Link Layer Technologies

Ethernet

$\rightarrow$ CSMA/CD for backward compatibility

$\rightarrow$ switched

$\rightarrow$ copper, fiber

Recent speeds:

- 1000Base-T (GigE): 1 Gbps over copper

  $\rightarrow$ version over fiber optic

- 10, 40, 100, 200, 400 Gbps Ethernet: fiber; cat 6, 7 copper

  $\rightarrow$ IEEE 802.3z

- 800 Gbps, 1.6 Tbps

  $\rightarrow$ IEEE 802.3dj

  $\rightarrow$ fiber

Ethernet technology evolution:
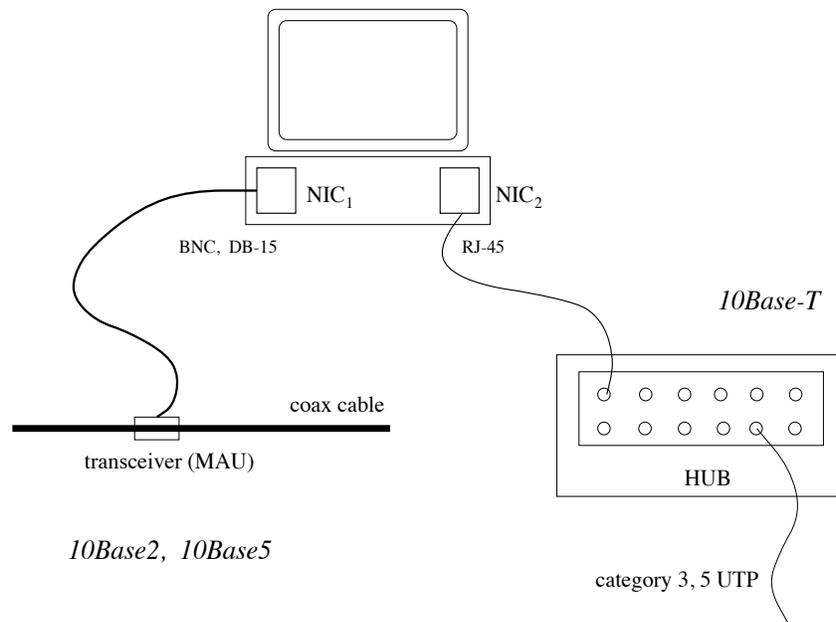
$\rightarrow$ each NIC assigned unique 48-bit address

From bus, hub to switch

$\rightarrow$ computer with specialized hardware support

But bus technology reused in powerline networks

$\rightarrow$ IEEE 1901 (HomePlug)

48-bit address:

$\rightarrow$ called MAC or hardware address
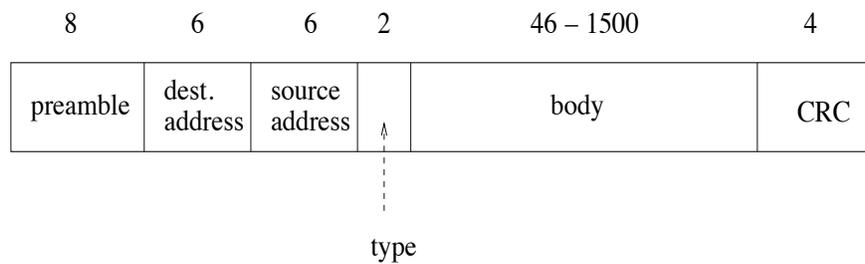
$\rightarrow$ broadcast address: all 1's

Each Ethernet frame has "From" (source) and "To" (destination) address.

$\rightarrow$ WiFi frame has 3–4 addresses

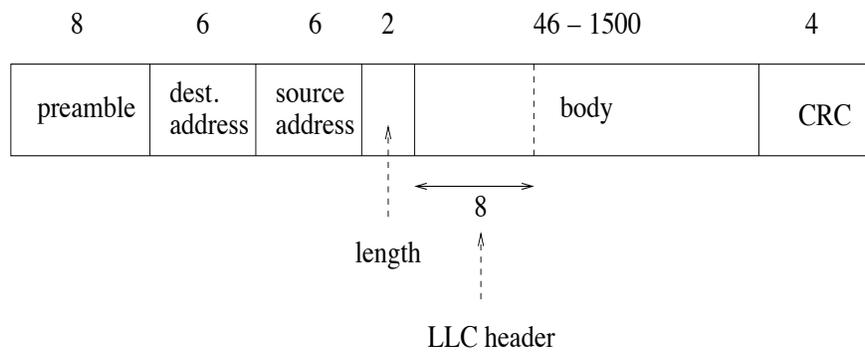NIC at receiver: accepts frames with matching destination address

- default
- can accept all frames

  $\rightarrow$ promiscuous mode

  $\rightarrow$ requires root privilege

  $\rightarrow$ useful for traffic monitoring/sniffing

DIX Ethernet frame:

8	6	6	2	46 – 1500	4
preamble	dest. address	source address		body	CRC

type

IEEE 802.3 Ethernet frame:

8	6	6	2		46 – 1500	4
preamble	dest. address	source address			body	CRC

length

8

LLC header

$\rightarrow$ type: e.g., 0x0800 for IPv4

$\rightarrow$ DIX dominant: incorporated into IEEE standard

Ethernet MAC protocol: CSMA/CD

- MA (Multiple Access): multiple nodes are allowed simultaneous access

  → just send

- CS (Carrier Sense): can detect if some other node is using the link

  → rule: if busy, wait until channel is not busy

- CD (Collision Detection): can detect frame collision stemming from simultaneous transmissions

  → rule: if collision, try later

Collision detection mechanism:

$\rightarrow$ while sending monitor link

$\rightarrow$ if other signal/activity is detected: collision

If collision occurs:

$\rightarrow$ failure to decode bits carried by arriving signal

$\rightarrow$ same holds at other end

Frame retransmitted when collision detected

$\rightarrow$ stop-and-wait with negative ACK

$\rightarrow$ collision event acts as negative ACK

$\rightarrow$ not through timeout

How to guarantee collision detection?

Through standard specification: 10 Mbps Ethernet specification

$\rightarrow$ link cannot be longer than 2500 m

$\rightarrow$ measured RTT: 51.2 μs

To ensure collision is detected:

$\rightarrow$ sender must stay active for 51.2 μ

$\rightarrow$ worst-case collision scenario

For 10 Mbps Ethernet

$\rightarrow$ frame size must be at least 512 bits

$\rightarrow$ i.e., 512 bits = 10 Mbps $\times$ 51.2 μs

$\rightarrow$ use padding if data is less than 64 bytes

Transmit at least 512 bits for CD:

$\rightarrow 6 + 6 + 2 + 46 + 4 = 64$ B $= 512$ bits

$\rightarrow$ minimum payload size of Ethernet frame

8	6	6	2	46 – 1500	4
preamble	dest. address	source address		body	CRC

type
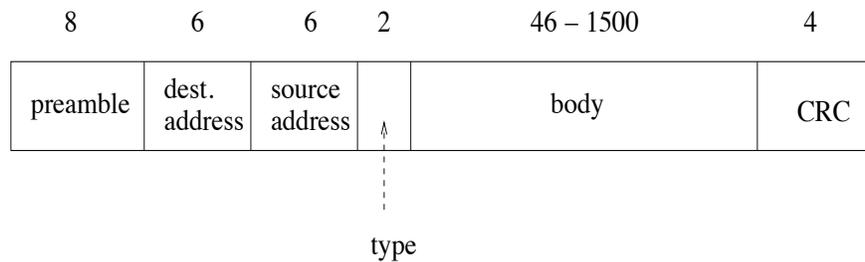
To achieve collision detection (CD) in 100 Mbps Ethernet:

$\rightarrow$ length restriction (250 m)

Too restrictive

$\rightarrow$ not needed in switched Ethernet

$\rightarrow$ two Ethernet switches connected via full-duplex link disable CD (and CS)

Upon collision: when attempt retransmission?

$\rightarrow$ try to avoid repeat collision

$\rightarrow$ randomize attempt period

$\rightarrow$ increase attempt period if collision repeats

Retransmission protocol: exponential backoff

1. Wait for random $0 \leq X \leq 51.2\ \mu s$ before 1st retry

2. Two consecutive collisions: wait for random $0 \leq X \leq 102.4\ \mu s$ before 2nd retry

3. Three consecutive collisions: wait for random $0 \leq X \leq 204.8\ \mu s$ before 3rd retry

4. i consecutive collisions: wait for $0 \leq X \leq 2^{i-1}\,51.2\ \mu s$ before next attempt

5. Give up if $i > 16$

$\rightarrow$ why exponential backoff?

Today: switched Ethernet with full-duplex links

- not shared bus anymore

  → every device connected by point-to-point link to switch

  → sender/receiver cannot collide

- switch: computer

  → with special hardware support to speed up packet handling

- arriving Ethernet frames subject to scheduling

  → e.g., FIFO, priority, fair queueing

  → finite buffers

  → frame losses due to buffer overflow, not collision