南京大學
NANJING UNIVERSITY

链路层和局域网

殷亚凤
智能软件与工程学院
苏州校区南雍楼东区225
yafeng@nju.edu.cn，https://yafengnju.github.io/

Outline

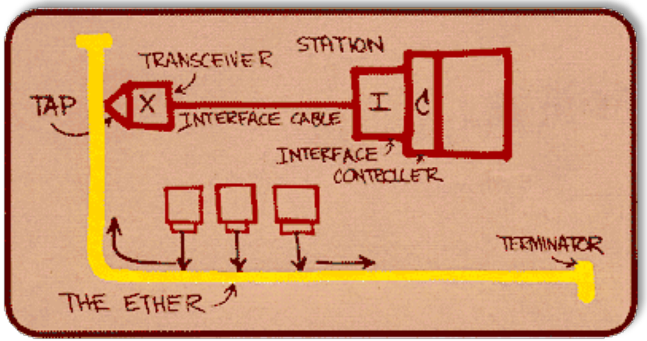
- Introduction
- Error detection, correction
- Multiple access protocols
- LANs
- Link virtualization: MPLS
- Data center networking
- A day in the life of a web request

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)

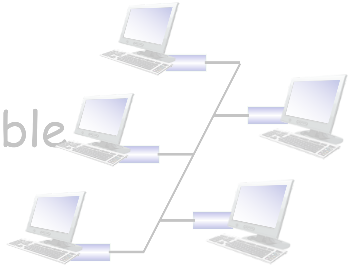
Metcalfe's Ethernet sketch



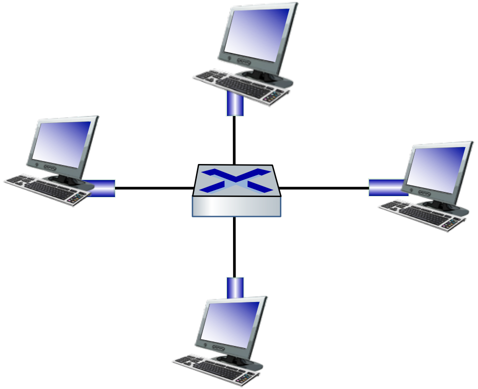
Bob Metcalfe: Ethernet co-inventor, 2022 ACM Turing Award recipient

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)



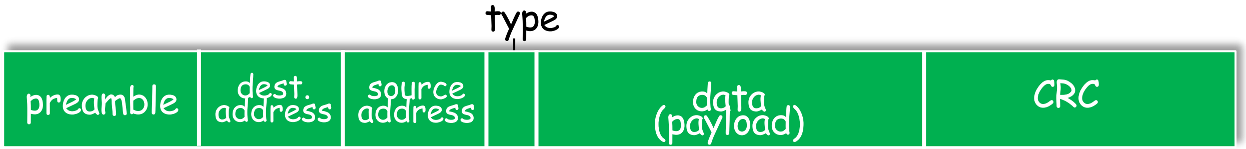
bus: coaxial cable



switched

Ethernet frame structure

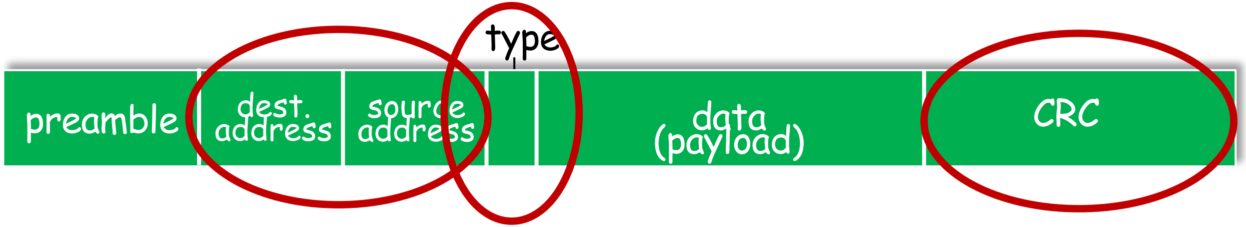
sending interface encapsulates IP datagram (or other network layer protocol packet) in Ethernet frame



preamble:

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

Ethernet frame structure (more)



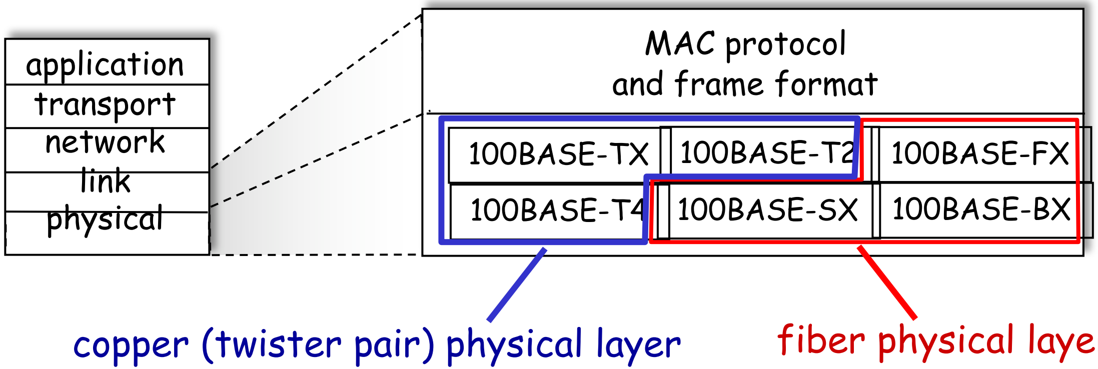
- **addresses:** 6 byte source, destination MAC addresses
  - if adapter receives frame with matching destination address, or with broadcast address (e.g., ARP packet), it passes data in frame to network layer protocol
  - otherwise, adapter discards frame
- **type:** indicates higher layer protocol
  - mostly IP but others possible, e.g., Novell IPX, AppleTalk
  - used to demultiplex up at receiver
- **CRC:** cyclic redundancy check at receiver
  - error detected: frame is dropped

Ethernet: unreliable, connectionless

- connectionless: no handshaking between sending and receiving NICs
- unreliable: receiving NIC doesn't send ACKs or NAKs to sending NIC
    - data in dropped frames recovered only if initial sender uses higher layer rdt (e.g., TCP), otherwise dropped data lost
- Ethernet's MAC protocol: unslotted CSMA/CD with binary backoff

802.3 Ethernet standards: link & physical layers

- many different Ethernet standards
  - common MAC protocol and frame format
  - different speeds: 2 Mbps, ... 100 Mbps, 1Gbps, 10 Gbps, 40 Gbps, 80 Gbps
    - ✓ different physical layer media: fiber, cable

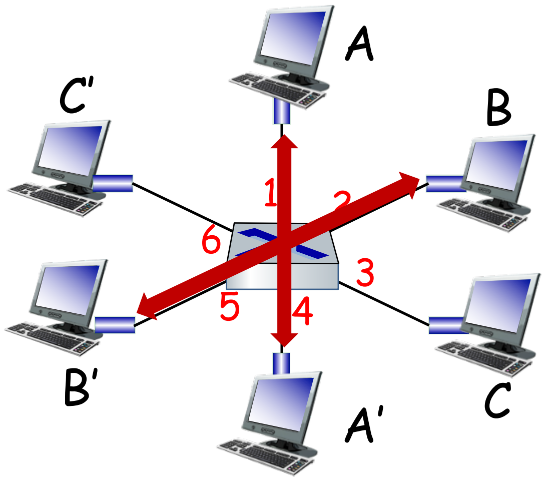


Ethernet switch

- 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

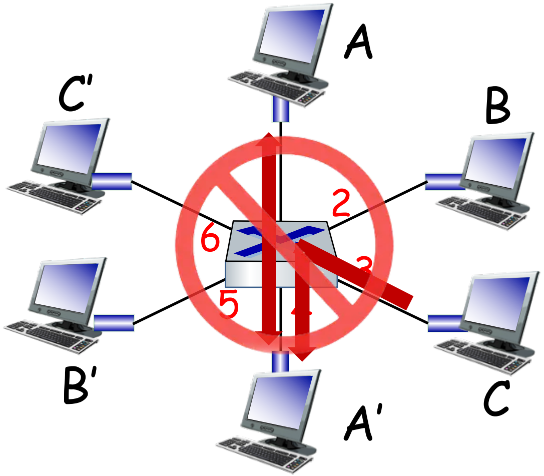
- hosts have dedicated, direct connection to switch
- switches buffer packets
- Ethernet protocol used on each incoming link, so:
  - no collisions; full duplex
  - each link is its own collision domain
- switching: A-to-A' and B-to-B' can transmit simultaneously, without collisions



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

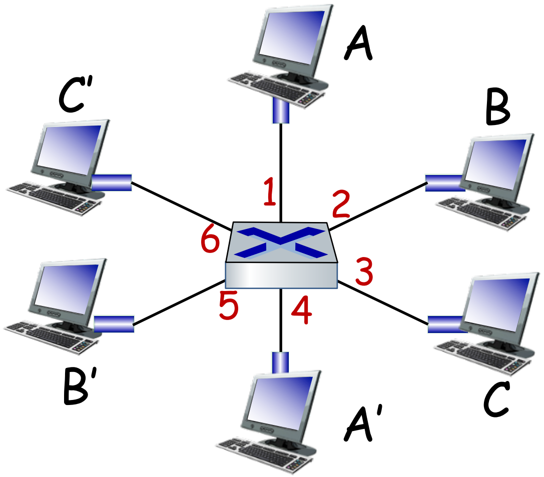
Q: how does switch know A′ reachable via interface 4, B′ reachable via interface 5?

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

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

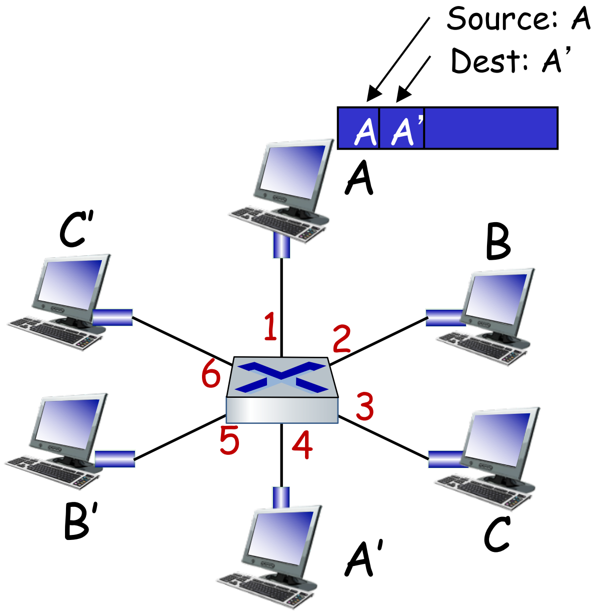
Q: how are entries created, maintained in switch table?

- 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)

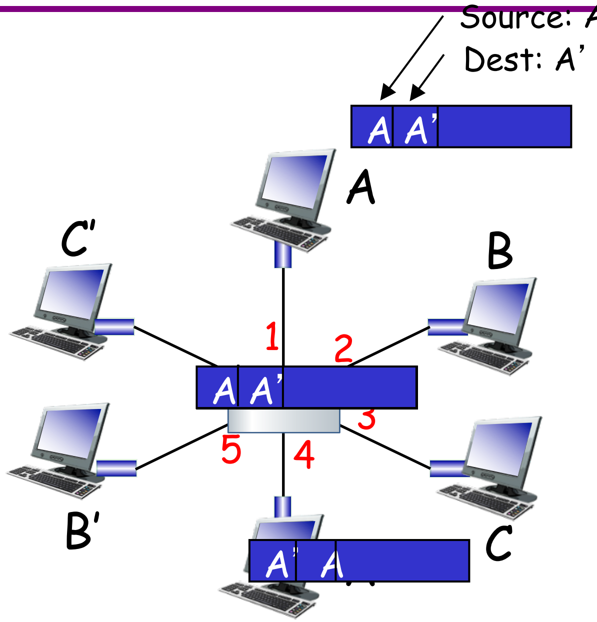
Switch: frame filtering/forwarding

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

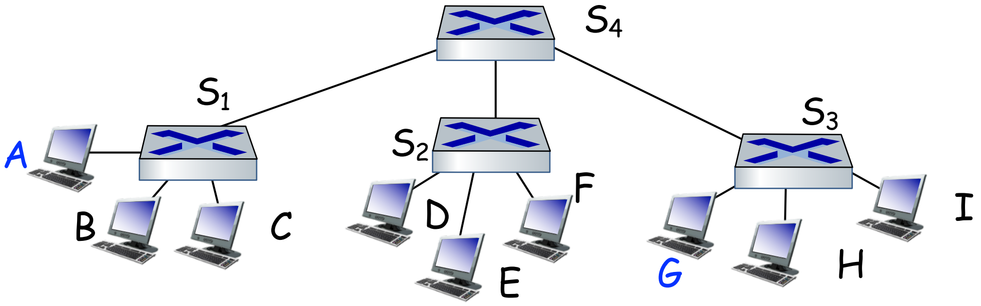


MAC addr	interface	TTL
A	1	60
A'	4	60

switch table (initially empty)

Interconnecting switches

self-learning switches can be connected together:

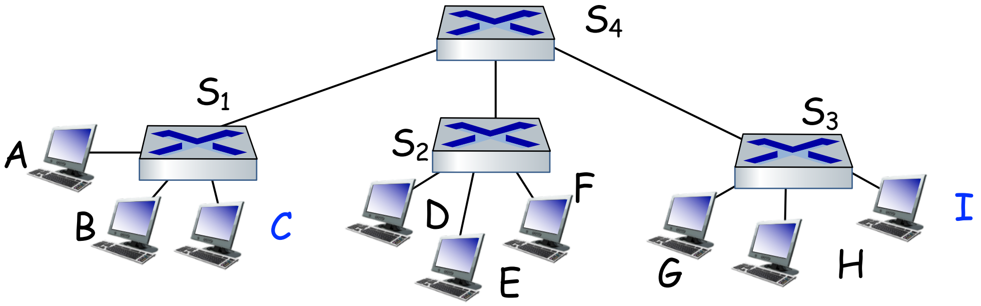


Q: sending from A to G - how does S_1 know to forward frame destined to G via S_4 and S_3?

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

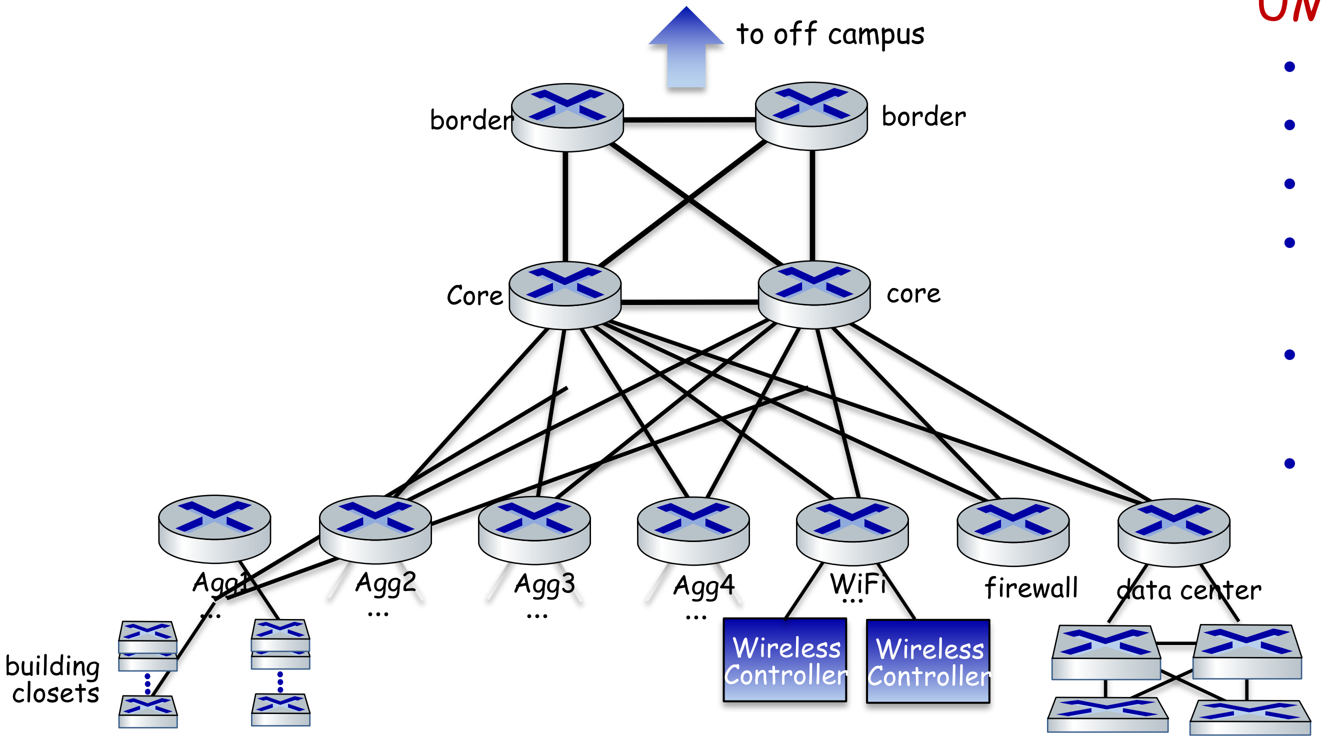
Self-learning multi-switch example

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

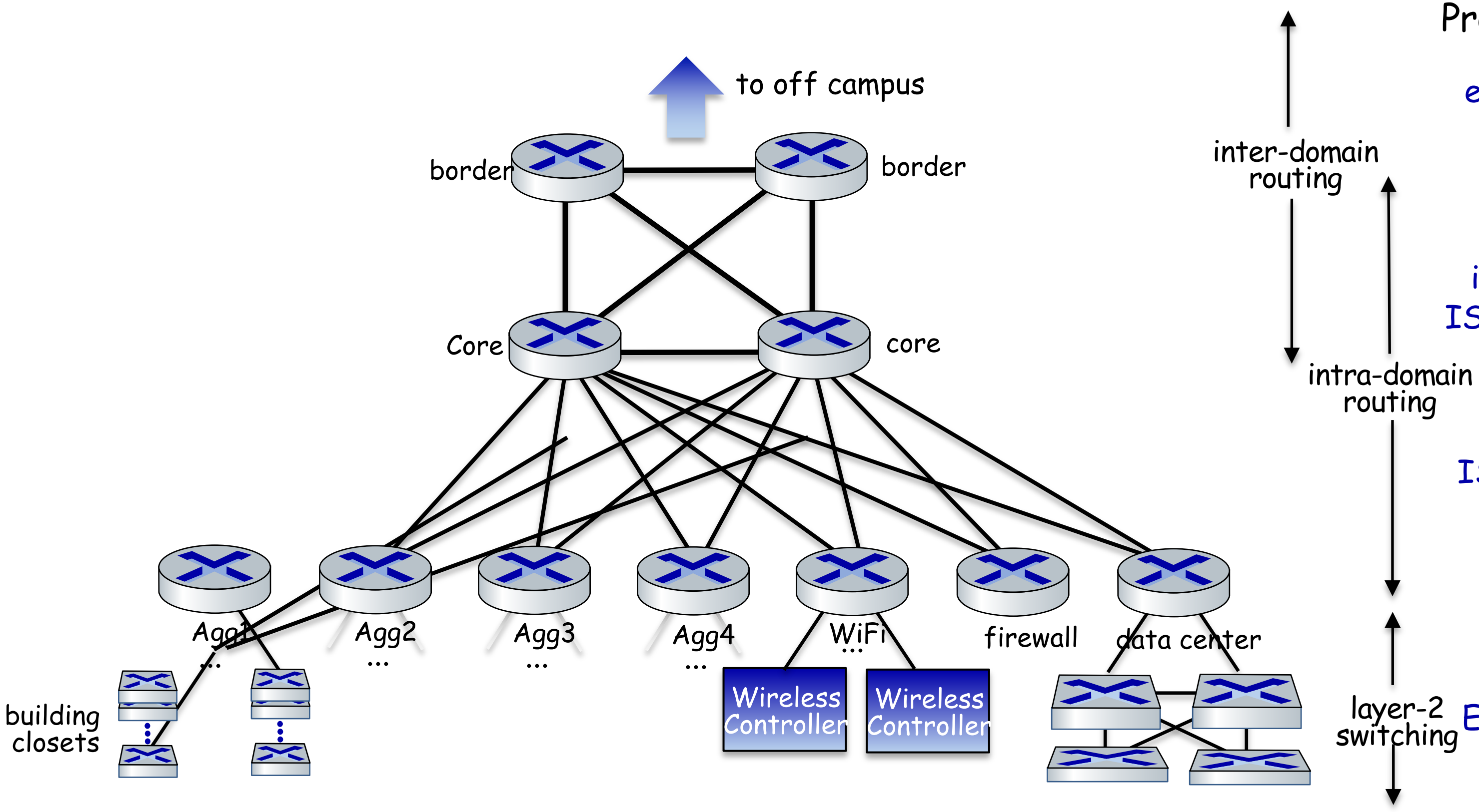


UMass network:

- 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



Protocols	Link Speeds
eBGP	10G; 100G pending
iBGP IS-IS	40G & 100G
IS-IS	40G
Ethernet	10G & 1G

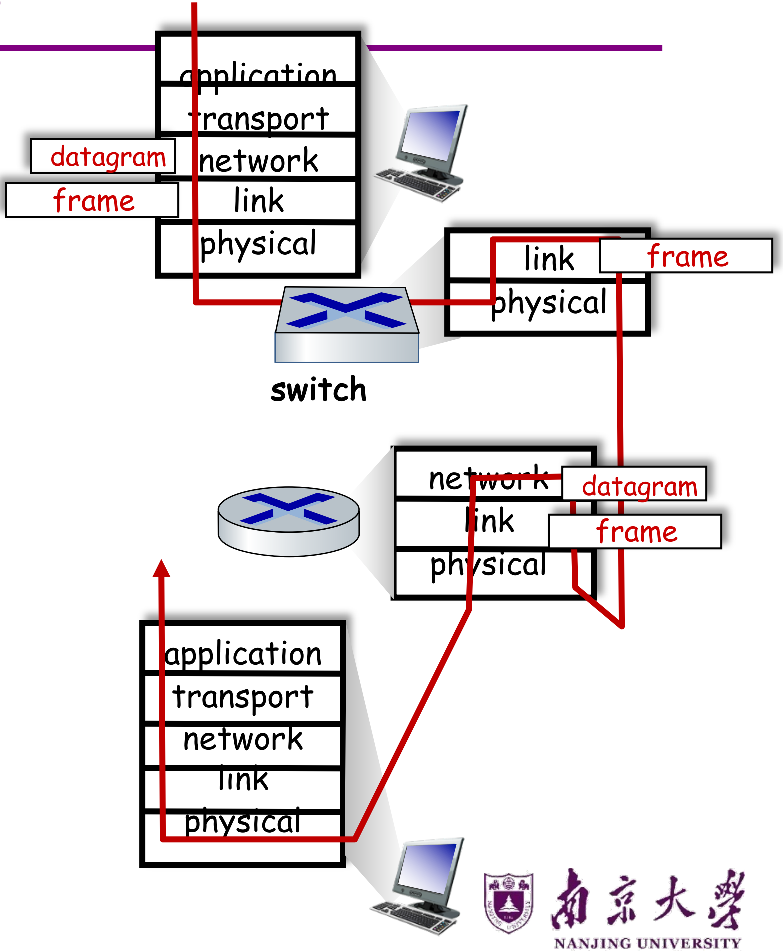
Switches vs. routers

both are store-and-forward:

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

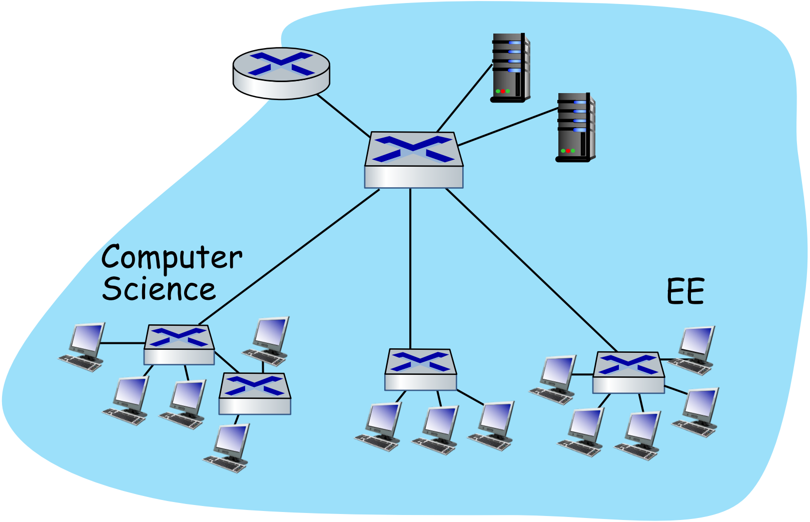
both have forwarding tables:

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



Virtual LANs (VLANs): motivation

Q: what happens as LAN sizes scale, users change point of attachment?

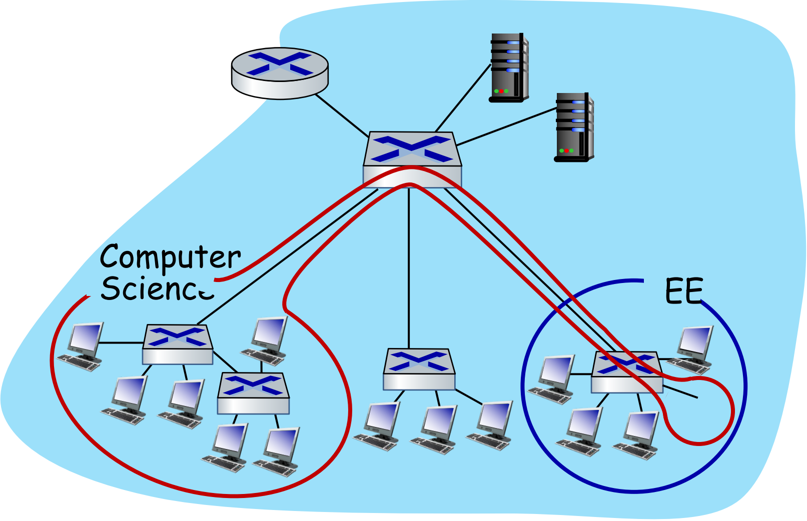


single broadcast domain:

- scaling: all layer-2 broadcast traffic (ARP, DHCP, unknown MAC) must cross entire LAN
- efficiency, security, privacy issues

Virtual LANs (VLANs): motivation

Q: what happens as LAN sizes scale, users change point of attachment?



single broadcast domain:

- scaling: all layer-2 broadcast traffic (ARP, DHCP, unknown MAC) must cross entire LAN
- efficiency, security, privacy issues

administrative issues:

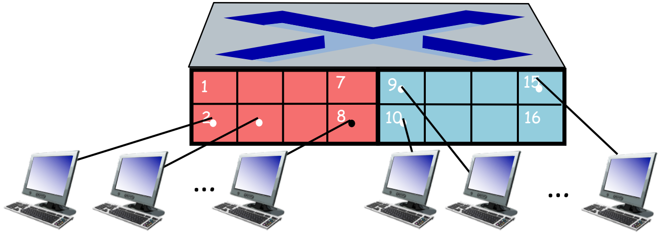
- CS user moves office to EE - physically attached to EE switch, but wants to remain logically attached to CS switch

Port-based VLANs

Virtual Local Area Network (VLAN)

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

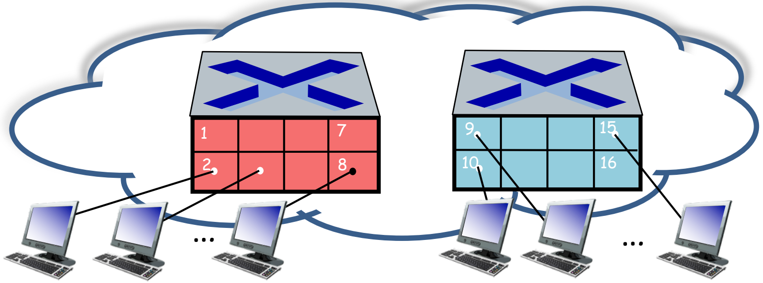
port-based VLAN: switch ports grouped (by switch management software) so that single physical switch ……



EE (VLAN ports 1-8)

CS (VLAN ports 9-15)

… operates as multiple virtual switches

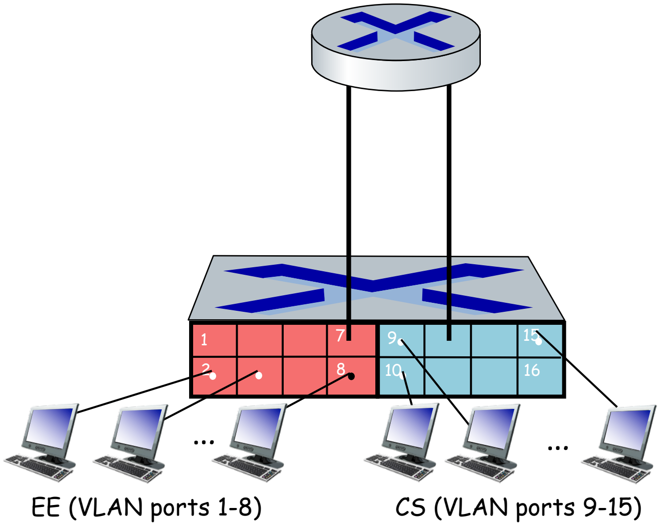


EE (VLAN ports 1-8)

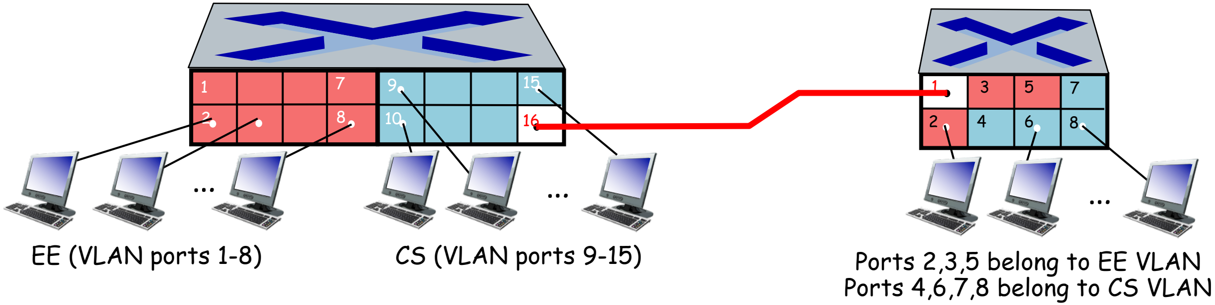
CS (VLAN ports 9-15)

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



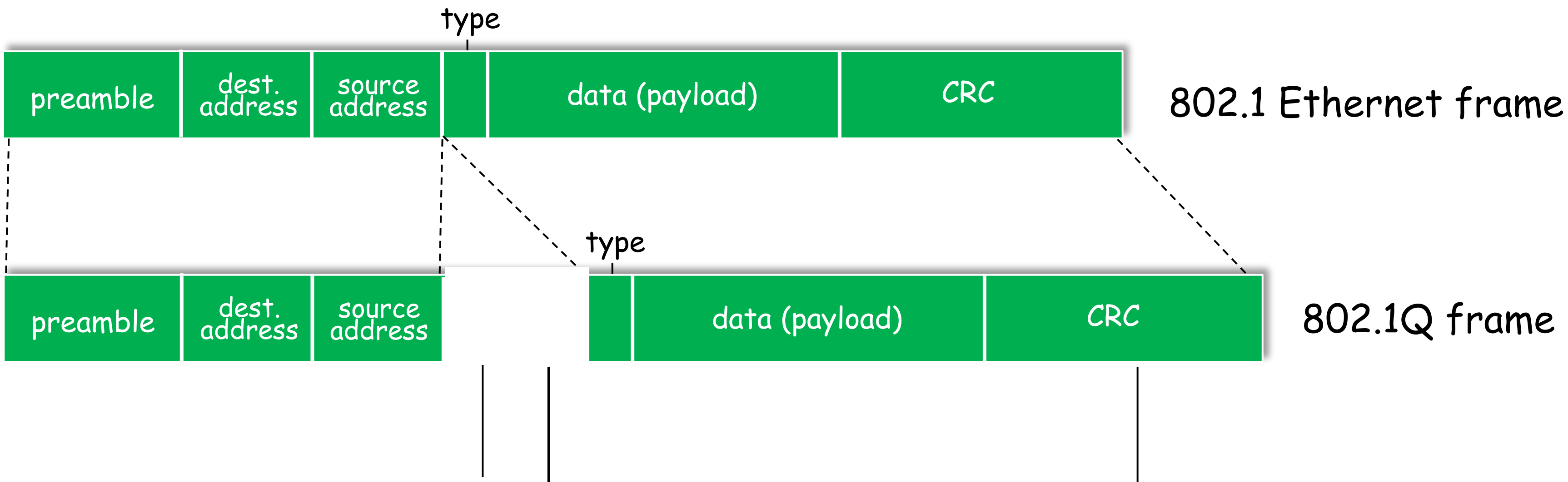
VLANS spanning multiple switches



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

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

802.1Q VLAN frame format

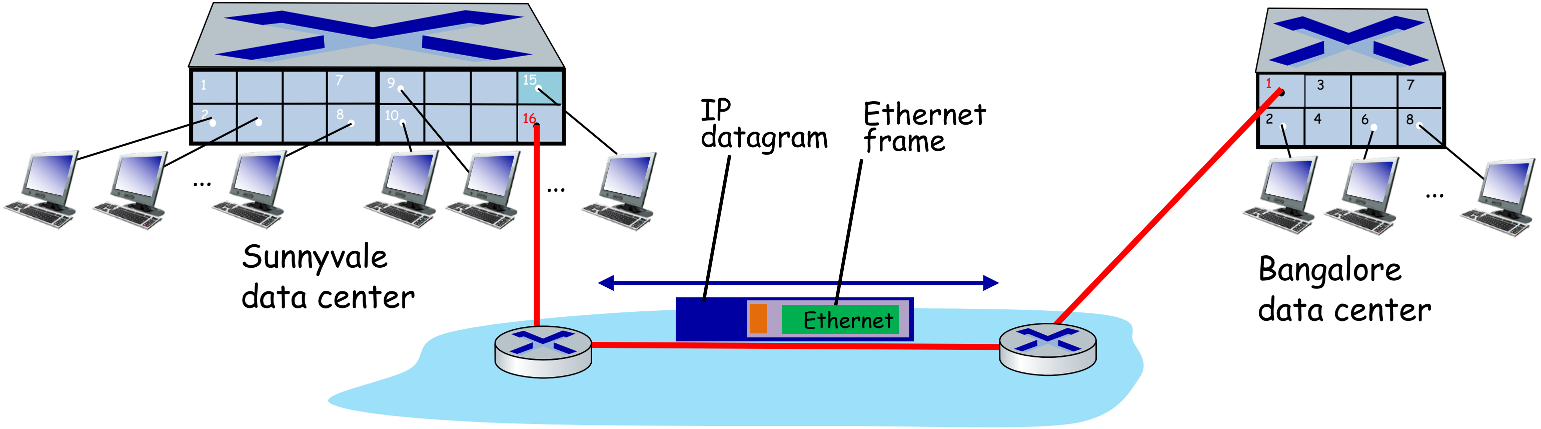


2-byte Tag Protocol Identifier
(value: 81-00)

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

Recomputed CRC

EVPN: Ethernet VPNs (aka VXLANs)



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

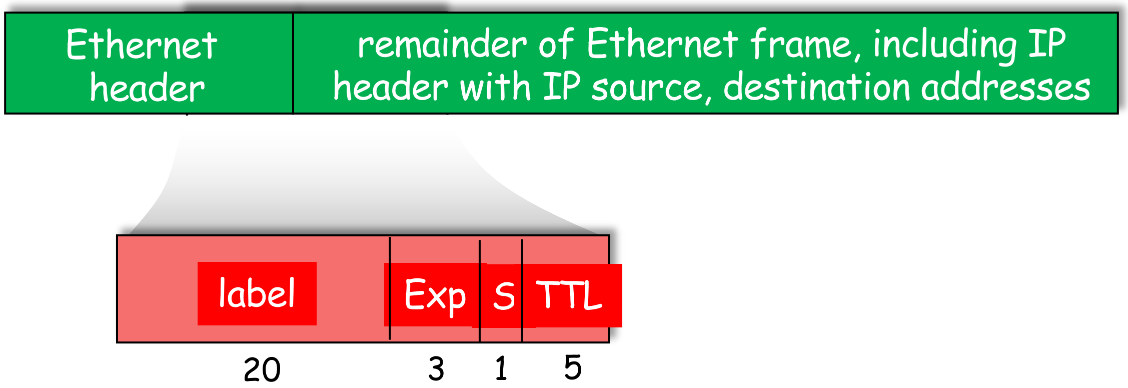
- Ethernet frames carried within IP datagrams between sites
- “tunneling scheme to overlay Layer 2 networks on top of Layer 3 networks ... runs over the existing networking infrastructure and provides a means to "stretch" a Layer 2 network.” [RFC 7348]

Outline

- Introduction
- Error detection, correction
- Multiple access protocols
- LANs
- Link virtualization: MPLS
- Data center networking
- A day in the life of a web request

Multiprotocol label switching (MPLS)

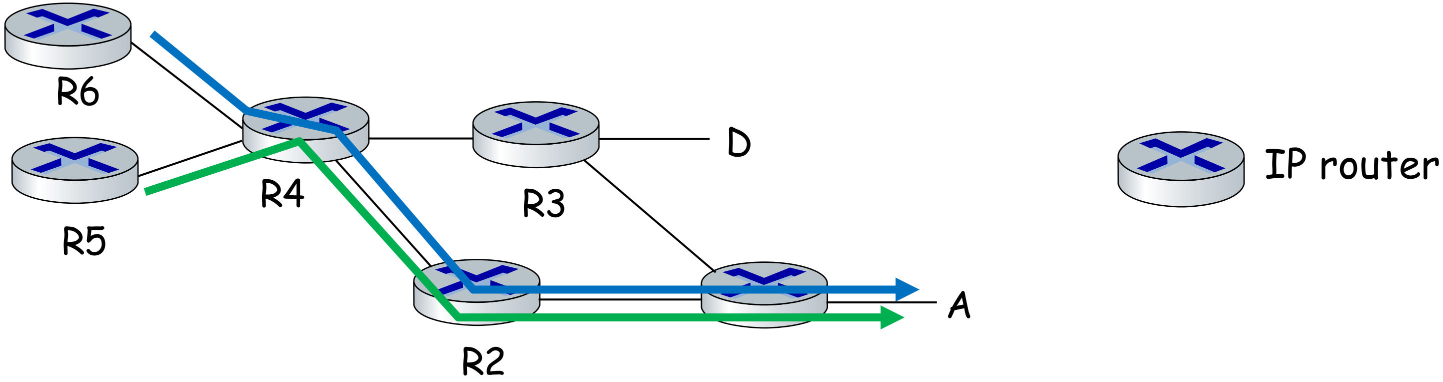
- goal: high-speed IP forwarding among network of MPLS-capable routers, using fixed length label (instead of shortest prefix matching)
  - faster lookup using fixed length identifier
  - borrowing ideas from Virtual Circuit (VC) approach
  - but IP datagram still keeps IP address!


Ethernet header
remainder of Ethernet frame, including IP header with IP source, destination addresses
label
Exp
S
TTL
20
3
1
5

MPLS capable routers

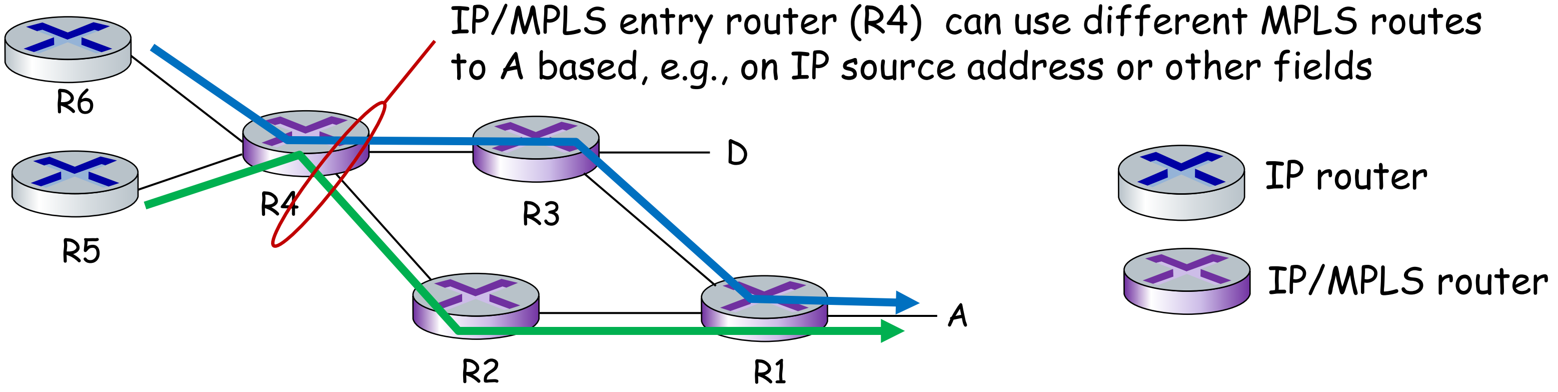
- a.k.a. label-switched router
- forward packets to outgoing interface based only on label value (don't inspect IP address)
  - MPLS forwarding table distinct from IP forwarding tables
- flexibility: MPLS forwarding decisions can differ from those of IP
  - use destination and source addresses to route 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

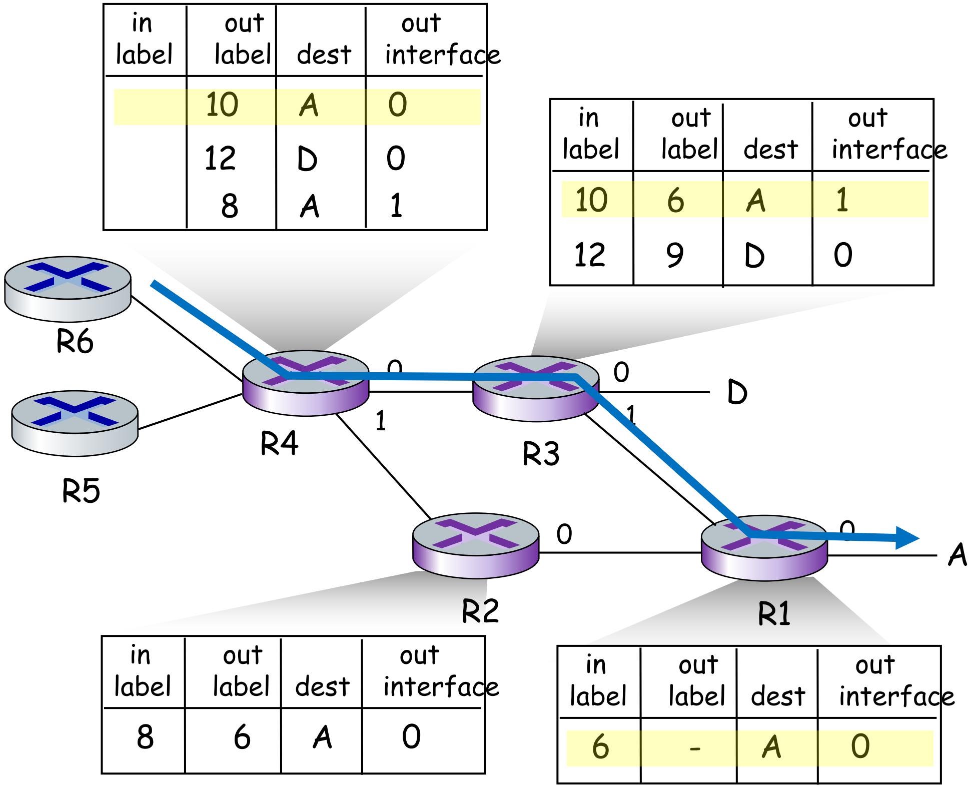
MPLS versus IP paths



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 forwarding tables



R4:

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

R3:

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

R2:

in label	out label	dest	out interface
8	6	A	0

R1:

in label	out label	dest	out interface
6	-	A	0

Outline

- Introduction
- Error detection, correction
- Multiple access protocols
- LANs
- 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:

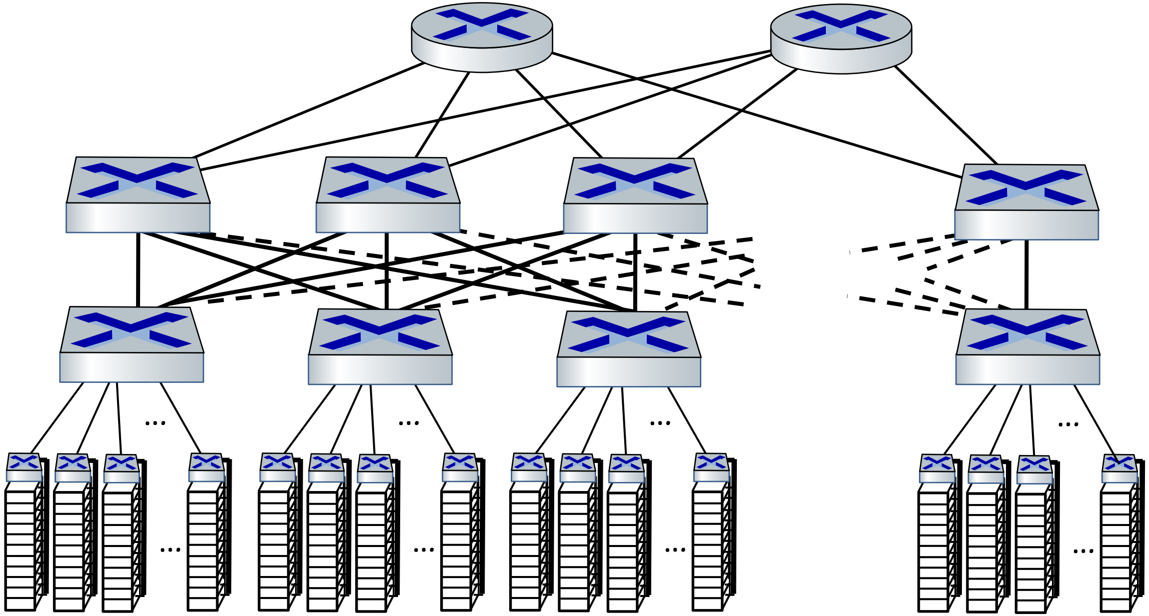
- e-business (e.g. Amazon)
- content-servers (e.g., YouTube, Akamai, Apple, Microsoft)
- search engines, data mining (e.g., Google)

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

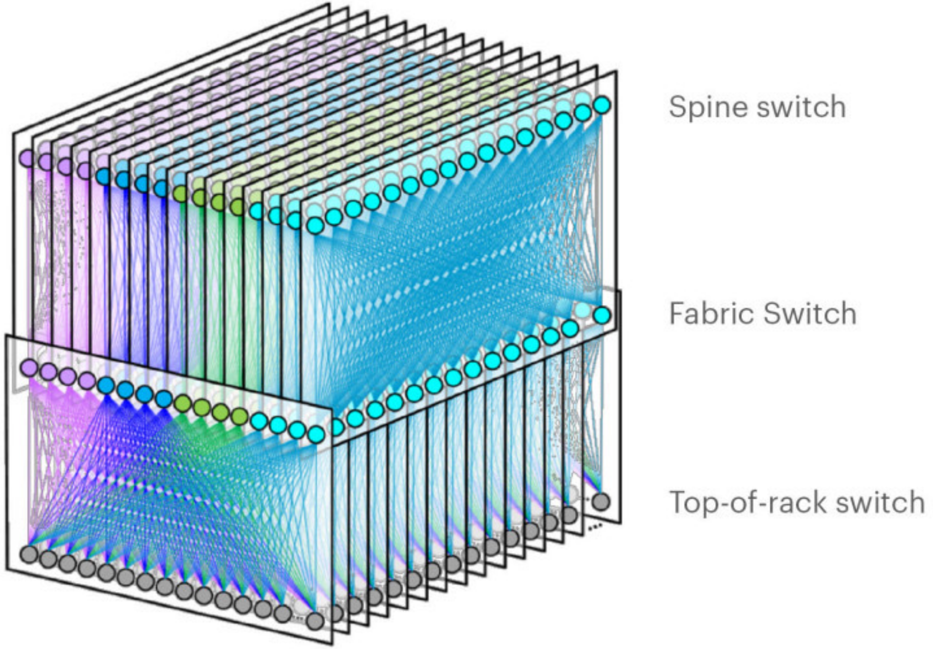
- one per rack
- 100G-400G Ethernet to blades

Server racks

- 20- 40 server blades: hosts

Datacenter networks: network elements

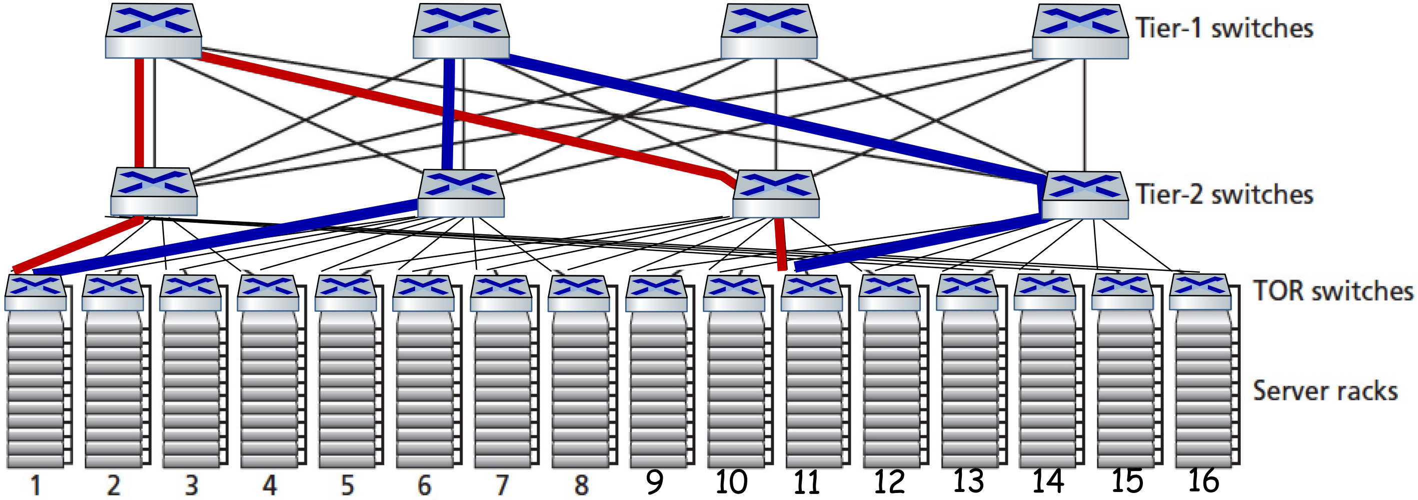
Facebook F16 data center network topology:



https://engineering.fb.com/data-center-engineering/f16-minipack/ (posted 3/2019)

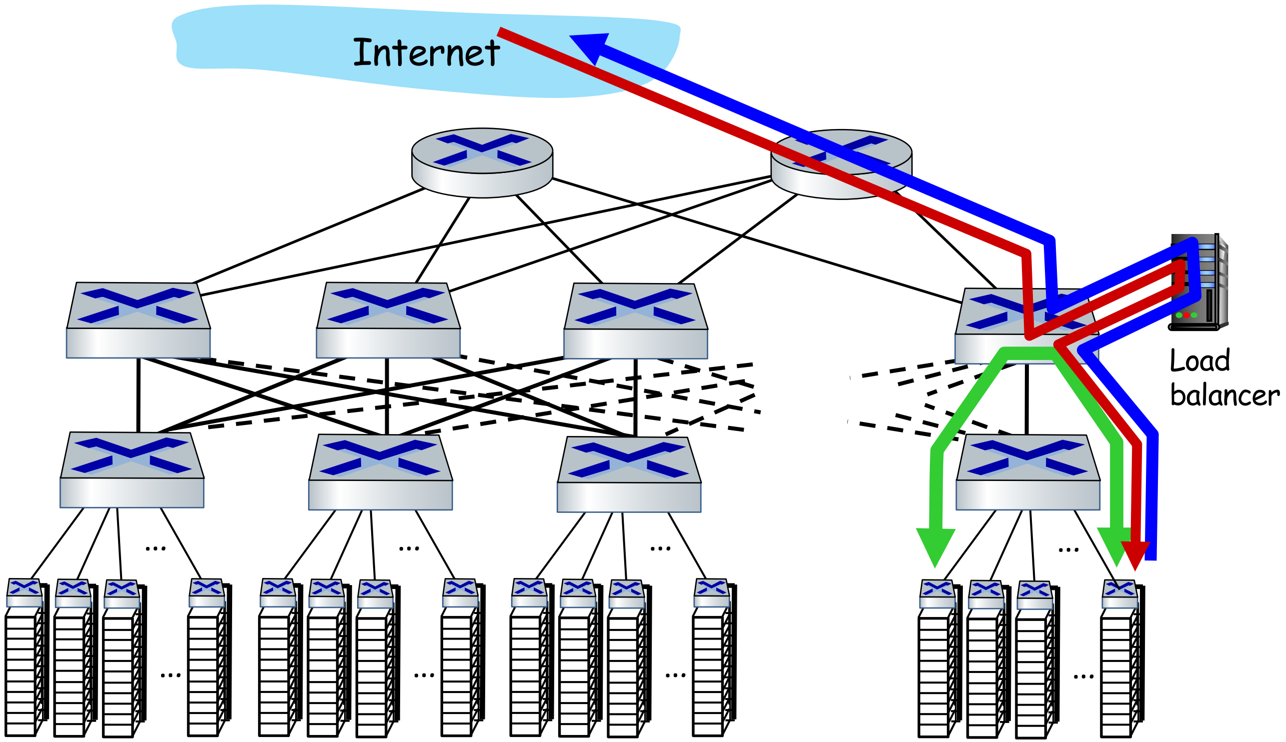
Datacenter networks: multipath

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



two disjoint paths highlighted between racks 1 and 11

Datacenter networks: application-layer routing



load balancer: application-layer routing

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

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/

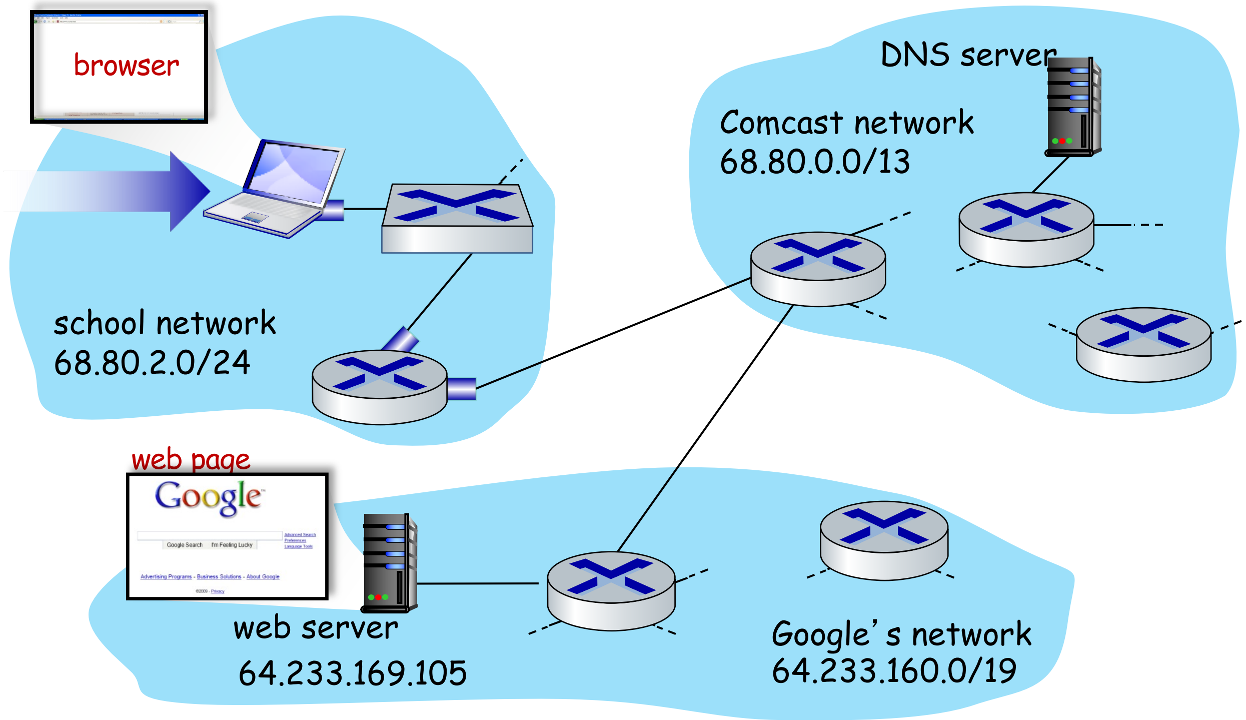
Outline

- Introduction
- Error detection, correction
- Multiple access protocols
- LANs
- 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

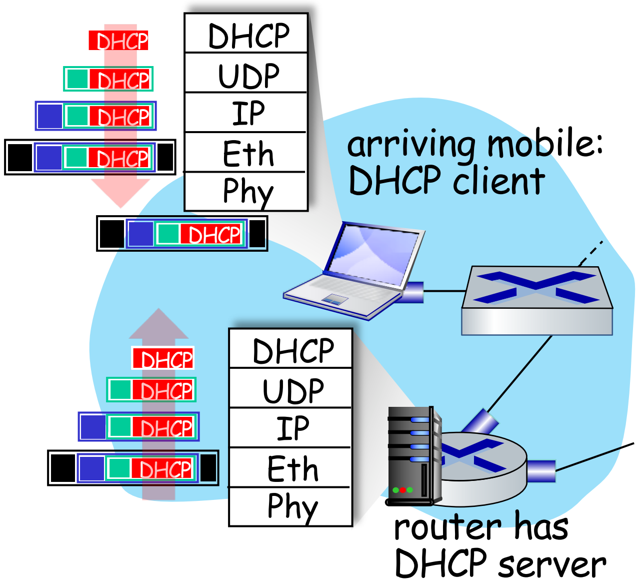


scenario:

- arriving mobile client attaches to network …
- requests web page: www.google.com

Sounds simple!

A day in the life: connecting to the Internet



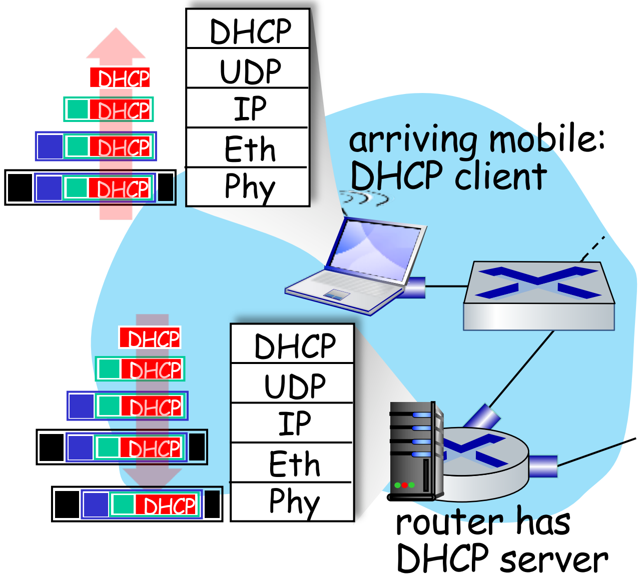
- connecting laptop needs to get its own IP address, addr of first-hop router, addr of DNS server: use DHCP

- 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

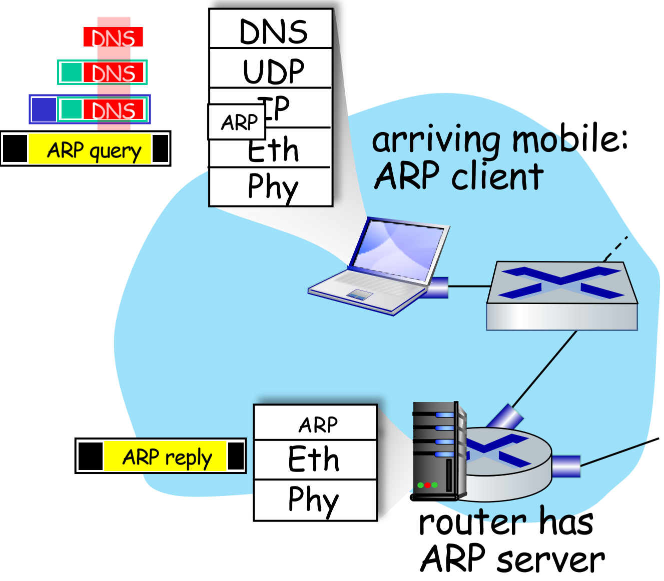
A day in the life: connecting to the Internet



- 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

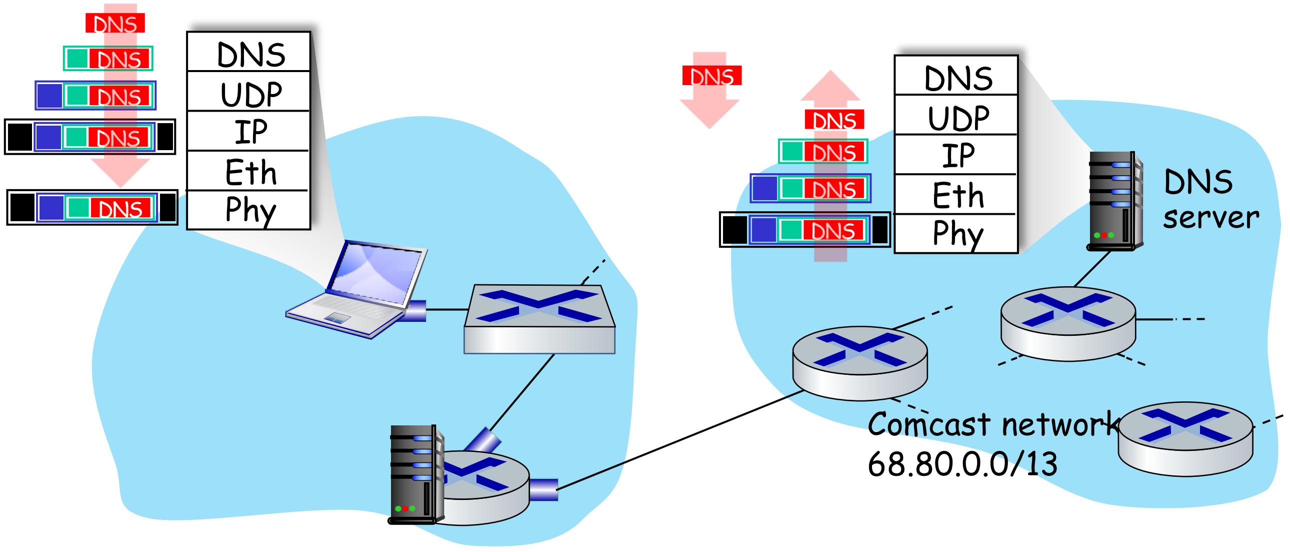
Client now has IP address, knows name & addr of DNS server, IP address of its first-hop router

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

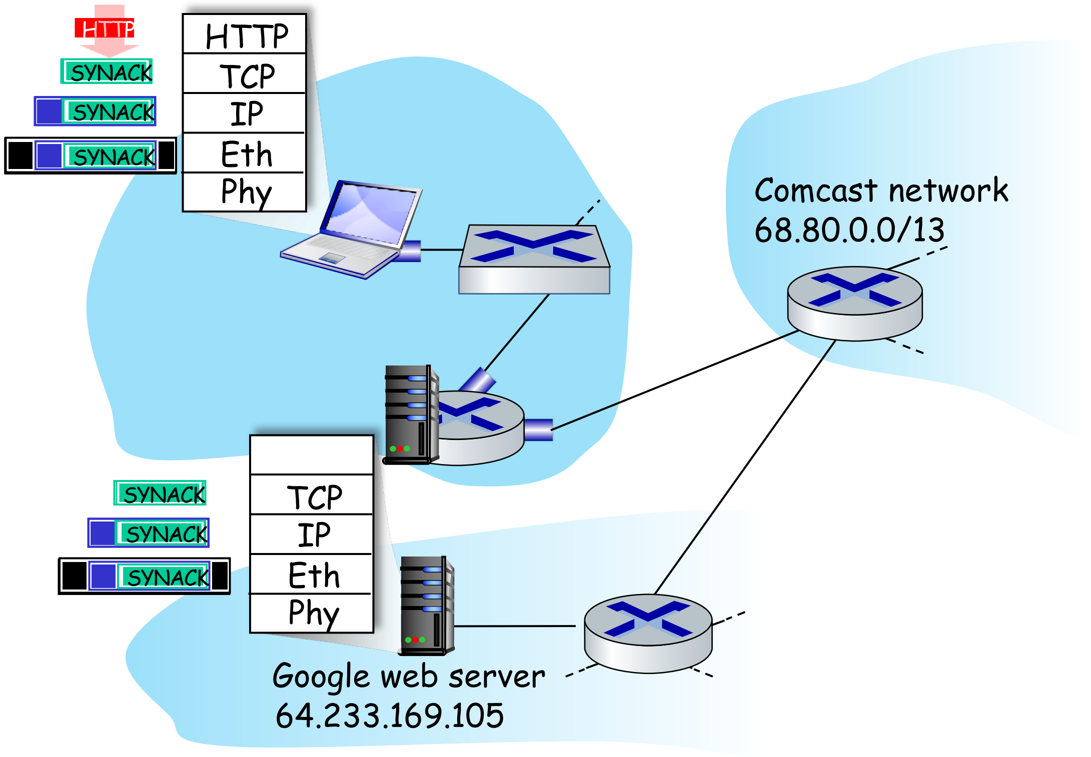


- de-muxed to DNS
- DNS replies to client with IP address of www.google.com

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

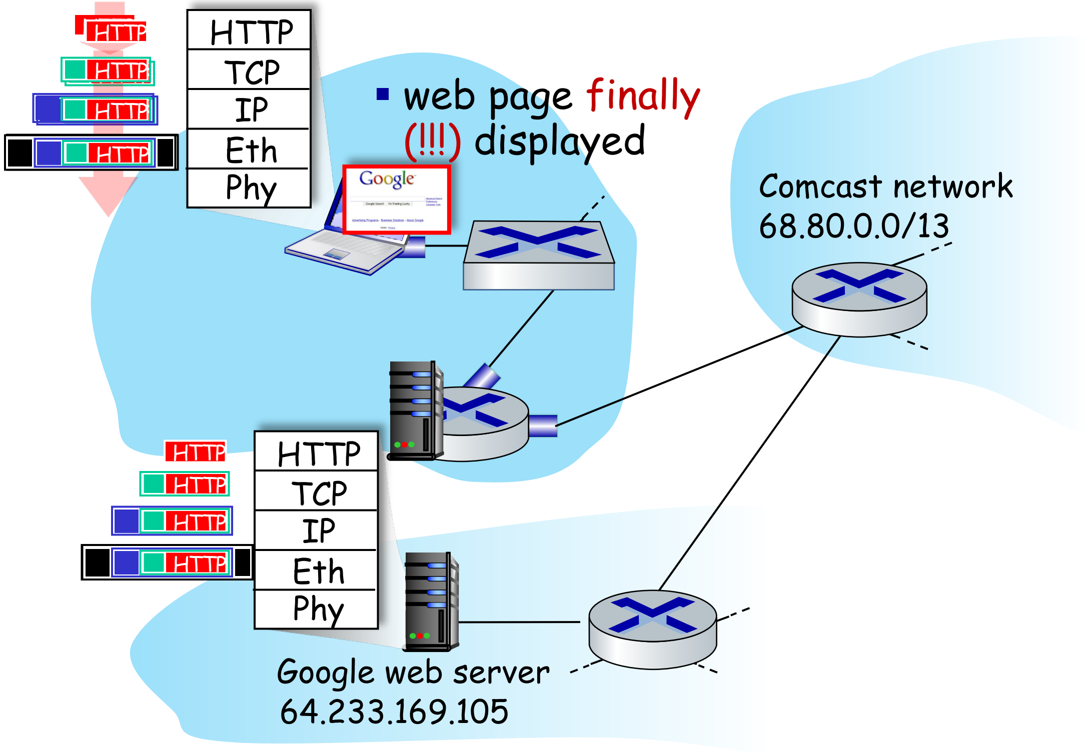
- IP datagram forwarded from campus network into Comcast network, routed (tables created by RIP, OSPF, IS-IS and/or BGP routing protocols) to DNS server

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

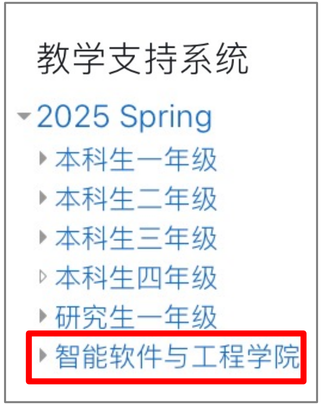


- web page finally (!!!) displayed

- 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

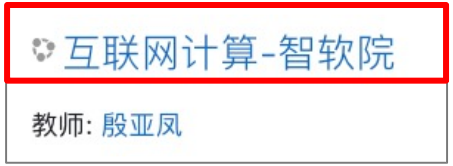
课程习题（作业）——截止日期：5月13日晚23:59

- **课本341-346页**：P18、P19、P23、P24、P25、P26题
- 提交方式：https://selearning.nju.edu.cn/（教学支持系统）



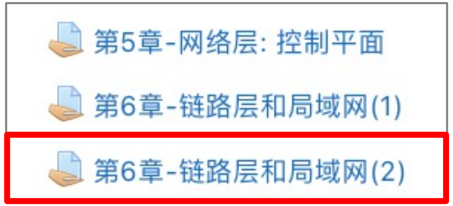
教学支持系统

- 2025 Spring
  - 本科生一年级
  - 本科生二年级
  - 本科生三年级
  - 本科生四年级
  - 研究生一年级
  - 智能软件与工程学院



互联网计算-智软院

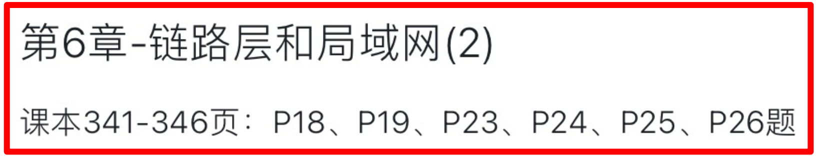
教师: 殷亚凤



第5章-网络层: 控制平面

第6章-链路层和局域网(1)

第6章-链路层和局域网(2)

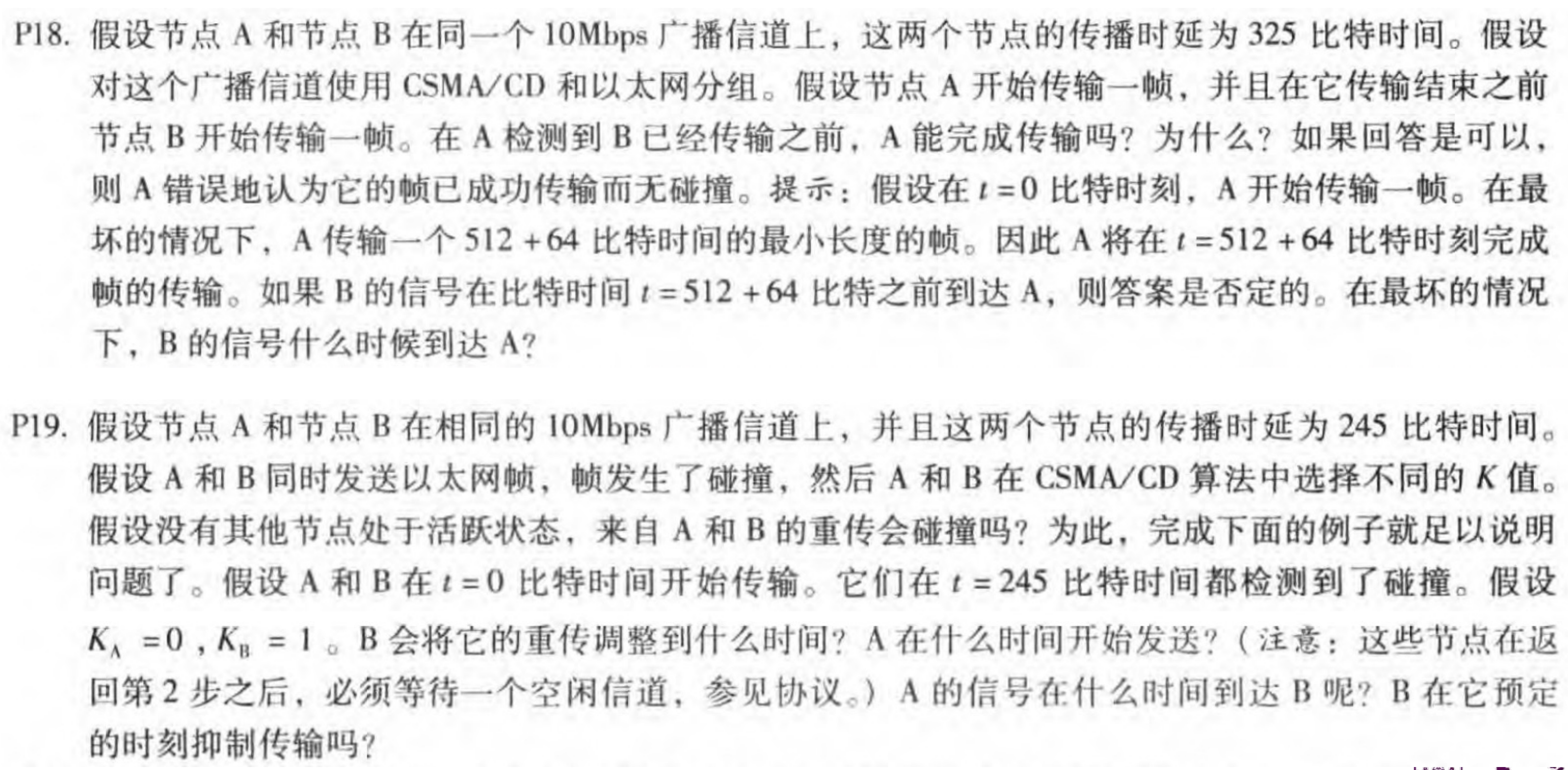


第6章-链路层和局域网(2)

课本341-346页：P18、P19、P23、P24、P25、P26题

- 命名：学号+姓名+第*章。
- 若提交遇到问题请及时发邮件或在下一次上课时反馈。

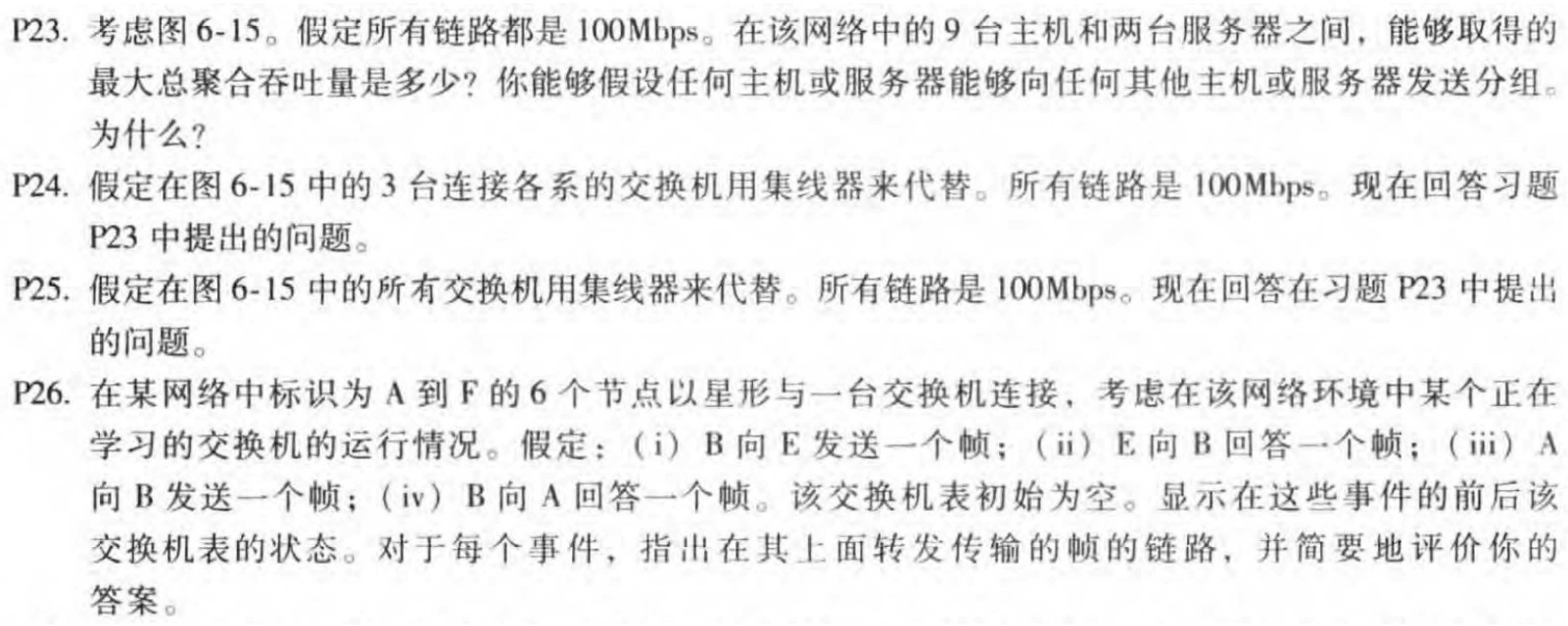
课程习题（作业）——截止日期：5月13日晚23:59



P18. 假设节点 A 和节点 B 在同一个 10Mbps 广播信道上，这两个节点的传播时延为 325 比特时间。假设对这个广播信道使用 CSMA/CD 和以太网分组。假设节点 A 开始传输一帧，并且在它传输结束之前节点 B 开始传输一帧。在 A 检测到 B 已经传输之前，A 能完成传输吗？为什么？如果回答是可以，则 A 错误地认为它的帧已成功传输而无碰撞。提示：假设在 $t=0$ 比特时刻，A 开始传输一帧。在最坏的情况下，A 传输一个 512 + 64 比特时间的最小长度的帧。因此 A 将在 $t=512+64$ 比特时刻完成帧的传输。如果 B 的信号在比特时间 $t=512+64$ 比特之前到达 A，则答案是否定的。在最坏的情况下，B 的信号什么时候到达 A？

P19. 假设节点 A 和节点 B 在相同的 10Mbps 广播信道上，并且这两个节点的传播时延为 245 比特时间。假设 A 和 B 同时发送以太网帧，帧发生了碰撞，然后 A 和 B 在 CSMA/CD 算法中选择不同的 K 值。假设没有其他节点处于活跃状态，来自 A 和 B 的重传会碰撞吗？为此，完成下面的例子就足以说明问题了。假设 A 和 B 在 $t=0$ 比特时间开始传输。它们在 $t=245$ 比特时间都检测到了碰撞。假设 $K_A=0$，$K_B=1$。B 会将它的重传调整到什么时间？A 在什么时间开始发送？（注意：这些节点在返回第 2 步之后，必须等待一个空闲信道，参见协议。）A 的信号在什么时间到达 B 呢？B 在它预定的时刻抑制传输吗？

课程习题（作业）——截止日期：5月13日晚23:59



P23. 考虑图 6-15。假定所有链路都是 100Mbps。在该网络中的 9 台主机和两台服务器之间，能够取得的最大总聚合吞吐量是多少？你能够假设任何主机或服务器能够向任何其他主机或服务器发送分组。为什么？

P24. 假定在图 6-15 中的 3 台连接各系的交换机用集线器来代替。所有链路是 100Mbps。现在回答习题 P23 中提出的问题。

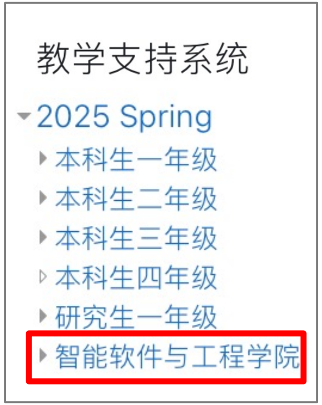
P25. 假定在图 6-15 中的所有交换机用集线器来代替。所有链路是 100Mbps。现在回答在习题 P23 中提出的问题。

P26. 在某网络中标识为 A 到 F 的 6 个节点以星形与一台交换机连接，考虑在该网络环境中某个正在学习的交换机的运行情况。假定：（i） B 向 E 发送一个帧；（ii） E 向 B 回答一个帧；（iii） A 向 B 发送一个帧；（iv） B 向 A 回答一个帧。该交换机表初始为空。显示在这些事件的前后该交换机表的状态。对于每个事件，指出在其上面转发传输的帧的链路，并简要地评价你的答案。

实验3——截止日期：5月20日晚23:59

- 实验3：响应ARP

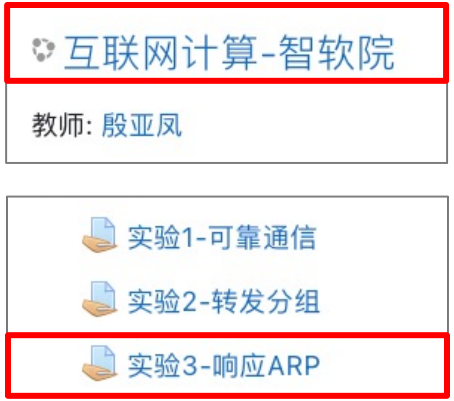
- 提交方式：https://selearning.nju.edu.cn/（教学支持系统）



教学支持系统

2025 Spring

- 本科生一年级
- 本科生二年级
- 本科生三年级
- 本科生四年级
- 研究生一年级
- 智能软件与工程学院



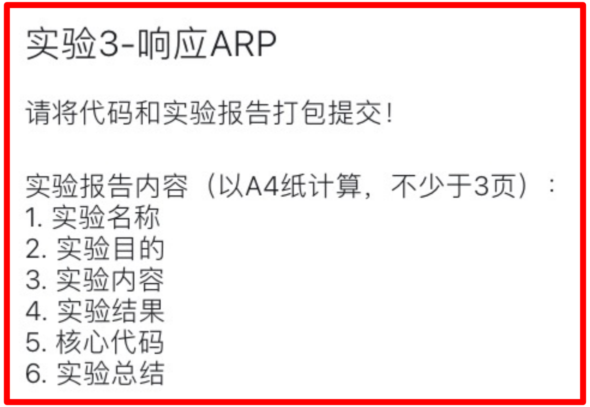
互联网计算-智软院

教师: 殷亚凤

实验1-可靠通信

实验2-转发分组

实验3-响应ARP



实验3-响应ARP

请将代码和实验报告打包提交!

实验报告内容（以A4纸计算，不少于3页）：
1. 实验名称
2. 实验目的
3. 实验内容
4. 实验结果
5. 核心代码
6. 实验总结

- 命名：学号+姓名+实验*。

- 若提交遇到问题请及时发邮件或在下一次上课时反馈。

实验3——截止日期：5月20日晚23:59

Lab 3: Respond to ARP

Overview

This is the first in a series of exercises that have the ultimate goal of creating an IPv4 router. The basic functions of an Internet router are to:

1. Respond to ARP (Address Resolution Protocol) requests for addresses that are assigned to interfaces on the router.
2. Make ARP requests for IP addresses that have no known Ethernet MAC address. A router will often have to send packets to other hosts, and needs Ethernet MAC addresses to do so.
3. Receive and forward packets that arrive on links and are destined to other hosts. Part of the forwarding process is to perform address lookups ("longest prefix match" lookups) in the forwarding information base. You will eventually just use "static" routing in your router, rather than implement a dynamic routing protocol like RIP or OSPF.
4. Respond to Internet Control Message Protocol (ICMP) messages like echo requests ("pings").
5. Generate ICMP error messages when necessary, such as when an IP packet's TTL (time to live) value has been decremented to zero.

The goal of this first stage of building the router is to accomplish item **#1** above: respond to ARP requests.

实验3——截止日期：5月20日晚23:59

Lab 3: Respond to ARP

Task 1: Preparation
Initiate your project with our template.
Start the task here

Task 2: Handle ARP Requests
Ready to make ARP work.
Start the task here

Task 3: Cached ARP Table
Maintain a correlation between each MAC address and its corresponding IP address.
Start the task here

提问

Q & A

殷亚凤
智能软件与工程学院
苏州校区南雍楼东区225
yafeng@nju.edu.cn，https://yafengnju.github.io/