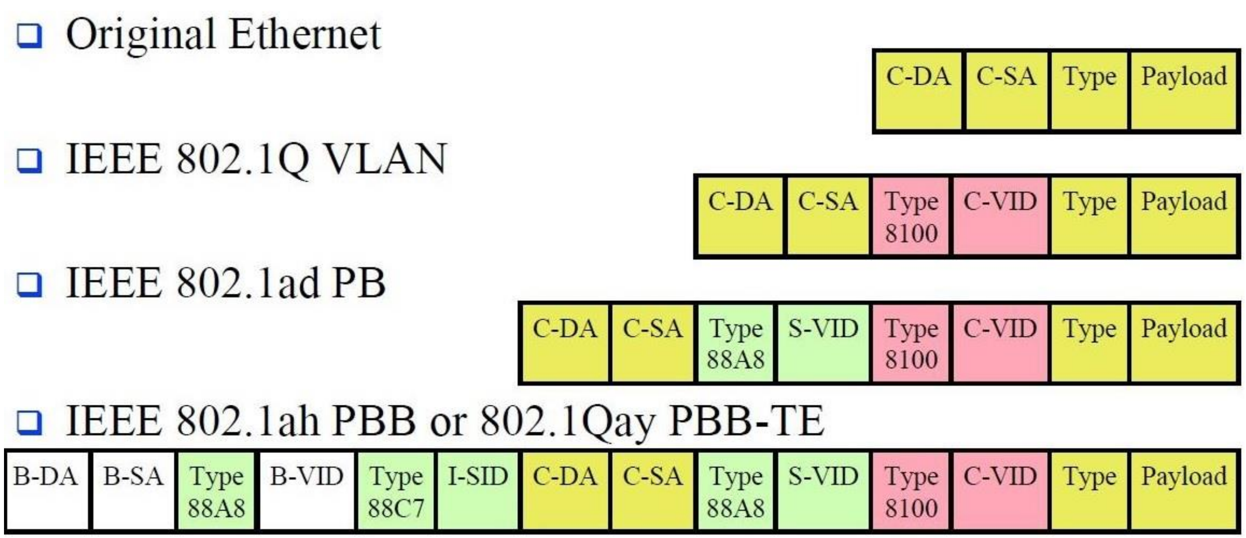
- Carrier Ethernet: extensions to Ethernet to enable telecommunications network providers to provide Ethernet services to customers and to utilize Ethernet technology in their networks (at MAN and WAN scales).

- Extensions to Ethernet for carriers (VLAN techniques used):
  - Provider Bridges (PB)
  - Provider Backbone Bridges (PBB)
  - Provider Backbone Bridge–Traffic Engineering (PBB-TE)

  (Also called ‘connection oriented Ethernet’ or Provider Backbone Transport (PBT))

- In these techniques, generally, the frame header and its processing are modified and the idea of stacked labels in MPLS is used. (Although these technologies were originally developed for carriers, they are also now used inside multi-tenant data centers (clouds))

Ethernet Frame Format Evolution



- Ethernet is heavily used in Data-centers.

Transmission over links and local area networks
Outline

- Context/overview
- Introduction and services
- Link types and overview of multiple access protocols
- Link-layer addressing
- Ethernet
- Link-layer switches
- PPP
- A day in the life of a web request

Point to Point Data Link Control

- one sender, one receiver, one link: easier than broadcast link:
  - no Media Access Control
  - no need for explicit MAC addressing
  - e.g., dialup link, ISDN line
- popular point-to-point DLC protocols:
  - PPP (point-to-point protocol)
  - HDLC: High level data link control (Data link used to be considered "high layer" in protocol stack!

PPP Design Requirements [RFC 1557]

- packet framing: encapsulation of network-layer datagram in data link frame
  - carry network layer data of any network layer protocol (not just IP) *at same time*
  - ability to demultiplex upwards
- bit transparency: must carry any bit pattern in the data field
- error detection (no correction)
- connection liveness: detect, signal link failure to network layer
- network layer address negotiation: endpoint can learn/configure each other's network address

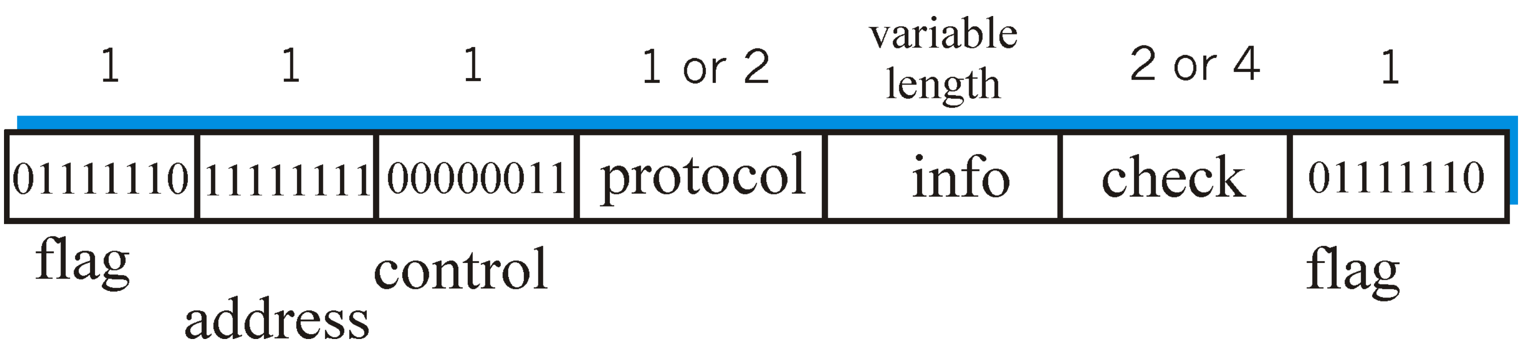
PPP non-requirements

- no error correction/recovery
- no flow control
- out of order delivery OK
- no need to support multipoint links (e.g., polling)

Error recovery, flow control, data re-ordering
all relegated to higher layers!

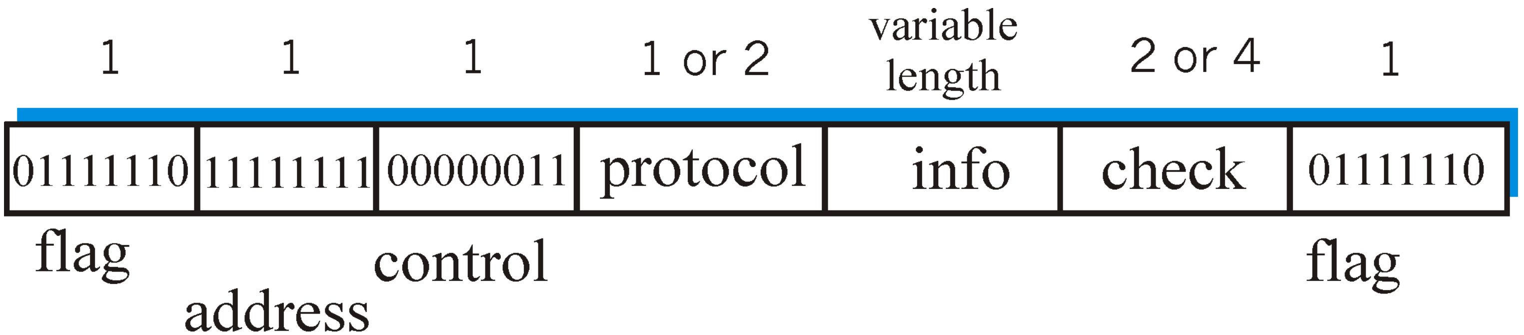
PPP Data Frame

- Flag: delimiter (framing)
- Address: does nothing (only one option)
- Control: does nothing; in the future possible multiple control fields
- Protocol: upper layer protocol to which frame delivered (eg, PPP-LCP, IP, IPCP, etc)



PPP Data Frame

- info: upper layer data being carried
- check: cyclic redundancy check for error detection

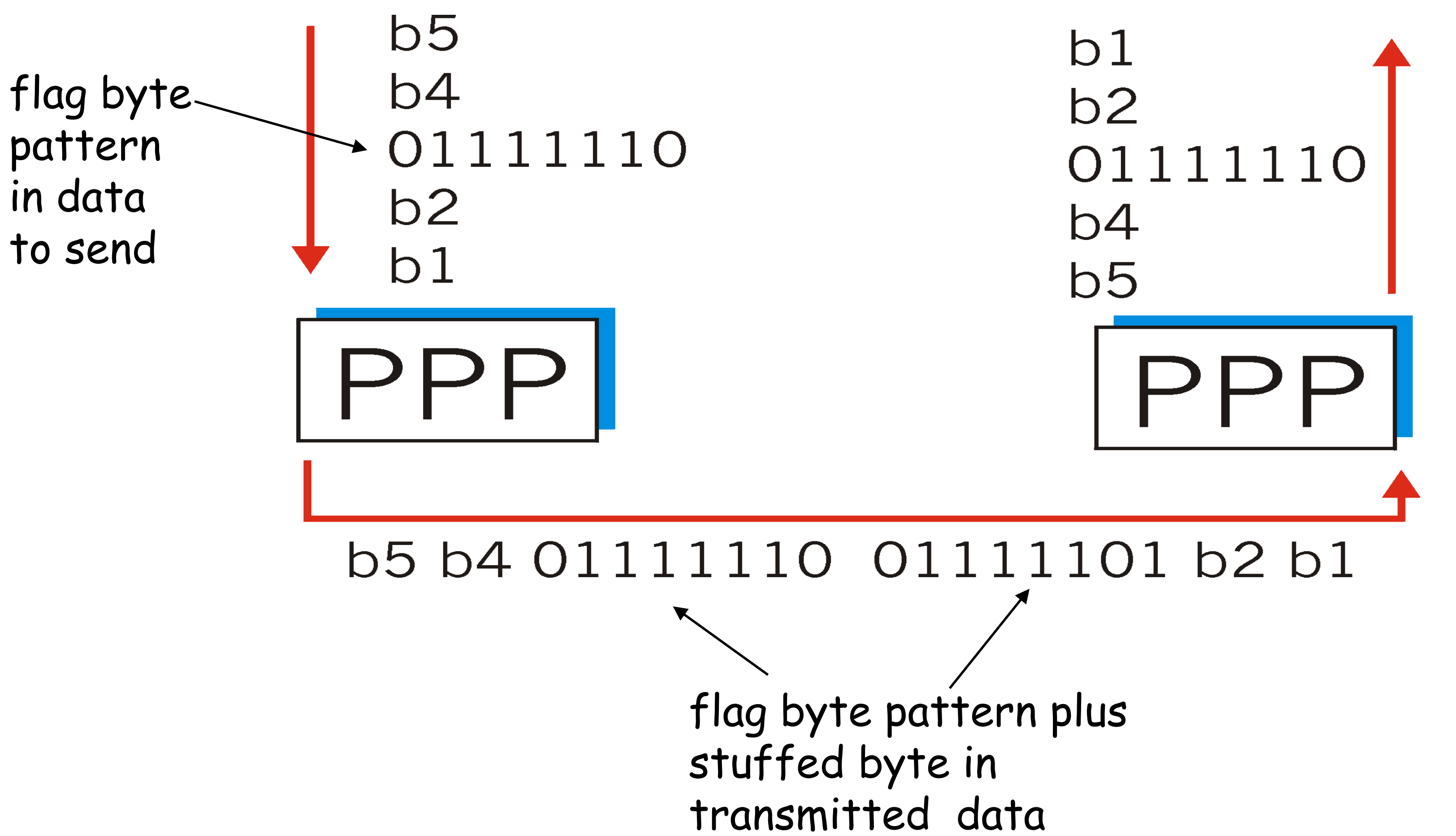


Byte Stuffing

- "data transparency" requirement: data field must be allowed to include flag pattern <01111110>
  - Q: is received <01111110> data or flag?

- Sender: adds ("stuffs") extra < 01111110> byte after each < 01111110> *data* byte
- Receiver:
  - two 01111110 bytes in a row: discard first byte, continue data reception
  - single 01111110: flag byte

Byte Stuffing



PPP Data Control Protocol

Before exchanging network-layer data, data link peers must

- configure PPP link (max. frame length, authentication)
- learn/configure network layer information
  - for IP: carry IP Control Protocol (IPCP) msgs (protocol field: 8021) to configure/learn IP address