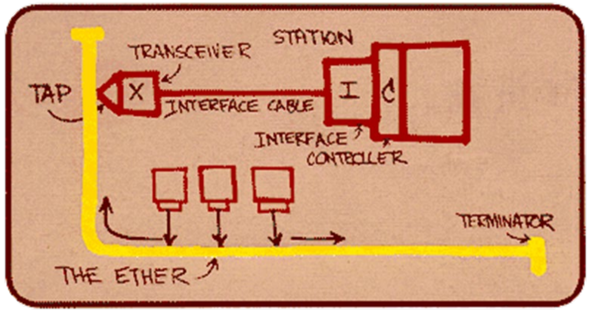
Data Link Layer: ETHERNET, Data Centre

A/PROF. DUY NGO

Ethernet

"dominant" wired LAN technology:

- single chip, multiple speeds (e.g., Broadcom BCM5761)
- first widely used LAN technology
- simpler, cheap
- kept up with speed race: 10 Mbps – 10 Gbps

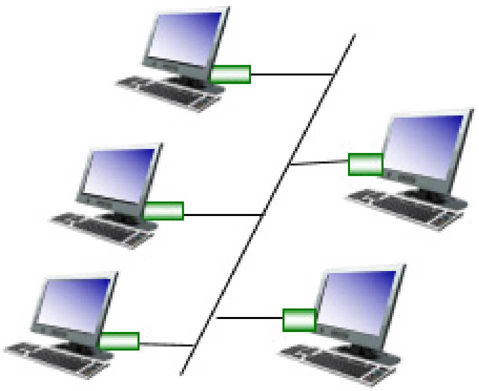


Metcalfe's Ethernet sketch

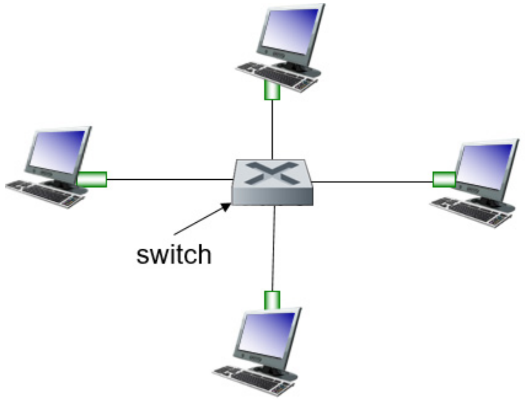
Ethernet: Physical Topology

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

bus: coaxial cable

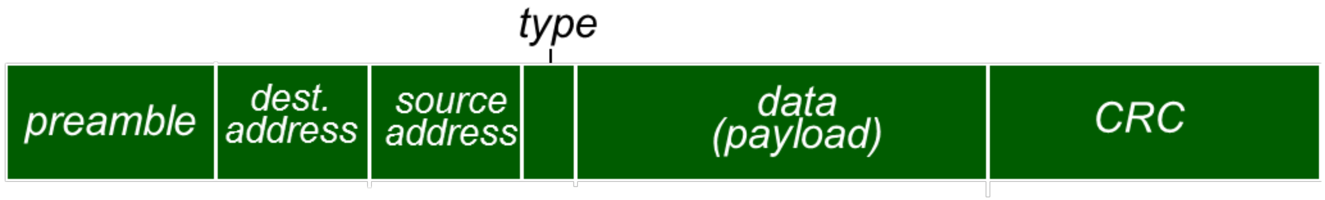


star



Ethernet Frame Structure (1 of 2)

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

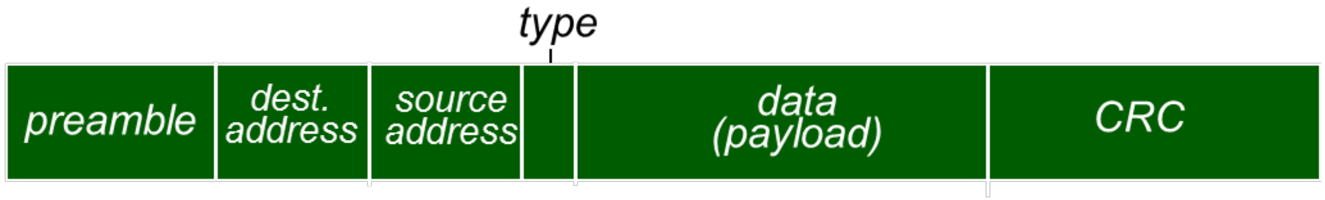


preamble:

- 7 bytes with pattern 10101010 followed by one byte with pattern 10101011
- used to synchronize receiver, sender clock rates

Ethernet Frame Structure (2 of 2)

- **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

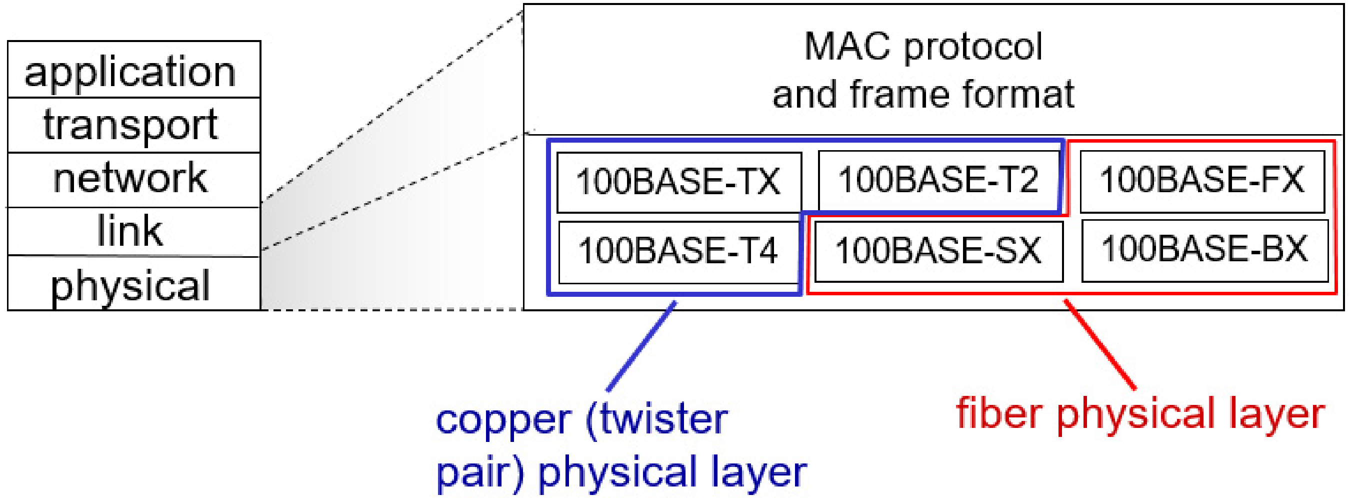


Ethernet: Unreliable, Connectionless

- **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** (for coaxial-cable-based and hub-based Ethernet; not for switch-based Ethernet)

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



application
transport
network
link
physical

MAC protocol
and frame format

100BASE-TX
100BASE-T2
100BASE-FX
100BASE-T4
100BASE-SX
100BASE-BX

copper (twister pair) physical layer

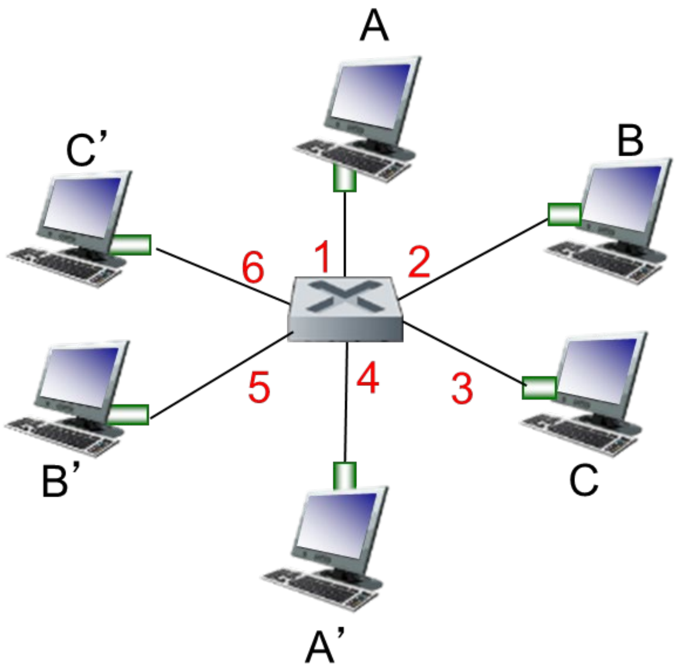
fiber physical layer

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
- **transparent**
  - hosts are 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, 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)

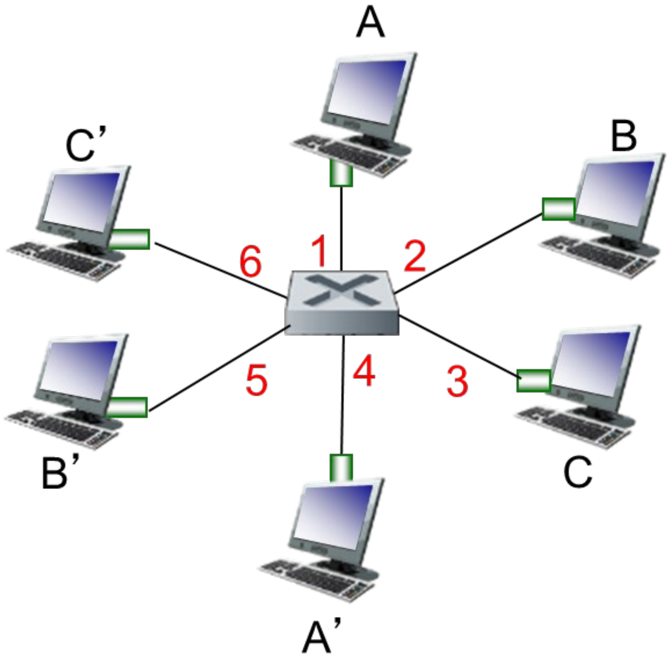
Switch Forwarding Table

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?

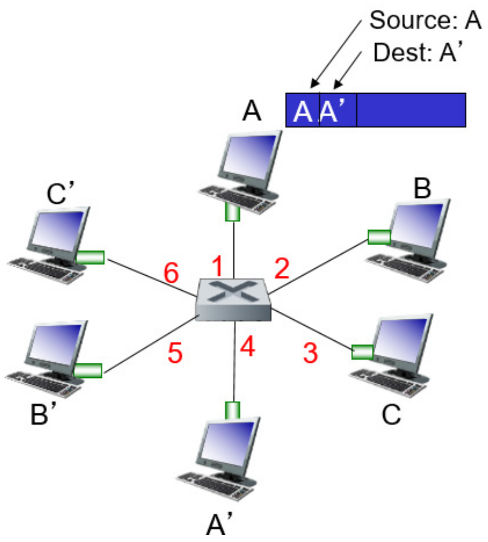
- something like a routing protocol?



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

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

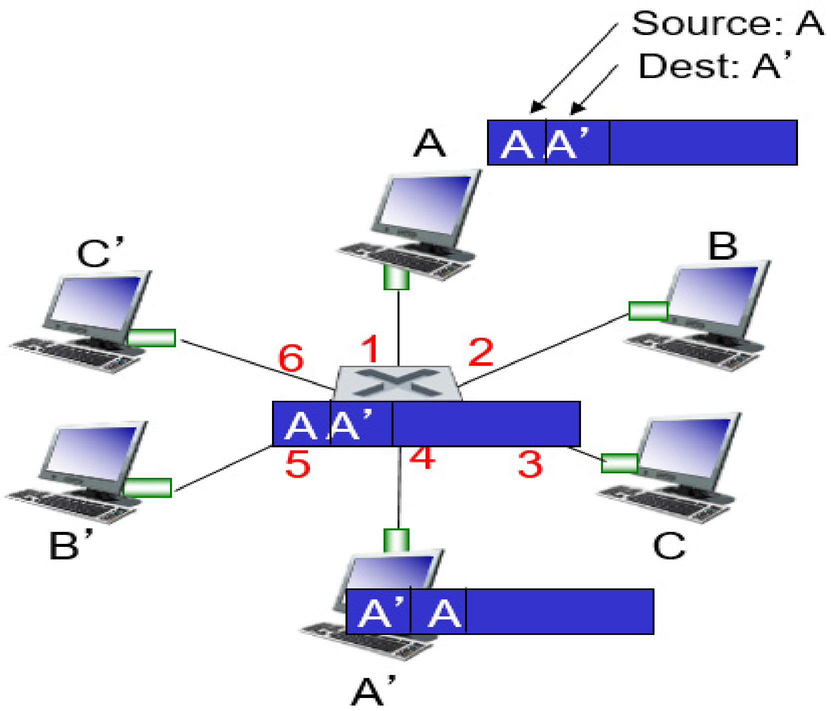
```
when frame received at switch:
1.  record incoming link, MAC address of sending host
2.  index switch table using MAC destination address
3.  if entry found for destination
        then {
        if destination on segment from which frame arrived
            then drop frame
            else forward 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 location known: **selectively send on just one link**

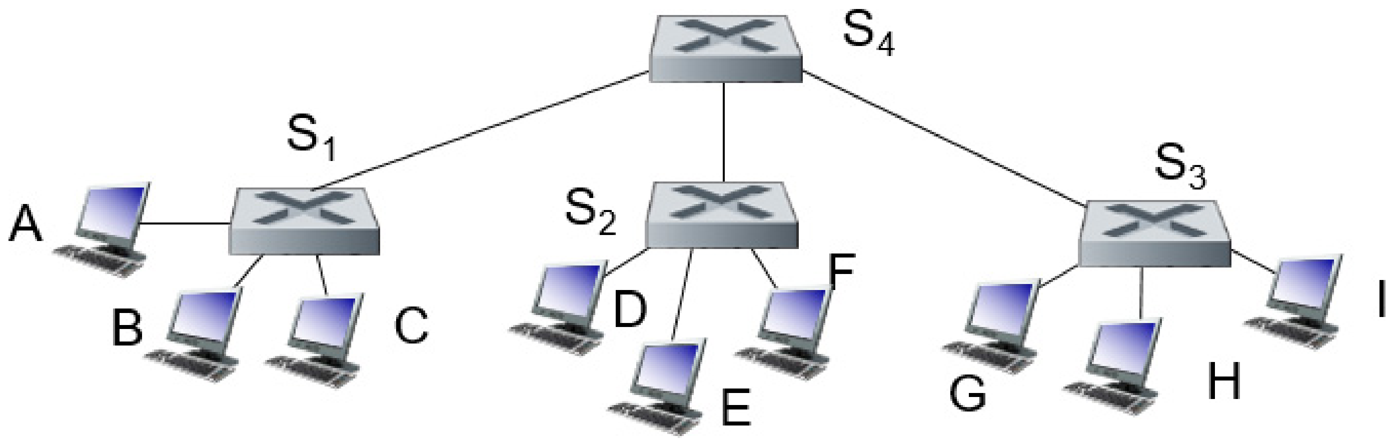
MAC addr	interface	TTL
A A'	1 4	60 60

switch table (initially empty)



Interconnecting Switches

self-learning switches can be connected together:

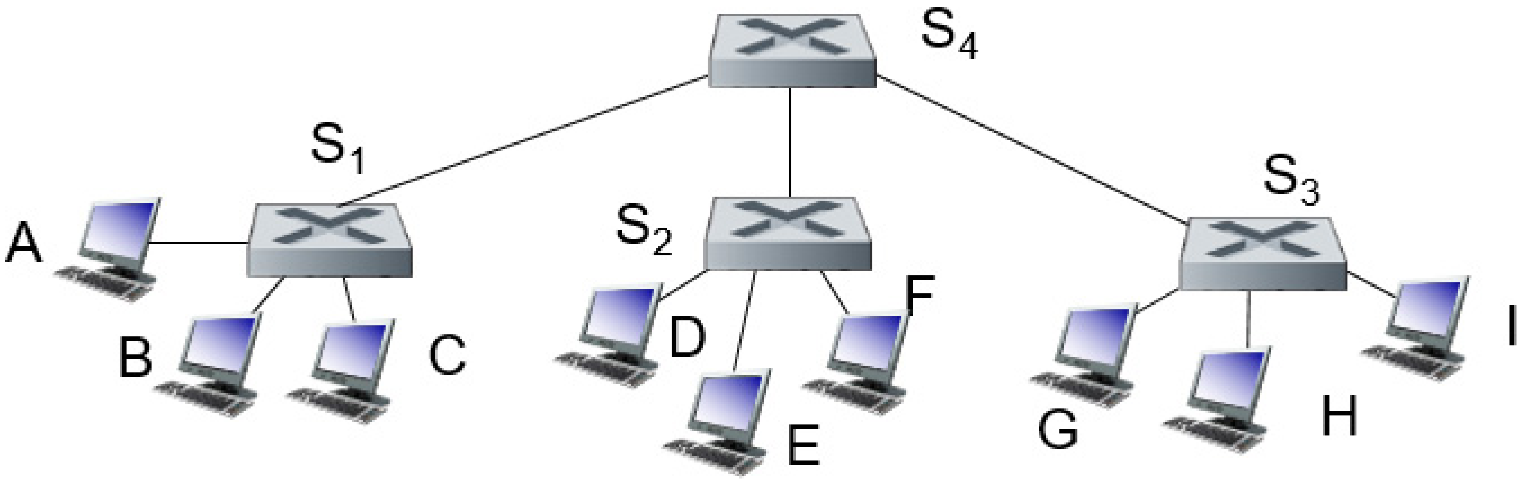


Q: sending from A to G - how does S_1 know to forward frame destined to G via S_4 and S_3?

- **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

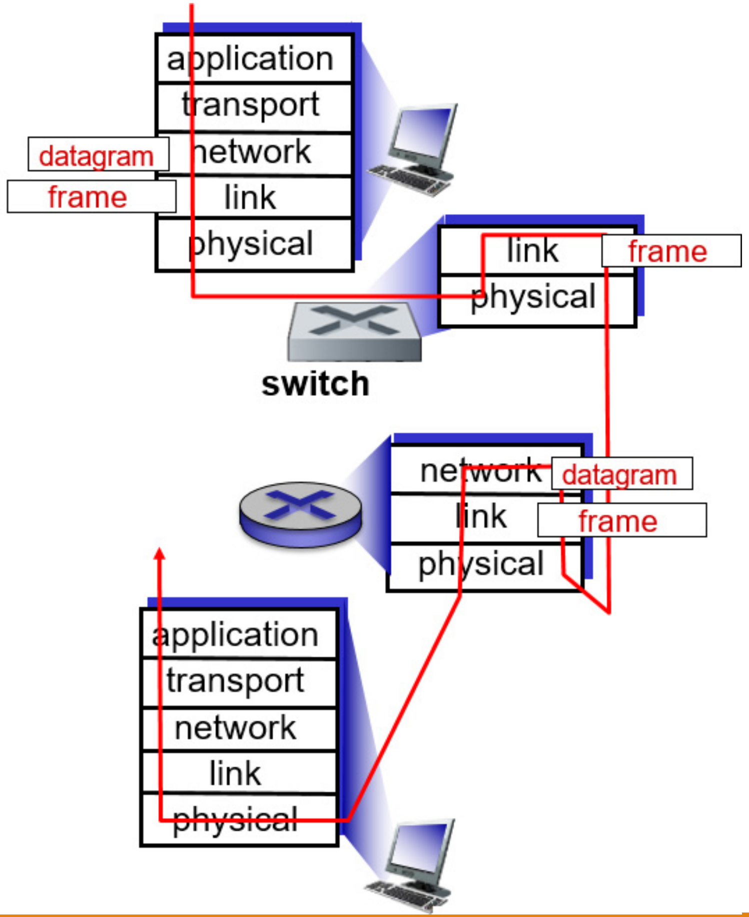
Switches vs. Routers

both are store-and-forward:

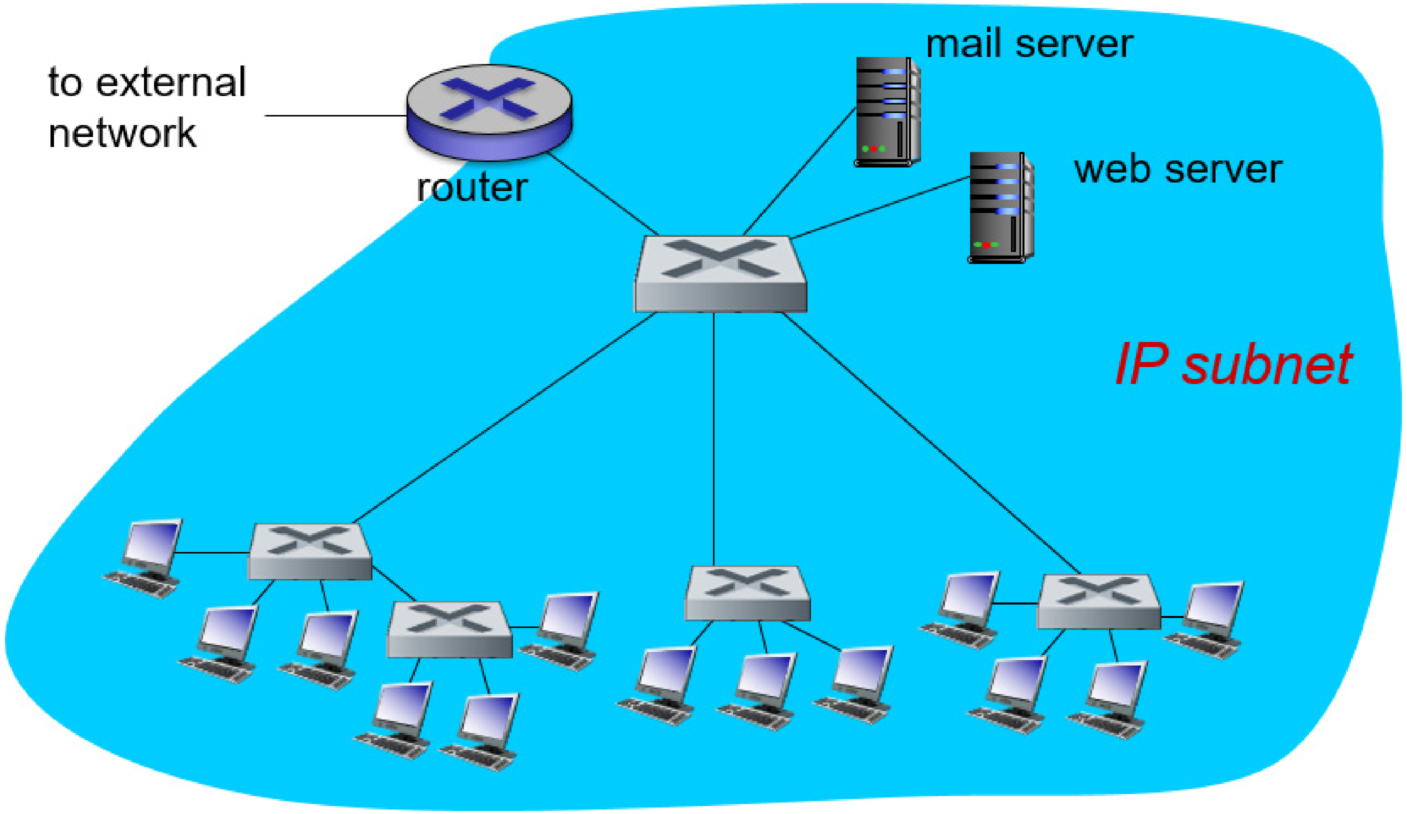
- **routers:** network-layer devices (examine network-layer headers)
- **switches:** link-layer devices (examine link-layer headers)

both have forwarding tables:

- **routers:** compute tables using routing algorithms, IP addresses
- **switches:** learn forwarding table using flooding, learning, MAC addresses



Institutional Network



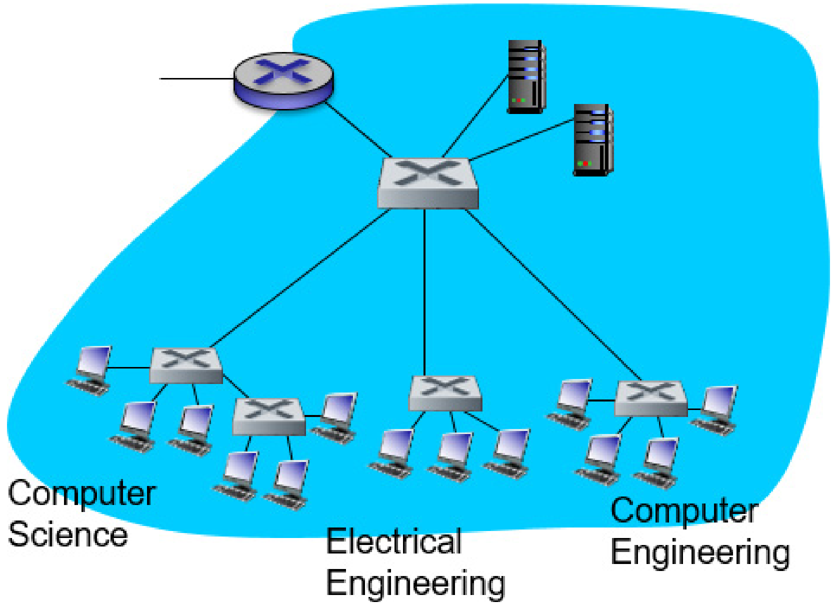
VLANs: Motivation

consider:

CS user moves office to EE, but wants connect to CS switch?

single broadcast domain - issues:

- all layer-2 broadcast traffic (ARP, DHCP, unknown location of destination MAC address) must cross entire LAN
- security/privacy, efficiency issues

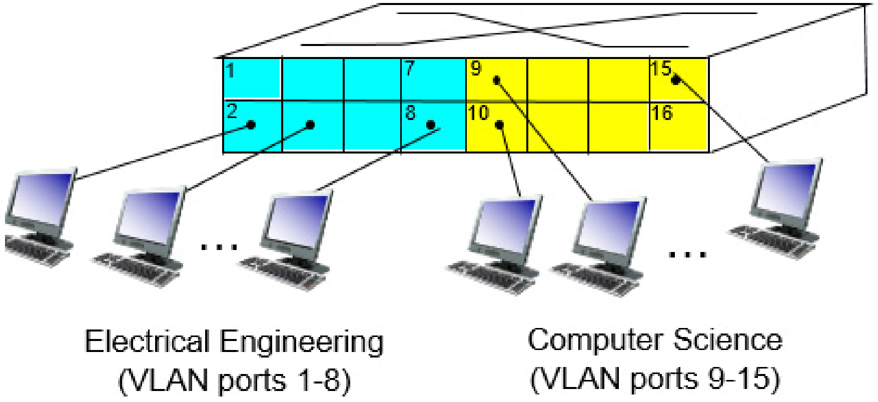


VLANs

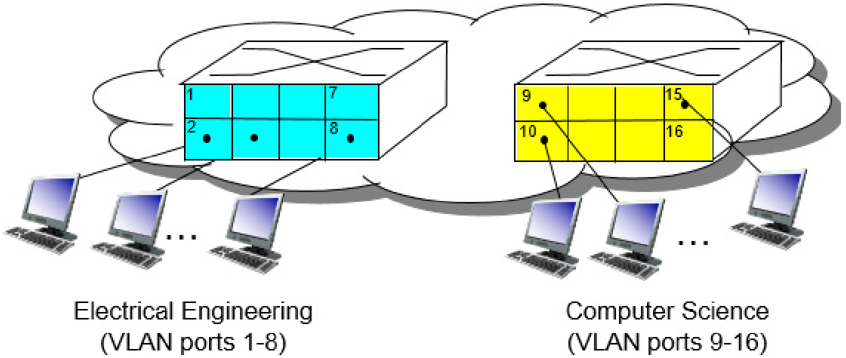
Virtual Local Area Network

switch(es) supporting VLAN capabilities can be configured to define multiple **virtual** LANS over single physical LAN infrastructure.

port-based VLAN: switch ports grouped (by switch management software) so that **single** physical switch ……

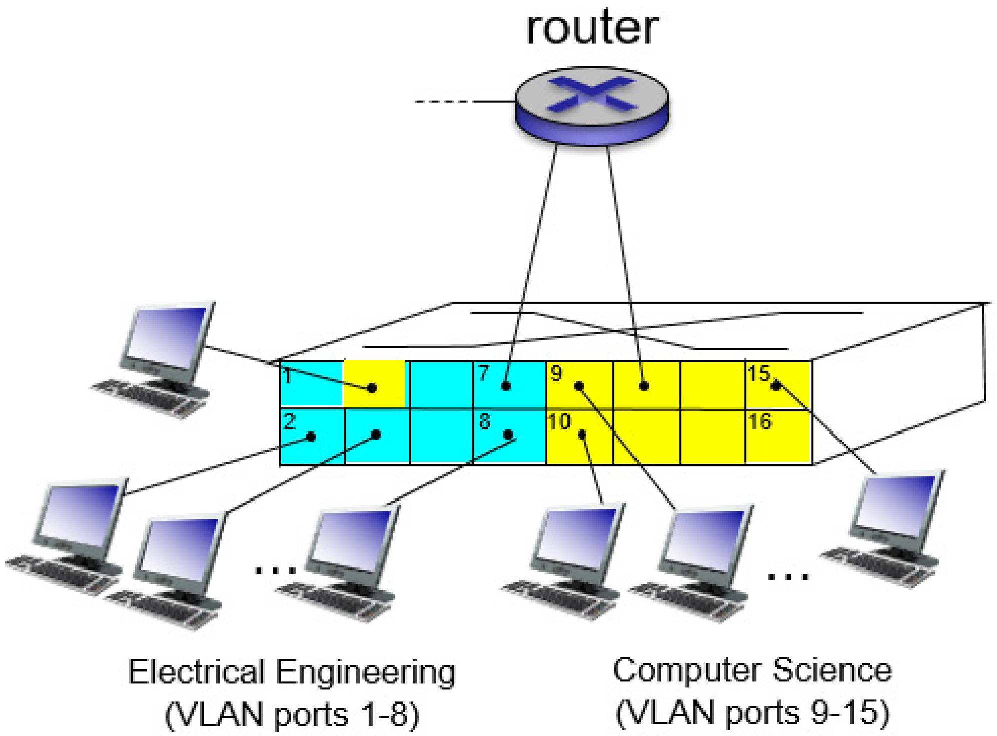


… operates as **multiple** virtual switches

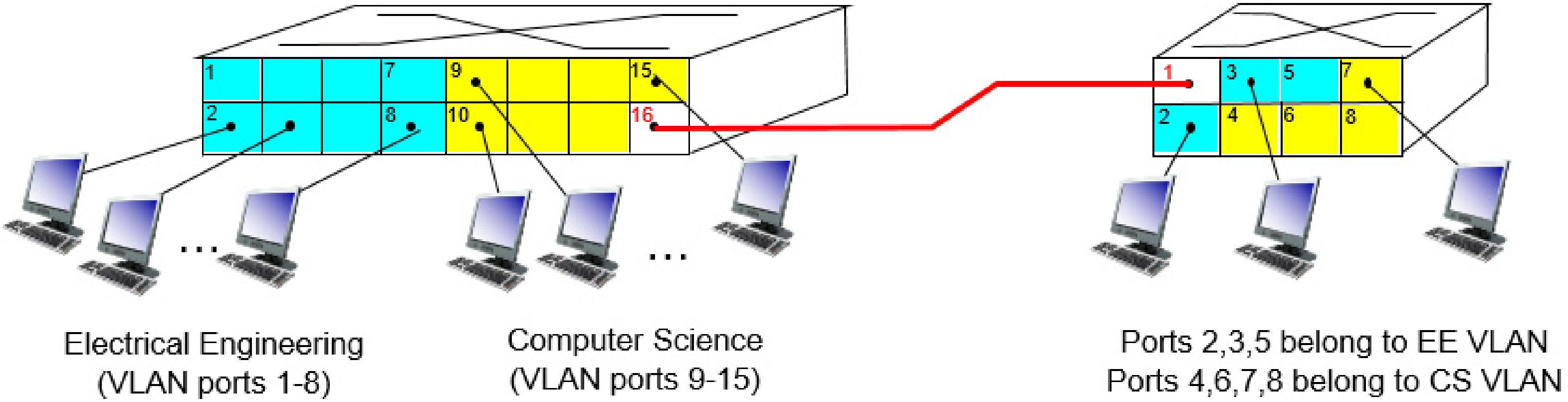


Port-Based VLAN

- **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
- **dynamic membership:** ports can be dynamically assigned among VLANs
- **forwarding between V LANS:** done via routing (just as with separate switches)
  - in practice vendors sell combined switches plus routers



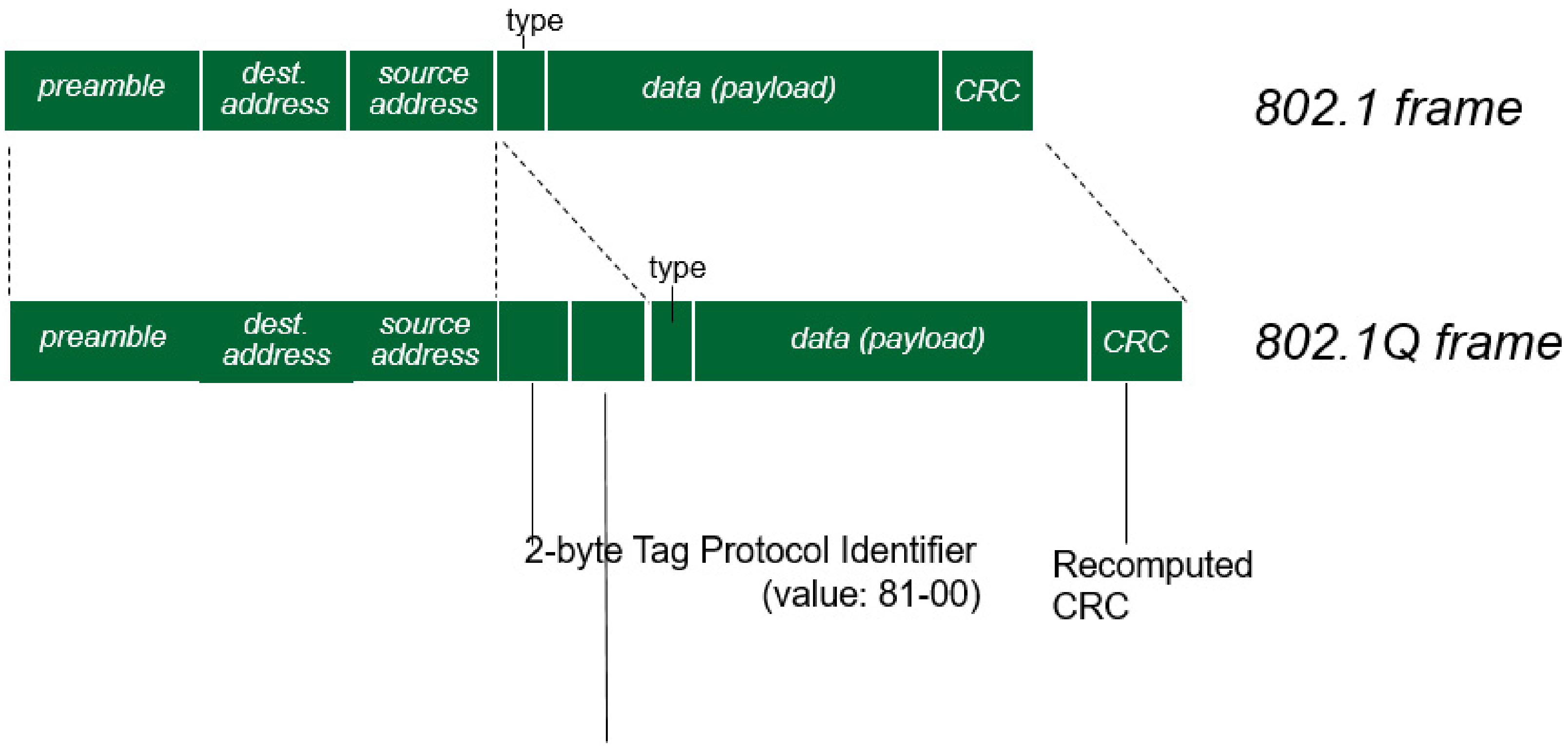
VLANS Spanning Multiple Switches



trunk port: carries frames between VLANs defined over multiple physical switches

- frames forwarded within VLAN between switches can't be vanilla 802.1 frames (must carry VLAN ID info)
- 802.1q protocol adds/removed additional header fields for frames forwarded between trunk ports

802.1Q VLAN Frame Format



2-byte Tag Protocol Identifier
(value: 81-00)

Recomputed
CRC

Tag Control Information (12 bit VLAN ID field,
3 bit priority field like IP TOS)

Data Center Networks (1 of 3)

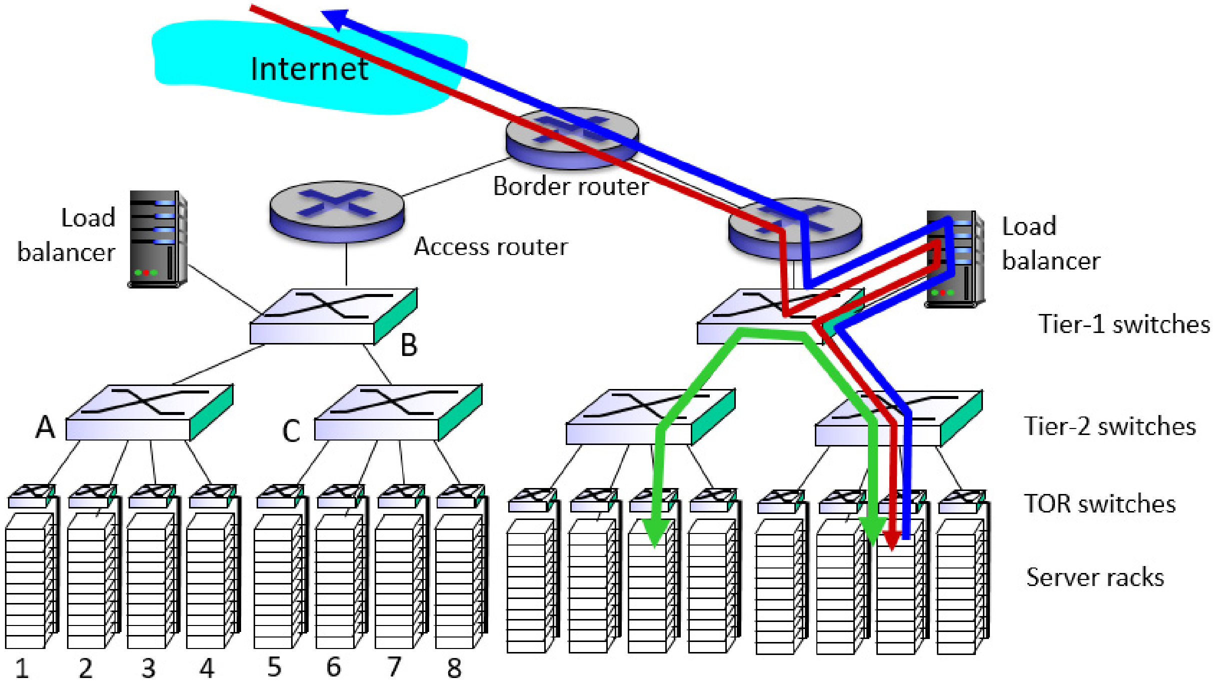
- 10’s to 100’s of thousands of hosts, often closely coupled, in close proximity:
  - e-business (e.g. Amazon)
  - content-servers (e.g., YouTube, Akamai, Apple, Microsoft)
  - search engines, data mining (e.g., Google)
- challenges:
  - multiple applications, each serving massive numbers of clients
  - managing/balancing load, avoiding processing, networking, data bottlenecks

Inside a 40-f t Microsoft container, Chicago data center

Data Center Networks (2 of 3)

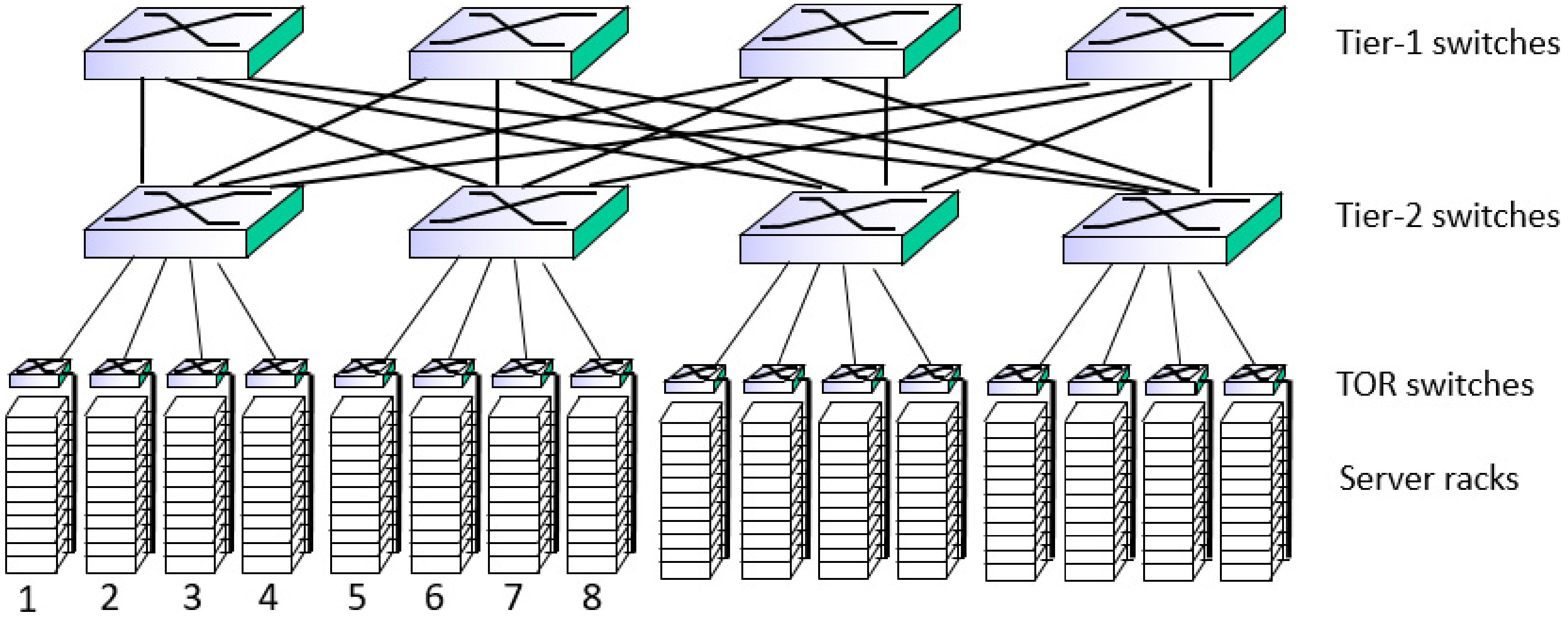
load balancer: application-layer routing

- receives external client requests
- directs workload within data center
- returns results to external client (hiding data center internals from client)



Data Center Networks (3 of 3)

- rich interconnection among switches, racks:
  - increased throughput between racks (multiple routing paths possible)
  - increased reliability via redundancy


Tier-1 switches
Tier-2 switches
TOR switches
Server racks
1
2
3
4
5
6
7
8