



Week # 5 – Lecture # 9 – Data link layer (Layer 2)

Lecture outline

- Multiple Access Protocols
 - Carrier Sense Multiple Access (CSMA)
 - Carrier Sense Multiple Access with Collision Detection (CSMA/CD)
- CSMA/CD Collision Recovery Mechanism (Binary Exponential Backoff)

Multiple Access Protocols:

The problem of multiple access:

There are two types of network links: point-to-point links and broadcast links. A point-to-point link consists of a single sender at one end of the link and a single receiver at the other end of the link. Many link-layer protocols have been designed for point-to-point links; the point-to-point protocol (PPP) and high-level data link control (HDLC) are two examples of such protocols. The second type of link, a broadcast link, can have multiple sending and receiving nodes all connected to the same, single, shared broadcast channel. The term broadcast is used here because when any one node transmits a frame, the channel broadcasts the frame and each of the other nodes receives a copy. Ethernet and wireless LANs are examples of broadcast link-layer technologies.

In a single shared broadcast channel, two or more simultaneous transmissions by nodes results in interference/collision of data (i.e., data loss). How to coordinate the access of multiple sending and receiving nodes to a shared broadcast channel—the multiple access problem.

In this lecture we shall explore different solutions to the problem of multiple access.

Here are couple of definitions which are important to remember before going into details:

Collision:

It means a node receives two or more signals at the same time.



Multiple access protocol:

Distributed algorithm that determines how nodes share channel, i.e., determine when node can transmit.

An ideal multiple access protocol:

Ideally, a multiple access protocol for a broadcast channel of rate R bits per second should have the following desirable characteristics:

1. When only one node has data to send, that node has a throughput of R bps.
2. When M nodes have data to send, each of these nodes has a throughput of R/M bps. This need not necessarily imply that each of the M nodes always has an instantaneous rate of R/M , but rather that each node should have an average transmission rate of R/M over some suitably defined interval of time.
3. The protocol is decentralized; that is, there is no master node that represents a single point of failure for the network.
4. The protocol is simple, so that it is inexpensive to implement.

However, it is an ideal case and practical implementations have some differences as we shall explore.

Over the years, dozens of multiple access protocols have been implemented in a variety of link-layer technologies. Nevertheless, we can classify just about any multiple access protocol as belonging to one of three categories:

- Channel partitioning protocols
- Random access protocols
- Taking-turns protocols

We'll cover these categories of multiple access protocols

Channel Portioning Protocols:

This scheme proposes to divide channel into smaller "pieces" (time slots, frequency) and allocate a piece to node for exclusive use. In this way collision can be avoided. There are two common techniques which can be used to partition a broadcast channel's bandwidth among all nodes sharing that channel. The first one is called the

time division multiple access (TDMA) whereas the second is called frequency division multiple access (FDMA).

Time Division Multiple Access - TDMA

Suppose the channel supports N nodes and that the transmission rate of the channel is R bps. TDMA divides time into time frames and further divides each time frame into N time slots. (The TDMA time frame should not be confused with the link-layer unit of data exchanged between sending and receiving adapters, which is also called a frame). In order to reduce confusion, we'll refer to the link-layer unit of data exchanged as a packet.) Each time slot is then assigned to one of the N nodes. Whenever a node has a packet to send, it transmits the packet's bits during its assigned time slot in the revolving TDM frame. Typically, slot sizes are chosen so that a single packet can be transmitted during a slot time. Figure 5.9 in book shows a simple four-node TDM example.

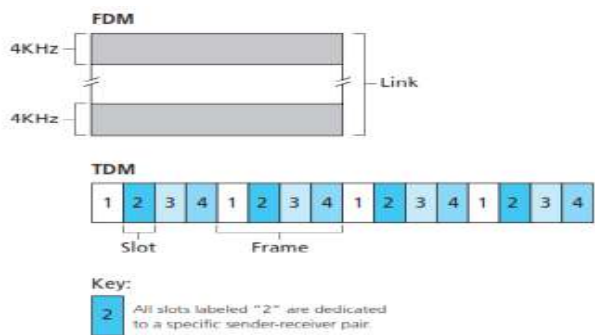
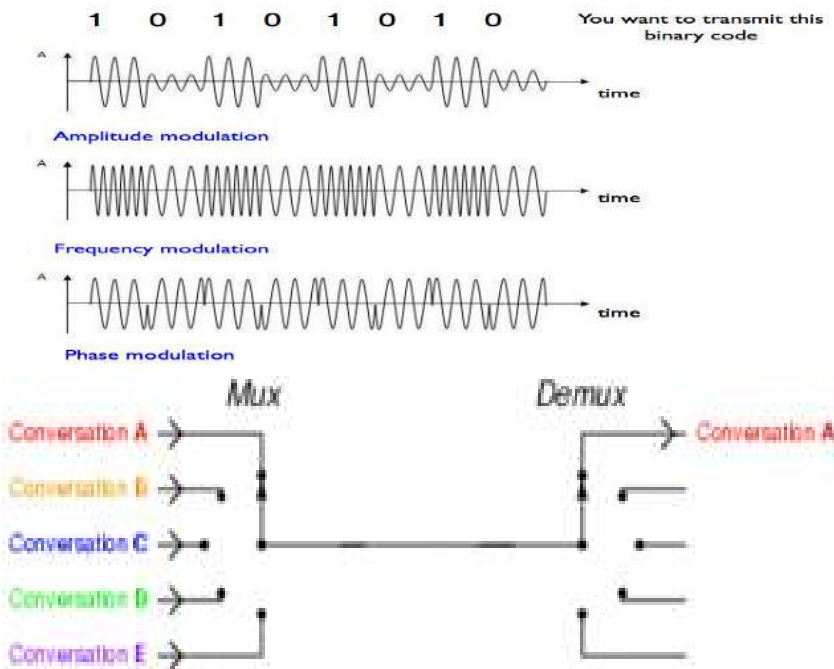
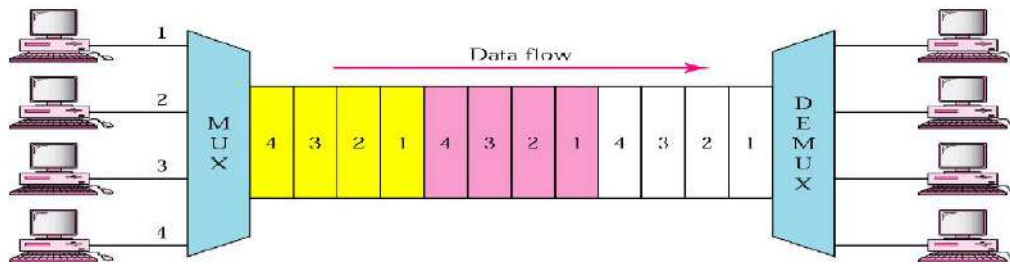


Figure 5.9 • A four-node TDM and FDM example





Here is the summary of TDMA functionality:

- Access to channel in "rounds"
- Each node gets fixed length time slot in each round
- Unused slots go idle. For instance, in 6-nodes LAN, let us say only nodes 1,3,4 have packets in a round, slots 2,5,6 will go idle

Pros and Cons of TDMA:

TDMA is appealing because it eliminates collisions and is perfectly fair: Each node gets a dedicated transmission rate of R/N bps during each frame time. However, it has two major drawbacks. First, a node is limited to an average rate of R/N bps even when it is the only node with packets to send. A second drawback is that a node must always wait for its turn in the transmission sequence—again, even when it is the only node with a frame to send.

FDMA:

While TDMA shares the broadcast channel in time, FDMA divides the R bps channel into different frequencies (each with a bandwidth of R/N) and assigns each frequency to one of the N nodes. FDMA thus creates N smaller channels of R/N bps out of the single, larger R bps channel.

Here is the summary of FDMA functionality:

- channel spectrum divided into frequency bands
- each node assigned fixed frequency band
- unused transmission time in frequency bands go idle
- example: 6-node LAN, only nodes 1,3,4 have packets, frequency bands 2,5,6 will go idle

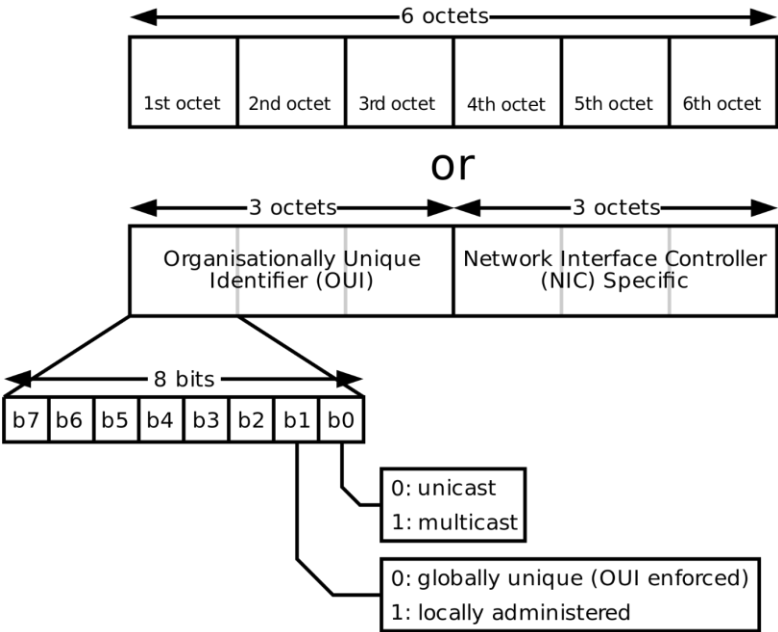


Pros and Cons of FDMA:

FDMA shares both the advantages and drawbacks of TDMA. It avoids collisions and divides the bandwidth fairly among the N nodes. However, FDMA also shares a principal disadvantage with TDMA—a node is limited to a bandwidth of R/N , even when it is the only node with packets to send.

In this lecture we covered channel portioning schemes. We shall continue our discussion on multiple access in next lecture. We shall look into other protocols which belong to random access and taking turn categorizes.

MAC Address: 48 bits



Port#	1	2		24
MAC#	PC4-MAC	PC2-MAC		PC1-MAC

L2 -- FRAME

Source MAC	Destination MAC	DATA	FCS
MAC1	MAC3		2



We looked into details of **channel partitioning protocols**. We discussed advantages and disadvantages of **TDMA** and **FDMA** protocols. We found that fundamental drawback with these approaches is limited share of bandwidth, even when there is only one node using a link. In today's lecture we shall continue our discussion and explore more solutions to address the problem of multiple access. In this regard, the next category of solutions is known as random access protocols.

Week # 5 – Lecture # 10 – Data link layer (Layer 2)

Multiple Access Protocols:

Random Access Protocols (random wait time to retransmit):

In a random access protocol, a transmitting node always **transmits at the full rate of the channel**, namely, R bps. There is no prior coordination between sending nodes in this case. It means **collision happens when two or more nodes transmit at the same time**. When there is a collision, each node involved in the collision repeatedly retransmits its frame (that is, packet) until its frame gets through without a collision. But when a node experiences a collision, it doesn't necessarily retransmit the frame right away. Instead, it waits a random delay (i.e., wait time) before retransmitting the frame. Each node involved in a collision chooses independent random delays. Because the random delays are independently chosen, it is possible that one of the nodes will pick a delay that is sufficiently less than the delays of the other colliding nodes and will therefore be able to transmit its frame into the channel without a collision.

The aim of random access protocols is **to achieve efficient link utilization** (high bandwidth usage of a link). But in this process collision might occur. So, a random access protocol must have a mechanism to detect the collision. Once the collision is detected, the next task is to **recover from the collision, which means retransmitting a frame without a collision** (e.g., via delayed retransmissions as discussed above).

There are many different implementations of random access protocols. Some common and the most popular examples of random access protocols are given below:



- **Carrier Sense Multiple Access (CSMA)**
- **Carrier Sense Multiple Access with Collision Detection (CSMA/CD)**

Carrier Sense Multiple Access (CSMA):

In both slotted and pure ALOHA, a node's decision to transmit is made independently of the activity of the other nodes attached to the broadcast channel. In particular, a node neither pays attention to whether another node happens to be transmitting when it begins to transmit, nor stops transmitting if another node begins to interfere with its transmission. CSMA makes the difference by addressing these aspects in order to avoid collision. To clarify this concept of communication, we take an example from polite human conversation. There are two important rules for polite human conversation:

1. Listen before speaking

If someone else is speaking, wait until they are finished. In the networking world, this is called **carrier sensing**—a node listens to the channel before transmitting. If a frame from another node is currently being transmitted into the channel, a node then waits until it detects no transmissions for a short amount of time and then begins transmission.

2. If someone else begins talking at the same time, stop talking

In the networking world, this is called **collision detection**—a transmitting node listens to the channel while it is transmitting. If it detects that another node is transmitting an interfering frame, it stops transmitting and waits a random amount of time before retransmission.

In short, CSMA sense the channel before transmitting, if channel sensed idle the frame is transmitted. Otherwise (channel sensed busy, some other node is sending frame), transmission is delayed for some time until an idle channel is available for transmission. Does this mechanism ensure that collision will never happen? The answer is NO (unfortunately). The reason is described in the next paragraph.

Collision Scenario:

Suppose two nodes A and B are connected to a shared broadcast media (e.g., Ethernet bus topology as shown in figure below). In this case all hosts are sharing the

same physical media or channel. Suppose both nodes A and B have frames to transmit and sense the channel idle at the same time and start transmission. It will result in collision indeed since the channel is shared. So, in addition to carrier sensing, we need a mechanism to detect collision and then recover from collision. Recovery is a mechanism which leads to the successful retransmission of a collided frame.



Carrier Sense Multiple Access with Collision Detection (CSMA/CD):

The operation of CSMA/CD can be summarized as follows:

1. The adapter (Network Interface Card) obtains a datagram from the network layer, prepares a link-layer frame, and puts the frame adapter buffer.
2. If the adapter senses that the channel is idle (that is, there is no signal energy entering the adapter from the channel), it starts to transmit the frame. If, on the other hand, the adapter senses that the channel is busy, it waits until it senses no signal energy and then starts to transmit the frame.
3. While transmitting, the adapter monitors for the presence of signal energy coming from other adapters using the broadcast channel.
4. If the adapter transmits the entire frame without detecting signal energy from other adapters, the adapter is finished with the frame. If, on the other hand, the adapter detects signal energy from other adapters while transmitting, it aborts the transmission (that is, it stops transmitting its frame).
5. After aborting, the adapter waits a random amount of time and then returns to step 2.

We explained how collision is detected in CSMA/CD. In next lecture, we continue this topic and explain collision recovery mechanism of CSMA/CD. The recovery mechanism is known as binary exponential backoff.



CSMA/CD Collision Recovery Mechanism (Binary Exponential Backoff)

We have seen in CSMA/CD that transmission is aborted for a random amount of time once collision is detected by a node (Lecture 9 notes, step 5 of CSMA/CD). This time is known as backoff time. The need to wait a random (rather than fixed) amount of time is important. If two nodes transmitted frames at the same time and then both waited the same fixed amount of time, they'd continue colliding forever.

The time interval (backoff time) is random for every node, but then next question is, what is a good interval of time from which to choose the random backoff time? If the interval is large and the number of colliding nodes is small, nodes are likely to wait a large amount of time (with the channel remaining idle) before repeating the sense-and-transmit when-idle channel. On the other hand, if the interval is small and the number of colliding nodes is large, it's likely that the chosen random values will be nearly the same, and transmitting nodes will again collide. What we'd like is an interval that is short when the number of colliding nodes is small, and long when the number of colliding nodes is large.

The **binary exponential backoff algorithm** elegantly solves this problem. Specifically, when transmitting a frame that has already experienced n collisions, a node chooses the value of K at random from $\{0, 1, 2, \dots, 2^n - 1\}$. Thus, **the more collisions experienced by a frame, the larger the interval from which K is chosen.**

Ethernet uses CSMA/CD to fix the problem of multiple access. The backoff time is the actual amount of time a node waits after collision detection. In case of Ethernet, backoff time is $K \times 512$ bit times. It means K times the amount of time needed to send 512 bits on to wire. The maximum value that n can take is limited to 10 (i.e., upper limit for value of n).

Pros and cons of random access protocols:

- efficient at low load: single node can fully utilize channel
- high load: collision overhead



Taking Turn Protocols:

The next and last category of multiple access protocols is known as taking turn protocols.

Until now, we have seen channel partitioning and random access protocols to fix the problem of multiple access. Channel partitioning protocols (TDMA, FDMA) are efficient at high traffic load but inefficient at low load. In contrast, random access protocols are efficient at low load but inefficient at high load due to collision overhead. Taking turn protocols (next category which we are going to study) look for the best from both categories of protocols, which means efficient bandwidth utilization for both high and low traffic load.

There are two most common and popular protocols which belong to this category (i.e. taking turn protocols):

- **Polling protocol**
- **Token passing protocol**

Polling Protocol:

The polling protocol requires one of the nodes to be designated as a master node. The master node polls each of the nodes in a round-robin fashion. In particular, the master node first sends a message to node 1, saying that it (node 1) can transmit up to some maximum number of frames. After node 1 transmits some frames, the master node tells node 2 it (node 2) can transmit up to the maximum number of frames. The master node can determine when a node has finished sending its frames by observing the lack of a signal on the channel. The procedure continues in this manner, with the master node polling each of the nodes in a cyclic manner. If a node has nothing to transmit (i.e., no frame), next node is offered for transmission without waiting. In this way, the polling protocol eliminates both collisions and empty slots. This allows polling to achieve a much higher efficiency. The **Bluetooth technology** uses polling protocol.

Drawbacks of polling protocol:

The first drawback is that the protocol introduces a **polling delay**—the amount of time required to notify a node that it can transmit. If, for example, only one node is active, then the node will transmit at a rate less than R bps, as the master node must poll



each of the inactive nodes in turn each time the active node has sent its maximum number of frames.

The second drawback, which is potentially more serious, is that **if the master node fails, the entire channel becomes inoperative.**

polling:

- ❖ master node “invites” slave nodes to transmit in turn
- ❖ typically used with “dumb” slave devices



Token Passing Protocol:

The second taking-turns protocol is the token-passing protocol. In this protocol there is no master node. **A small, special-purpose frame known as a token is exchanged among the nodes in some fixed order.** For example, node 1 might always send the token to node 2, node 2 might always send the token to node 3, and node N might always send the token to node 1. When a node receives a token, it holds onto the token only if it has some frames to transmit; otherwise, it immediately forwards the token to the next node. If a node does have frames to transmit when it receives the token, it sends up to a maximum number of frames and then forwards the token to the next node. Token passing is decentralized and highly efficient.

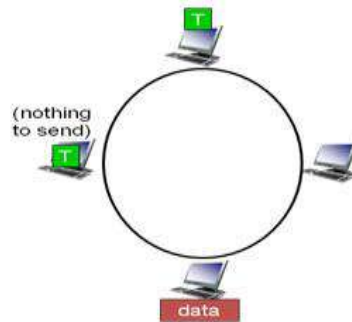


Computer Networks (CS-577)



token passing:

- ❖ control *token* passed from one node to next sequentially.



Drawback of token passing protocol:

The failure of one node can crash the entire channel. Or if a node accidentally neglects to release the token, then some recovery procedure must be invoked to get the token back in circulation. Over the years many token-passing protocols have been developed, including the **Fiber Distributed Data Interface (FDDI) protocol** and the **IEEE 802.5 token ring protocol**, and each one had to address these issues.

In this lecture we have completed our discussion on multiple access protocols. We covered three different categories of multiple access protocols which include channel portioning protocols, random access protocols and taking turn protocols.