Lecture 4

In this lecture, we will discuss about the

- Fabrication of data link layer into MAC and LLC
- Channel Allocation Problem
- Multiple Access Protocol
- Data Link Layer protocols for Local Network (802.3 et al)

Data Link Layer is further subdivided into two parts, as shown below

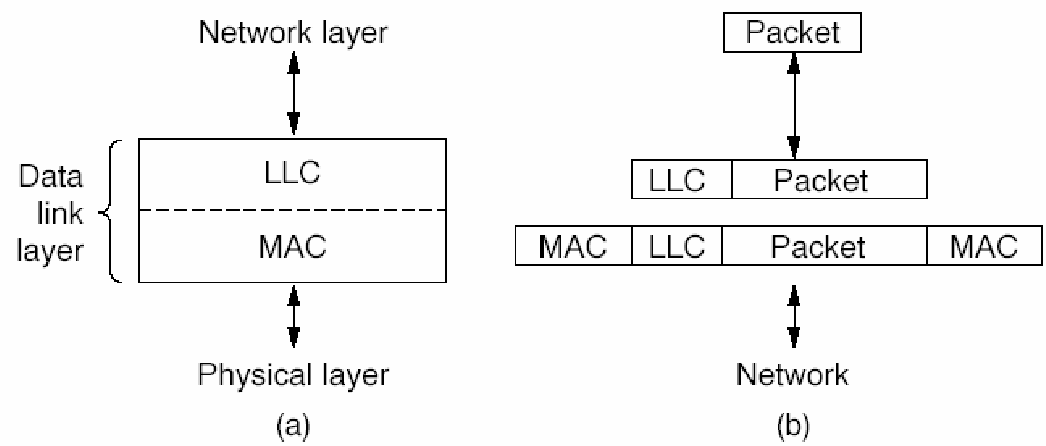


Fig. 4-24. (a) Position of LLC. (b) Protocol formats.

One part of division is LLC (Logical Link Control), and other part is MAC (Medium Access Control).
LLC sub layer attaches its own header to the packet received from the network layer, and give it to MAC. After this MAC sub layer attaches its header and trailer, thus making it frame.

Over here, we are much interested for MAC sub layer. The very first thing is why do we require it?
As we know, networks can be divided into two categories: point to point channels and the other one multipoint channels. In a multipoint channel, many users can access the channel simultaneously. Such channels are also known as multi-access channels or random access channels. The protocols used to determine who goes next on a multi-access channel belong to the sub layer of the data link layer called the MAC (Medium Access Control) sub layer. The MAC sub layer is important in LAN where multi-access channels are used.

So now, we will discuss how to allocate a single broad cast channel among multiple users. This problem of allocating the single broad cast channel among multiple users is called as channel allocation problem.

We can further classify the solution to this problem into types

1. Static Channel Allocation in LAN and MAN
   (i) FDM
   (ii) TDM
2. Dynamic Channel Allocation
   (i) Pure Aloha
   (ii) Slotted Aloha
   (iii) Persistent CSMA
   (iv) Non Persistent CSMA
   (v) CSMA/CD
   (vi) Bit Map Protocol
   (vii) Binary Countdown

Static channel allocation is one of the solutions of allocating channel. If we use FDM, i.e. allocate the bandwidth of the channel among n users, and TDM, where a frame is divided into n slots. We had a discussion on these topics earlier also.

Dynamic Channel Allocation

(i) Pure ALOHA

The very first algorithm we will start with is pure ALOHA. The basic idea of an ALOHA system is simple; let users transmit whenever they have data to be sent. There will be collisions (If two frames are transmitted simultaneously, they overlap in time and the resulting signal is garbled, this event is called as **collision**. Sometime the first bit of one frame collides with the last bit of the frame ahead of it). Due to collisions colliding frame will be destroyed, and there will be a feed back to the transmitting stations. Actually after transmitting a frame, station listens for an amount of time equal to the propagation delay on the network. If the station fails to receive an acknowledgment or if the frame is destroyed, the sender waits for a random period of time, then retransmit the frame. The efficiency of pure ALOHA is $\frac{1}{2e}$ i.e. 18 %.

(ii) Slotted ALOHA

Slotted ALOHA is an improvement over Pure ALOHA. It decreases the number of collisions and thus the efficiency is almost doubled compared to Pure ALOHA. This approach require to users to agree of slot boundaries, which means that divide the time into discrete intervals, each interval corresponding to one frame. A station can transmit a frame only at the start of each interval. This means that, even if the user has data ready with him, he cannot send the data, he has to wait for the arrival of next slot. Thus number of collisions reduced, or the vulnerable period is reduced. In Other words, continuous Pure ALOHA has turned onto discrete Slotted ALOHA. The efficiency of slotted aloha is $\frac{1}{e}$ i.e. 36 %, which is twice that of pure ALOHA. Refer to figure below

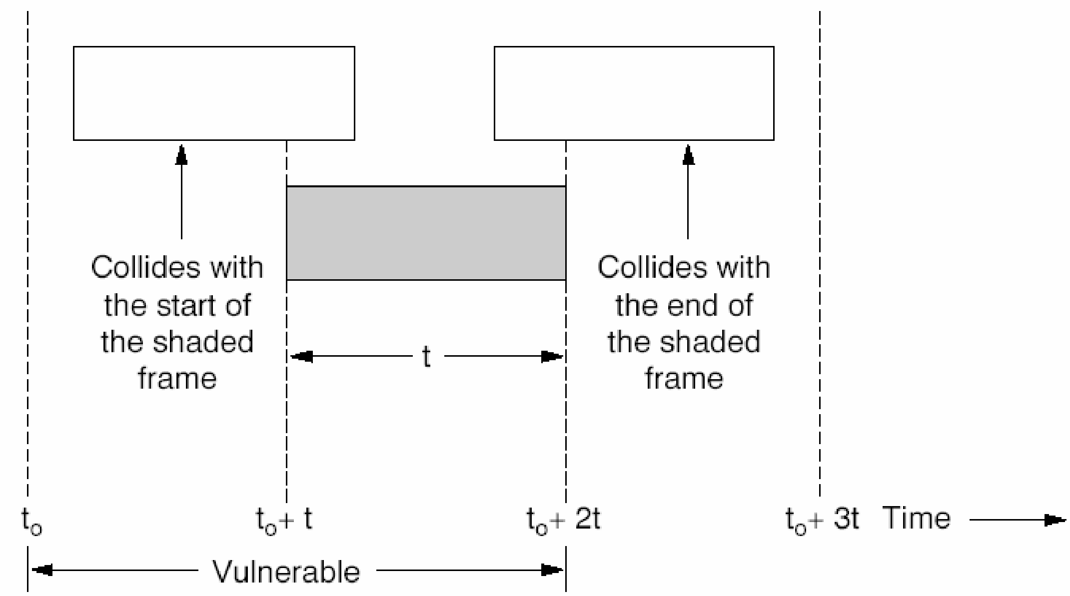
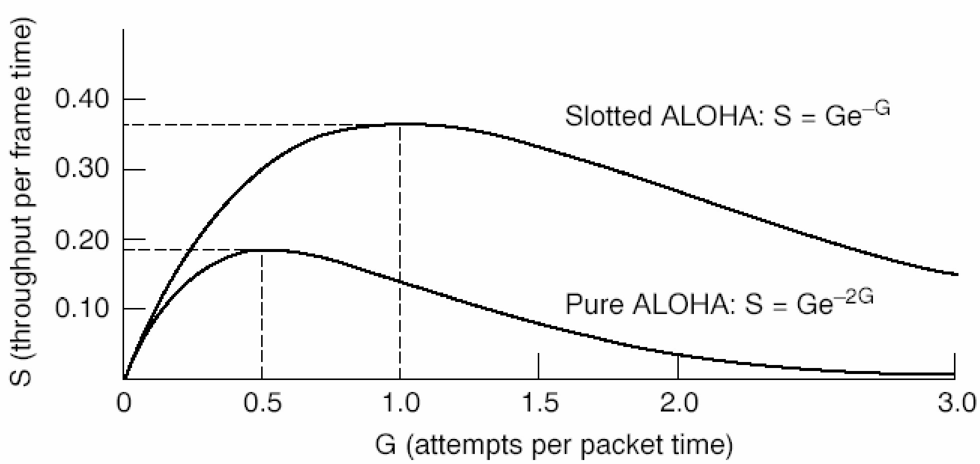


Fig. 4-2. Vulnerable period for the shaded frame.



The drawback with slotted ALOHA is, station start transmitting blindly at the start of an interval, without paying attention to what other stations are doing, this will lead to many collisions. There were chances of improvement, which are proposed in the algorithms to come.

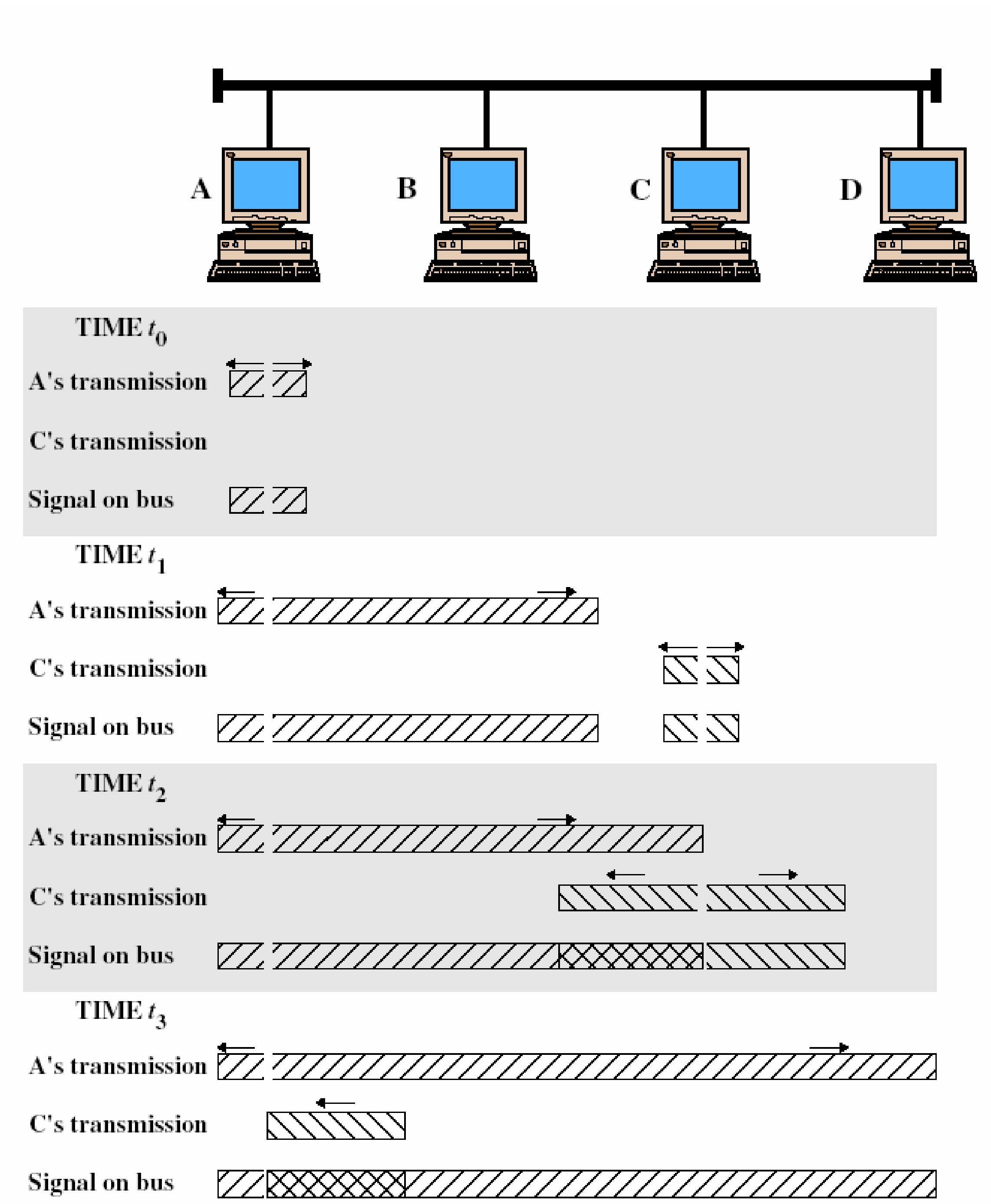
Persistent CSMA

CSMA stands for **C**arrier **S**ense **M**ultiple **A**ccess. Carrier Sense means, stations has an additional property with them, that they can sense the channel (carrier) and tell if the channel is in use or not. What we want, that at the start of the slot, stations should sense the channel first, and then act accordingly. In **persistent CSMA**, when a station has data to send, he first listens to the channel to see if the channel is busy or not or someone else is transmitting at that moment. If the channel is free, station transmits, otherwise, the station continuously sense the channel, until the channel is free, as soon as channel is find to be free, station start with the transmission. So here stations are taking an **impatient** approach, means they are continuously sensing the channel unless they find the channel to be free for transmission. This lead to problem, how, let us consider a scenario. If two stations become ready in the middle of some third station's transmission, both will wait politely until the transmission ends, and then both will transmit simultaneously, resulting in a collision. This protocol is anyway better than pure aloha, because stations have certain decency, that they first sense the channel, on finding it busy wait otherwise transmit. So persistent CSMA has better performance than pure ALOHA. But still there exist some drawbacks, which we discussed above.

Non Persistent CSMA

Non Persistent CSMA is an improvement over CSMA. In this protocol, an attempt has been made to make the stations more patient. Before transmitting, station sense the channel as always, if the channel is free, goes on with the transmission, but if the channel is busy, station moves back for a random period of time, come back again to sense the channel, if the channel is free, go on with the transmission otherwise, again move back and so on. This algorithm leads to a better utilization of channel compared to persistent CSMA.

How Frames Collide



The above figure illustrates the scenario of transmission for a base-band bus.

- At time t_0, station A begins transmitting a packet addressed to D.
- At time t_1, both B and C are ready to transmit, but B senses a transmission but C find the medium idle, as he is nearer to A (or you can say, the packet transmitted from A has reached till B but not till C). So B defers and C starts with its transmission

- At time t_2, station C starts with its transmission, and also by time t_2, A's transmission has reached station C, and a collision occurs.
- Station C detects the collision, and ceases transmission. The effect of collision propagates back to A, which will take some time.
- At time t_3, station A detects the collision, and he also defers the collision.

One can see how this strategy would be effective for networks in which the average **frame transmission time is much longer then the propagation time**. Collisions can occur only when more than one user begins transmitting with a short time (the period of propagation delay). If a station begins to transmit a frame and there are no collisions during the time it takes for the leading edge of the packet to propagate to the farthest station, then there will be no collision for that frame, because all other stations are now aware of the transmission.

The maximum utilization depends upon the length of frame and on the propagation time; the longer the frame or shorter the propagation time, the higher the utilization.

CSMA/CD

CSMA/CD is an abbreviation for Carrier Sense Multiple Access with Collision Detection. Persistent CSMA and non persistent CSMA are definitely an improvement over ALOHA but we can have another improvement, by ensuring that stations abort their transmission as soon as they detect the collision. In the cases, which we discussed earlier, stations used to get feedback, whether there is a collision or not. But now stations are improved, they can themselves detect the collision. So after a station detects the collision, it either aborts its transmission, or if intent to transmit, waits for a random period of time and then tries again. CSMA/CD is an important protocol which will find its application in 802.3 (discussed later). In CSMA/CD if a collision occurs, it is resolved through **binary exponential back off algorithm.** According to this algorithm, if a collision occurs then stations backs off and then waits for a random period of time, before transmitting again. How much is this random period of time, is decided by binary exponential back off algorithm. Time is divided into discrete slots; length of each slot is equal to worst case round trip propagation time. After the first collision, each stations waits either 0 or 1 slot times before trying again. If say, two stations collide, and each one picks up the same number, they will collide again. After second collision, each one picks up either 0,1,2,3. If a third collision occurs, then the next time the number of slots to wait is chosen from 0 to 2^3-1. In general, after all collisions, it will be 0 to 2^i-1. However after ten collisions have been reached, the randomization interval is freezed at a maximum of 0 to 1023 slots. After 16 collisions, reports the failure back to computer, and the transmission restarts as a whole.

Bit-Map Protocol

This is a collision free protocol, means the probability of collision is very less. N contention slots are defined, if there are N stations in the network. If station 0 has frame to send, it transmits a 1 bit during the zeroth slot otherwise 0. If station 1 has a frame to send it transmits a 1 bit during the first slot, similarly if station N has frame to send, it transmits a 1 bit during the last slot. After all N stations have passed by, each station has complete knowledge of which stations wish to transmit, **and at this point they start transmitting in numerical order**. Since everyone agrees on who goes next, there will never be any collision. The drawback with this method is it incurs some overhead, of transmitting a bit-map of N slots, which actually reserves slot for their transmission.

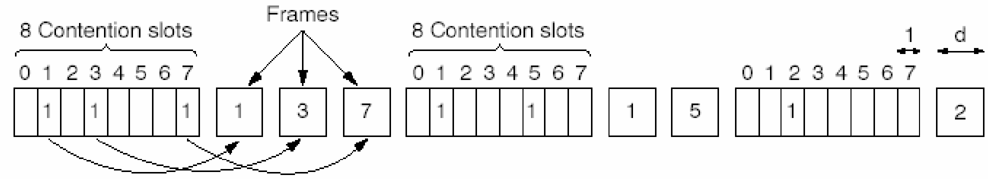


Fig. 4-6. The basic bit-map protocol.

Binary Countdown

This is another collision free protocol, but this has also some overhead. All stations wanting to use the channel now broadcast their addresses as a binary bit string. The bits in each address position are binary ORed together. For example: if stations with addresses **0**010, **0**100, **1**001, and **1**010 are all trying to get the channel, the first bit that they transmit is 0, 0, 1, 1 respectively. These are ORed together to form a 1. Stations 0010 and 0100 see the 1 as result, backs off. The result as a 1 means that some higher number station is competing for accessing the channel, so the stations with first bit as 0 backs off automatically. Now only two stations with addresses 1**0**01 and 1**0**10 are left, their second bit in address is ORed, i.e. 0 and 0, the result is 0, so no one backs off. Then their third bit in address is ORed, i.e. 0 and 1, the result is 1, so the one with third bit as 0 backs off, while the other one with third bit as 1 stays, and he is the ultimate chosen one, who has gain access to the channel.

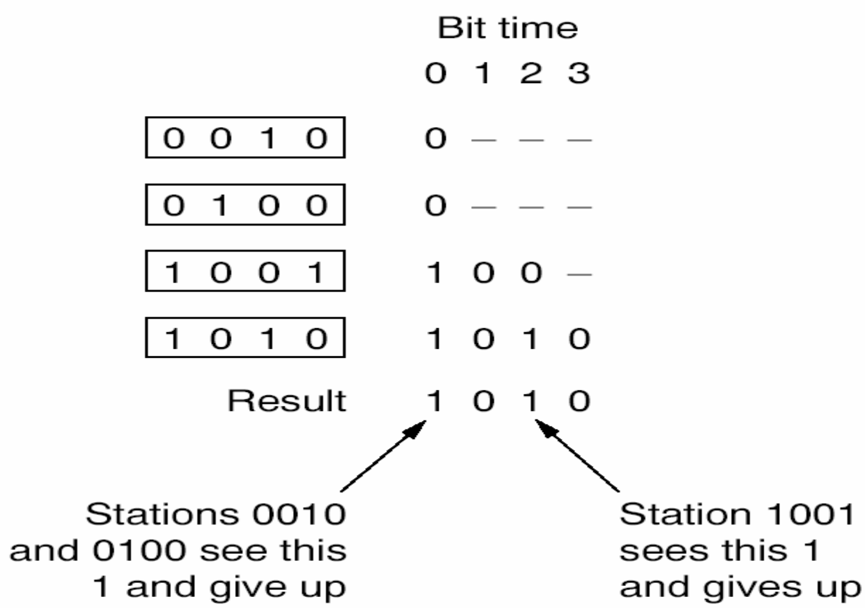


Fig. 4-7. The binary countdown protocol. A dash indicates silence.