Computer Networks

Interworking

Jianping Pan
Fall 2020

Software engineering or Computer Science?

Graduating soon – or just graduated- this session is for you!

CO-OP + CAREER CONVERSATION:
PREPARING TO LAUNCH YOUR CAREER
Targeted to Computer Science & Software Engineering students, but all students welcome!

Tuesday, December 1 | 12pm Pacific Time | Zoom

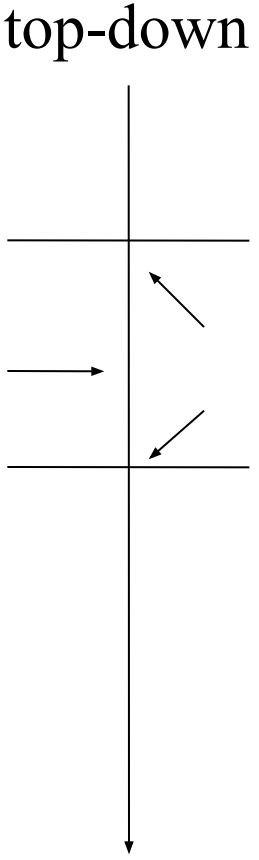
Are you nearing the end of your degree, or have you recently graduated? Starting to think about where your computer science or software engineering degree might take you career-wise in a post-graduation world?

Join **John Fagan** (UVic Career Educator) and **Ian Frazer** (President of Canadian Information Processing Society - CIPS) to learn more about preparing to graduate, planning your career, and the industry available for new computer science and software engineering grads. Bring questions and get answers

Register on LIM to receive a link - https://learninginmotion.uvic.ca/events.htm?evtId=4344

Review: protocols

- Application layer
  - HTTP, DNS
- Transport layer
  - TCP, UDP
- Network layer
  - IP/ICMP; RIP, OSPF, BGP
- Link layer
  - HDLC, IEEE 802.3, IEEE 802.11



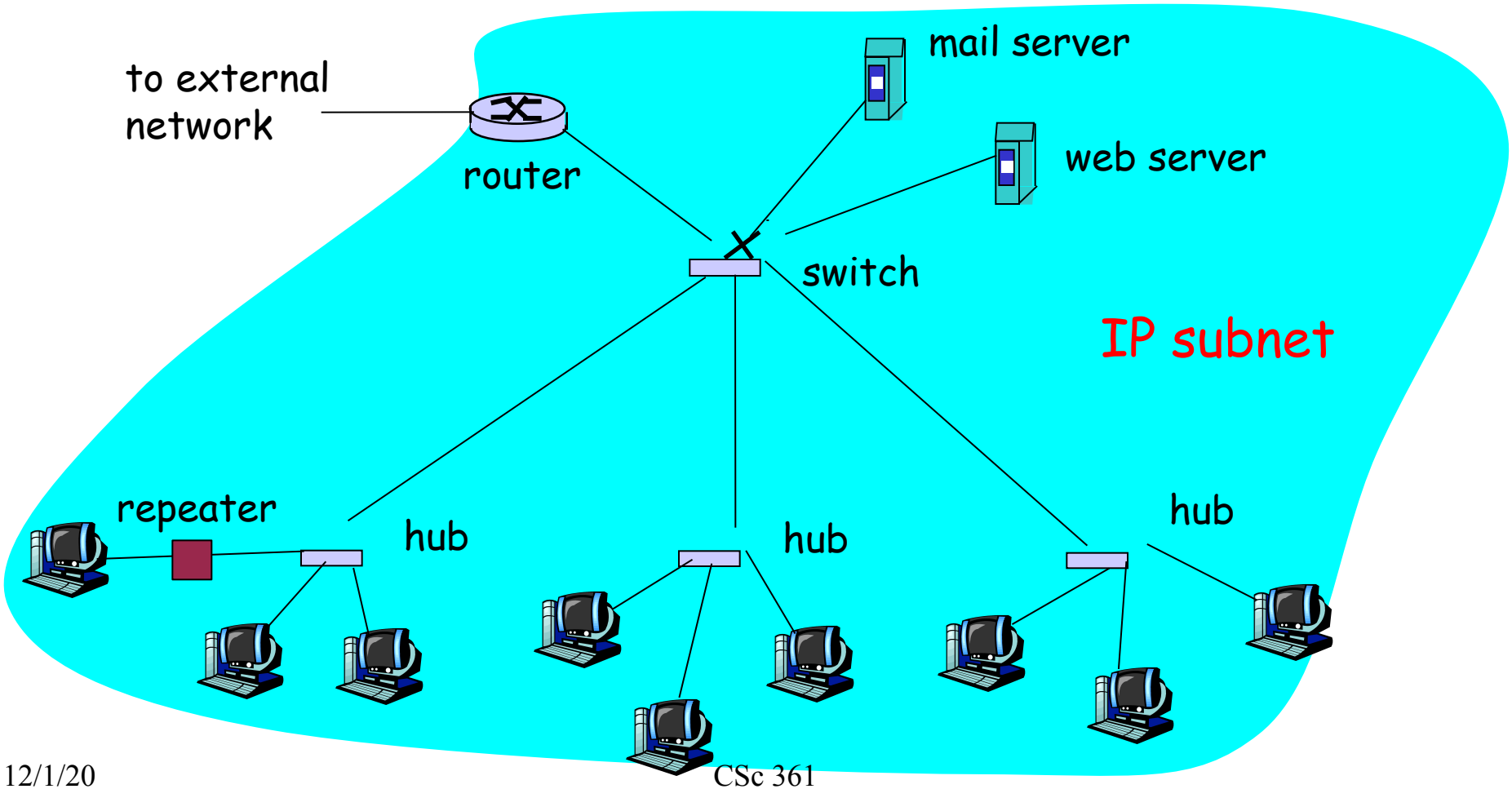
* messages, segments, packets, frames, etc?

Today's topics

- Interworking
  - now a “bottom-up” approach
  - devices
    - repeater, hub, switch, router, gateway
  - protocols
    - STP: spanning tree protocol
    - ARP: address resolution protocol

* recall DNS?

Example network



* WiFi range extender? DSL/cable home gateway?

Interworking devices

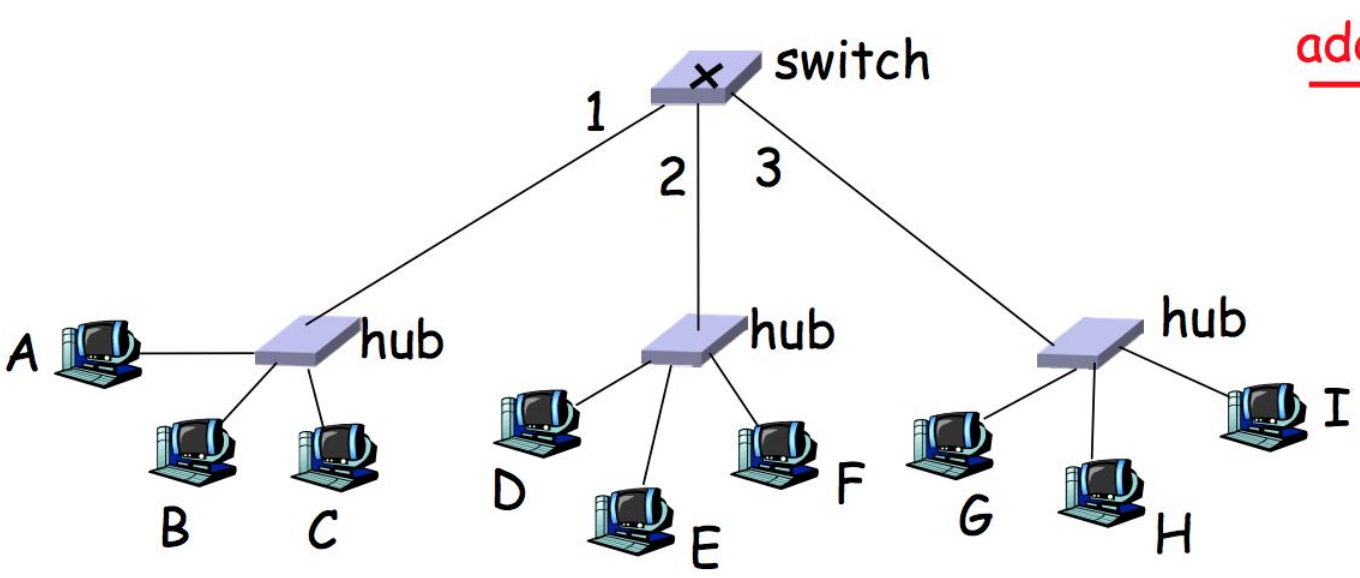
- Repeater: signal amplifier/regenerators
- Hub (e.g., thick vs thin-cable Ethernet)
  - Layer 1 device: forward from one link to all others
  - larger collision domain
- Switch
  - Layer 2 device, selective forward, self-learning
  - transparent to end hosts
- Router: Layer 3 device, routing involved
- Gateway: higher-layer protocol specific

Self-learning switch

- A switch has a switch table, which is built automatically, dynamically, and autonomously—without any intervention from a network admin
- entry in switch table:
  - (MAC Address, Interface, Time Stamp)--Interface leads to the MAC Address
  - stale entries in table dropped (TTL can be 60 min)
- switch *learns* which hosts can be reached through which interfaces
  - when a frame is received, switch "learns" location of sender/source: incoming LAN segment
  - records sender/location pair in switch table

Self-learning switch example

Suppose C sends a frame to D

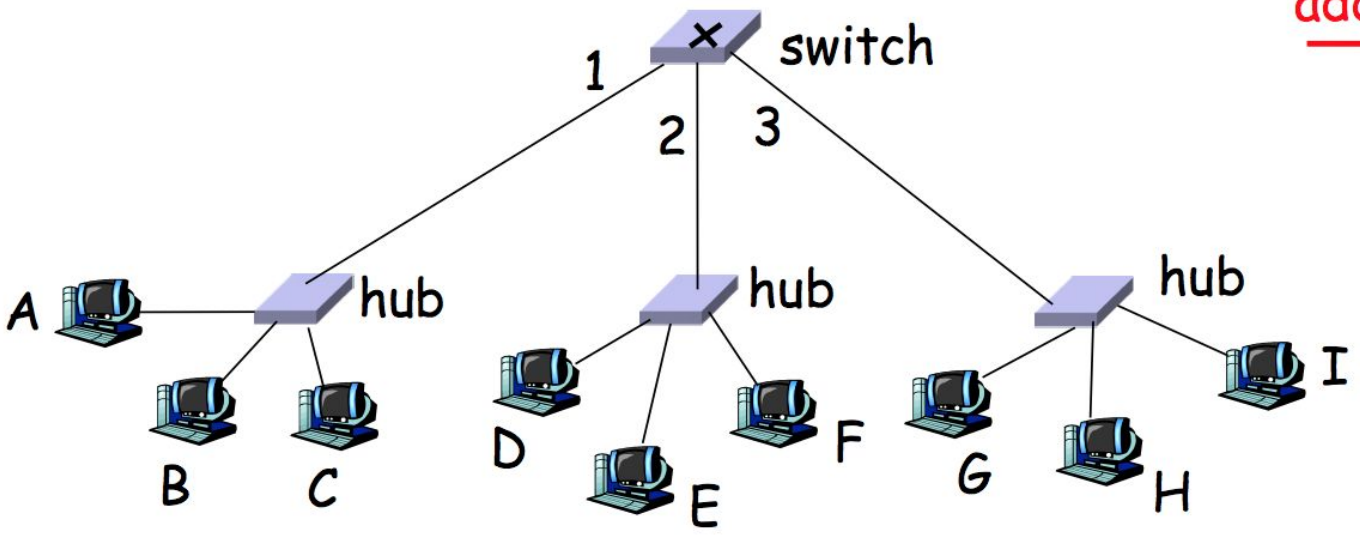


address	interface
A	1
B	1
E	2
G	3

- Switch receives frame from C
  - notes in switch table that C is on interface 1
  - because D is not in table, switch forwards frame into interfaces 2 and 3--flooding

Example continued

Suppose D replies back with frame to C.

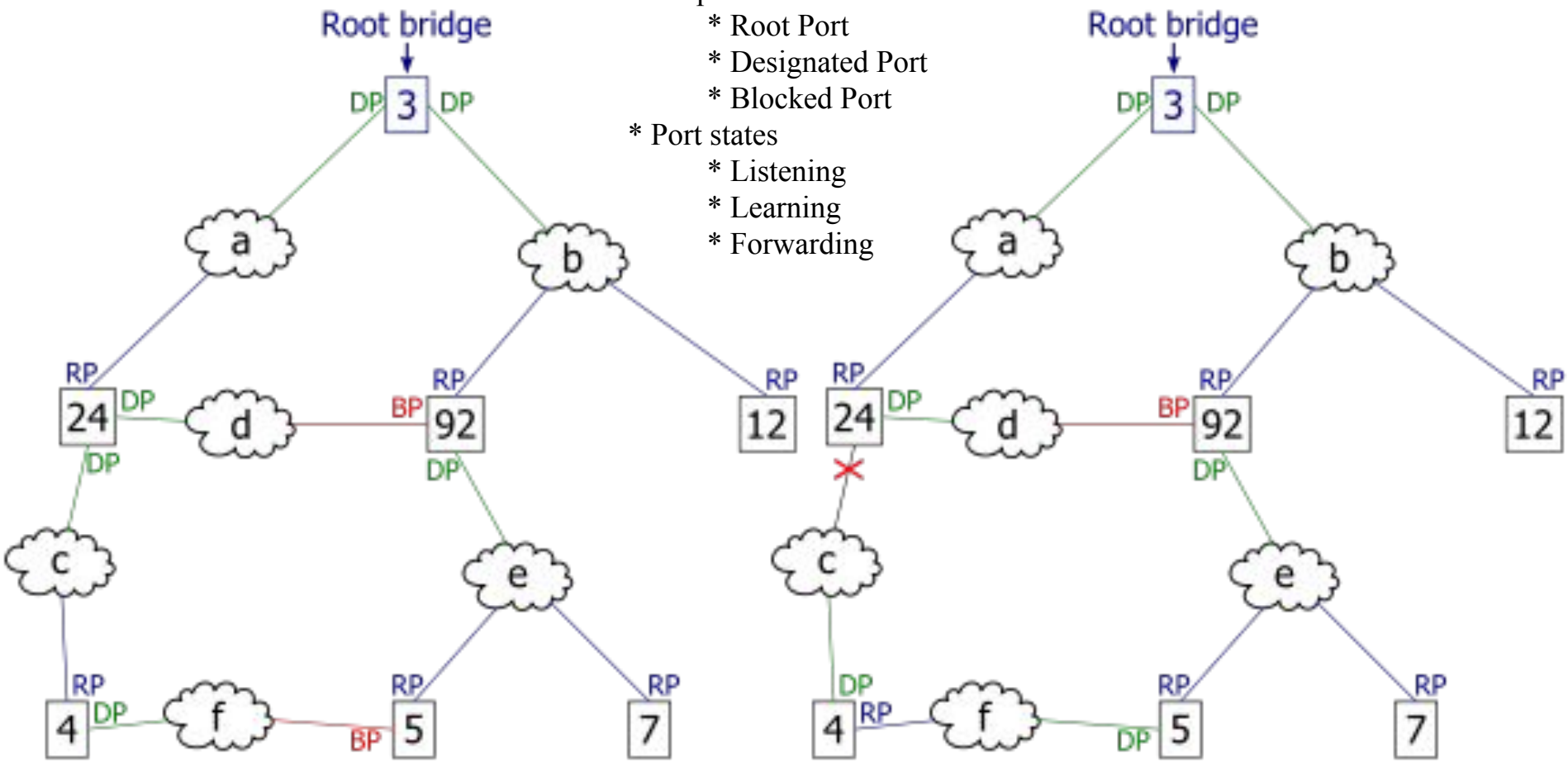


address	interface
A	1
B	1
E	2
G	3
C	1

- Switch receives frame from D
  - notes in switch table that D is on interface 2
  - because C is in table, switch forwards frame only to interface 1

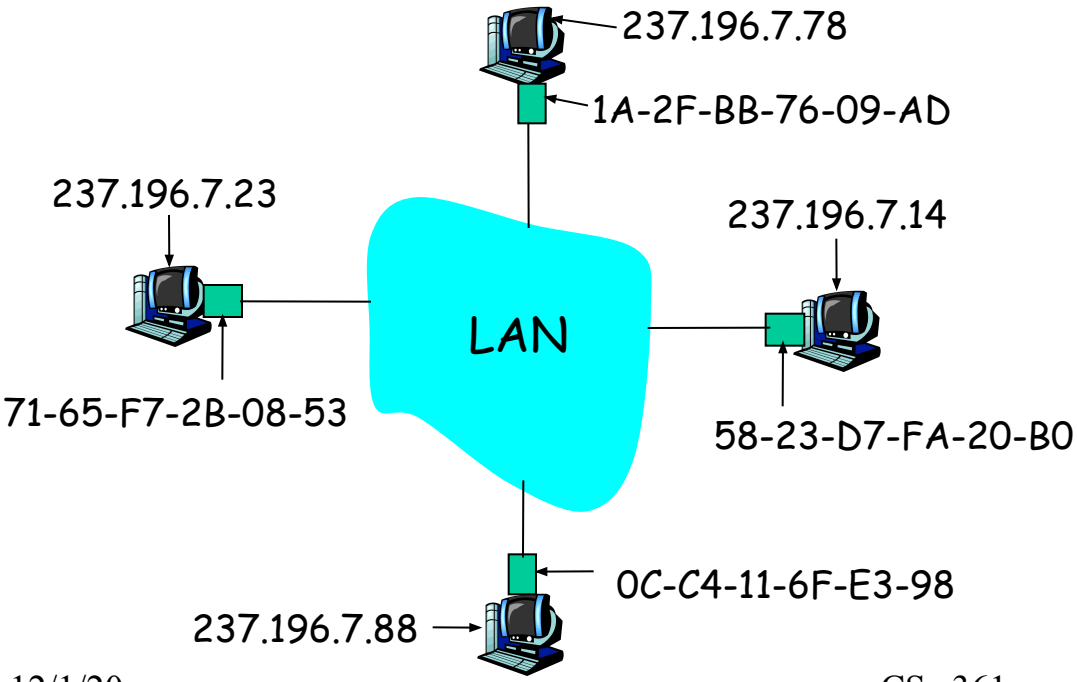
Spanning tree protocol

* Graph -> Tree
    * Root Port
    * Designated Port
    * Blocked Port
* Port states
    * Listening
    * Learning
    * Forwarding



ARP: Address Resolution Protocol

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



- Each IP node (Host, Router) on LAN has ARP table
- ARP 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)

$ arp

* how to know B's IP address?

* subnet test?

ARP protocol: Same LAN (network)

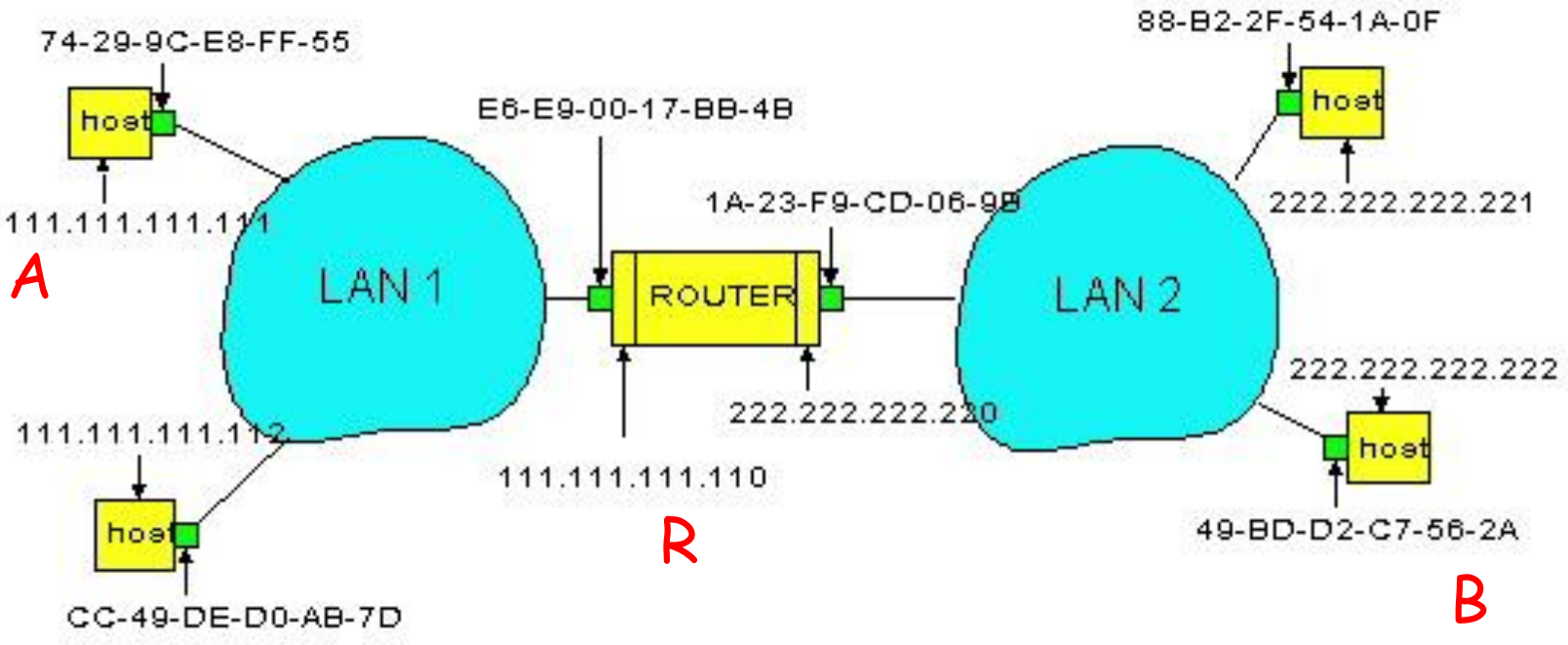
- **A** wants to send datagram to **B**, and **B**'s MAC address not in A's ARP table.
- **A** broadcasts ARP query packet, containing **B**'s IP address
  - Dest MAC address = FF-FF-FF-FF-FF-FF
  - all machines on LAN receive ARP query
- **B** receives ARP packet, replies to **A** with its (**B**'s)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

- **A** caches (saves) IP-to-MAC address pair in its ARP table until information becomes old (times out)
  - soft state: information that times out (goes away) unless refreshed
- ARP is "plug-and-play":
  - nodes create their ARP tables without intervention from net administrator

Routing to another LAN

walkthrough: send datagram from A to B via R

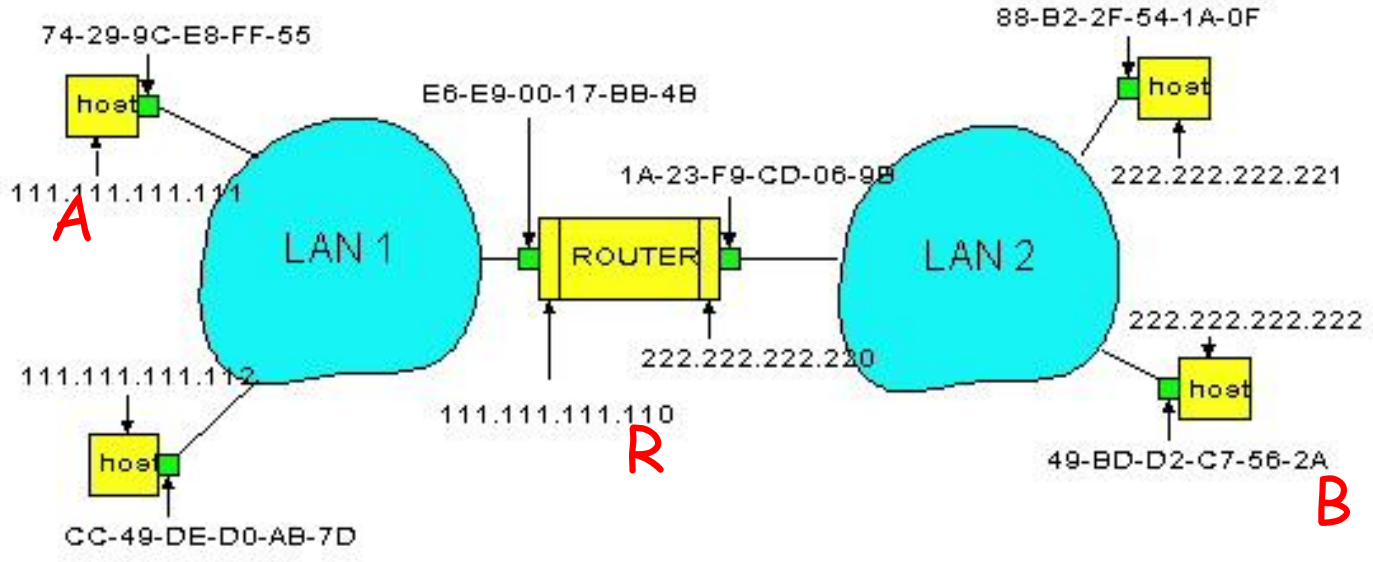
assume **A** knows **B** IP address



- Two ARP tables in router **R**, one for each IP network (LAN)

* how to know "another" network?

- A creates datagram with source A, destination B
- A uses ARP to get R's MAC address for 111.111.111.110
- A creates link-layer frame with R's MAC address as dest, frame contains A-to-B IP datagram
- A's adapter sends frame
- R's adapter receives frame
- R removes IP datagram from Ethernet frame, sees its destined to B, and then forward the datagram from 111.111.111.110 to 222.222.222.220 (forwarding table).
- R uses ARP to get B's MAC address
- R creates frame containing A-to-B IP datagram sends to B



We will see ARP in today and tomorrow's lab

- What we have played in http://picohub.csc.uvic.ca
  - Lab 1: **PicoNet** with *ping* and *tcpdump*
  - Lab 2: HTTP (persistent vs non-persistent): *nc* with *nginx*
  - Lab 3: P1 (SWS)
  - Lab 4: DNS (recursive vs iterative): *nslookup* with *deadwood*
  - Lab 5: TCP Connection Management and Flow Control: *iperf* with *tc* netem delay
  - Lab 6: TCP Error control: *iperf* with *tc* netem delay and loss
    - P2 (RDP)
  - Lab 7: TCP Congestion Control: *iperf* with *tc* netem delay and loss, and congestion modules
  - Lab 8: IP Addressing: *ip* a and *iptables* for NAT
  - Lab 9: IP Routing: *ip* r with *iptables* and *traceroute*
    - P3 (SoR)
  - Lab A: **Interworking (all things together): *arp* and forwarding and routing**
- You have used a lot of network diagnosis tools
  - *ping*, *tcpdump*, *traceroute*: network connectivity and protocol interaction
  - *ip* address, link, neighbor, route: host configuration (and more---how PicoNet was built)
    - *iptables*: NAT, firewall and more; *tc*: traffic control, network emulation and more
  - *nc*: arbitrary TCP and UDP endpoints
    - *iperf*: performance metrics

* also well-known client/server apps: *wget*/*nginx*, *nslookup*/*deadwood*, etc

Midterm 3 (M3) preparation

- Friday, December 4, 2020, 9:30am Victoria Time (GMT-**8**)
- Coverage
  - All lectures, tutorials and labs, and their corresponding text materials, slides, assignments, etc
  - Right after M2 to Wednesday, December 2, 2020, all inclusive
- Type of questions
  - Concepts questions: describe and compare related concepts
  - Role-playing questions: If I am the computer (network), what shall I do?
  - Play-a-role questions: If I was given the (real) problem to solve, what shall I do?
- Format of the exam
  - Part 1 (7%): Friday 9:30am to 10:20am, Victoria time, through connex -> Tests & Quizzes
    - If you have CAL accommodation, the time duration (50 minutes) will be prorated
    - Very important: only one device and one browser window or tab, to access the test page
    - Save your work often, and when timeout, your work will be saved by the system too
  - Part 2 (8%): Friday 3pm to Saturday 3pm, Victoria time, through connex -> Tests & Quizzes
    - Same guideline as above but this time, you only have at most 4 hours to finish it
- All exams are open book/Internet, but have to be done by yourself alone