The University Of Alabama in Huntsville Computer Science

cs570

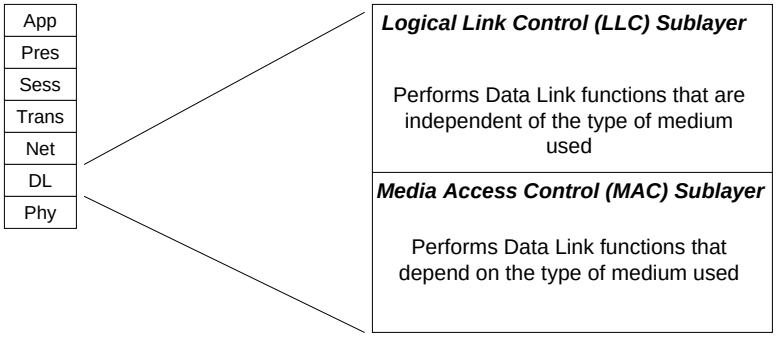
The MAC Sublayer of L2

The University Of Alabama in Huntsville Computer Science

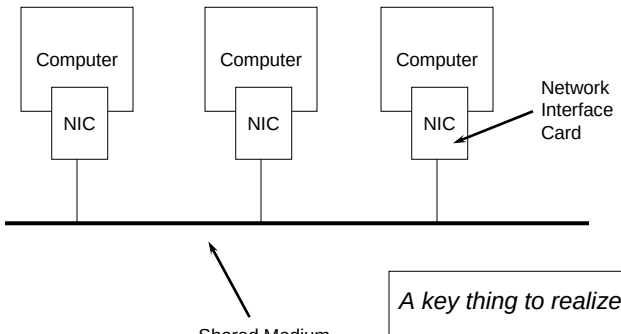
cs570

The MAC Sublayer

The data link layer is usually thought of (and often implemented) as two distinct sublayers performing different functions



Shared ("Multidrop") LANs



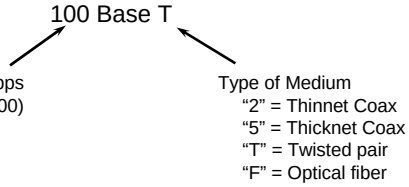
A key thing to realize:

Every node on the LAN sees every frame of data

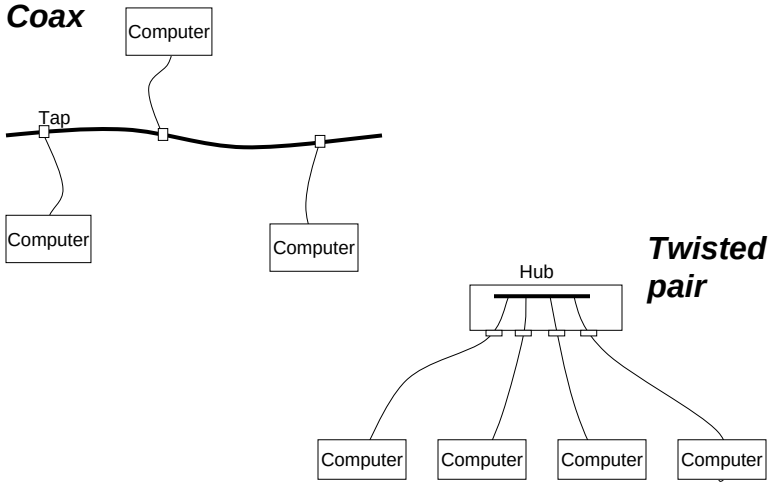
Ethernet

- The most common LAN technology
- IEEE Standard 802.3
- Simple electronics, simple protocols
- Versions with Bandwidths from 10Mbps to 10Gbps

Nomenclature:



Ethernet wiring



How does Ethernet work?

The basic ideas:

- *carrier sense* (CS): Every station can tell if the line is idle
- *multiple access* (MA): All stations have equal access to the line
- *collision detection* (CD): Stations monitor the line while they send. They can tell if there is a "collision" with another station's frame. (This is called "listen while talk")

= "CSMA/CD"

Ethernet frame format

bytes:	8	6	6	2	0 to 1500	0 to 46	4
	Preamble	**Destination Address**	**Source Address**	**Type/ Size**	**Data**	**Pad**	**CRC**
	101010…. used as a synch signal	6-byte MAC addresses (e.g., 04:A2:61:B4:CD:00) Destination addresses with high-order bit = 1 are multicast addresses. All 1's is a broadcast. Every MAC address is unique world-wide		# bytes in Data field	An Ethernet frame can carry up to 1500 bytes of data. The frame, exclusive of Preamble, must be at least 64 bytes long. So there must be at least 46 bytes in the Data field. If there are fewer, pad bytes are added.		CRC-32 error detection

Example Ethernet frames

100 bytes of data

6	6	2	100	4
Destination Address	**Source Address**	**100**	**Data**	**CRC**

10 bytes of data

6	6	2	10	36	4
Destination Address	**Source Address**	**10**	**Data**	**Pad**	**CRC**

Data field padded to 46 bytes

Ethernet MAC Algorithms

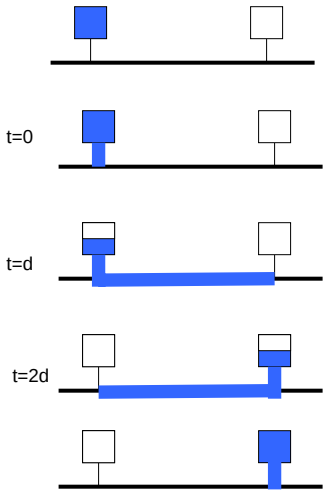
SEND

While there are frames to send
    Listen for a period of time ("Inter-Frame Gap").
    If line is idle, then
        Listen while sending the frame.
        If there is a collision while sending then
            send a "Jamming signal" and stop
            wait an amount of time determined by the "exponential backoff" algorithm
        End
    End
End

RECEIVE

When the start of a frame is sensed
    Check the destination address. Copy the frame into the receive buffer if:
        1. It is addressed to my MAC address, or
        2. It is a broadcast frame, or
        3. It is a multicast frame addressed to my multicast group, or
        4. I am in "promiscuous" mode

Sending a frame



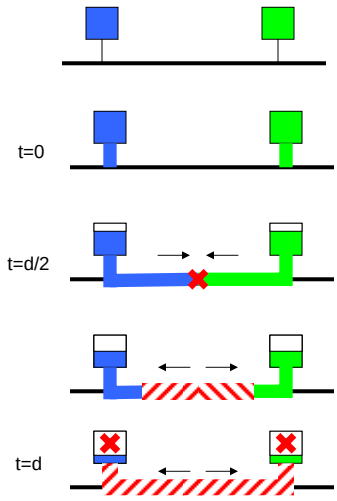
Sender senses the line and find it idle

Begins sending to a node at the far end of the LAN

Call the end-to-end latency of the ethernet "d"

The amount of time needed to send a frame can't be less than 2d (we'll see why later)

Collisions



Two nodes sense the line and find it idle. Assume they are at the opposite ends of the Ethernet.

Both begin sending

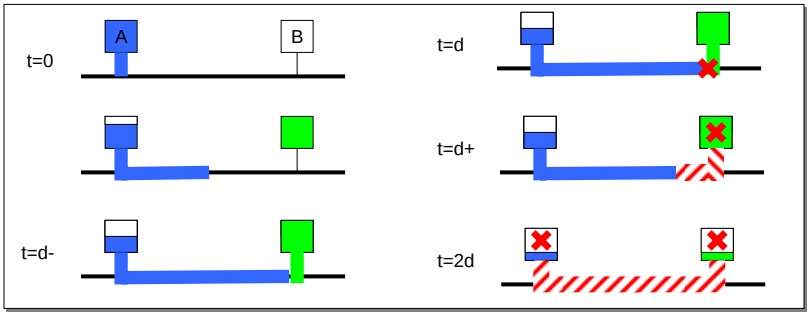
The frames "collide"

The result of the collided frames is an interference pattern that is gibberish. The gibberish propagates throughout the Ethernet.

The sending nodes, doing "listen while talk" see that the data on the line doesn't match what they are sending. This tells them a collision has occurred.

The "longest" collision

Two nodes, A and B, at opposite ends of the Ethernet. A sends, then B sends just as A's frame reaches B



In order for A to know about the collision of a frame sent at t=0, A has to still be sending at t=2d.

This means that **frames must be long enough to take at least 2d to send**.

Ethernet parameters

The parameters of a particular type of Ethernet are specified in the particular IEEE standard. These are usually set as a result of engineering trades and safety margins (and they usually don't match the theoretical values).

Parameters:

- Min length of a frame
- Min time required to send the min frame ("slot time")
- Max length of the Ethernet (coax) or of connections to the hub (TWP)
- Max number of computers, repeaters, etc

	Min frame length	Slot time
10Mbps	512 bits	51.2 uS
100Mbps	512 bits	5.12 uS
1Gbps	4096 bits	4.096 uS

The Exponential Backoff algorithm

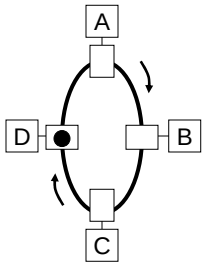
Problem:
After a collision, how long do the senders wait before trying again? We need a way to stagger or randomize their wait intervals, but we want to minimize the wait times.

Exponential Backoff Algorithm forces senders to wait a # of slot times that is randomly picked from a list. The list grows as the number of successive collisions grows. After n collisions, there are 2^n numbers in the list.

Number of collisions in a row	Number of Slot times colliding senders will wait before re-trying	Probability of another collision on the next round
1	0 or 1	0.5
2	0, 1, 2, or 3	0.25
3	0,1,2,3,4,5,6 or 7	0.125
n	$0 \ldots 2^n-1$	2^{-n}

Note: n is limited so that NIC failures can be detected (type: n<=16).

Token Ring (IEEE 802.5)

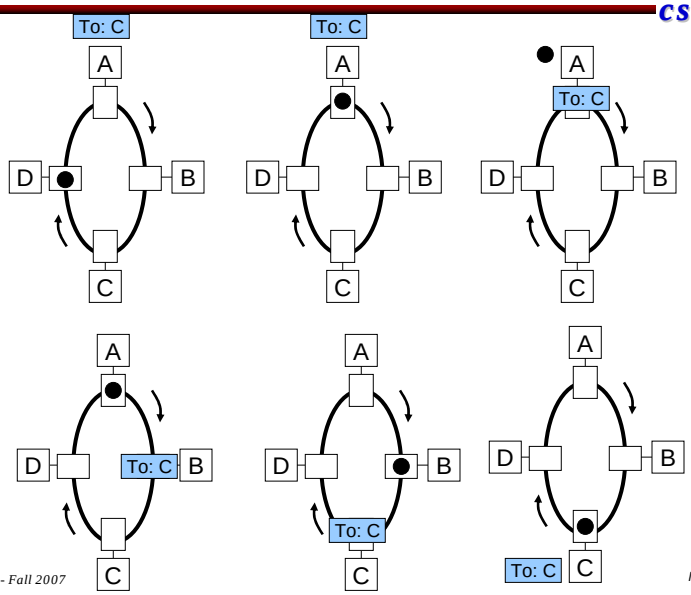


Frames are passed from node to node around the ring.

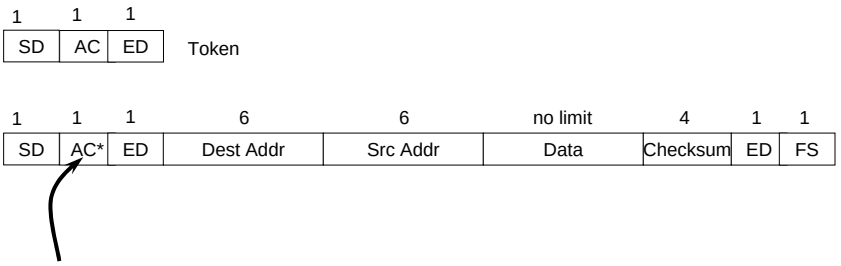
There is a special “token” frame that circulates on the ring.

To send, a node waits for the token, removes it from the ring, sends its frame, then replaces the token.

Token Ring operation



Token Ring operation



Bit flipped to indicate that this is a data frame, not the token

Note similarity to Ethernet frame, especially MAC address fields

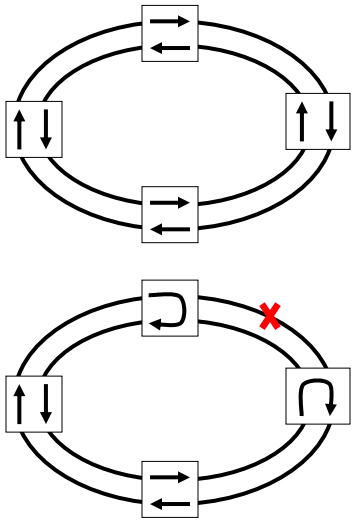
Some Token Ring Parameters

- 4 and 16 Mbps
- 250 stations per ring max
- Token Holding Time (max time a node can hold the token -- prevents hogging): default=10msec

A problem with token rings

When there is a break anywhere in the ring, all communication everywhere is lost

Fiber Distributed Data Interface (FDDI)



Dual counter-rotating rings provide some ability to tolerate a broken ring

FDDI Parameters
- 100 Mbps
- 500 stations max
- up to 2 Km between stations