CS 455/555
Intro to Networks and Communications

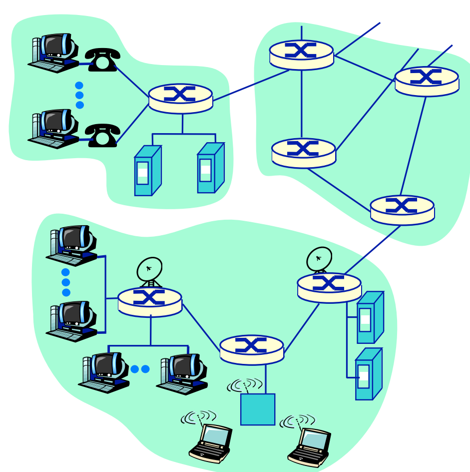
Link Layer Addressing, Ethernet, and a Day in the Life of a Web Request

Dr. Michele Weigle
Department of Computer Science
Old Dominion University
mweigle@cs.odu.edu

http://www.cs.odu.edu/~mweigle/CS455-S13/

Link Layer
Outline

- Introduction and services
- Error detection and correction
- Multiple access protocols
- Link-layer addressing
- Ethernet
- Link-Layer switches
  - » VLANs
- A Day in the Life…



Link-Layer Addressing
Ethernet MAC addresses

IP address:

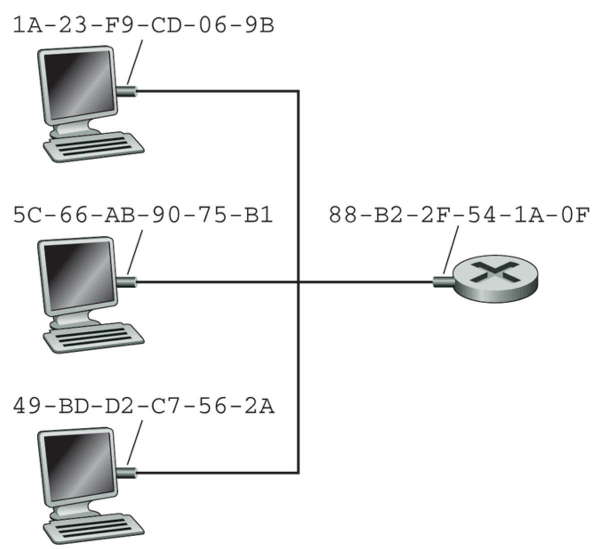
- *32-bit network-layer* address
- used to get datagram to destination network

Ethernet (*or MAC or physical*) address:

- used to get datagram from one interface to another physically-connected interface (same network)
- 48-bit MAC address burned in the adapter ROM

Link-Layer Addressing
Ethernet MAC addresses

- Each adapter has unique (in the universe) address
- Ethernet MAC address allocation administered by IEEE
  - » manufacturer buys portion of MAC address space (to assure uniqueness)
- MAC unstructured address => portability



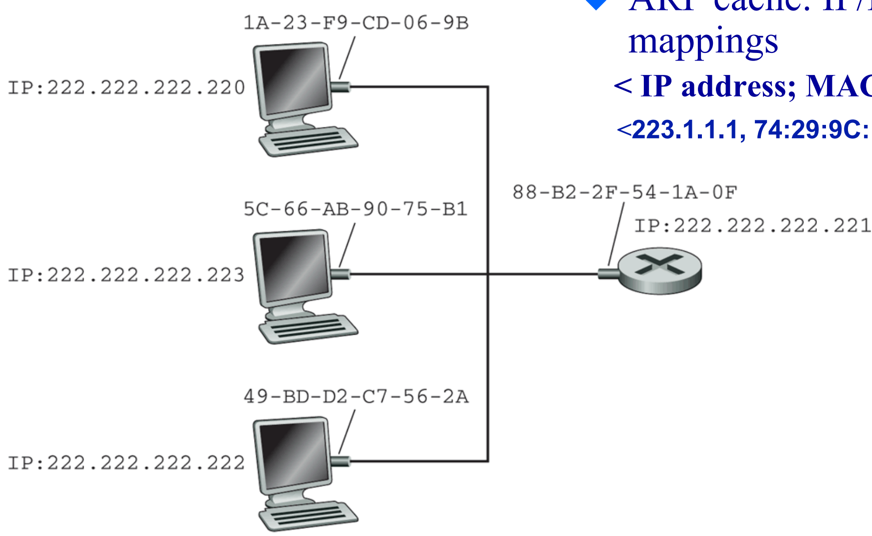
Link-Layer Addressing
ARP: Address Resolution Protocol

Question: how to determine MAC address of B given B's IP address?

- Each IP node (Host, Router) has ARP module & cache
- ARP cache: IP/MAC address mappings

< IP address; MAC address; TTL>

<223.1.1.1, 74:29:9C:E8:FF:55, 1200>



Link-Layer Addressing
ARP protocol

- Host A knows B's IP address, wants to learn the MAC address of B
- A *broadcasts* ARP query packet, containing B's IP address
  - » destination MAC address: FF-FF-FF-FF-FF-FF
  - » all hosts on Ethernet receive ARP query
- B receives ARP packet, replies to A with its (B's) MAC address
  - » frame sent to A's MAC address
- A caches IP-to-MAC address pairs until information becomes old
- *ARP is "plug-and-play"*
  - » *nodes create their ARP tables without human intervention*

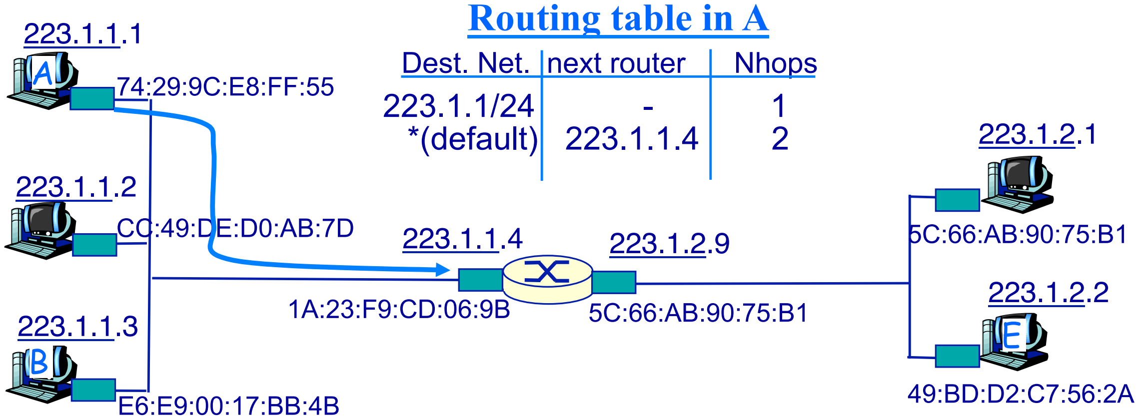
Routing IP Datagrams

Routing to a remote destination

misc fields	223.1.1.1	223.1.2.2	data

Routing table in A

Dest. Net.	next router	Nhops
223.1.1/24	-	1
*(default)	223.1.1.4	2



- A finds 223.1.1.4 in routing table as default router
- A uses ARP to find 1A:23:F9:CD:06:9B as Ethernet address for 223.1.1.4
- A creates Ethernet frame with 1A:23:F9:CD:06:9B as destination
  - » Ethernet frame contains IP datagram in data field
- A's link layer (adapter driver) sends Ethernet frame

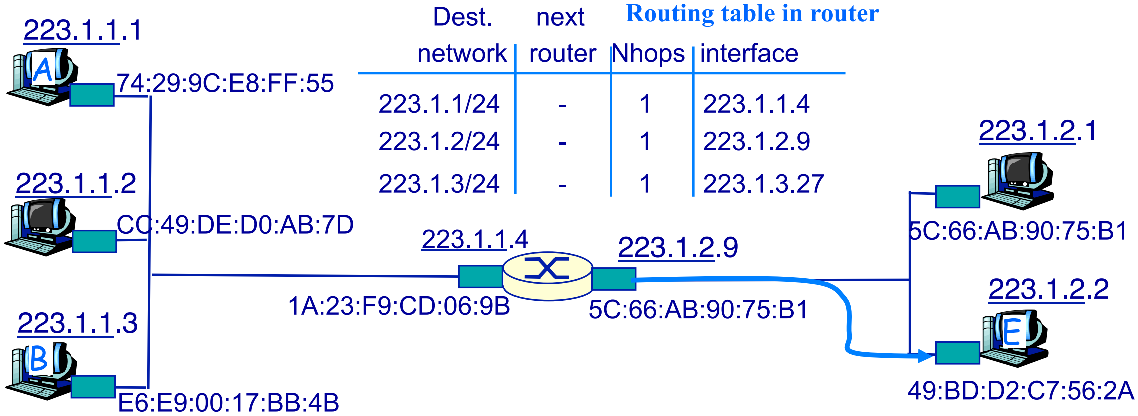
Routing IP Datagrams

Routing to a remote destination

misc fields	223.1.1.1	223.1.2.2	data

Routing table in router

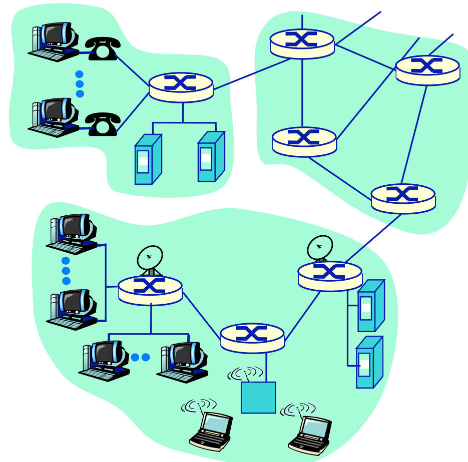
Dest. network	next router	Nhops	interface
223.1.1/24	-	1	223.1.1.4
223.1.2/24	-	1	223.1.2.9
223.1.3/24	-	1	223.1.3.27



- Router finds 223.1.2.9 in routing table as destination network interface
- Router uses ARP to find 49:BD:D2:C7:56:2A as Ethernet address for 223.1.2.2
- Router creates Ethernet frame with 49:BD:D2:C7:56:2A as destination;
  - » Ethernet frame contains IP datagram in data field
- Router's link layer (adapter driver) sends Ethernet frame

Link Layer
Outline

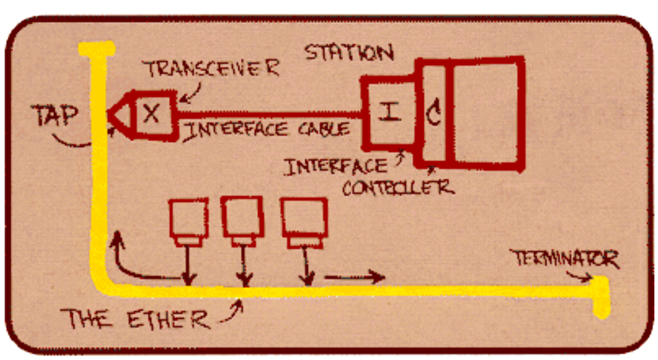
- Introduction and services
- Error detection and correction
- Multiple access protocols
- Link-layer addressing
- Ethernet
- Link-Layer switches
  - » VLANs
- A Day in the Life…



Link Layer
Ethernet (IEEE 802.3)

"Dominant" wired LAN technology:

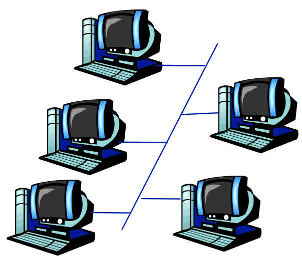
- cheap
  - » $20 for NIC
- first widely-used LAN technology
- simpler, cheaper than token LANs and ATM
- kept up with speed race: 10 Mbps – 10 Gbps

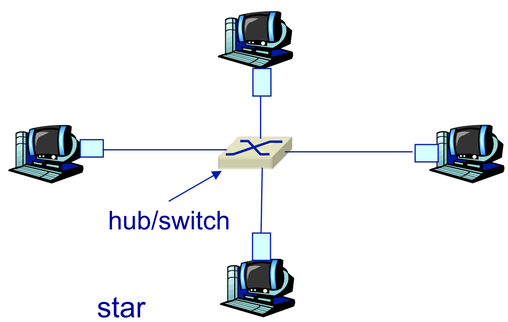


Metcalfe's Ethernet sketch

Ethernet
Star Topology

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

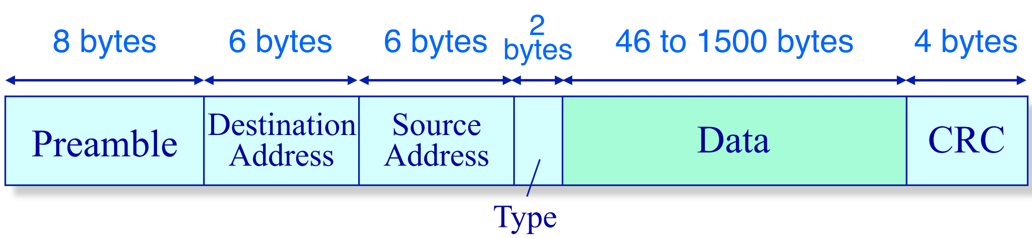

bus: coaxial cable



star

Ethernet
Frame Structure

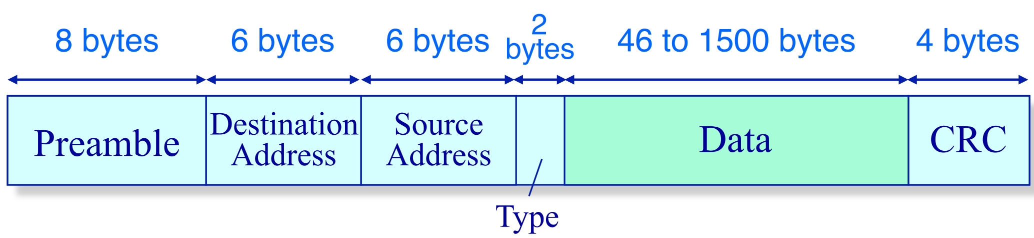
- Sending adapter encapsulates IP datagram (or other network layer protocol packet) in Ethernet frame
- Preamble
  - » 7 bytes with pattern 10101010 followed by one byte with pattern 10101011
  - » used to synchronize receiver, sender clock rates



Ethernet

Frame Structure

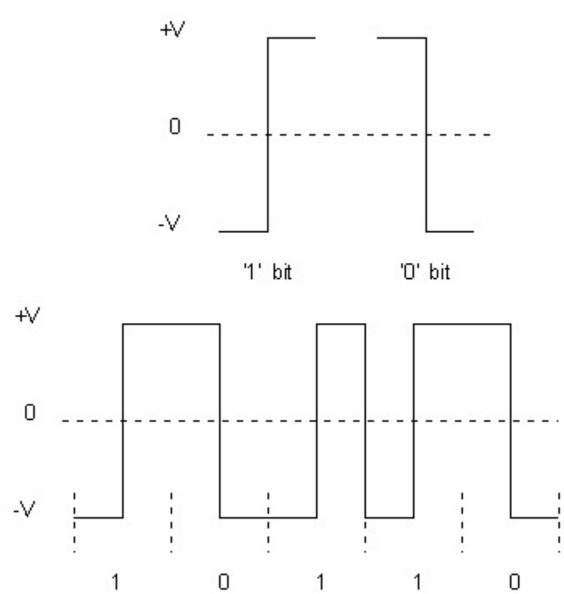
- Addresses: 6 bytes each
  - » if adapter receives frame with matching destination address, or with broadcast address (*e.g.* ARP packet), it passes data in frame to the network layer
  - » otherwise, adapter discards frame
- Type: indicates higher layer protocol (mostly IP, but others possible, *e.g.*, Novell IPX, AppleTalk)
- CRC: checked at receiver, if error is detected, frame is dropped



Ethernet

Manchester Encoding

- Bits encoded as voltage changes instead of fixed voltage
- 0 encoded with falling edge
- 1 encoded with rising edge
- Voltage change made exactly halfway through time slot
  - » need for *preamble*



Ethernet

Unreliable, connectionless

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

Ethernet

CSMA/CD algorithm

- NIC receives datagram from network layer, creates frame
- If NIC senses channel idle, starts frame transmission
  - » if channel busy, waits until channel idle, then transmits
- If NIC transmits entire frame without detecting another transmission, NIC is done with frame
- If NIC detects another transmission while transmitting, aborts and sends jam signal
- After aborting, NIC enters *exponential backoff*
  - » After m^{th} collision, NIC chooses K at random from $\{0,1,2,\ldots,2^m-1\}$
  - » NIC waits $K \cdot 512$ bit times, tries to send (sense channel)

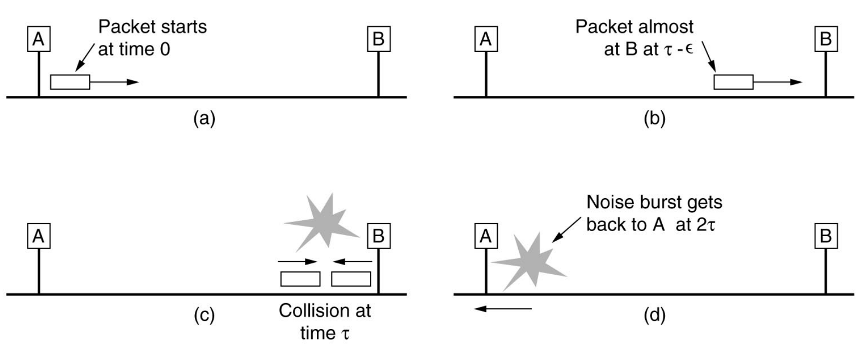
Ethernet
CSMA/CD algorithm (more)

- Jam Signal
  - » make sure all other transmitters are aware of collision
  - » 48 bits
- Bit time
  - » 0.1 µs for 10 Mbps Ethernet
  - » for K=1023, wait time is about 50 ms

<u>Exponential Backoff</u>

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

Ethernet
Collision Detection



- 10 Mbps Ethernet has max length of 2500 meters
  - » worst-case RTT about 50 µs (microseconds)
- Min frame must take at least worst-case RTT to transmit
  - » at 10 Mbps, can transmit 500 bits in 50 µs
  - » number rounded to 512 bits (64 bytes)

CSMA/CD

Efficiency

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

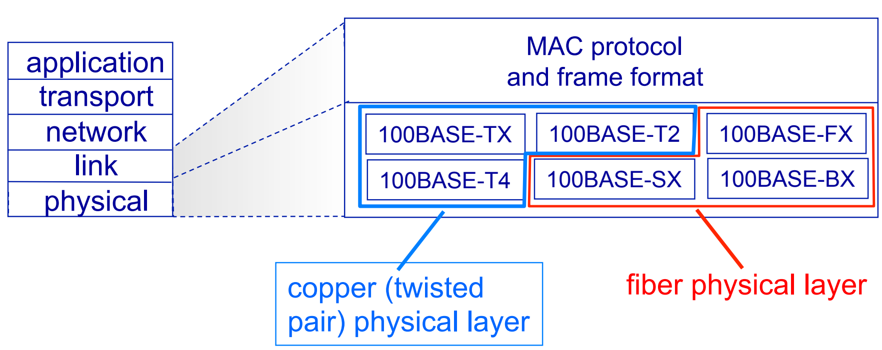
$$efficiency = \frac{1}{1 + 5t_{prop}/t_{trans}}$$

- Efficiency goes to 1
  - » as t_{prop} goes to 0
  - » as t_{trans} goes to infinity
- Better performance than ALOHA
- Simple, cheap, decentralized

802.3 Ethernet

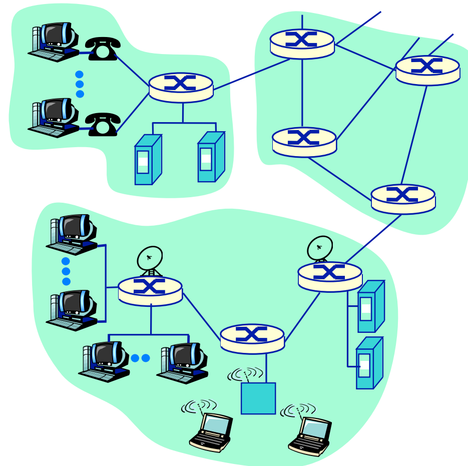
Standards: Link & Physical Layers

- Common MAC protocol and frame format
- Different speeds: 2 Mbps, 10 Mbps, 100 Mbps, 1 Gbps, 10 Gbps
- Different physical layer media: fiber, cable



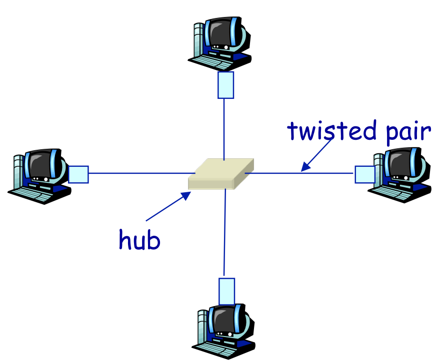
Link Layer
Outline

- Introduction and services
- Error detection and correction
- Multiple access protocols
- Link-layer addressing
- Ethernet
- Link-Layer switches
  - » VLANs
- A Day in the Life…



Link Layer
Hubs

- Physical-layer ("dumb") repeaters:
  - » bits coming in one link go out all other links at same rate
  - » all nodes connected to hub can collide with one another
  - » no frame buffering
  - » no CSMA/CD at hub: host NICs detect collisions

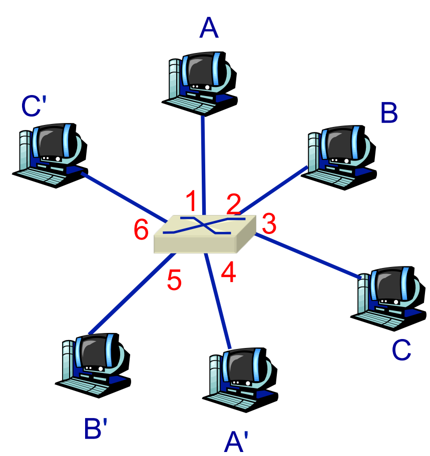


Link Layer
Switch

- Link-layer device: smarter than hubs, take active role
  - » store, forward Ethernet frames
  - » examine incoming frame's MAC address, *selectively* forward frame to one-or-more outgoing links when frame is to be forwarded on segment, uses CSMA/CD to access segment
- Transparent
  - » hosts are unaware of presence of switches
- Plug-and-play, self-learning
  - » switches do not need to be configured

Switches
Allows *multiple* simultaneous transmissions

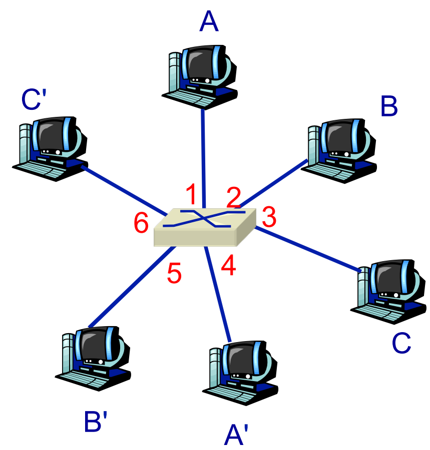
- Hosts have dedicated, direct connection to switch
- Switches buffer packets
- Ethernet protocol used on *each* incoming link, but no collisions; full duplex
  - » each link is its own collision domain
- *switching:* A-to-A' and B-to-B' simultaneously, without collisions
  - » not possible with dumb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



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

Switches
Table

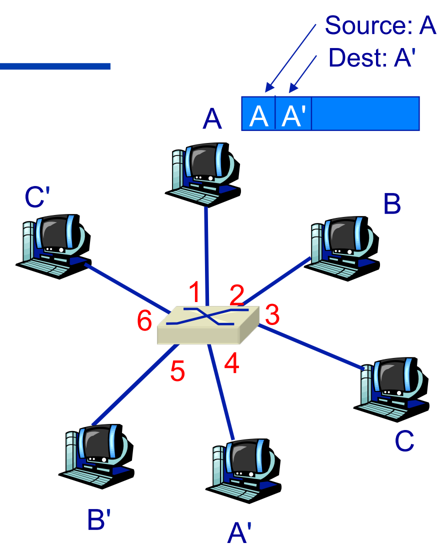
- How does switch know that A' reachable via interface 4 and B' reachable via interface 5?
  - » each switch has a *switch table*
  - » each entry contains
    - MAC address of host
    - interface to reach host
    - timestamp
- How are entries created, maintained in switch table?
  - » something like a routing protocol?



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

Switches
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)

Switches
Frame filtering/forwarding

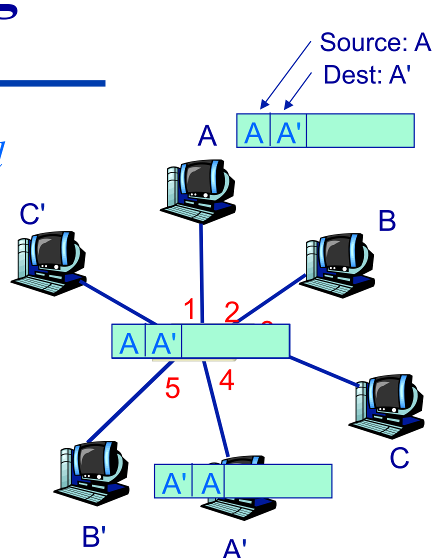
When frame received:

1. record link associated with sending host
2. index switch table using MAC dest address
3. **if** entry found for destination
   then {
   if dest on segment from which frame arrived
   then drop the frame
   else forward the frame on interface indicated
   }
   else flood

forward on all but the interface on which the frame arrived

Self-learning, forwarding
Example

- Frame destination unknown: *flood*
- Destination A location known: *selective send*

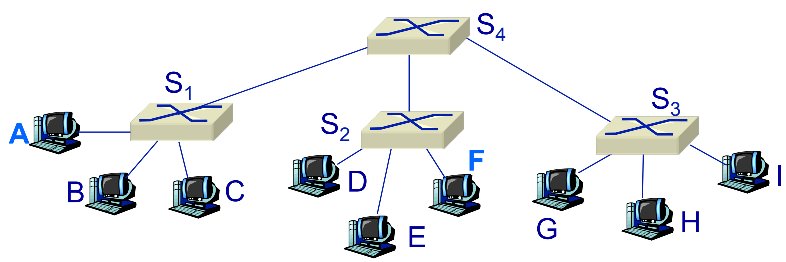


MAC addr	interface	TTL
A	*1*	*60*
A'	*4*	*60*

Switch table (initially empty)

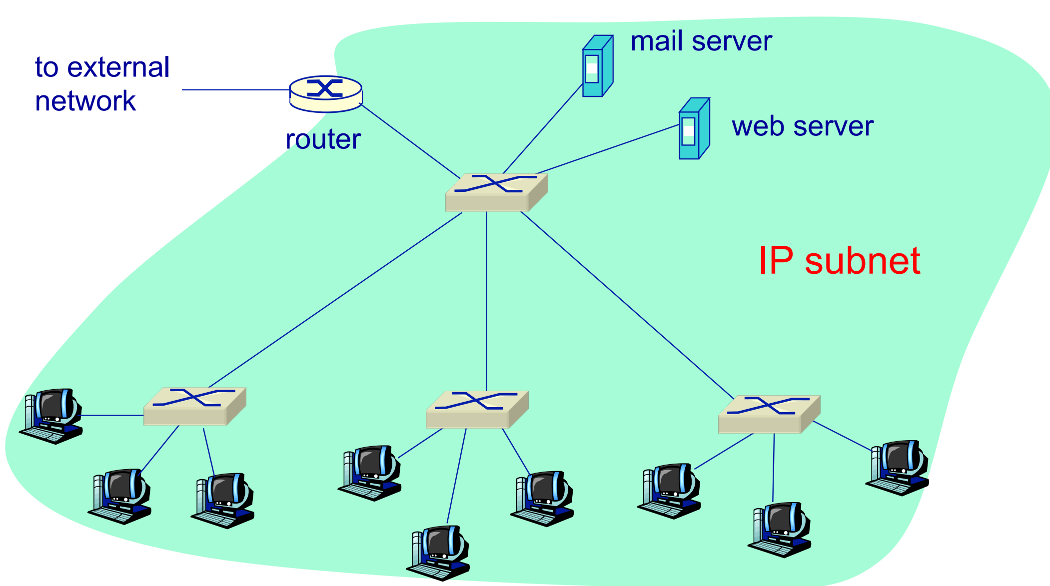
Switches
Interconnecting switches

- Switches can be connected together



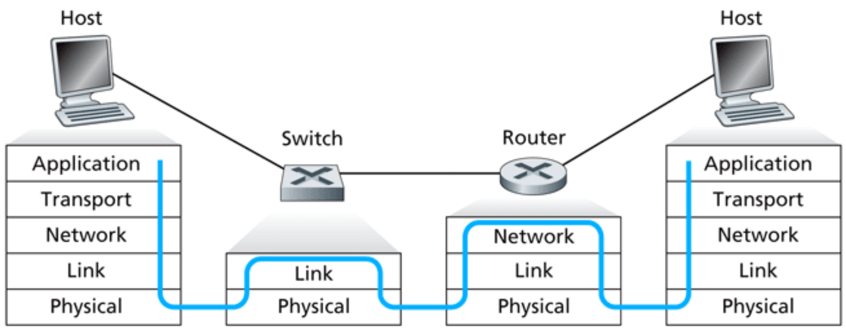
- Sending from A to F - how does S_1 know to forward frame destined to F via S_4 and S_2?
  - » Self learning! (works exactly the same as in single-switch case!)

Switches
Institutional network



Switches vs. Routers

- Both store-and-forward devices
  - » routers: network layer devices (examine network layer headers)
  - » switches are link layer devices
- Routers maintain routing tables, implement routing algorithms
- Switches maintain switch tables, implement filtering, learning algorithms

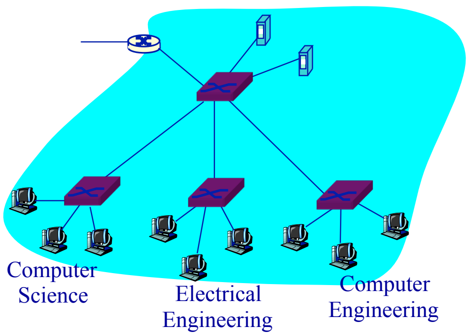


Hubs vs. Switches vs. Routers

- Hubs
  - » physical layer devices
  - » connected devices share bandwidth
    - ❖ all hosts / networks connected to a 10 Mbps hub share 10 Mbps
  - » all connections must be same physical layer speed
- Routers
  - » network layer devices (Layer 3)
  - » MAC address and IP address
  - » run routing algorithm for forwarding
  - » forwards based on IP destination address
- Switches
  - » link layer devices (Layer 2)
  - » connected devices do not share bandwidth
    - ❖ each host / network connected to a 10 Mbps switch gets 10 Mbps
  - » can handle different physical layer speeds
  - » no MAC address
  - » forwards based on MAC destination address

Link Layer
Virtual LANs (VLANs)

What's wrong with this picture?



What happens if:
- CS user moves office to EE, but wants connect to CS switch?

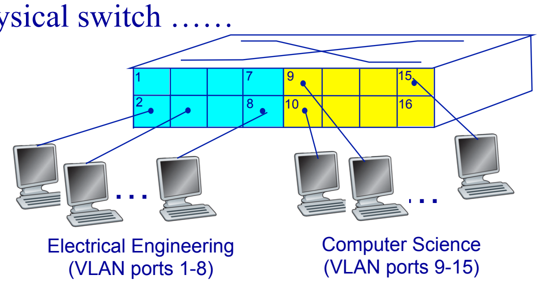
Single broadcast domain:
- all layer-2 broadcast traffic (ARP, DHCP) crosses entire LAN (security/privacy, efficiency issues)
- each lowest level switch has only few ports in use

VLANs

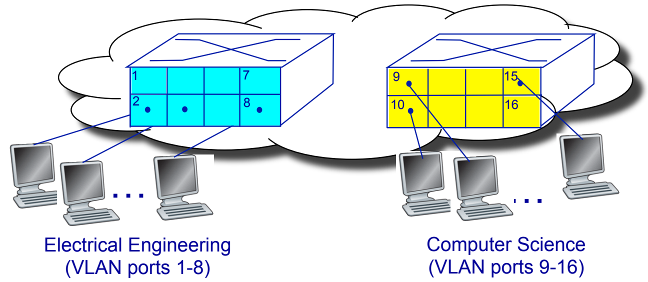
Virtual LAN

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 ……

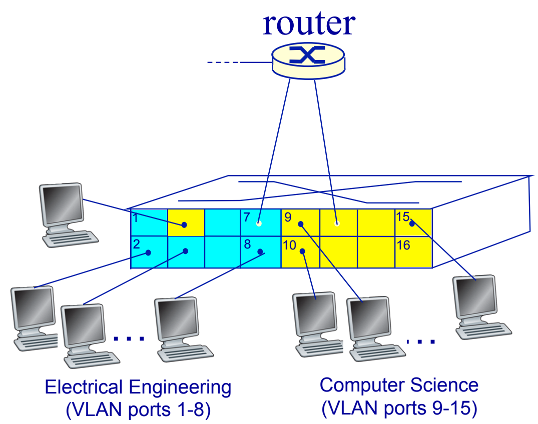


… operates as multiple virtual switches

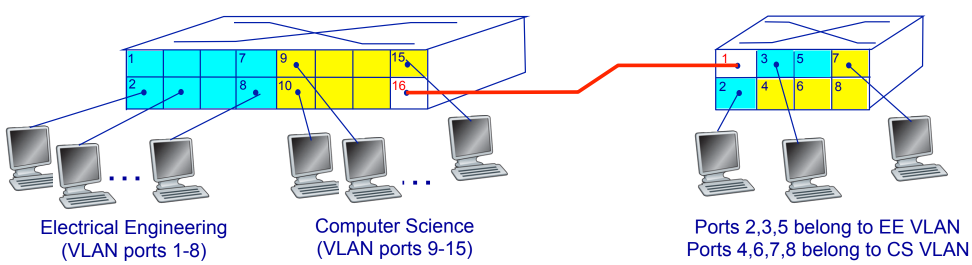


Port-based VLAN

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



VLANs
Spanning multiple switches



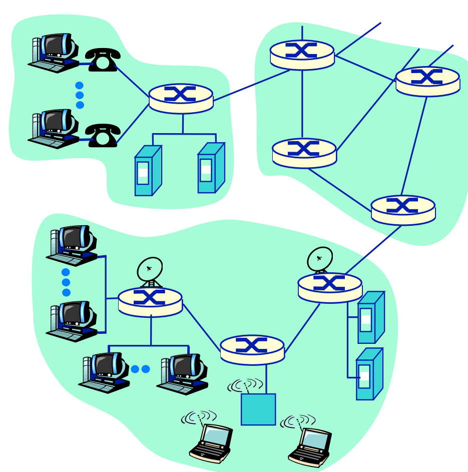
- *trunk port:* carries frames between VLANS defined over multiple physical switches
  - » frames forwarded within VLAN between switches can't be vanilla 802.1 frames (must carry VLAN ID info)
  - » 802.1q protocol adds/removed additional header fields for frames forwarded between trunk ports

Chapter 5: 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
  - » VLANs

Link Layer
Outline

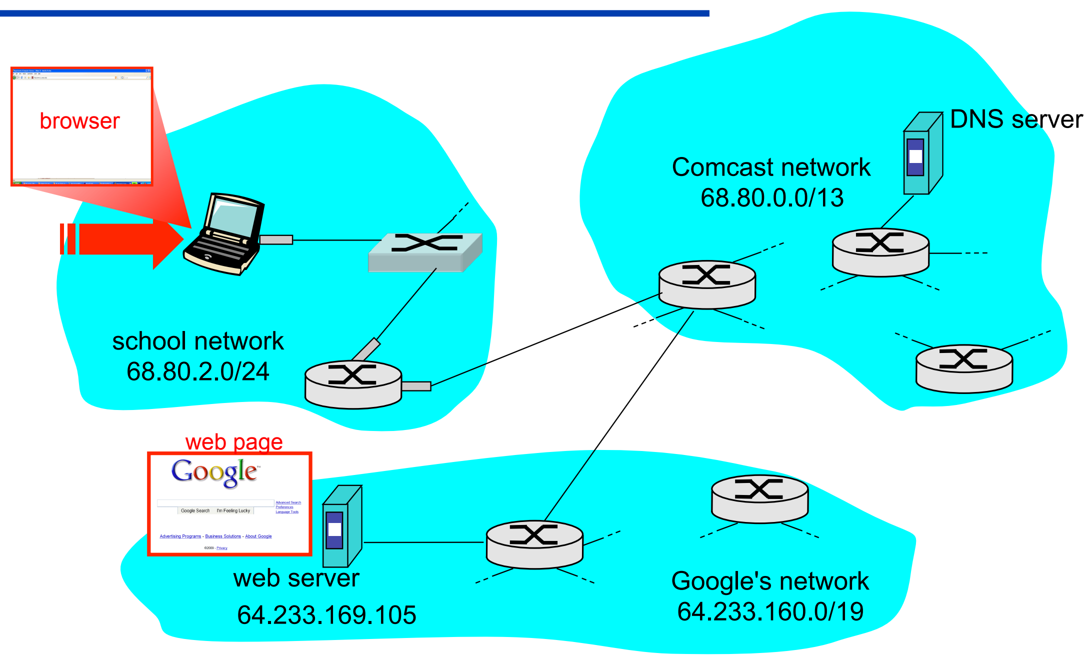
- Introduction and services
- Error detection and correction
- Multiple access protocols
- Link-layer addressing
- Ethernet
- Link-Layer switches
  - » VLANs
- A Day in the Life…



Synthesis
A day in the life of a web request

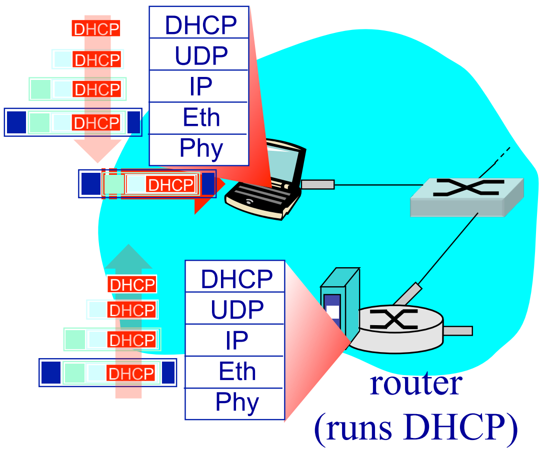
- Our journey down the protocol stack is complete!
  - » application, transport, network, link
- Let's put it all together
  - » Goal: identify, review, understand protocols (at all layers) involved in seemingly simple scenario: requesting www page
  - » Scenario: student attaches laptop to campus network, requests/receives www.google.com

A Day in the Life…
Scenario



A Day in the Life…

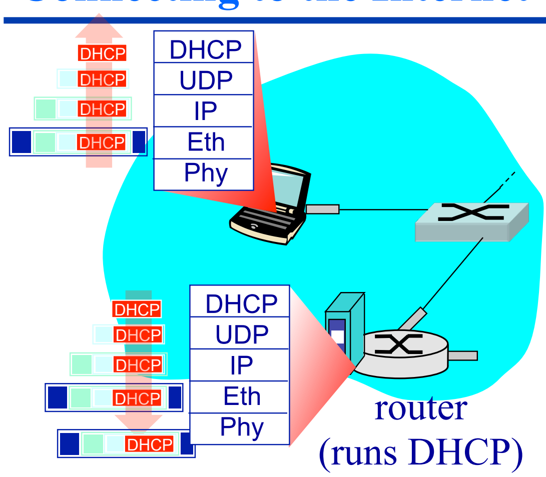
Connecting to the Internet



- 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 **802.1** Ethernet
  - Ethernet frame **broadcast** (dest: FFFFFFFFFFFF) on LAN, received at router running **DHCP** server
  - Ethernet **demux'ed** to IP demux'ed to UDP demux'ed to DHCP

A Day in the Life…

Connecting to the Internet

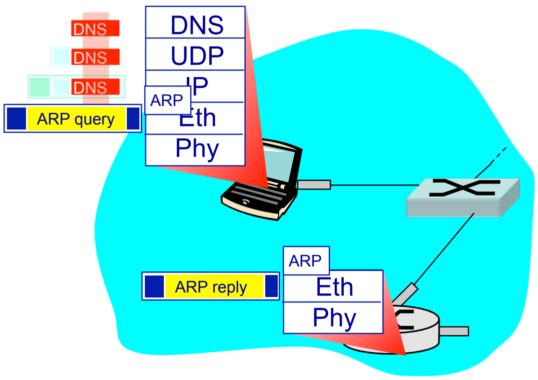


- DHCP server formulates DHCP ACK containing client's IP address, IP address of first-hop router for client, name & IP address of DNS server
  - encapsulation at DHCP server, frame forwarded (**switch learning**) through LAN, demultiplexing at client
  - DHCP client receives DHCP ACK reply

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

A Day in the Life…

ARP (before DNS, before HTTP)



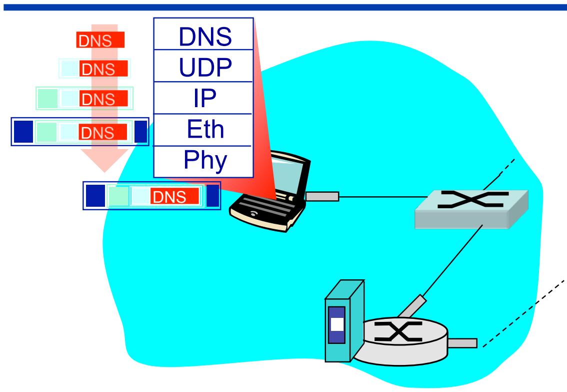
- Before sending HTTP request, need IP address of www.google.com: DNS

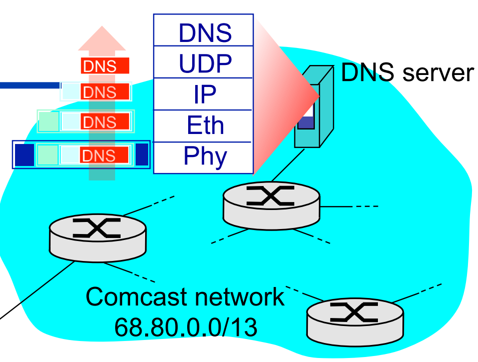
- DNS query created, encapsulated in UDP, encapsulated in IP, encapsulated in Eth. In order to send frame to router, need MAC address of router interface: **ARP**
- **ARP query** broadcast, received by router, which replies with **ARP reply** giving MAC address of router interface

Client now knows MAC address of first hop router, so can now send frame containing DNS query

A Day in the Life…

Using DNS



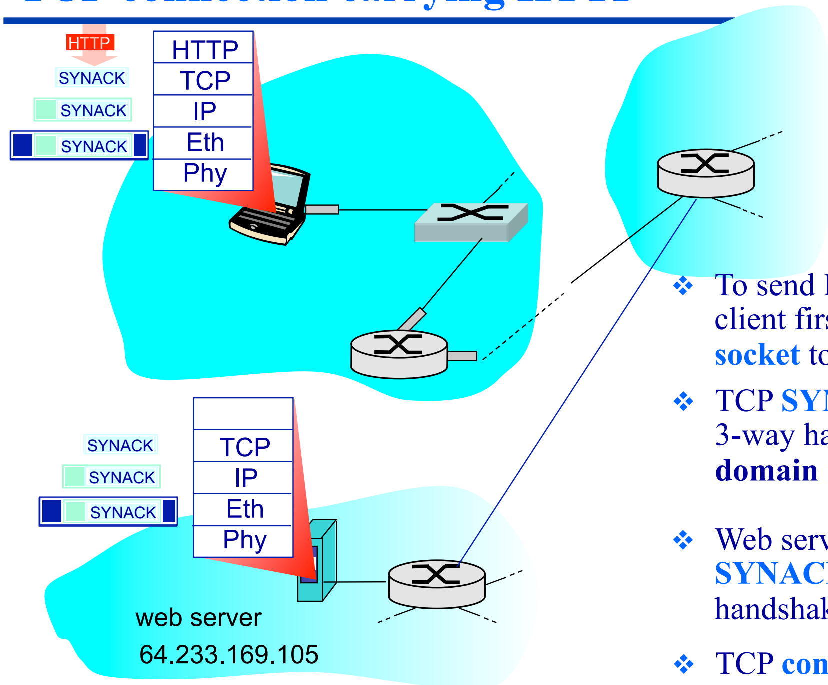


- IP datagram containing DNS query forwarded via LAN switch from client to 1st hop router

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

A Day in the Life…

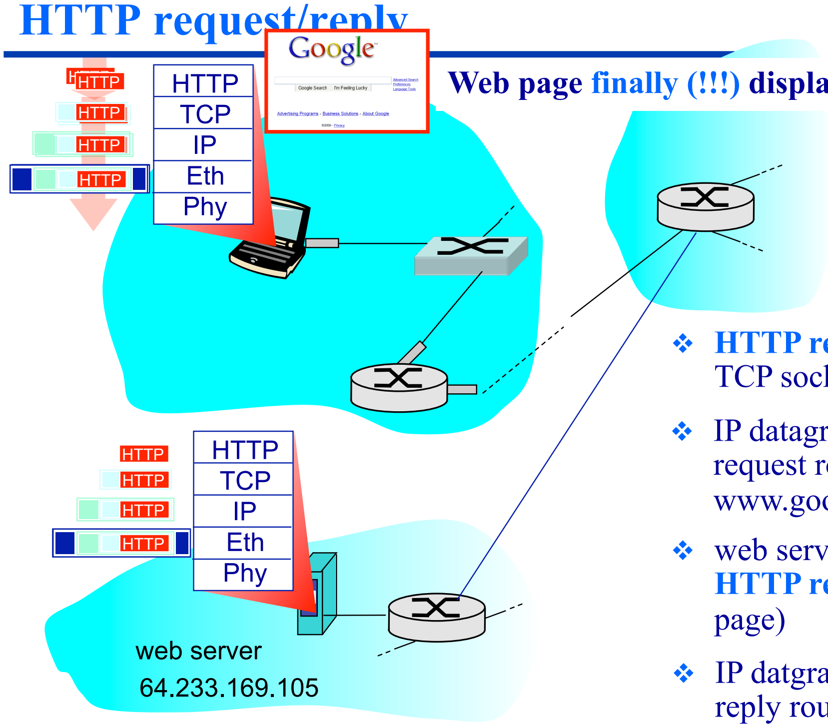
TCP connection carrying HTTP



- 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



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 datgram containing HTTP reply routed back to client