Data Link Communication: Technology and Access Control

Point-to-point communication

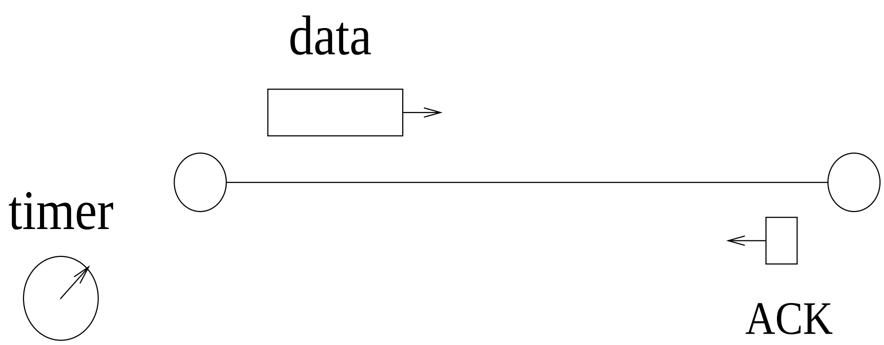
Already seen digital/analog transmission of digital data including coding and error detection.

Reliable transmission

Principal methodology: ARQ (Automatic Repeat reQuest) or PAR (Positive Acknowledgment with Retransmission) or backward error correction (BEC).

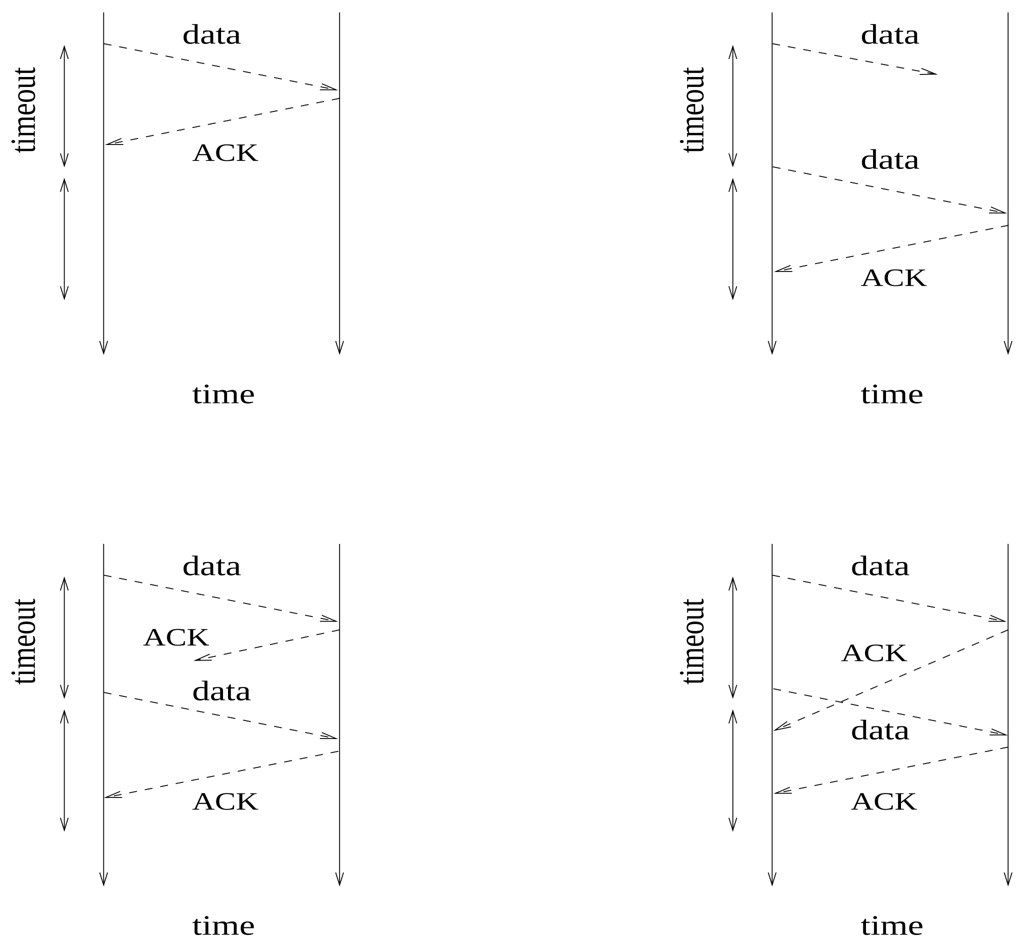
Three components:

- acknowledgment (ACK)
- timeout
- retransmit



Stop-and-wait

Assumption: Frame is "lost" due to corruption; discarded by NIC after error detection.



Issue of RTT (Round-Trip Time) and timer management:

- what is proper value of timer?
- RTT estimation
- easier for single link than *path* in an internetwork
- largely independent of queueing effect

More serious problem: Not keeping the pipe full.

$\longrightarrow$ bandwidth-delay product

Literally, volume of data in on the link.

To achieve high utilization, want to keep volume of traffic flowing close to the bandwidth limit.

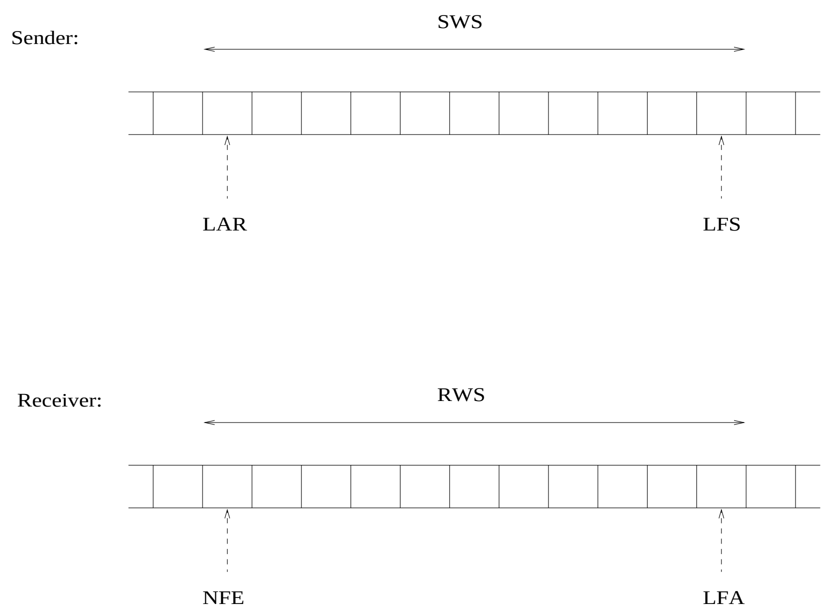
Example: Link BW 1.5 Mbps, 45 ms RTT; bandwidth-delay product = 1.5 Mbps $\times$ 45 ms = 67.5 kb $\approx$ 8 kB.

If frame size 1 kB, then (effective) throughput is $1024 \times 8/0.045 = 182$ kbps; utilization is only 0.125.

Solution: Other things being equal, must increase frame size.

- straightforward increase of frame size is problematic; why?
- send blocks of data, i.e., sequence of frames
- creates management problem

Sliding window protocol



- SWS: Send Window Size
- RWS: Receiver Window Size
- LAR: Last ACK Received
- LFS: Last Frame Sent
- NFE: Next Frame Expected
- LFA: Last Frame Acceptable

Assign sequence number (SeqNum) to individual frames.

Maintain invariants:

- $\text{LFS} - \text{LAR} + 1 \leq \text{SWS}$
- $\text{LFA} - \text{NFE} + 1 \leq \text{RWS}$

Sender: Update LAR, send more frames, then update LFS.

Receiver: Cumulative ACK; let SeqNumToAck denote the largest sequence number not yet acknowledged.

- $\text{NFE} \leftarrow \text{SeqNumToAck} + 1$
- $\text{LFA} \leftarrow \text{SeqNumToAck} + \text{RWS}$

ACK variants:

- piggyback
- negative ACKs
- selective ACKs

Sequence number wrap-around problem:

$$\text{SWS} < (\text{MaxSeqNum} + 1)/2.$$

$\longrightarrow$ similar to stop-and-wait (binary)

Further optimization/control variables in end-to-end case?

Why can packets still be lost given that link layer achieves reliability?

Link-based flow/congestion control revival (H. T. Kung). Achieve flow control/multiplexing (buffer sharing)/reliability at link level.

Multi-access communication

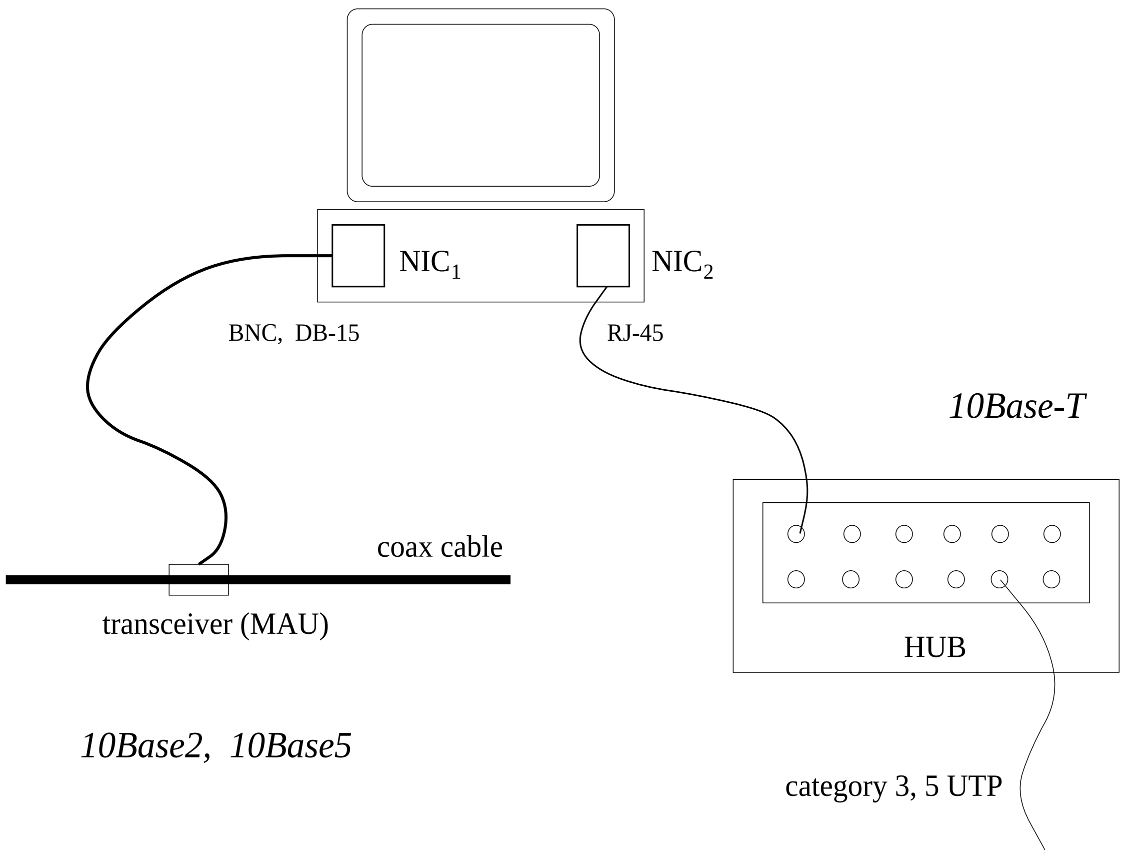
Ethernet and CSMA/CD

$\longrightarrow$ copper, optical fiber

Types:

- 10Base2 (ThinNet): coax, segment length 200 m, 30 nodes/segment
- 10Base5 (ThickNet): coax, segment length 500 m, 100 nodes/segment
- 10Base-T: twisted pair, segment length 100 m, 1024 nodes/segment
- 10Base-F: fiber, segment length 2000 m, 1024 nodes/segment
- 100Base-T (Fast Ethernet): category 5 UTP, fiber (also 100VG-AnyLAN)
- Gigabit Ethernet: fiber, category 5 UTP

Connectivity example:



⟶ bus/star configuration

⟶ multihomed/singlehomed

⟶ unique Ethernet address per NIC

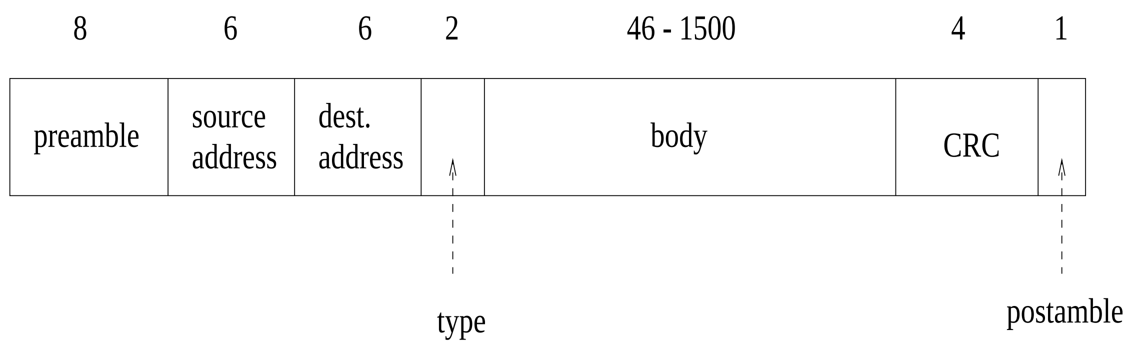
Segments can be hooked up by repeaters, bridges, gateways, (hub) switches.

- maximum of 2 (4 for IEEE 802.3) repeaters between two hosts; 1500 m
- for Fast Ethernet, 2 repeater hops

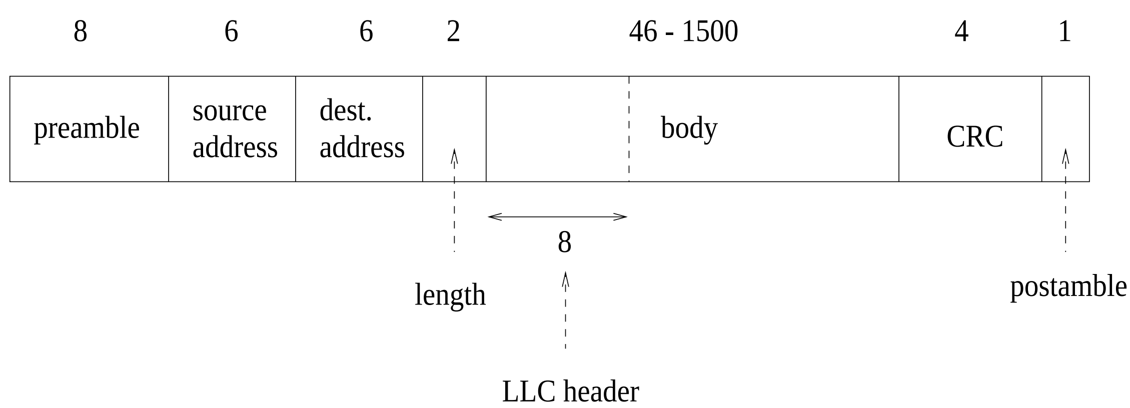
High-bandwidth Ethernets have *shorter* network diameter.

- about 2500 m for 10 Mbps Ethernet
- about 200 m for 100 Mbps Ethernet
- even shorter for 1 Gbps Ethernet

DIX Ethernet frame:



IEEE 802.3 Ethernet frame:



$\longrightarrow$ IEEE 802.2 LLC (Logical Link Control)

Encoding: Manchester

Addressing:

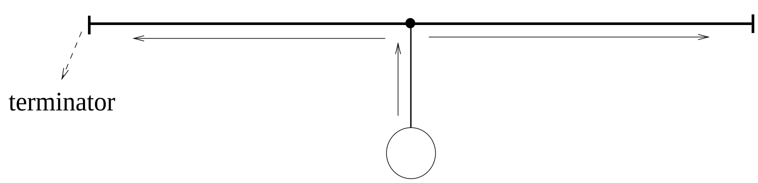
- 48 bit unique address
- point-to-point
- broadcast (all 1's)

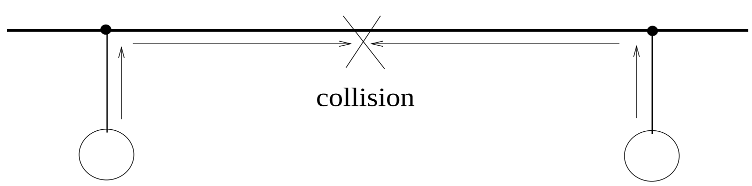
Receiver: Ethernet adaptor accepts frames with relevant address.

- accepts only own frame address
- accepts all frames (promiscuous mode)

MAC (Medium Access Control): CSMA/CD

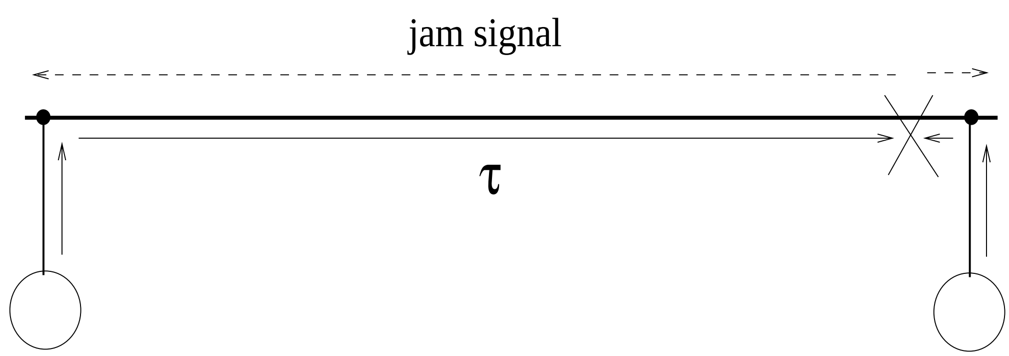
- *CS (Carrier Sense)*: Can detect if some other node is using the link.
- *MA (Multiple Access)*: Many nodes are allowed to simultaneously access the link.
- *CD (Collision Detection)*: Can detect if simultaneous access has occured (corrupted signal).





Ethernet is 1-*persistent* MA scheme; more generally, p-persistent where $0 < p \leq 1$.

Worst-case collision scenario:



- sender (worst case) needs to wait 2τ sec before detecting collision
- for 2500 m length, 51.2 μs round-trip time (2τ)
- enforce 51.2 μs slot (jam) time
- at 10 Mbps, 512 bits; i.e., minimum frame size

Hence, upon collision detection:

- Make sure to transmit at least 512 bits

  $\longrightarrow$ $2 \times$ bandwidth-delay product

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

- exponential backoff; wait for $0 \leq X \leq 51.2\ \mu$s before next attempt
- on i'th collision, wait for $0 \leq X \leq (2^i - 1)51.2\ \mu$s before next attempt, $i \leq 16$
- $X = 0\ \mu\text{s}, 51.2\ \mu\text{s}, 2 \times 51.2\ \mu\text{s}, 3 \times 51.2\ \mu\text{s}, \ldots$

$\longrightarrow$ distributed bus arbitration mechanism