Data Link and Physical Layers and 10 GbE Protocol

Hakim Weatherspoon

Assistant Professor, Dept of Computer Science

CS 5413: High Performance Systems and Networking

September 12, 2014

Slides used and adapted judiciously from Computer Networking, A Top-Down Approach

Goals for Today

- Link Layer and Physical Layer
  - Abstraction / services
  - Switches and Local Area Networks
    - Addressing, ARP (address resolution protocol)
    - Ethernet
    - Ethernet Switch
  - Multiple Access Protocols

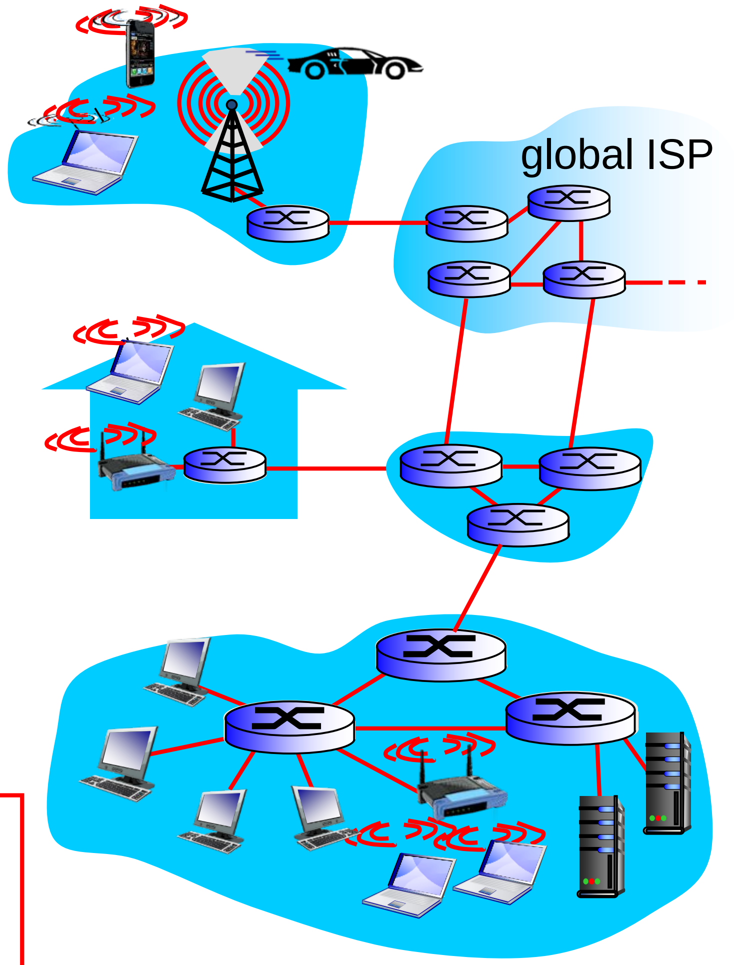
- Data Center Network
  - 10GbE (10 Gigabit Ethernet)

- Backup Slides
  - Virtual Local Area Networks (VLAN)
  - Multiple Access Protocols
  - Putting it all together: A day and a life of a web request

Link Layer

terminology:

- hosts and routers: nodes
- communication channels that connect adjacent nodes along communication path: links
  - wired links
  - wireless links
  - LANs
- layer-2 packet: frame, encapsulates datagram



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

Link Layer

- 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 rdt 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

Link Layer

Services

- *framing, link access:*
  - encapsulate datagram into frame, adding header, trailer
  - channel access if shared medium
  - "MAC" addresses used in frame headers to identify source, dest
    - different from IP address!
- *reliable delivery between adjacent nodes*
  - we learned how to do this already (chapter 3)!
  - seldom used on low bit-error link (fiber, some twisted pair)
  - wireless links: high error rates
    - *Q:* why both link-level and end-end reliability?

Link Layer

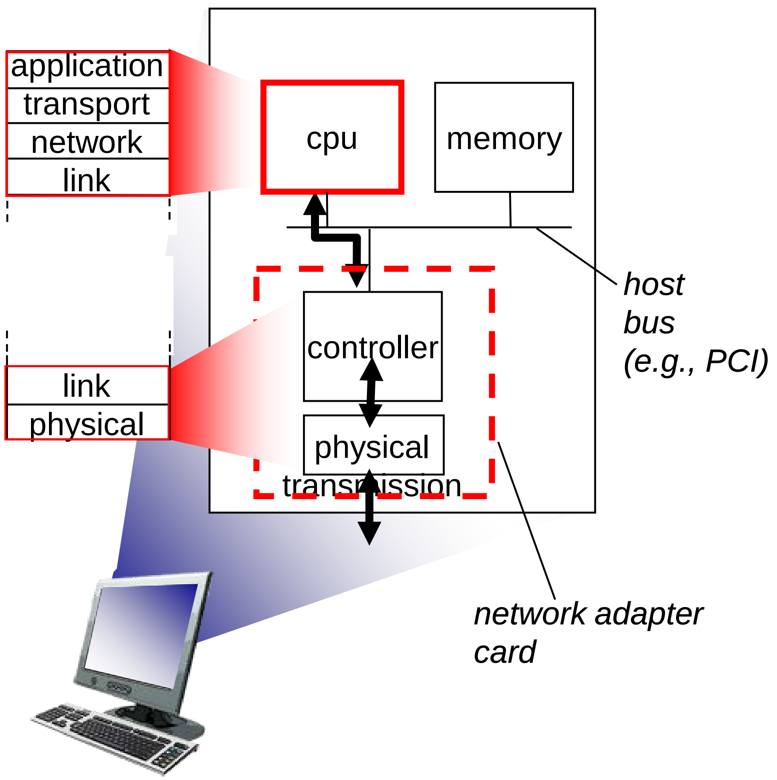
Services

- ***flow control:***
  - pacing between adjacent sending and receiving nodes
- ***error detection*:**
  - errors caused by signal attenuation, noise.
  - receiver detects presence of errors:
    - signals sender for retransmission or drops frame
- ***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

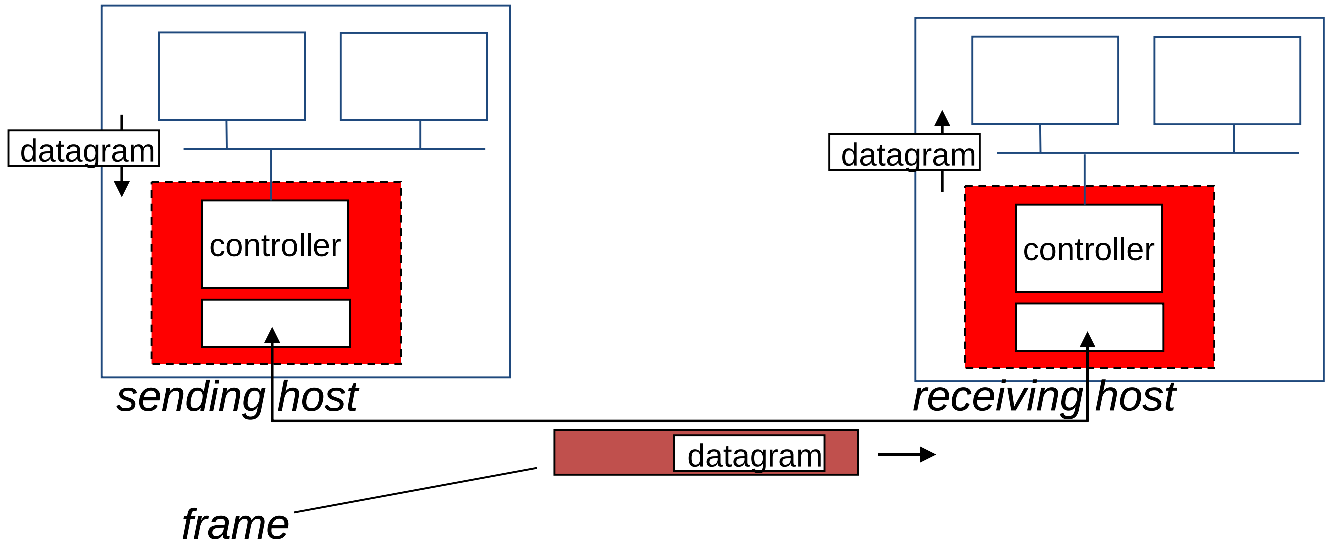
Link Layer

Where is the link layer implemented?

- in each and every host
- link layer implemented in "adaptor" (aka *network interface card* NIC) or on a chip
  - Ethernet card, 802.11 card; Ethernet chipset
  - implements link, physical layer
- attaches into host's system buses
- combination of hardware, software, firmware



Adapters communicating



- sending side:
  - encapsulates datagram in frame
  - adds error checking bits, rdt, flow control, etc.
- receiving side
  - looks for errors, rdt, flow control, etc
  - extracts datagram, passes to upper layer at receiving side

Goals for Today

- Link Layer and Physical Layer
  - Abstraction / services
  - Switches and Local Area Networks
    - Addressing, ARP (address resolution protocol)
    - Ethernet
    - Ethernet switch
  - Multiple Access Protocols

- Data Center Network
  - 10GbE (10 Gigabit Ethernet)

- Backup Slides
  - Virtual Local Area Networks (VLAN)
  - Multiple Access Protocols
  - Putting it all together: A day and a life of a web request

Switches and Local Area Networks

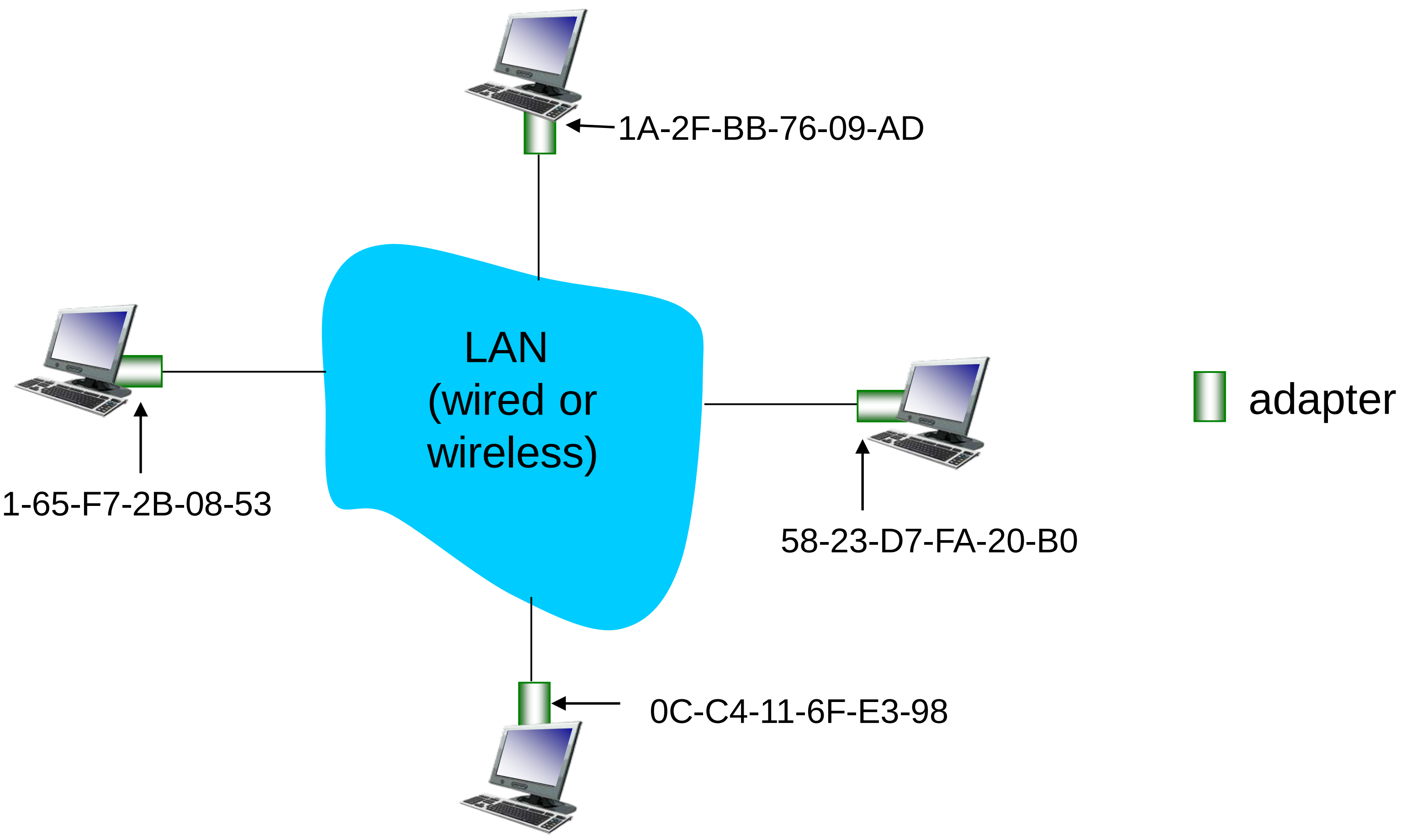
MAC (medium access control) addresses and ARP (address resolution protocol)

- 32-bit IP address:
  - *network-layer* address for interface
  - used for layer 3 (network layer) forwarding
- MAC (or LAN or physical or Ethernet) address:
  - function: *used 'locally" to get frame from one interface to another physically-connected interface (same network, in IP-addressing sense)*
  - 48 bit MAC address (for most LANs) burned in NIC ROM, also sometimes software settable
  - e.g.: 1A-2F-BB-76-09-AD

hexadecimal (base 16) notation
(each "number" represents 4 bits)

LAN (MAC) addresses and ARP

each adapter on LAN has unique *LAN* address



Switches and Local Area Networks

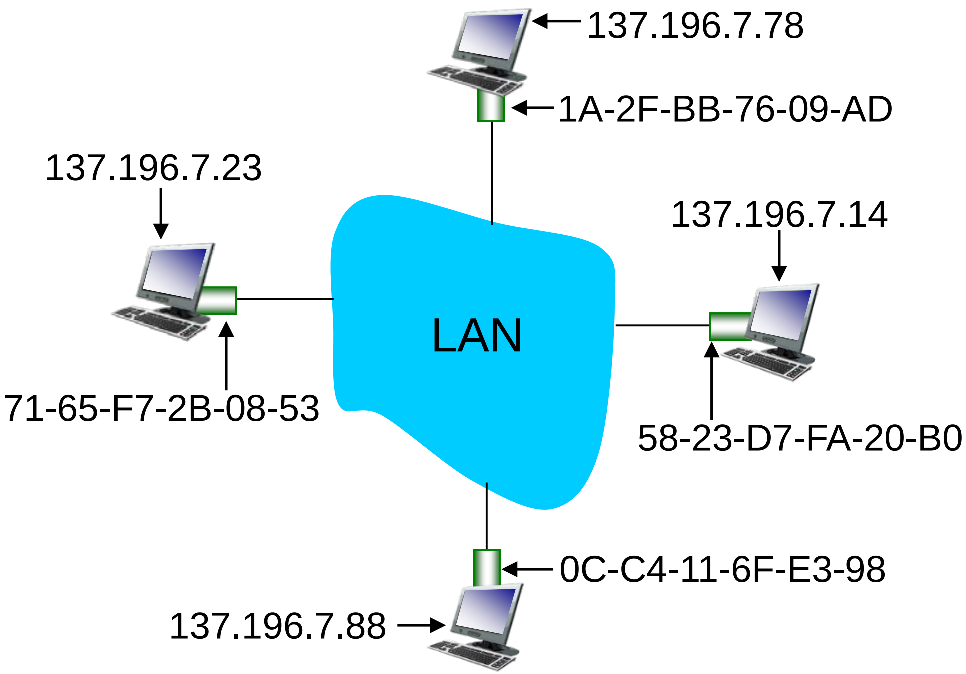
LAN (MAC) addresses and ARP

- 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

Switches and Local Area Networks

LAN (MAC) addresses and ARP: Address Resolution Protocol

Question: how to determine interface's MAC address, knowing its IP address?



ARP table: each IP node (host, router) on LAN has 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)

Switches and Local Area Networks

ARP: Address Resolution Protocol; Same LAN

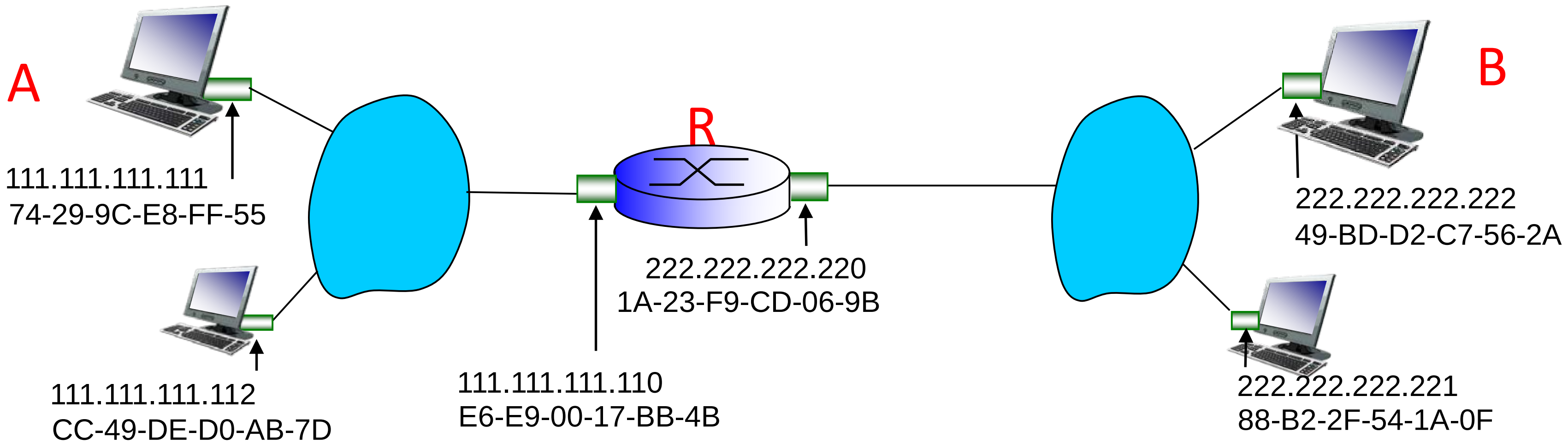
- A wants to send datagram to B
  - B's MAC address not in A's ARP table.
- A broadcasts ARP query packet, containing B's IP address
  - dest MAC address = FF-FF-FF-FF-FF-FF
  - all nodes on LAN receive ARP query
- B receives ARP packet, replies to A with its (B's) MAC address
  - frame sent to A's MAC 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*

Switches and Local Area Networks

ARP: Address Resolution Protocol; different LAN

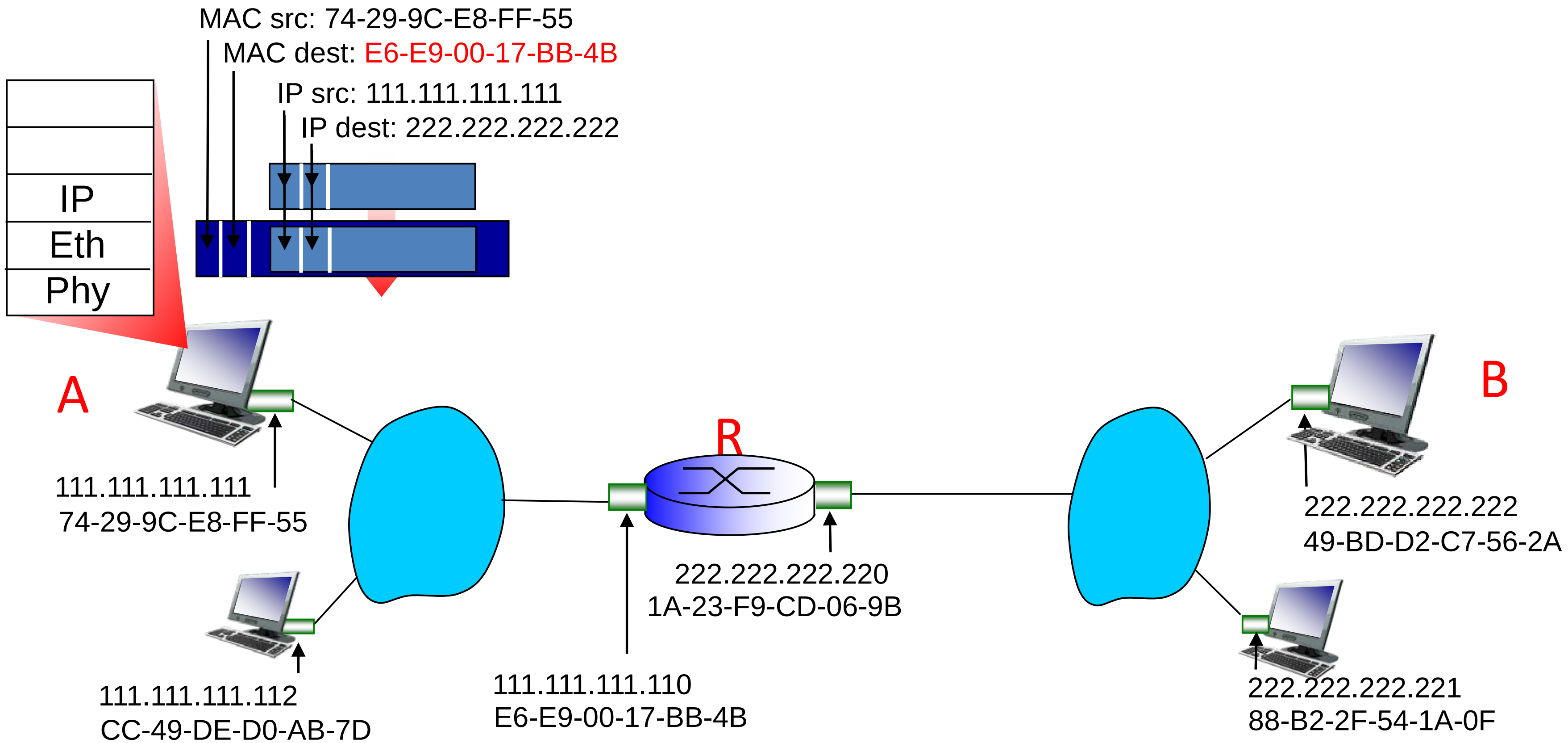
walkthrough: send datagram from A to B via R

- focus on addressing – at IP (datagram) and MAC layer (frame)
- assume A knows B's IP address
- assume A knows IP address of first hop router, R (how?)
- assume A knows R's MAC address (how?)



ARP: Address Resolution Protocol; different LAN

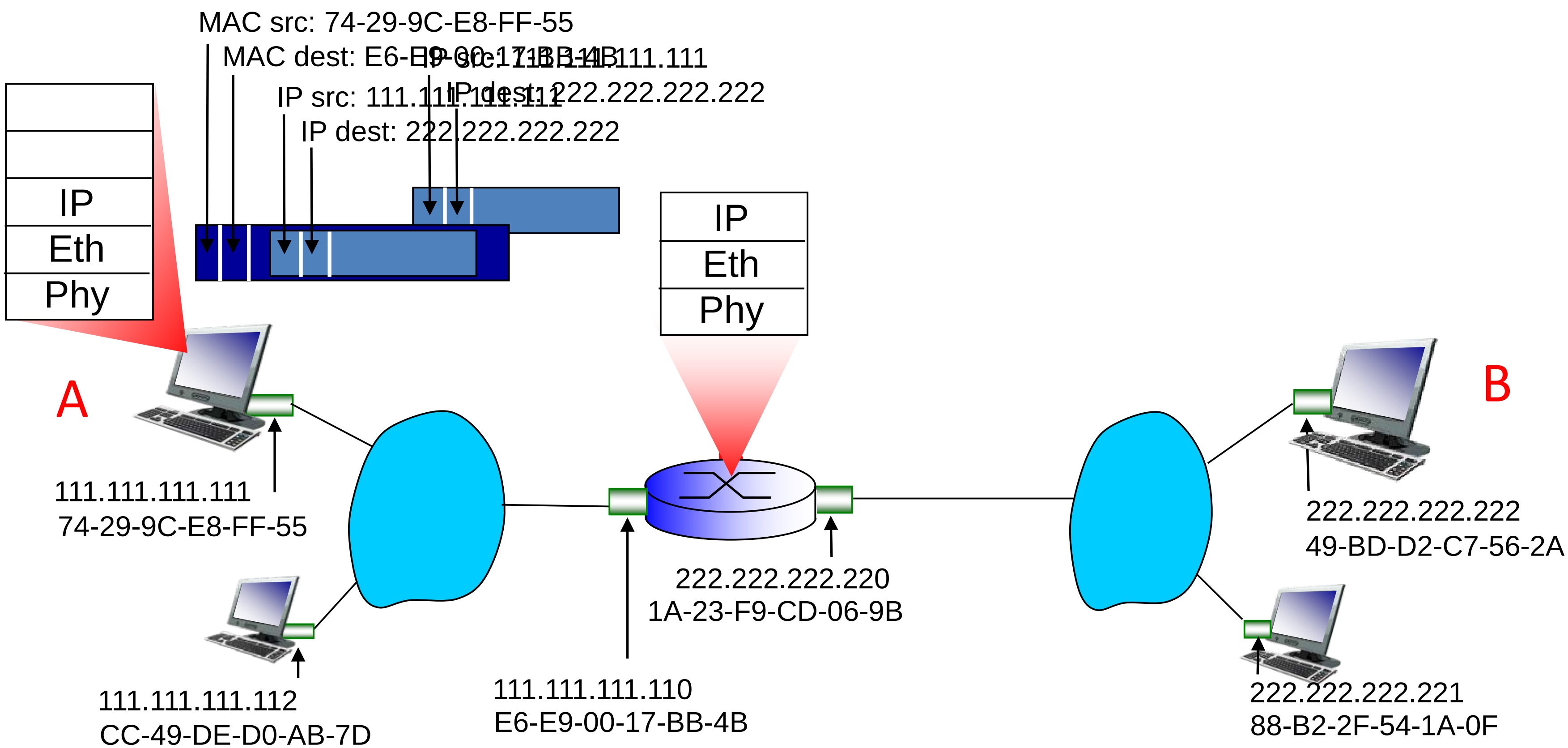
- A creates IP datagram with IP source A, destination B
- A creates link-layer frame with R's MAC address as dest, frame contains A-to-B IP datagram



Switches and Local Area Networks

ARP: Address Resolution Protocol; different LAN

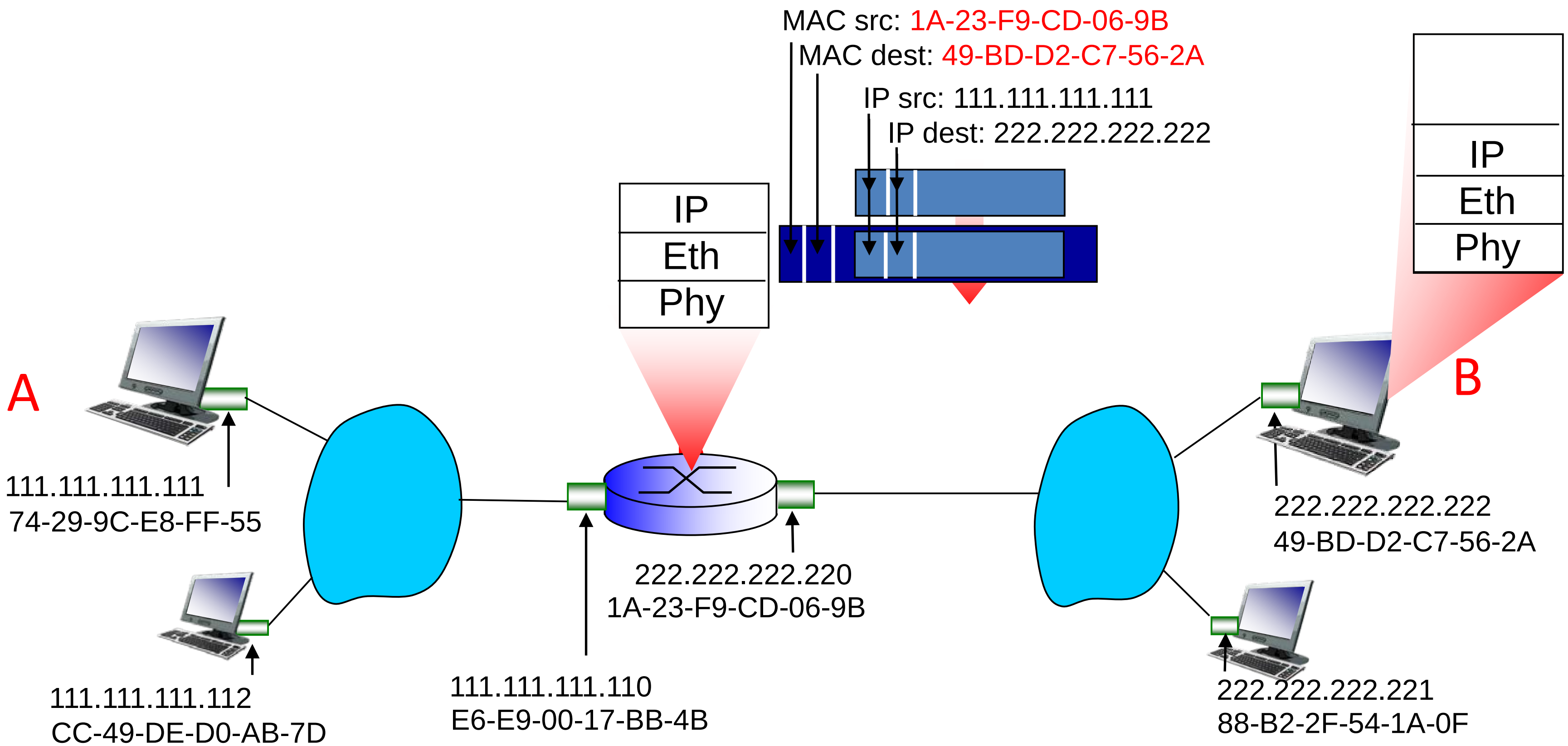
- frame sent from A to R
- frame received at R, datagram removed, passed up to IP



Switches and Local Area Networks

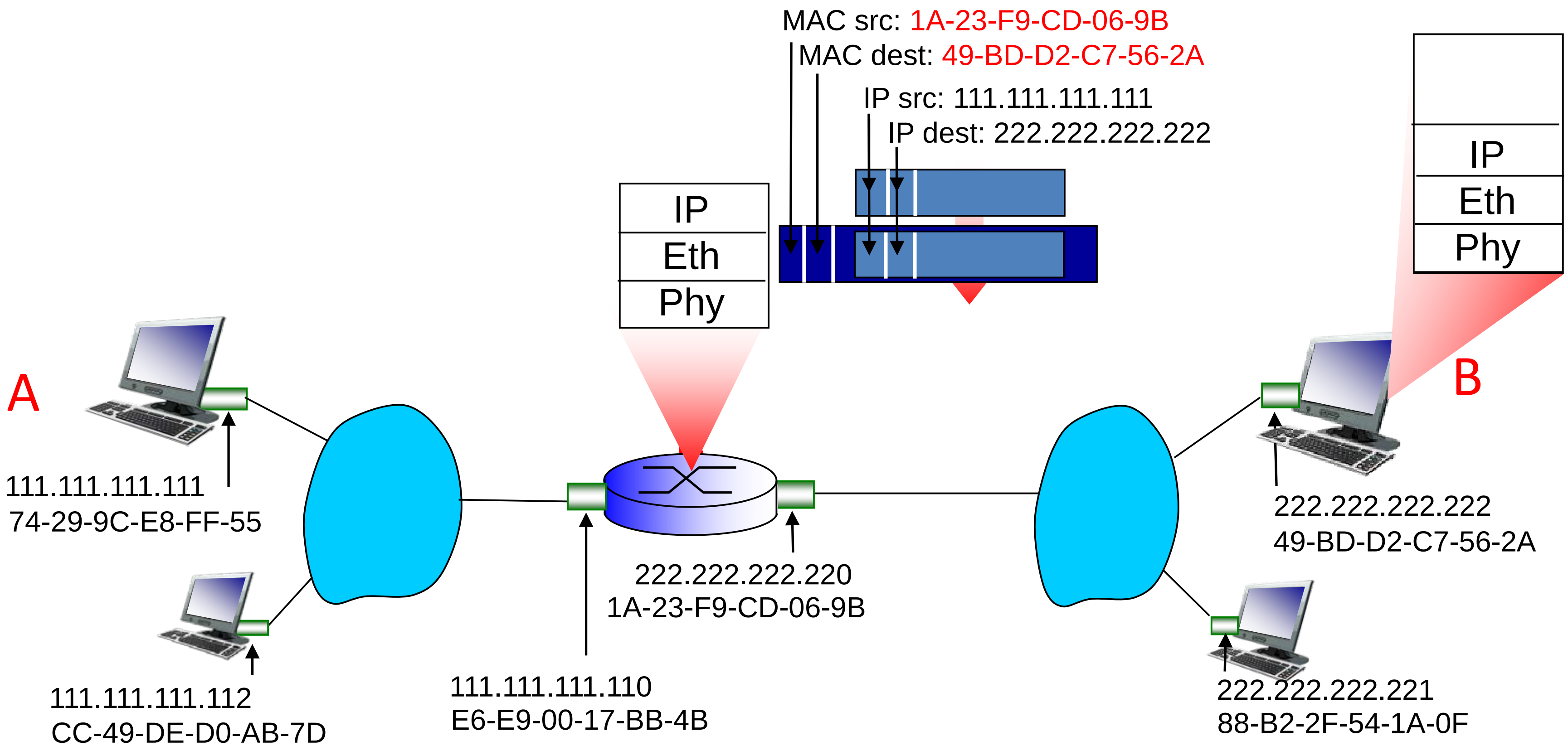
ARP: Address Resolution Protocol; different LAN

- R forwards datagram with IP source A, destination B
- R creates link-layer frame with B's MAC address as dest, frame contains A-to-B IP datagram



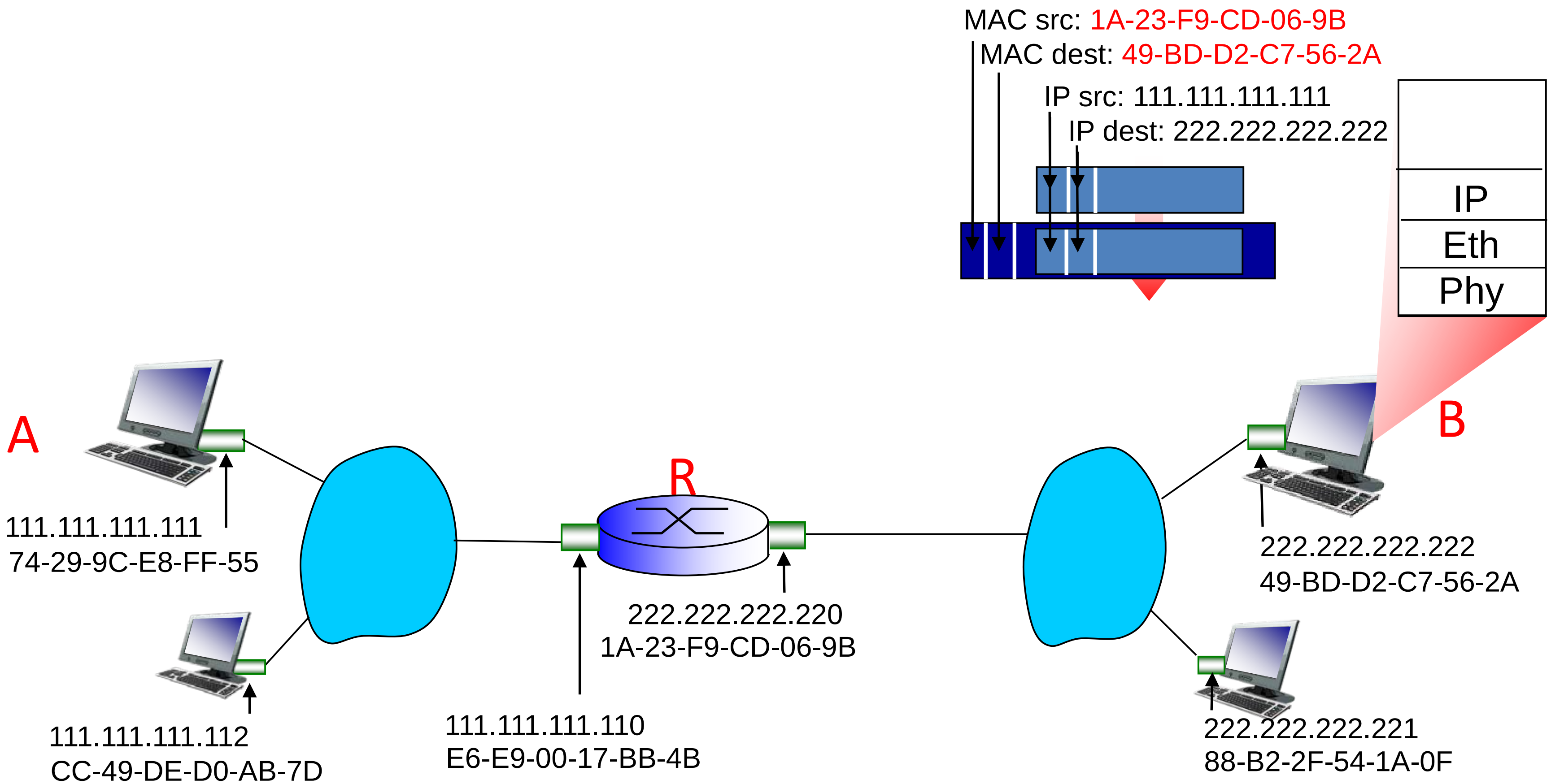
ARP: Address Resolution Protocol; different LAN

- R forwards datagram with IP source A, destination B
- R creates link-layer frame with B's MAC address as dest, frame contains A-to-B IP datagram



ARP: Address Resolution Protocol; different LAN

- R forwards datagram with IP source A, destination B
- R creates link-layer frame with B's MAC address as dest, frame contains A-to-B IP datagram



Goals for Today

- Link Layer and Physical Layer
  - Abstraction / services
  - Switches and Local Area Networks
    - Addressing, ARP (address resolution protocol)
    - Ethernet
    - Ethernet switch
  - Multiple Access Protocols

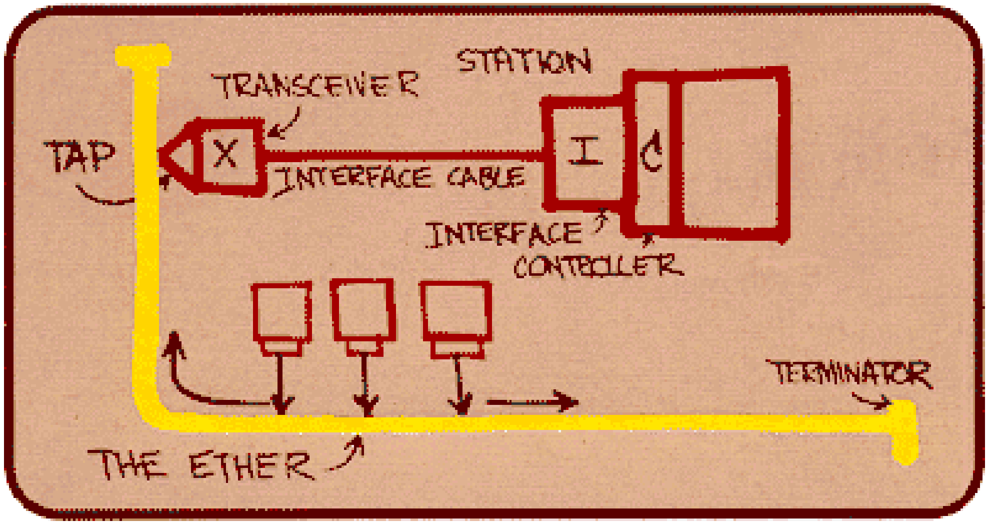
- Data Center Network
  - 10GbE (10 Gigabit Ethernet)

- Backup Slides
  - Virtual Local Area Networks (VLAN)
  - Multiple Access Protocols
  - Putting it all together: A day and a life of a web request

Link and Physical Layer: Ethernet

"dominant" wired LAN technology:

- cheap $20 for NIC
- first widely used LAN technology
- simpler, cheaper than token LANs and ATM
- kept up with speed race: 10 Mbps – 10 Gbps

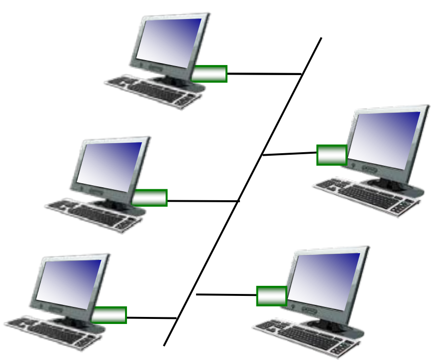


Metcalfe's Ethernet sketch

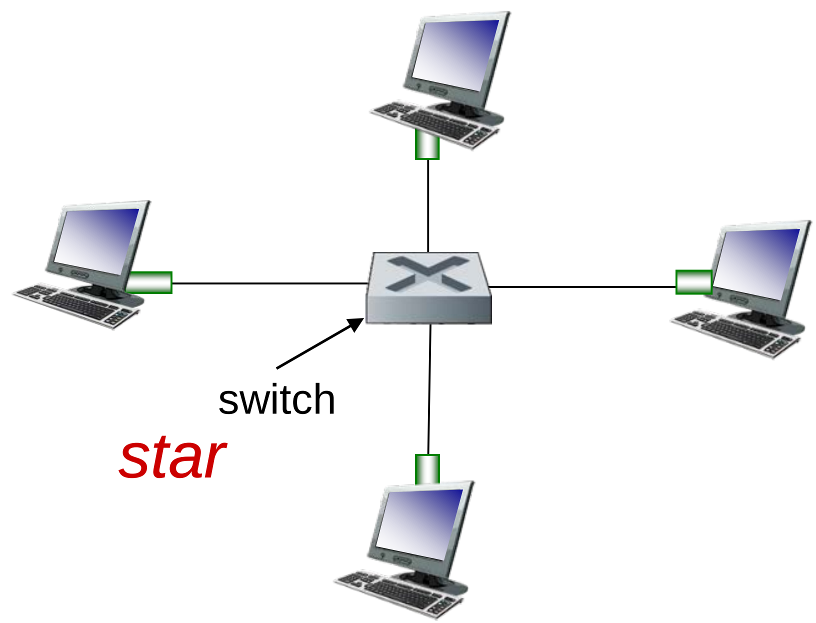
Link and Physical Layer: Ethernet

Ethernet Physical Topologies

- *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

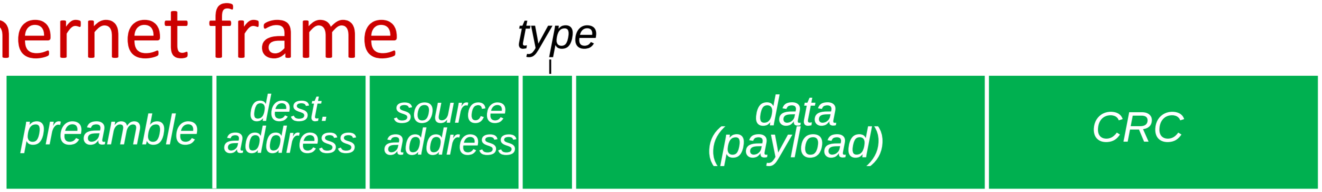


star

Link and Physical Layer: Ethernet

Ethernet frame structure/format

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



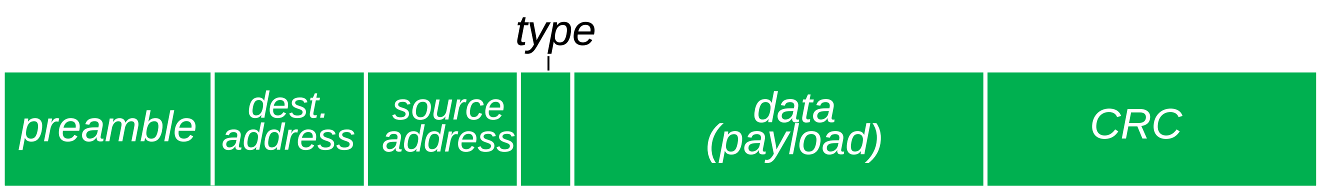
preamble:

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

Link and Physical Layer: Ethernet

Ethernet frame structure/format

- *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



Link and Physical Layer: Ethernet

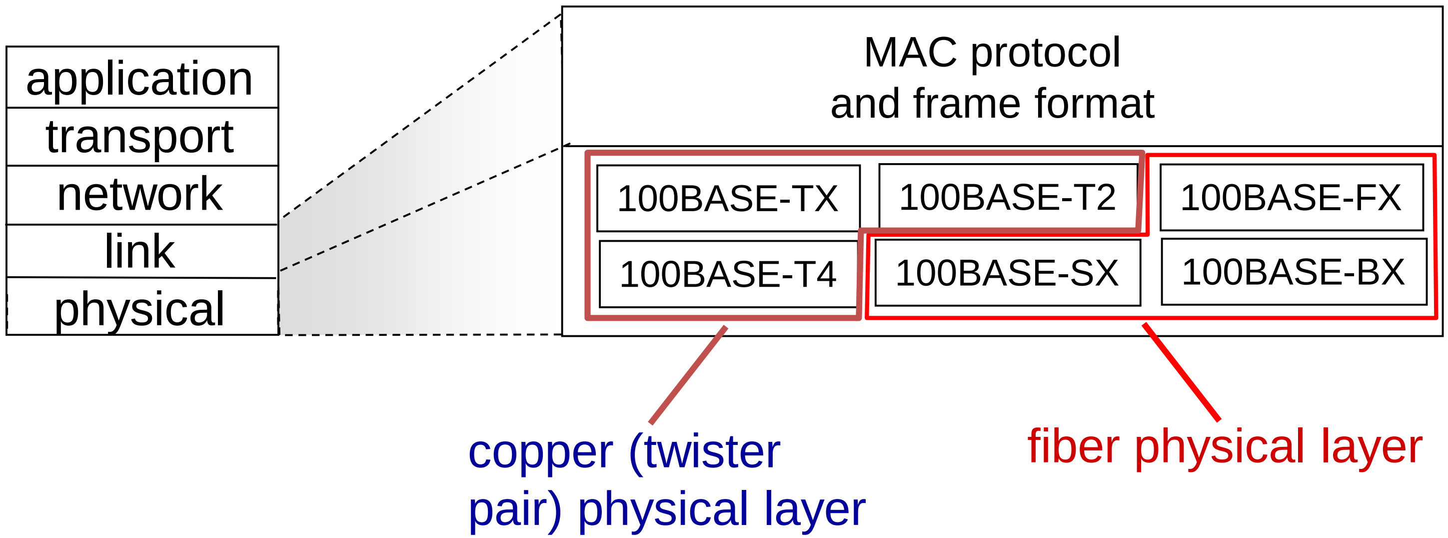
Ethernet service abstraction and implementation

- *connectionless:* no handshaking between sending and receiving NICs
- *unreliable:* receiving NIC doesnt 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 wth binary backoff*

Link and Physical Layer: Ethernet

IEEE 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



Goals for Today

- Link Layer and Physical Layer
  - Abstraction / services
  - Switches and Local Area Networks
    - Addressing, ARP (address resolution protocol)
    - Ethernet
    - Ethernet switch
  - Multiple Access Protocols

- Data Center Network
  - 10GbE (10 Gigabit Ethernet)

- Backup Slides
  - Virtual Local Area Networks (VLAN)
  - Multiple Access Protocols
  - Putting it all together: A day and a life of a web request

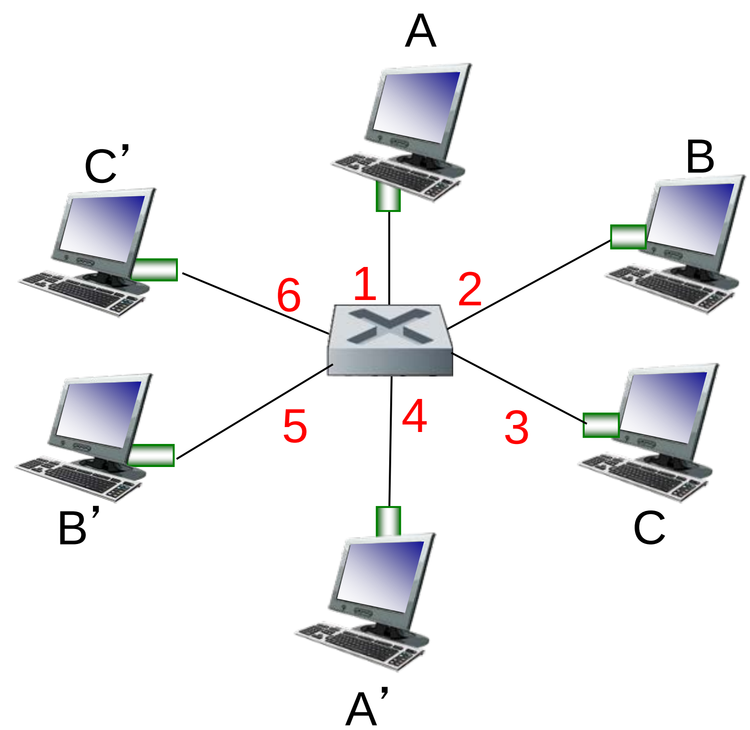
Ethernet Switch

- 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 Switch

Switch: Multiple Simultaneous Transmission

- 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



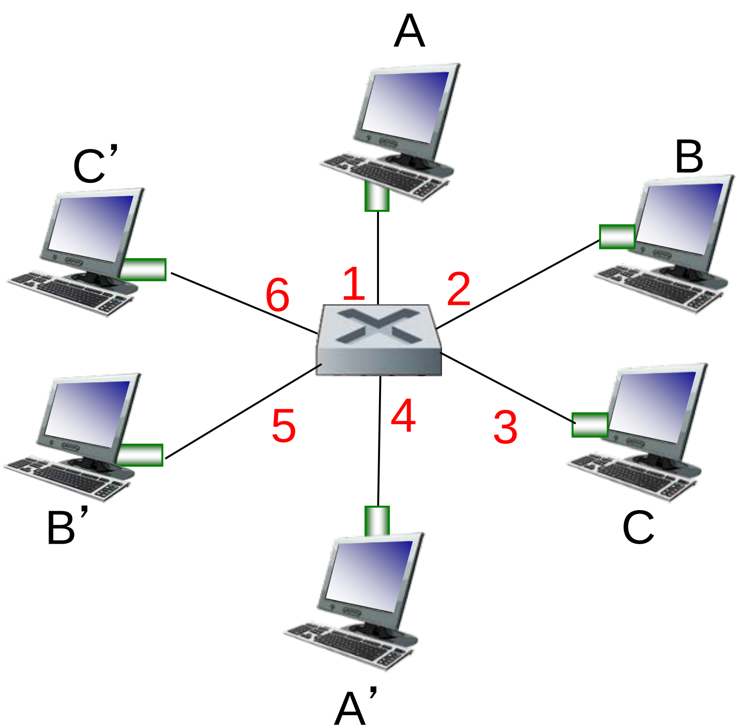
switch with six interfaces (1,2,3,4,5,6)

Ethernet Switch

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!



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

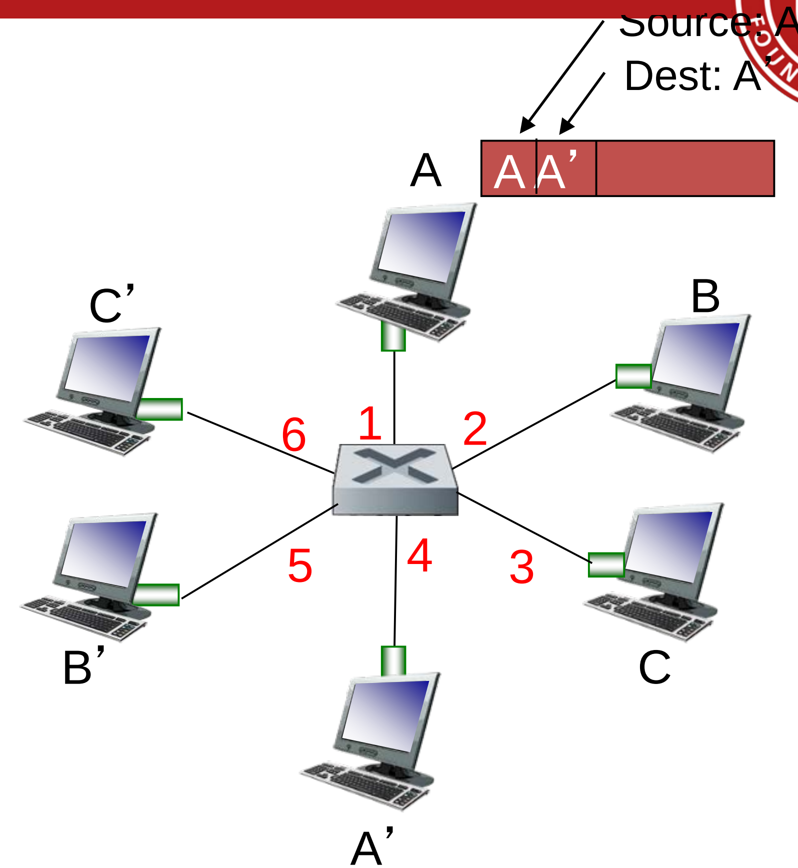
Q: how are entries created, maintained in switch table?

- something like a routing protocol?

Ethernet Switch

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)

Ethernet Switch

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

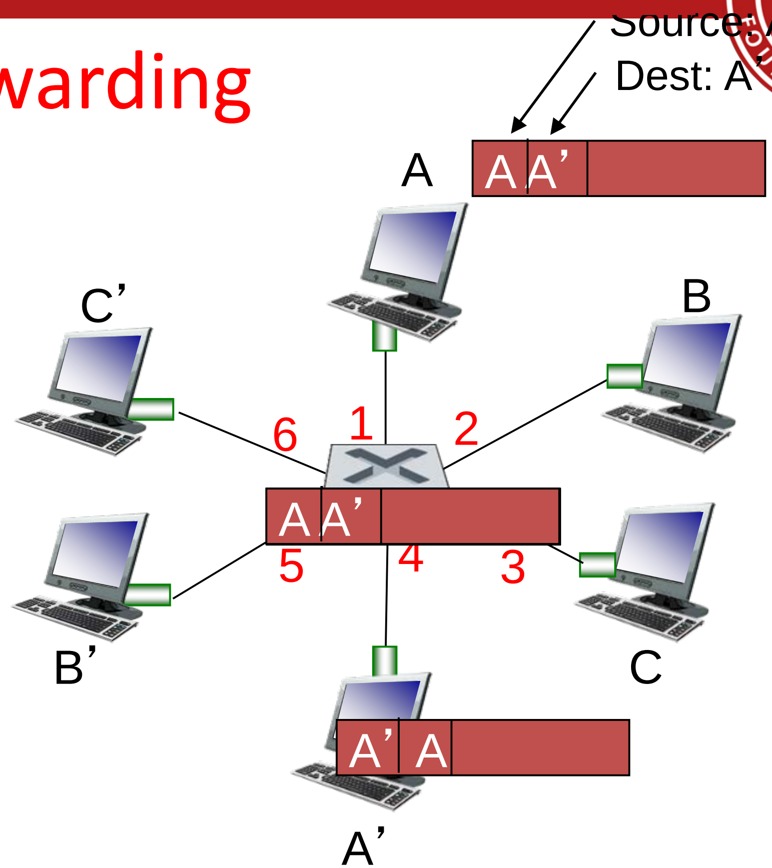
```
then {
  if destination on segment from which frame arrived
     then drop frame
     else forward frame on interface indicated by entry
}
else flood  /* forward on all interfaces except arriving
                 interface */
```

Ethernet Switch

Example: Self-learning, forwarding

- frame destination, A', locaton unknown: *flood*

- destination A location known: selectively send on just one link



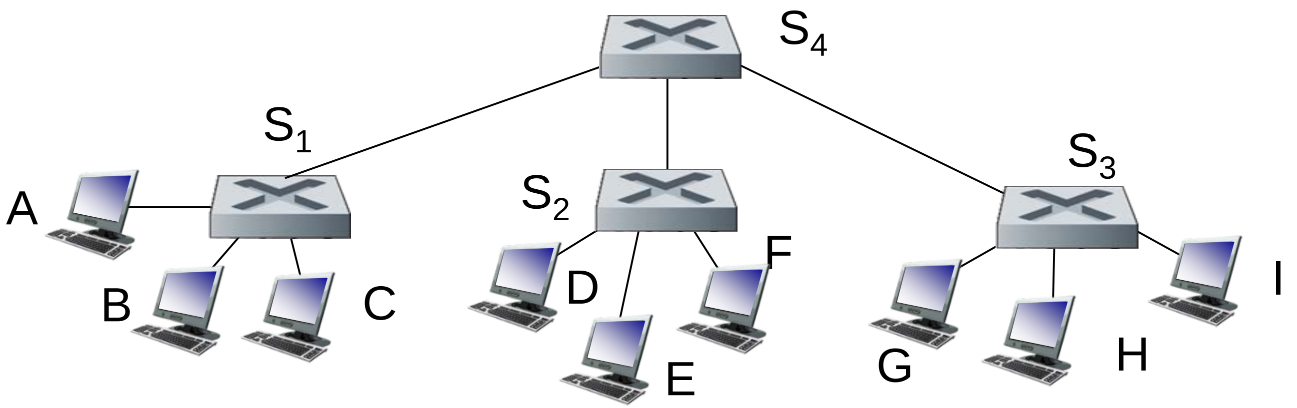
MAC addr	interface	TTL
A	*1*	*60*
A'	*4*	*60*

switch table (initially empty)

Ethernet Switch

Interconnecting Switches

- switches can be connected together

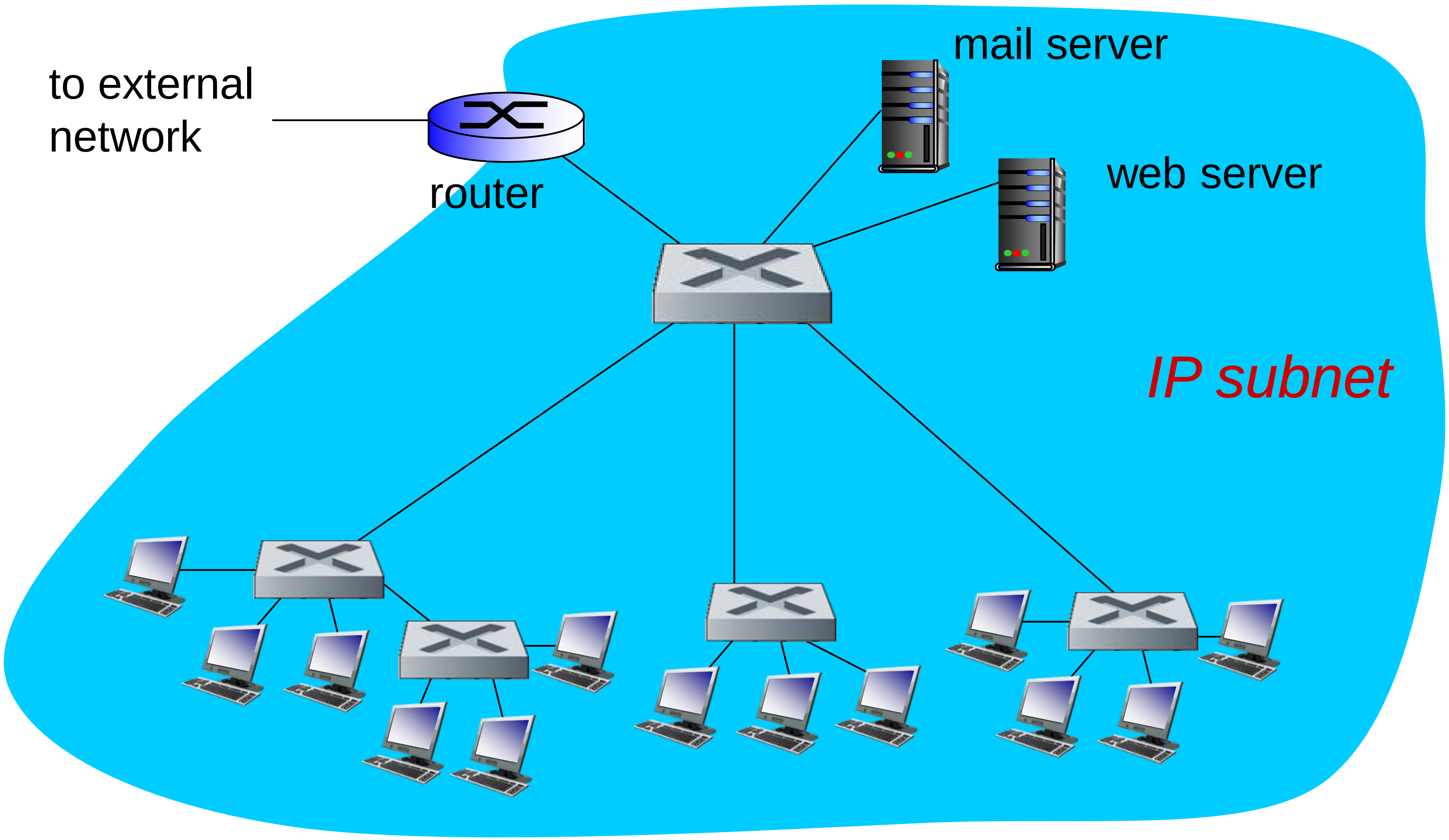


Q: sending from A to G - how does S_1 know to forward frame destined to F via S_4 and S_3?

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

Ethernet Switch

Institutional Network



Ethernet Switch

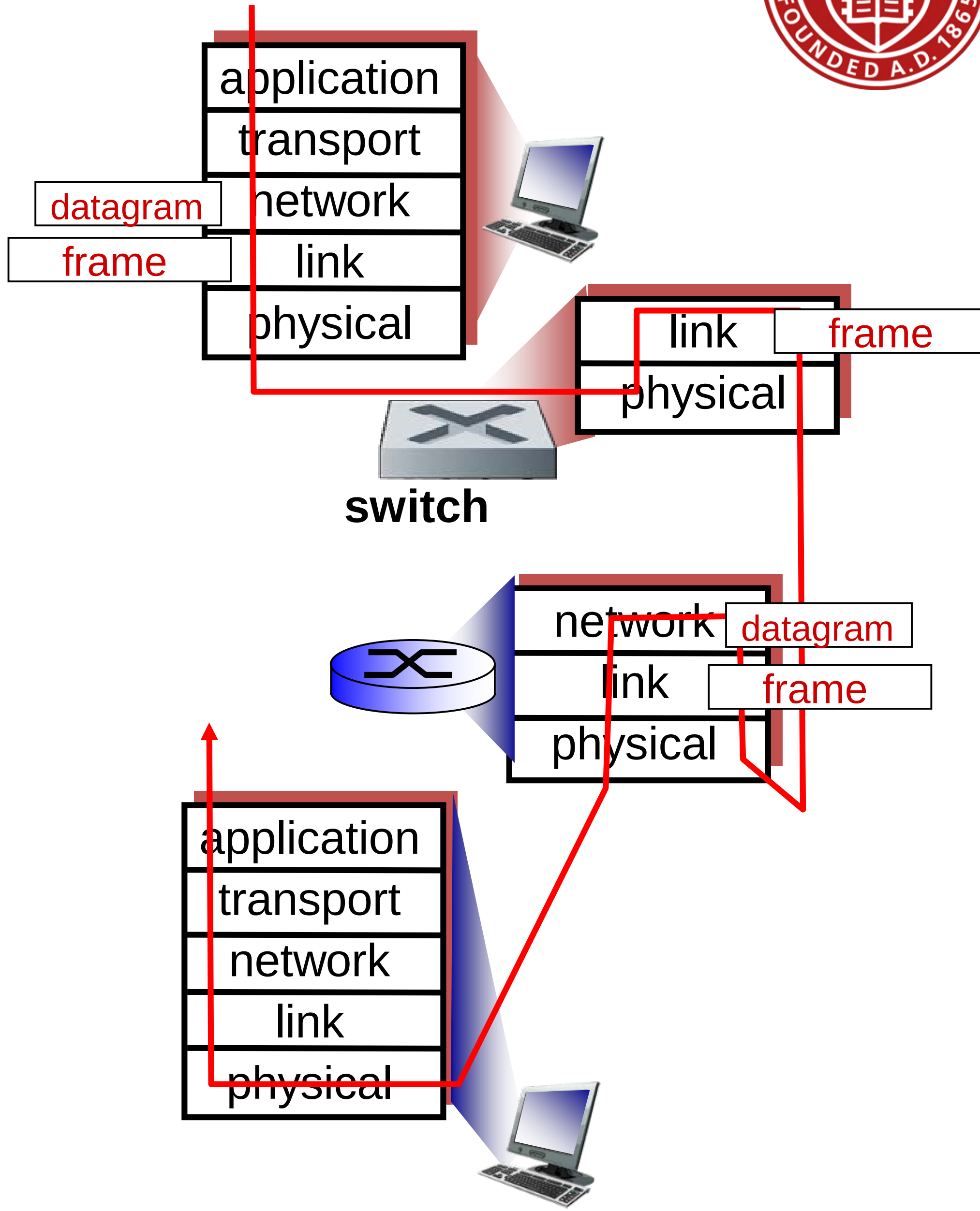
Switches vs Routers

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



Goals for Today

- Link Layer and Physical Layer
  - Abstraction / services
  - Switches and Local Area Networks
    - Addressing, ARP (address resolution protocol)
    - Ethernet
    - Ethernet switch
  - Multiple Access Protocols
- Data Center Network
  - 10GbE (10 Gigabit Ethernet)
- Backup Slides
  - Virtual Local Area Networks (VLAN)
  - Multiple Access Protocols
  - Putting it all together: A day and a life of a web request

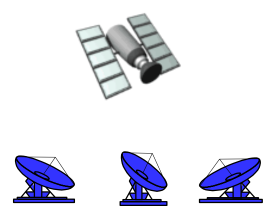
Multiple Access Links, Protocols

two types of "links":

- point-to-point
  - PPP for dial-up access
  - point-to-point link between Ethernet switch, host
- *broadcast (shared wire or medium)*
  - old-fashioned Ethernet
  - upstream HFC
  - 802.11 wireless LAN

shared wire (e.g., cabled Ethernet)

shared RF (e.g., 802.11 WiFi)



shared RF (satellite)

humans at a cocktail party (shared air, acoustical)

Multiple Access Links, Protocols

MAC Protocols: Taxonomy

three broad classes:

- *channel partitioning*
  - divide channel into smaller "pieces" (time slots, frequency, code)
  - allocate piece to node for exclusive use
- *random access*
  - channel not divided, allow collisions
  - "recover" from collisions
- *"taking turns"*
  - nodes take turns, but nodes with more to send can take longer turns

Multiple Access Links, Protocols

MAC Protocols: Tradeoffs

channel partitioning MAC 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 MAC protocols

- efficient at low load: single node can fully utilize channel
- high load: collision overhead

"taking turns" protocols

look for best of both worlds!

Multiple Access Links, Protocols

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

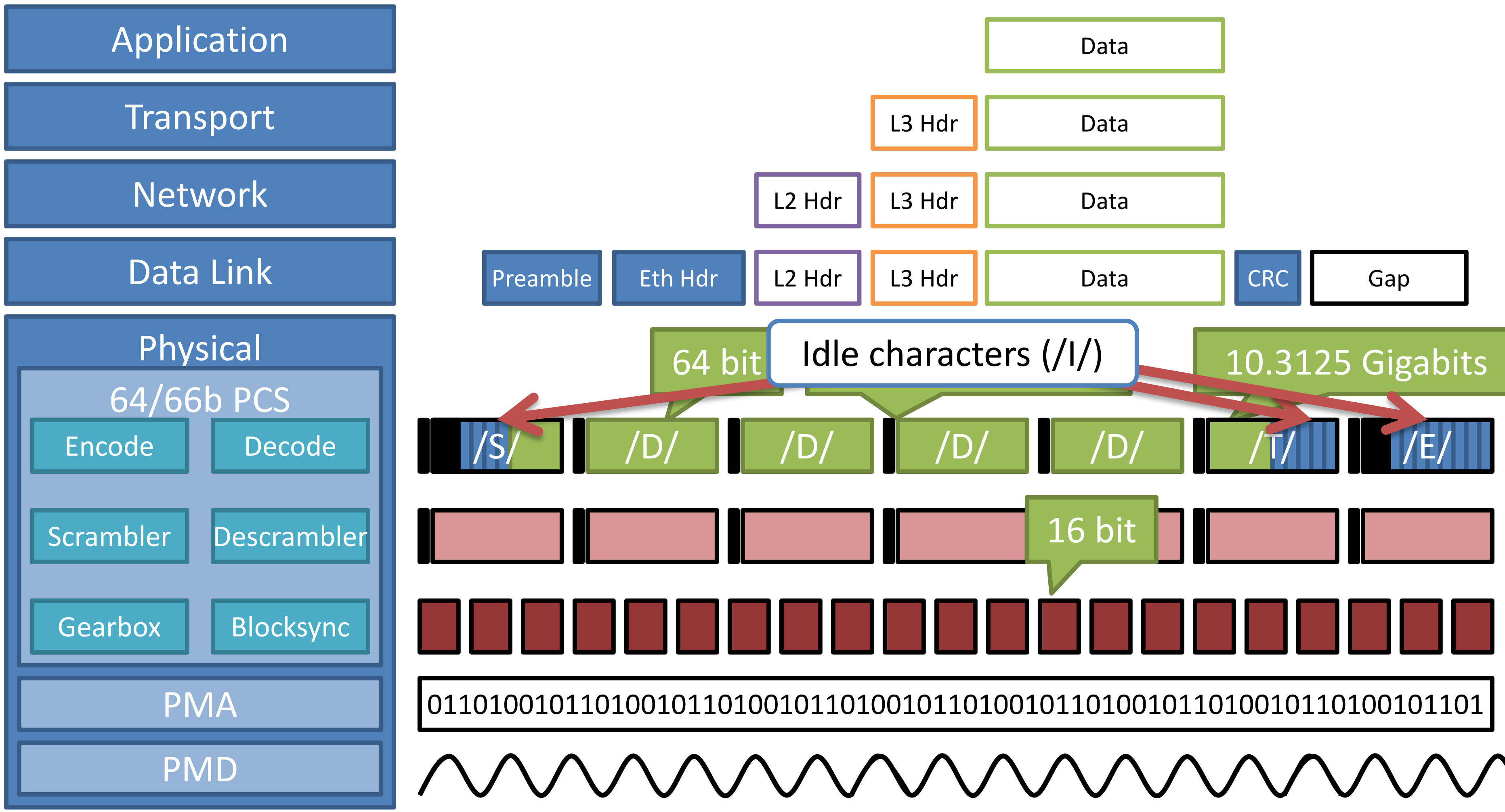
Goals for Today

- Link Layer and Physical Layer
  - Abstraction / services
  - Switches and Local Area Networks
    - Addressing, ARP (address resolution protocol)
    - Ethernet
    - Ethernet switch
  - Multiple Access Protocols

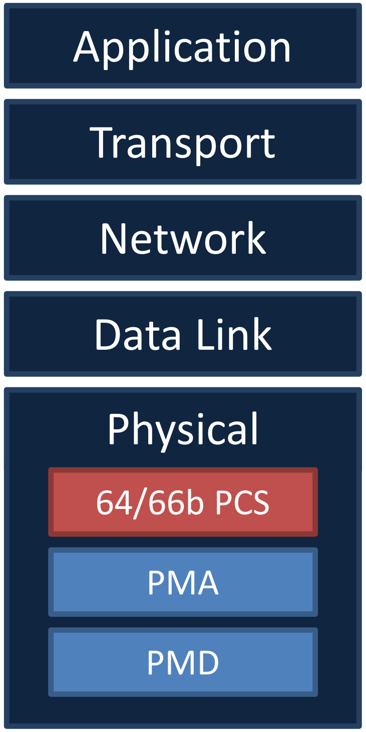
- Data Center Network
  - 10GbE (10 Gigabit Ethernet)

- Backup Slides
  - Virtual Local Area Networks (VLAN)
  - Multiple Access Protocols
  - Putting it all together: A day and a life of a web request

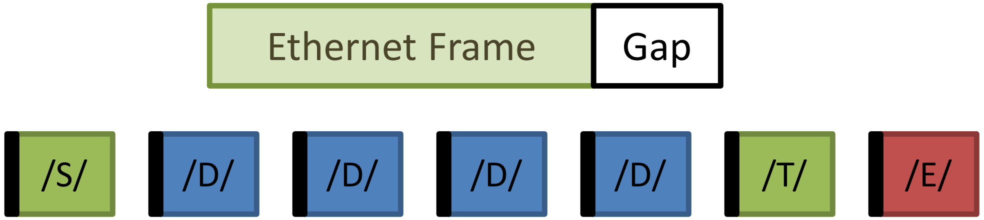
10GbE (10 gigabit Ethernet)



10GbE (10 gigabit Ethernet)



	Sync	Block Payload								
		0	8	16	24	32	40	48	56	65
Data Block	01	D0	D1	D2	D3	D4	D5	D6	D7	
		Block Type								
/E/	10	0x1e	C0	C1	C2	C3	C4	C5	C6	C7
/S/	10	0x33	C0	C1	C2	C3		D5	D6	D7
	10	0x78	D1	D2	D3	D4	D5	D6	D7	
/T/	10	0x87		C1	C2	C3	C4	C5	C6	C7
	10	0x99	D0		C2	C3	C4	C5	C6	C7
	10	0xaa	D0	D1		C3	C4	C5	C6	C7
	10	0xb4	D0	D1	D2		C4	C5	C6	C7
	10	0xcc	D0	D1	D2	D3		C5	C6	C7
	10	0xd2	D0	D1	D2	D3	D4		C6	C7
	10	0xe1	D0	D1	D2	D3	D4	D5		C7
	10	0xff	D0	D1	D2	D3	D4	D5	D6	



/S/ Start of Frame block

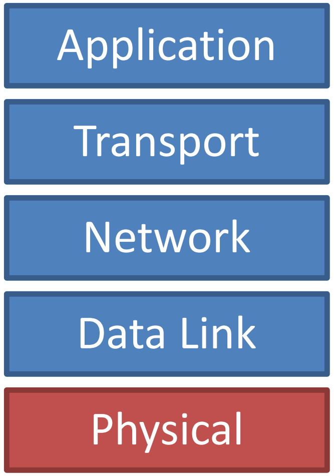
/T/ End of Frame block

/E/ Idle block

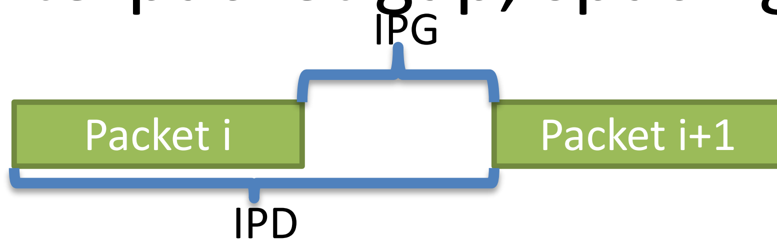
/D/ Data block

SoNIC: Software network interface card

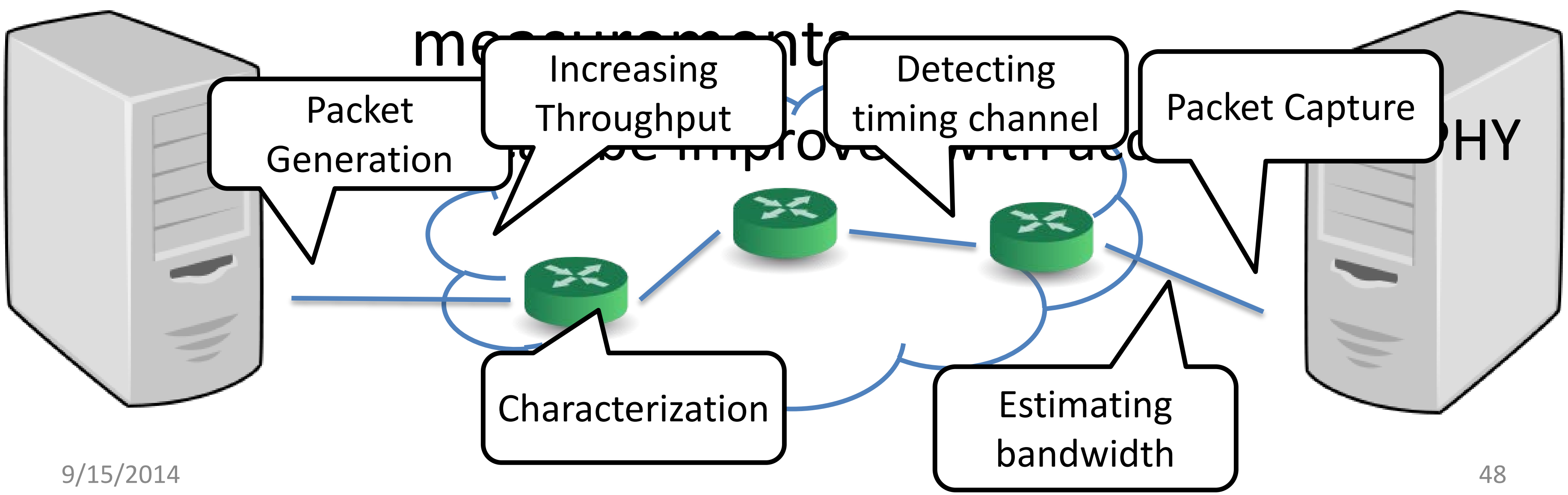
Hyper-fidelity network measurements



- Interpacket gap, spacing, arrival time, …



- Important metric for network measurements



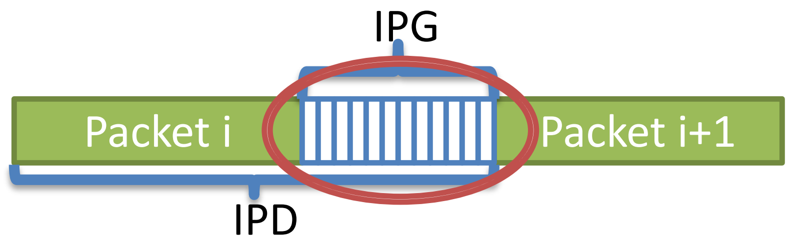
10GbE (10 gigabit Ethernet)

SoNIC: Software network interface card

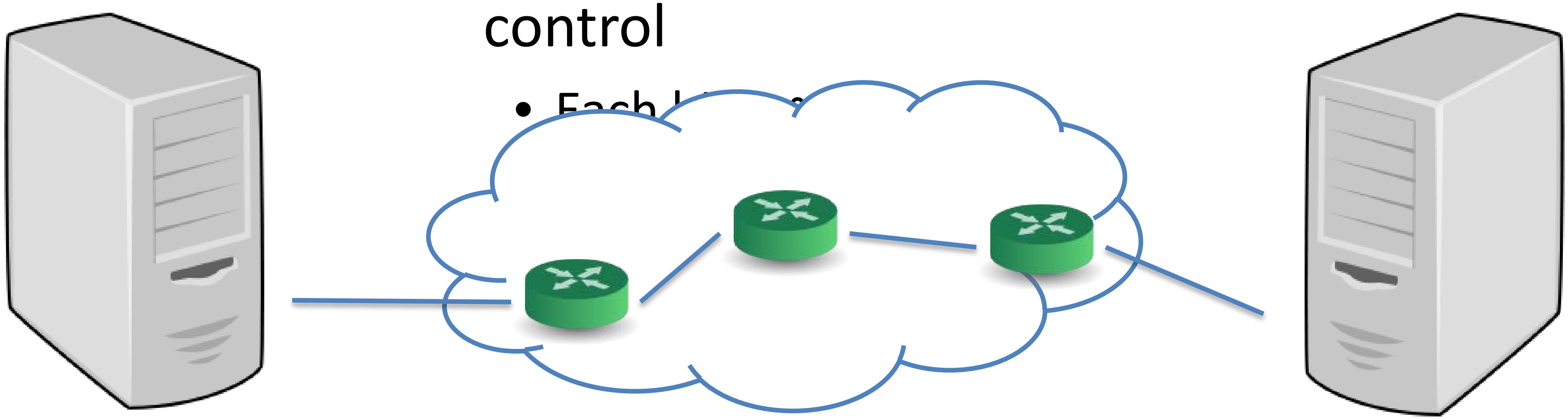
Hyper-fidelity network measurements

- Application
- Transport
- Network
- Data Link
- Physical

- Valuable information: Idle characters



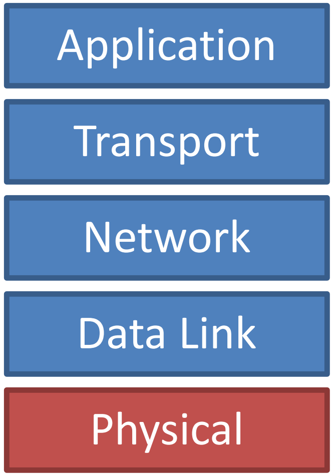
  – Can provide precise timing base for control
    - Each



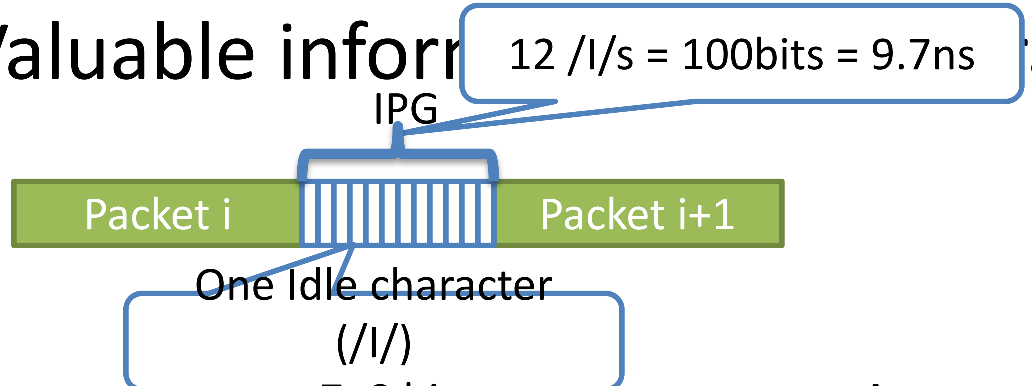
10GbE (10 gigabit Ethernet)

SoNIC: Software network interface card

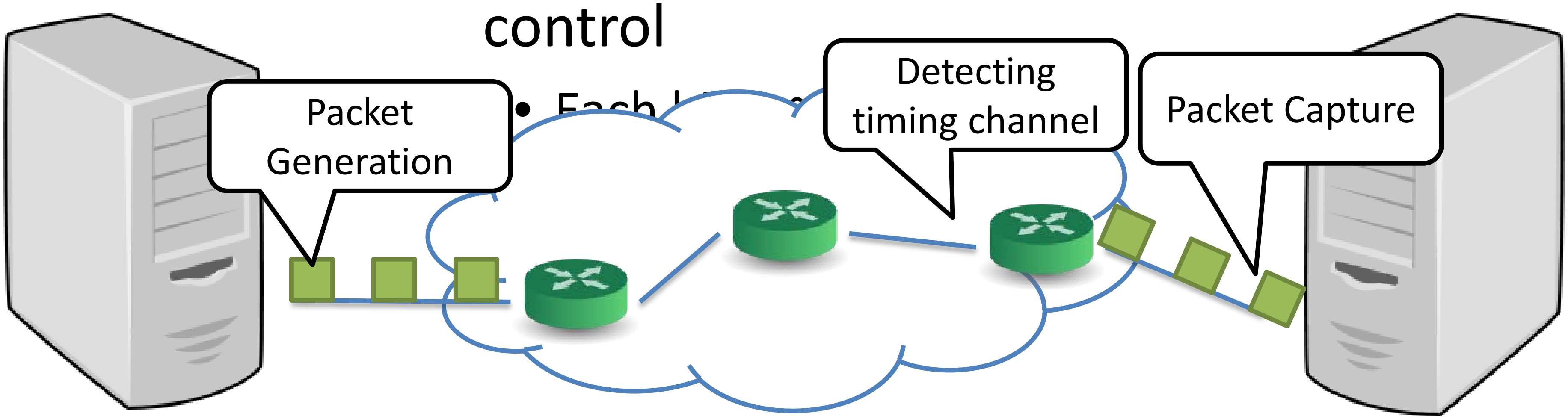
Hyper-fidelity network measurements



- Valuable inforr … acters



  – Can provide precise timing base for control
    - Each I



Goals for Today

- Link Layer and Physical Layer
  - Abstraction / services
  - Switches and Local Area Networks
    - Addressing, ARP (address resolution protocol)
    - Ethernet
    - Ethernet switch
  - Multiple Access Protocols

- Data Center Network
  - 10GbE (10 Gigabit Ethernet)

- Backup Slides
  - Virtual Local Area Networks (VLAN)
  - Multiple Access Protocols
  - Putting it all together: A day and a life of a web request

Perspective

- principles behind data link layer services:
  - link layer addressing
  - sharing a broadcast channel: multiple access
  - error detection, correction
- instantiation and implementation of various link layer technologies
  - Ethernet
  - switched LANS

Perspective

- Journey down protocol stack *complete*
- Basic understanding of networking practices and principles
- *lots* of interesting topics in data center and high performance systems and networks

Before Next time

- Project Proposal
  - **due today, Friday**
  - Meet with groups, TA, and professor
- Lab1
  - Single threaded TCP proxy
  - **Due today, Friday**

- ***Required review and reading***
  - "A 50-Gb/s IP Router," Craig Partridge , Senior Member , Philip P. Carvey , Isidro Castineyra , Tom Clarke , John Rokosz , Joshua Seeger , Michael Sollins , Steve Starch , Benjamin Tober , Gregory D. Troxel , David Waitzman , Scott Winterble. *IEEE/ACM Transactions on Networking (ToN),* Volume 6, Issue 3 (June 1998), pages 237-248.
  - http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=700888
  - http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.129.3926&rep=rep1&type=pdf

- Check website for updated schedule

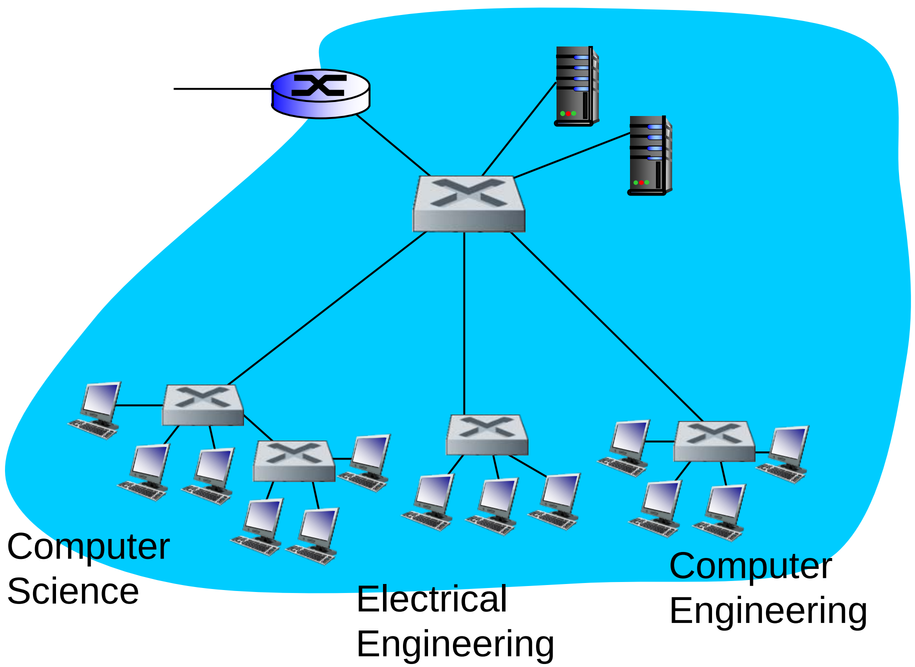
Goals for Today

- Link Layer and Physical Layer
  - Abstraction / services
  - Switches and Local Area Networks
    - Addressing, ARP (address resolution protocol)
    - Ethernet
    - Ethernet switch
  - Multiple Access Protocols

- Data Center Network
  - 10GbE (10 Gigabit Ethernet)

- Backup Slides
  - Virtual Local Area Networks (VLAN)
  - Multiple Access Protocols
  - Putting it all together: A day and a life of a web request

Virtual Local Area Networks (VLAN)



consider:

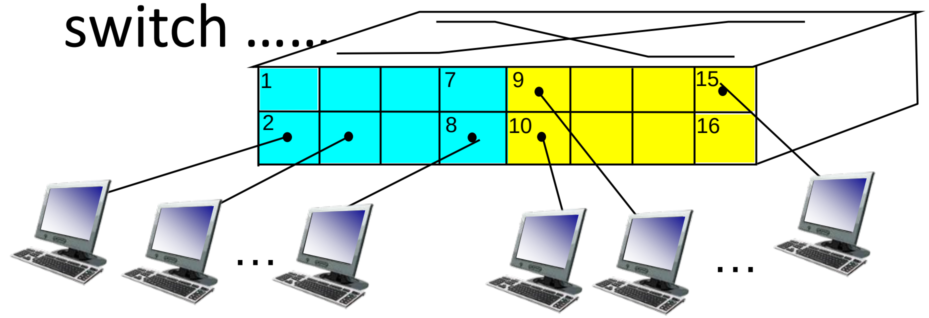
- CS user moves office to EE, but wants connect to CS switch?
- single broadcast domain:
  - all layer-2 broadcast traffic (ARP, DHCP, unknown location of destination MAC address) must cross entire LAN
  - security/privacy, efficiency issues

Virtual Local Area Networks (VLAN)

Virtual Local Area Network

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

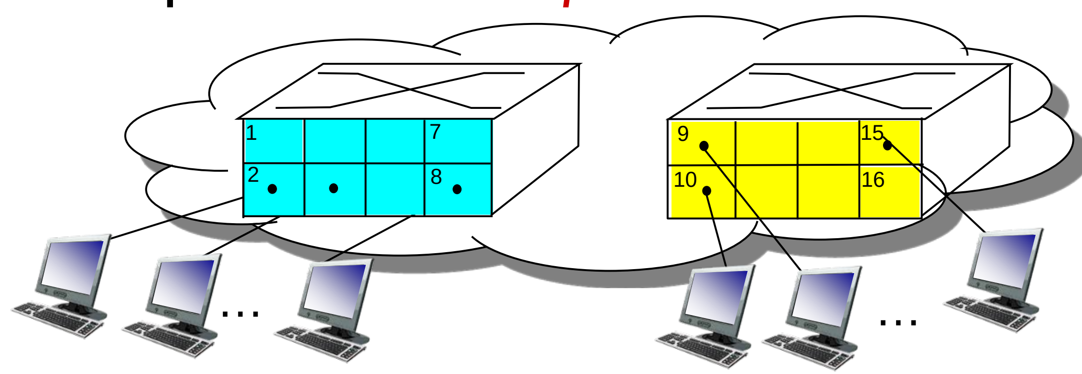
port-based VLAN: switch ports grouped (by switch management software) so that *single* physical switch ......



Electrical Engineering
(VLAN ports 1-8)

Computer Science
(VLAN ports 9-15)

… operates as *multiple* virtual switches



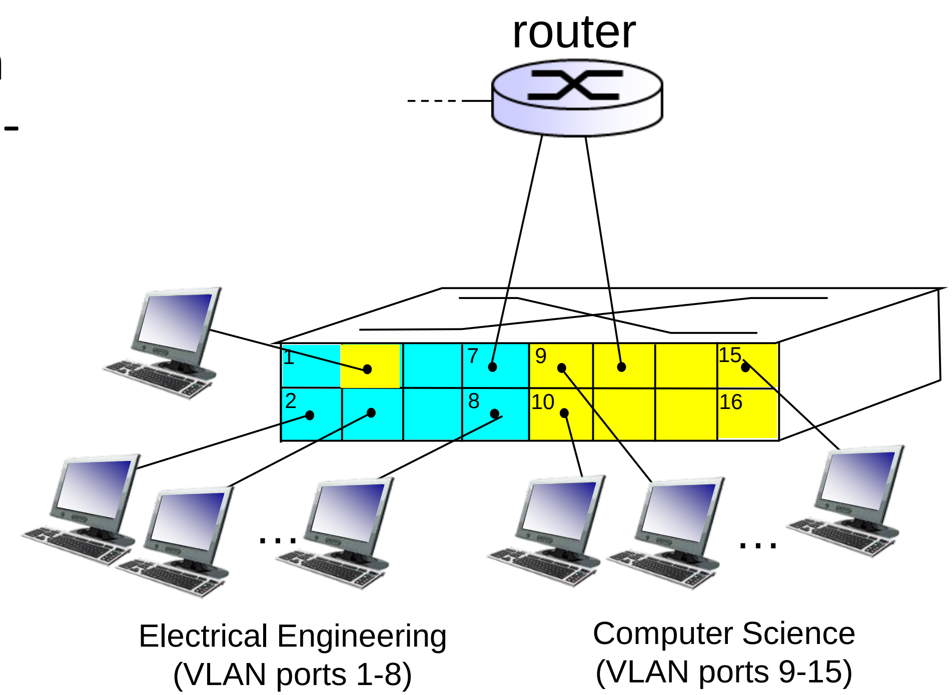
Electrical Engineering
(VLAN ports 1-8)

Computer Science
(VLAN ports 9-16)

Virtual Local Area Networks (VLAN)

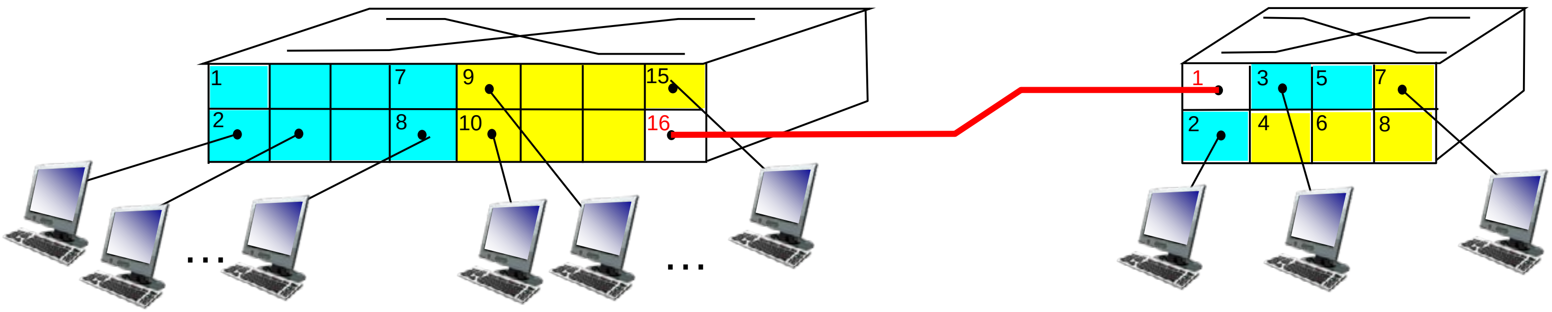
Port-based VLAN

- *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 VLANS: done via routing (just as with separate switches)
  - in practice vendors sell combined switches plus routers



Virtual Local Area Networks (VLAN)

VLANs spanning multiple switches



Electrical Engineering
(VLAN ports 1-8)

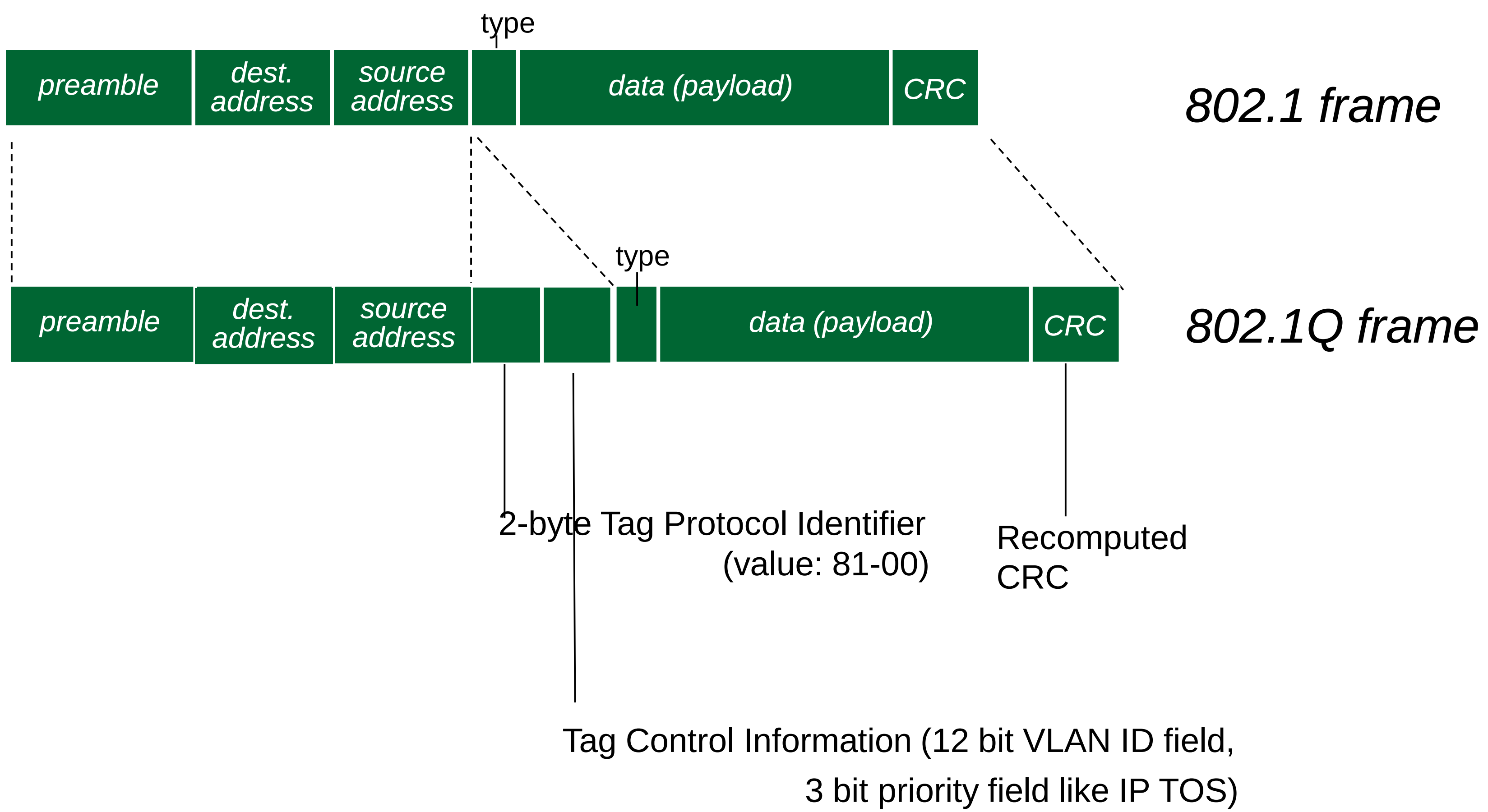
Computer Science
(VLAN ports 9-15)

Ports 2,3,5 belong to EE VLAN
Ports 4,6,7,8 belong to CS VLAN

- *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

Virtual Local Area Networks (VLAN)

802.1Q VLAN frame format



Goals for Today

- Link Layer and Physical Layer
  - Abstraction / services
  - Switches and Local Area Networks
    - Addressing, ARP (address resolution protocol)
    - Ethernet
    - Ethernet switch
  - Multiple Access Protocols

- Data Center Network
  - 10GbE (10 Gigabit Ethernet)

- Backup Slides
  - Virtual Local Area Networks (VLAN)
  - Multiple Access Protocols
  - Putting it all together: A day and a life of a web request

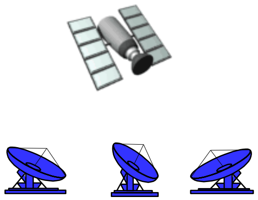
Multiple Access Links, Protocols

two types of "links":

- point-to-point
  - PPP for dial-up access
  - point-to-point link between Ethernet switch, host
- *broadcast (shared wire or medium)*
  - old-fashioned Ethernet
  - upstream HFC
  - 802.11 wireless LAN

shared wire (e.g., cabled Ethernet)

shared RF (e.g., 802.11 WiFi)



shared RF (satellite)

humans at a cocktail party (shared air, acoustical)

Multiple access protocols

- single shared broadcast channel
- two or more simultaneous transmissions by nodes: interference
  - *collision* if node receives two or more signals at the same time

multiple access protocol

- distributed algorithm that determines how nodes share channel, i.e., determine when node can transmit
- communication about channel sharing must use channel itself!
  - no out-of-band channel for coordination

An ideal multiple access protocol

given: broadcast channel of rate R bps

desiderata:

1. when one node wants to transmit, it can send at rate R.
2. when M nodes want to transmit, each can send at average rate R/M
3. fully decentralized:
   - no special node to coordinate transmissions
   - no synchronization of clocks, slots
4. simple

Multiple Access Links, Protocols

MAC Protocols: Taxonomy

three broad classes:

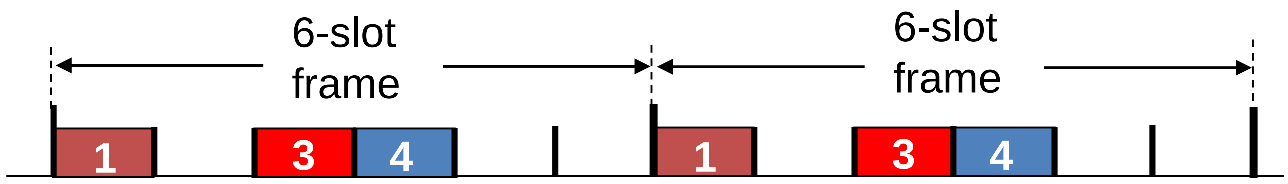
- *channel partitioning*
  - divide channel into smaller "pieces" (time slots, frequency, code)
  - allocate piece to node for exclusive use
- *random access*
  - channel not divided, allow collisions
  - "recover" from collisions
- *"taking turns"*
  - nodes take turns, but nodes with more to send can take longer turns

Multiple Access Links, Protocols

Channel Partitioning MAC protocols: TDMA

TDMA: time division multiple access

- access to channel in "rounds"
- each station gets fixed length slot (length = pkt trans time) in each round
- unused slots go idle
- example: 6-station LAN, 1,3,4 have pkt, slots 2,5,6 idle

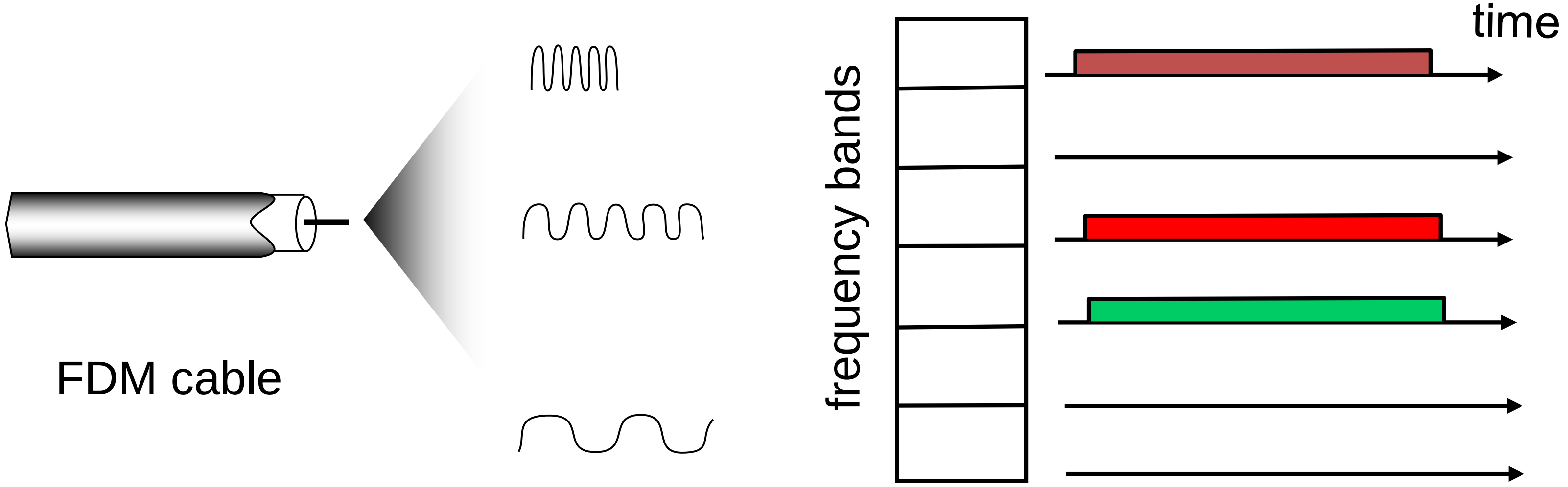


Multiple Access Links, Protocols

Channel Partitioning MAC protocols: FDMA

FDMA: frequency division multiple access

- channel spectrum divided into frequency bands
- each station assigned fixed frequency band
- unused transmission time in frequency bands go idle
- example: 6-station LAN, 1,3,4 have pkt, frequency bands 2,5,6 idle



Multiple Access Links, Protocols

Random Access Protocols

- when node has packet to send
  - transmit at full channel data rate R.
  - no *a priori* coordination among nodes
- two or more transmitting nodes ➜ "collision",
- random access MAC protocol specifies:
  - how to detect collisions
  - how to recover from collisions (e.g., via delayed retransmissions)
- examples of random access MAC protocols:
  - slotted ALOHA
  - ALOHA
  - CSMA, CSMA/CD, CSMA/CA

Multiple Access Links, Protocols

Slotted ALOHA

assumptions:

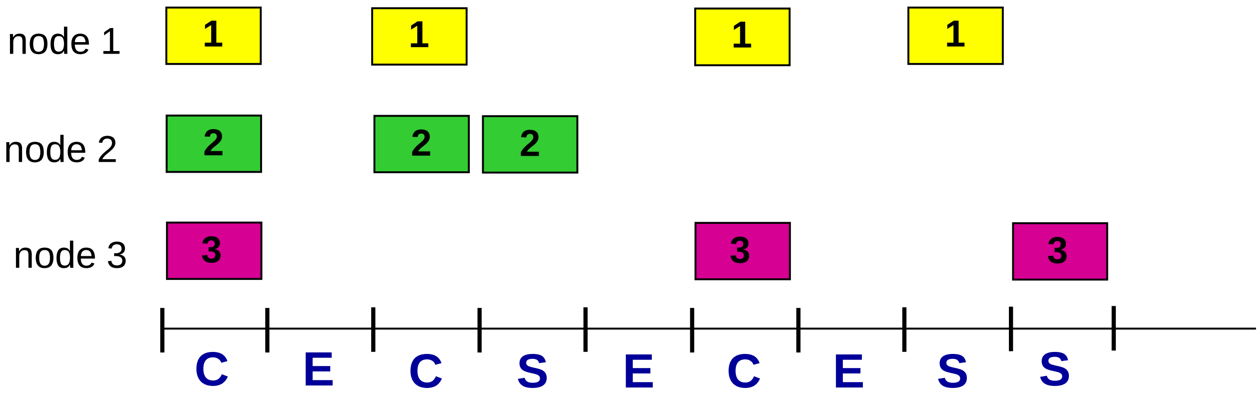
- all frames same size
- time divided into equal size slots (time to transmit 1 frame)
- nodes start to transmit only slot beginning
- nodes are synchronized
- if 2 or more nodes transmit in slot, all nodes detect collision

operation:

- when node obtains fresh frame, transmits in next slot
  - *if no collision:* node can send new frame in next slot
  - *if collision:* node retransmits frame in each subsequent slot with prob. p until success

Multiple Access Links, Protocols

Slotted ALOHA



Pros:

- single active node can continuously transmit at full rate of channel
- highly decentralized: only slots in nodes need to be in sync
- simple

Cons:

- collisions, wasting slots
- idle slots
- nodes may be able to detect collision in less than time to transmit packet
- clock synchronization

Slotted ALOHA: Efficiency

efficiency: long-run fraction of successful slots (many nodes, all with many frames to send)

- *suppose: N* nodes with many frames to send, each transmits in slot with probability *p*
- prob that given node has success in a slot = $p(1-p)^{N-1}$
- prob that *any* node has a success = $Np(1-p)^{N-1}$

- max efficiency: find p* that maximizes $Np(1-p)^{N-1}$
- for many nodes, take limit of $Np^*(1-p^*)^{N-1}$ as N goes to infinity, gives:

max efficiency = 1/e = .37

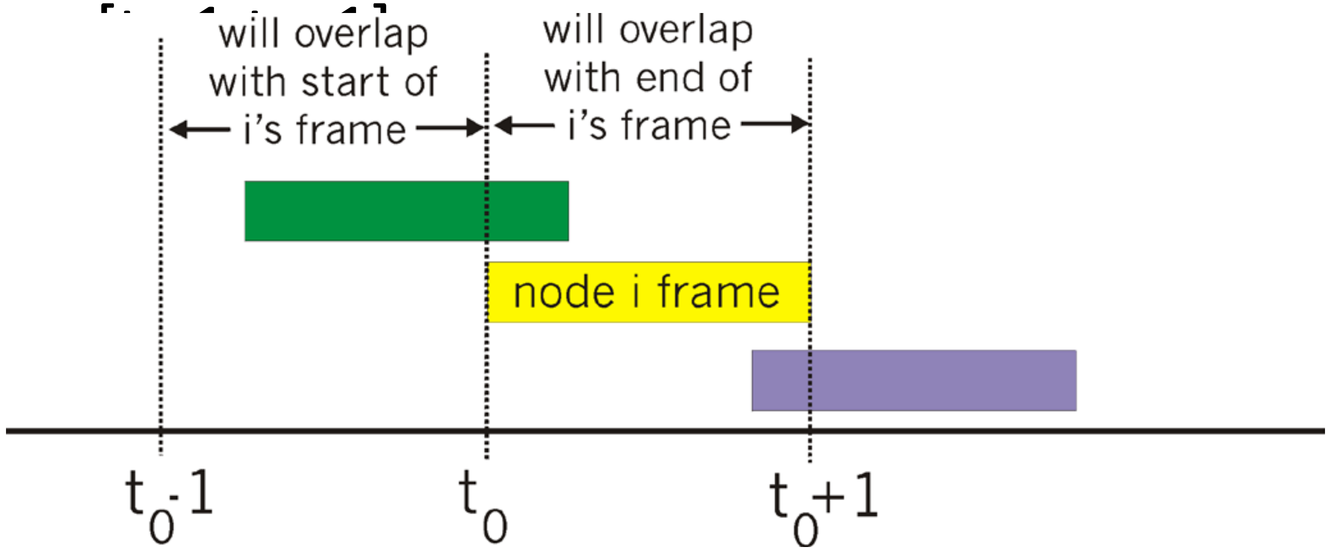
at best: channel used for useful transmissions 37% of time!

!

Multiple Access Links, Protocols

Pure (unslotted) ALOHA

- unslotted Aloha: simpler, no synchronization
- when frame first arrives
  - transmit immediately
- collision probability increases:
  - frame sent at t_0 collides with other frames sent in



Multiple Access Links, Protocols

Pure (unslotted) ALOHA

P(success by given node) = P(node transmits) ·

P(no other node transmits in $[t_0\text{-}1, t_0]$ ·

P(no other node transmits in $[t_0\text{-}1, t_0]$

$$= p \cdot (1-p)^{N-1} \cdot (1-p)^{N-1}$$

$$= p \cdot (1-p)^{2(N-1)}$$

… choosing optimum p and then letting n $\rightarrow \infty$

= 1/(2e) = .18

even *worse* than slotted Aloha!

Multiple Access Links, Protocols

CSMA: Carrier Sense Multiple Access

CSMA: listen before transmit:

if channel sensed idle: transmit entire frame

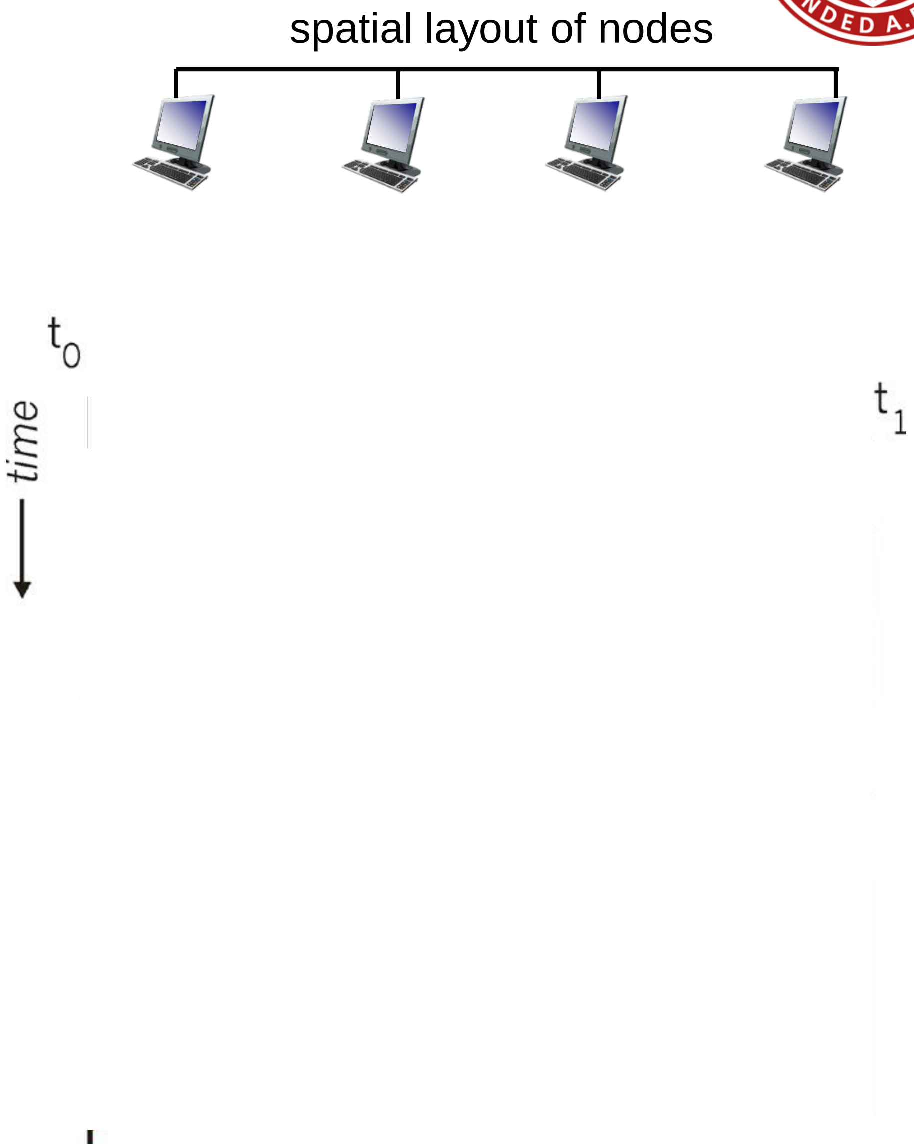
- if channel sensed busy, defer transmission

- human analogy: don't interrupt others!

Multiple Access Links, Protocols

CSMA collisions

- collisions *can* still occur: propagation delay means two nodes may not hear each other's transmission
- collision: entire packet transmission time wasted
  - distance & propagation delay play role in in determining collision probability



Multiple Access Links, Protocols

CSMA/CD (collision detection)

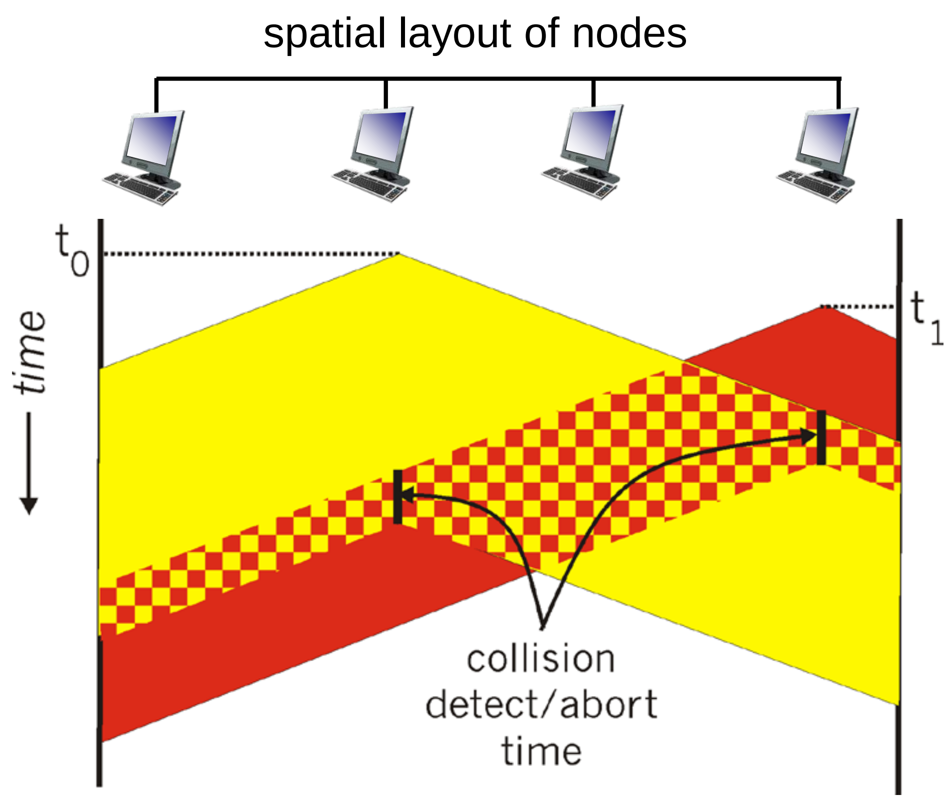
CSMA/CD: carrier sensing, deferral as in CSMA

- collisions *detected* within short time
- colliding transmissions aborted, 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
- human analogy: the polite conversationalist

Multiple Access Links, Protocols

CSMA/CD (collision detection)



Multiple Access Links, Protocols

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 *binary (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
   - longer backoff interval with more collisions

CSMA/CD efficiency

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

- efficiency goes to 1
  - as t_{prop} goes to 0
  - as t_{trans} goes to infinity
- better performance than ALOHA: and simple, cheap, decentralized!

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

Multiple Access Links, Protocols

channel partitioning MAC 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 MAC protocols

- efficient at low load: single node can fully utilize channel
- high load: collision overhead

"taking turns" protocols

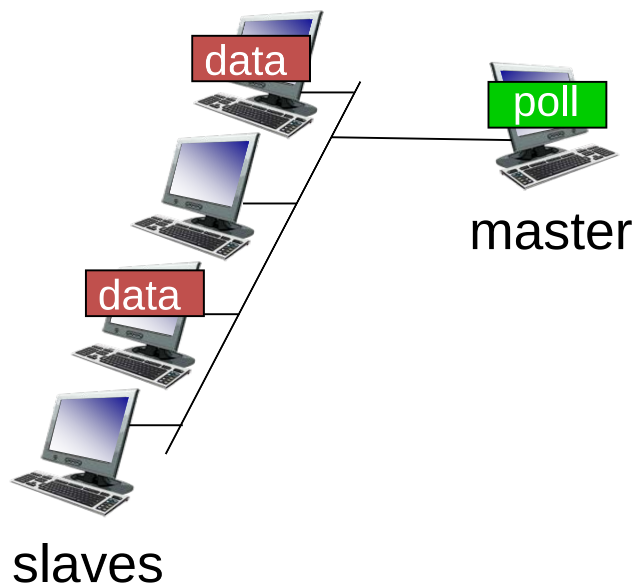
look for best of both worlds!

Multiple Access Links, Protocols

"Taking turns" MAC protocols

polling:

- master node "invites" slave nodes to transmit in turn
- typically used with "dumb" slave devices
- concerns:
  - polling overhead
  - latency
  - single point of failure (master)

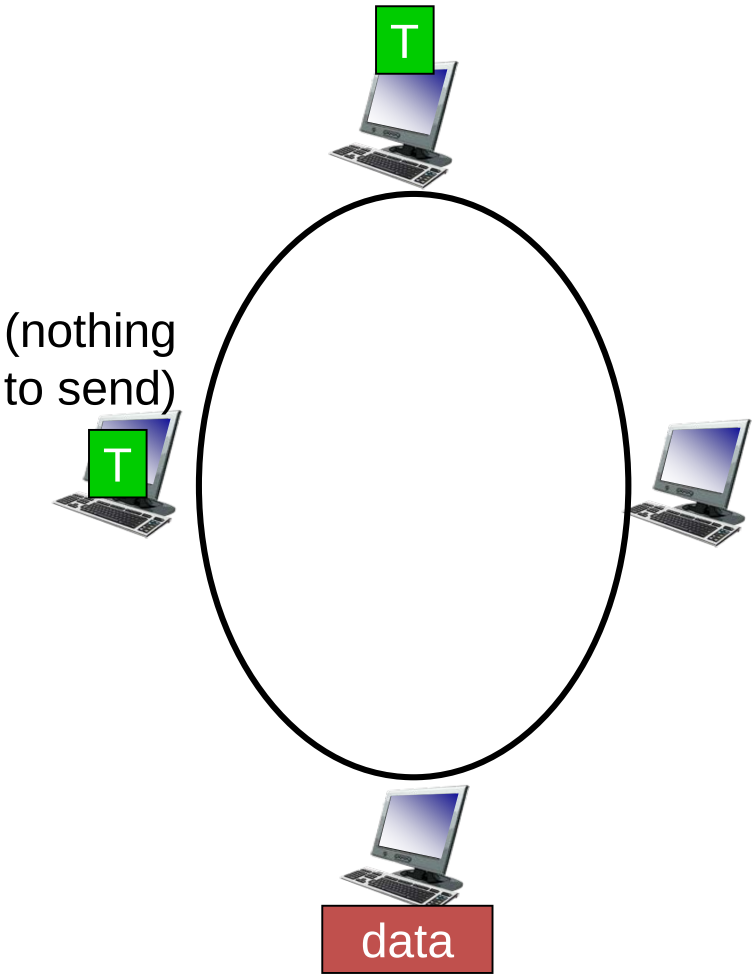


Multiple Access Links, Protocols

"Taking turns" MAC protocols

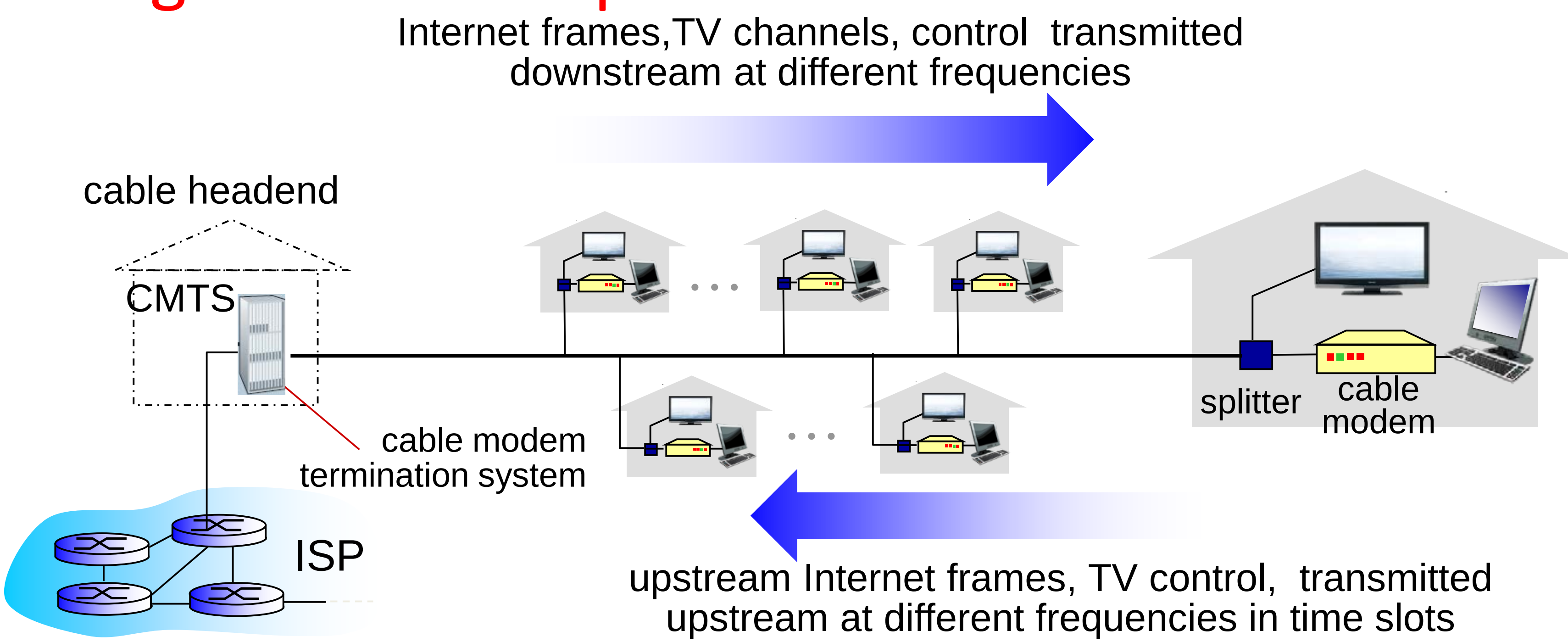
token passing:

- control token passed from one node to next sequentially.
- token message
- concerns:
  - token overhead
  - latency
  - single point of failure (token)



Multiple Access Links, Protocols

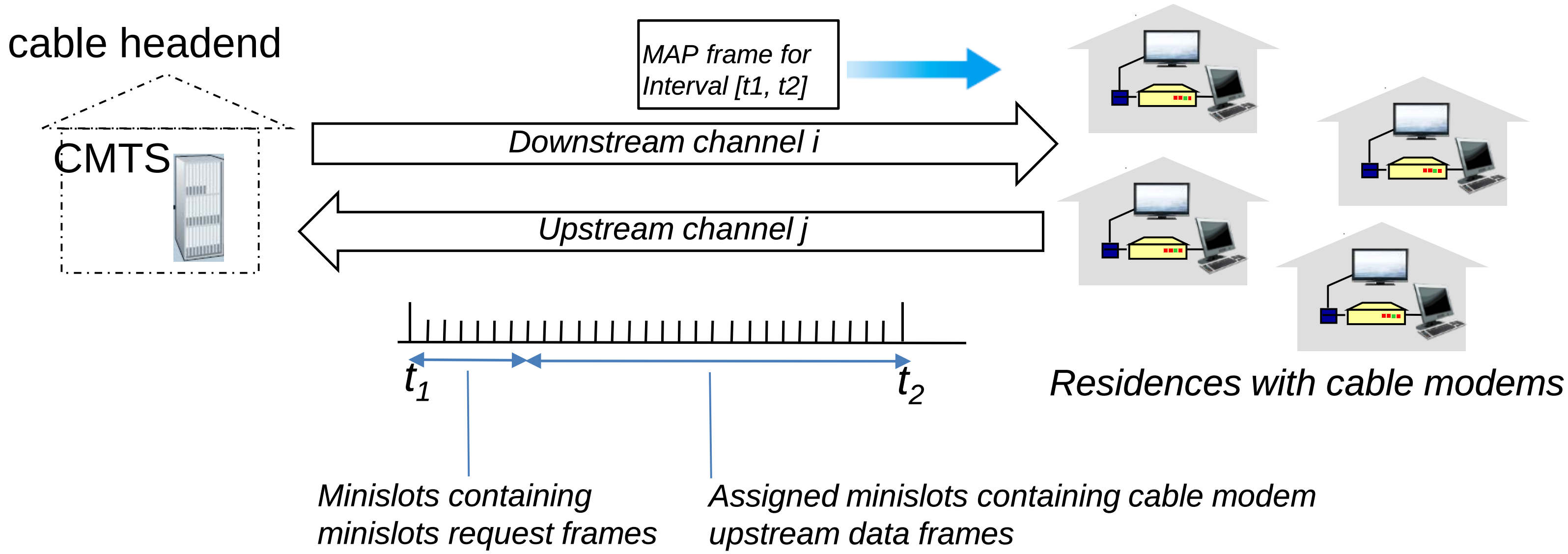
"Taking turns" MAC protocols: Cable Access Networks



- *multiple* 40Mbps downstream (broadcast) channels
  - single CMTS transmits into channels
- *multiple* 30 Mbps upstream channels
  - *multiple access: all* users contend for certain upstream channel time slots (others assigned)

Multiple Access Links, Protocols

"Taking turns" MAC protocols: Cable Access Networks



DOCSIS: data over cable service interface spec

- FDM over upstream, downstream frequency channels
- TDM upstream: some slots assigned, some have contention
  - downstream MAP frame: assigns upstream slots
  - request for upstream slots (and data) transmitted random access (binary backoff) in selected slots

Multiple Access Links, Protocols

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

Goals for Today

- Link Layer and Physical Layer
  - Abstraction / services
  - Switches and Local Area Networks
    - Addressing, ARP (address resolution protocol)
    - Ethernet
    - Ethernet switch
  - Multiple Access Protocols

- Data Center Network
  - 10GbE (10 Gigabit Ethernet)

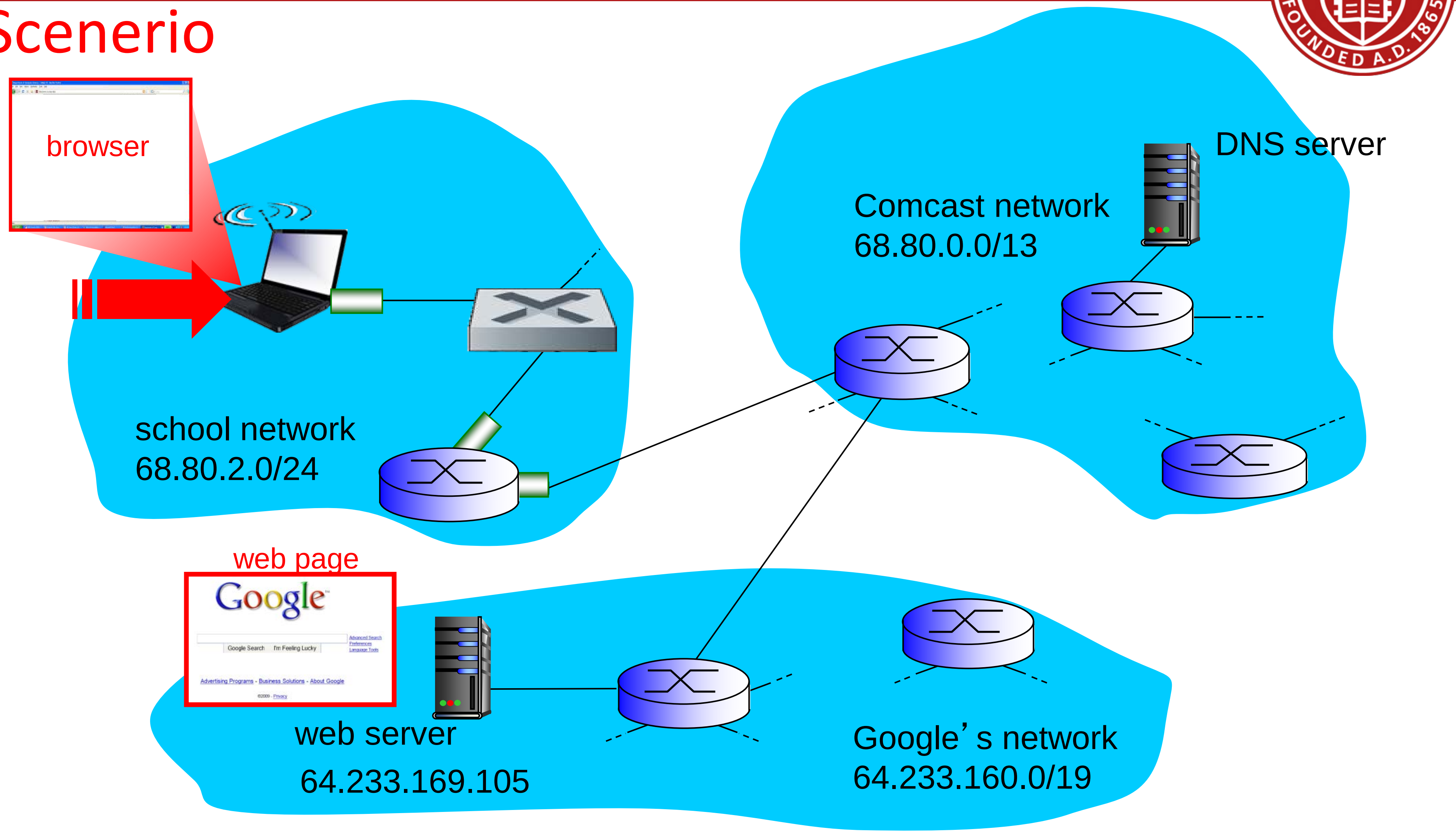
- Backup Slides
  - Virtual Local Area Networks (VLAN)
  - Multiple Access Protocols
  - Putting it all together: A day and a life of a web request

Putting it all together: A day and a life of a web req

- 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

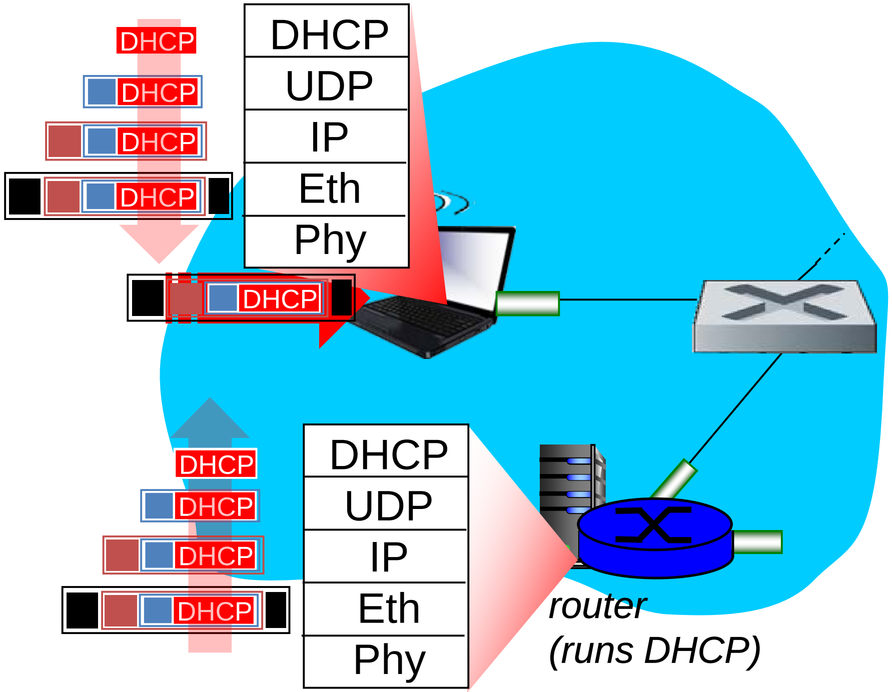
Putting it all together: A day and a life of a web req

Scenerio



Putting it all together: A day and a life of a web req

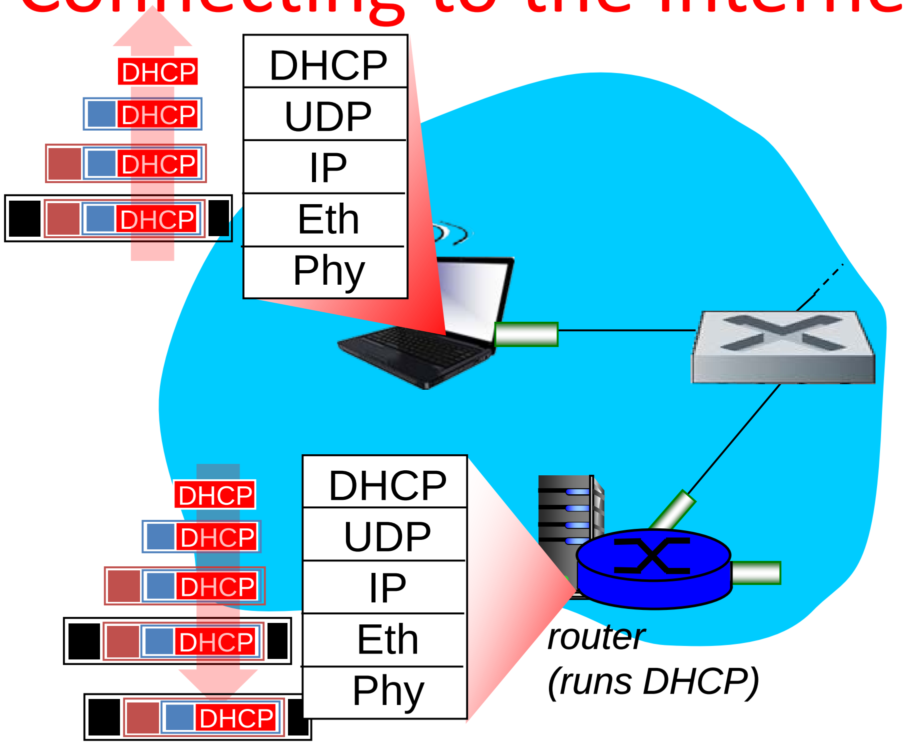
Connecting to the Internet



- connecting laptop needs to get its own IP address, addr of first-hop router, addr of DNS server: use *DHCP*
- DHCP request encapsulated in UDP, encapsulated in IP, encapsulated in 802.3 Ethernet
- Ethernet frame broadcast (dest: FFFFFFFFFFFF) on LAN, received at router running DHCP server
- Ethernet demuxed to IP demuxed, UDP demuxed to DHCP

Putting it all together: A day and a life of a web req

Connecting to the Internet

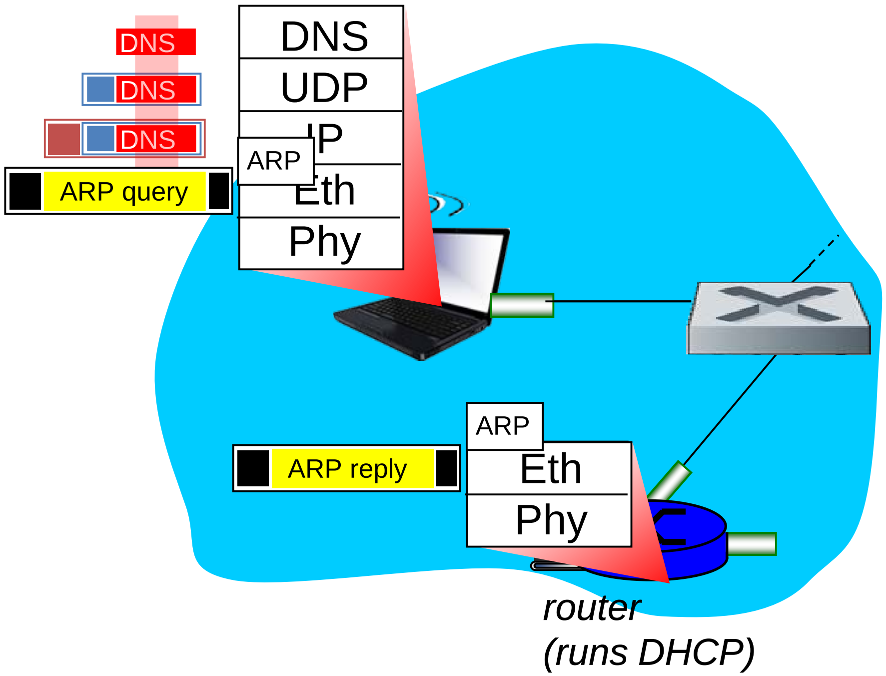


- DHCP server formulates *DHCP ACK* containing client's IP address, IP address of first-hop router for client, name & IP address of DNS server
- encapsulation at DHCP server, frame forwarded (switch learning) through LAN, demultiplexing at client
- DHCP client receives DHCP ACK reply

Client now has IP address, knows name & addr of DNS server, IP address of its first-hop router

Putting it all together: A day and a life of a web req

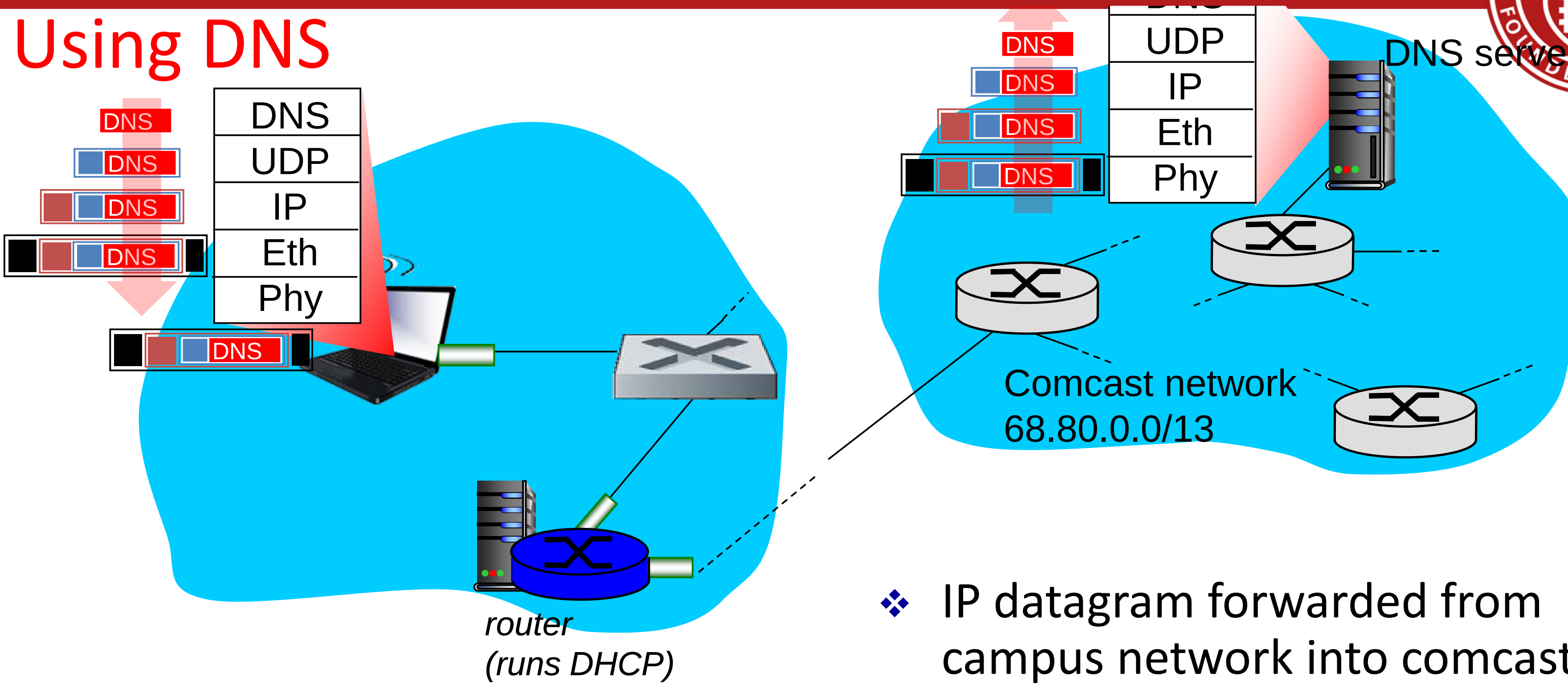
ARP (before DNS, before HTTP)



- before sending *HTTP* request, need IP address of www.google.com: *DNS*
- DNS query created, encapsulated in UDP, encapsulated in IP, encapsulated in Eth. 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
- client now knows MAC address of first hop router, so can now send frame containing DNS query

Putting it all together: A day and a life of a web req

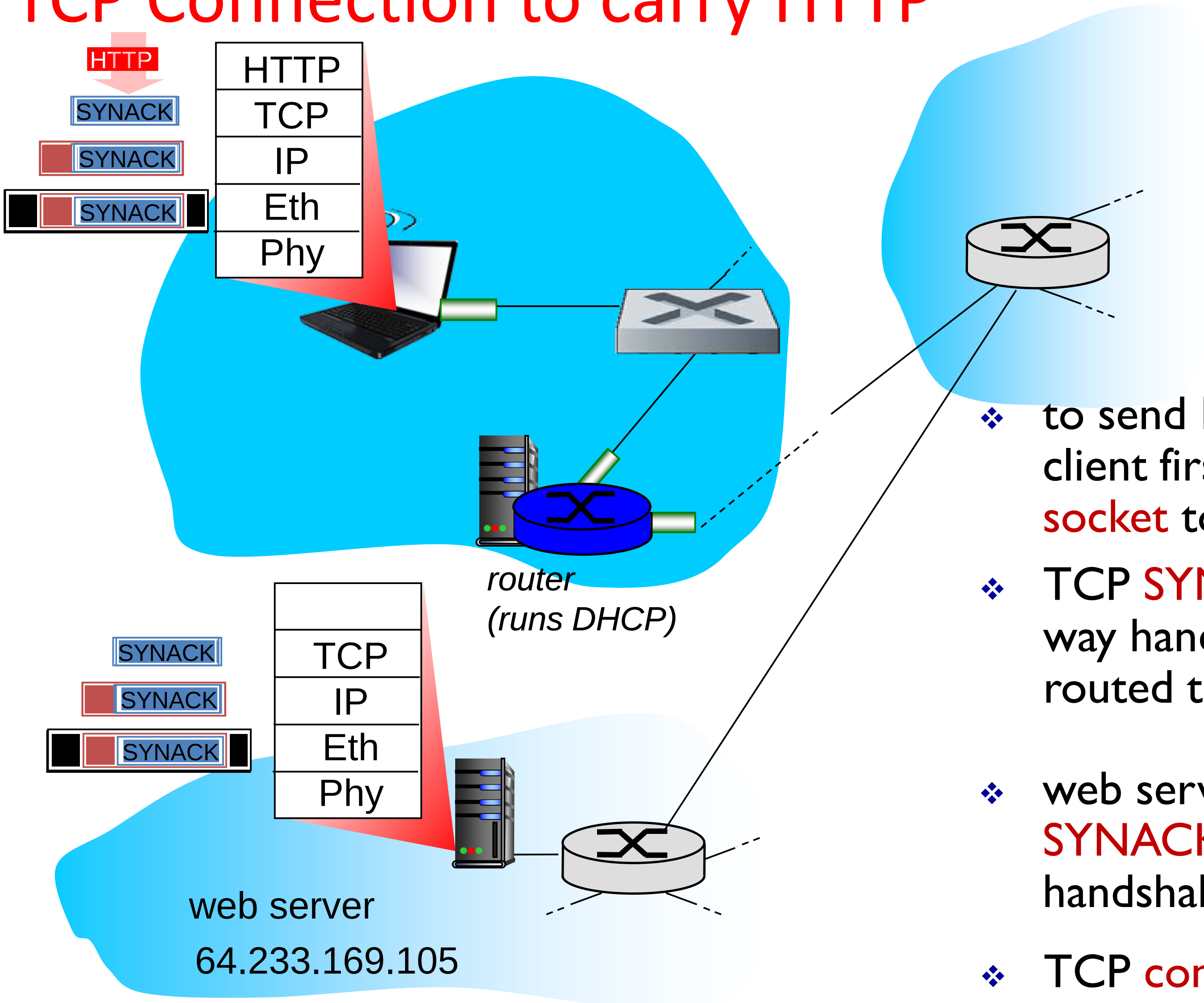
Using DNS



- 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 (tables created by RIP, OSPF, IS-IS and/or BGP routing protocols) to DNS server
- demux'ed to DNS server
- DNS server replies to client with IP address of www.google.com

TCP Connection to carry HTTP

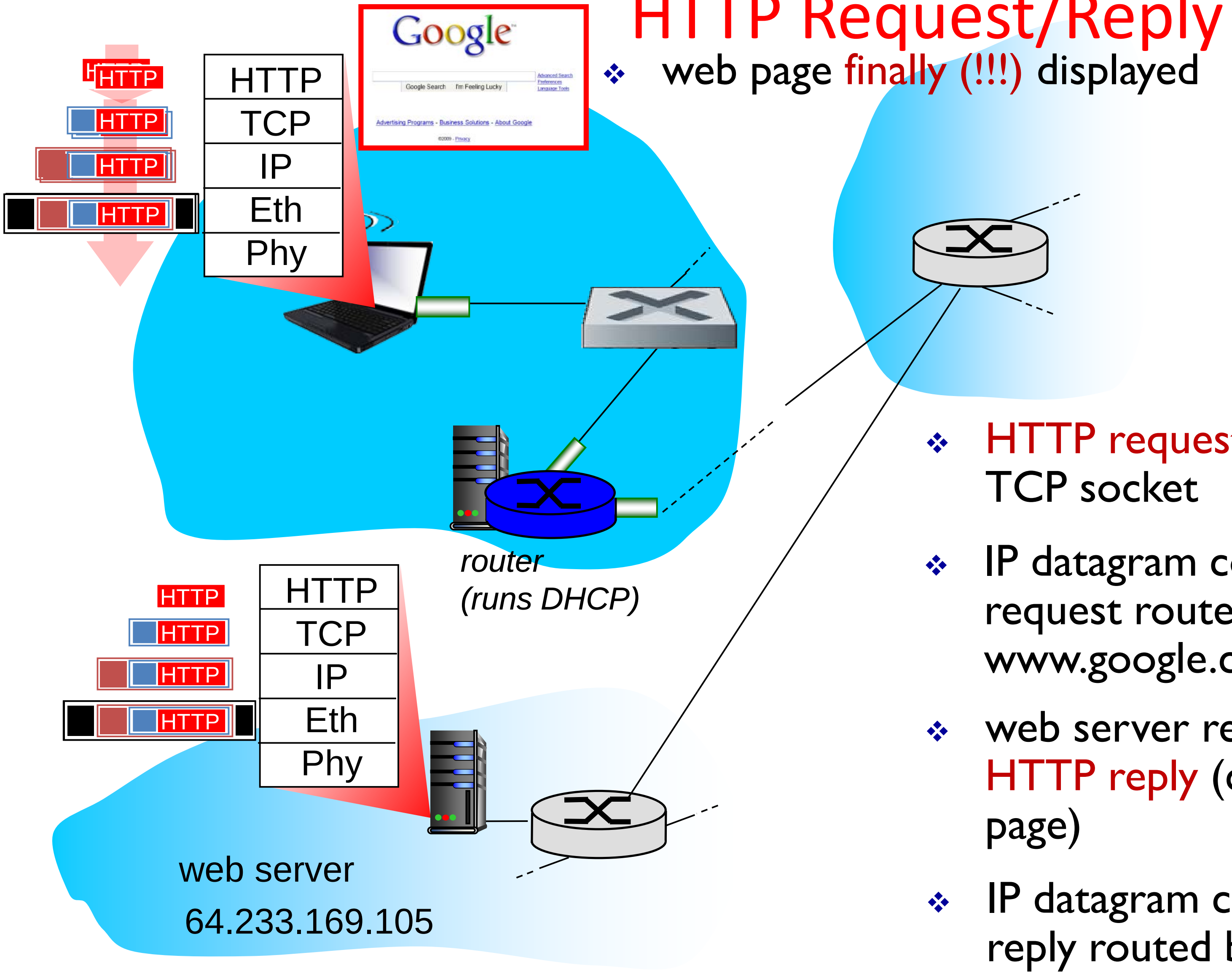


- to send HTTP request, client first opens TCP socket to web server
- TCP SYN segment (step 1 in 3-way handshake) inter-domain routed to web server
- web server responds with TCP SYNACK (step 2 in 3-way handshake)
- TCP connection established!

Putting it all together: A day and a life of a web req

HTTP Request/Reply

- web page finally (!!!) displayed



- HTTP request sent into 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