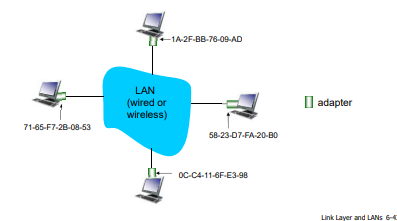
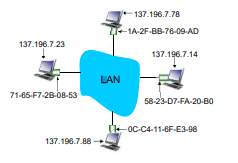
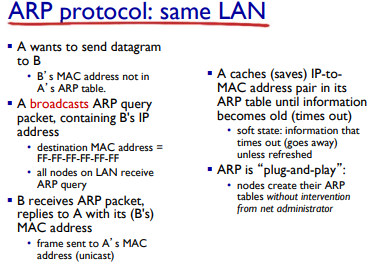
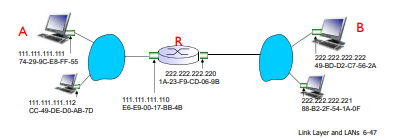
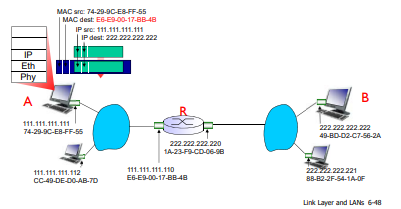
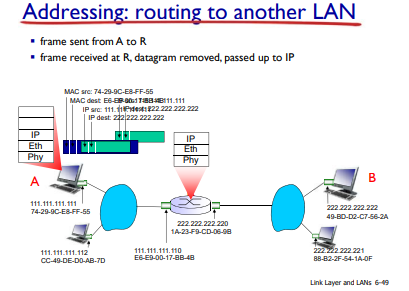
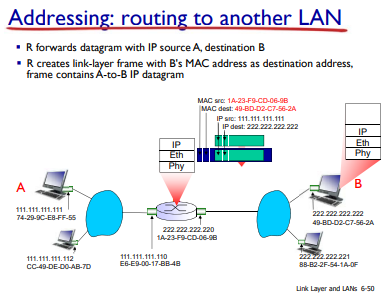
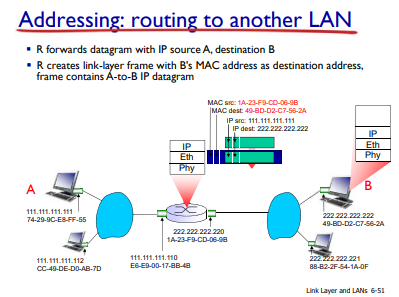
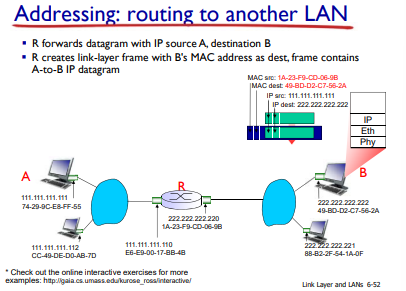
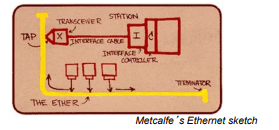
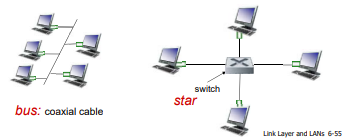
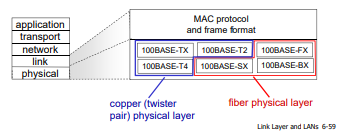
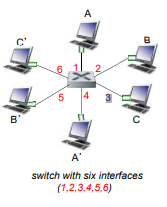
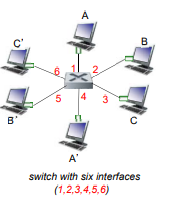
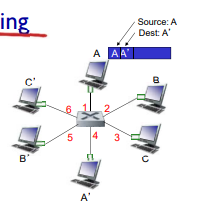
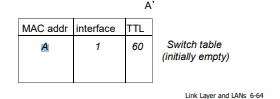
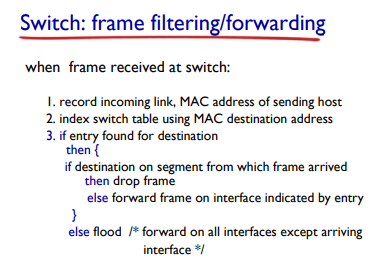
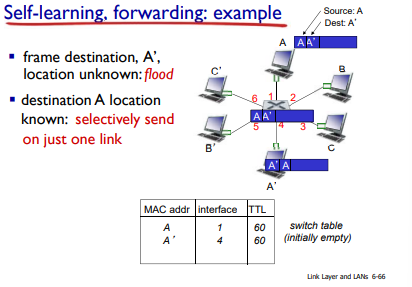
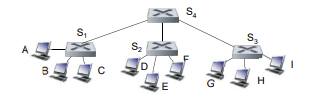
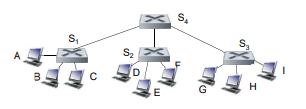
* MAC addresses and ARP
  + 32-bit IP address:
    - • network-layer address for interface
    - • used for layer 3 (network layer) forwarding
  + MAC (or LAN or physical or Ethernet) address:
    - • function: used ‘locally” to get frame from one interface to another physically-connected interface (same network, in IPaddressing sense)
    - • 48 bit MAC address (for most LANs) burned in NIC ROM, also sometimes software settable
    - • e.g.: 1A-2F-BB-76-09-AD
    - 
* LAN Addresses and ARP
  + Each adapter on LAn has unique LAN address
    - 
  + MAC address allocation administered by IEEE
  + manufacturer buys portion of MAC address space (to assure uniqueness)
  + analogy:
    - • MAC address: like Social Security Number
    - • IP address: like postal address
  + MAC flat address ➜ portability
    - • can move LAN card from one LAN to another
  + IP hierarchical address not portable
    - • address depends on IP subnet to which node is attached
* ARP: Address Resolution Protocol
  + ARP table: each IP node (host, router) on LAN has table
  + • IP/MAC address mappings for some LAN nodes:
  + < IP address; MAC address; TTL>
  + • TTL (Time To Live): time after which address mapping will be forgotten (typically 20 min)
  + 
* ARP Protocol: Same LAN
  + 
* Addressing: Routing to another LAN
  + walkthrough: send datagram from A to B via R
  + focus on addressing – at IP (datagram) and MAC layer (frame)
  + assume A knows B’s IP address
  + assume A knows IP address of first hop router, R (how?)
  + assume A knows R’s MAC address (how?)
  + 
* Addressing; Routing to another LAN
  + A creates IP datagram with IP source A, destination B
  + A creates link-layer frame with R's MAC address as destination address, frame contains A-to-B IP datagram
  + 
* 
* 
* 
* 
* Ethernet
  + “dominant” wired LAN technology:
  + single chip, multiple speeds (e.g., Broadcom BCM5761)
  + first widely used LAN technology
  + simpler, cheap
  + kept up with speed race: 10 Mbps – 10 Gbps
  + 
* Ethernet Physical Topology
  + bus: popular through mid 90s
    - • all nodes in same collision domain (can collide with each other)
  + star: prevails today
    - • active switch in center
    - • each “spoke” runs a (separate) Ethernet protocol (nodes do not collide with each other)
  + 
* Ethernet Frame Structure
  + sending adapter encapsulates IP datagram (or other network layer protocol packet) in Ethernet frame
  + 
  + preamble:
  + 7 bytes with pattern 10101010 followed by one byte with pattern 10101011
  + used to synchronize receiver, sender clock rates
* Ethernet: Unreliable, connectionless
  + connectionless: no handshaking between sending and receiving NICs
  + unreliable: receiving NIC doesn't send acks or nacks to sending NIC
  + • data in dropped frames recovered only if initial sender uses higher layer rdt (e.g., TCP), otherwise dropped data lost
  + Ethernet’s MAC protocol: unslotted CSMA/CD with binary backoff
* 802.3 Ethernet standards: link & physical layers
  + many different Ethernet standards
  + • common MAC protocol and frame format
  + • different speeds: 2 Mbps, 10 Mbps, 100 Mbps, 1Gbps, 10 Gbps, 40 Gbps
  + • different physical layer media: fiber, cable
  + 
* Ethernet Switch
  + link-layer device: takes an active role
    - • store, forward Ethernet frames
    - • examine incoming frame’s MAC address, selectively forward frame to one-or-more outgoing links when frame is to be forwarded on segment, uses CSMA/CD to access segment
  + transparent
    - • hosts are unaware of presence of switches
  + plug-and-play, self-learning
    - • switches do not need to be configured
* Switch: multiple simultaneous transmissions
  + hosts have dedicated, direct connection to switch
  + switches buffer packets
  + Ethernet protocol used on each incoming link, but no collisions; full duplex
    - • each link is its own collision domain
  + switching: A-to-A’ and B-to-B’ can transmit simultaneously, without collisions
    - 
* Switch Forwarding Table
  + Q: how does switch know A’ reachable via interface 4, B’ reachable via interface 5? switch with six interfaces (1,2,3,4,5,6) A A’ B B’ C C’ 1 2 5 4 3
  + A: each switch has a switch 6 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
  + 
  + 
* Switch: Frame Filtering/Forwarding
* 
* Self-Learning, forwarding: example
  + 
* Interconnecting Switches
  + self-learning switches can be connected together:
  + 
  + Q:sending from A to G - how does S1 know to forward frame destined to G via S4 and S3?
    - 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 S1, S2, S3, S4