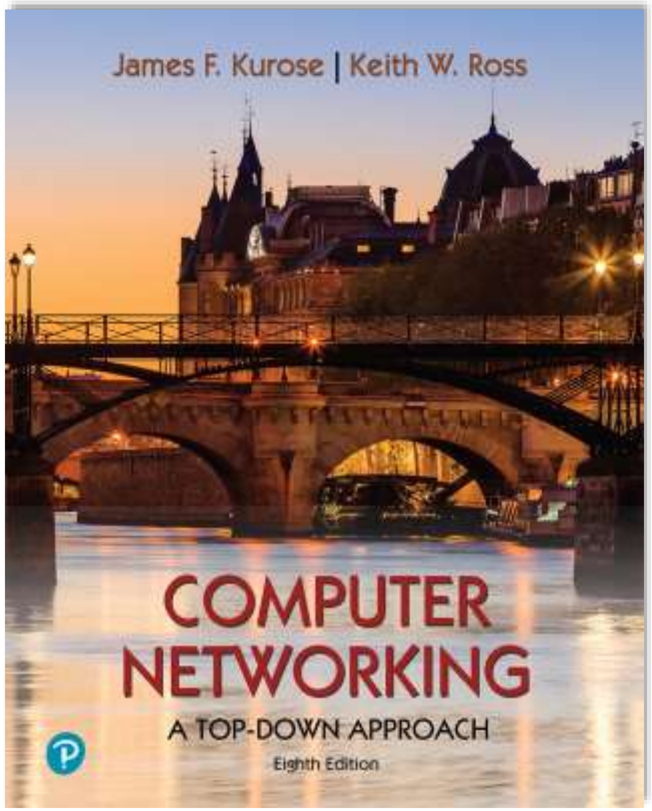
Chapter 6
The Link Layer and LANs

A note on the use of these PowerPoint slides:
We're making these slides freely available to all (faculty, students, readers). They're in PowerPoint form so you see the animations; and can add, modify, and delete slides (including this one) and slide content to suit your needs. They obviously represent a *lot* of work on our part. In return for use, we only ask the following:

- If you use these slides (e.g., in a class) that you mention their source (after all, we'd like people to use our book!)
- If you post any slides on a www site, that you note that they are adapted from (or perhaps identical to) our slides, and note our copyright of this material.

For a revision history, see the slide note for this page.

Thanks and enjoy! JFK/KWR



Computer Networking: A Top-Down Approach
8th edition
Jim Kurose, Keith Ross
Pearson, 2020

Link layer and LANs: our goals

- understand principles behind link layer services:
  - error detection, correction
  - sharing a broadcast channel: multiple access
  - link layer addressing
  - local area networks: Ethernet, VLANs
- datacenter networks

- instantiation, implementation of various link layer technologies

Link layer, LANs: roadmap

- introduction
- error detection, correction
- multiple access protocols
- LANs
  - addressing, ARP
  - Ethernet
  - switches
  - VLANs
- link virtualization: MPLS
- data center networking

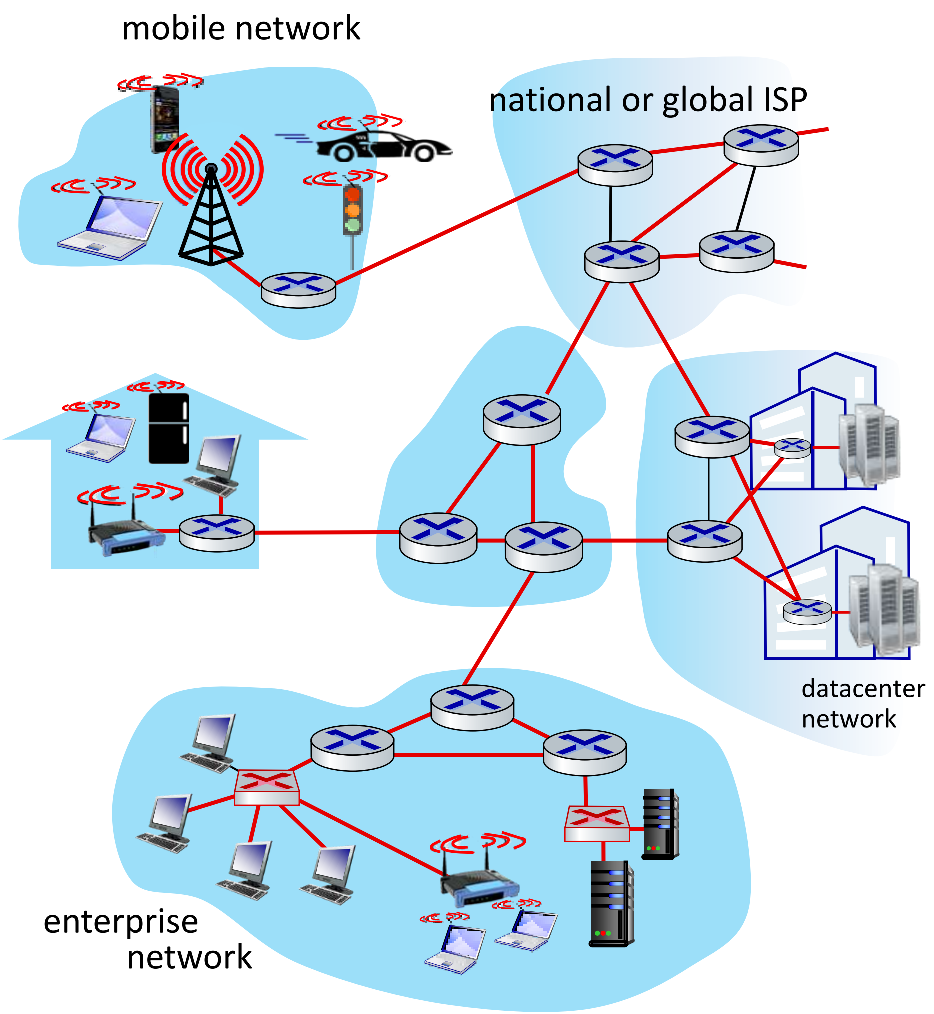
- a day in the life of a web request

Link layer: introduction

terminology:

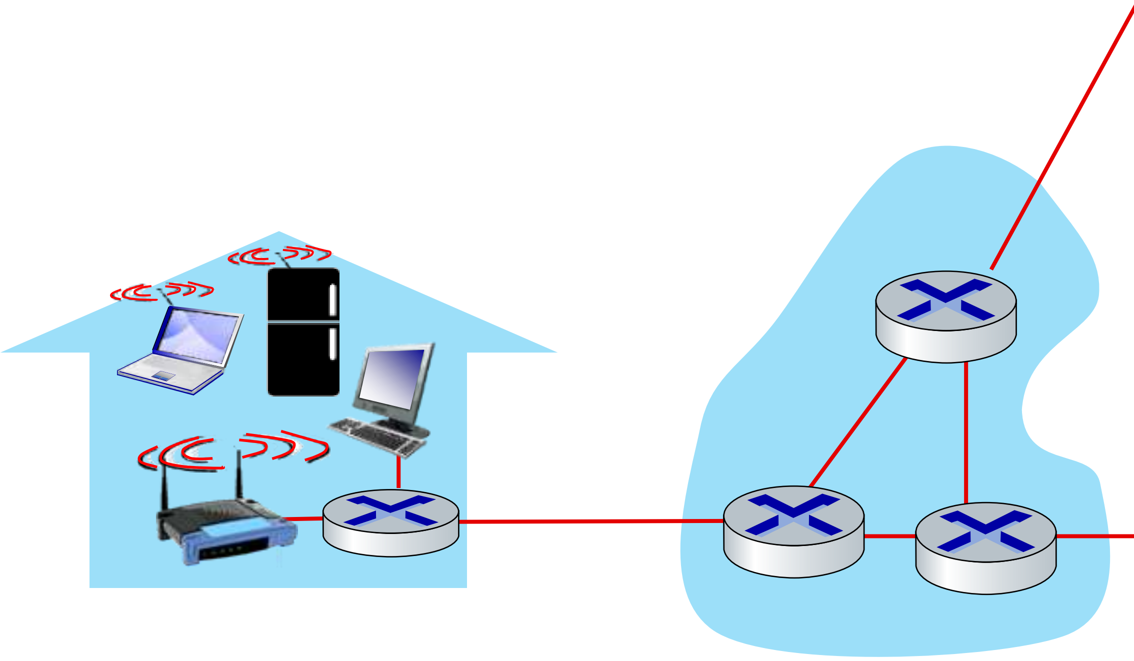
- Nodes – any device that runs a link-layer (hosts, routers, switches, wifi).
- communication channels that connect adjacent nodes along communication path: links
  - wired , wireless
  - LANs
- layer-2 packet: *frame*, encapsulates datagram

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



Link layer: context

- datagram transferred by different link protocols over different links:
  - e.g., WiFi on first link, Ethernet on next link
- each link protocol provides different services
  - e.g., may or may not provide reliable data transfer over link

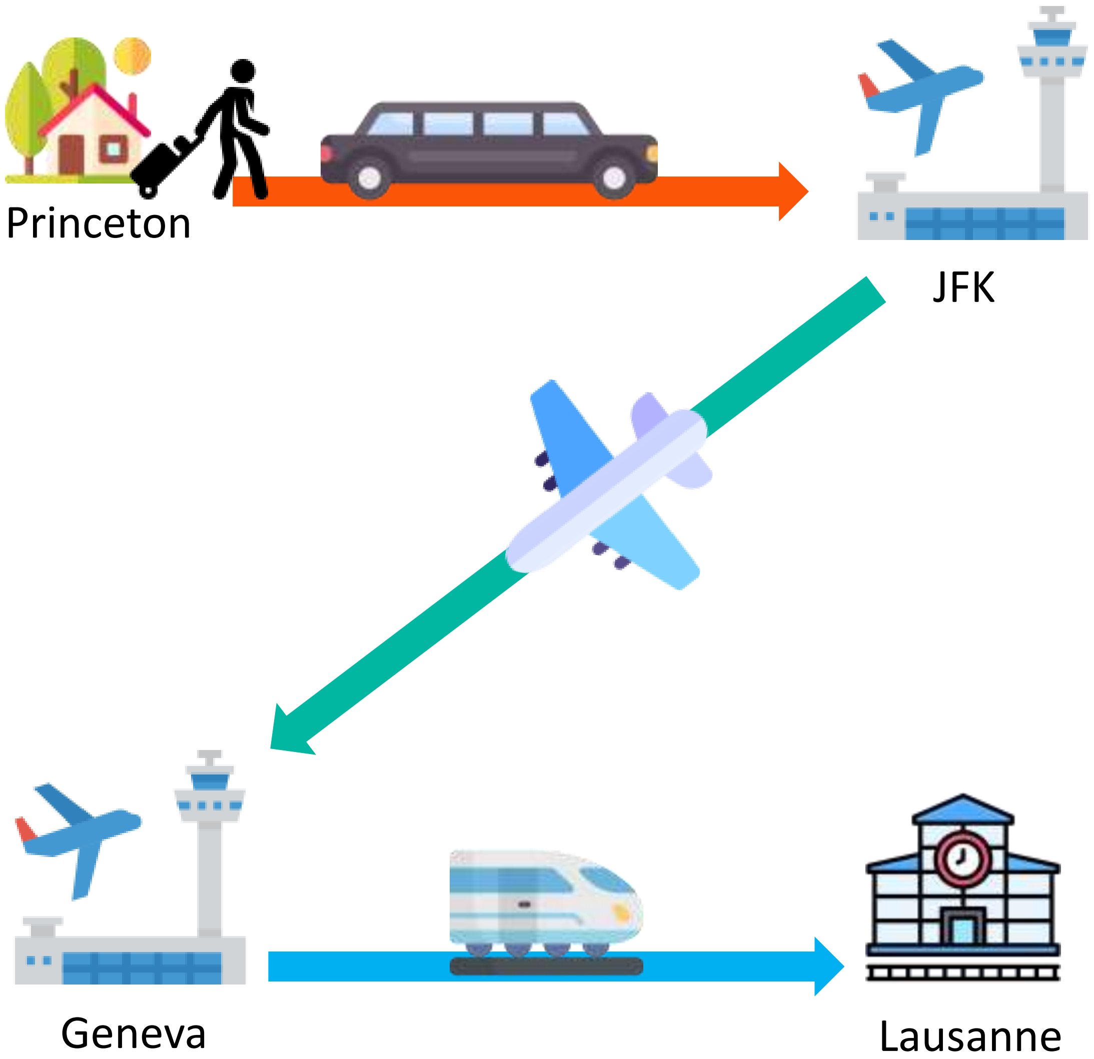


Link Layer Types

Two different types of link-layer channels:

- First type are broadcast channels, which connect multiple hosts in wireless LAN, satellite and hybrid fiber-coaxial cable
  - Many hosts are connected to the same broadcast communication channel, medium access protocol is needed to coordinate frame transmission.
    - Central controller or hosts themselves coordinate the transmission
- Second type of link-layer channel is the point-to-point communication link; office computer to nearby Ethernet.
  - Coordinating access to a point-to-point link is simpler; Point-to-Point Protocol (PPP), which is used in settings ranging from dial-up service over a telephone line to high-speed point-to-point frame transport over fiber-optic links.

Transportation analogy

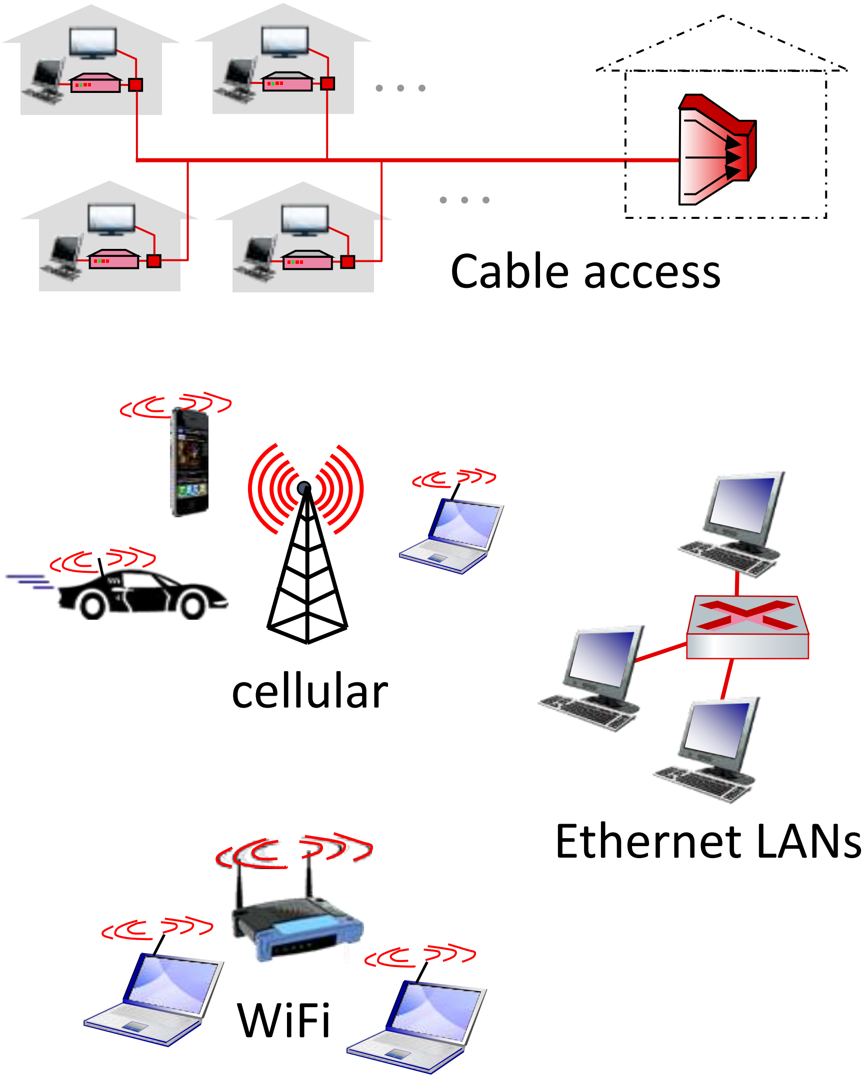


transportation analogy:

- trip from Princeton to Switzerland
  - 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:**
  - encapsulate datagram into frame before transmission over the link
- **Link access:**
  - Medium Access Control (MAC) addresses in frame headers identify source, destination (different from IP address!)
- **Reliable Delivery:**
  - When a link-layer protocol provides reliable delivery service, it guarantees to move each network-layer datagram across the link without error.



Link Layer: Services

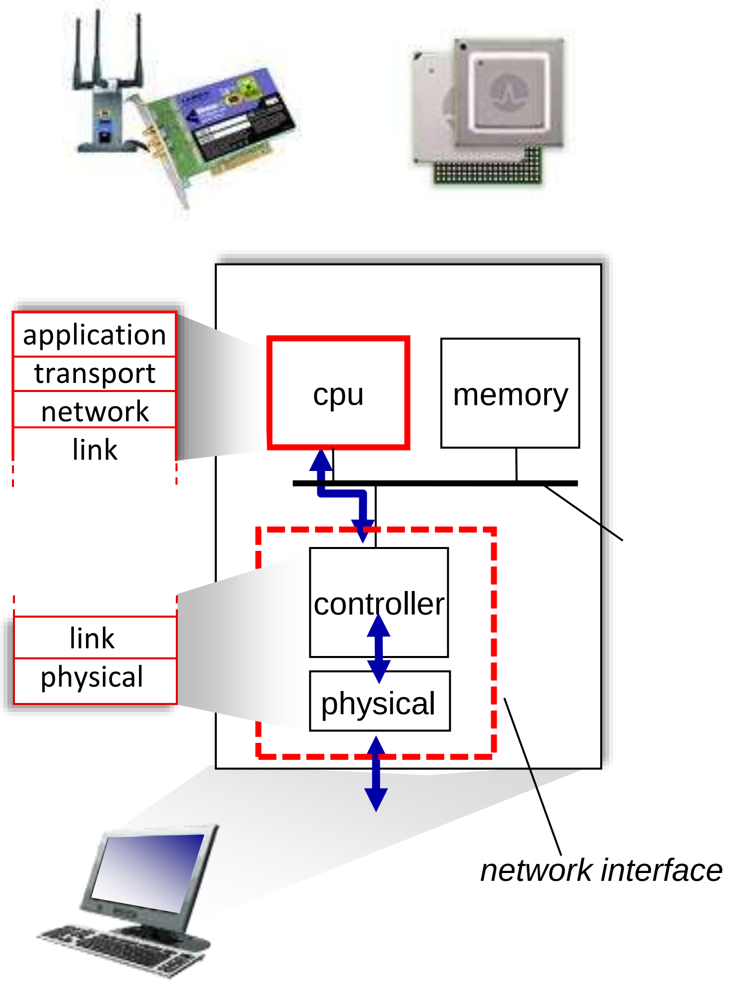
- Reliable Delivery (continued):
- A link-layer reliable delivery service is often used for links that are prone to high error rates, such as a wireless link, with the goal of correcting an error locally—on the link where the error occurs—rather than forcing an end-to-end retransmission of the data by a transport- or application-layer protocol.
- However, link-layer reliable delivery can be considered an unnecessary overhead for low bit-error links, including fiber, coax, and many twisted-pair copper links. For this reason, many wired link-layer protocols do not provide a reliable delivery service.

Link Layer: Services

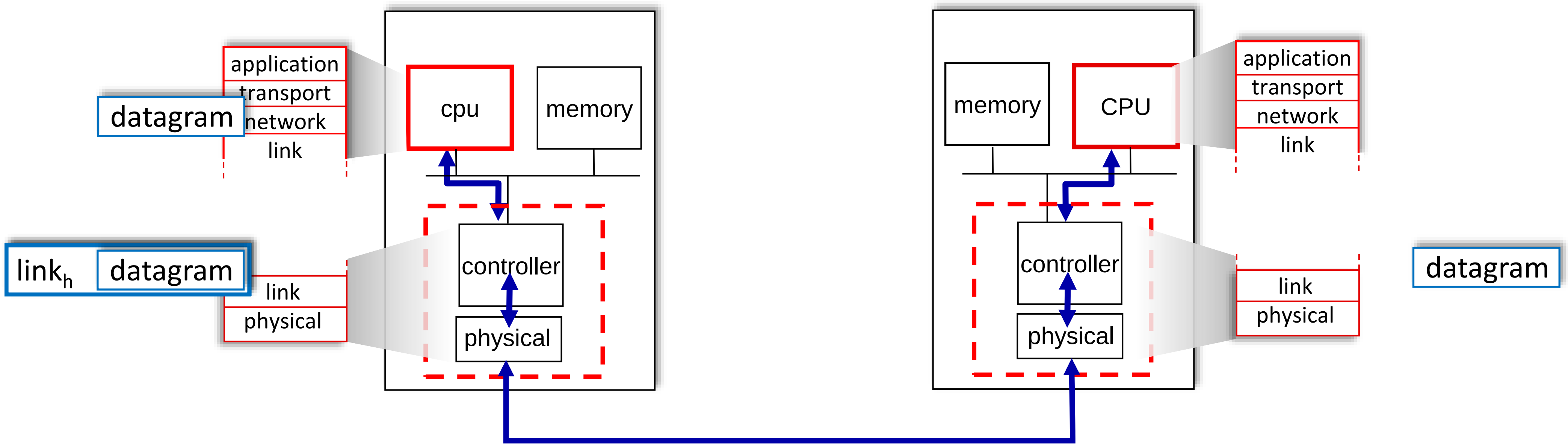
- Error Detection and Correction:
- The link-layer hardware in a receiving node can incorrectly decide that a bit in a frame is zero when it was transmitted as a one, and vice versa. Such bit errors are introduced by signal attenuation and electromagnetic noise
  - Many link-layer protocols provide a mechanism to detect such bit errors.
  - This is done by having the transmitting node include error-detection bits in the frame, and having the receiving node perform an error check
- Network layer provide limited error detection - checksum; transport layer (TCP) provides some detection and correction services
- Error detection in the link layer is usually more sophisticated and is implemented in hardware. Receiver not only detects when bit errors have occurred in the frame but also determines exactly where in the frame the errors have occurred and then corrects these errors.

Host link-layer implementation

- in each-and-every host; most of link layer is implemented in hardware; part of it is implemented in software that runs on the host's CPU.
  - Software: activating controller's hardware
  - Handling error conditions
  - Passing datagram up to the network layer
- link layer implemented on-chip or in network interface card (NIC)
  - implements link, physical layer
- attaches into host's system buses
- combination of hardware, software, firmware



Interfaces communicating



sending side:

- Controller encapsulates datagram into frame
- Transmits the frame into communication link
- adds error checking bits, reliable data transfer, flow control, etc.

receiving side:

- Controller receives the entire frame, extracts network layer datagram
- If link layer performs error detection, then the receiving controller performs error detection and correction

Link layer, LANs: roadmap

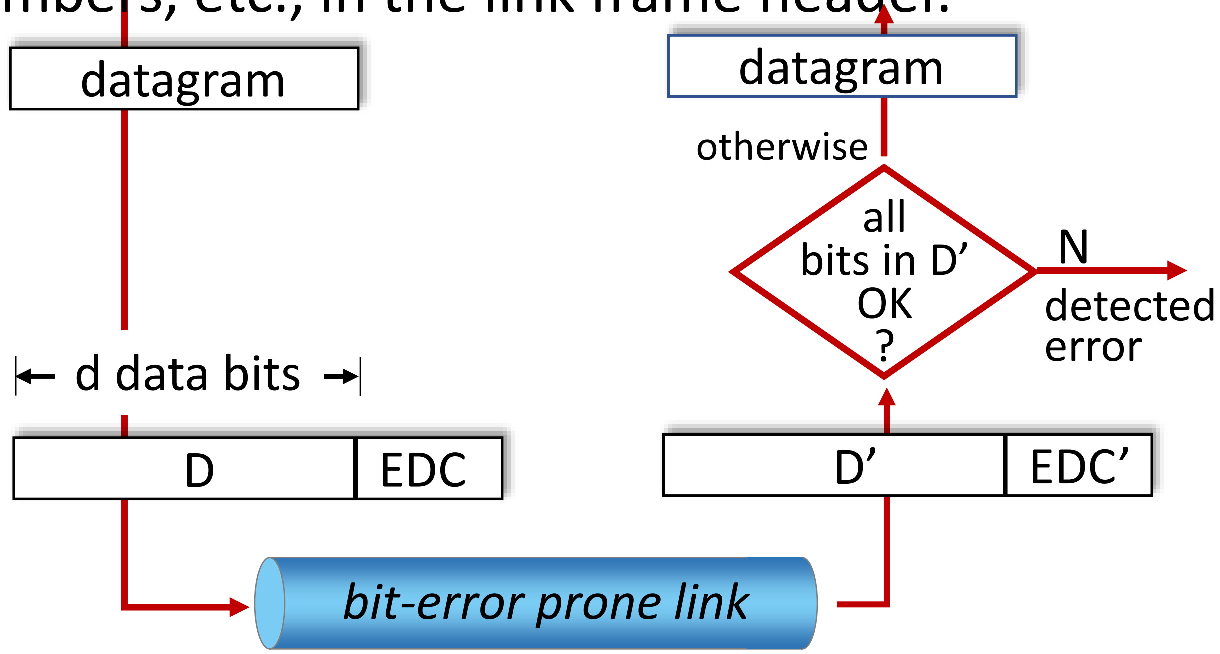
- introduction
- error detection, correction
- multiple access protocols
- LANs
  - addressing, ARP
  - Ethernet
  - switches
  - VLANs
- link virtualization: MPLS
- data center networking

- a day in the life of a web request

Bit-level error detection

Sending node: data, D, is augmented with error detection and correction bits (EDC)

Not only around datagram, but link-level addressing info, sequencing numbers, etc., in the link frame header.



Error detection not 100% reliable!

- protocol may miss some errors, but rarely
- larger EDC field yields better detection and correction

Parity checking

single bit parity:

- detect single bit errors

0111000110101011	1

← *d* data bits →

parity bit

Even/odd parity: set parity bit so there is an even/odd number of 1’sdepending on the schema

At receiver:

- compute parity of *d* received bits
- compare with received parity bit – if different than error detected
- “bursts” of errors can occur

Can detect *and* correct errors (without retransmission!)

- **two-dimensional parity**: detect *and correct* single bit errors

row parity

column parity

$$
\begin{array}{cccc|c}
d_{1,1} & \cdots & d_{1,j} & & d_{1,j+1} \\
d_{2,1} & \cdots & d_{2,j} & & d_{2,j+1} \\
\cdots & \cdots & \cdots & & \cdots \\
d_{i,1} & \cdots & d_{i,j} & & d_{i,j+1} \\
\hline
d_{i+1,1} & \cdots & d_{i+1,j} & & d_{i+1,j+1}
\end{array}
$$

no errors:

$$
\begin{array}{ccccc|c}
1 & 0 & 1 & 0 & 1 & 1 \\
1 & 1 & 1 & 1 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 1 \\
\hline
1 & 0 & 1 & 0 & 1 & 0
\end{array}
$$

detected and correctable single-bit error:

$$
\begin{array}{ccccc|c}
1 & 0 & 1 & 0 & 1 & 1 \\
1 & 0 & 1 & 1 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 1 \\
\hline
1 & 0 & 1 & 0 & 1 & 0
\end{array}
$$

parity error

parity error

Internet checksum (review, see section 3.3)

Goal: detect errors (*i.e.*, flipped bits) in transmitted segment

sender:

- treat contents of UDP segment (including UDP header fields and IP addresses) as sequence of 16-bit integers
- checksum value put into UDP checksum field

receiver:

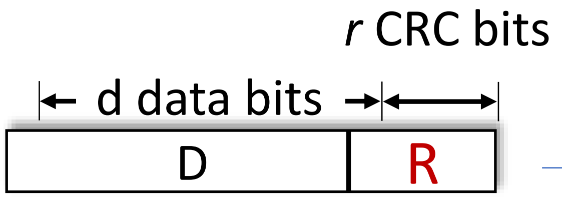
- compute checksum of received segment
- check if computed checksum equals checksum field value:
  - not equal - error detected
  - equal - no error detected. *But maybe errors nonetheless?* More later ….

Cyclic Redundancy Check (CRC)

- more powerful error-detection coding; **polynomial codes**
- Data bits (given, think of these as a binary number)
- Since it is possible to view the bit string to be sent as a polynomial whose coefficients are the 0 and 1 values in the bit string, with operations on the bit string interpreted as polynomial arithmetic.
- Consider the d-bit piece of data, **D**, that the sending node wants to send to the receiving node.
- The sender and receiver must first agree on an r + 1 bit pattern, known as a generator, which we will denote as **G**.
- We will require that the most significant (leftmost) bit of G be a 1.
- For a given piece of data, D, the sender will choose r additional bits, R, and append them to D such that the resulting d + r bit pattern (interpreted as a binary number) is exactly divisible by G (i.e., has no remainder) using modulo-2 arithmetic

Cyclic Redundancy Check (CRC)

The receiver divides the $d + r$ received bits by G. If the remainder is nonzero, the receiver knows that an error has occurred; otherwise, the data is accepted as being correct.



bits to send

<D,R> = $D * 2^r$ XOR R — formula for these bits

sender: compute *r* CRC bits, R, such that <D,R> *exactly* divisible by G (mod 2)

- receiver knows G, divides <D,R> by G. If non-zero remainder: error detected!
- can detect all burst errors less than r+1 bits
- All CRC calculations are done in modulo-2 arithmetic without carries in addition or borrows in subtraction. This means that addition and subtraction are identical, and both are equivalent to the bitwise exclusive-or (XOR) of the operands.
- widely used in practice (Ethernet, 802.11 WiFi)

Link layer, LANs: roadmap

- introduction
- error detection, correction
- **multiple access protocols**
- LANs
  - addressing, ARP
  - Ethernet
  - switches
  - VLANs
- link virtualization: MPLS
- data center networking

- a day in the life of a web request

Multiple access links, protocols

two types of "links":

- point-to-point
  - point-to-point link between switch, host
  - Single sender at one end of the the link and single receiver at the other end of the link
  - Link layer protocols: PPP (point to point protocol), HDLC (high-level data link).

- broadcast (shared wire or medium)
  - Multiple sending and receiving nodes all connected to the same single shared broadcast channel.
  - Ethernet and wireless LANs
  - upstream HFC in cable-based access network
  - 802.11 wireless LAN, 4G/4G. satellite

Multiple access protocols

Problem of central importance to the link layer:

- how to coordinate the access of multiple sending and receiving nodes to a shared broadcast channel—the **multiple access problem.**
- Broadcast channels are often used in LANs, networks that are geographically concentrated in a single building.
- Broadcasting examples: instructor and students.
  - **Who gets to talk?** (transmit into a channel), **and when?**

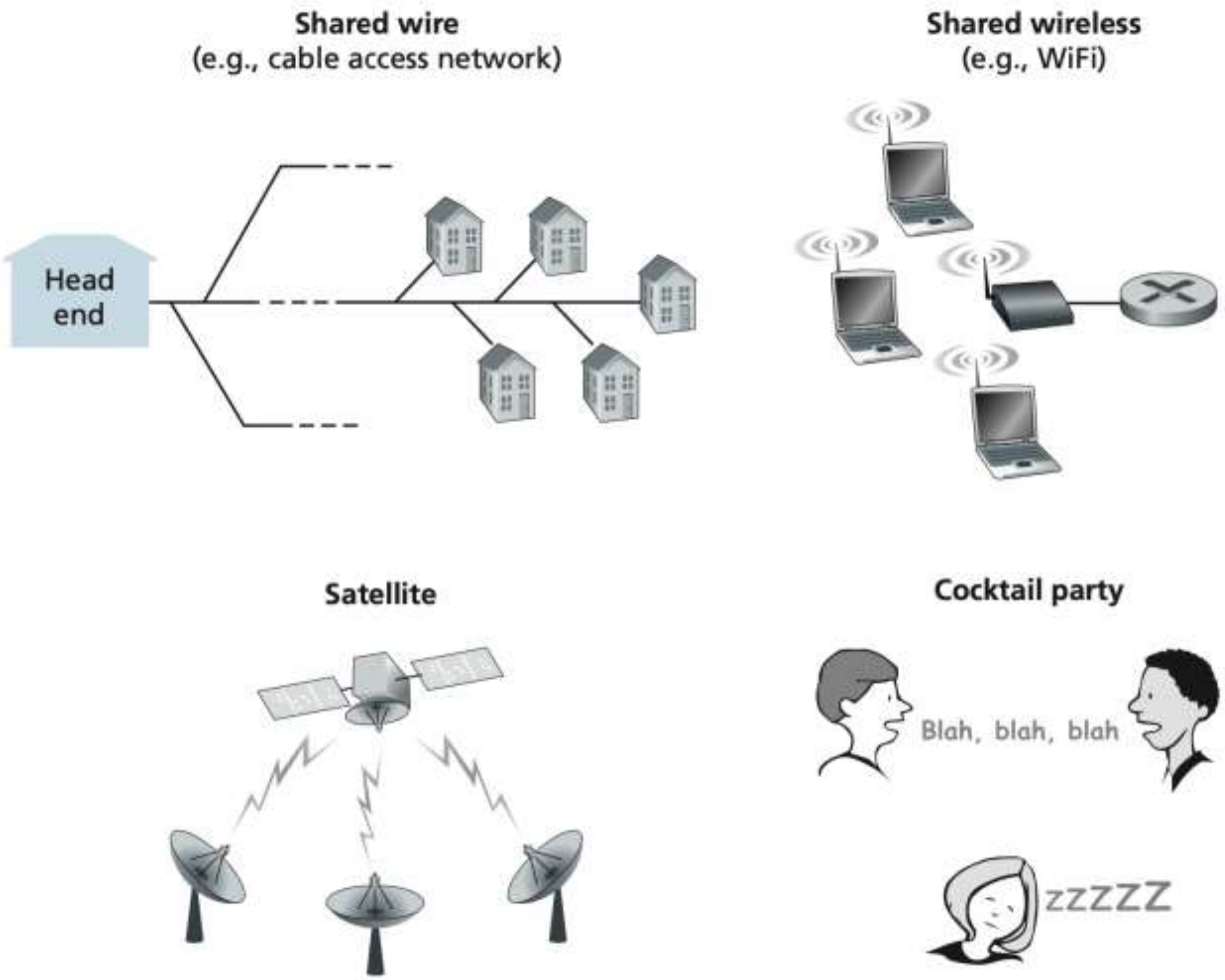
Multiple access protocols

- single shared broadcast channel

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

Multiple access protocols needed in:



Multiple Access Protocol Taxonomy

- All nodes are capable of transmitting frames, more than two nodes can transmit frames at the same time
- When this happens, all of the nodes receive multiple frames at the same time; that is, the transmitted frames collide at all of the receivers.
  - During collision. None of the receiving nodes can make any sense of any frames that were transmitted
  - All of the frames involved in the collision are lost, and the broadcast channel is wasted during the collision interval.
    - Multiply that by “many” and you get multiple channels no longer usable.

Multiple Access Protocol Taxonomy

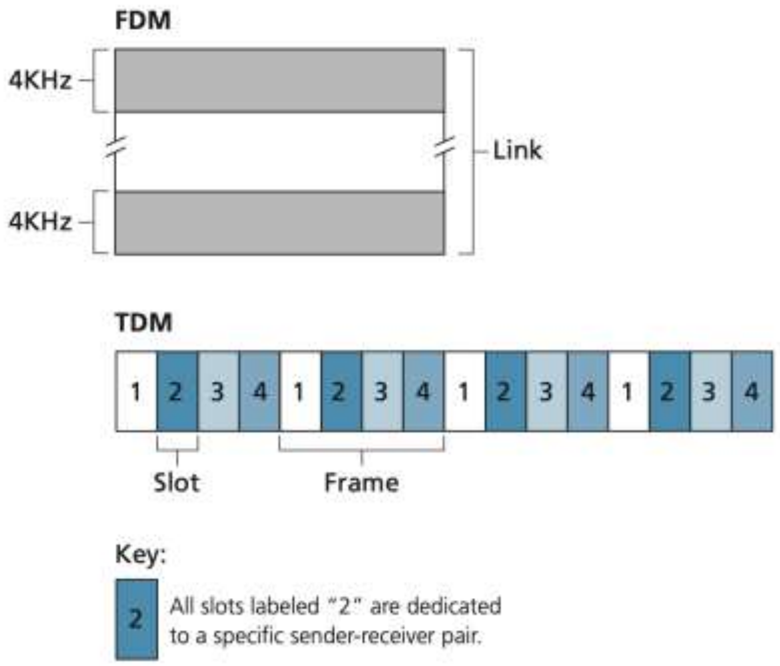
- Active nodes transmission coordination is needed – multiple access protocol
  - Newly emerging links requiring new types of multiple access protocols
- **Multiple Access Protocols**:
  - Channel partitioning
  - Random access protocols
  - Taking-turns protocols

An ideal multiple access protocol

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. The protocol is decentralized; that is, there is no master node that represents a single point of failure for the network
4. Protocol is simple, inexpensive to implement

Channel Partitioning Protocols

- time-division multiplexing (TDM) and frequency-division multiplexing (FDM) are two techniques that can be used to partition a broadcast channel's bandwidth among all nodes sharing that channel



Channel Partitioning Protocols - TDM

Two main drawbacks of TDM:

- First, a node is limited to an average rate of R/N bps even when it is the only node with packets to send
- A second drawback is that a node must always wait for its turn in the transmission sequence—even when it is the only node with a frame to send

Channel Partitioning Protocols - FDM

- While TDM shares the broadcast channel in time, FDM divides the R bps channel into different frequencies (each with a bandwidth of R/N)
  - and assigns each frequency to one of the N nodes
  - Same advantages and disadvantages of TDM
  - Avoids collision and divides the bandwidth among N nodes.
- **However, FDM also shares a principal disadvantage with TDM—a node is limited to a bandwidth of R/N, even when it is the only node with packets to send.**

Channel Partitioning Protocols - CDMA

- Code Division Multiple Access (CDMA).
- Assigns different code to each node; each node uses its unique code to encode data bits
- Successful simultaneous transmission
- Used in Military Systems (due to anti-jamming properties), and now days in telephony.

Random Access Protocols

- transmitting node always transmits at the full rate of the channel R
- When collision occurs, each node involved repeatedly retransmits its frame until it gets through without collision
  - Retransmission occurs during a random delay
  - Each node involved chooses independent random delays
- ALOHA – popular and widely used random access protocols.

Slotted ALOHA

- All frames consist of exactly L bits.
- Time is divided into slots of size L/R seconds (that is, a slot equals the time to transmit one frame).
- Nodes start to transmit frames only at the beginnings of slots.
- The nodes are synchronized so that each node knows when the slots begin.
- If two or more frames collide in a slot, then all the nodes detect the collision event before the slot ends.

Slotted ALOHA

- Let p be a probability, that is, a number between 0 and 1. The operation of slotted ALOHA in each node is simple:
- When the node has a fresh frame to send, it waits until the beginning of the next slot and transmits the entire frame in the slot.
- If there isn't a collision, the node has successfully transmitted its frame and thus need not consider retransmitting the frame.
- If there is a collision, the node detects the collision before the end of the slot. The node retransmits its frame in each subsequent slot with probability p until the frame is transmitted without a collision.

Slotted ALOHA

- Unlike channel partitioning, slotted ALOHA allows a node to transmit continuously at the full rate, R, when that node is the only active node
- Slotted ALOHA is also highly decentralized, because each node detects collisions and independently decides when to retransmit.

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

Carrier Sense Multiple Access CSMA/CD

- In ALOHA, a node's decision to transmit is made independently of the activity of the other nodes attached to the broadcast channel
- Using human discussion analogy:
  - Listen before speaking. If someone else is speaking, wait until they are finished. In the networking world, this is called carrier sensing—a node listens to the channel before transmitting.
  - If someone else begins talking at the same time, stop talking. In the networking world, this is called collision detection (CD)

CSMA (carrier sense multiple access)

simple CSMA: listen before transmit:

- if channel sensed idle: transmit entire frame
- if channel sensed busy: defer transmission

- human analogy: don't interrupt others!

CSMA/CD: CSMA with *collision detection*

- collisions *detected* within short time
- colliding transmissions aborted, reducing channel wastage
- collision detection easy in wired, difficult with wireless

- human analogy: the polite conversationalist

Taking turns protocols

- Two desirable properties of a multiple access protocol are:
  - (1) when only one node is active, the active node has a throughput of R bps
  - (2) when M nodes are active, then each active node has a throughput of nearly R/M bps
    - ALOHA and CMS protocols have the #1, missing property #2.
- Taking turn protocols designed to address the missing property (#2)
  - **Polling protocol –** requires one nodes to be designated a master node throughout the transmission, which eliminates the collision.
  - **Token-passing protocol –** no master node; special frame known as **token** is exchanged among the nodes in fixed order. The token is kept by the node during the transmission only (hot-potato analogy)

Draw-backs with Taking Turn Protocols

- Drawbacks of polling protocol:
  - If master node fails, the entire channel becomes unavailable
  - Polling delay – the time required to notify a node that it can transmit
    - Bluetooth is an example of polling protocol

- Drawbacks of token-passing protocol:
  - The failure of one node can crash the entire channel
  - If a node accidentally didn’t release the token, then recovery is needed to get token back in the circulation.

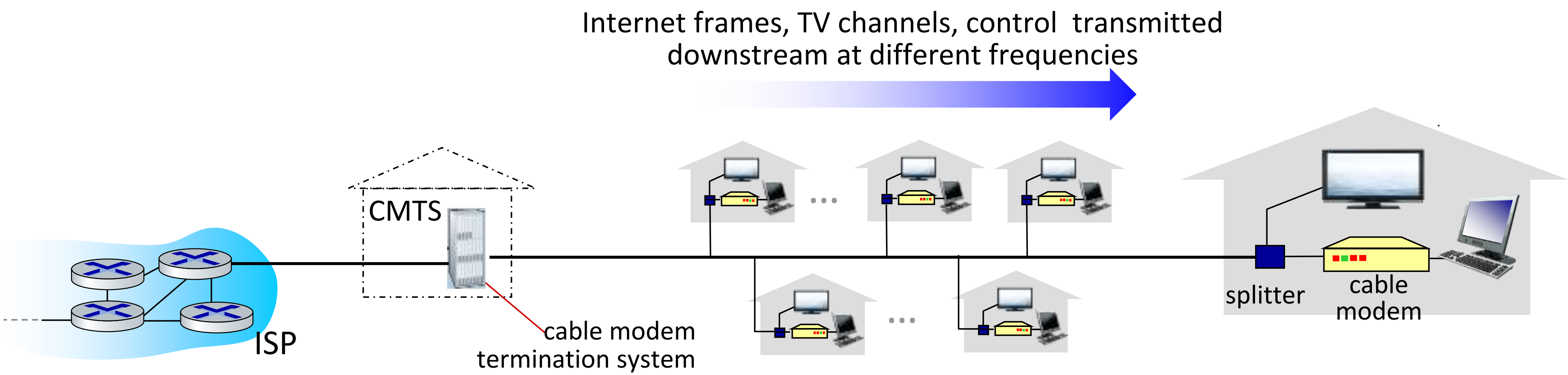
DOCSIS – Link-layer protocol for cable internet

- Data-Over-Cable Service Interface Specification (DOCSIS)
  - Specifies the cable data network architecture and its protocols
  - DOCSIS uses FDM to divide the downstream (CMTS to modem) and upstream (modem to CMTS) network segments into multiple frequency channels.
  - Each downstream channel is between 24 MHz and 192 MHz wide, with a maximum throughput of approximately 1.6 Gbps per channel; each upstream channel has channel widths ranging from 6.4 MHz to 96 MHz, with a maximum upstream throughput of approximately 1 Gbps
    - **CMTS** – cable modem termination system
  - **Incorporates all 3 classes of multiple access protocols** (partition, random access, and taking turns)

DOCSIS

- Each upstream and downstream channel is a broadcast channel
- Frames transmitted on the downstream channel by the CMTS are received by all cable modems receiving that channel; single CMTS transmitting into the downstream channel results in no multiple access problem
- The upstream direction, multiple cable modems share the same upstream channel (frequency) to the CMTS, and thus collisions can occur.

Cable access network: FDM, TDM *and* random access!



- multiple downstream (broadcast) FDM channels: up to 1.6 Gbps/channel
  - single CMTS transmits into channels
- multiple upstream channels (up to 1 Gbps/channel)
  - multiple access: all users contend (random access) for certain upstream channel time slots; others assigned TDM

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

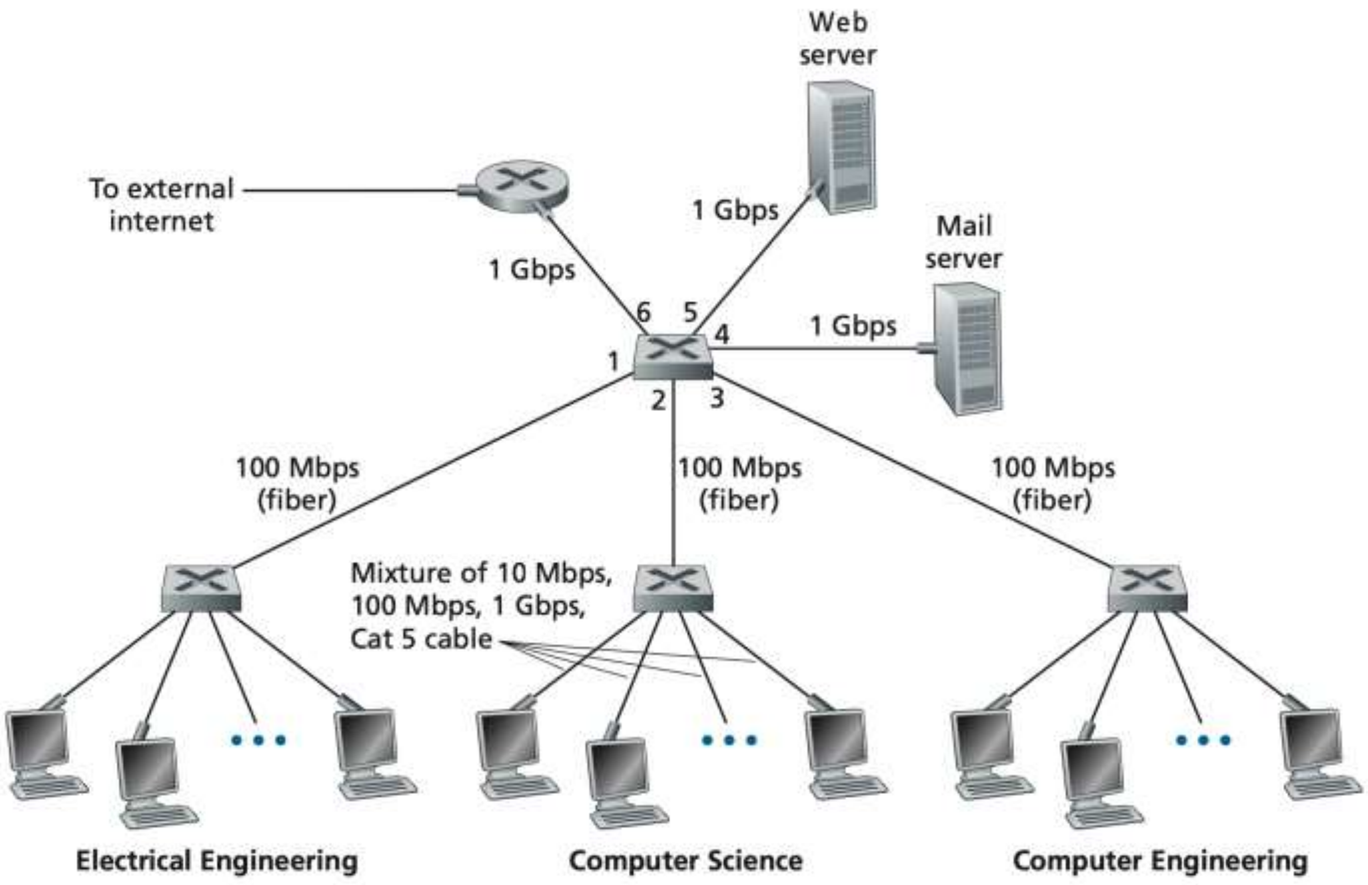
Link layer, LANs: roadmap

- introduction
- error detection, correction
- multiple access protocols
- LANs
  - addressing, ARP
  - Ethernet
  - switches
  - VLANs
- link virtualization: MPLS
- data center networking

- a day in the life of a web request

Switched Local Area Network

Portion of university infrastructure: 3 departments, 2 servers, router and 4 switches



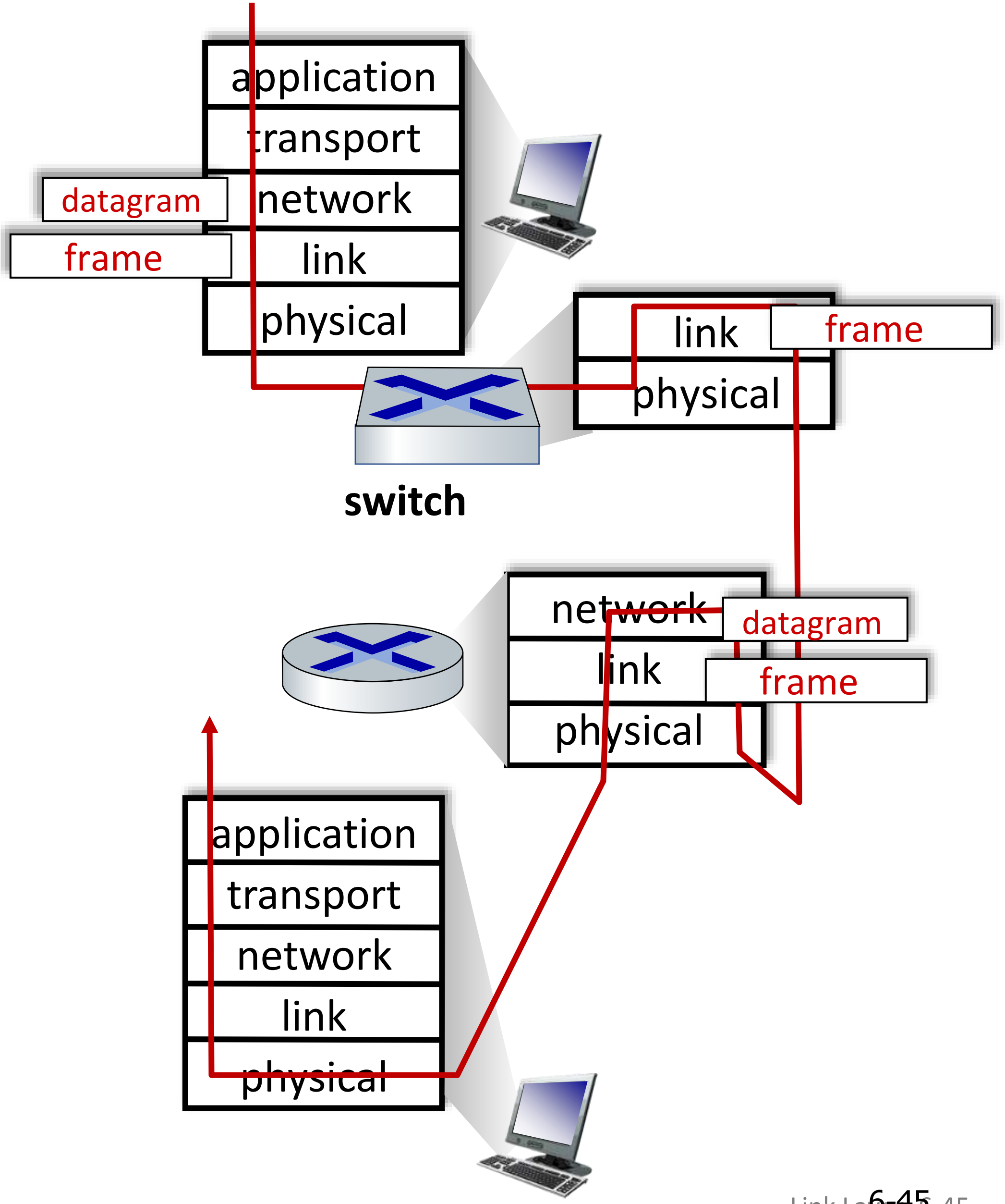
Switches vs. routers

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



Switches

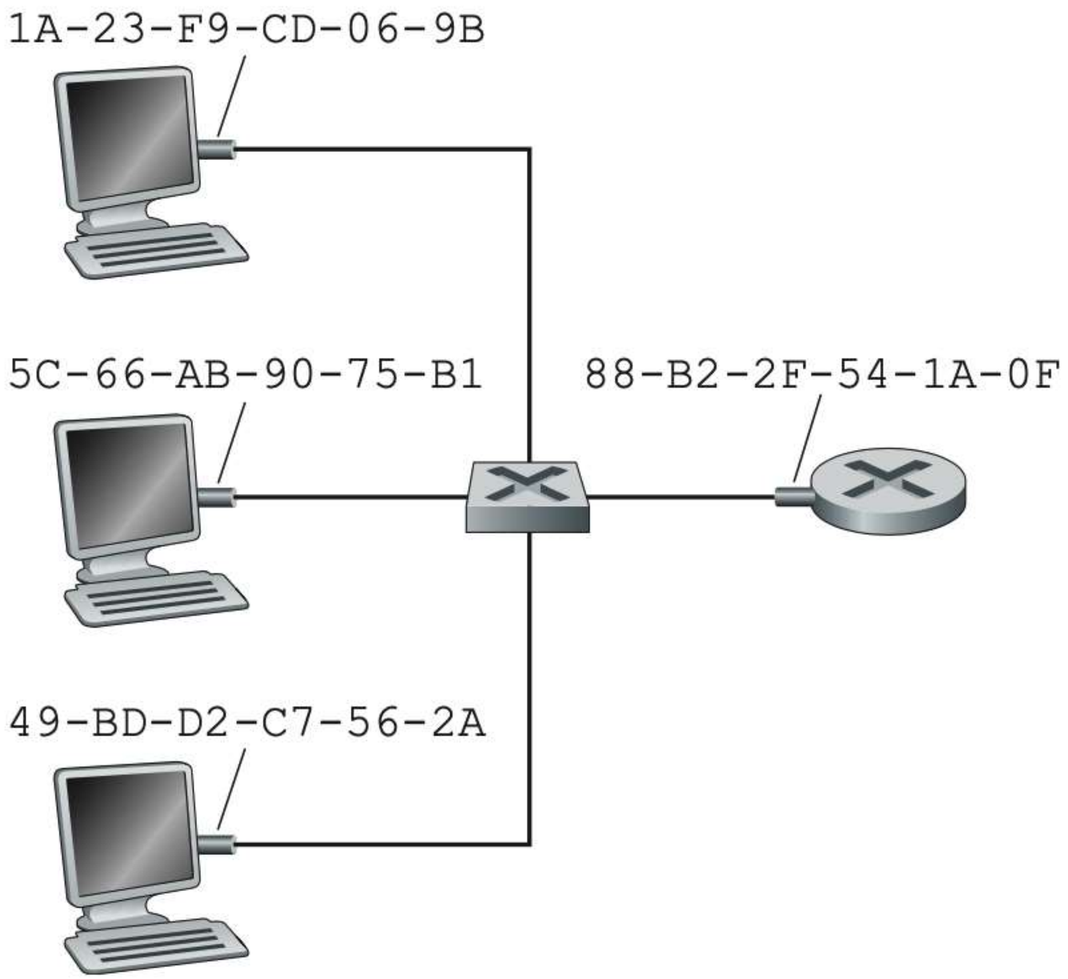
- Mainly operate at link layer (layer 2)
- Link-layer frames are exchanged
- Majority do not “read” network –layer datagrams
  - **No access to the network-layer addresses**
  - **Do not use routing algorithm, like OSPF**
- Link-layer addresses are used to forward frames: MAC address

MAC addresses

- 32-bit IP address:
  - *network-layer* address for interface
  - used for layer 3 (network layer) forwarding
  - e.g.: 128.119.40.136
- MAC (or LAN or physical or Ethernet) address:
  - function: used “locally” to get frame from one interface to another physically-connected interface (same subnet, in IP-addressing sense)
  - 48-bit MAC address (for most LANs) hard-coded into NIC, also sometimes software settable
  - **Unique!** e.g.: 1A-2F-BB-76-09-AD

hexadecimal (base 16) notation
(each “numeral” represents 4 bits)

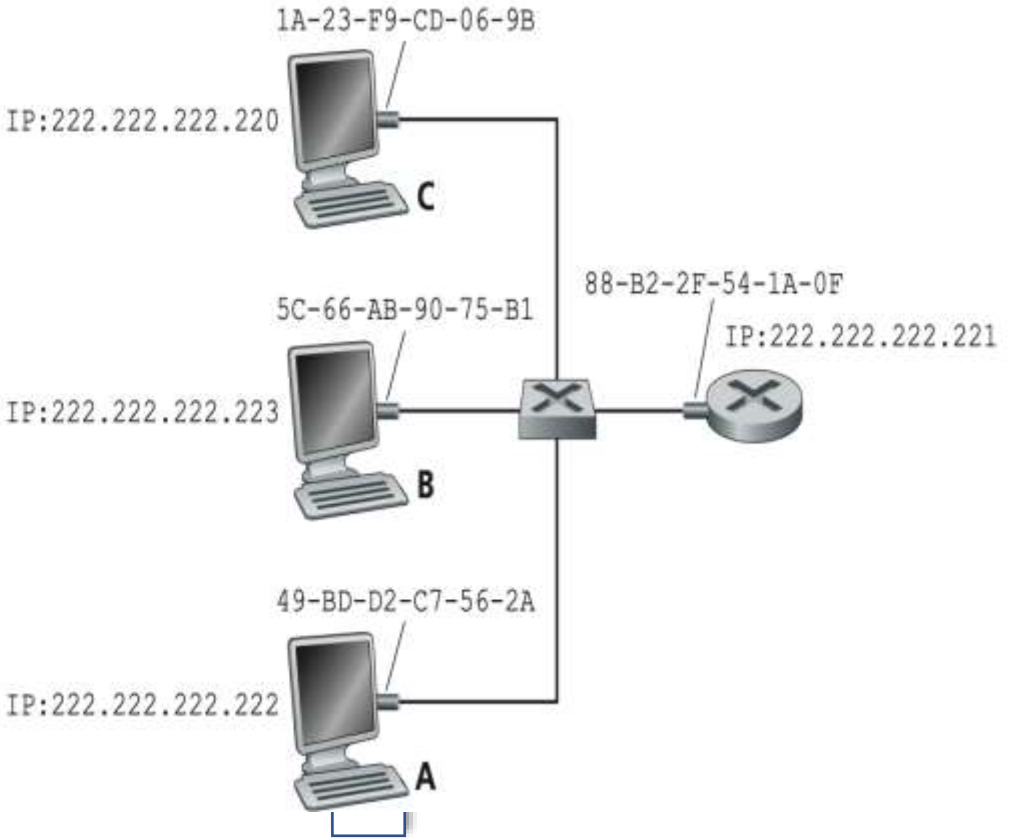
MAC addresses



MAC addresses

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

ARP: address resolution protocol



ARP table in memory: 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)

One important difference between the two resolvers is that DNS resolves host names for hosts anywhere in the Internet, whereas ARP resolves IP addresses only for hosts and router interfaces on the same subnet

ARP protocol in action

example: A wants to send datagram to B

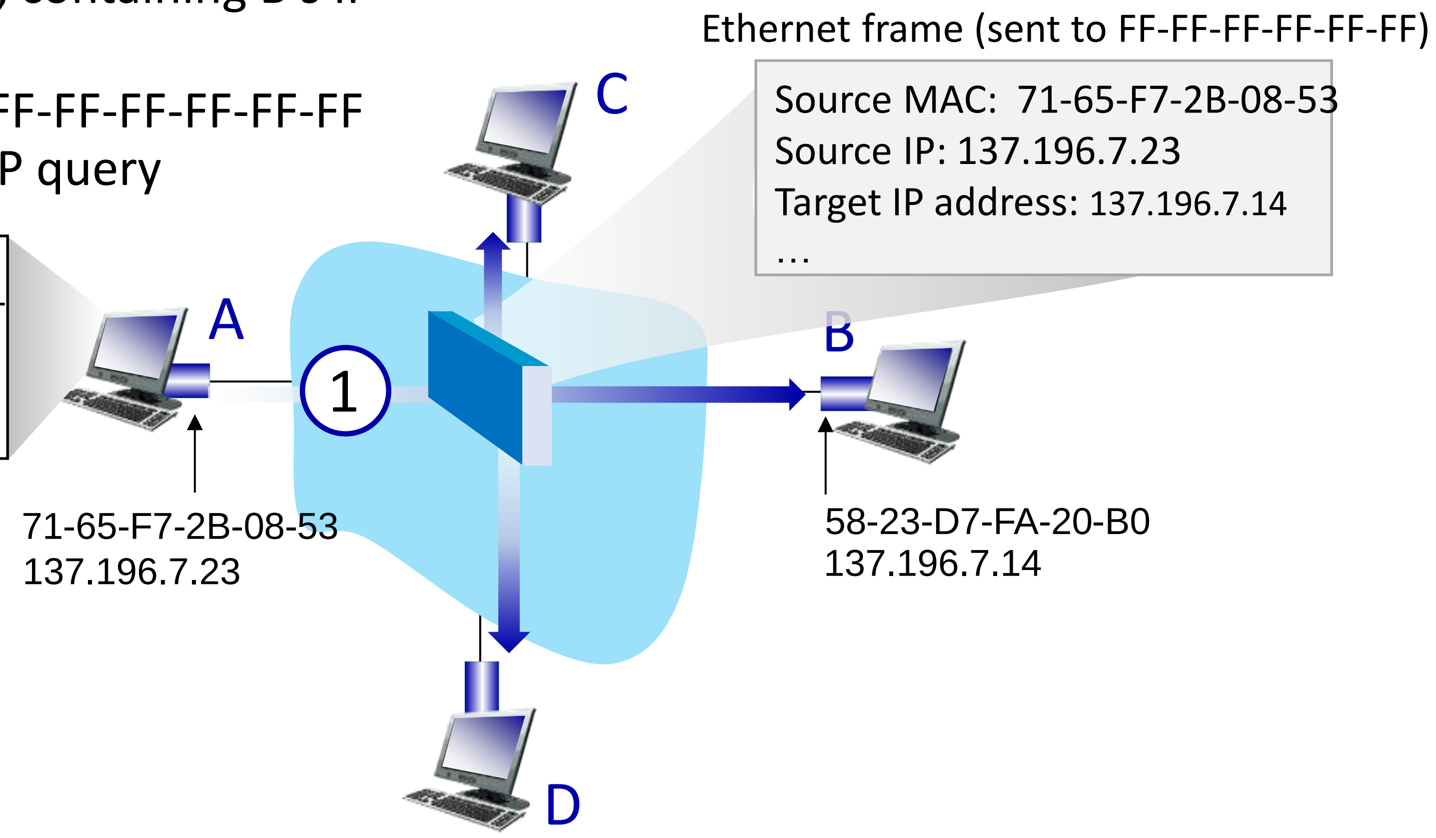
- B's MAC address not in A's ARP table, so A uses ARP to find B's MAC address

1 A broadcasts ARP query/packet, containing B's IP addr

- destination MAC address = FF-FF-FF-FF-FF-FF
- all nodes on LAN receive ARP query

ARP table in A

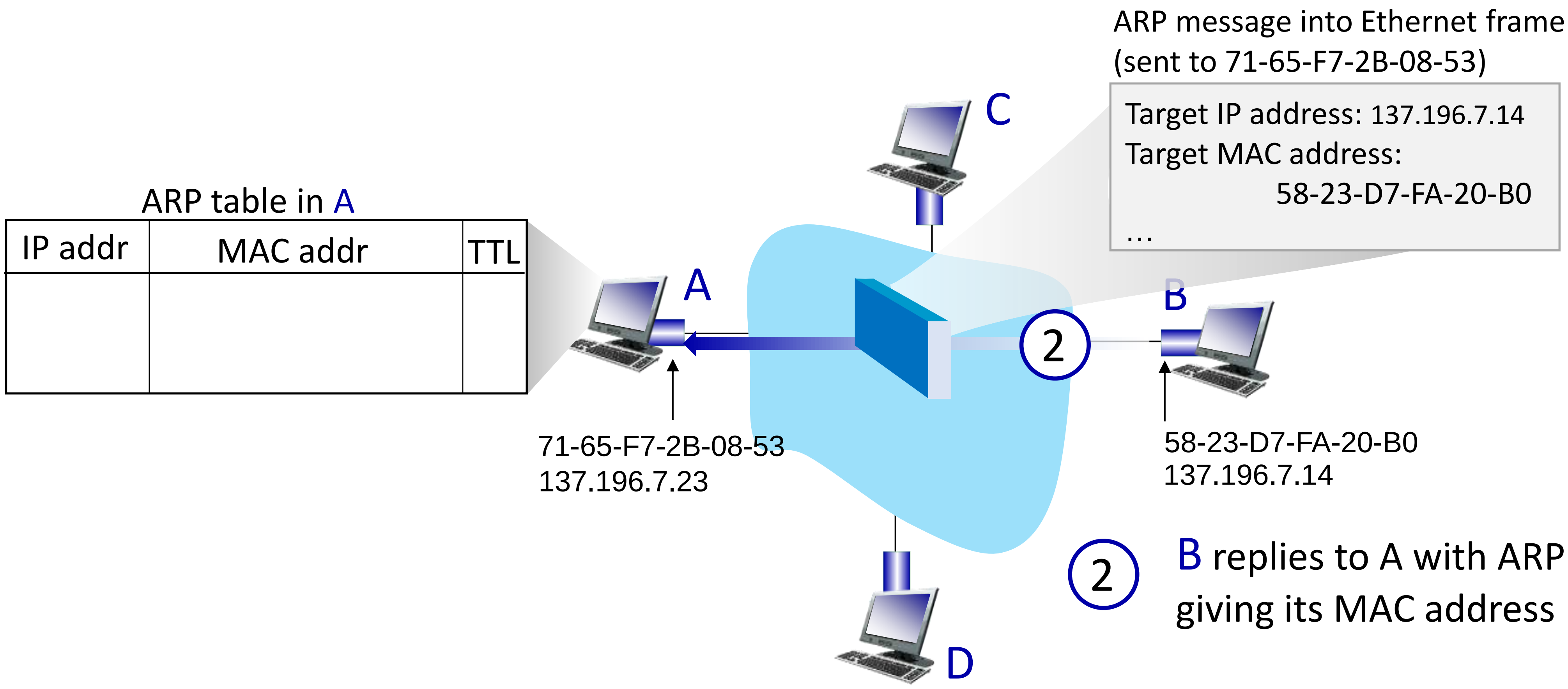
IP addr	MAC addr	TTL



ARP protocol in action

example: A wants to send datagram to B

- B's MAC address not in A's ARP table, so A uses ARP to find B's MAC address



2 B replies to A with ARP response, giving its MAC address

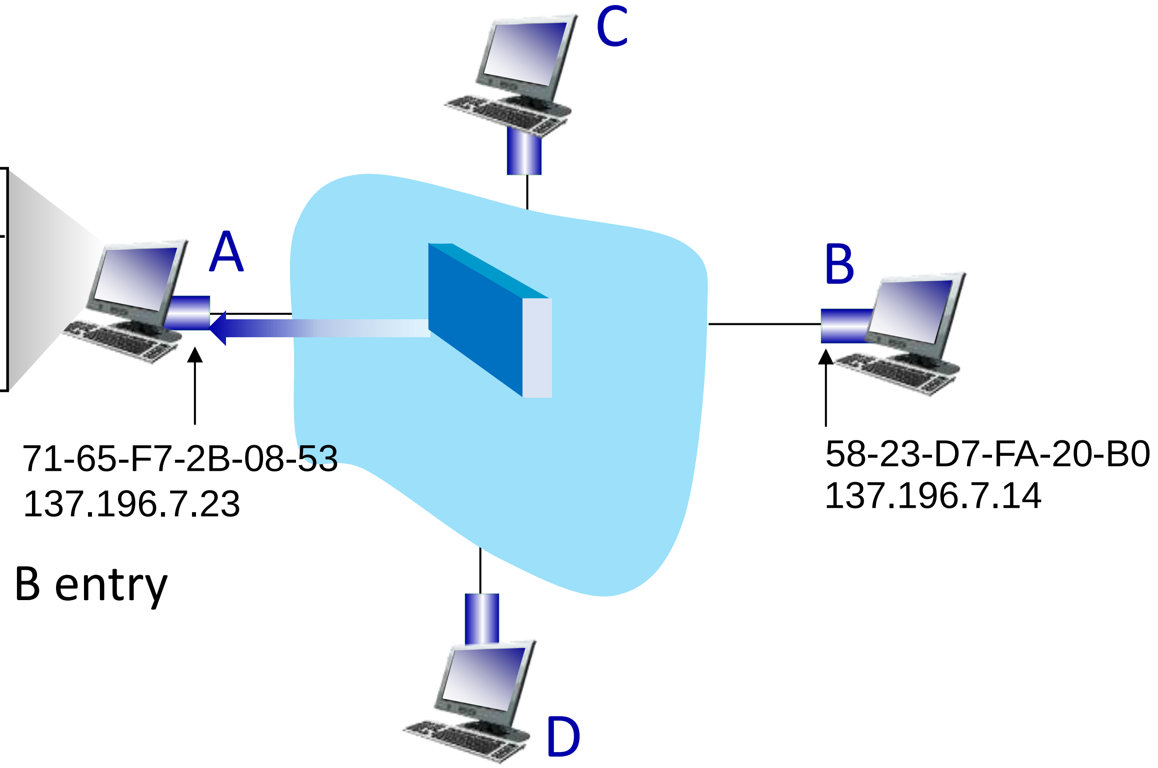
ARP protocol in action

example: A wants to send datagram to B

- B's MAC address not in A's ARP table, so A uses ARP to find B's MAC address

ARP table in A

IP addr	MAC addr	TTL
137.196.7.14	58-23-D7-FA-20-B0	500

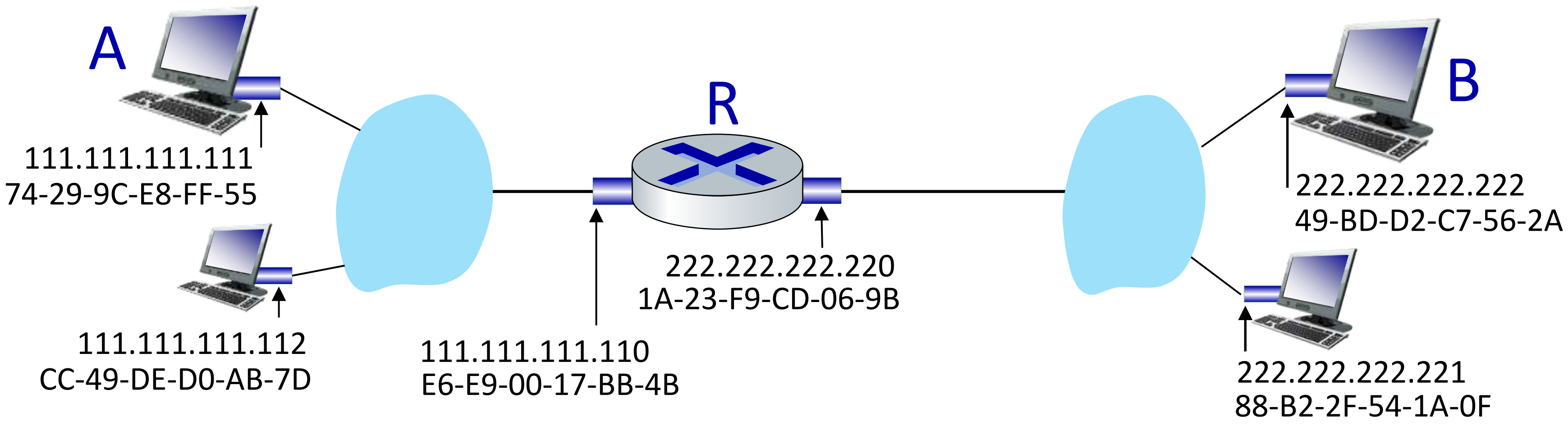


3 A receives B's reply, adds B entry into its local ARP table

Routing to another subnet: addressing

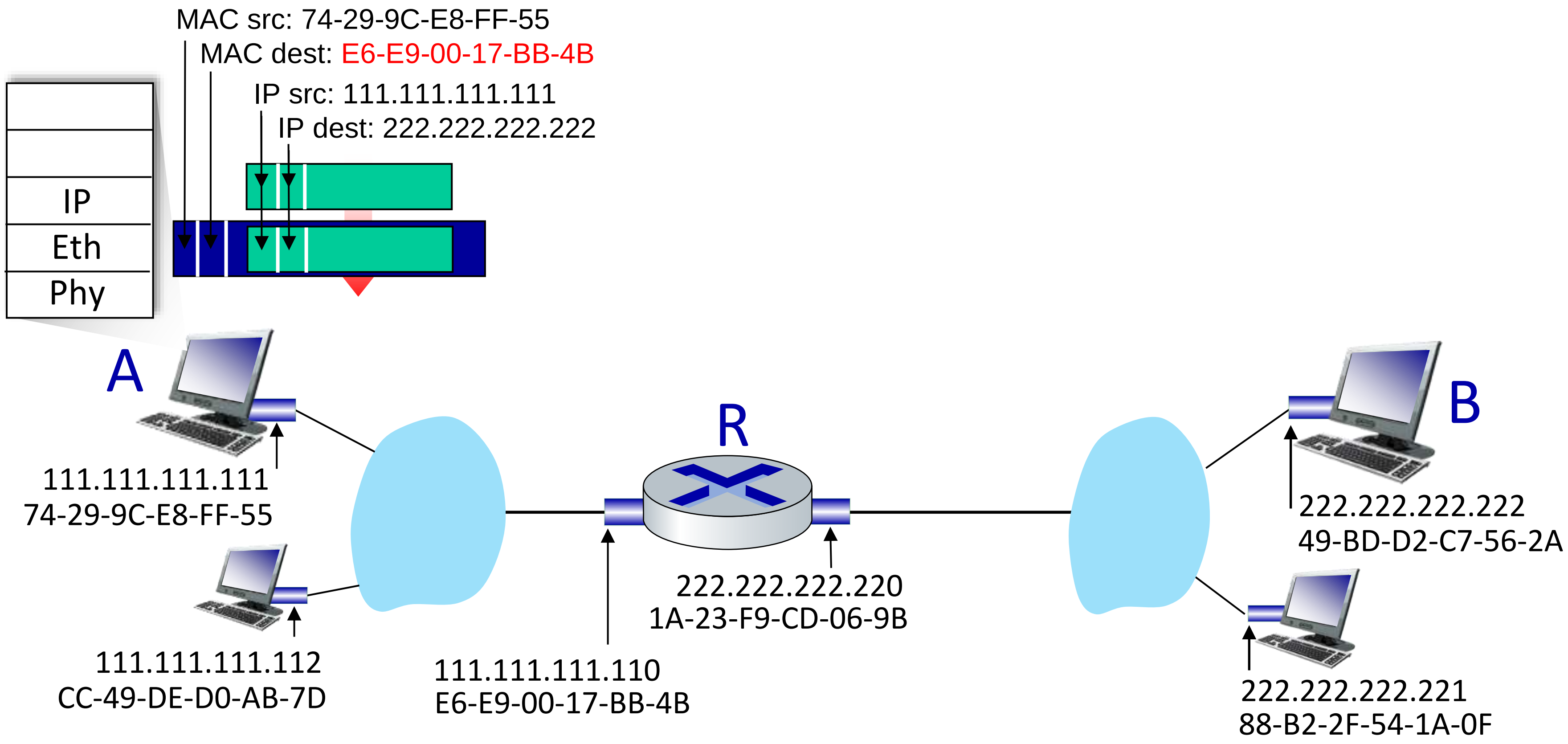
walkthrough: sending a datagram from *A* to *B* via *R*

- focus on addressing – at IP (datagram) and MAC layer (frame) levels
- assume that:
  - A knows B’s IP address
  - A knows IP address of first hop router, R
  - A knows R’s MAC address



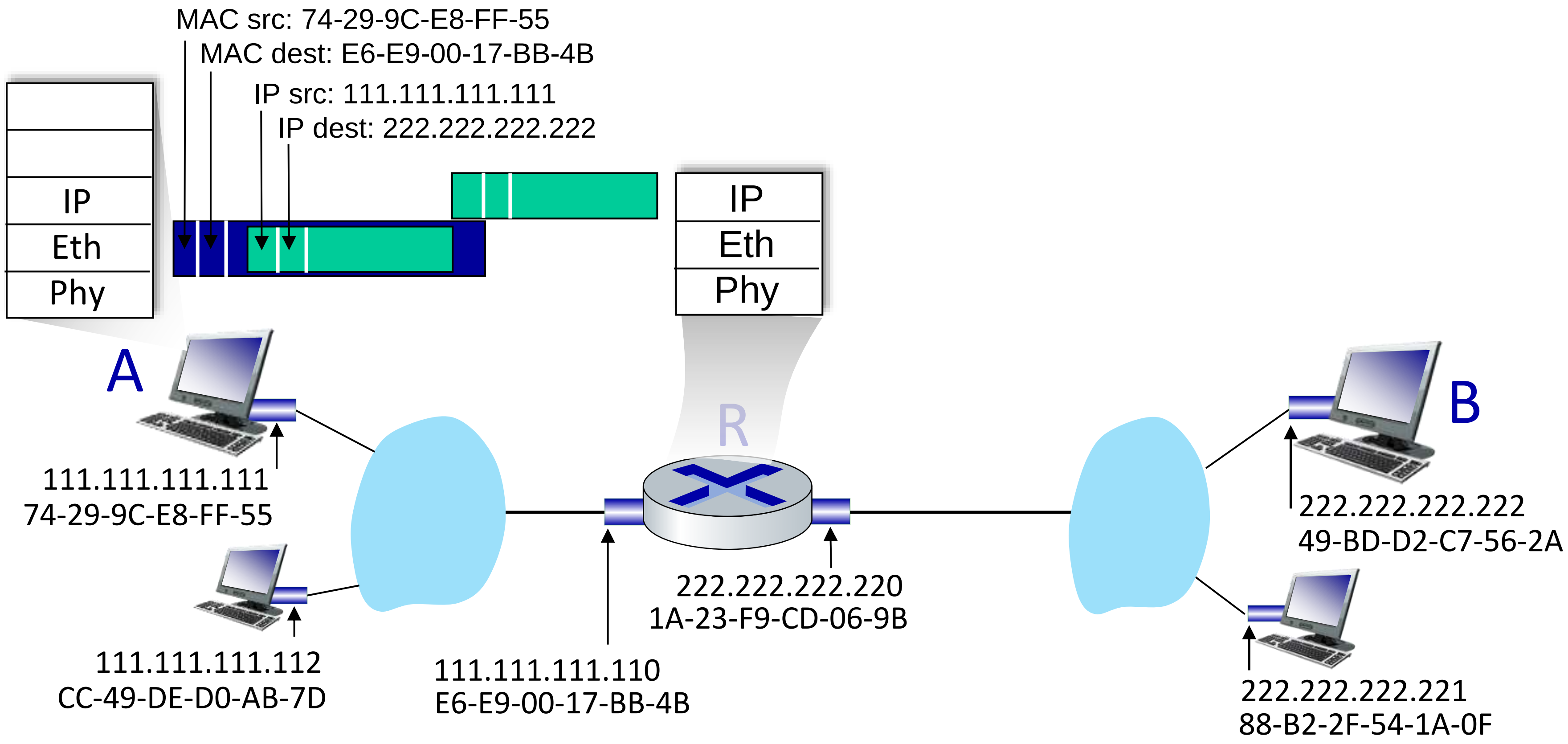
Routing to another subnet: addressing

- A creates IP datagram with IP source A, destination B
- A creates link-layer frame containing A-to-B IP datagram
  - R's MAC address is frame's destination



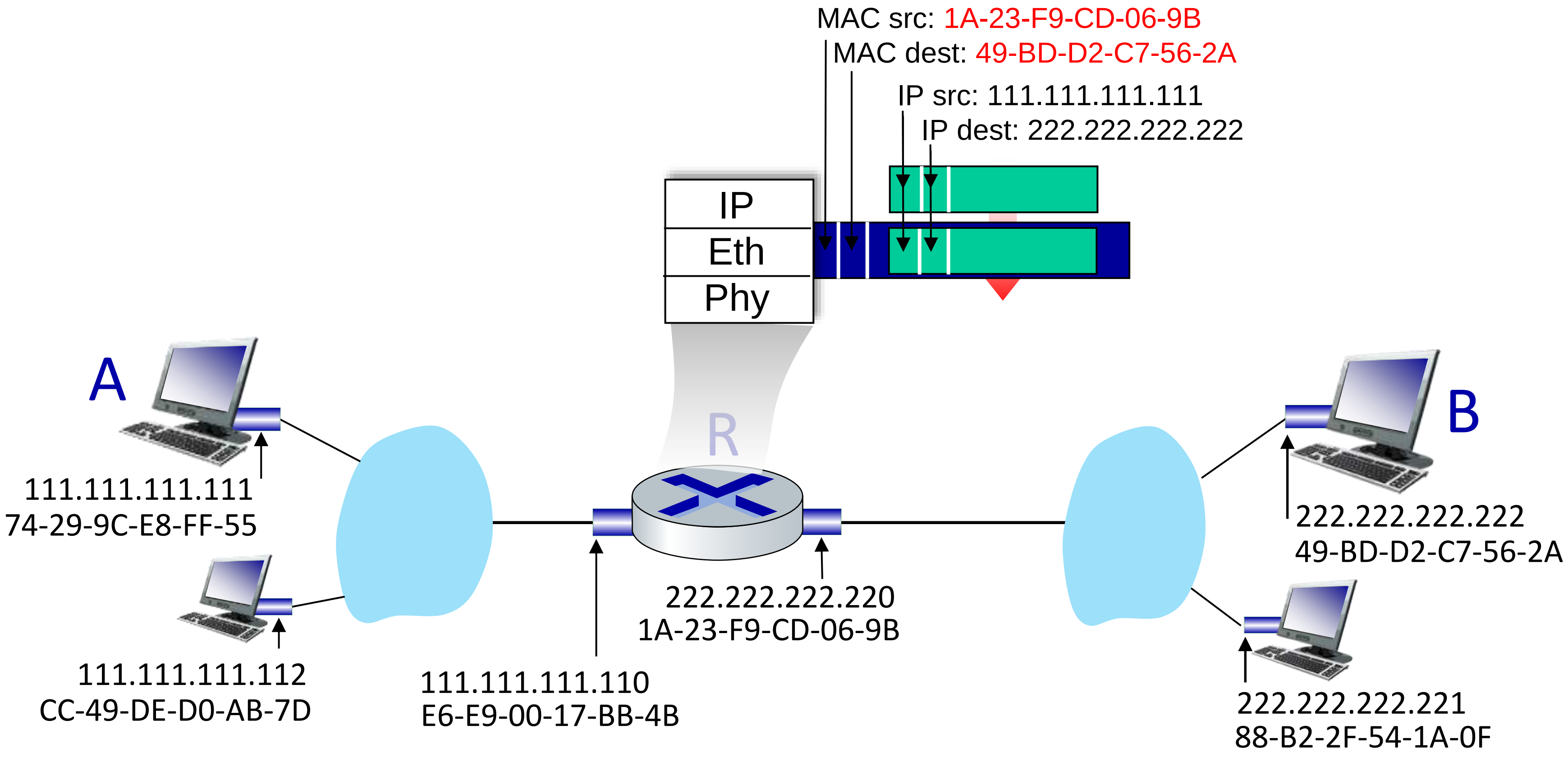
Routing to another subnet: addressing

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



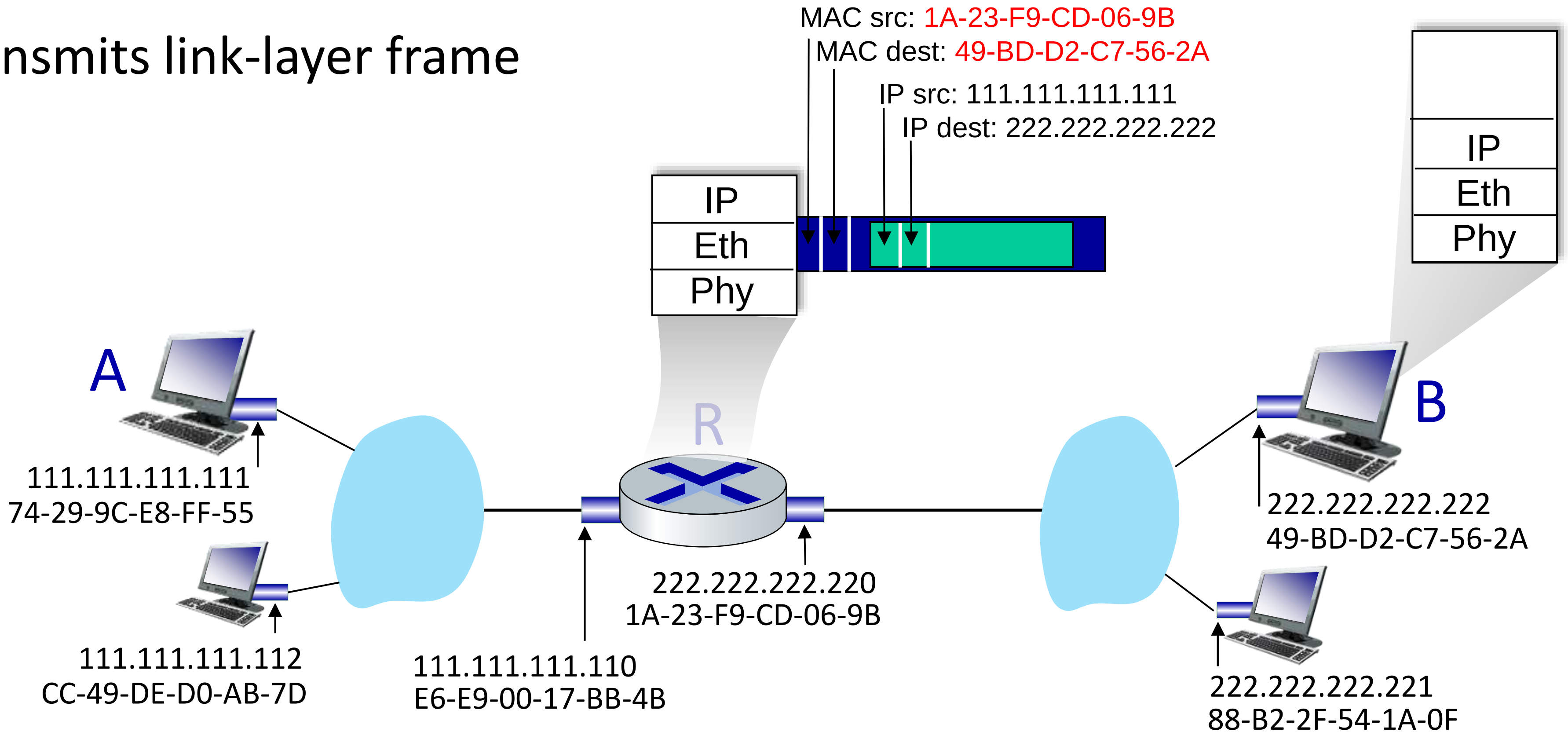
Routing to another subnet: addressing

- R determines outgoing interface, passes datagram with IP source A, destination B to link layer
- R creates link-layer frame containing A-to-B IP datagram. Frame destination address: B's MAC address



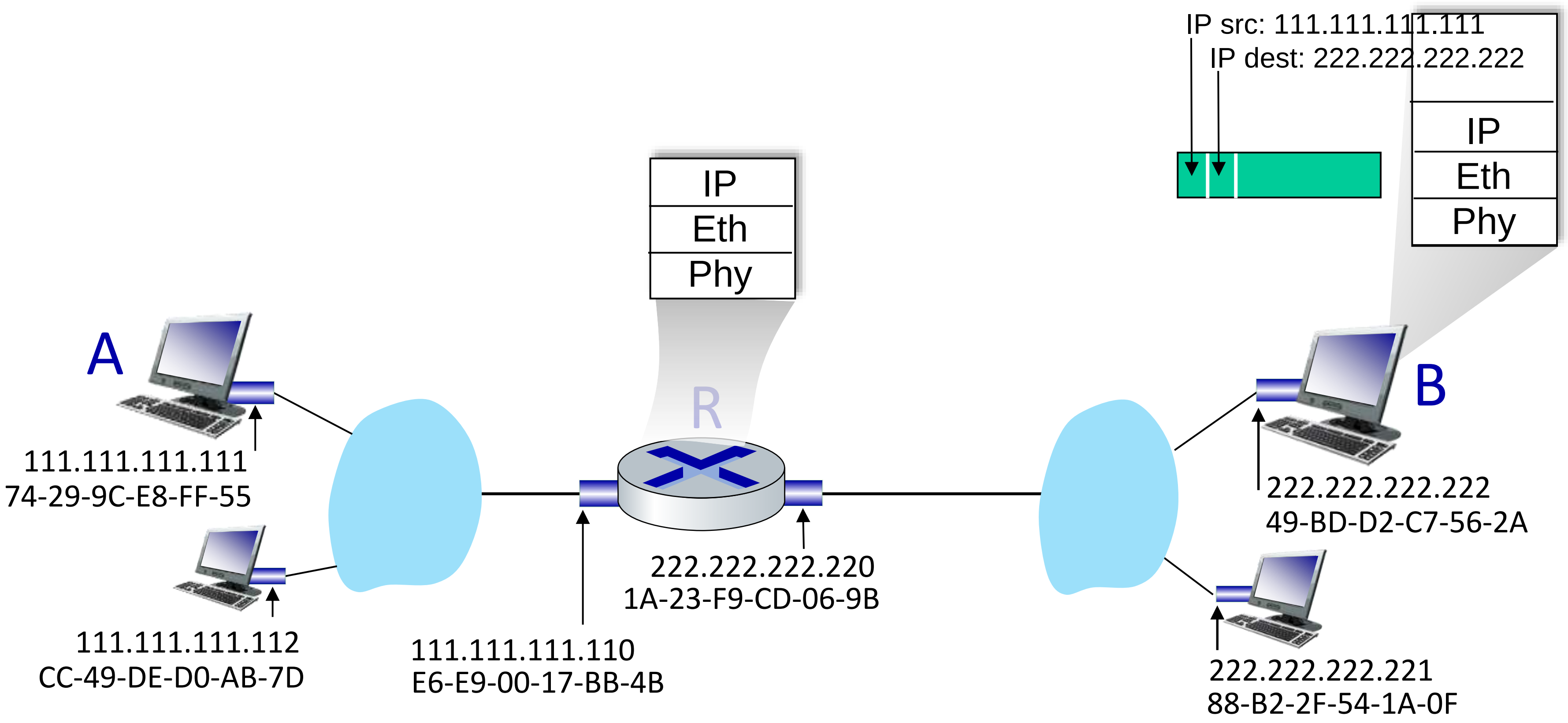
Routing to another subnet: addressing

- R determines outgoing interface, passes datagram with IP source A, destination B to link layer
- R creates link-layer frame containing A-to-B IP datagram. Frame destination address: B's MAC address
- transmits link-layer frame



Routing to another subnet: addressing

- B receives frame, extracts IP datagram destination B
- B passes datagram up protocol stack to IP



Link layer, LANs: roadmap

- introduction
- error detection, correction
- multiple access protocols
- LANs
  - addressing, ARP
  - Ethernet
  - switches
  - VLANs
- link virtualization: MPLS
- data center networking

- a day in the life of a web request

Ethernet frame structure

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

preamble:

- serves to “wake up” the receiving adapters and to sync their clocks, to sender’s clock.

Destination address:

- MAC address of the destination adapter

Source Address:

- MAC address of the adapter that transmits the frame onto the LAN

Ethernet frame structure (more)

Data:

- IP datagram (max size 1,500 bytes)

Type:

Permits Ethernet to multiplex

CRC:

- Bit errors detection in the frame

Ethernet

- No handshaking (**connectionless**)
- Analogous to IPS's layer 3 datagram service and UDP's layer 4 connectionless service
- Ethernet provides **unreliable** service to the network layer
  - **No ACKs or NACKs**
    - Frame is discarded
- Ethernet is unaware if it re-transmits the frame, or sends a brand new one
- Simple and Cheap

IEE 802.3 Ethernet standards: link & physical layers

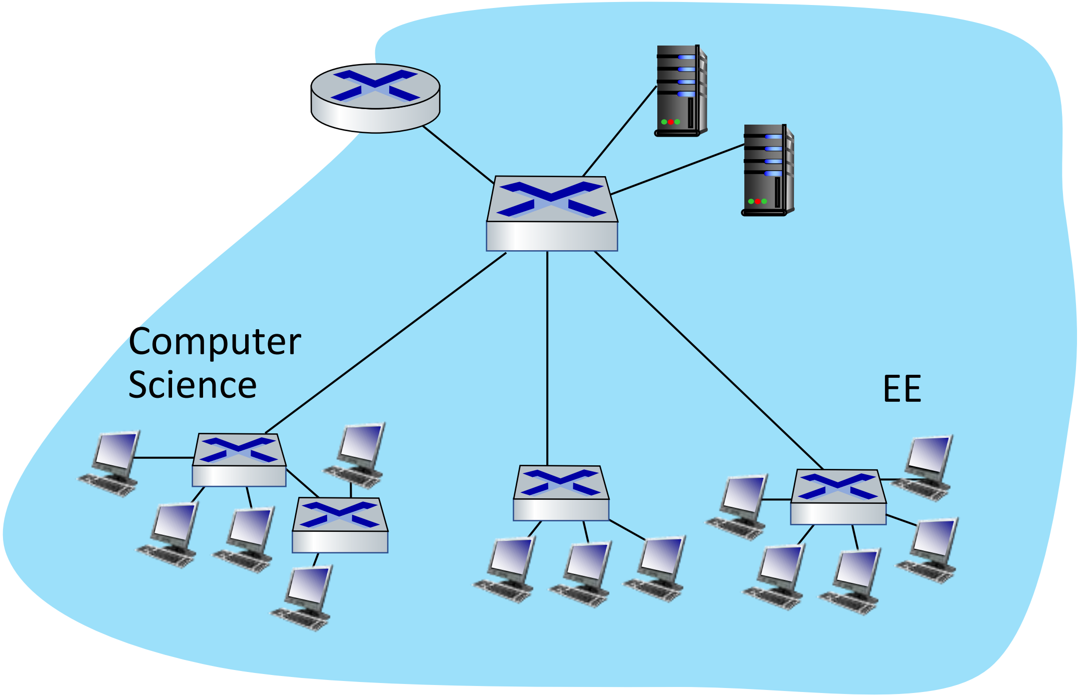
- Many different protocols and standards under IEEE 802.3 (Ethernet):
  - 10BASE-T, 10BASE-2, 100BASE-T,1000BASE-LX, 10GBASE-T, 40GBASE-T
    - First number refers to speed in Megabit per second
    - "BASE" – baseband Ethernet, physical media carries only Ethernet traffic
    - Last portion of the acronym – physical media itself (coaxial cable, copper wire, fiber)

Link layer, LANs: roadmap

- introduction
- error detection, correction
- multiple access protocols
- LANs
  - addressing, ARP
  - Ethernet
  - switches
  - VLANs
- link virtualization: MPLS
- data center networking

- a day in the life of a web request

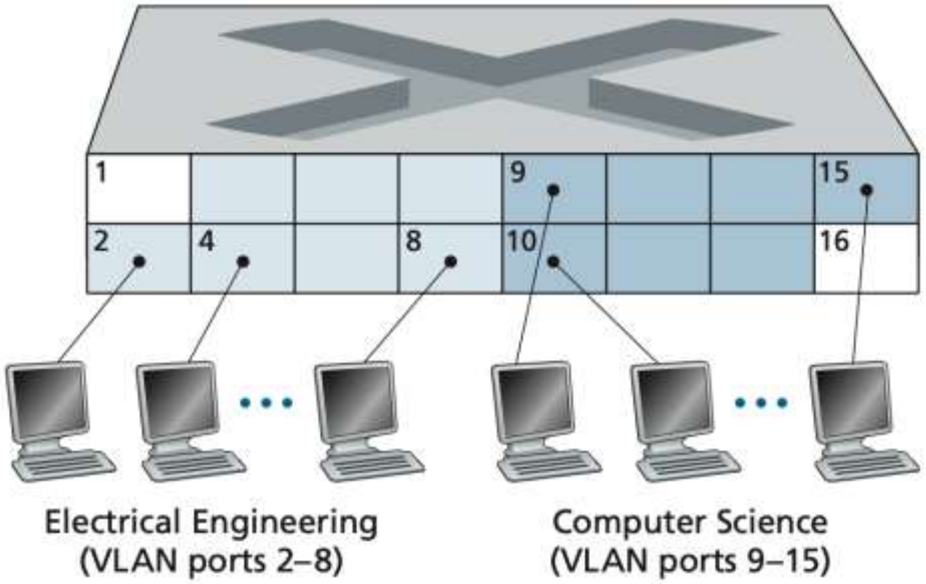
Virtual LANs (VLANs): motivation



Three drawbacks:

- Lack of traffic isolation
- Inefficient use of switches
- Managing users

Virtual LANs (VLANs): motivation

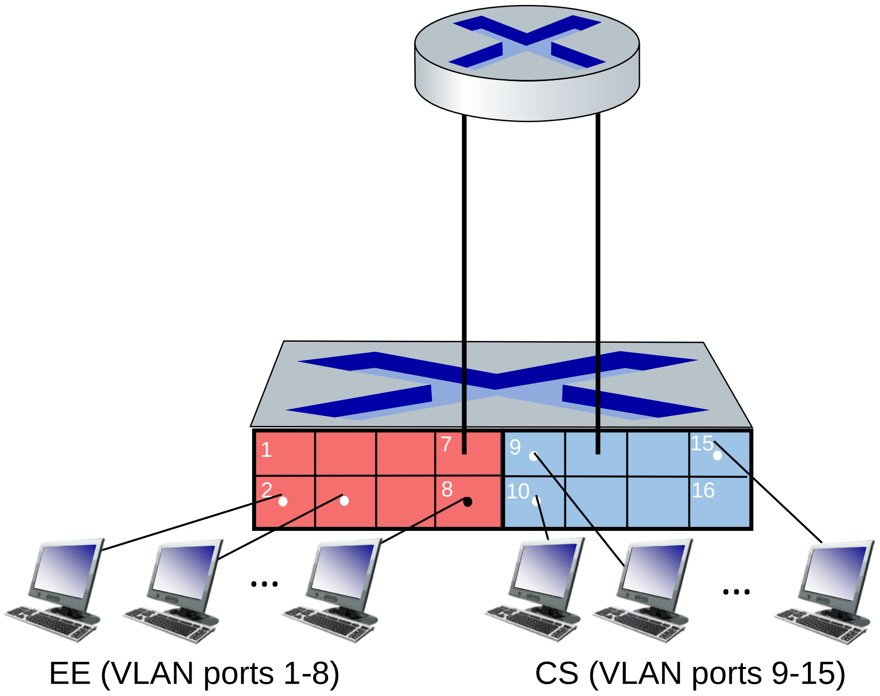


Switch supports VLANs; multiple virtual local area networks to be defined over a single physical local area network infrastructure.

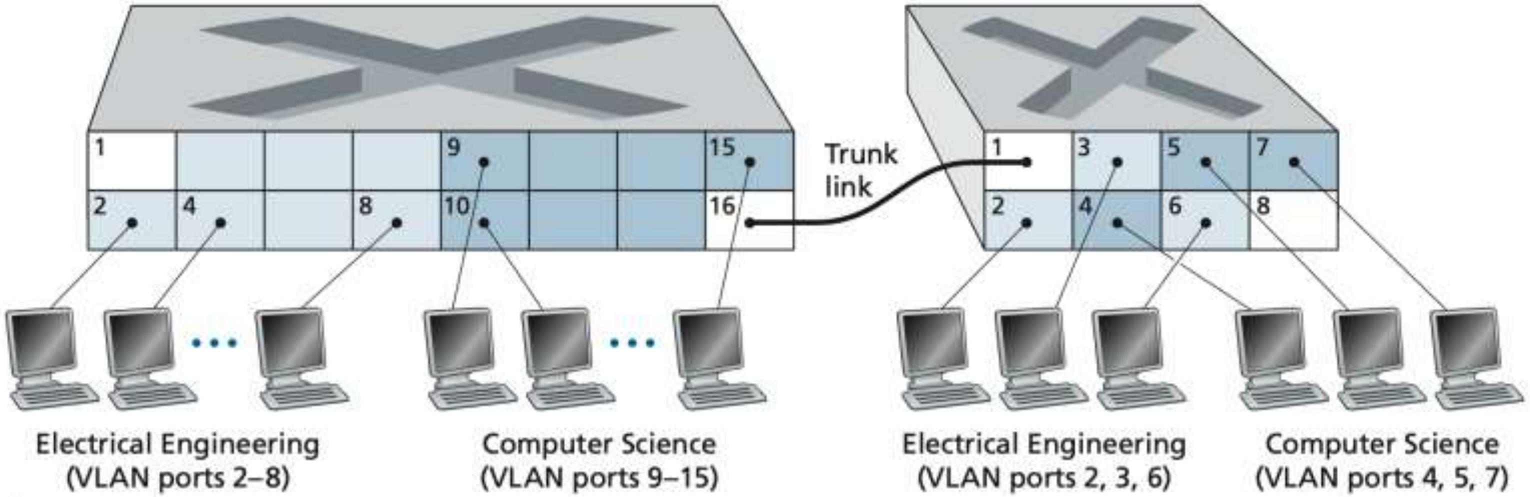
- Hosts communicate with each other as if they were connected to the switch
- Port-based VLANs: switch ports (interfaces/APIs) divided into groups by network manager
- Solves the problems from previously articulated problems with switches

Port-based VLANs

- 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



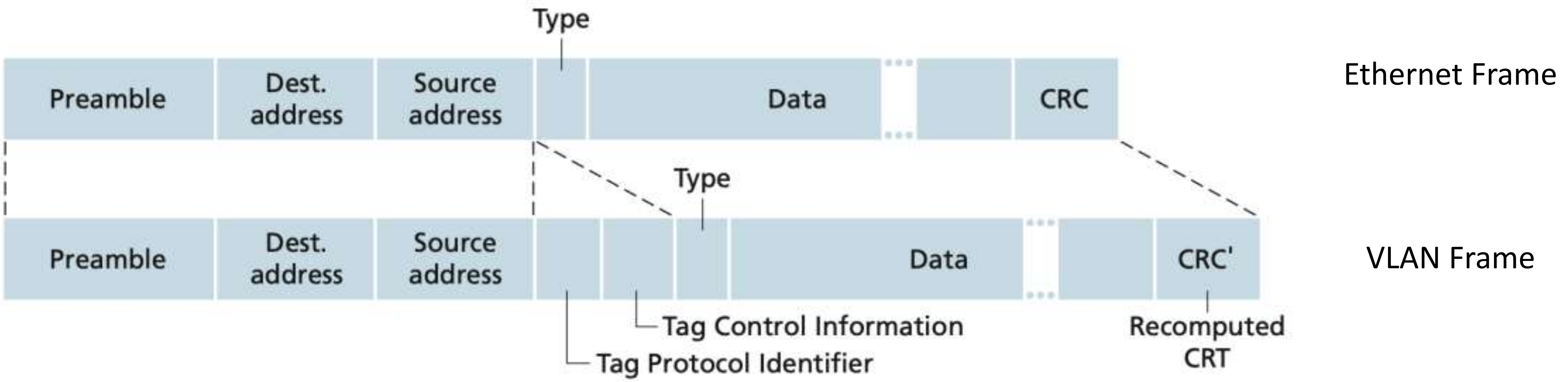
VLANS spanning multiple switches



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

- Interconnects the two VLAN switches
- Trunk port belongs to all VLAN switches
- Frames sent to any VLAN are forwarded over the trunk link to the other switch
- 8021.Q – standard defining how frames cross VLANs

802.1Q VLAN frame format



- **VLAN Tag** –carries the identity of the VLAN to which the frame belongs
- Added by the sending side of the VLAN trunk, removed by the receiving side of the trunk
  - TPI & TCI - VLAN identifier field and priority field

Link layer, LANs: roadmap

- introduction
- error detection, correction
- multiple access protocols
- LANs
  - addressing, ARP
  - Ethernet
  - switches
  - VLANs
- link virtualization: MPLS
- data center networking

- a day in the life of a web request

Datacenter networks

10's to 100's of thousands of hosts, often closely coupled, in close proximity:

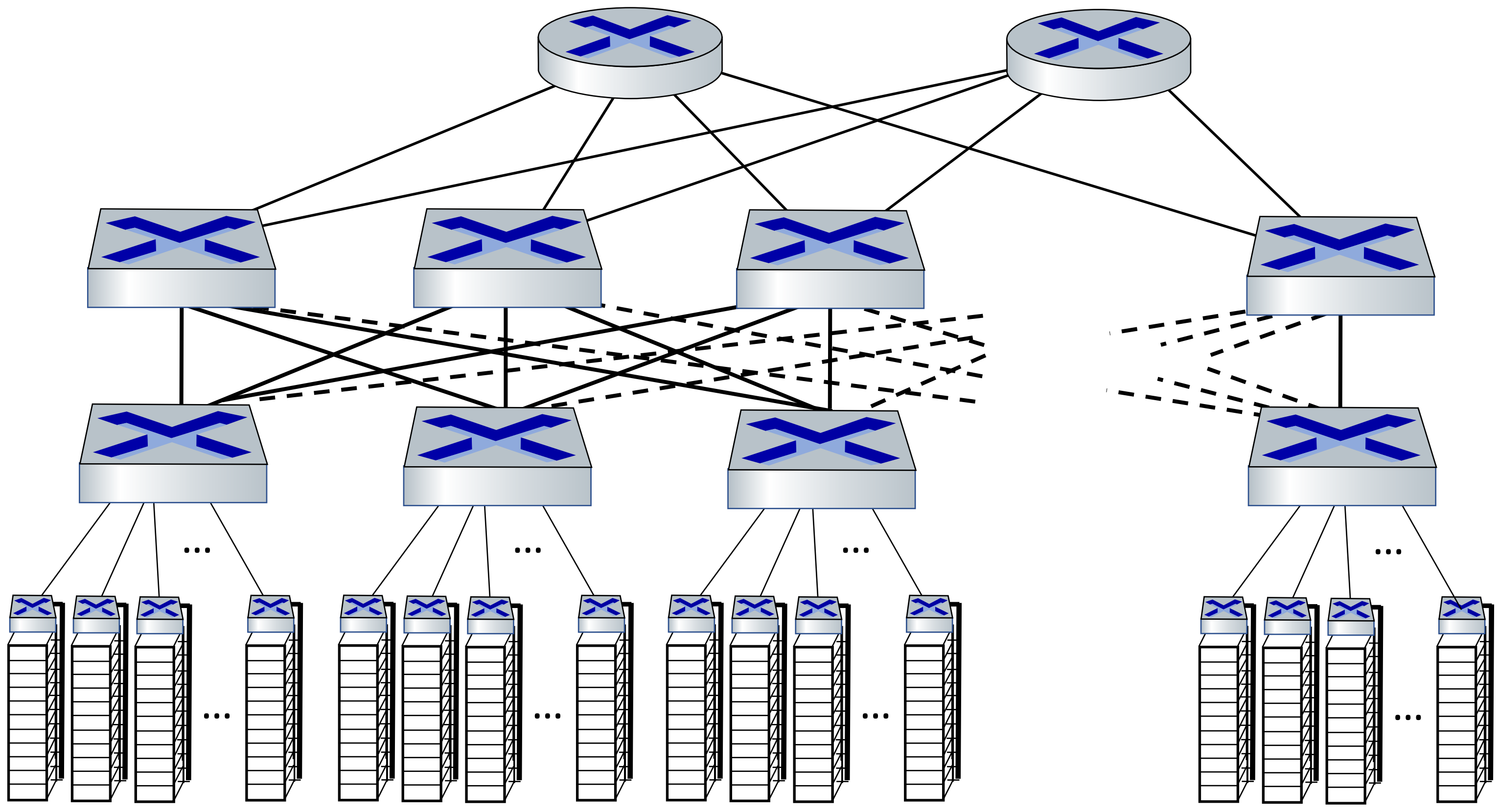
- e-business (e.g. Amazon) & cloud computing (AWS, Azure, GCP)
- content-servers (e.g., YouTube, Akamai, Apple, Microsoft)
- search engines, data mining (e.g., Google)

challenges:

- multiple applications, each serving massive numbers of clients
- reliability
- managing/balancing load, avoiding processing, networking, data bottlenecks

Inside a 40-ft Microsoft container, Chicago data center

Datacenter networks: network elements



Border routers
- connections outside datacenter

Tier-1 switches
- connecting to ~16 T-2s below

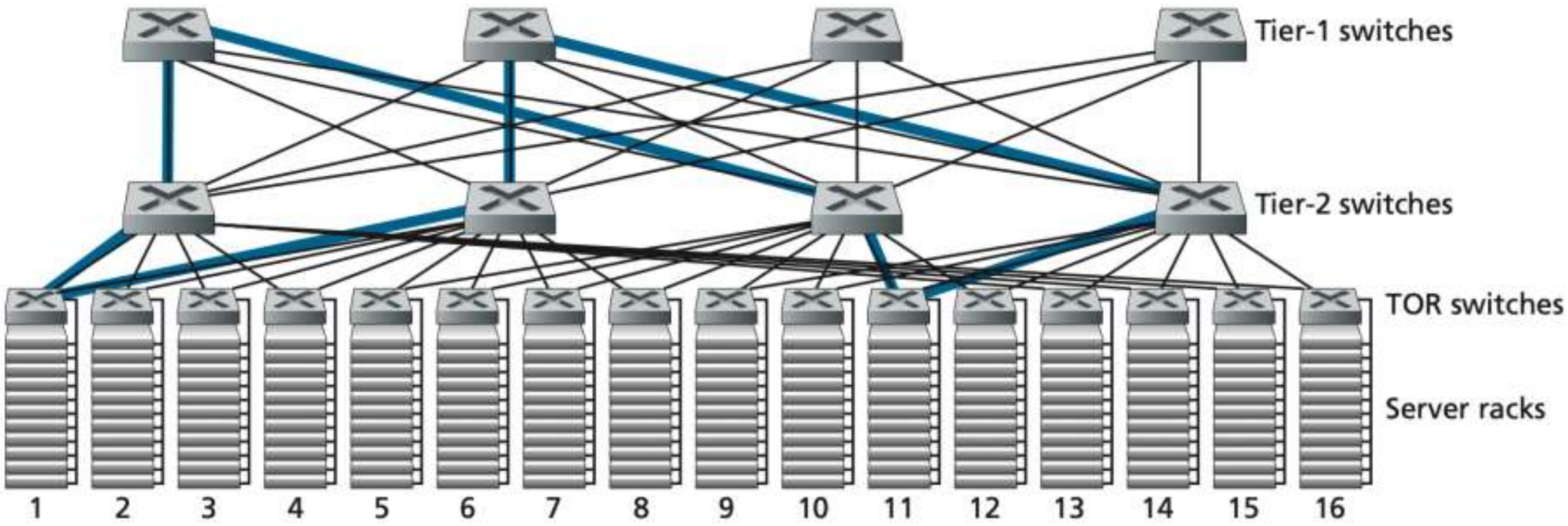
Tier-2 switches
- connecting to ~16 TORs below

Top of Rack (TOR) switch
- one per rack
- 100G-400G Ethernet to blades

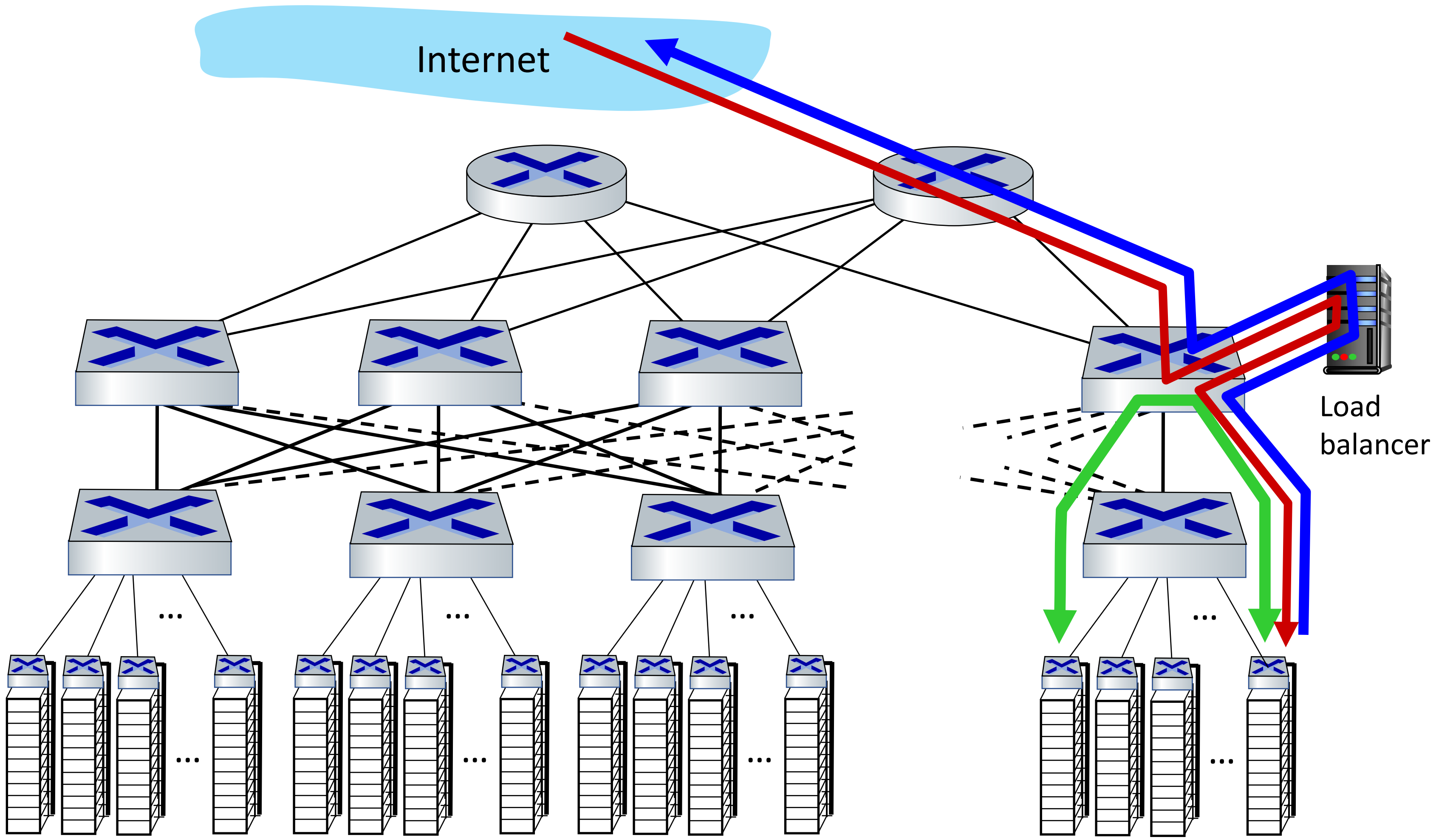
Server racks
- 20- 40 server blades: hosts

Datacenter networks: multipath

- rich interconnection among switches, racks: each Top of Rack switch is connected to 4 different tier-2 switches, and each tier-2 switch is connected to 4 different tier-1 switch
  - increased throughput between racks (multiple routing paths possible)
  - increased reliability via redundancy



Datacenter networks: Load Balancer



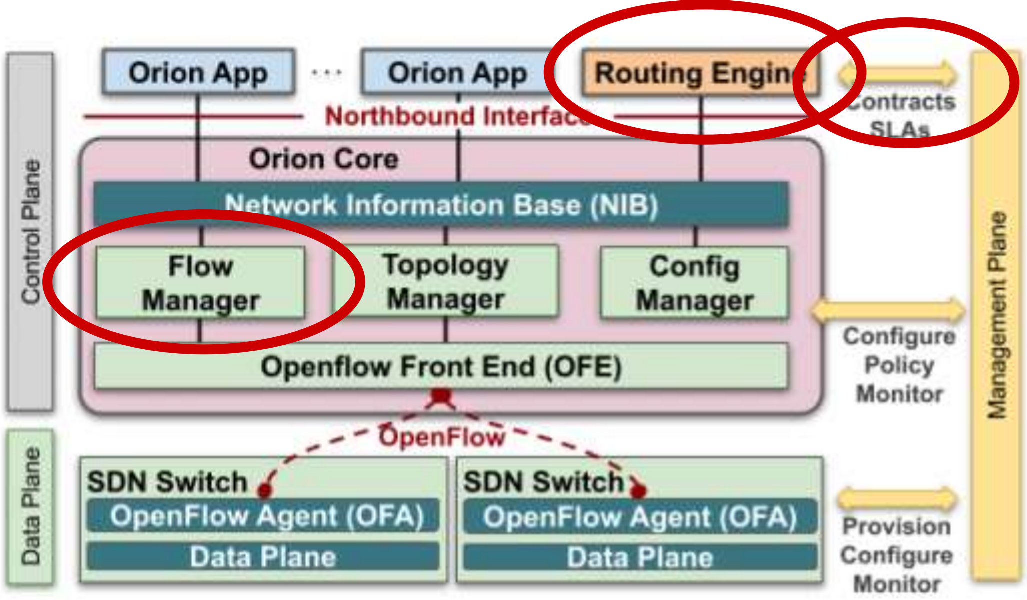
load balancer: layer-4 switch

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

ORION: Google's new SDN control plane for internal datacenter (Jupiter) + wide area (B4) network

- routing (intradomain, iBGP), traffic engineering: implemented in *applications* on top of ORION core
- edge-edge flow-based controls (e.g., CoFlow scheduling) to meet contract SLAs
- management: pub-sub distributed microservices in Orion core, OpenFlow for switch signaling/monitoring

Orion SDN architecture and core apps



Note:

- no routing protocols, congestion control (partially) also managed by SDN rather than by protocol
- are protocols dying?

Link layer, LANs: roadmap

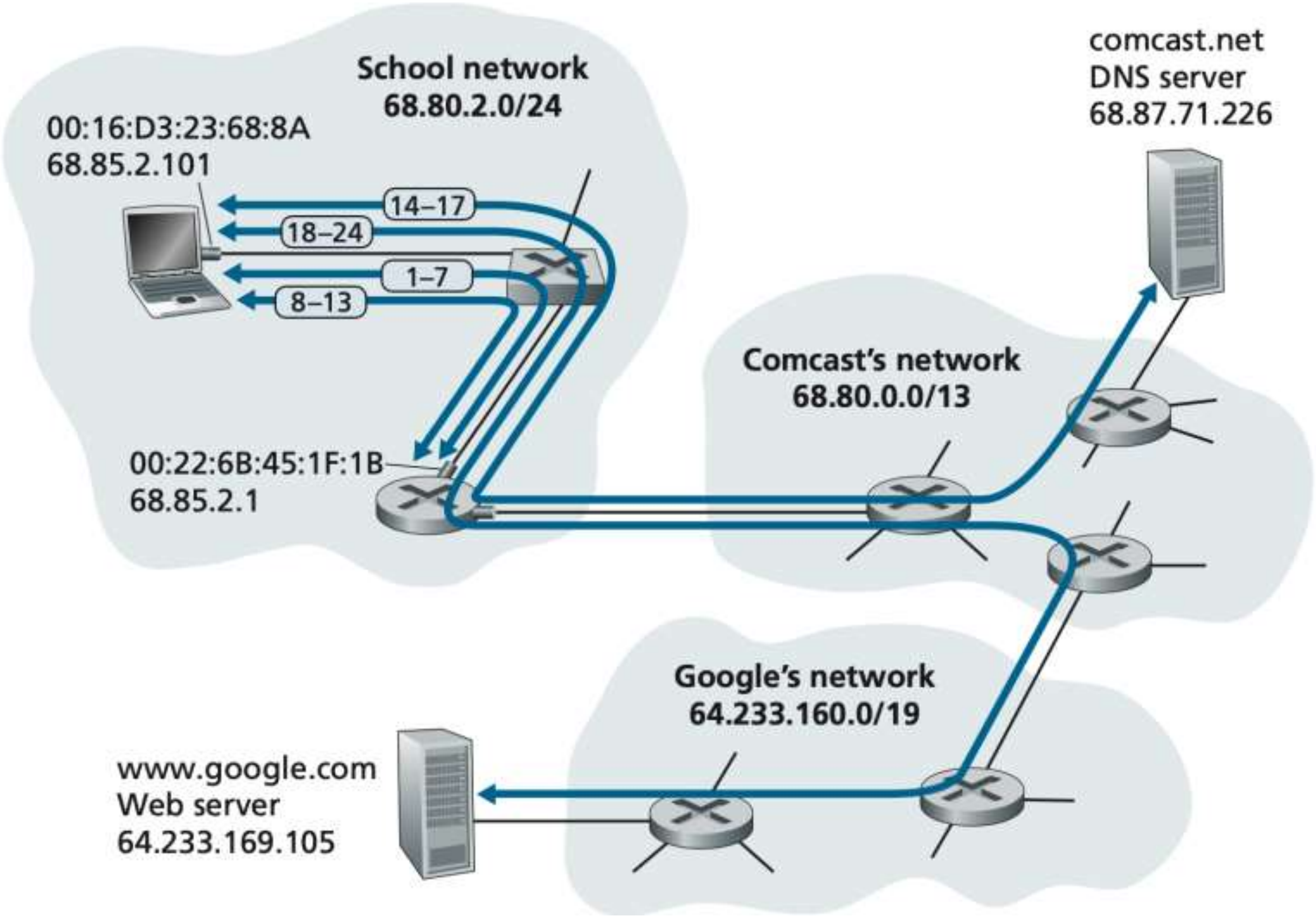
- introduction
- error detection, correction
- multiple access protocols
- LANs
  - addressing, ARP
  - Ethernet
  - switches
  - VLANs
- link virtualization: MPLS
- data center networking

- a day in the life of a web request

Synthesis: a day in the life of a web request

- our journey down the protocol stack is now 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

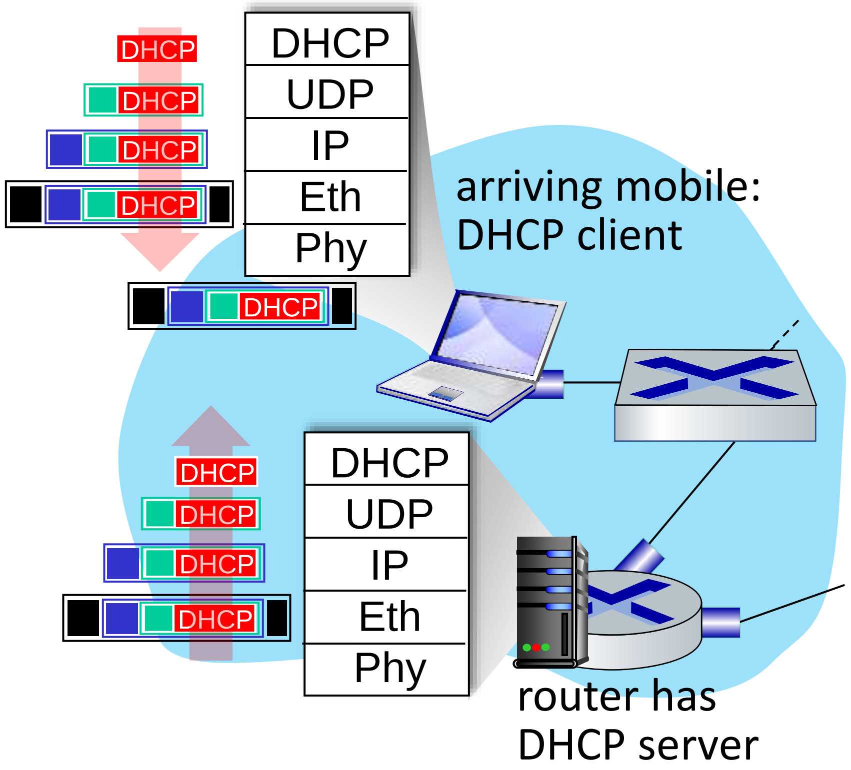
A day in the life: scenario



scenario:

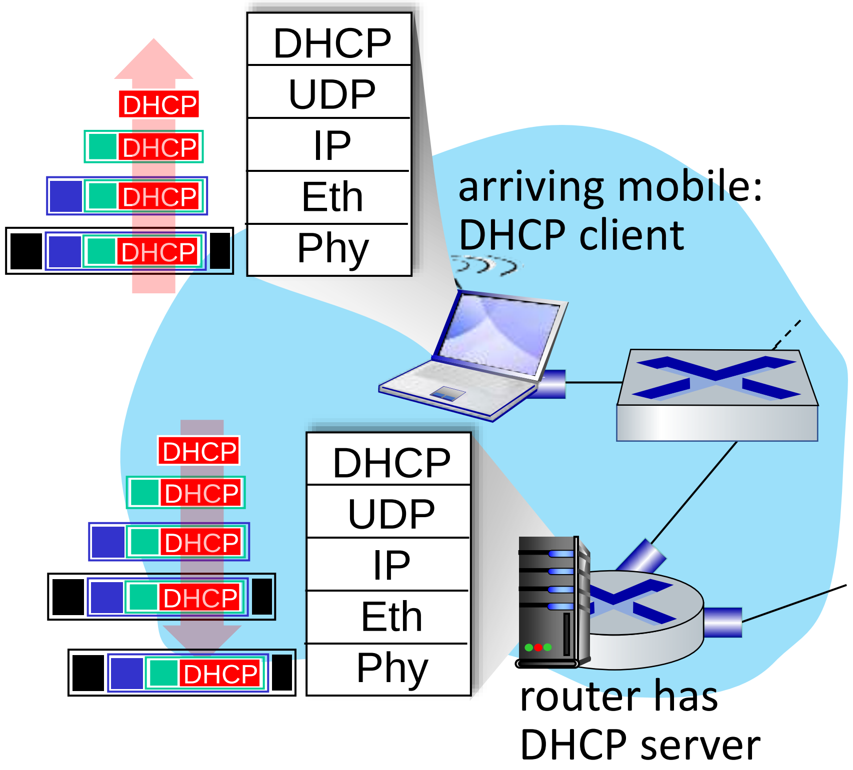
- requests web page: www.google.com

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 802.3 Ethernet
- Ethernet frame broadcast (dest: FFFFFFFFFFFF) on LAN, received at router running DHCP server
- Ethernet de-muxed to IP de-muxed, UDP de-muxed to DHCP

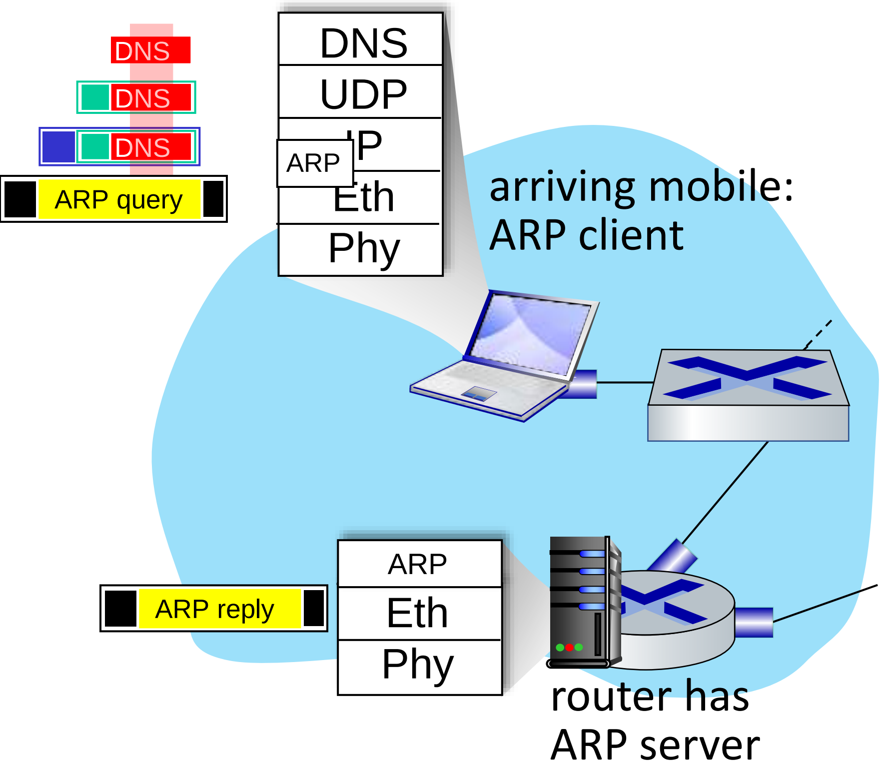
A day in the life: connecting to the Internet



- 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

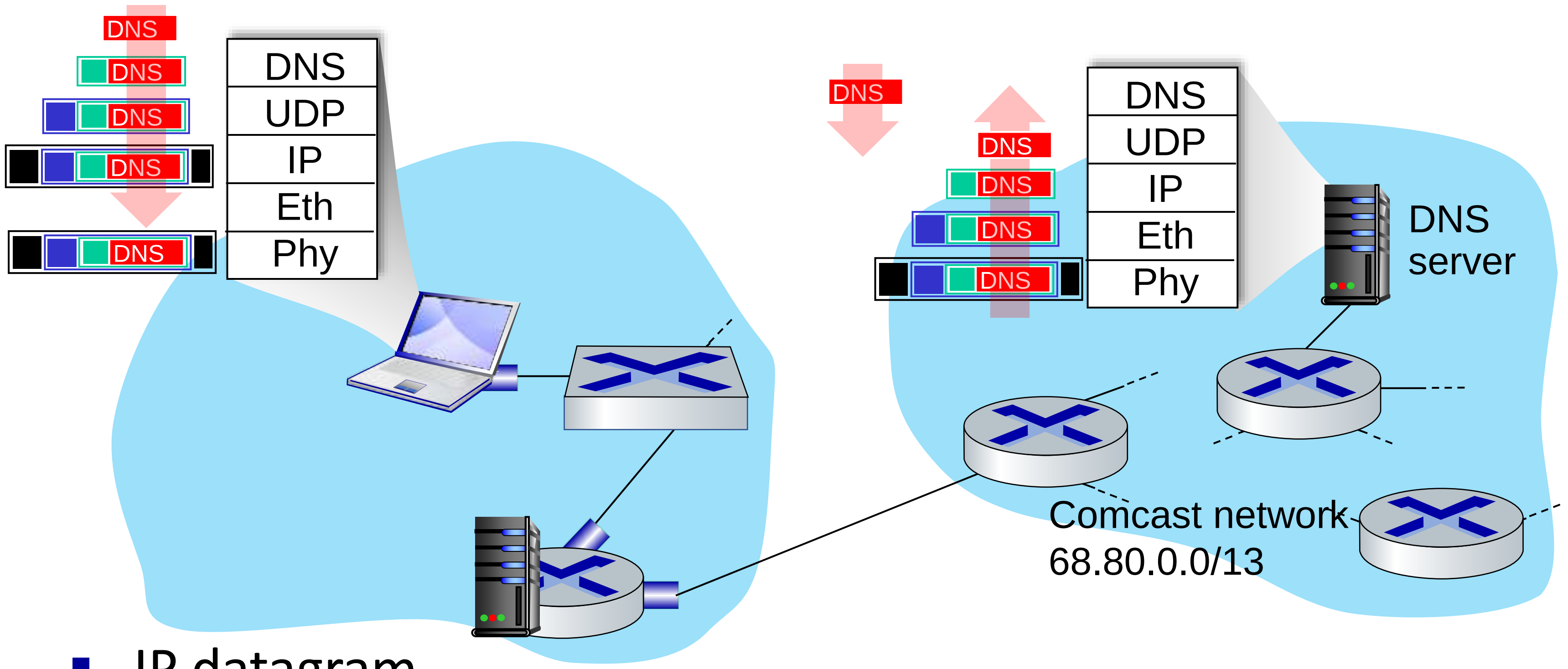
Client now has IP address, knows name & addr of DNS server, IP address of its first-hop router

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. 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

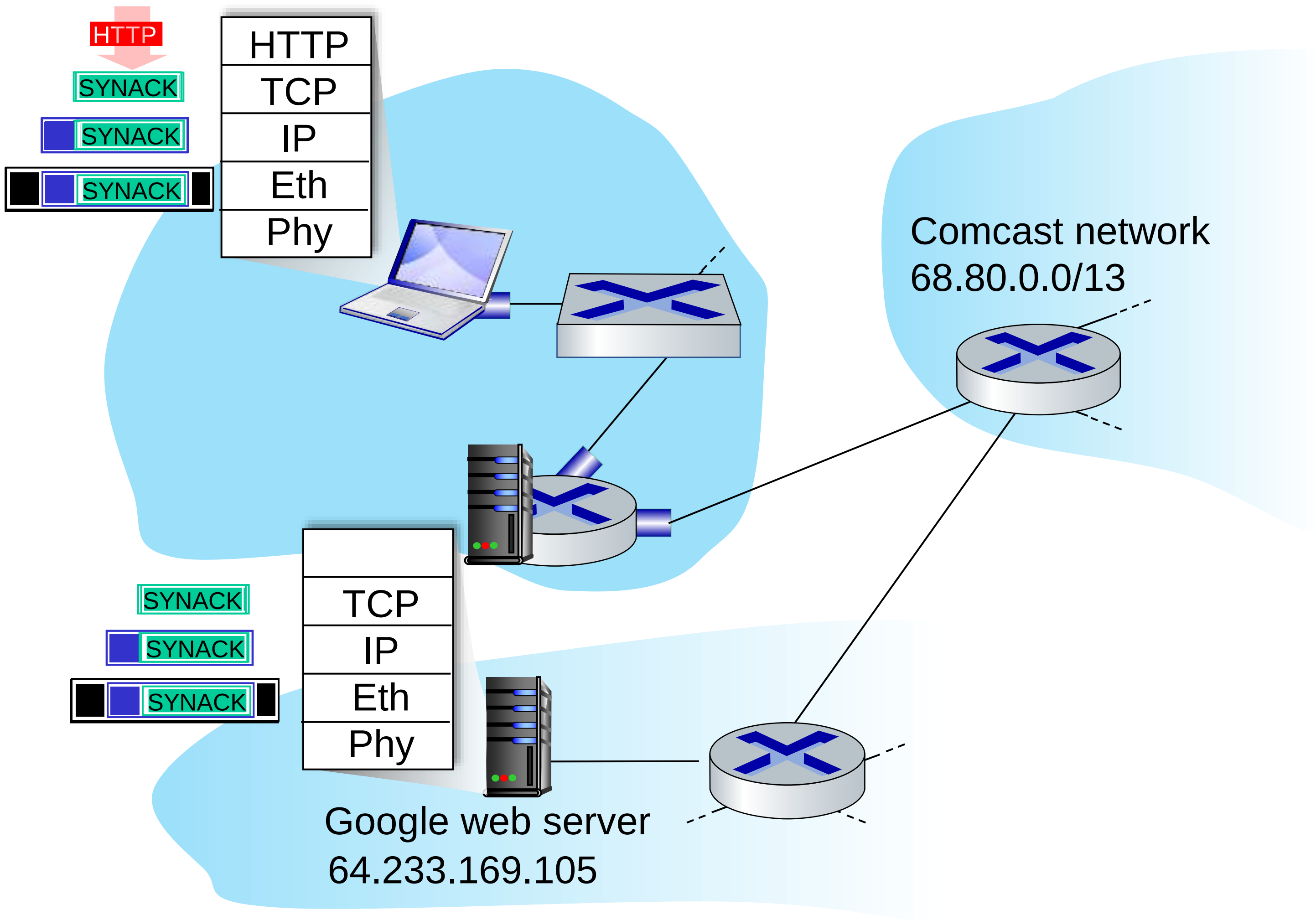


- de-muxed to DNS
- DNS replies to client with IP address of www.google.com

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

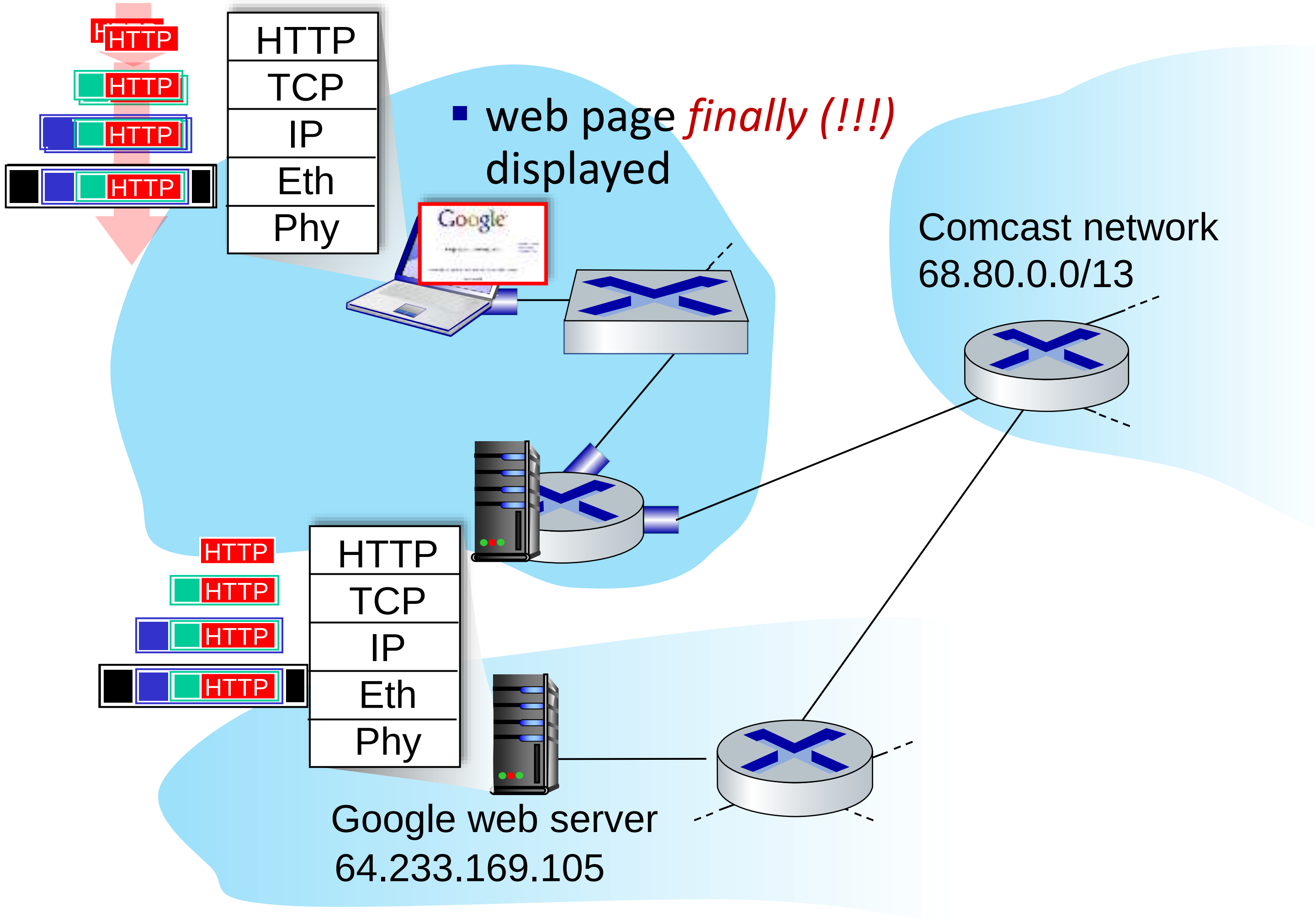
- IP datagram forwarded from campus network into Comcast network, routed (tables created by RIP, OSPF, IS-IS and/or BGP routing protocols) to DNS server

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 TCP 3-way handshake) inter-domain routed to web server
- web server responds with TCP SYNACK (step 2 in TCP 3-way handshake)
- TCP connection established!

A day in the life... HTTP request/reply



- 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

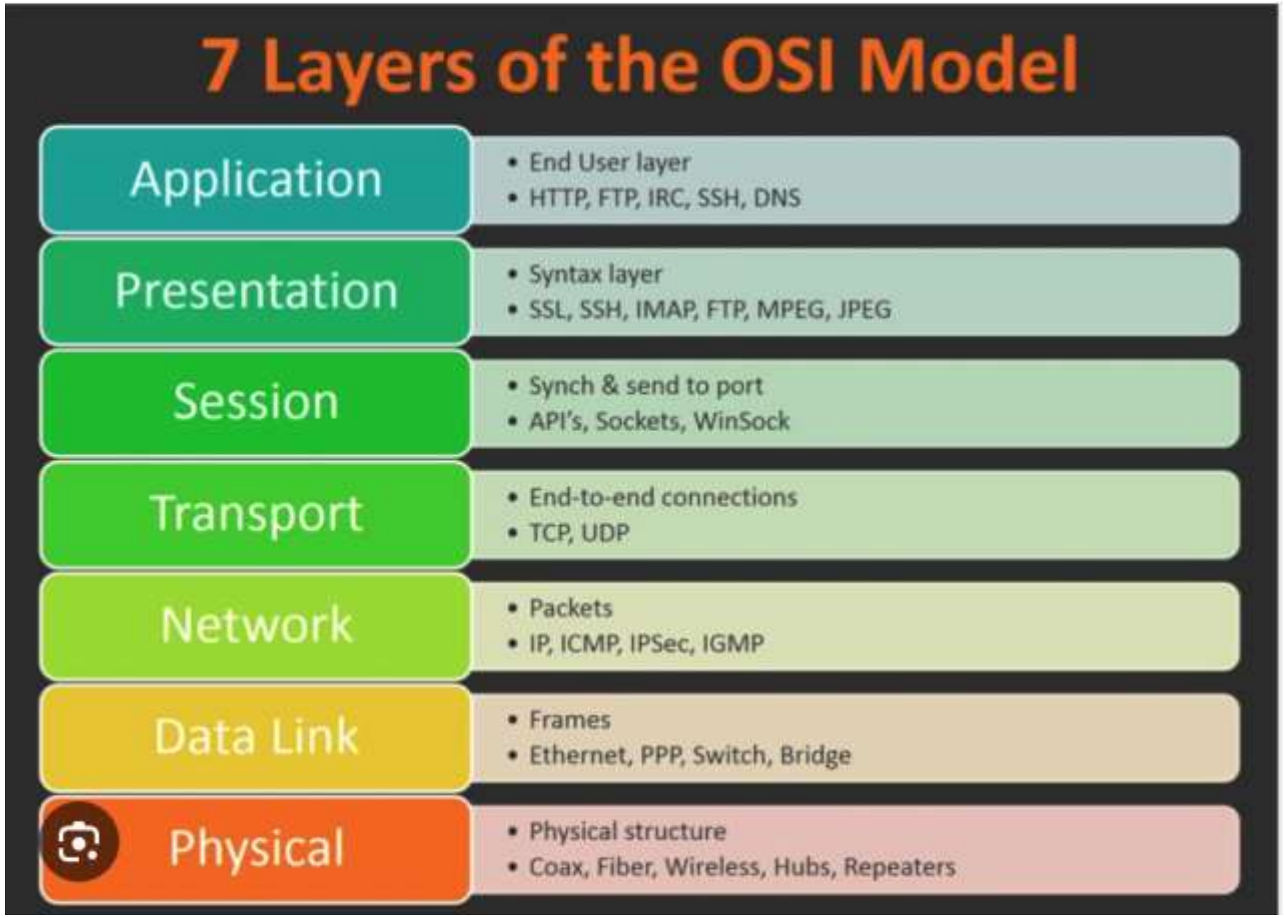
Chapter 6: Summary

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

Chapter 6: let's take a breath

- journey down protocol stack *complete* (except PHY)
- solid understanding of networking principles, practice!
- ….. could stop here …. but *more* interesting topics!
  - wireless
  - security

OSI 7-layer Model



Additional Chapter 6 slides

Pure ALOHA efficiency

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!

Cyclic Redundancy Check (CRC): example

Sender wants to compute R such that:

$D \cdot 2^r \;\; XOR \;\; R = nG$

... or equivalently (XOR R both sides):

$D \cdot 2^r = nG \;\; XOR \;\; R$

... which says:

if we divide $D \cdot 2^r$ by G, we want remainder R to satisfy:

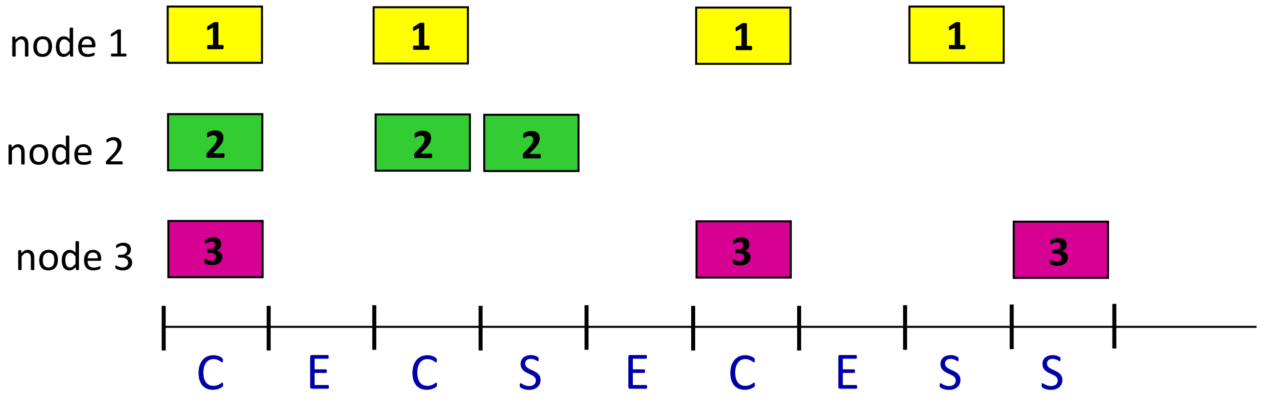
$$R = remainder\left[\frac{D \cdot 2^r}{G}\right]$$

algorithm for computing R

```
     G                      1 0 1 0 1 1
                      _________________
  1 0 0 1           ) 1 0 1 1 1 0 0 0 0      D*2^r  (here, r=3)
                      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 0
                          1 0 0 1
                            1 0 1 0
                            1 0 0 1
                              0 1 1
                                R
```

* Check out the online interactive exercises for more examples: http://gaia.cs.umass.edu/kurose_ross/interactive/

Slotted ALOHA



C: collision
S: success
E: empty

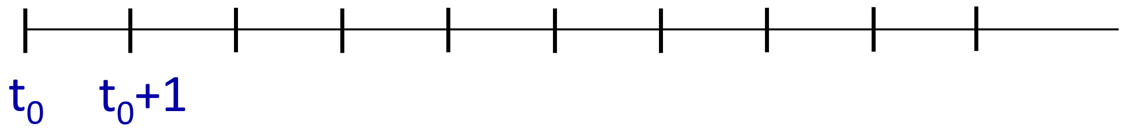
Pros:

- 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



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 probability *p* until success

randomization – *why*?

Slotted ALOHA: efficiency

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

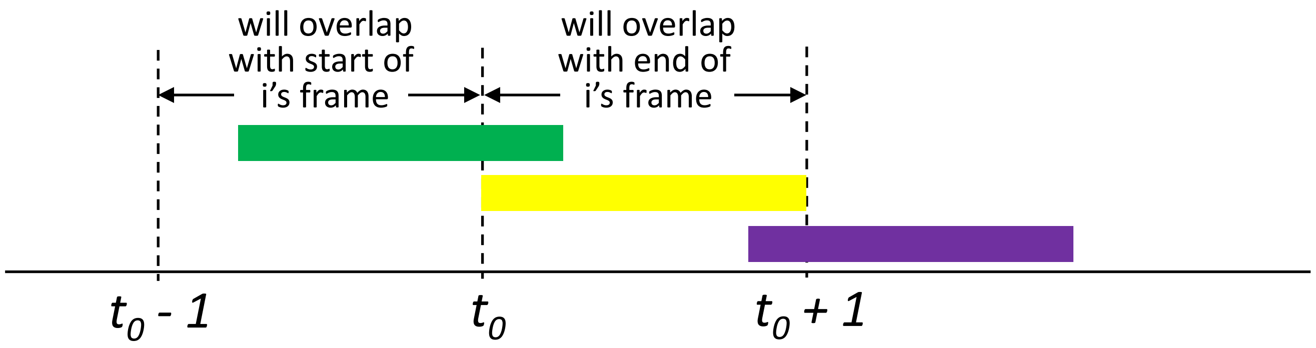
- *suppose:* N nodes with many frames to send, each transmits in slot with probability p
  - prob that given node has success in a slot $= p(1-p)^{N-1}$
  - prob that *any* node has a success $= Np(1-p)^{N-1}$
  - max efficiency: find p^* that maximizes $Np(1-p)^{N-1}$
  - for many nodes, take limit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

Pure ALOHA

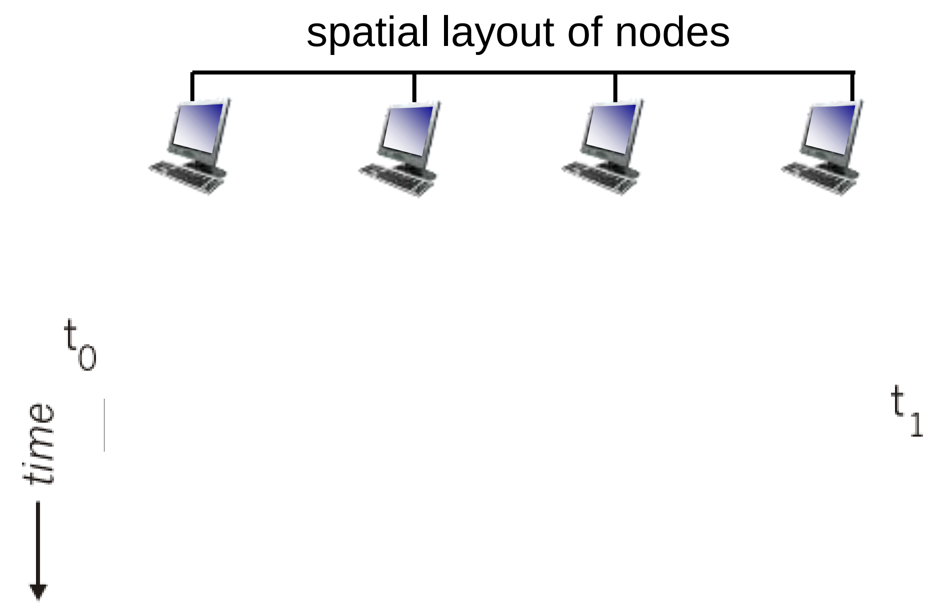
- unslotted Aloha: simpler, no synchronization
  - when frame first arrives: transmit immediately
- collision probability increases with no synchronization:
  - frame sent at t_0 collides with other frames sent in $[t_0-1,t_0+1]$



- pure Aloha efficiency: 18% !

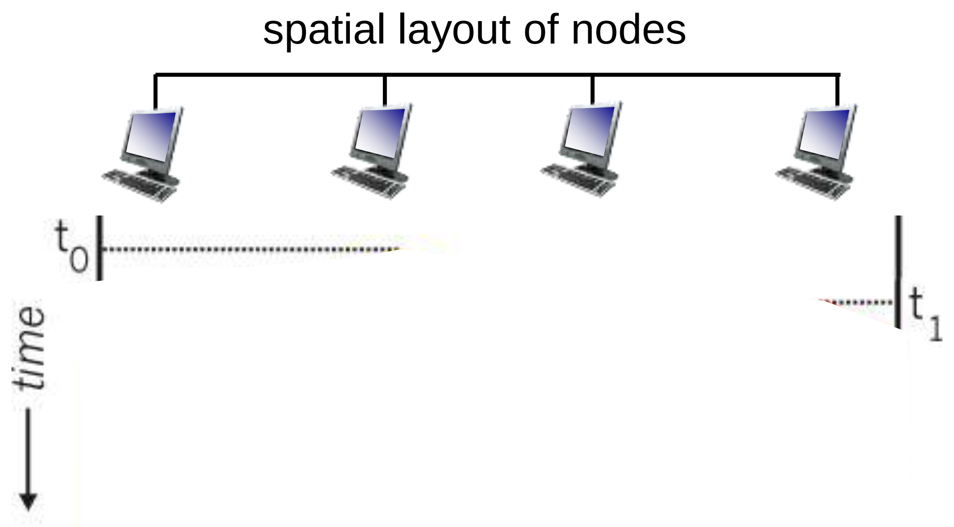
CSMA: collisions

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



CSMA/CD:

- CSMA/CD reduces the amount of time wasted in collisions
  - transmission aborted on collision detection



Ethernet CSMA/CD algorithm

1. Ethernet receives datagram from network layer, creates frame
2. If Ethernet senses channel:

   if idle: start frame transmission.

   if busy: wait until channel idle, then transmit
3. If entire frame transmitted without collision - done!
4. If another transmission detected while sending: abort, send jam signal
5. After aborting, enter *binary (exponential) backoff:*
   - after *m*th collision, chooses *K* at random from $\{0,1,2, \ldots, 2^m-1\}$. Ethernet waits $K \cdot 512$ bit times, returns to Step 2
   - more collisions: longer backoff interval

CSMA/CD efficiency

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

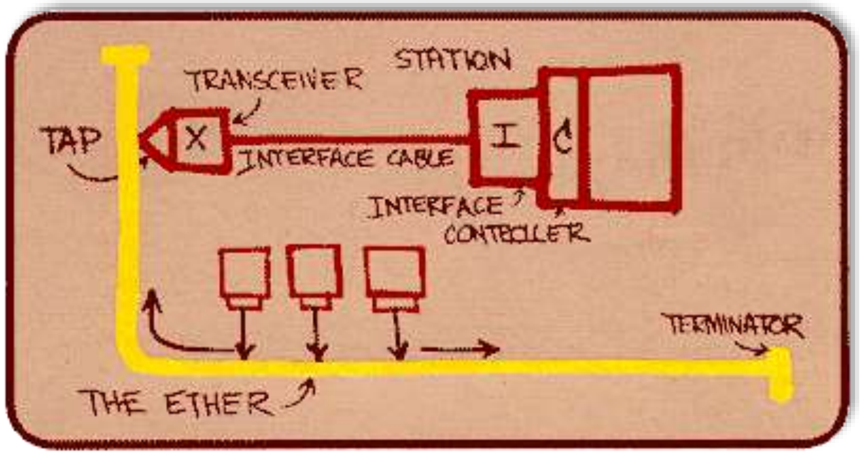
$$efficiency = \frac{1}{1 + 5t_{prop}/t_{trans}}$$

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

Ethernet

"dominant" wired LAN technology:
- first widely used LAN technology
- simpler, cheap
- kept up with speed race: 10 Mbps – 400 Gbps
- single chip, multiple speeds (e.g., Broadcom BCM5761)

Metcalfe's Ethernet sketch



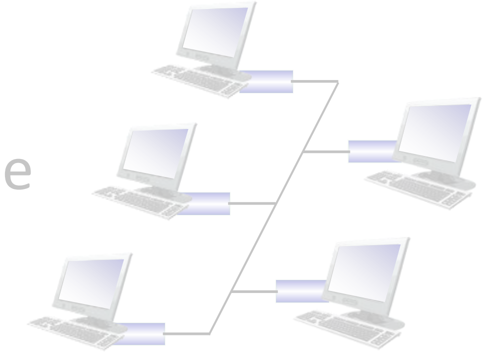
Bob Metcalfe: Ethernet co-inventor, 2022 ACM Turing Award recipient

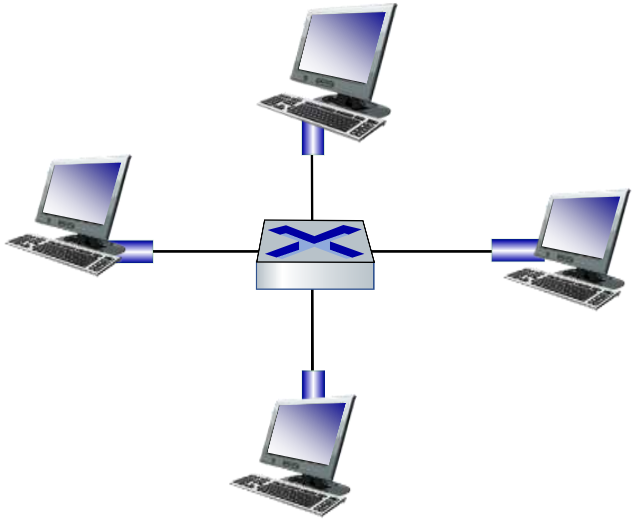
https://www.uspto.gov/learning-and-resources/journeys-innovation/audio-stories/defying-doubters

Ethernet: physical topology

- bus: popular through mid 90s
  - all nodes in same collision domain (can collide with each other)
- switched: prevails today
  - active link-layer 2 *switch* in center
  - each "spoke" runs a (separate) Ethernet protocol (nodes do not collide with each other)

bus: coaxial cable





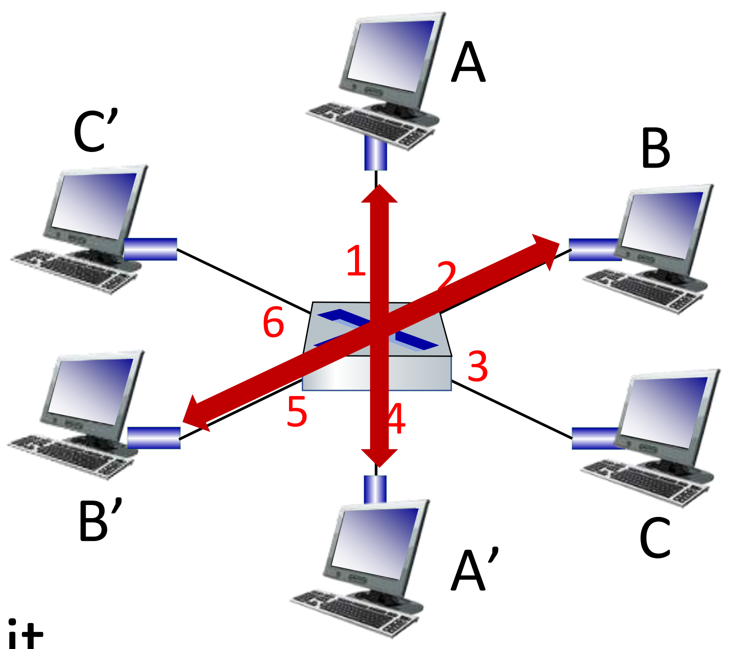
switched

Ethernet switch

- Switch is a link-layer device: takes an *active* role
  - store, forward Ethernet (or other type of) 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 *unaware* of presence of switches
- plug-and-play, self-learning
  - switches do not need to be configured

Switch: multiple simultaneous transmissions

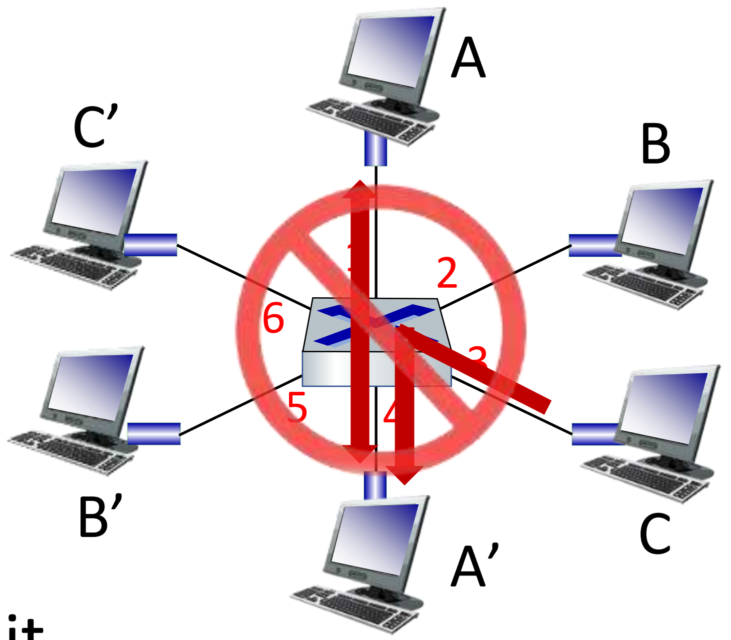
- hosts have dedicated, direct connection to switch
- switches buffer packets
- Ethernet protocol used on *each* incoming link, so:
  - no collisions; full duplex
  - each link is its own collision domain
- switching: A-to-A' and B-to-B' can transmit simultaneously, without collisions



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

Switch: multiple simultaneous transmissions

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



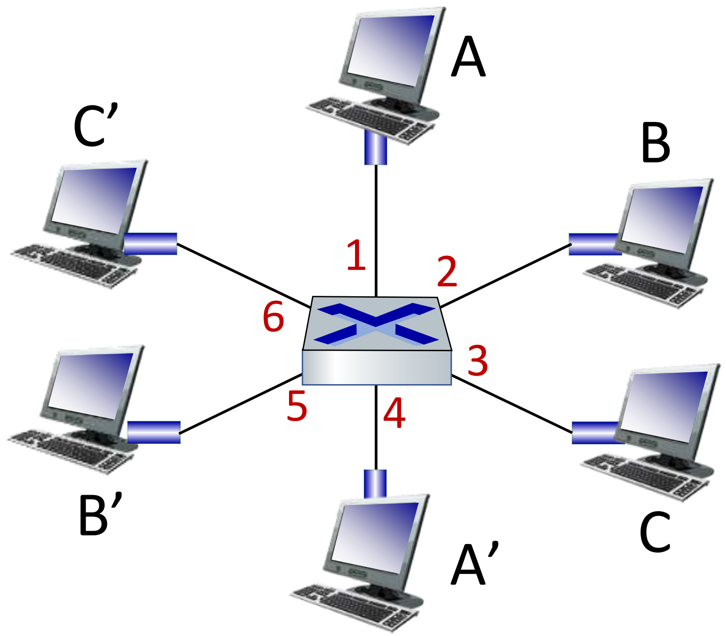
switch with six interfaces (1,2,3,4,5,6)

Switch forwarding table

Q: how does switch know A' reachable via interface 4, B' reachable via interface 5?

A: each switch has a switch table, each entry:

- (MAC address of host, interface to reach host, time stamp)
- looks like a routing table!

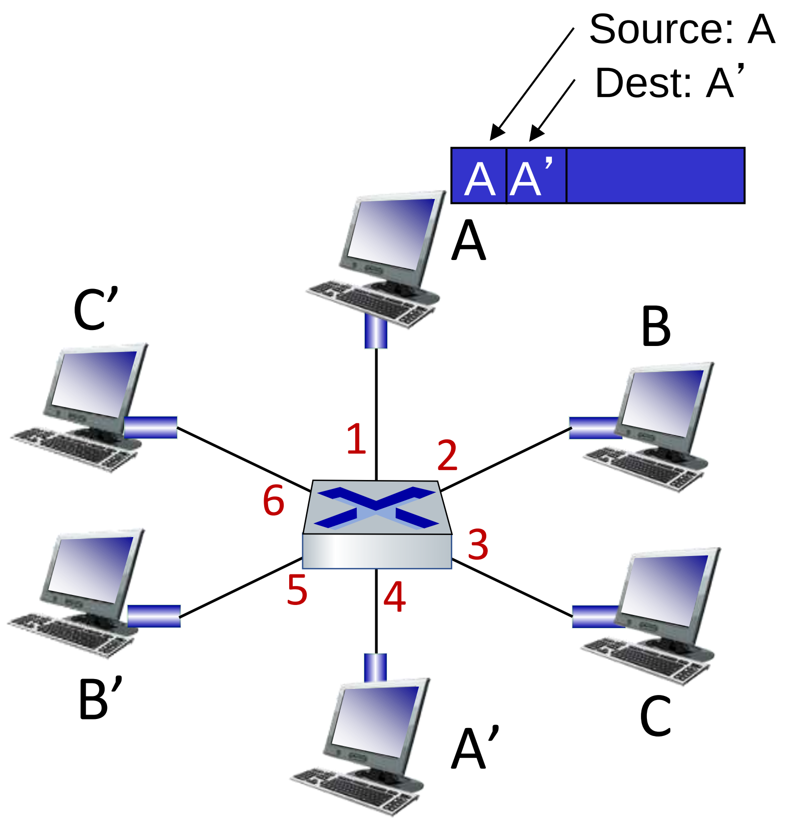


Q: how are entries created, maintained in switch table?

- something like a routing protocol?

Switch: self-learning

- switch *learns* which hosts can be reached through which interfaces
  - when frame received, switch "learns" location of sender: incoming LAN segment
  - records sender/location pair in switch table



MAC addr	interface	TTL
A	*1*	*60*

Switch table (initially empty)

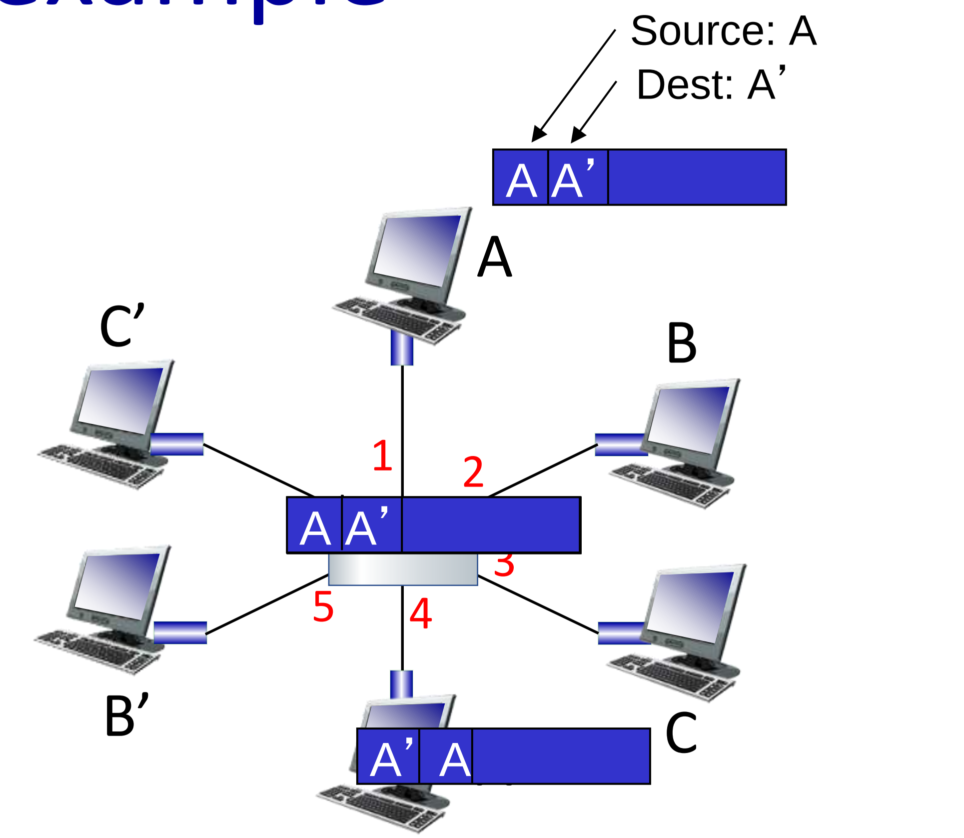
Switch: frame filtering/forwarding

when frame received at switch:

1. record incoming link, MAC address of sending host
2. index switch table using MAC destination address
3. if entry found for destination
   then {
   if destination on segment from which frame arrived
      then drop frame
      else forward frame on interface indicated by entry
   }
   else flood /* forward on all interfaces except arriving interface */

Self-learning, forwarding: example

- frame destination, A’, location unknown: flood
- destination A location known: selectively send on just one link

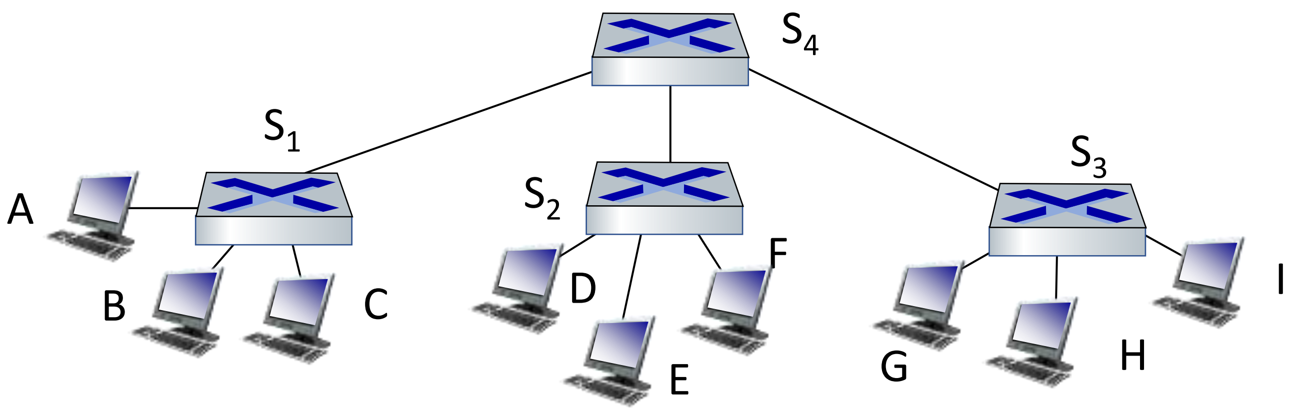


MAC addr	interface	TTL
A	*1*	*60*
A’	*4*	*60*

switch table (initially empty)

Interconnecting switches

self-learning switches can be connected together:

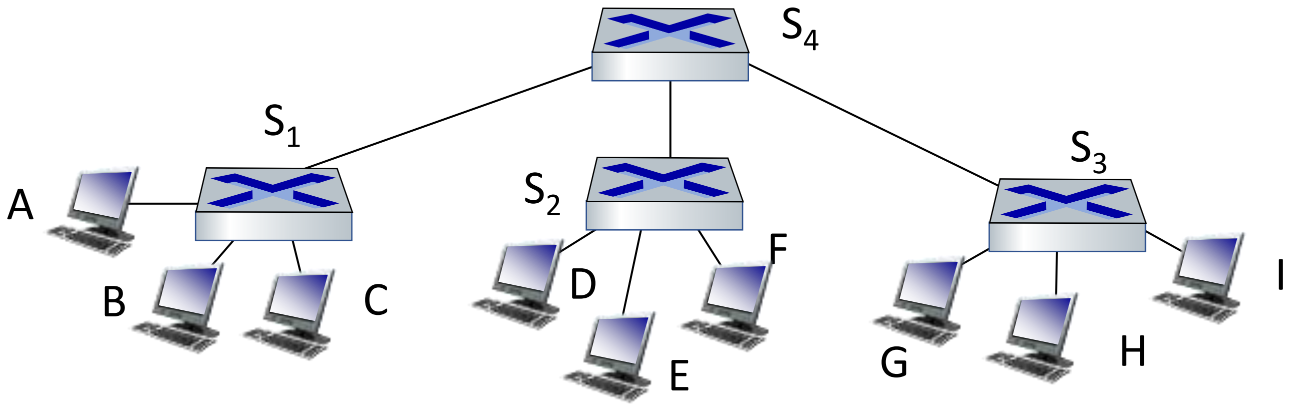


Q: sending from A to G - how does S_1 know to forward frame destined to G via S_4 and S_3?

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

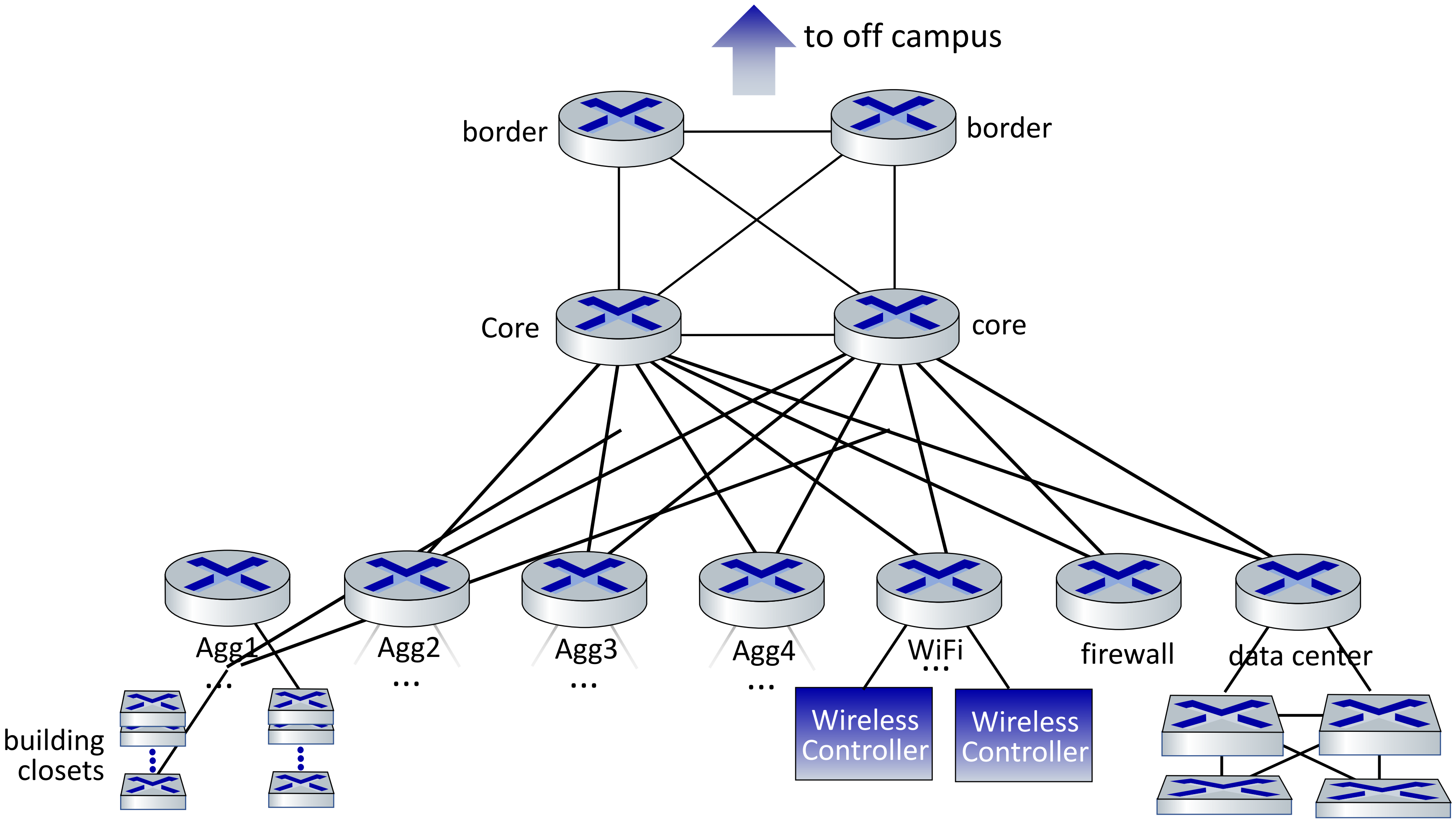
Self-learning multi-switch example

Suppose C sends frame to I, I responds to C



Q: show switch tables and packet forwarding in S_1, S_2, S_3, S_4

UMass Campus Network - Detail

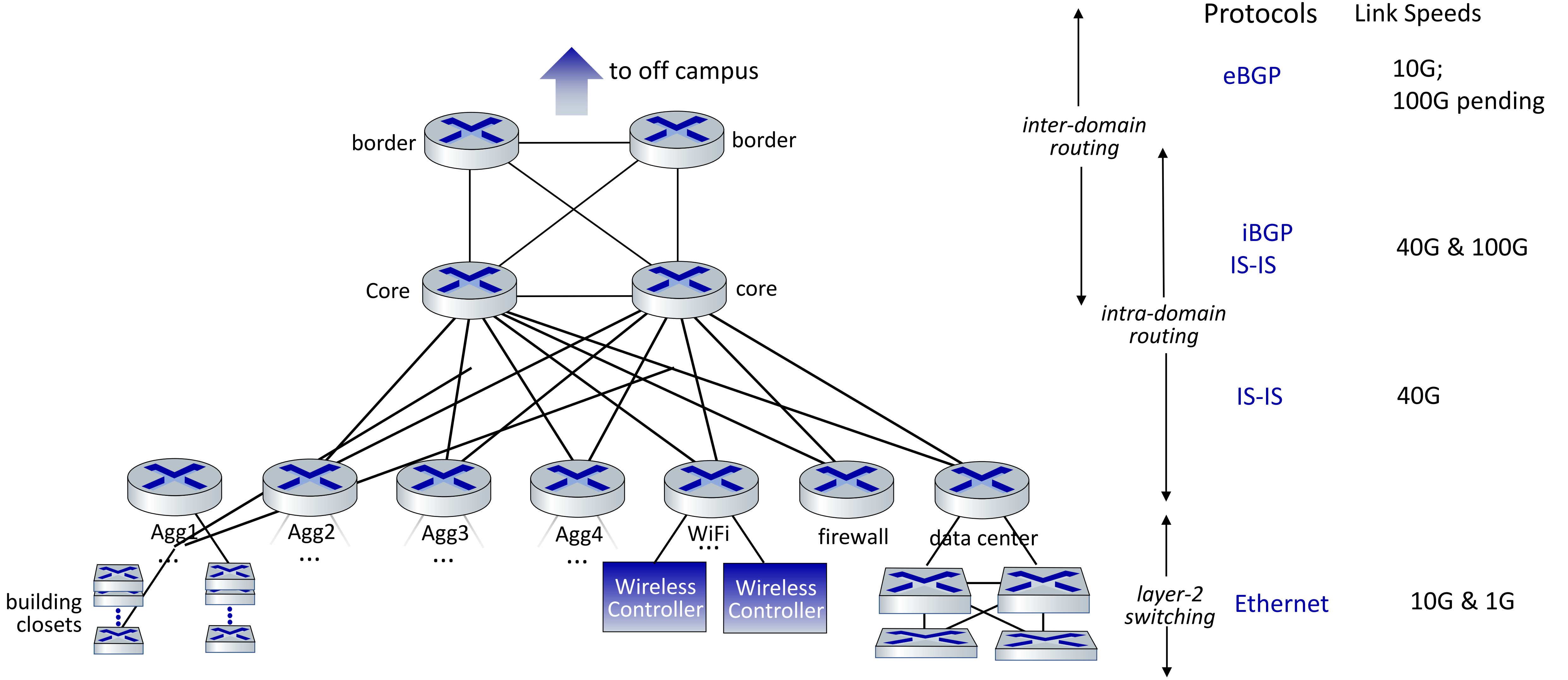


UMass network:

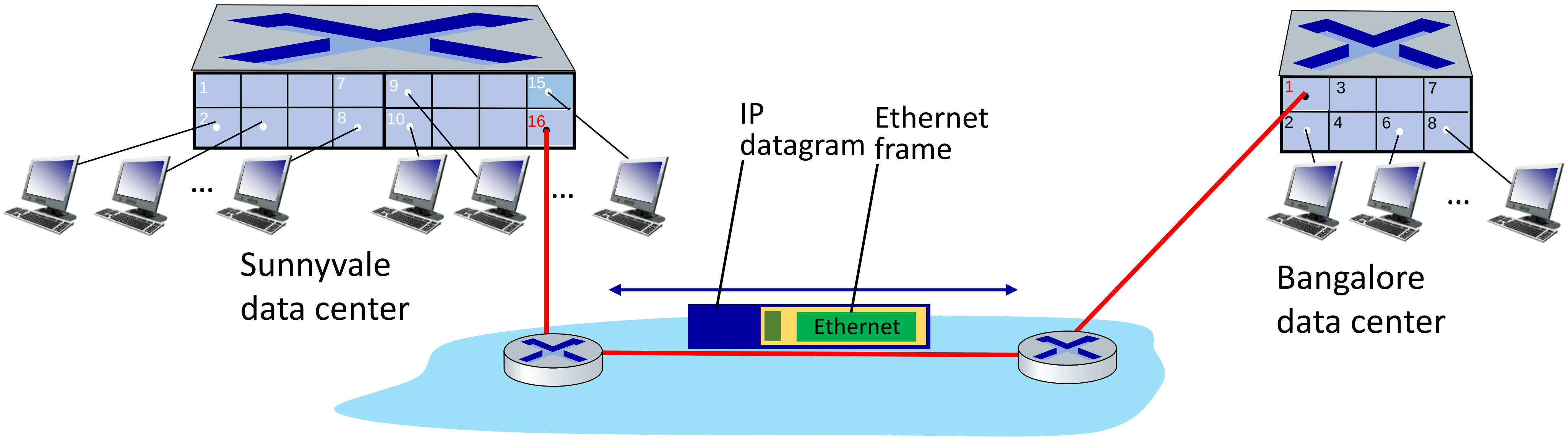
- 4 firewalls
- 10 routers
- 2000+ network switches
- 6000 wireless access points
- 30000 active wired network jacks
- 55000 active end-user wireless devices

... all built, operated, maintained by ~15 people

UMass Campus Network - Detail



EVPN: Ethernet VPNs (aka VXLANs)



Layer-2 Ethernet switches *logically* connected to each other (e.g., using IP as an *underlay*)

- Ethernet frames carried *within* IP datagrams between sites
- "*tunneling* scheme to *overlay Layer 2 networks on top of Layer 3 networks* ... runs over the existing networking infrastructure and provides a means to "stretch" a Layer 2 network." [RFC 7348]

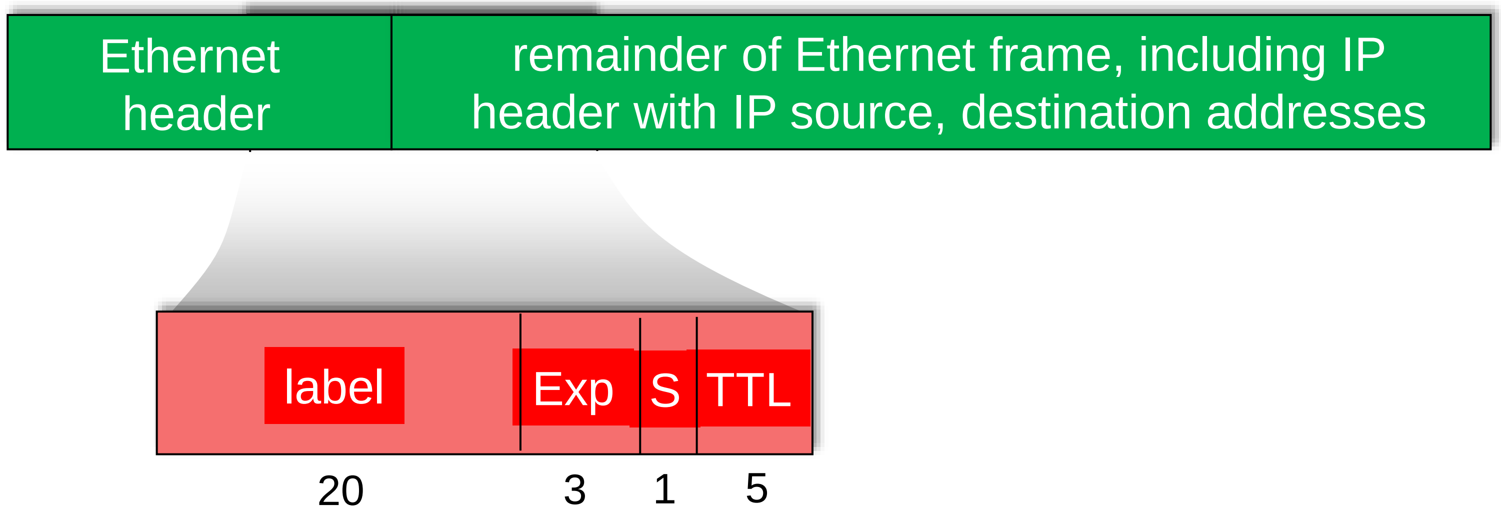
Link layer, LANs: roadmap

- introduction
- error detection, correction
- multiple access protocols
- LANs
  - addressing, ARP
  - Ethernet
  - switches
  - VLANs
- link virtualization: MPLS
- data center networking

- a day in the life of a web request

Multiprotocol label switching (MPLS)

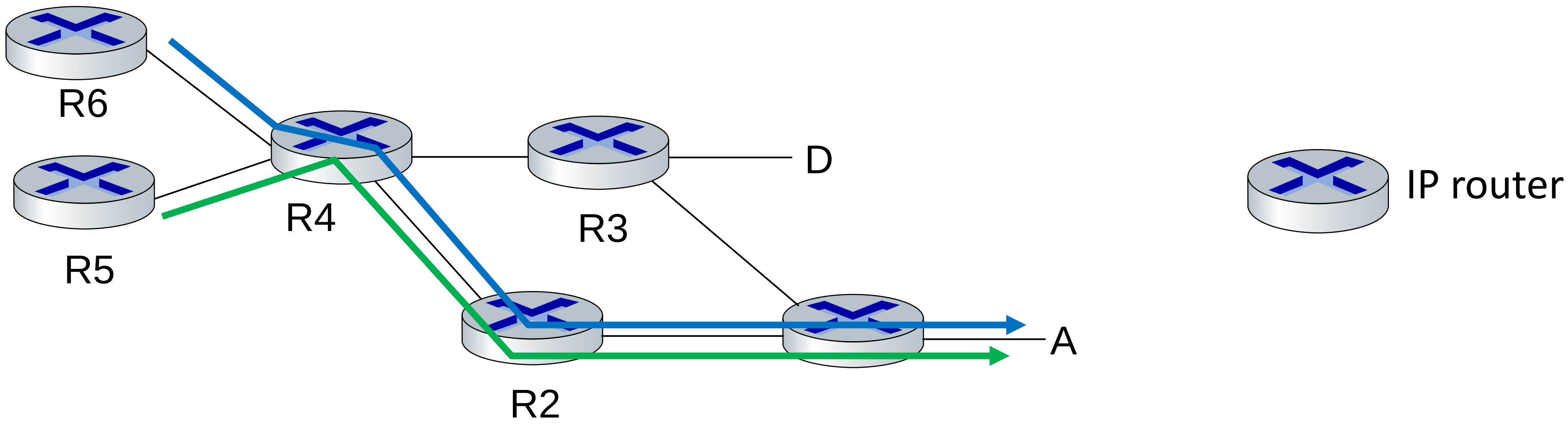
- goal: high-speed IP forwarding among network of MPLS-capable routers, using fixed length label (instead of shortest prefix matching)
  - faster lookup using fixed length identifier
  - borrowing ideas from Virtual Circuit (VC) approach
  - but IP datagram still keeps IP address!



MPLS capable routers

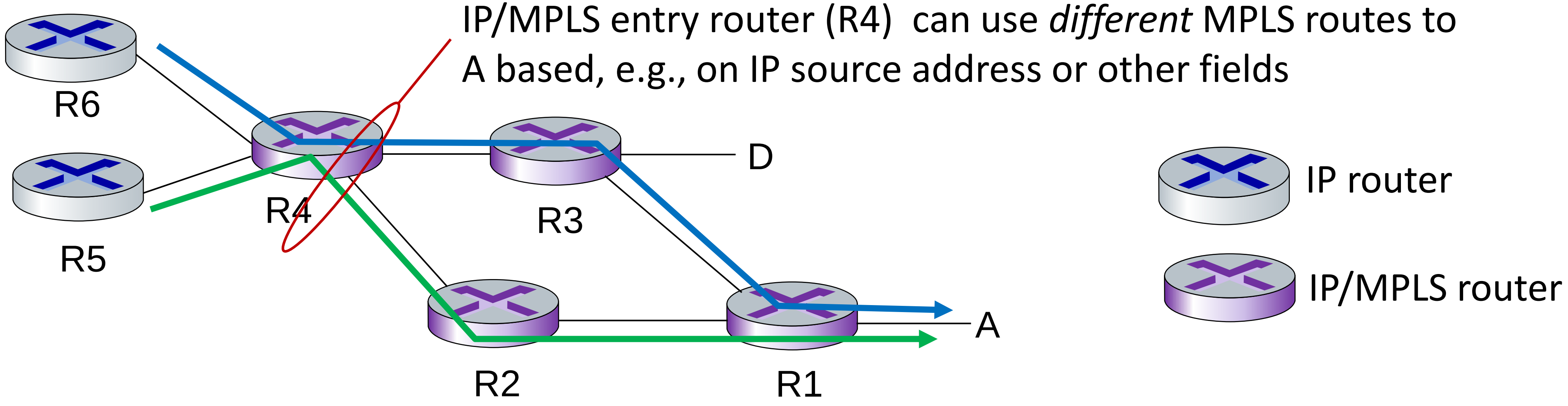
- a.k.a. label-switched router
- forward packets to outgoing interface based only on label value (*don't inspect IP address*)
  - MPLS forwarding table distinct from IP forwarding tables
- *flexibility:* MPLS forwarding decisions can *differ* from those of IP
  - use destination *and* source addresses to route flows to same destination differently (traffic engineering)
  - re-route flows quickly if link fails: pre-computed backup paths

MPLS versus IP paths



- IP routing: path to destination determined by destination address alone

MPLS versus IP paths

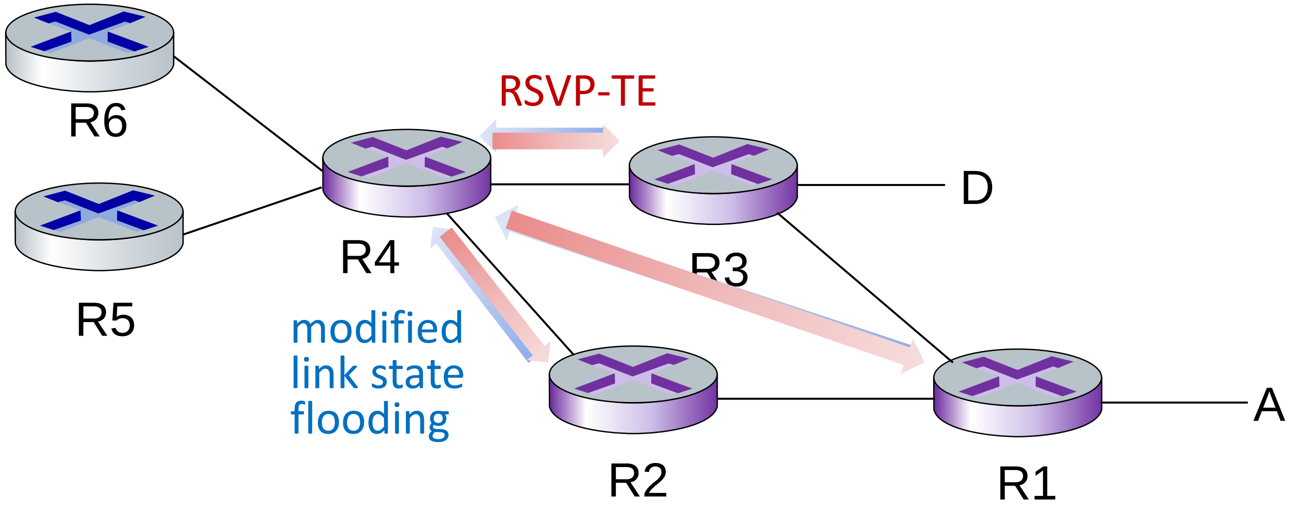


IP/MPLS entry router (R4) can use *different* MPLS routes to A based, e.g., on IP source address or other fields

- IP routing: path to destination determined by destination address alone
- MPLS routing: path to destination can be based on source *and* destination address
  - flavor of generalized forwarding (MPLS 10 years earlier)
  - *fast reroute:* precompute backup routes in case of link failure

MPLS signaling

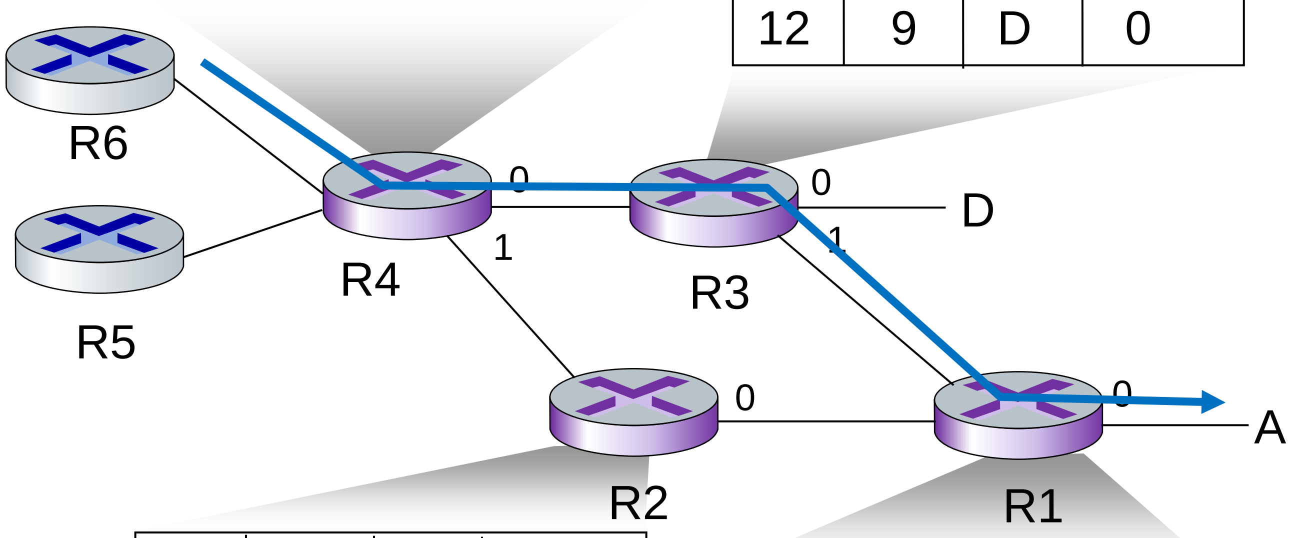
- modify OSPF, IS-IS link-state flooding protocols to carry info used by MPLS routing:
  - e.g., link bandwidth, amount of “reserved” link bandwidth
- entry MPLS router uses RSVP-TE signaling protocol to set up MPLS forwarding at downstream routers



MPLS forwarding tables

in label	out label	dest	out interface
	10	A	0
	12	D	0
	8	A	1

in label	out label	dest	out interface
10	6	A	1
12	9	D	0



in label	out label	dest	out interface
8	6	A	0

in label	out label	dest	out interface
6	-	A	0

Datacenter networks: protocol innovations

- link layer:
  - RoCE: remote DMA (RDMA) over Converged Ethernet
- transport layer:
  - ECN (explicit congestion notification) used in transport-layer congestion control (DCTCP, DCQCN)
  - experimentation with hop-by-hop (backpressure) congestion control
- routing, management:
  - SDN widely used within/among organizations' datacenters
  - place related services, data as close as possible (e.g., in same rack or nearby rack) to minimize tier-2, tier-1 communication

Google Networking: Infrastructure and Selected Challenges (Slides: https://networkingchannel.eu/google-networking-infrastructure-and-selected-challenges/