CS 330: Network Applications & Protocols

Link Layer

Galin Zhelezov
Department of Physical Sciences
York College of Pennsylvania

Overview of Link Layer

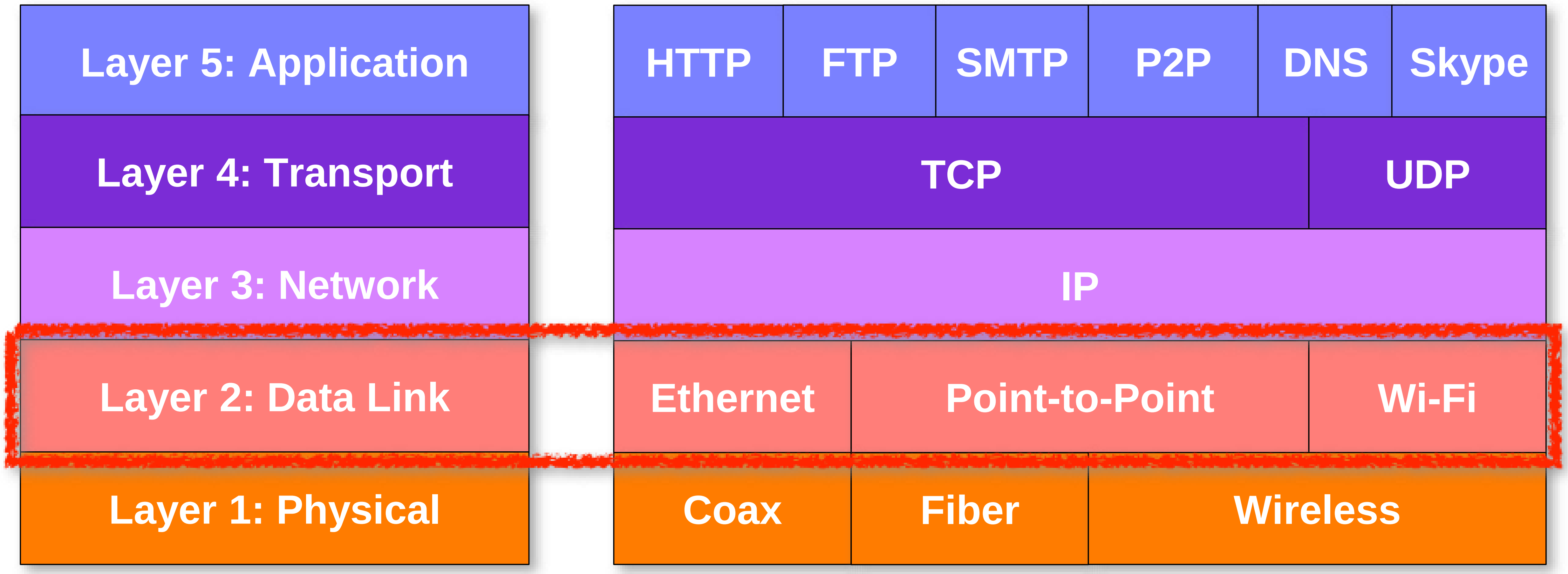
- **Introduction, Services**
- **Error Detection, Correction**
- **Multiple Access Protocols**
- **LANs**
  - Addressing, ARP
  - Ethernet
  - Switches
- **A Day in the Life**

Overview of Link Layer

- **Introduction, Services**
- **Error Detection, Correction**
- **Multiple Access Protocols**
- **LANs**
  - Addressing, ARP
  - Ethernet
  - Switches
- **A Day in the Life**

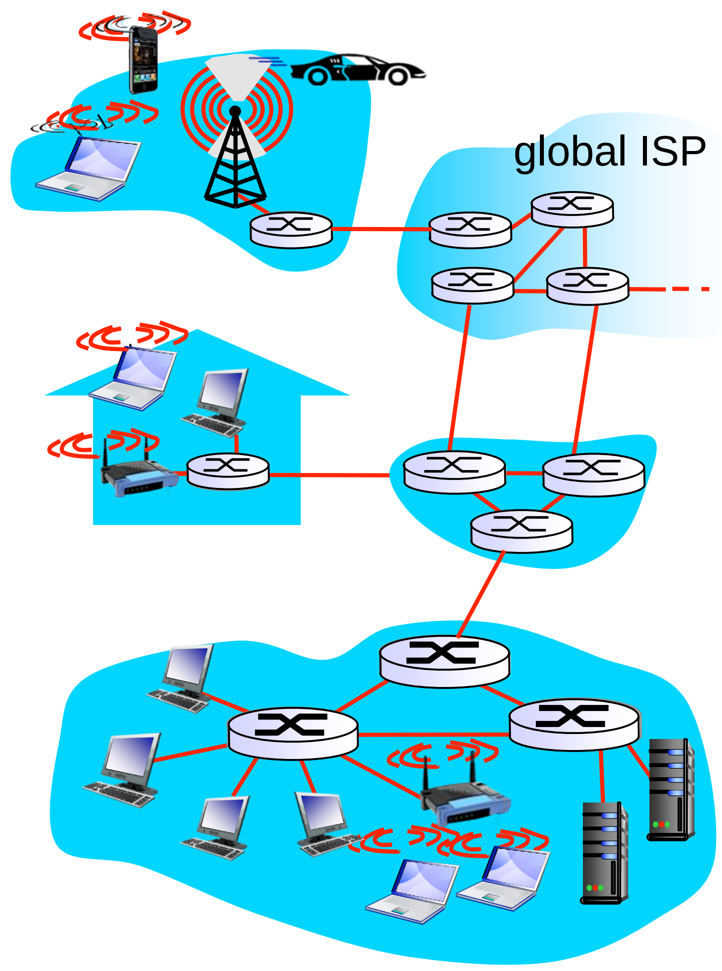
Protocol Layers

- **Top-Down Approach**



Link Layer: Introduction

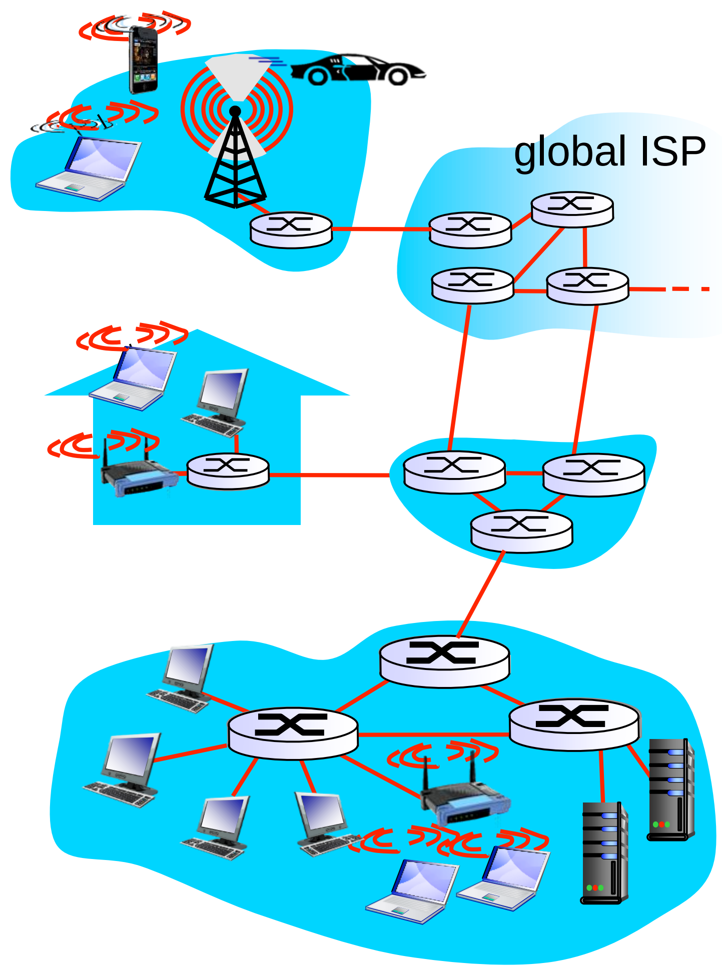
- **Terminology**
  - Hosts, routers, switches, access points are all nodes
  - Links connect adjacent nodes
    - Wired links
    - Wireless links
  - Layer-2 messages are transmitted as frames (encapsulates a datagram)



- **Data-link layer has responsibility of transferring datagram from one node to another physically adjacent node over a link**

Link Layer: Context

- **Datagrams can be transferred by different link protocols over different links**
  - For example, Ethernet on first link, frame relay on intermediate links, and 802.11 (wireless) on the last link



- **Each link protocol provides different services**
  - For example, some links may provide reliable data transfer where others may not

Link Layer Services

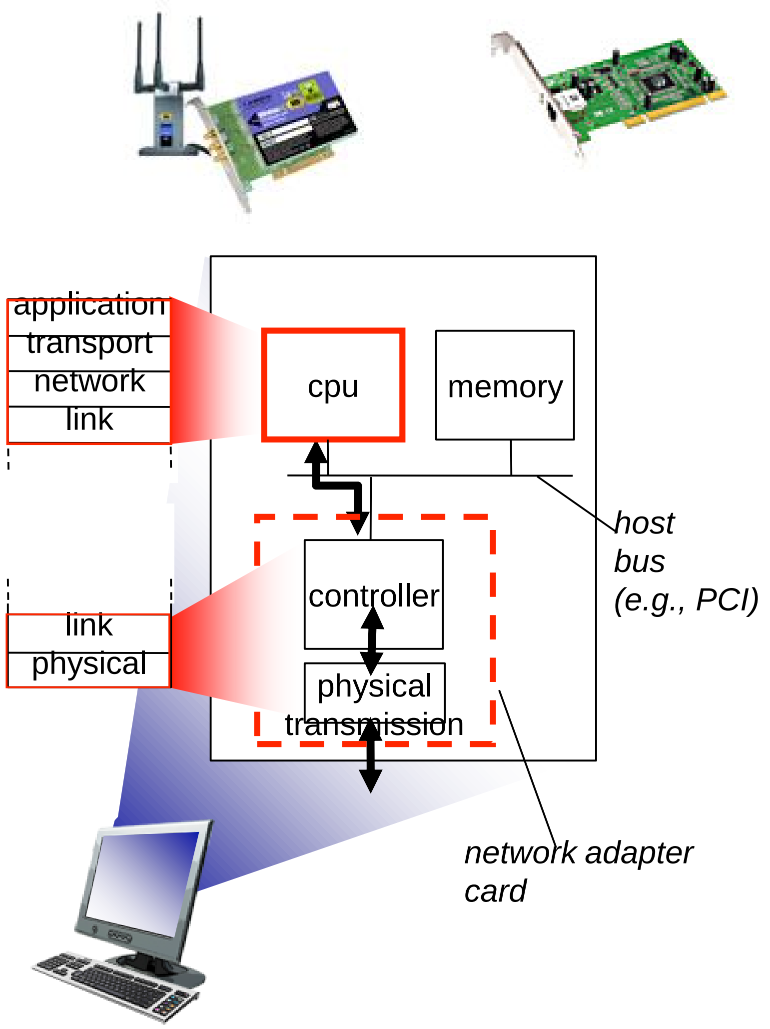
- **Framing:**
  - Encapsulate datagram into frame, adding header, trailer
  - “MAC” addresses used in frame headers to identify source and destination
    - Different from IP address
- **Link access:**
  - Coordinates frame transmission of many nodes using shared medium
- **Reliable delivery between adjacent nodes**
  - Provides reliable transmission on lossy links
    - Useful on wireless links where error rates are high
    - Seldom used on low bit-error link (fiber, some twisted pair)

Link Layer Services (Cont.)

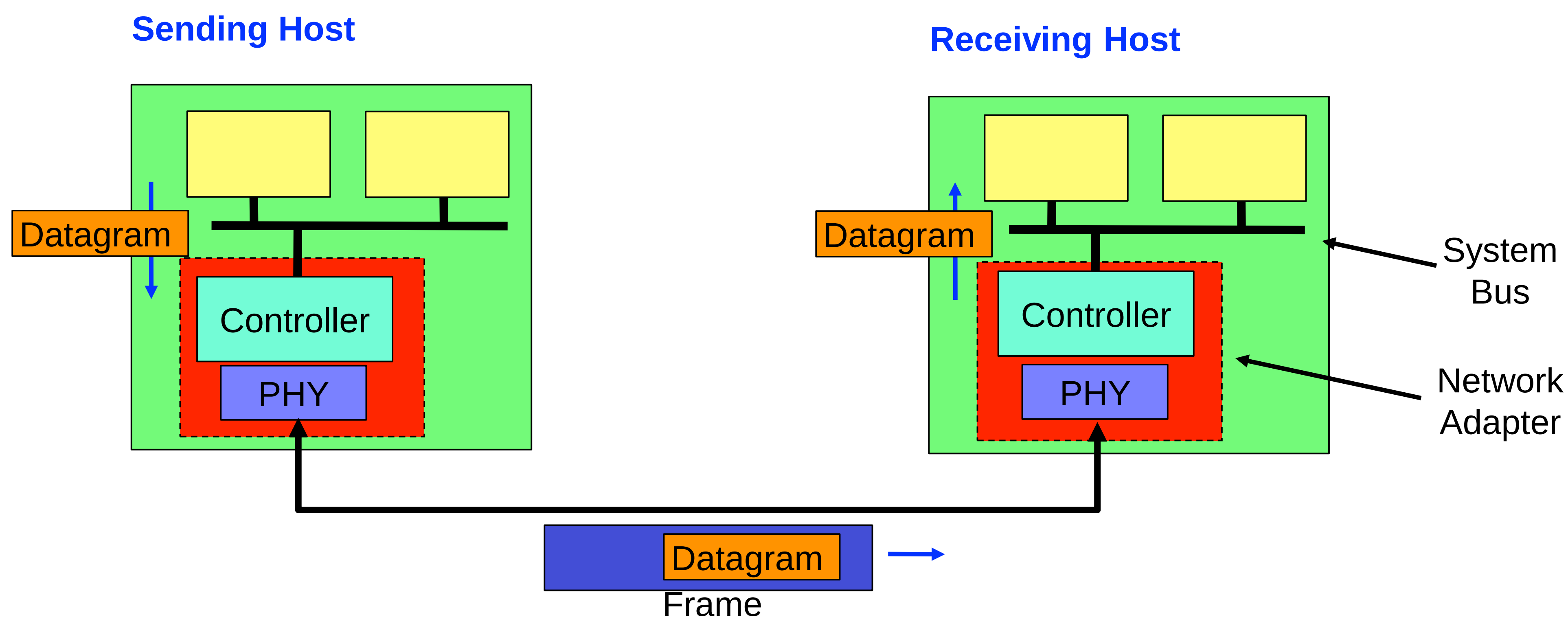
- **Flow control**
  - Controls pacing between adjacent sending and receiving nodes
- **Error detection**
  - Errors can be caused by signal attenuation, noise on links
  - Receiver detects presence of errors
    - May signals sender for retransmission
- **Error correction**
  - Receiver identifies and corrects bit error(s) without resorting to retransmission
- ***half-duplex and full-duplex***
  - with half duplex, nodes at both ends of link can transmit, but not at same time

Link Layer Implementation

- **Implemented in each and every host**
- **Link Layer implemented in network adapter (i.e. Network Interface Card (NIC) or chipset)**
  - NIC or chipset typically implements both link layer and physical layer
  - Single chip provides most of link layer services
- **Attaches into host's system buses**
- **Typically implemented as a combination of hardware, software, firmware**



Adapters Communicating



- **Sending side**
  - Encapsulates datagram in frame
  - Adds error checking bits, rdt info, flow control info, etc.
- **Receiving side**
  - Looks for bit errors, rdt, flow control info, etc.
  - Extracts datagram, passes to upper layer at receiving side

Overview of Link Layer

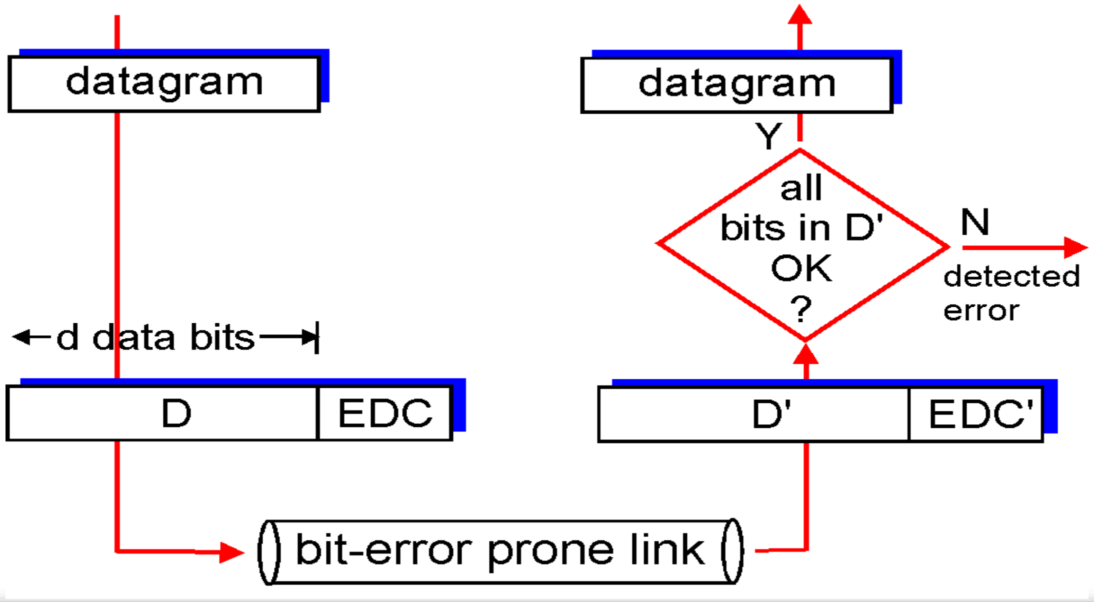
- **Introduction, Services**
- **Error Detection, Correction**
- **Multiple Access Protocols**
- **LANs**
  - Addressing, ARP
  - Ethernet
  - Switches
- **A Day in the Life**

Error Detection

- **Link layer provides error detection methods to detect errors that occur on a single link between nodes**
  - Protects original datagram and the frame fields added by the link layer
  - Different techniques possible for error detection
    - Parity checks
    - Checksums
    - Cyclic Redundancy Checks (CRC)
- **Error detection is not 100% reliable, may not always detect errors**
  - More EDC bits means more likely to detect/correct errors
  - More EDC bits also means more complex, time consuming to check/compute EDC bits

EDC = Error Detection and Correction bits (redundancy bits)

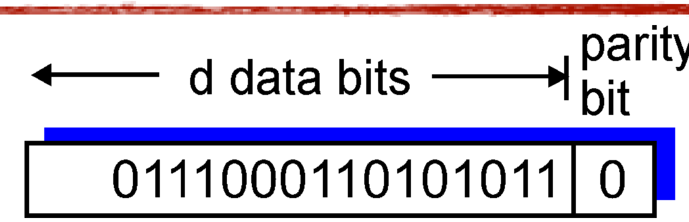
D = Data protected by error checking, may include header fields



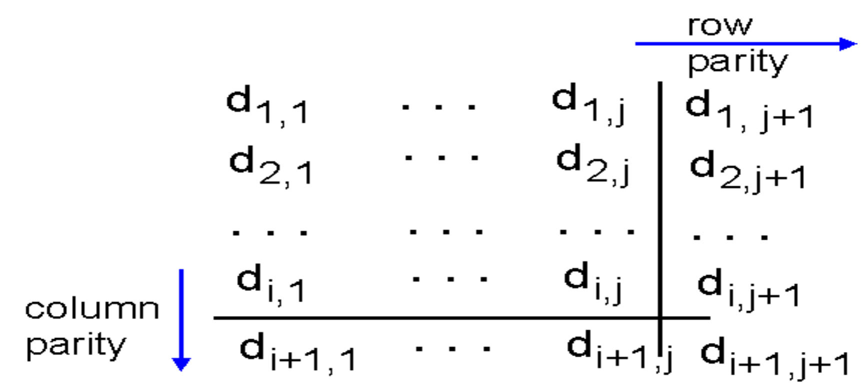
Parity Checking

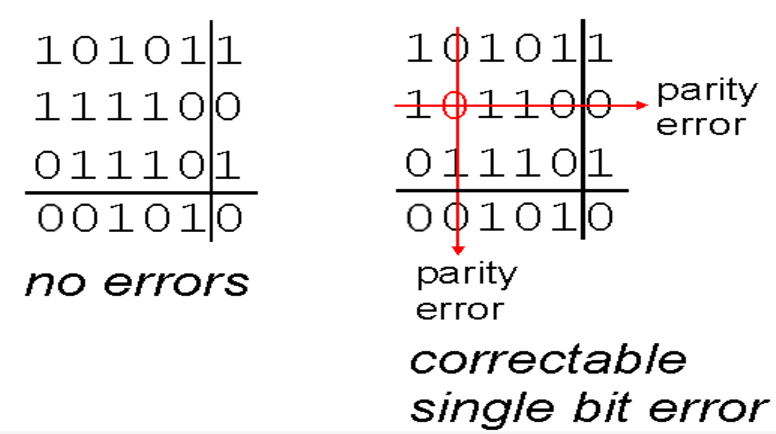
- **Simple form of error detection**
- **In single-bit parity scheme, add a single bit to data such that**
  - The total number of 1s in the data is an even number (for even parity)
  - The total number of 1s in the data is an odd number (for odd parity)
- **Can be performed in one or more dimensions**
  - Multidimensional parity allows for some error correction
- **It is possible to miss errors**
  - What if two 1s become 0s in an even parity scheme?

Single bit parity detects single bit errors



Two-dimensional bit parity detect and correct single bit errors



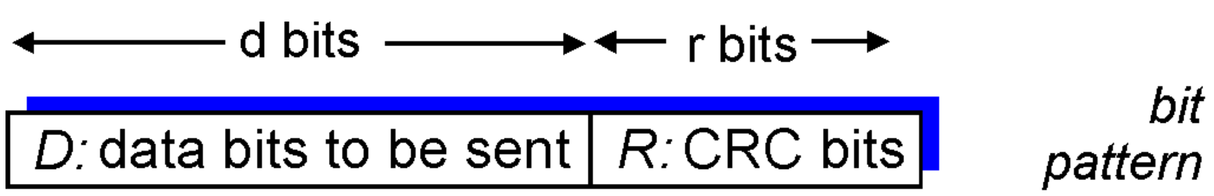


Internet Checksum (review)

- **Used to detect errors (e.g. flipped bits) in transmitted data**
- **Sender:**
  - Treat segment contents, including the header fields, as sequence of 16-bit integers
  - Perform one's complement sum of segment contents, then take one's complement of that sum
  - Insert checksum value into UDP checksum field
- **Receiver:**
  - Compute one's complement sum of received segment (including checksum field)
  - Check if computed sum equals **`0xFFFF`**
    - YES - no error detected. But may have errors nonetheless? More later ….
    - NO - error detected

Cyclic Redundancy Check (CRC)

- **More powerful error-detection coding**
  - Treat data bits, D, as a binary number
    - The data consists of d bits
  - Append CRC bits R
    - The CRC consists of r bits
  - Choose a r+1 bit pattern called the generator, G



- **Want to choose r CRC bits, R, such that**
  - Concatenated bits <D,R> is exactly divisible by G (using modulo-2 arithmetic)
  - Receiver knows G, divides <D,R> by G
    - If non-zero remainder then error detected!
- **Can detect all burst errors less than r+1 bits**
- **Widely used in practice (Ethernet, 802.11 WiFi, ATM)**

CRC Example (Sender's Computation)

- **Want:**

$$D\cdot 2^{r} \text{ XOR } R = nG$$

- **Equivalent to:**

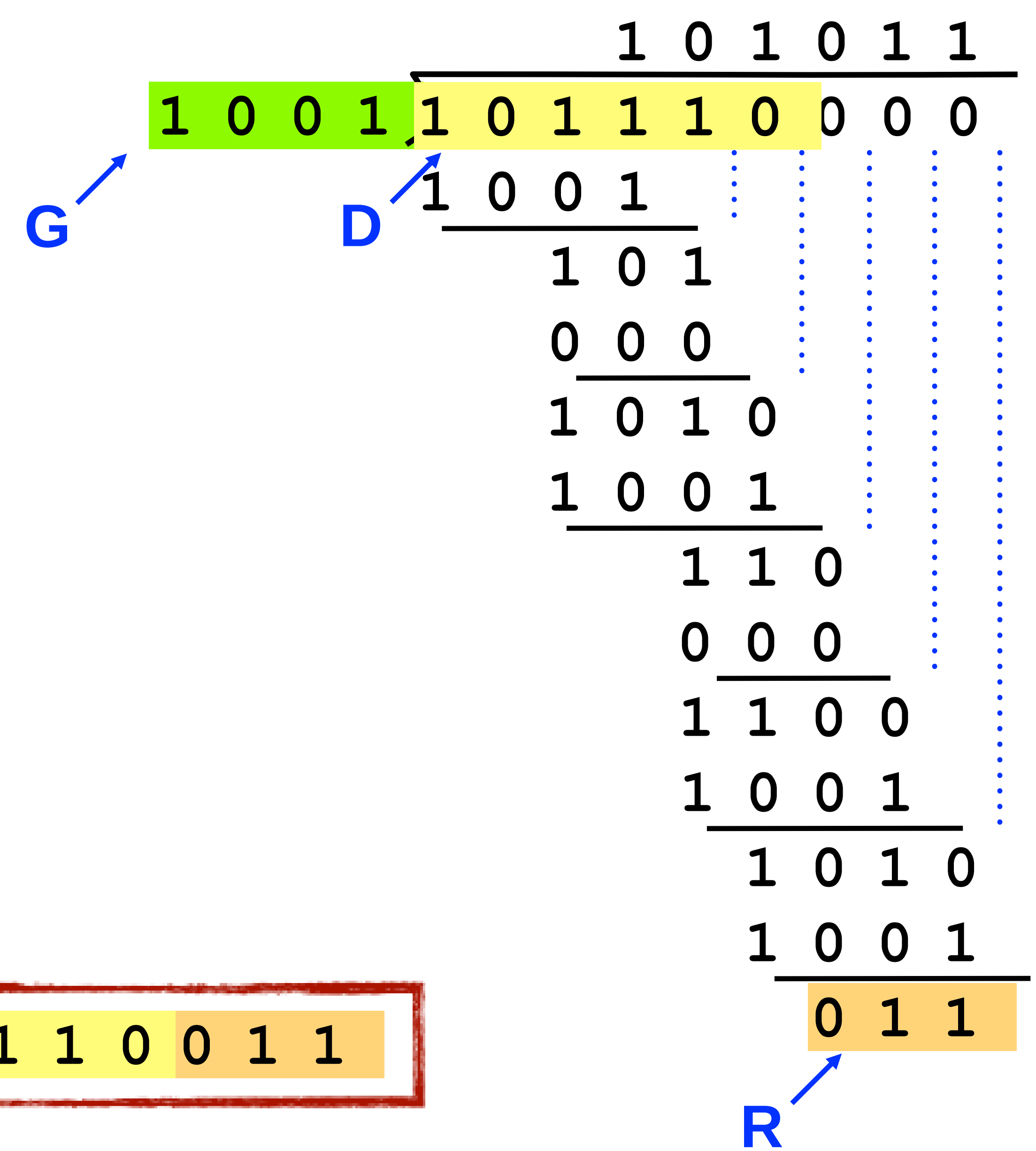
$$D\cdot 2^{r} = nG \text{ XOR } R$$

- **Equivalently:**
  - If $D\cdot 2^{r}$ is divided by G, want remainder R to satisfy:

$$R = D\cdot 2^{r} \bmod G$$

- **Data transmitted is <D,R>:** 1 0 1 1 1 0 0 1 1

Example with $r = 3$



CRC Example (Receiver’s Computation)

- **The receiver verifies the data doing a similar computer**
  - Divide received data, <D,R> by the same generator, G, used by the source
    - Remainder of 0 indicates there were no errors
    - Non-zero remainder indicates an error occurred

```
<D,R> ÷ G == 0
```

```
                 1 0 1 0 1 1
         ___________________
1 0 0 1 ) 1 0 1 1 1 0 0 1 1
          1 0 0 1
          -------
            1 0 1
            0 0 0
            -----
            1 0 1 0
            1 0 0 1
            -------
              1 1 0
              0 0 0
              -----
              1 1 0 1
              1 0 0 1
              -------
                1 0 0 1
                1 0 0 1
                -------
                  0 0 0
```

G

<D, R>

Remainder

Overview of Link Layer

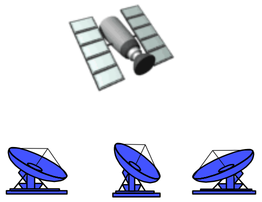
- **Introduction, Services**
- **Error Detection, Correction**
- **Multiple Access Protocols**
- **LANs**
  - Addressing, ARP
  - Ethernet
  - Switches
- **A Day in the Life**

Multiple Access Links

- **Two types of links**
  - Point-to-point link
    - PPP (point-to-point protocol) for dial-up access
    - Point-to-point link between Ethernet switch, host
    - Single sender, single receiver
  - Broadcast link (shared wire or medium)
    - Old-fashioned Ethernet
    - Upstream HFC (hybrid fiber-coaxial)
    - 802.11 wireless LAN
    - Multiple sending, receiving nodes
      - How should this be handled?

Shared wire
(e.g. cabled Ethernet)

Shared RF
(e.g. 802.11 WiFi)



Shared RF
(e.g. satellite)

Multiple Access Protocols

- **Broadcast links allows multiple senders and multiple receivers to share the same broadcast channel**
- **Two or more simultaneous transmissions by nodes may cause interference**
  - A collision occurs if multiple nodes attempt to broadcast at the same time
    - Node detects collision if it receives two or more signals at the same time
- **A multiple access protocol is designed to allow multiple senders/receivers to use the same channel**
  - Distributed algorithm that determines how nodes share channel
    - Determines when node can transmit data on shared medium
  - Communication about channel sharing must use channel itself
    - No out-of-band channel for coordination

An Ideal Multiple Access Protocol

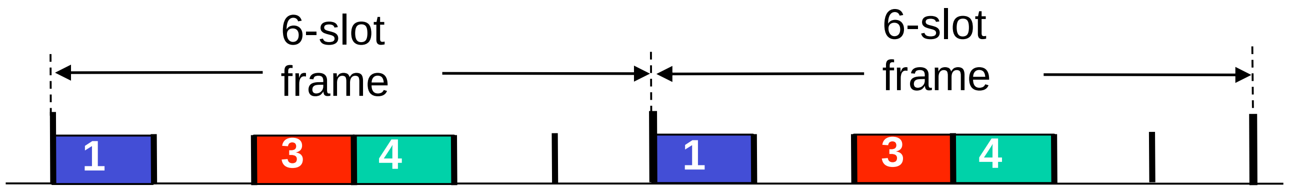
- **Given a broadcast channel of rate R bps**
- **Desirable Characteristics:**
  - When one node wants to transmit, it can send at rate R
  - When M nodes want to transmit, each can send at average rate R/M
  - Fully decentralized
    - No special node to coordinate transmissions
    - No synchronization of clocks, slots
  - Simple and easy to implement

Multiple Access Protocols

- **Three broad classes of multiple access protocols**
  - Channel partitioning protocols
    - Divide channel into smaller “pieces” (time slots, frequency, code)
    - Allocate a piece to each node for exclusive use
  - Random access protocols
    - Channel not divided
    - Collisions are possible
      - Must be able to recover from collisions
  - Taking-turns protocols
    - Nodes take turns sending data
    - Nodes with more to send can take longer turns

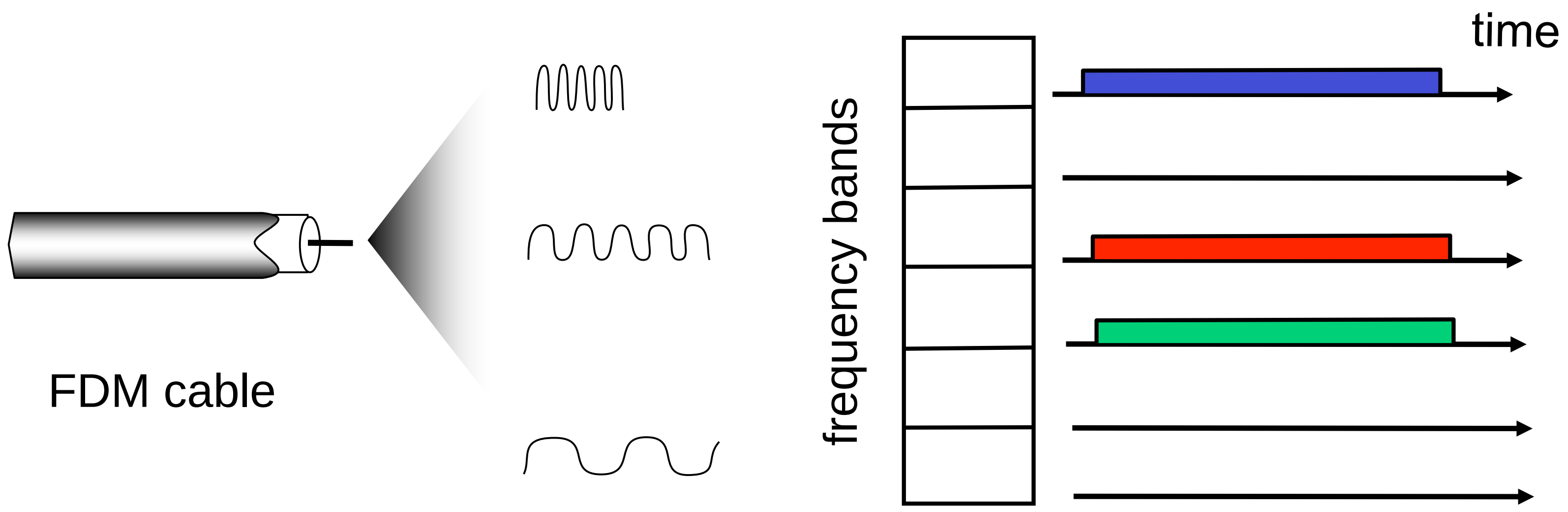
Channel Partitioning Protocols: TDMA

- **TDMA: Time Division Multiple Access**
  - Nodes are provided access to channel in "rounds"
  - Each station gets fixed length slot in each round
    - Length = time required to transmit packet
  - Unused slots go idle
    - A node cannot use another node’s time slot
    - If a node has no data to send, it’s time slot is wasted
  - Example: 6-station LAN, 1,3,4 have packets, slots 2,5,6 idle



Channel Partitioning Protocols: FDMA

- **FDMA: Frequency Division Multiple Access**
  - Channel spectrum divided into frequency bands
  - Each station assigned fixed frequency band
    - Multiple stations can transmit simultaneously since they are on different frequencies
  - Unused transmission time in frequency bands go idle
  - Example: 6-station LAN, 1,3,4 have packet, frequency bands 2,5,6 idle



Random Access Protocols

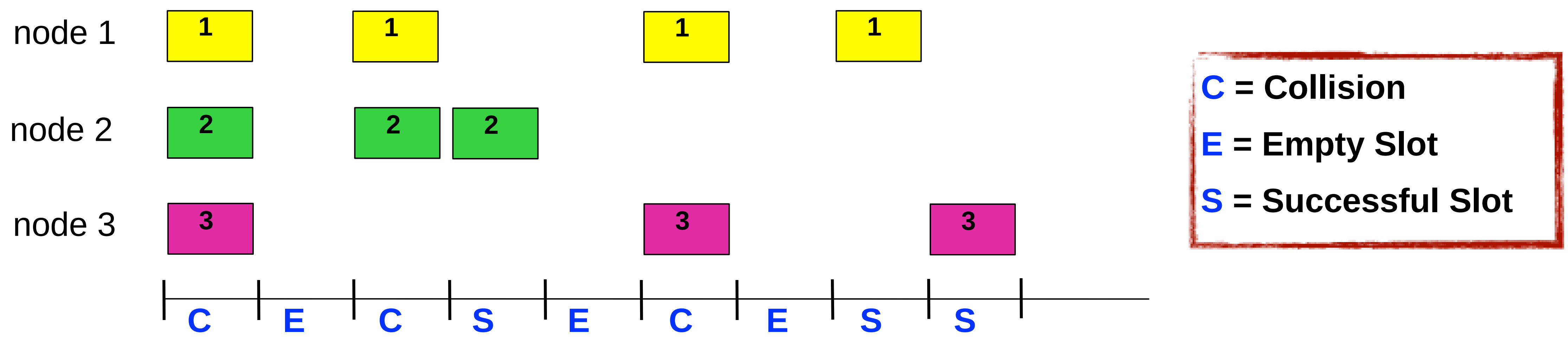
- **When a node has a packet to send**
  - Transmit at full channel data rate R
  - No a priori coordination among nodes
- **If two or more nodes are transmitting a collision may occur**
- **Random access protocols specify**
  - How collisions are detected
  - How to recover from collisions (e.g. via randomly delayed retransmissions)
- **Examples of random access MAC protocols:**
  - Slotted ALOHA
  - ALOHA
  - CSMA, CSMA/CD, CSMA/CA

Random Access Protocols: Slotted ALOHA

- **Assumptions:**
  - All frames are the same size
  - Time divided into equal size slots (time to transmit 1 frame)
  - Nodes start to transmit only at beginning of slot
  - Nodes are synchronized so each node knows when slots begin
  - If 2 or more nodes transmit in the same slot, all nodes detect a collision before end of slot

- **Operation:**
  - When node needs to send a frame it waits for the beginning of the next slot and then transmits the frame
    - If no collision is detected, the frame was transmitted successfully; node can send new frame in next slot
    - If collision is detected node retransmits frame in each subsequent slot with some probability until success

Random Access Protocols: Slotted ALOHA



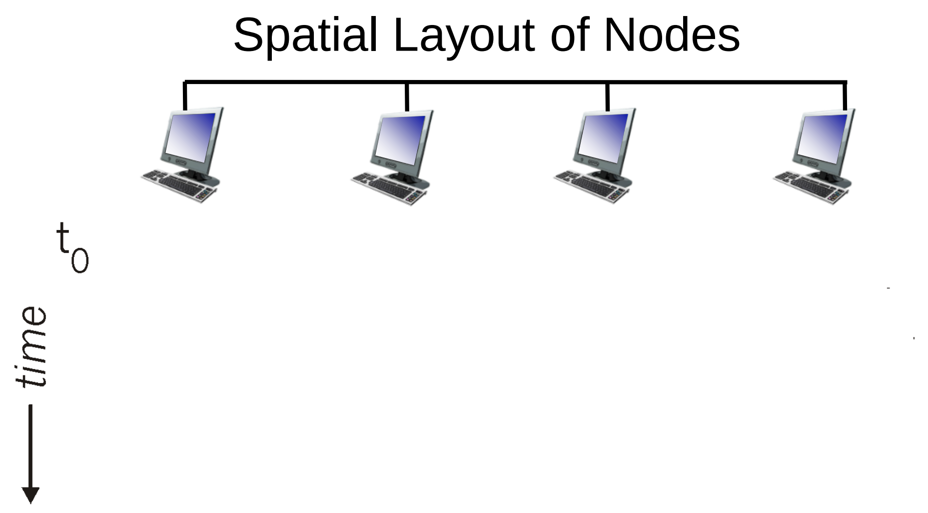
- **Pros:**
  - Single active node can continuously transmit at full rate of channel
  - Highly decentralized
  - Simple to implement
- **Cons:**
  - Collisions are possible which wastes slots
  - May have idle slots
  - Nodes may be able to detect collision in less than time to transmit frame
  - Must synchronize nodes
  - Only 37% of slots result in successful transmissions

Random Access Protocols: CSMA

- **Carrier Sense Multiple Access (CSMA)**
  - A node listens (carrier sense) to the channel before sending data
    - If node senses that another node is transmitting, then don’t transmit frame
    - If node doesn’t sense another node transmitting, then that node transmits its entire frame
  - The human analogy: don’t interrupt someone else who’s speaking!

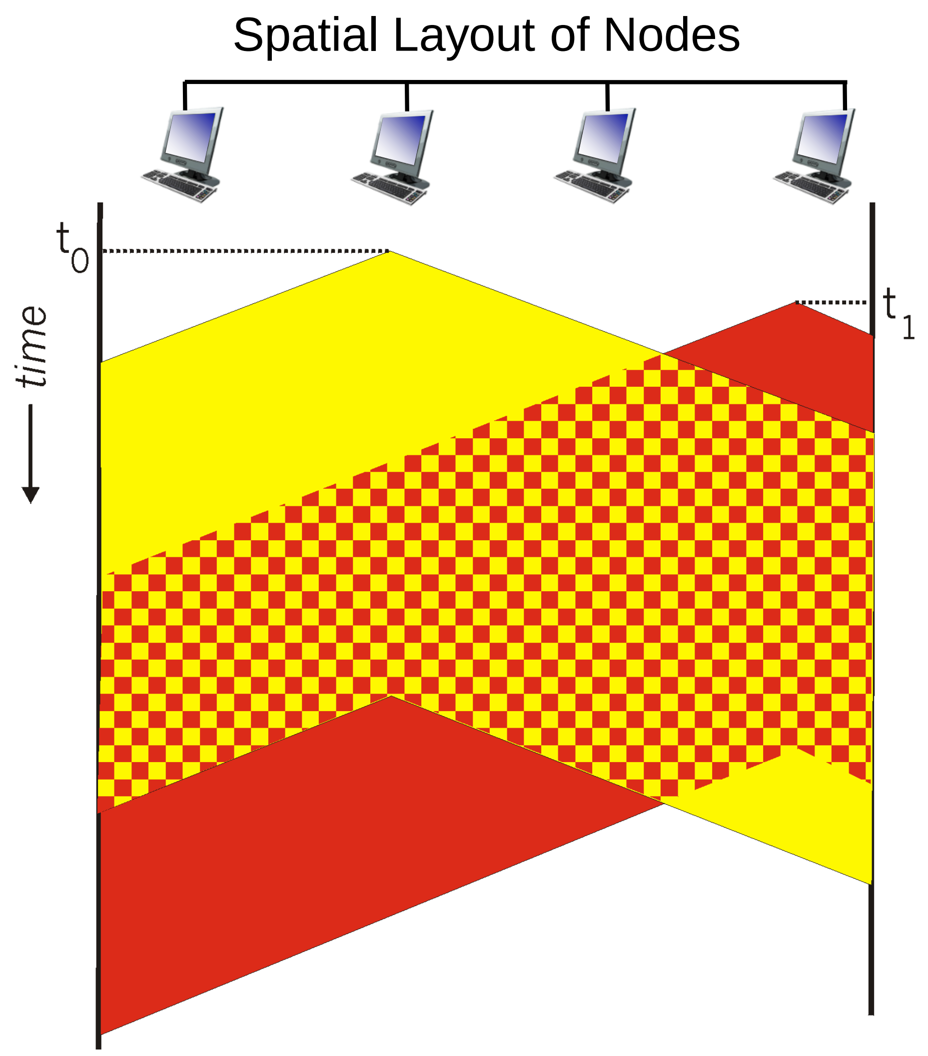
CSMA Collisions

- **Collisions can still occur**
  - Propagation delay means two nodes may not hear each other's transmission before transmitting
- **If collision occurs then the entire packet transmission time is wasted**
  - Distance and propagation delay play role in in determining collision probability



CSMA Collisions

- **Collisions can still occur**
  - Propagation delay means two nodes may not hear each other's transmission before transmitting
- **If collision occurs then the entire packet transmission time is wasted**
  - Distance and propagation delay play role in in determining collision probability

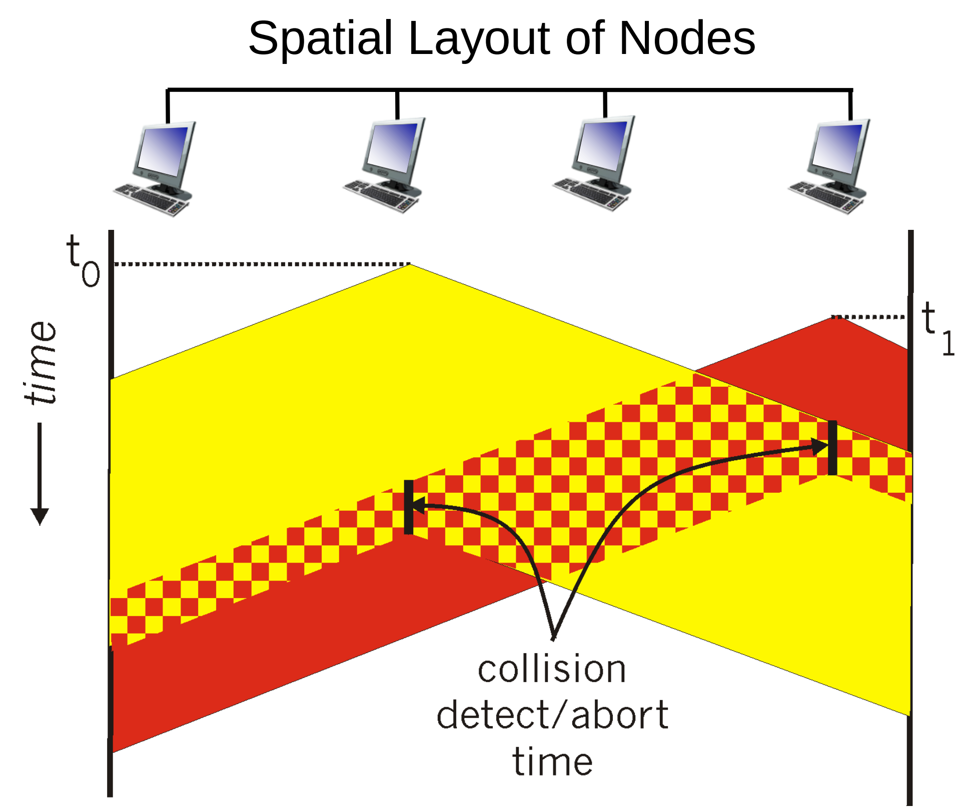


Random Access Protocols: CSMA/CD

- **Carrier Sense Multiple Access with Collision Detection**
  - Carrier sensing and deferral just like CSMA
    - Collisions can be detected within short time
    - Colliding transmissions aborted immediately, reducing channel wastage
  - Collision Detection
    - Easy in wired LANs - measure signal strengths, compare transmitted, received signals
    - Difficult in wireless LANs - received signal strength overwhelmed by local transmission strength
  - The human analogy: the polite conversationalist

CSMA/CD Collisions

- **If collision occurs:**
  - Node stops transmitting as soon as collision is detected



Ethernet CSMA/CD Algorithm

(1) NIC receives datagram from Network Layer and creates a frame

(2) NIC checks channel

- If NIC senses channel is idle, it starts frame transmission
- If NIC senses channel busy, it waits until channel is 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, it aborts and sends a jam signal

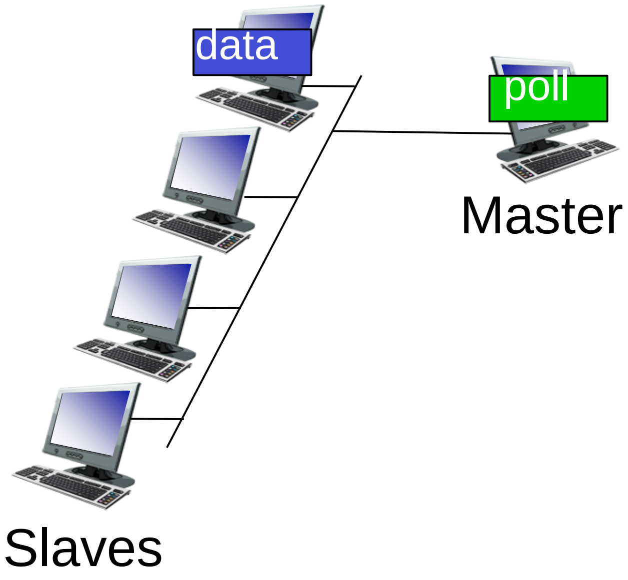
- After aborting, NIC enters binary (exponential) backoff:
  - After m^{th} collision, NIC chooses some value, K, at random from $\{0,1,2, \ldots, 2^m-1\}$. NIC waits $K \cdot 512$ bit times, returns to Step 2
  - Longer backoff interval with more collisions

Taking-turns Protocols

- **Channel partitioning protocols**
  - Share channel efficiently and fairly at high load
  - Inefficient at low load
    - Delay in channel access
    - 1/N bandwidth allocated even if only 1 active node
- **Random access protocols**
  - Efficient at low load - single node can fully utilize channel
  - High load results in collision overhead
- **Taking-turns protocols**
  - Look for best of both worlds

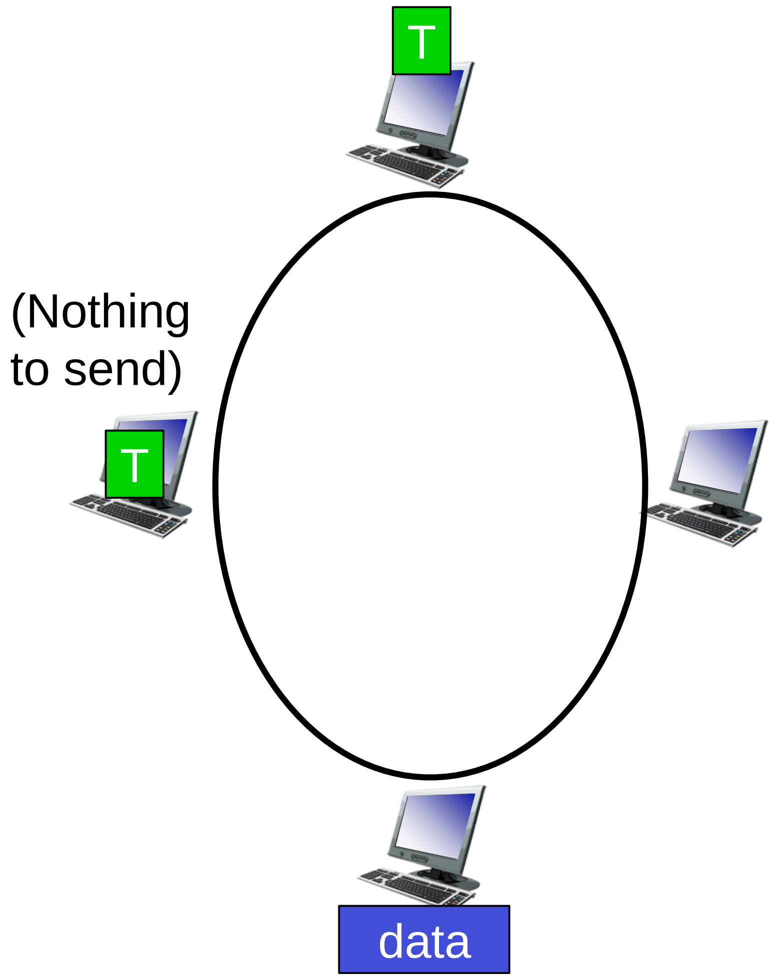
Taking-turns Protocols

- **Polling:**
  - Master node “invites” slave nodes to transmit in turn
    - Slave node transmit data if it has any
    - Master can detect when slave is done because it’s listening
  - Typically used with “dumb” slave devices
  - Concerns
    - Polling overhead
    - Latency
    - Single point of failure (the master node)



Taking-turns Protocols

- **Token passing:**
  - A control token is passed from one node to next sequentially
    - Token is a small, special-purpose frame
    - If nothing to send, pass token along
    - If data to send, keep token until done transmitting data
  - Concerns:
    - Token overhead
    - Latency
    - Single point of failure (the token)



Summary of MAC protocols

- ***channel partitioning*, by time, frequency or code**
  - Time Division, Frequency Division
- ***random access* (dynamic),**
  - ALOHA, S-ALOHA, CSMA, CSMA/CD
  - carrier sensing: easy in some technologies (wire), hard in others (wireless)
  - CSMA/CD used in Ethernet
  - CSMA/CA used in 802.11
- ***taking turns***
  - polling from central site, token passing
  - Bluetooth, FDDI, token ring

Overview of Link Layer

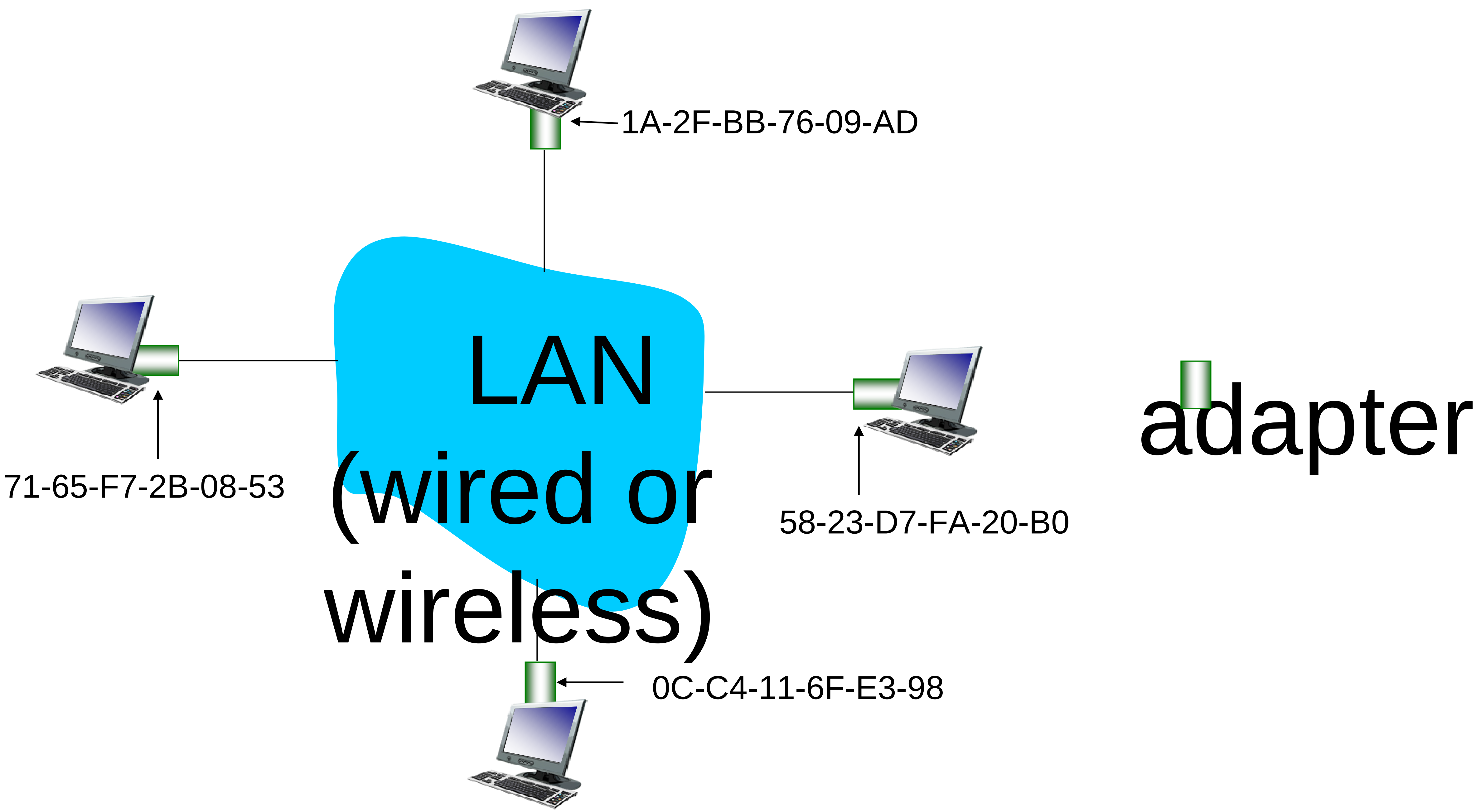
- **Introduction, Services**
- **Error Detection, Correction**
- **Multiple Access Protocols**
- **LANs**
  - Addressing, ARP
  - Ethernet
  - Switches
- **A Day in the Life**

MAC Addresses

- **Media Access Control address (also known as link-layer address, Ethernet address, or physical address)**
  - Used to address link-layer frames to destination
  - A 48-bit (6-byte) value that is associated with a physical NIC
    - Example: 1A-2F-BB-76-09-AD
  - MAC address burned in NIC ROM (sometimes software settable)
  - No two NICs should have the same MAC address
    - Even though sometimes they do, just make sure they're no on the same network
  - Unlike and IP address, a MAC address does NOT change when a host moves from network to network
  - A host on a network "listens" to ALL frames but ignores frames that are not addressed to it
    - Frames that are addressed to a host are passed up to the Network Layer

LAN addresses and ARP

each adapter on LAN has unique *LAN* address

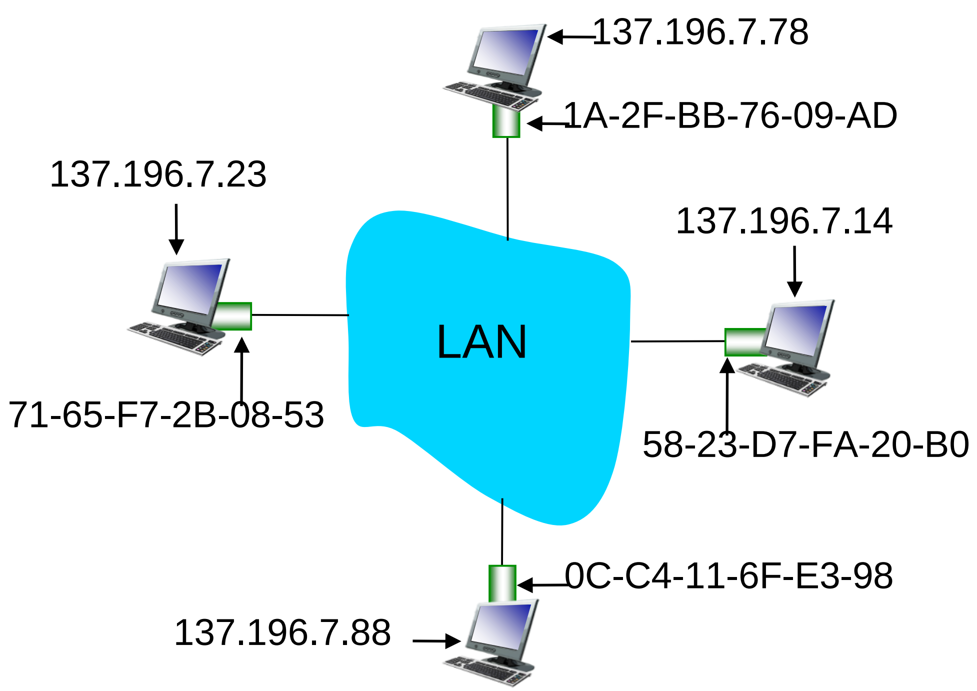


LAN addresses (more)

- **MAC address allocation administered by IEEE**
- **manufacturer buys portion of MAC address space (to assure uniqueness)**
- **analogy:**
  - MAC address: like Social Security Number
  - IP address: like postal address
- **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

ARP: Address Resolution Protocol

Question: How can a host determine the MAC address of a destination machine knowing only its IP address?



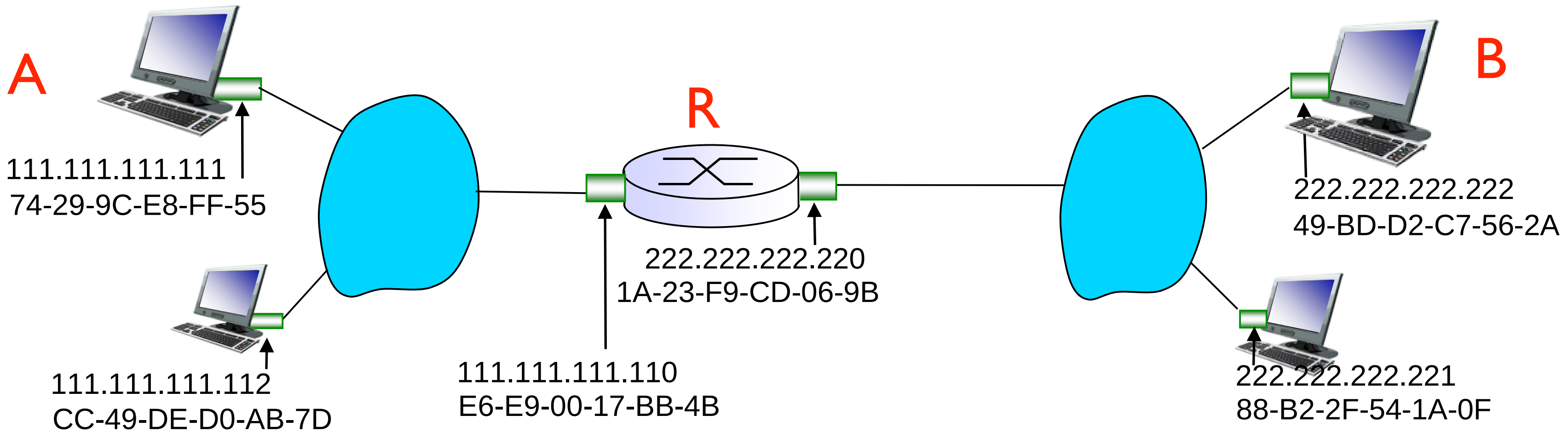
- **ARP table - every IP node (hosts and routers) on LAN maintains an ARP table**
  - IP/MAC address mappings for some LAN nodes:
    - < IP address , MAC address , TTL >
  - TTL (Time To Live) represents the time after which address mapping will be forgotten (typically 20 minutes)

ARP Protocol to Find Host on Same LAN

- **Host A wants to send a datagram to Host B**
  - Host B's MAC address not in Host A's ARP table
- **Host A broadcasts an ARP Request packet, containing Host B's IP address**
  - "Who has this IP address? What is your physical address?"
  - To broadcast, set destination MAC address to FF-FF-FF-FF-FF-FF
  - All nodes on the LAN receive ARP query
    - If IP address does not match a hosts IP address, the host just ignores the ARP request
- **Host B receives the ARP packet and replies to Host A with its MAC address**
  - "Hey, that's my IP address, here's my physical address."
  - Frame is sent directly to Host A's MAC address (unicast)
- **Host A caches 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 user or network administrator

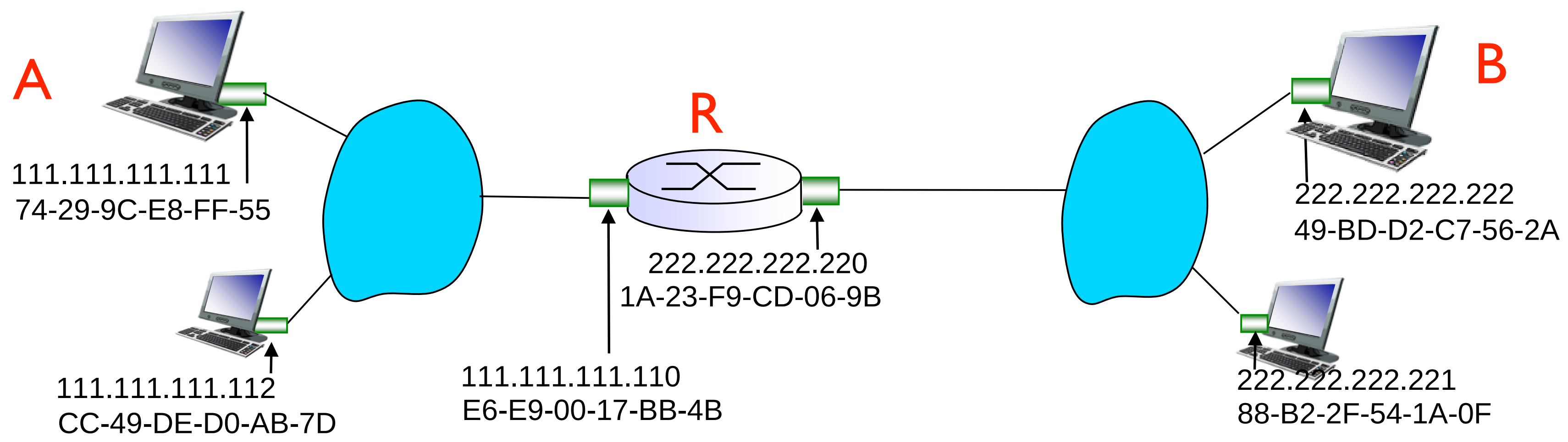
ARP Protocol to Find Host on Different LAN

- **Not all hosts are on the same LAN**
- **What if Host A wants to send a message to a host on a different LAN, Host B?**
  - Can't send directly to Host B's physical address because Host B is not on the same network
  - Can't ARP for Host B's physical address because Host B is not on the same network



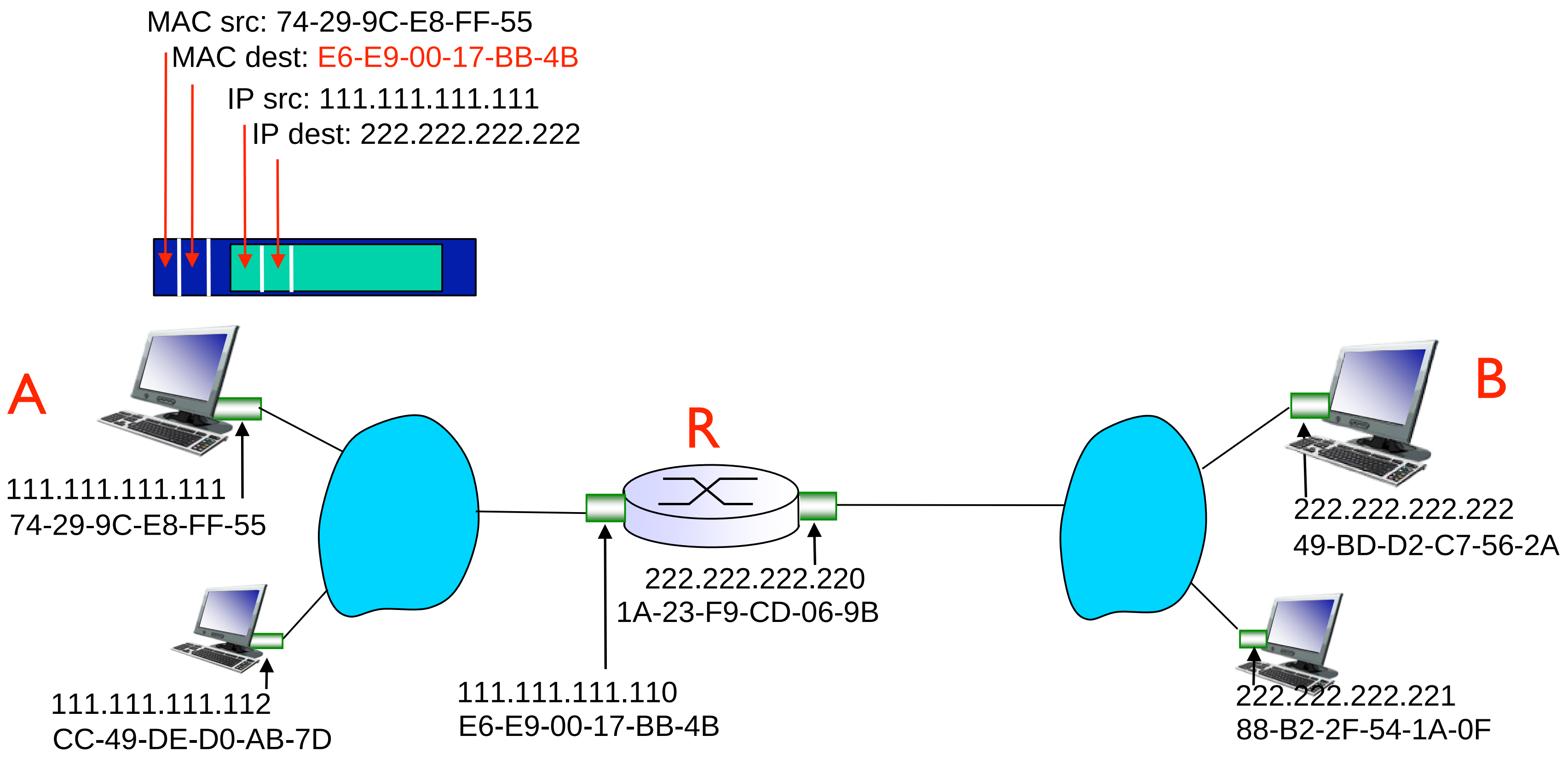
ARP Protocol to Find Host on Different LAN

- **Host A recognizes that IP address of Host B is on a different subnet**
  - Instead of sending frame directly to Host B, send the frame to router that knows path to Host B (router R)
  - But wait, Host A now needs to know the MAC address of Router R!
    - Host A can use ARP to find the MAC address of Router R



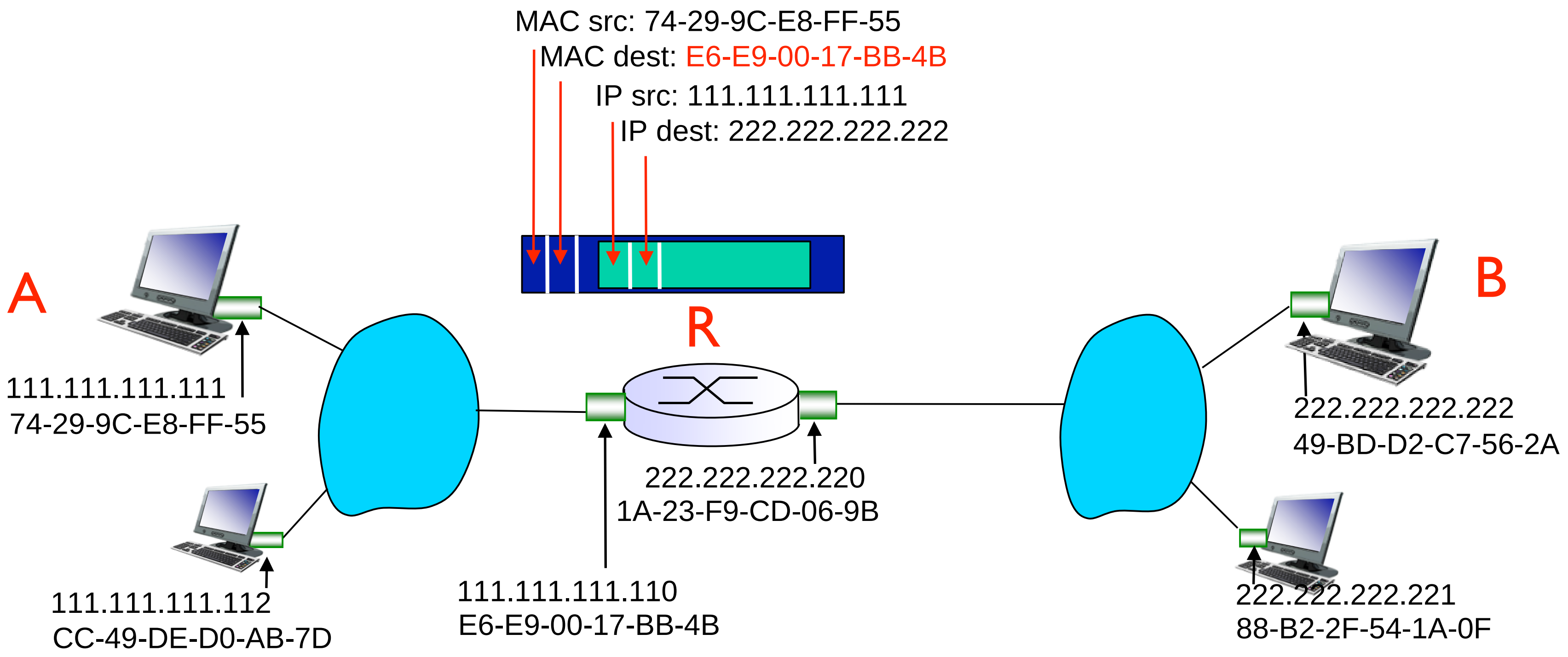
ARP Protocol to Find Host on Different LAN

- **Next, Host A puts the IP datagram in a link-layer frame with the following information:**
  - IP Source Address = Host A's IP Address
  - IP Destination Address = Host B's IP Address
  - Source MAC Address = Host A's MAC Address
  - Destination MAC Address = Router R's MAC Address



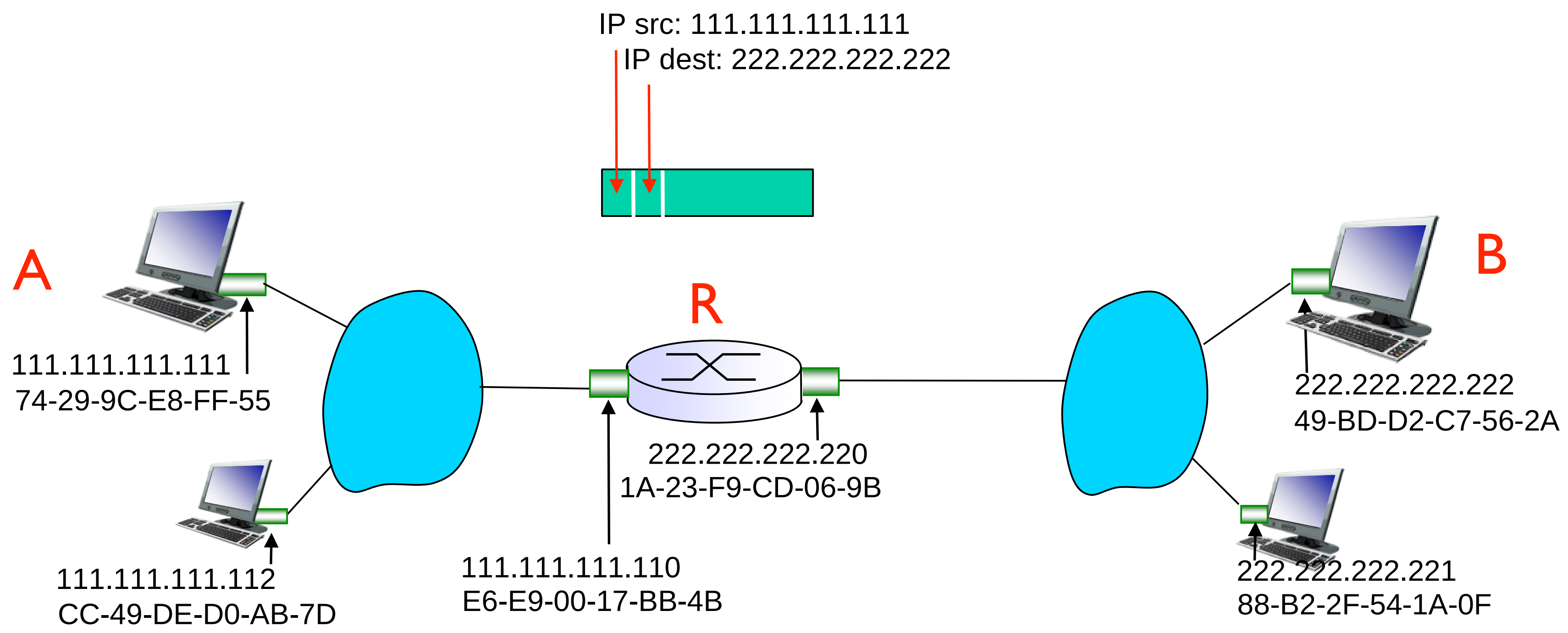
ARP Protocol to Find Host on Different LAN

- **Frame is sent from Host A to Router R**



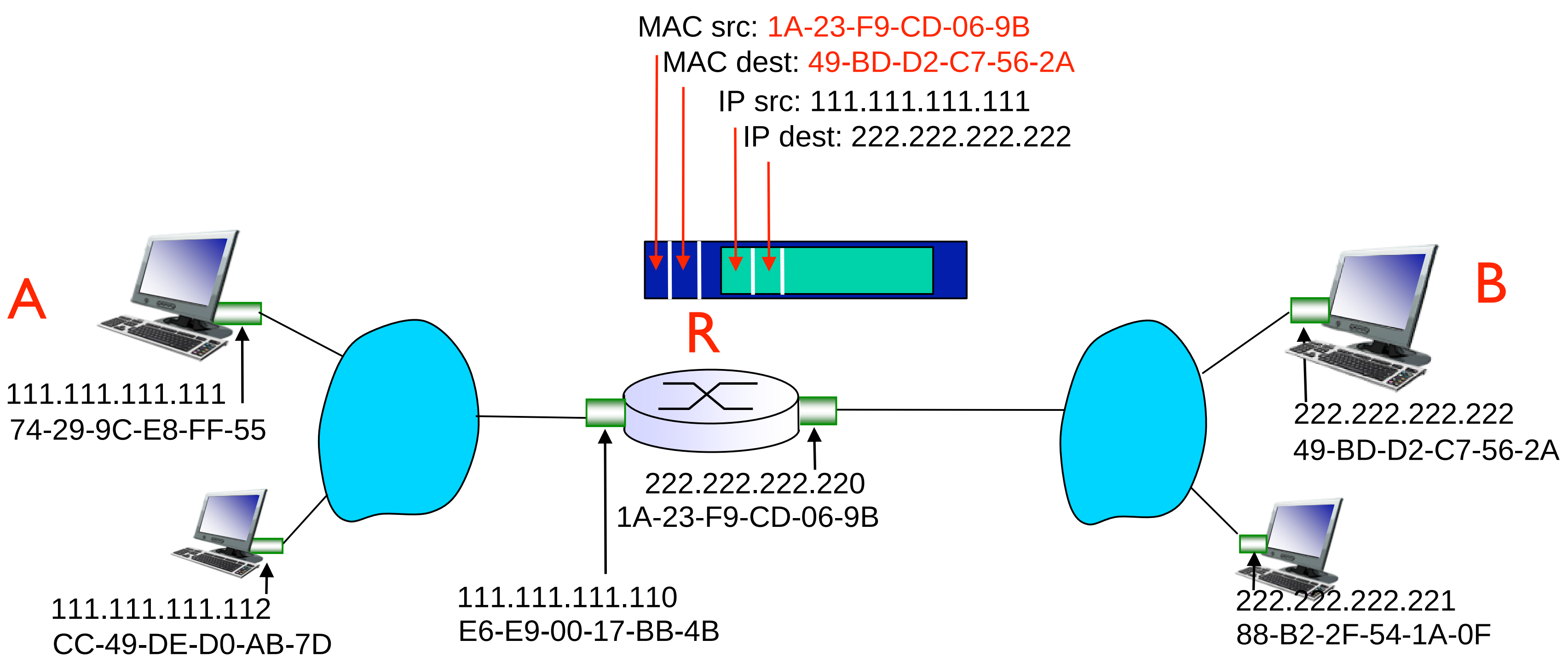
ARP Protocol to Find Host on Different LAN

- **When frame is received at Router R, datagram is removed from frame and passed up to Network Layer**
- **Router must then encapsulate the datagram in a NEW link-layer frame to be sent on the 222.222.222.0/24 subnet**
  - But wait, Router R now needs to know the MAC address of Host B
    - Router R can use ARP to find the MAC address of Host B



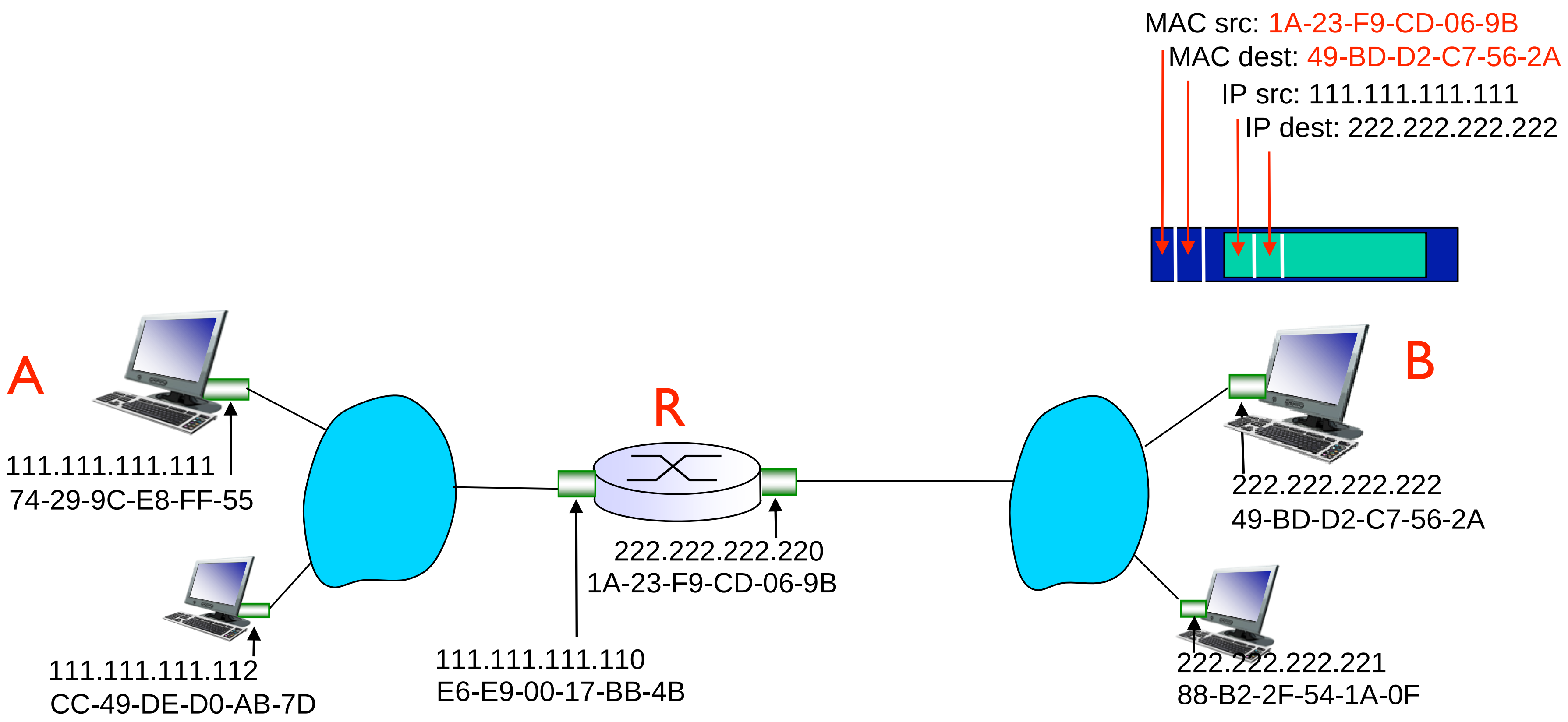
ARP Protocol to Find Host on Different LAN

- **After successfully ARPing for Host B, Router R can encapsulate the datagram in a new frame destined for Host B**



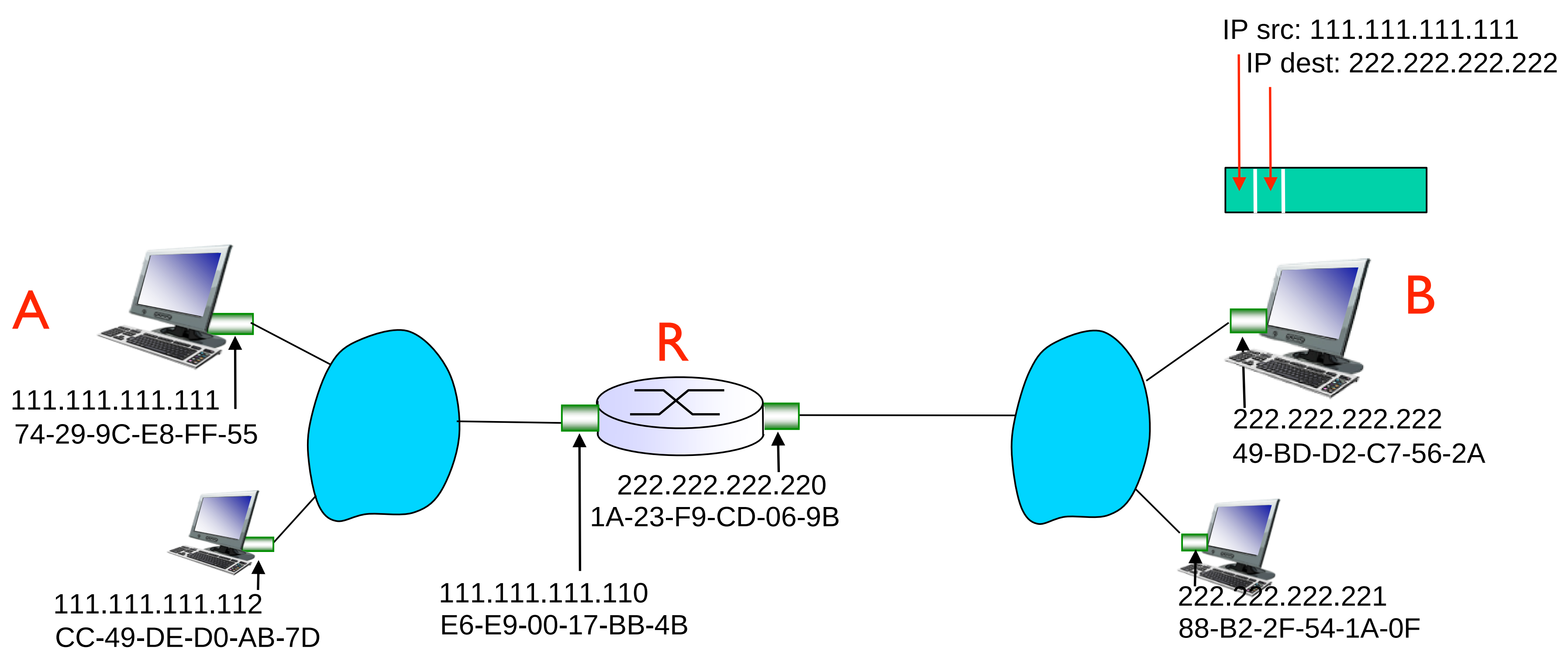
ARP Protocol to Find Host on Different LAN

- **Frame is sent from Router R to Host B**



ARP Protocol to Find Host on Different LAN

- **Host B can then extract the datagram from the frame and send it up to the Network Layer for further processing**

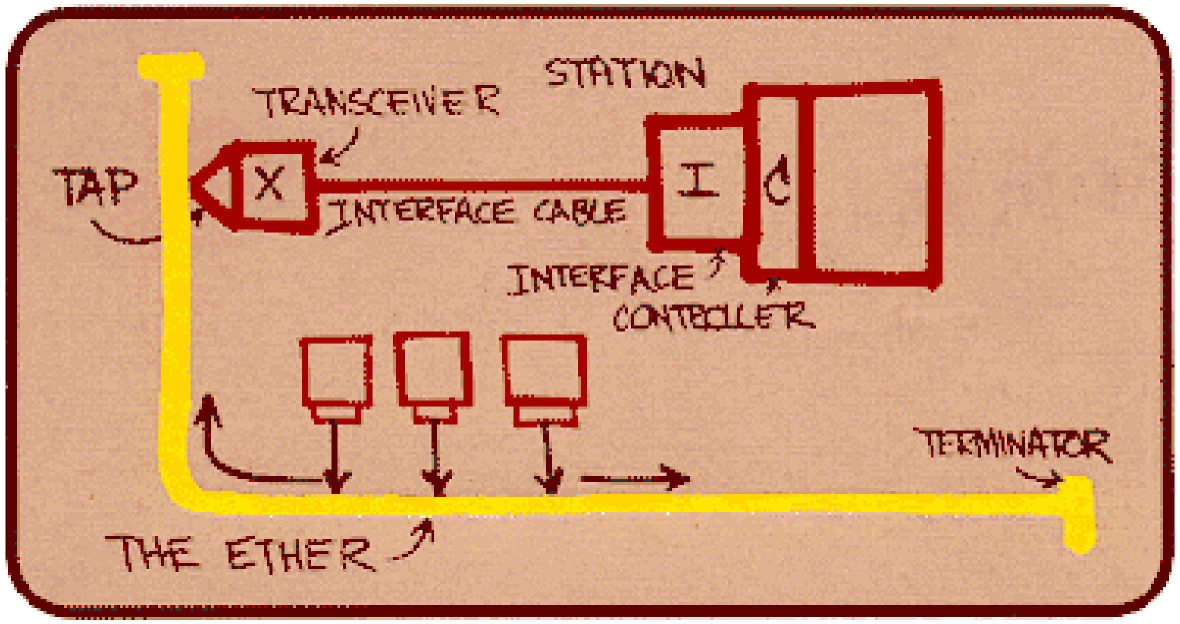


Overview of Link Layer

- **Introduction, Services**
- **Error Detection, Correction**
- **Multiple Access Protocols**
- **LANs**
  - Addressing, ARP
  - Ethernet
  - Switches
- **A Day in the Life**

Ethernet

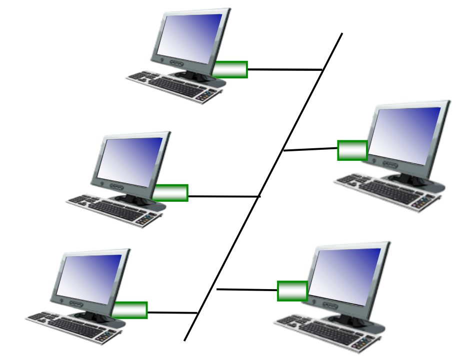
- **The dominant wired LAN technology**
  - Cheap $20 for NIC
  - First widely used LAN technology
  - Simpler, cheaper than token-ring LANs and ATM
  - Kept up with speed race: 10 Mbps - 10 Gbps



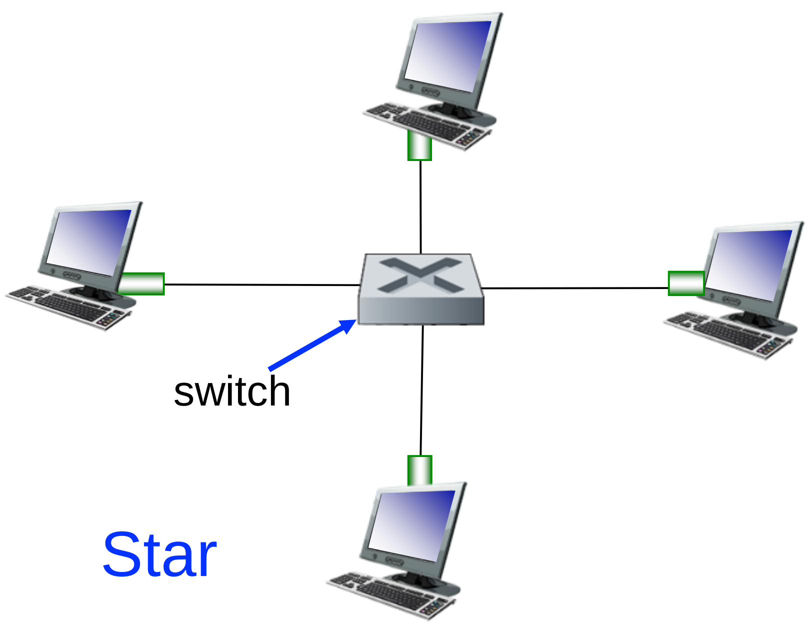
Metcalfe's Ethernet sketch

Ethernet: Physical Topology

- **Bus topology**
  - Popular through mid 1990s
  - All nodes in same collision domain
    - Can collide with each other
  - Network segments connected with hubs
- **Star topology (switched Ethernet)**
  - Most prevalent topology today
  - Active switch in center
  - Each “spoke” runs independent of others
    - Nodes do not collide with each other



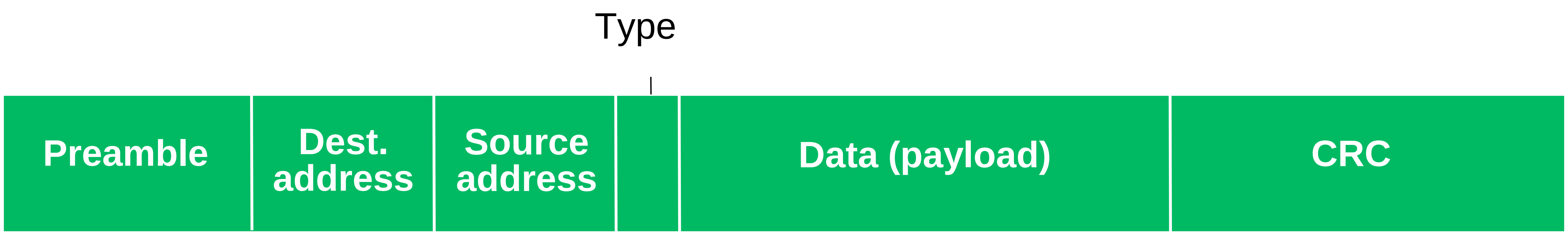
Bus



Star

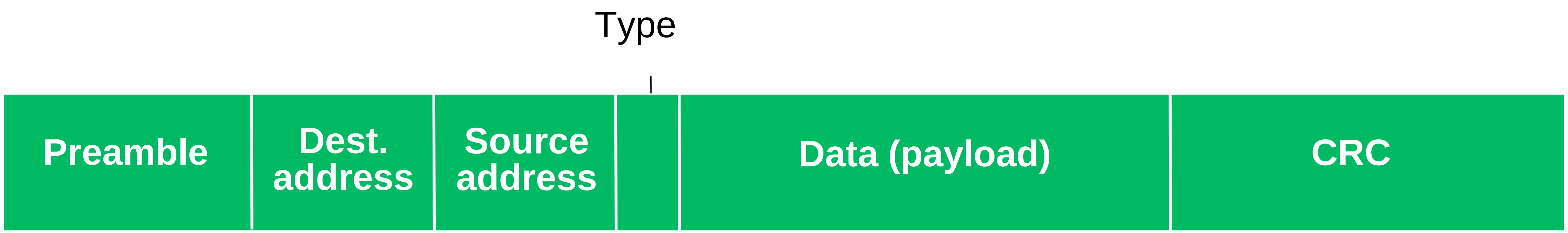
Ethernet Frame Structure

- **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 (Cont.)



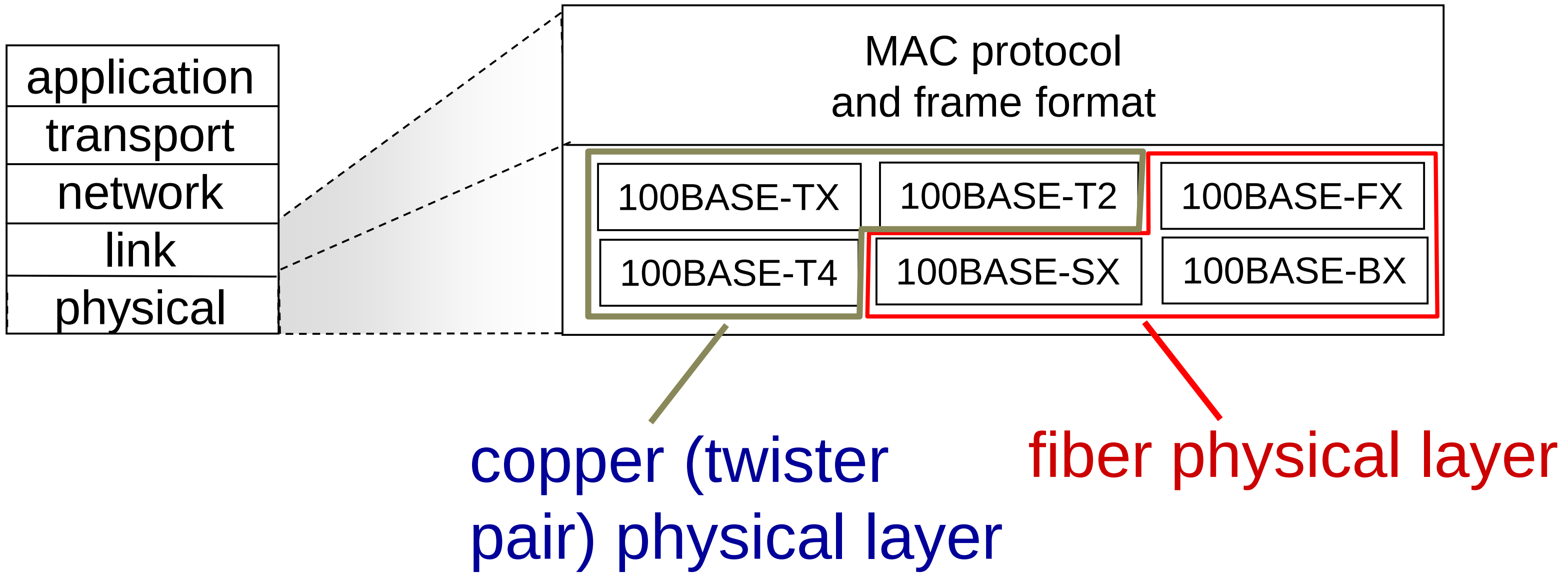
- **Addresses**
  - 6-byte source and destination MAC addresses
    - If adapter receives frame with matching destination address, or with broadcast address (e.g. ARP packet), it passes data in frame to network layer protocol
    - Otherwise, adapter discards frame
- **Type**
  - Indicates higher layer protocol (mostly IP but others possible, e.g., Novell IPX, AppleTalk)
- **CRC**
  - Cyclic redundancy check at receiver
    - If error is detected, frame is dropped

Ethernet: Unreliable and 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 reliable data transfer (e.g., TCP), otherwise dropped data is lost
- **Ethernet's MAC protocol is unslotted CSMA/CD with binary exponential backoff**
  - Ethernet networks that are fully switched don't really need collision detection

802.3 Ethernet Standards

- **There are many different Ethernet standards**
  - All have common MAC protocol and frame format at the Link Layer
  - Have different speeds
    - 2 Mbps, 10 Mbps, 100 Mbps, 1 Gbps, 10 Gbps
  - May have different Physical Layer media: coax, fiber, twisted pair
    - First number refers to the speed
    - Characters after the dash refer to the media type



Overview of Link Layer

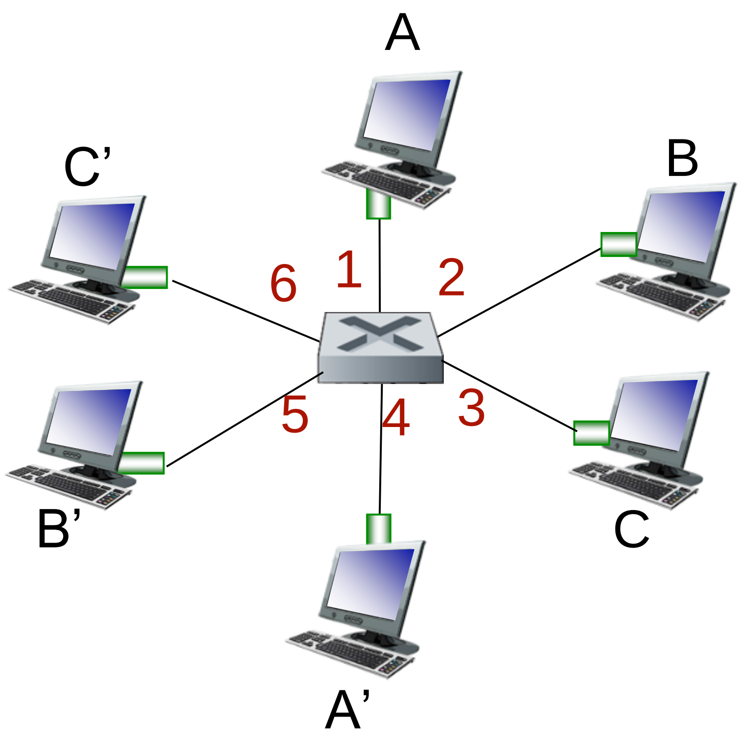
- **Introduction, Services**
- **Error Detection, Correction**
- **Multiple Access Protocols**
- **LANs**
  - Addressing, ARP
  - Ethernet
  - Switches
- **A Day in the Life**

Ethernet Switch

- **Link-layer device**
  - Store and forward Ethernet frames
  - Examine incoming frame’s MAC address, selectively forward frame to one-or-more outgoing links
- **Transparent**
  - Hosts are unaware of the presence of switches
  - No need to address a frame to a switch
- **Plug-and-play, self-learning**
  - Switches do not need to be configured

Switch: Multiple Simultaneous Transmissions

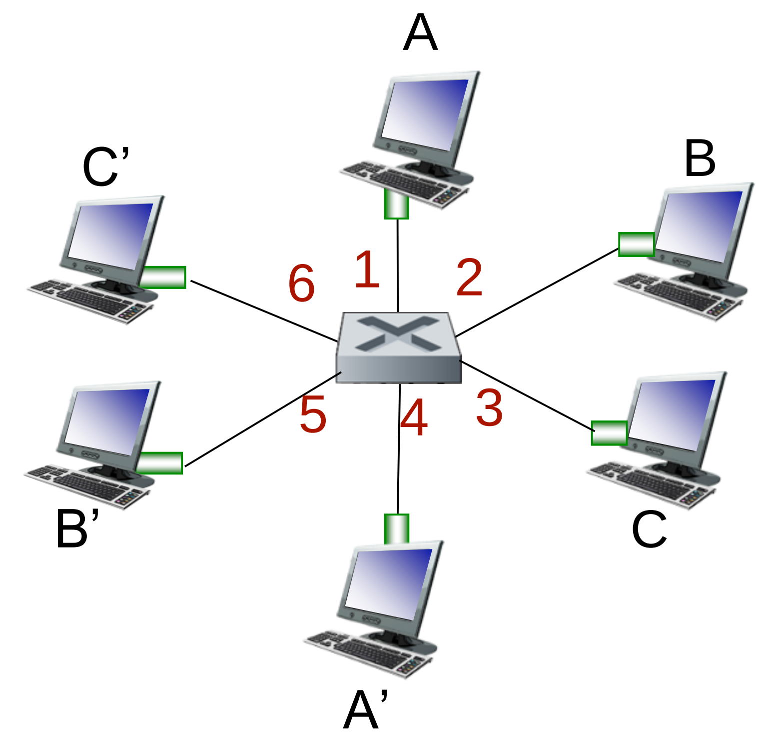
- **Hosts have dedicated, direct connection to switch**
- **Switches buffer frames**
- **Ethernet protocol used on each incoming link**
  - But no collisions
    - Each link is its own collision domain
  - Full duplex
    - Can send and receive simultaneously
- **Switching**
  - A-to-A' and B-to-B' can transmit simultaneously, without collisions



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

Switch Forwarding Table

- ***Question:*** **How does switch know A' is reachable via interface 4, or that B' reachable via interface 5?**
- ***Answer:*** **Each switch maintains a switch table, where each entry contains the following:**
  - MAC address of host, interface to reach that host, and a time stamp
    - Time stamp acts as a time-to-live

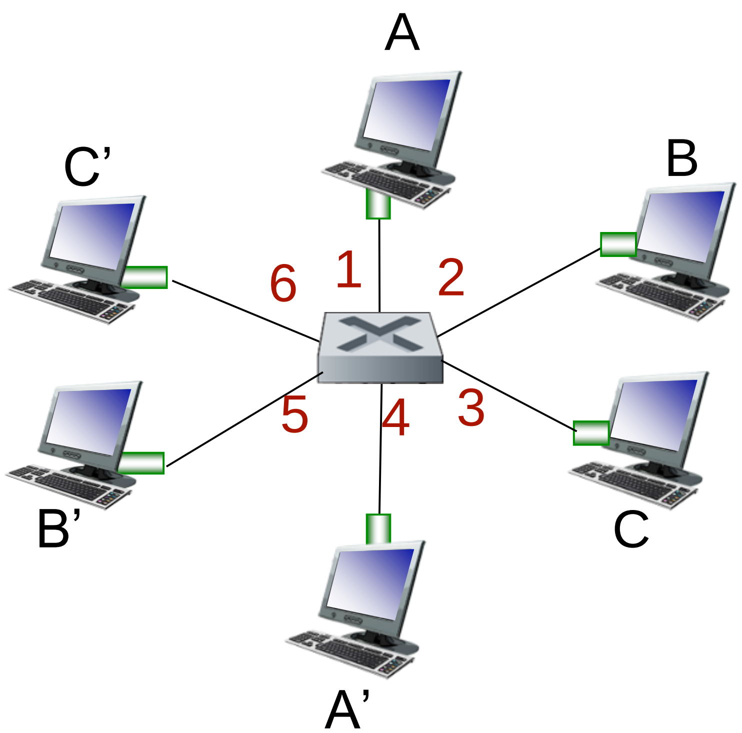


Switch with six interfaces (1,2,3,4,5,6)

Address	Interface	Time
01-12-23-34-45-56	2	9:39
62-FE-F7-11-89-A3	1	9:32
7C-BA-B2-B4-91-10	3	9:36

Switches are Self-learning

- **Switch table initially starts out empty**
- **Switch learns which hosts can be reached through which interfaces**
  - When a frame is received, switch "learns" location of sender
    - Knows which interface it came in on, looks at source address of incoming frame to construct (sender/interface) pair
- **Example, if Host A send a frame to Host C, switch learns that Host A is on Interface #1**



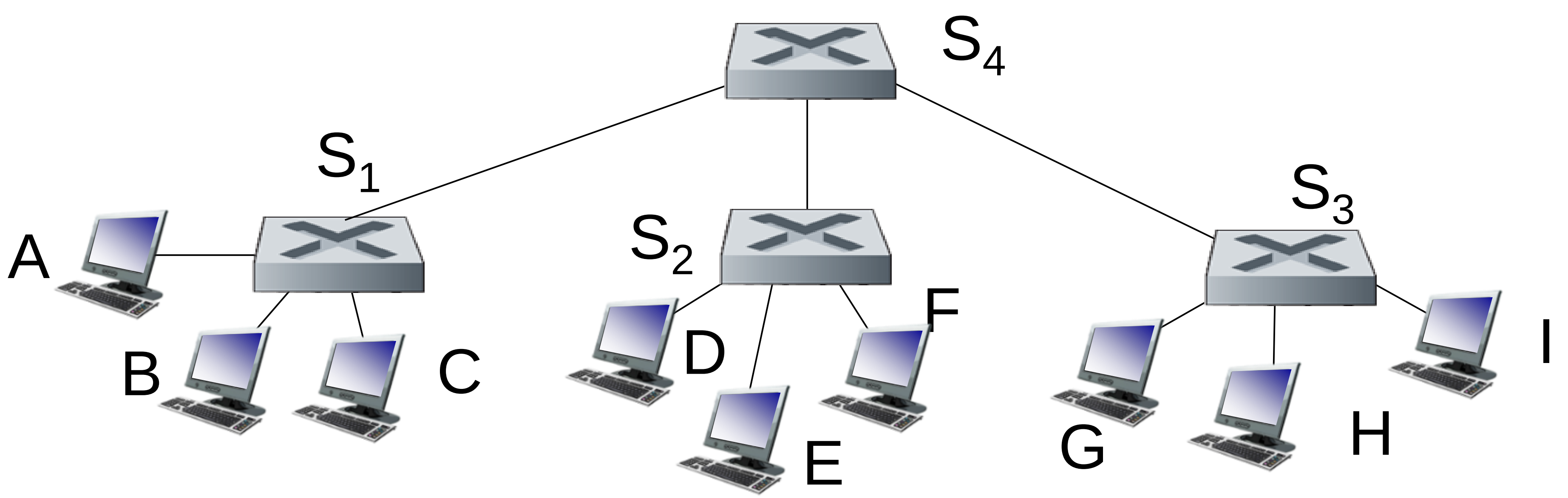
Switch with six interfaces (1,2,3,4,5,6)

Switch: Frame Filtering & Forwarding

- **When a frame arrives at a switch, there are three possible scenarios**
  - There is no entry in the switch table for the destination host
    - Switch forwards copies of the frame on all outgoing interfaces *except* the one on which it arrived (i.e. broadcast)
  - There is an entry in the switch table for the destination host, but the destination host is on the same interface on which the frame arrived
    - Switch simply discards the frame
  - There is an entry in the switch table for the destination host and it is on a different interface than the interface on which the frame arrived
    - Switch forwards the frame only on the interface on which the destination host is connected

Interconnecting switches

self-learning switches can be connected together:

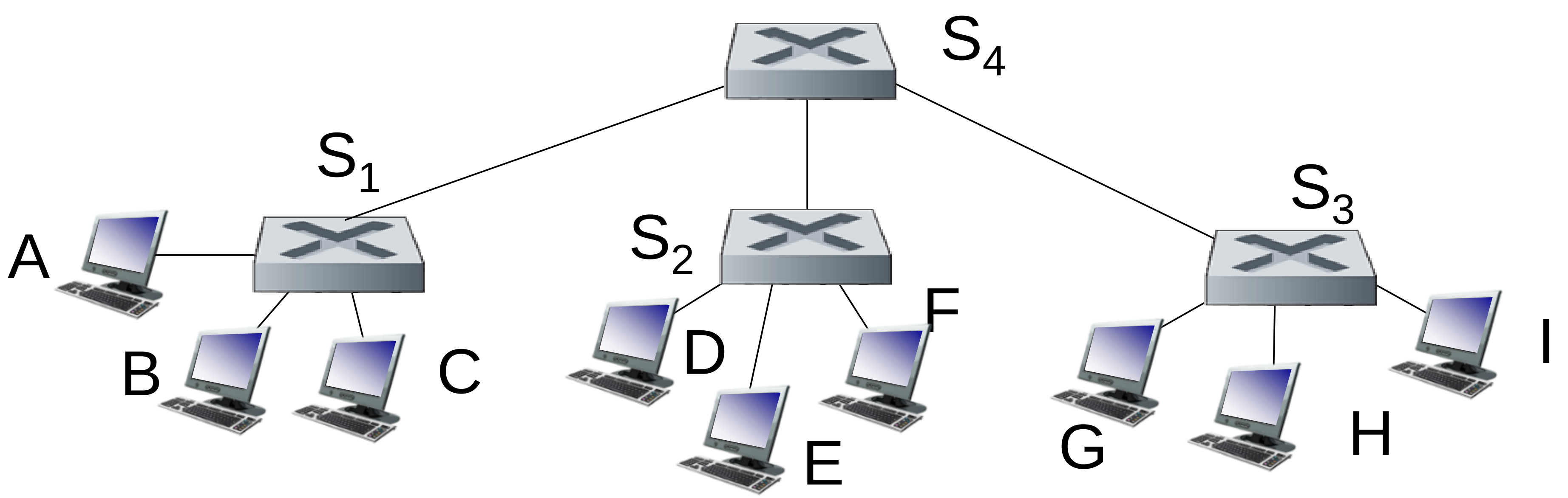


Q: sending from A to G - how does S_1 know to forward frame destined to G via S_4 and S_3?

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

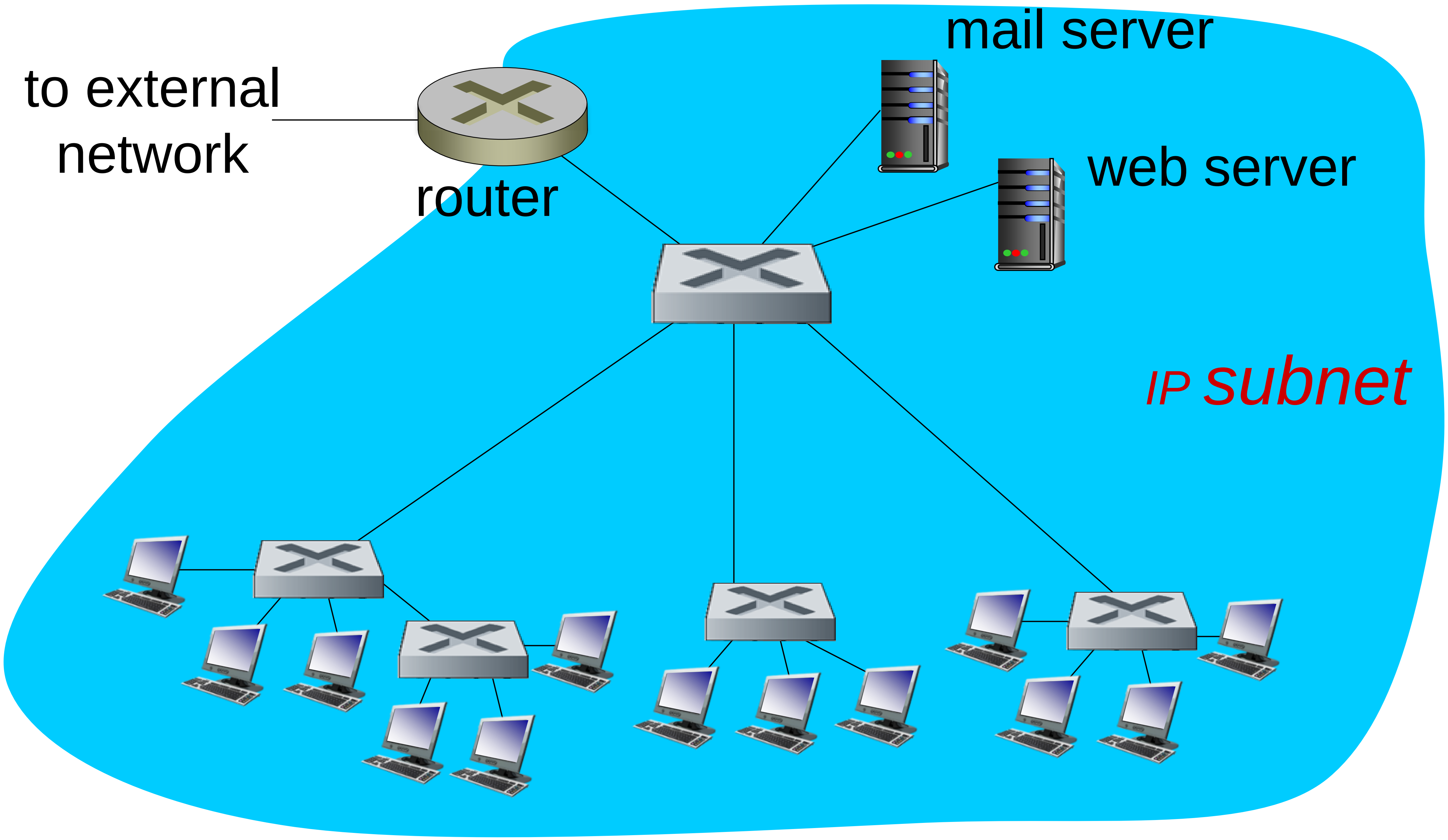
Self-learning multi-switch example

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

Institutional network



Switches vs. Routers

- **Both are store-and-forward devices**
  - Routers are network-layer devices (examine network-layer headers)
  - Switches are link-layer devices (examine link-layer headers)

- **Both have forwarding tables**
  - Routers compute tables using routing algorithms and IP addresses
  - Switches learn forwarding table using flooding, learning, MAC addresses

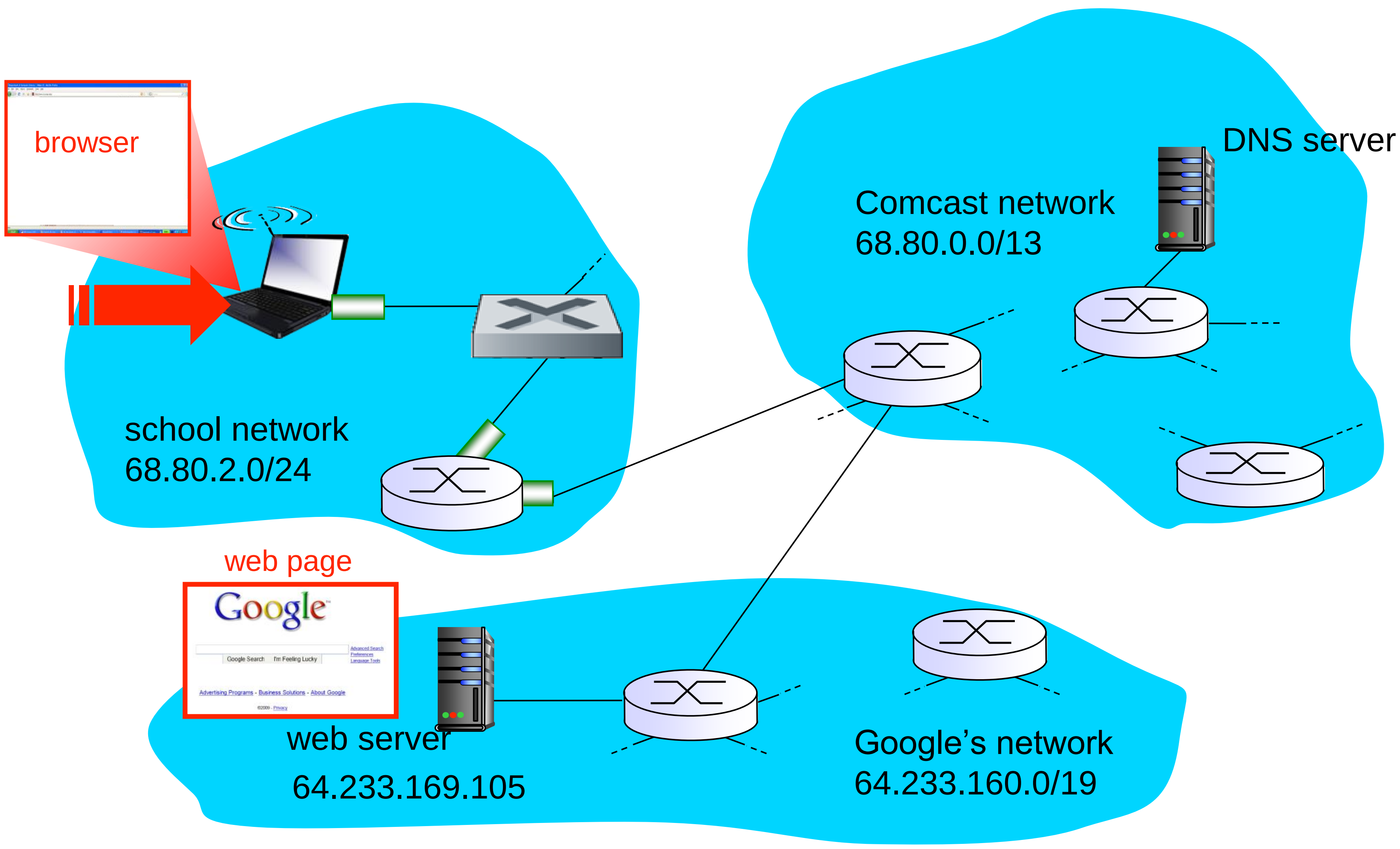
Overview of Link Layer

- **Introduction, Services**
- **Error Detection, Correction**
- **Multiple Access Protocols**
- **LANs**
  - Addressing, ARP
  - Ethernet
  - Switches
- **A Day in the Life**

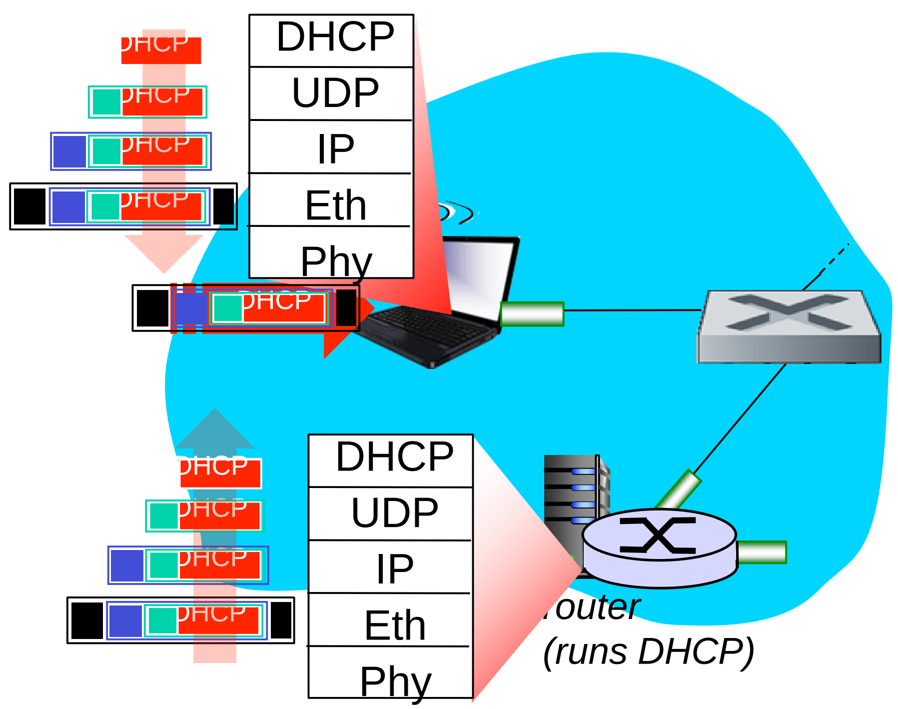
Synthesis: a day in the life of a web request

- **journey down protocol stack complete!**
  - application, transport, network, link
- **putting-it-all-together: synthesis!**
  - *goal:* identify, review, understand protocols (at all layers) involved in seemingly simple scenario: requesting www page
  - *scenario:* student attaches laptop to campus network, requests/receives www.google.com

A Day in the Life of a Web Request: Scenario

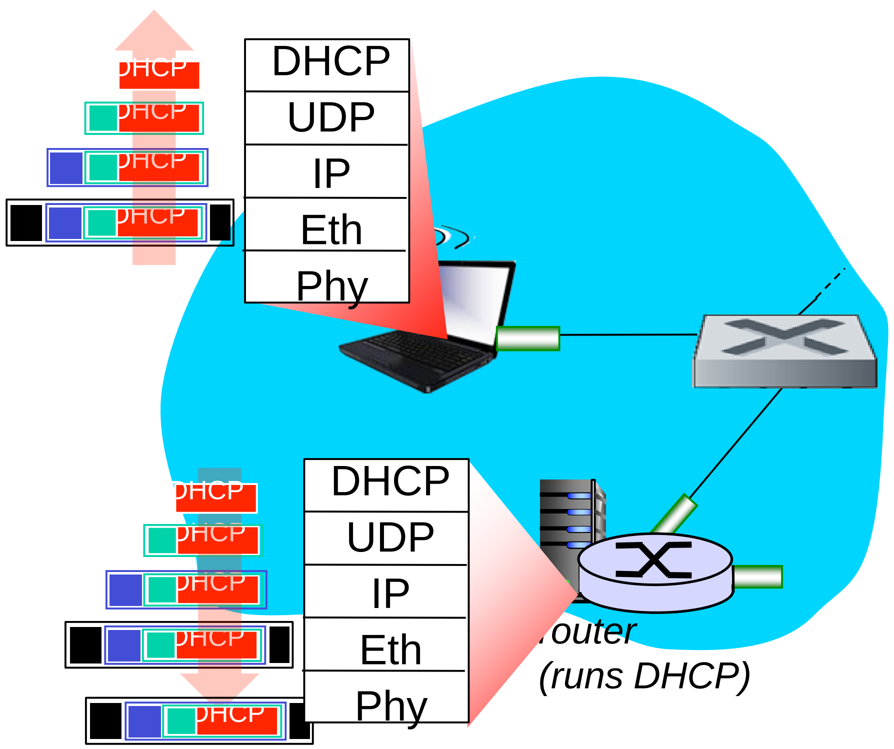


A Day in the Life… Connecting to the Internet



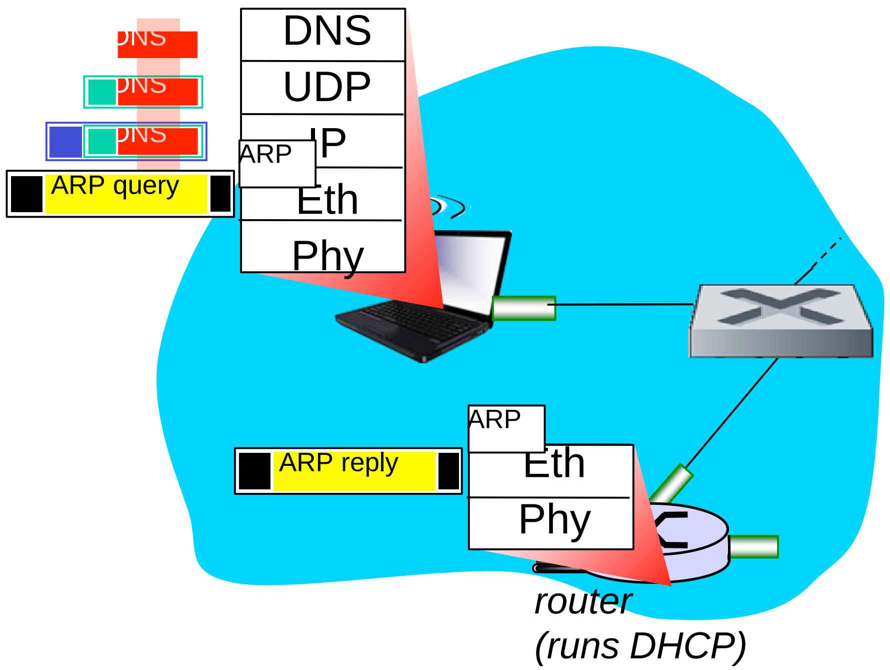
- **Connecting laptop needs to get its own IP address, the address of the first-hop router, and address of DNS server -- use DHCP**
- **DHCP request *encapsulated* in UDP, encapsulated in IP, encapsulated in 802.3 Ethernet**
- **Ethernet frame broadcast on LAN, received at router running DHCP server**
  - Dest MAC addr = FF-FF-FF-FF-FF-FF
  - Dest IP addr = 255.255.255.255:67
- **Ethernet demuxed to IP demuxed, UDP demuxed to DHCP**

A Day in the Life… Connecting to the Internet



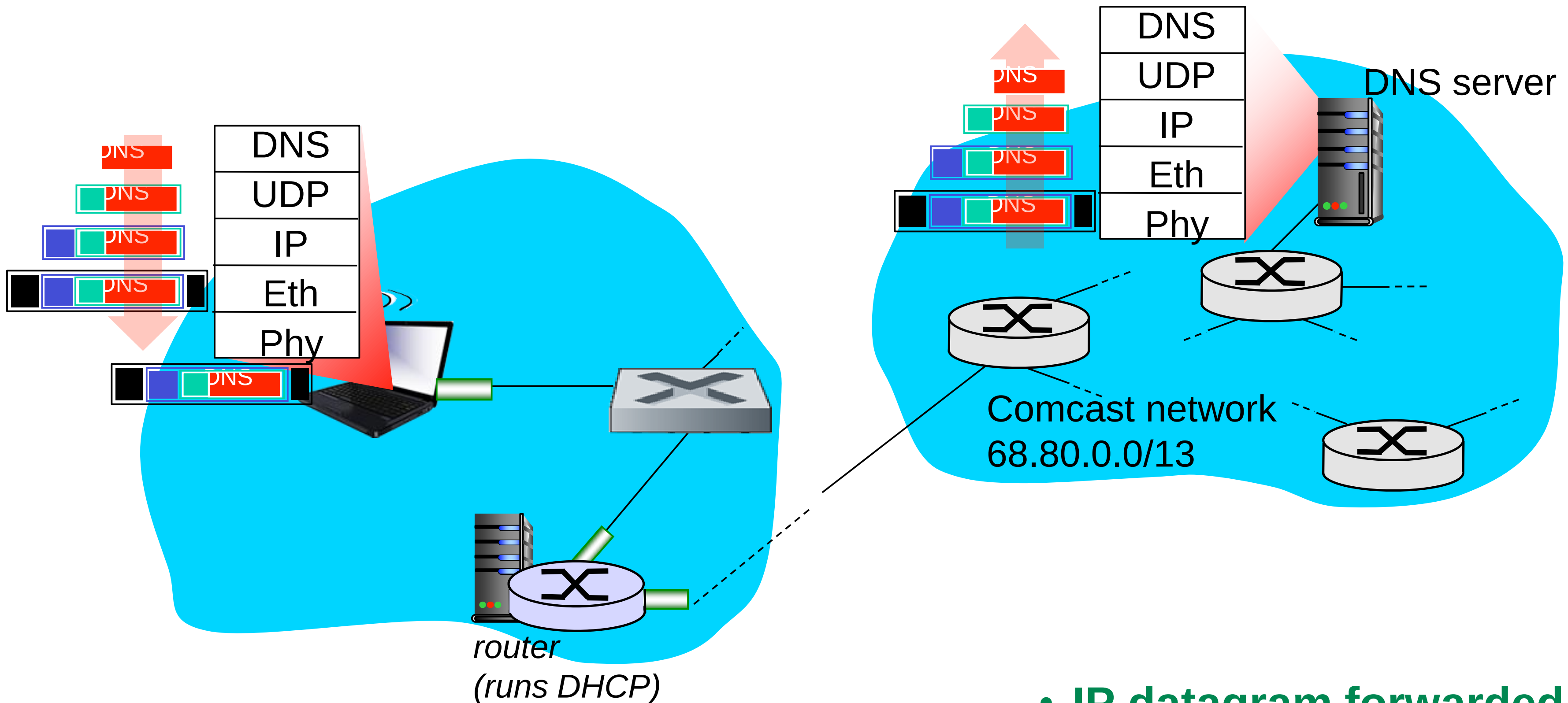
- DHCP server formulates **DHCP ACK** containing client's IP address, IP address of first-hop router for client, and name and IP address of DNS server
- Encapsulation at DHCP server, frame forwarded (switch now knows MAC address of laptop learning) through LAN, demultiplexing at client
- DHCP client receives DHCP ACK reply
- Client now has IP address, knows the name and address of the DNS server, and the IP address of its first-hop router

A Day in the Life… ARP



- Before sending HTTP request, need IP address of www.google.com: DNS
- DNS query created, encapsulated in UDP, encapsulated in IP, encapsulated in Ethernet frame
- To send frame to router, need MAC address of router interface: ARP
- ARP query broadcast, received by router, which replies with ARP reply giving MAC address of router interface to laptop
- Client now knows MAC address of first hop router, so can now send frame containing DNS query

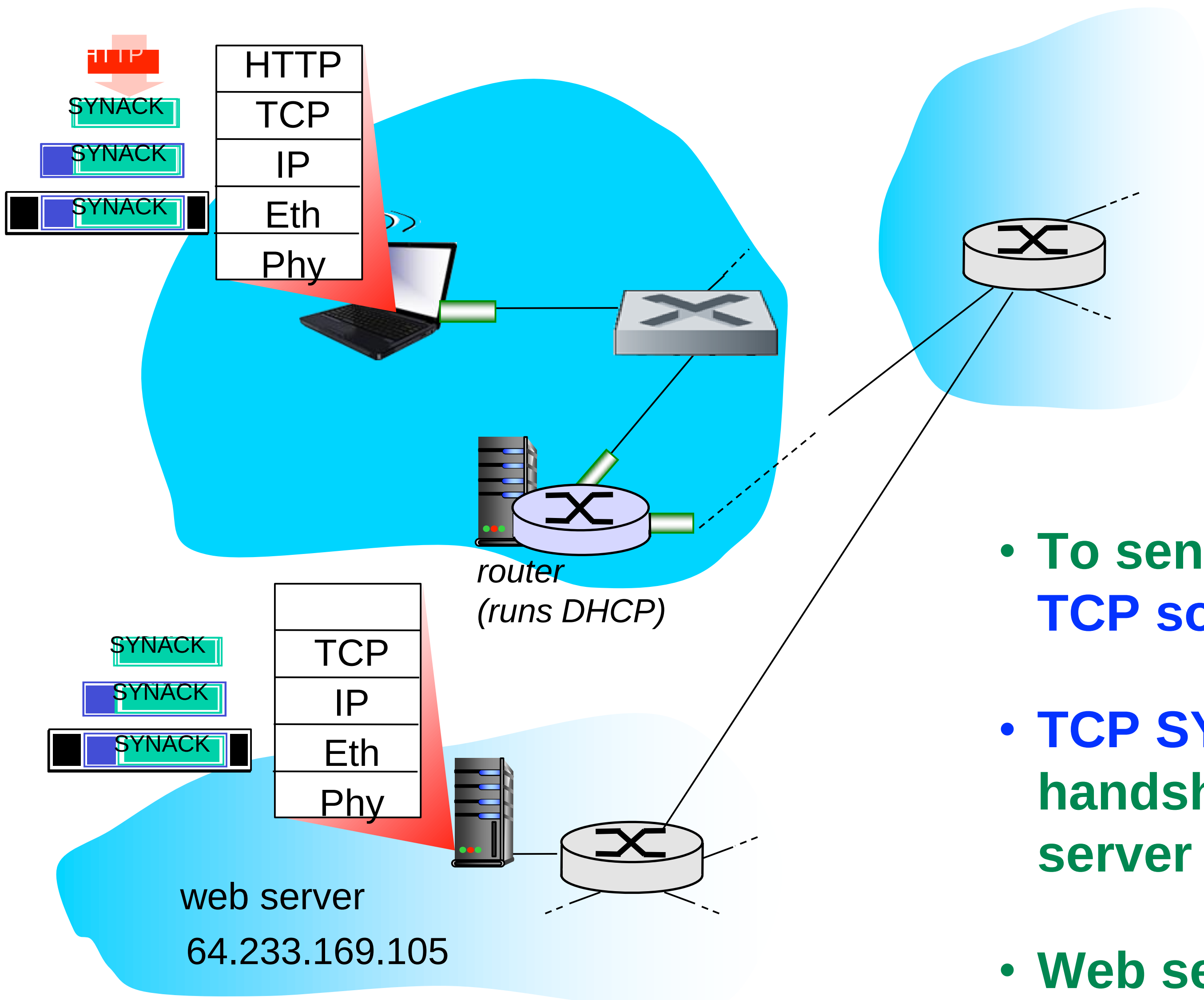
A Day in the Life… Using DNS



- **IP datagram containing DNS query forwarded via LAN switch from client to 1st hop router**

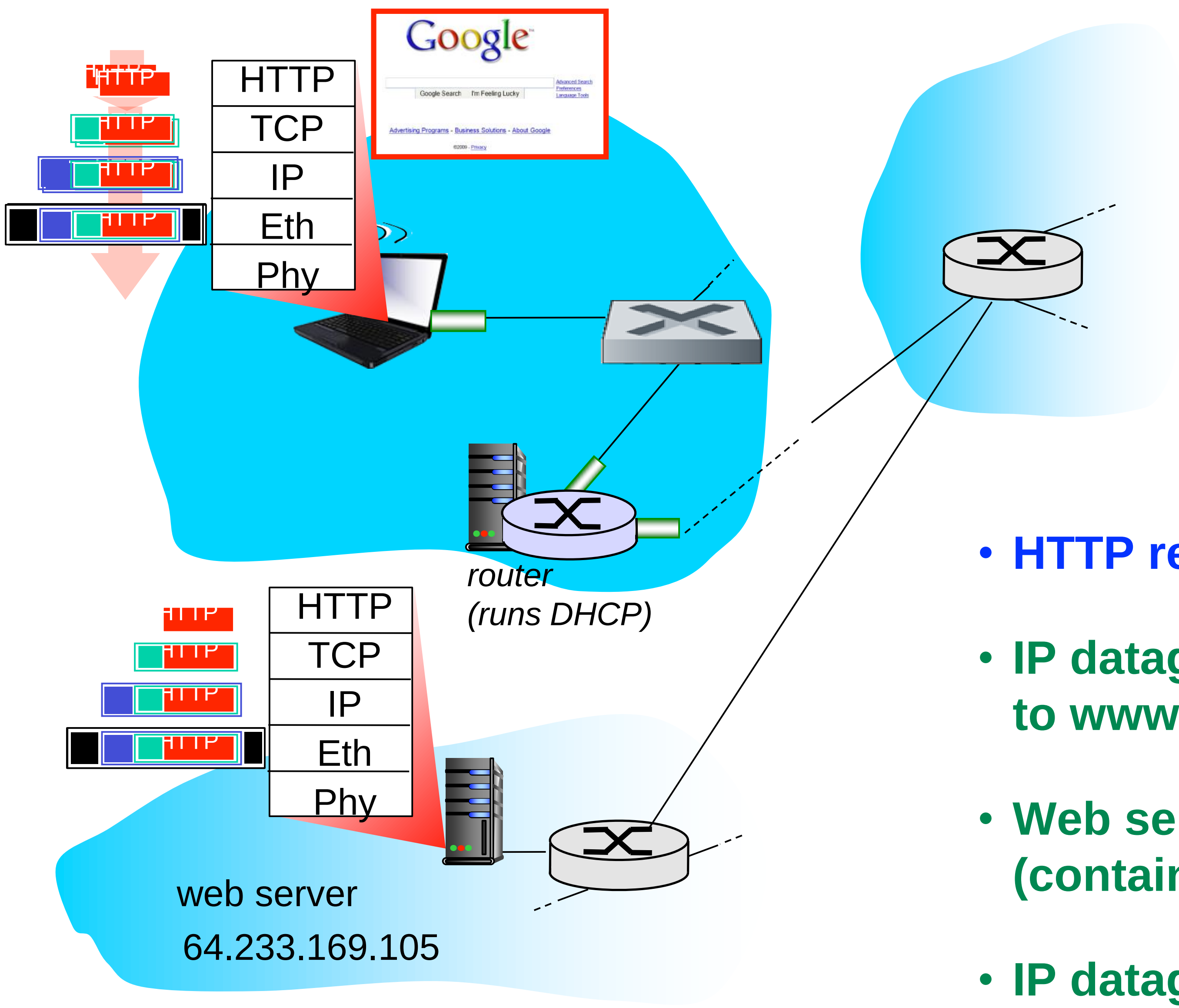
- **IP datagram forwarded from campus network into Comcast network,**
  - Routed to DNS server (tables created by RIP, OSPF, IS-IS and/or BGP routing protocols)
- **Demux'ed to DNS server**
- **DNS server replies to client with IP address of www.google.com**

A Day in the Life…TCP Connection Carrying HTTP



- To send **HTTP request**, client first opens **TCP socket to web server**
- **TCP SYN segment** (step 1 in 3-way handshake) inter-domain routed to web server
- Web server responds with **TCP SYNACK** (step 2 in 3-way handshake)
- **TCP connection established!**

A Day in the Life… HTTP Request/Reply



- **HTTP request sent to TCP socket**
- **IP datagram containing HTTP request routed to www.google.com**
- **Web server responds with HTTP reply (containing web page)**
- **IP datagram containing HTTP reply routed back to client**
- **Web page finally displayed!**

Chapter 6: Summary

- **principles behind data link layer services:**
  - error detection, correction
  - sharing a broadcast channel: multiple access
  - link layer addressing
- **instantiation and implementation of various link layer technologies**
  - Ethernet
  - switched LANS
- **synthesis: a day in the life of a web request**