Lecture 27

- Sections 6.4.2 and 6.4.3
  - Switched local area networks
    - Ethernet
    - Link layer switches

57

Link Layer

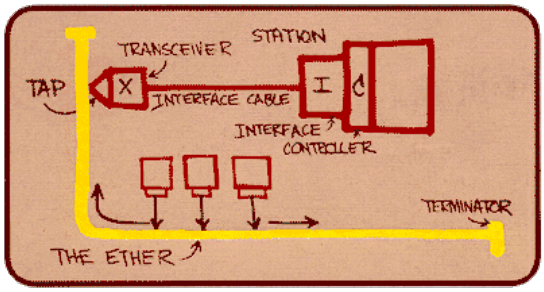
6.1 Introduction and services
6.2 Error detection and correction
6.3Multiple access protocols
6.4 Switched Local Area Networks
6.4.1 Link-Layer Addressing and ARP
6.4.2 Ethernet

58

Ethernet

"dominant" wired LAN technology:

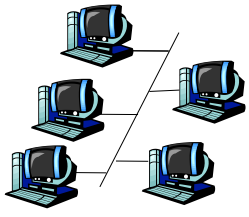
- cheap $20 (or less) for NIC
- first widely used LAN technology
- simpler, cheaper than token LANs and ATM
- kept up with speed race: 10 Mbps – 100 Gbps



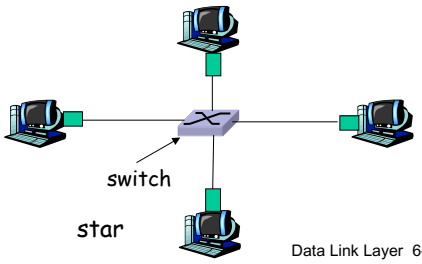
Metcalfe's Ethernet sketch

Star topology

- bus topology popular through mid 90s
  - nodes can collide with each other
- bus topology was replaced by a *hub* (dumb repeater), with each node connecting to the hub using UTP:
  - collisions may still occur
- today: star topology prevails
  - active *switch* in center (instead of hub)
  - each "spoke" runs a (separate) Ethernet protocol (nodes do not collide with each other – switch prevents collisions)



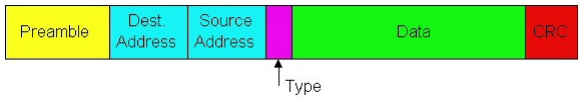
bus: coaxial cable



star

Ethernet Frame Structure

Sending adapter encapsulates IP datagram (or other network layer protocol packet) in Ethernet frame



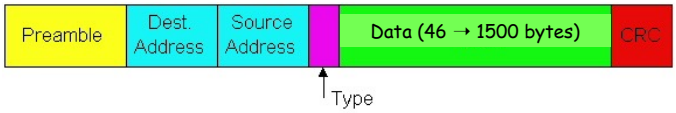
Preamble:

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

61

Ethernet Frame Structure (more)

- Addresses: 6 bytes
  - 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):
  - In IEEE 802.3 it can be used for length (values between 46 and 1500); types values are ≥ 1536
- CRC: checked at receiver, if error is detected, frame is dropped



62

Ethernet: Unreliable, connectionless

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

63

Ethernet CSMA/CD algorithm

1. NIC receives datagram from network layer, creates frame
2. If NIC senses channel idle, starts frame transmission If NIC senses channel busy, waits until channel idle, then transmits
3. If NIC transmits entire frame without detecting another transmission, NIC is done with frame !
4. If NIC detects another transmission while transmitting, aborts and sends jam signal
5. After aborting, NIC enters exponential backoff: after *m*th collision, NIC chooses *K* at random from $\{0,1,2,\ldots,2^m-1\}$. NIC waits K·slot times (slot =512 bits), returns to Step 2

64

Ethernet's CSMA/CD (more)

Jam Signal: make sure all other transmitters are aware of collision; 48 bits

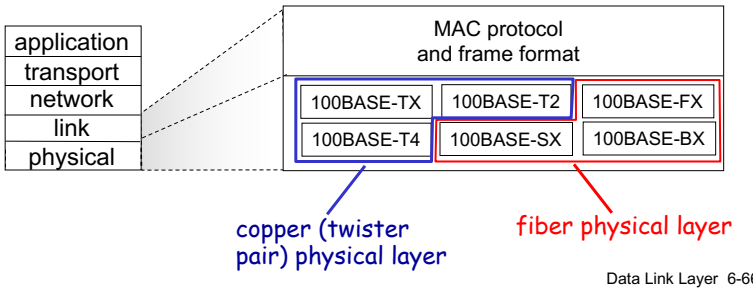
Bit time: .1 microsec for 10 Mbps Ethernet ; for K=1023, wait time is about 50 msec

Exponential Backoff:

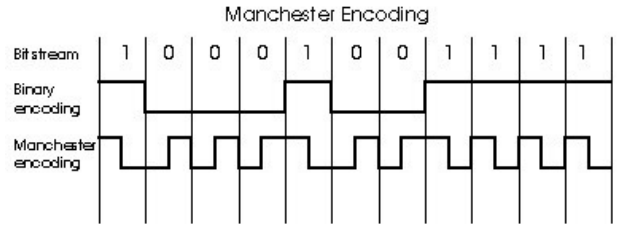
- *Goal*: adapt retransmission attempts to estimated current load
  - heavy load: random wait will be longer
- first collision: choose K from {0,1}; delay is K· 512 bit transmission times
- after second collision: choose K from {0,1,2,3}…
- after ten collisions, choose K from {0,1,2,3,4,…,1023}

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, 10Gbps, 40Gbps, 100Gbps
  - different physical layer media: fiber, cable



Clock Synchronization: Manchester encoding



- used in 10BaseT
- each bit has a transition
- allows clocks in sending and receiving nodes to synchronize to each other
  - no need for a centralized, global clock among nodes!

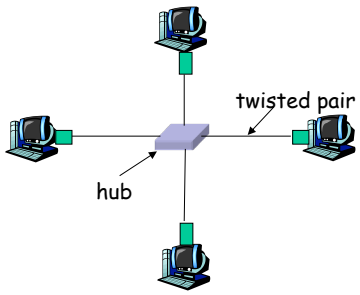
Link Layer

6.1 Introduction and services
6.2 Error detection and correction
6.3Multiple access protocols
6.4 Switched Local Area Networks
  6.4.1 Link-Layer Addressing and ARP
  6.4.2 Ethernet
  6.4.3 Link layer switches and LANs

Ethernet Hubs

… physical-layer ("dumb") repeaters:

- bits coming in one link go out *all* other links at same rate
- all nodes connected to hub can collide with one another
- no frame buffering
- no CSMA/CD at hub: host NICs detect collisions



69

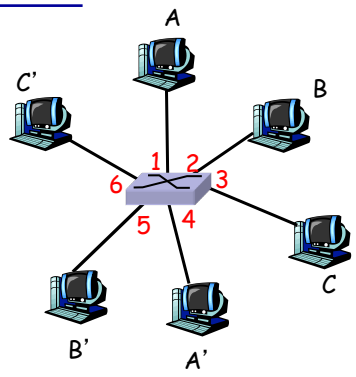
Ethernet Switch

- link-layer device: smarter than hubs, take *active* role
  - supports multiple transmissions
  - stores & forwards Ethernet frames (to avoid collisions)
  - examines 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 are not pre-configured: they learn MAC addresses of nodes connected to switch ports

70

Ethernet Switch: allows *multiple* simultaneous transmissions

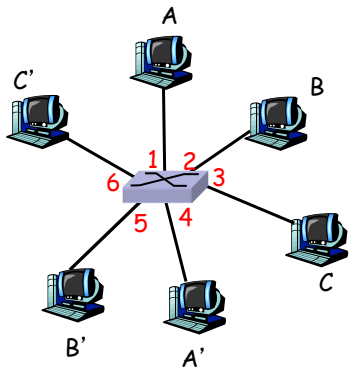
- hosts have dedicated, direct connection to switch
- switches buffer packets
- Ethernet protocol used on *each* incoming link, but no collisions; full duplex
  - each link is its own collision domain
- *switching:* A-to-A' and B-to-B' simultaneously, without collisions
  - not possible with dumb hub



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

Switch Table

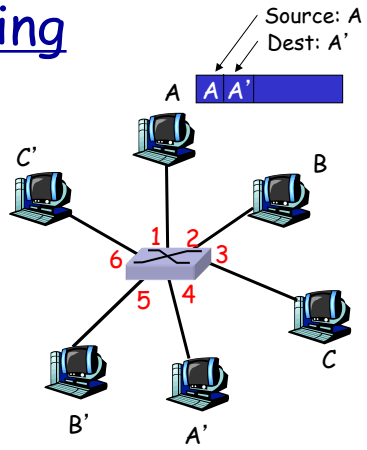
- *Q:* how does switch know that A' reachable via interface 4, B' reachable via interface 5?
- *A:* each switch has a switch table, each entry:
  - (MAC address of host, interface to reach host, time stamp)
- looks like a routing table!
- *Q:* how are entries created, maintained in switch table?



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

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

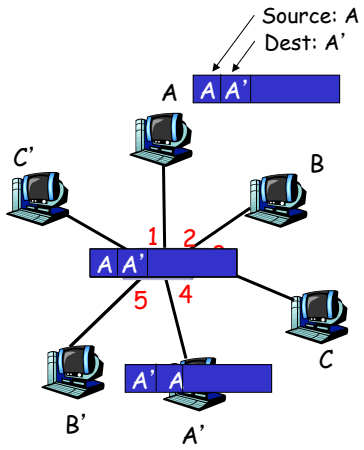
When frame received:

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

forward on all but the interface on which the frame arrived

Self-learning, forwarding: example

- frame destination unknown: *flood*
- destination A location known: *selective send*

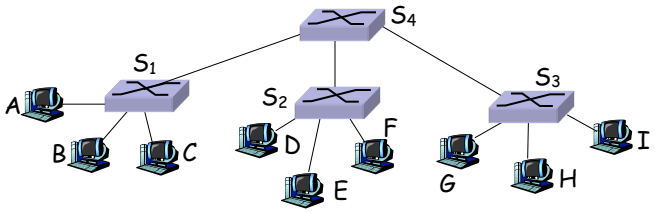


MAC addr	interface	TTL
A	*1*	*60*
A'	*4*	*60*

Switch table (initially empty)

Interconnecting switches

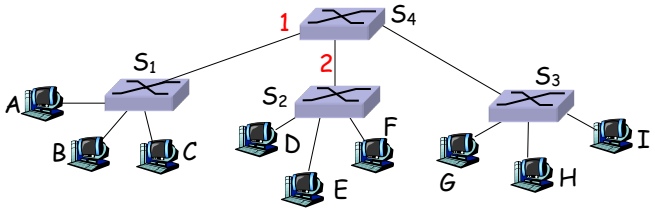
- switches can be connected together



- *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

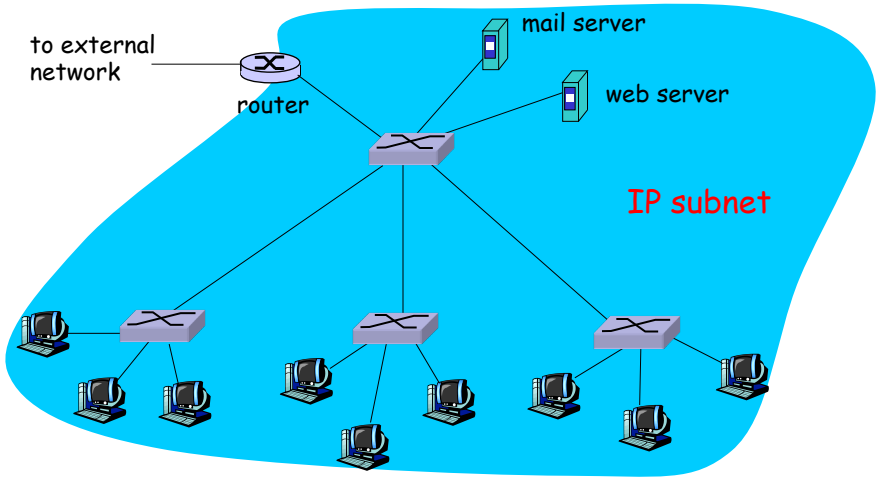
Suppose C sends frame to I, I responds to C



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

77

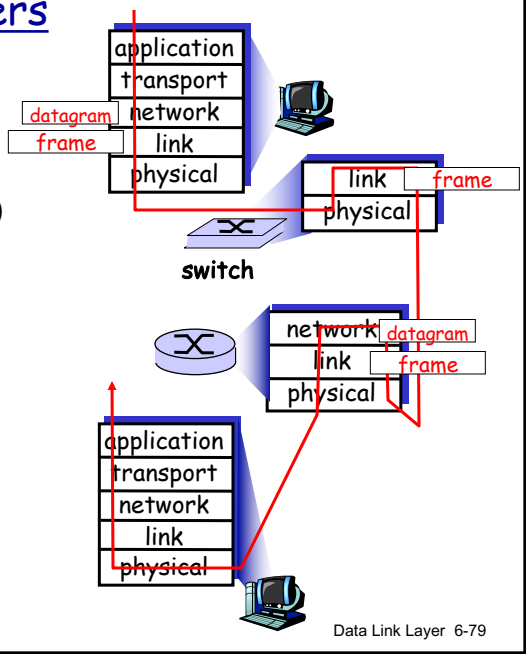
Institutional network



78

Switches vs. Routers

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



Chapter 6: Summary

- principles behind data link layer services:
  - error detection, correction
  - MAC protocols and multiple access: channel partitioning, random access, and taking turns
  - link layer addressing
- instantiation and implementation of various link layer technologies
  - Ethernet
  - switched LANS