Computer Networks

Ethernet
Hubs and Switches

Based on Computer Networking, 4th Edition by Kurose and Ross

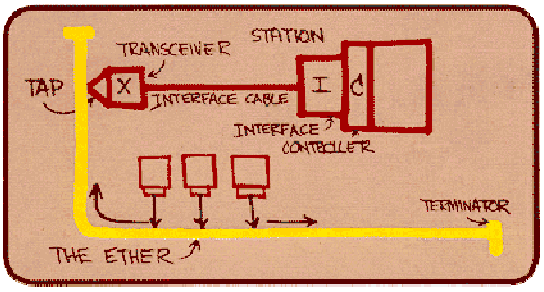
Stan Kurkovsky

Ethernet

"dominant" wired LAN technology:

- cheap $20 for NIC
- first widely used LAN technology
- Simpler, cheaper than token LANs and ATM
- Kept up with speed race: 10 Mbps – 10 Gbps
- Also referred to as 802.3 LAN (IEEE 802.3 working group)

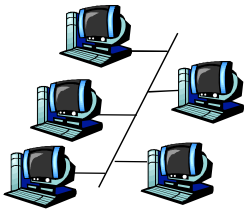
Metcalfe's Ethernet sketch



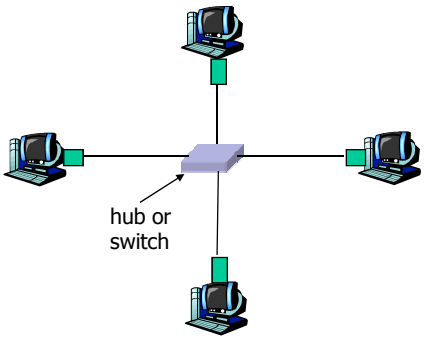
Stan Kurkovsky

Star topology

- Bus topology popular through mid 90s
  - all nodes in same collision domain (can collide with each other)
- Now star topology prevails
- Connection choices: hub or switch (more later)
  - active *switch* in center
  - each "spoke" runs a (separate) Ethernet protocol (nodes do not collide with each other)



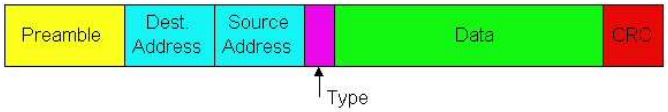
bus: coaxial cable



Stan Kurkovsky

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
- Addresses: 6 bytes
  - if adapter receives frame with matching destination address, or with broadcast address (eg ARP packet), it passes data in frame to net-layer protocol
  - otherwise, adapter discards frame
- Type: indicates the higher layer protocol (mostly IP but others may be supported such as Novell IPX and AppleTalk)
- CRC: checked at receiver, if error is detected, the frame is simply dropped

Stan Kurkovsky

Unreliable, connectionless service

- Connectionless: No handshaking between sending and receiving adapter.
- Unreliable: receiving adapter doesn't send acks or nacks to sending adapter
  - stream of datagrams passed to network layer can have gaps
  - gaps will be filled if app is using TCP
  - otherwise, app will see the gaps

- Ethernet's MAC protocol: unslotted CSMA/CD
  - adapter doesn't transmit if it senses that some other adapter is transmitting, that is, carrier sense
  - transmitting adapter aborts when it senses that another adapter is transmitting, that is, collision detection
  - Before attempting a retransmission, adapter waits a random time, that is, random access

Stan Kurkovsky

Ethernet CSMA/CD algorithm

1. Adaptor receives datagram from net layer & creates frame
2. If adapter senses channel idle, it starts to transmit frame. If it senses channel busy, waits until channel idle and then transmits
3. If adapter transmits entire frame without detecting another transmission, the adapter is done with frame!
4. If adapter detects another transmission while transmitting, aborts and sends jam signal
5. After aborting, adapter enters **exponential backoff**: after the mth collision, adapter chooses a K at random from $\{0,1,2,\ldots,2^m-1\}$. Adapter waits $K \cdot 512$ bit times and returns to Step 2

Stan Kurkovsky

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}

Stan Kurkovsky

CSMA/CD efficiency

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

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

- Efficiency goes to 1 as t_{prop} goes to 0
- Goes to 1 as t_{trans} goes to infinity
- Much better than ALOHA, but still decentralized, simple, and cheap

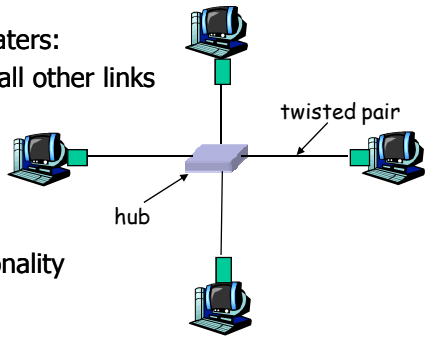
Stan Kurkovsky

10BaseT and 100BaseT

- 10/100 Mbps rate; latter called "fast ethernet"
- T stands for Twisted Pair
- Nodes connect to a hub: "star topology"; 100 m max distance between nodes and hub

Hubs are essentially physical-layer repeaters:

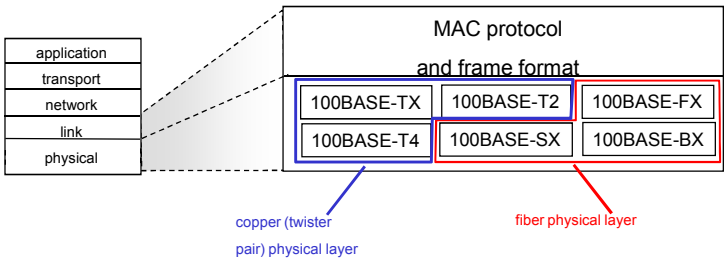
- bits coming from one link go out all other links
- at the same rate
- no frame buffering
- no CSMA/CD at hub: adapters detect collisions
- provides net management functionality



Stan Kurkovsky

802.3 Ethernet Standards: Link & Physical Layers

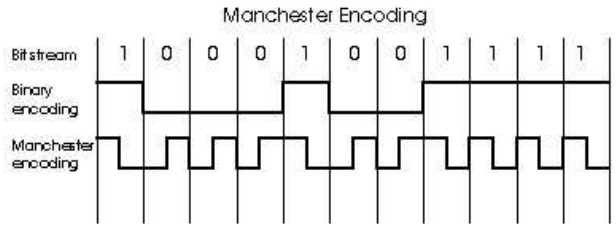
- *many* different Ethernet standards
  - common MAC protocol and frame format
  - different speeds: 2 Mbps, 10 Mbps, 100 Mbps, 1Gbps, 10G bps
  - different physical layer media: fiber, cable



Stan Kurkovsky

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!
- Hey, this is physical-layer stuff!



Stan Kurkovsky

Gbit Ethernet

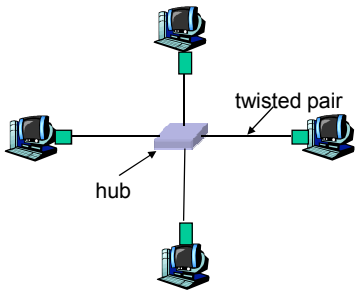
- 802.3z
- uses standard Ethernet frame format
- allows for point-to-point links and shared broadcast channels
- in shared mode, CSMA/CD is used; short distances between nodes required for efficiency
- uses hubs, called here "Buffered Distributors"
- Full-Duplex at 1 Gbps for point-to-point links
- 10 Gbps now (802.3ae)

Stan Kurkovsky

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



Stan Kurkovsky

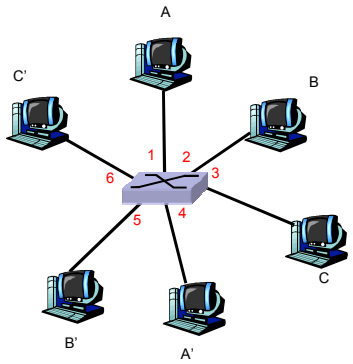
Switch

- link-layer device: smarter than hubs, take *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

Stan Kurkovsky

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 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?
    - something like a routing protocol?

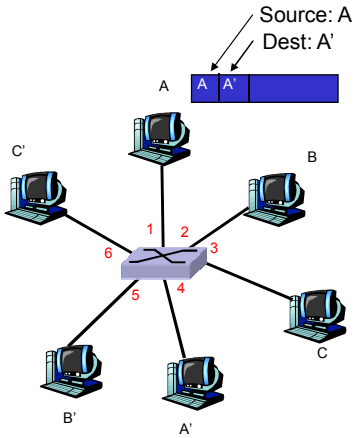


switch with six interfaces
(1,2,3,4,5,6)

Stan Kurkovsky

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)

Stan Kurkovsky

Filtering/Forwarding

When switch receives a frame:

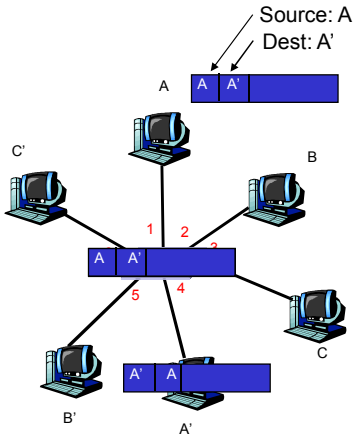
1. record link associated with sending host
2. index switch table using MAC dest address
3. if entry found for destination

```
        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
```

Stan Kurkovsky

Self-learning, forwarding: example

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

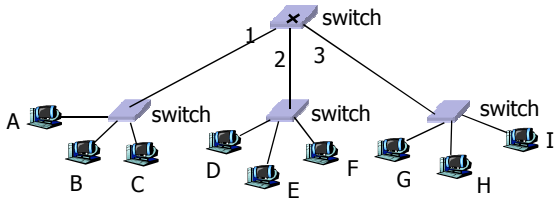


MAC addr	interface	TTL
A	1	60
A'	4	60

Switch table
(initially empty)

Stan Kurkovsky

Interconnecting switches



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

Suppose C sends frame to D

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

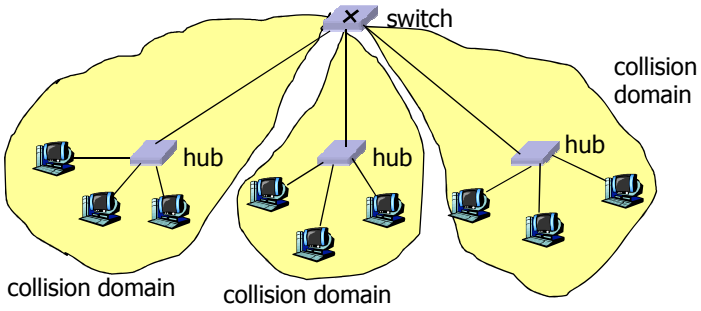
Suppose D replies back with frame to C

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

Stan Kurkovsky

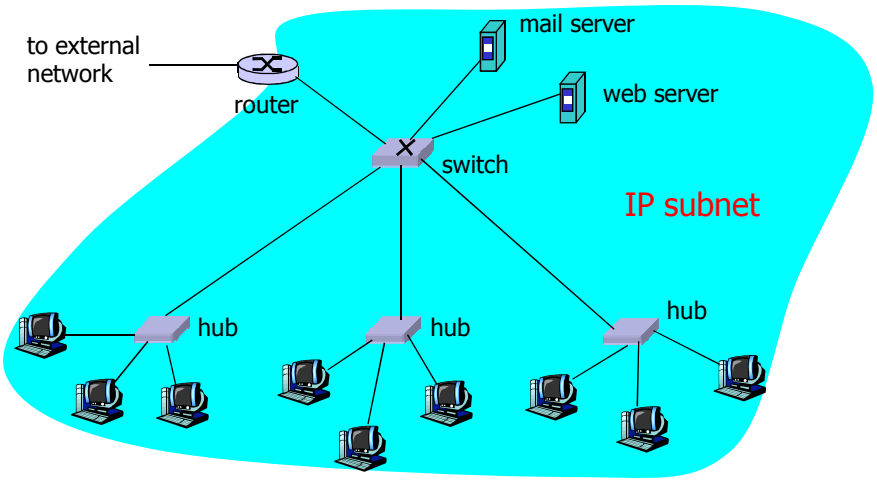
Switch: traffic isolation

- switch installation breaks subnet into LAN segments
- switch filters packets:
  - same-LAN-segment frames not usually forwarded onto other LAN segments
  - segments become separate collision domains



Stan Kurkovsky

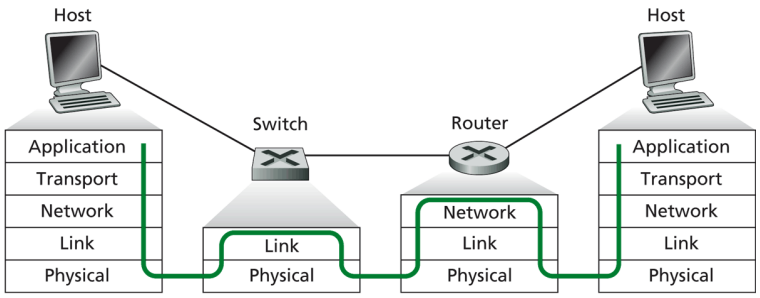
Institutional network



Stan Kurkovsky

Switches vs. Routers

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



Stan Kurkovsky

Summary comparison

	hubs	routers	switches
traffic isolation	no	yes	yes
plug & play	yes	no	yes
optimal routing	no	yes	no
cut through	yes	no	yes

Stan Kurkovsky