Unit- 4

The Medium Access Sub-layer

Content:

1. The channel allocation problems
2. Multiple access protocols
3. Ethernet
4. Data link layer switching

I. Introduction-

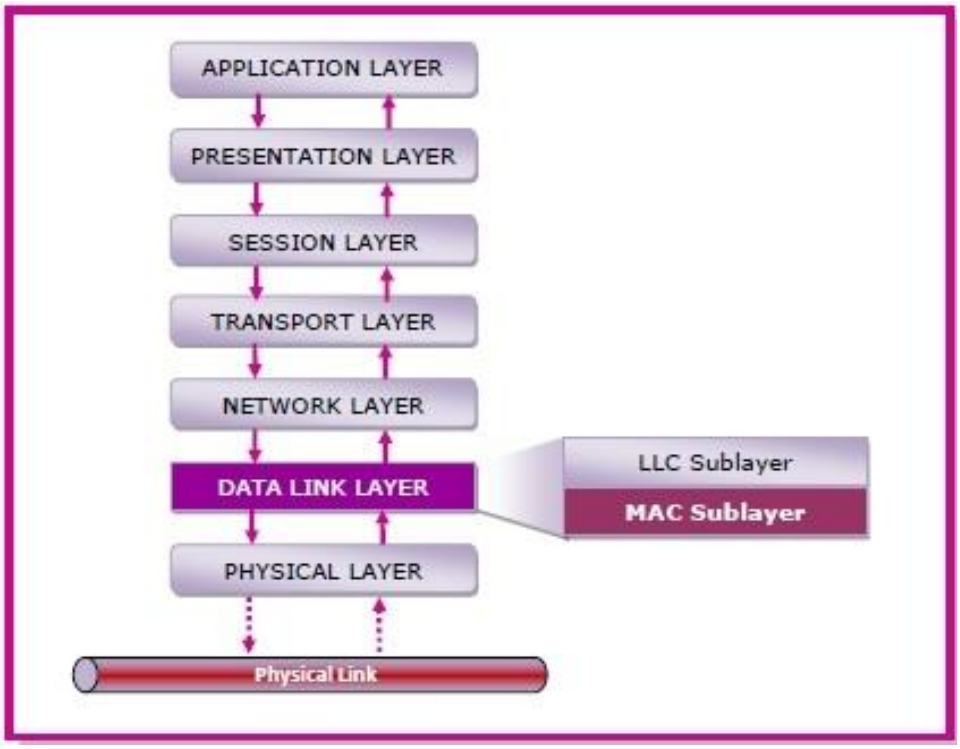
The medium access control (MAC) is a sublayer of the data link layer of the open system interconnections (OSI) reference model for data transmission. It is responsible for flow control and multiplexing for transmission medium. It controls the transmission of data packets via remotely shared channels. It sends data over the network interface card.

MAC Layer in the OSI Model:

The Open System Interconnections (OSI) model is a layered networking framework that conceptualizes how communications should be done between heterogeneous systems. The data link layer is the second lowest layer. It is divided into two sublayers –

1. The logical link control (LLC) sublayer
2. The medium access control (MAC) sublayer

The following diagram depicts the position of the MAC layer –



Functions of MAC Layer:

- It provides an abstraction of the physical layer to the LLC and upper layers of the OSI network.
- It is responsible for encapsulating frames so that they are suitable for transmission via the physical medium.
- It resolves the addressing of source station as well as the destination station, or groups of destination stations.
- It performs multiple access resolutions when more than one data frame is to be transmitted. It determines the channel access methods for transmission.
- It also performs collision resolution and initiating retransmission in case of collisions.
- It generates the frame check sequences and thus contributes to protection against transmission errors.

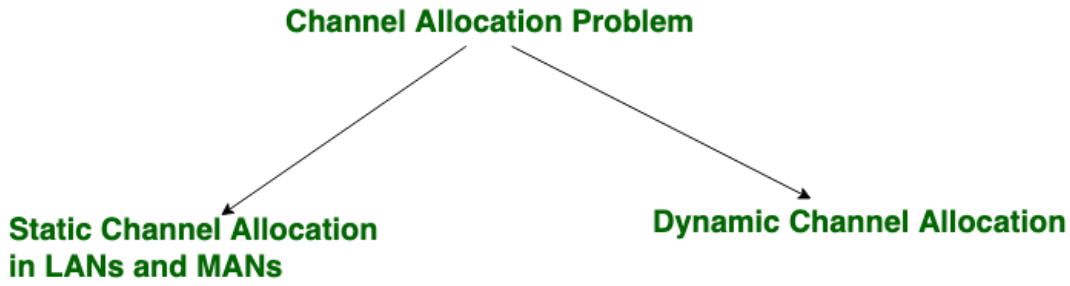
MAC Addresses:

MAC address or media access control address is a unique identifier allotted to a network interface controller (NIC) of a device. It is used as a network address for data transmission within a network segment like Ethernet, Wi-Fi, and Bluetooth.

MAC address is assigned to a network adapter at the time of manufacturing. It is hardwired or hard-coded in the network interface card (NIC). A MAC address comprises of six groups of two hexadecimal digits, separated by hyphens, colons, or no separators. An example of a MAC address is 00:0A:89:5B:F0:11.

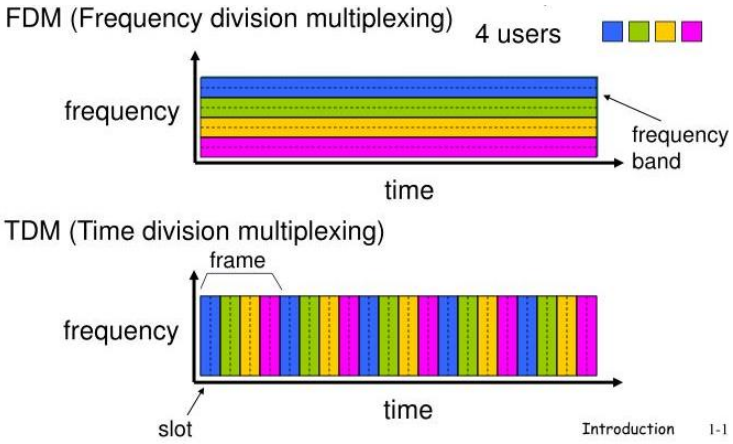
4.1 The Channel Allocation Problem:

Channel allocation is a process in which a single channel is divided and allotted to multiple users in order to carry user specific tasks. There is user's quantity may vary every time the process takes place. The central theme of this chapter is how to allocate a single broadcast channel among competing users.



4.1.1 Static Channel Allocation in LANs and MANs:

The traditional way of allocating a single channel, such as a telephone trunk, among multiple competing users is Frequency Division Multiplexing (FDM).



If there are *N* users, the bandwidth is divided into *N* equal-sized portions, each user being assigned one portion. Since each user has a private frequency band, there is no interference between users.

When there is only a small and constant number of users, each of which has a heavy (buffered) load of traffic (e.g., carriers' switching offices), FDM is a simple and efficient allocation mechanism.

However, when the number of senders is large and continuously varying or the traffic is bursty, FDM presents some problems.

If the spectrum is cut up into *N* regions and fewer than *N* users are currently interested in communicating, a large piece of valuable spectrum will be wasted.

If more than *N* users want to communicate, some of them will be denied permission for lack of bandwidth, even if some of the users who have been assigned a frequency band hardly ever transmit or receive anything.

The poor performance of static FDM can easily be seen from a simple queueing theory calculation.

Let us start with the mean time delay, *T*, for a channel of capacity *C* bps, with an arrival rate of λ frames/sec, each frame having a length drawn from an exponential probability density function with mean 1/μ bits/frame.

With these parameters the arrival rate is λ frames/sec and the service rate is μ*C* frames/sec.

From queueing theory it can be shown that for Poisson arrival and service times,

$$T = \frac{1}{\mu C - \lambda}$$

Now let us divide the single channel into *N* independent subchannels, each with capacity *C/N* bps. The mean input rate on each of the subchannels will now be λ*/N.* Recomputing *T* we get,

$$T_{\text{FDM}} = \frac{1}{\mu(C/N) - (\lambda/N)} = \frac{N}{\mu C - \lambda} = NT$$

4.1.2 Dynamic Channel Allocation in LANs and MANs:

Underlying all the work done in this area are five key assumptions, described below.

Station Model:

The model consists of *N* independent stations (e.g., computers, telephones, or personal communicators), each with a program or user that generates frames for transmission.

Stations are sometimes called terminals. The probability of a frame being generated in an interval of length Δ*t* is λΔ*t*, where λ is a constant (the arrival rate of new frames).

Once a frame has been generated, the station is blocked and does nothing until the frame has been successfully transmitted.

Single Channel Assumption:

A single channel is available for all communication. All stations can transmit on it and all can receive from it. As far as the hardware is concerned, all stations are equivalent, although protocol software may assign priorities to them.

Collision Assumption:

If two frames are transmitted simultaneously, they overlap in time and the resulting signal is garbled. This event is called a collision. All stations can detect collisions. A collided frame must be transmitted again later. There are no errors other than those generated by collisions.

4a. Continuous Time. Frame transmission can begin at any instant. There is no master clock dividing time into discrete intervals.

4b. Slotted Time. Time is divided into discrete intervals (slots). Frame transmissions always begin at the start of a slot. A slot may contain 0, 1, or more frames, corresponding to an idle slot, a successful transmission, or a collision, respectively.

5a. Carrier Sense. Stations can tell if the channel is in use before trying to use it. If the channel is sensed as busy, no station will attempt to use it until it goes idle.

5b. No Carrier Sense. Stations cannot sense the channel before trying to use it. They just go ahead and transmit. Only later can they determine whether the transmission was successful.

Random Access Protocol: In this, all stations have same superiority that is no station has more priority than another station. Any station can send data depending on medium's state (idle or busy). It has two features:

1. There is no fixed time for sending data
2. There is no fixed sequence of stations sending data

The Random-access protocols are further subdivided as:

(a) ALOHA – It was designed for wireless LAN but is also applicable for shared medium. In this, multiple stations can transmit data at the same time and can hence lead to collision and data being garbled.

- **Pure Aloha:**
  When a station sends data it waits for an acknowledgement. If the acknowledgement doesn't come within the allotted time then the station waits for a random amount of time called back-off time (Tb) and re-sends the data. Since different stations wait for different amount of time, the probability of further collision decreases.
- Vulnerable Time = 2* Frame transmission time
- Throughput = G exp{-2*G}

  Maximum throughput = 0.184 for G=0.5

- **Slotted Aloha:**
  It is similar to pure aloha, except that we divide time into slots and sending of data is allowed only at the beginning of these slots. If a station misses out the allowed time, it must wait for the next slot. This reduces the probability of collision.
- Vulnerable Time = Frame transmission time
- Throughput = G exp{-*G}

  Maximum throughput = 0.368 for G=1

  For more information on ALOHA refer – LAN Technologies

(b) CSMA – Carrier Sense Multiple Access ensures fewer collisions as the station is required to first sense the medium (for idle or busy) before transmitting data. If it is idle then it sends data, otherwise it waits till the

channel becomes idle. However there is still chance of collision in CSMA due to propagation delay. For example, if station A wants to send data, it will first sense the medium.If it finds the channel idle, it will start sending data. However, by the time the first bit of data is transmitted (delayed due to propagation delay) from station A, if station B requests to send data and senses the medium it will also find it idle and will also send data. This will result in collision of data from station A and B.

CSMA access modes-

- **1-persistent:** The node senses the channel, if idle it sends the data, otherwise it continuously keeps on checking the medium for being idle and transmits unconditionally(with 1 probability) as soon as the channel gets idle.
- **Non-Persistent:** The node senses the channel, if idle it sends the data, otherwise it checks the medium after a random amount of time (not continuously) and transmits when found idle.
- **P-persistent:** The node senses the medium, if idle it sends the data with p probability. If the data is not transmitted ((1-p) probability) then it waits for some time and checks the medium again, now if it is found idle then it send with p probability. This repeat continues until the frame is sent. It is used in Wifi and packet radio systems.
- **O-persistent:** Superiority of nodes is decided beforehand and transmission occurs in that order. If the medium is idle, node waits for its time slot to send data.

(c) CSMA/CD – Carrier sense multiple access with collision detection. Stations can terminate transmission of data if collision is detected. For more details refer – Efficiency of CSMA/CD

(d) CSMA/CA – Carrier sense multiple access with collision avoidance. The process of collisions detection involves sender receiving acknowledgement signals. If there is just one signal (its own) then the data is successfully sent but if there are two signals(its own and the one with which it has collided)

then it means a collision has occurred. To distinguish between these two cases, collision must have a lot of impact on received signal. However, it is not so in wired networks, so CSMA/CA is used in this case.

CSMA/CA avoids collision by:

1. **Interframe space –** Station waits for medium to become idle and if found idle it does not immediately send data (to avoid collision due to propagation delay) rather it waits for a period of time called Interframe space or IFS. After this time, it again checks the medium for being idle. The IFS duration depends on the priority of station.
2. **Contention Window –** It is the amount of time divided into slots. If the sender is ready to send data, it chooses a random number of slots as wait time which doubles every time medium is not found idle. If the medium is found busy it does not restart the entire process, rather it restarts the timer when the channel is found idle again.
3. **Acknowledgement –** The sender re-transmits the data if acknowledgement is not received before time-out.

4.1.3 Collision-Free Protocols

Although collisions do not occur with CSMA/CD once a station has unambiguously captured the channel, they can still occur during the contention period. These collisions adversely affect the system performance, especially when the cable is long (i.e., large τ) and the frames are short. And CSMA/CD is not universally applicable. In this section, we will examine some protocols that resolve the contention for the channel without any collisions at all, not even during the contention period.

4.1.3.1 A Bit-Map Protocol:

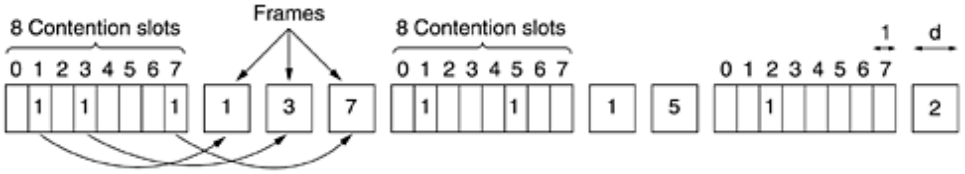
In our first collision-free protocol, the basic bit-map method, each contention period consists of exactly *N* slots. If station 0 has a frame to send,

it transmits a 1 bit during the zeroth slot. No other station is allowed to transmit during this slot. Regardless of what station 0 does,

station 1 gets the opportunity to transmit a 1 during slot 1, but only if it has a frame queued. In general, station *j* may announce that it has a frame to send by inserting a 1 bit into slot *j.* After all *N* slots have passed by, each station has complete knowledge of which stations wish to transmit. At that point, they begin transmitting in numerical order.

Since everyone agrees on who goes next, there will never be any collisions. After the last ready station has transmitted its frame, an event all stations can easily monitor, another *N* bit contention period is begun.

If a station becomes ready just after its bit slot has passed by, it is out of luck and must remain silent until every station has had a chance and the bit map has come around again. Protocols like this in which the desire to transmit is broadcast before the actual transmission are called **reservation protocols**.



4.1.3.2 Binary Countdown:

Binary Countdown Protocol is a collision-free protocol that operates in the Medium Access Control (MAC) layer of the OSI model. In computer networks, when more than one station tries to transmit simultaneously via a shared channel, the transmitted data is garbled, an event called collision. Collision-free protocols resolves channel access while the stations are contending for the shared channel, thus eliminating any possibilities of collisions.

Working Principle of Binary Countdown

In a binary countdown protocol, each station is assigned a binary address. The binary addresses are bit strings of equal lengths. When a station wants to transmit, it broadcasts its address to all the stations in the channel, one bit at a time starting with the highest order bit.

In order to decide which station gets the channel access, the addresses of the stations which are broadcasted are ORed. The higher numbered station gets the channel access.

Example

Suppose that six stations contend for channel access which have the addresses: 1011, 0010, 0101, 1100, 1001 and 1101.

The iterative steps are −

- All stations broadcast their most significant bit, i.e. 1, 0, 0, 1, 1, 1. Stations 0010 and 0101 sees 1 bit in other stations, and so they give up competing for the channel.
- The stations 1011, 1100, 1001 and 1101 continue. They broadcast their next bit, i.e. 0, 1, 0, 1. Stations 1011 and 1001 sees 1 bit in other stations, and so they give up competing for the channel.
- The stations 1100 and 1101 continue. They broadcast their next bit, i.e. 0, 0. Since both of them have same bit value, both of them broadcast their next bit.
- The stations 1100 and 1101 broadcast their least significant bit, i.e. 0 and 1. Since station 1101 has 1 while the other 0, station 1101 gets the access to the channel.
- After station 1101 has completed frame transmission, or there is a time-out, the next contention cycle starts.

Station Address	Bit Time 0	1	2	3	Station Status
1011	1	0	-	-	Gives up after bit time 1
0010	0	-	-	-	Gives up after bit time 0
0101	0	-	-	-	Gives up after bit time 0
1100	1	1	0	0	Gives up after bit time 3
1001	1	0	-	-	Gives up after bit time 0
1101	1	1	0	1	Gets channel access

4.1.3.3 Limited Contention Protocols:

- Collision based protocols (pure and slotted ALOHA, CSMA/CD) are good when the network load is low.
- Collision free protocols (bitmap, binary Countdown) are good when load is high.
- How about combining their advantages?

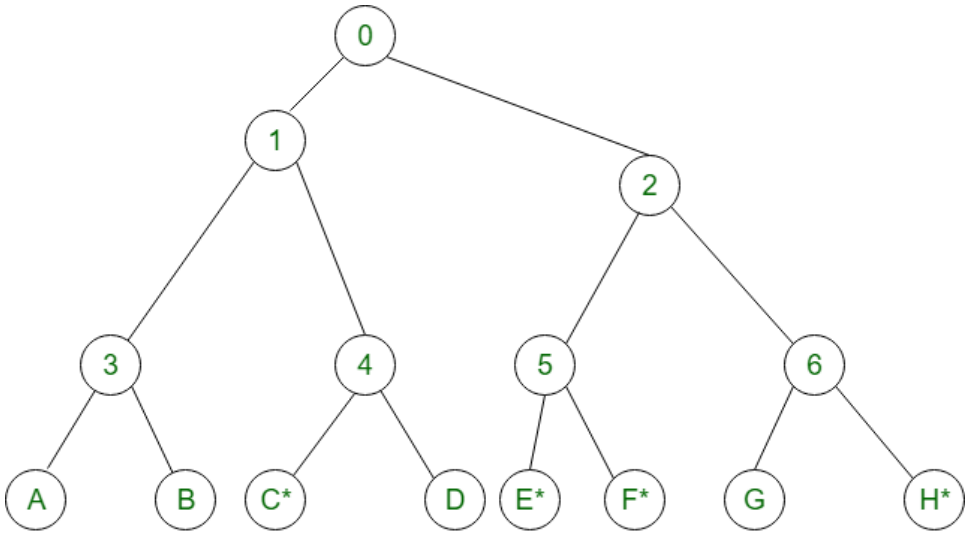
1. Behave like the ALOHA scheme under light load
2. Behave like the bitmap scheme under heavy load.

4.1.3.3.1 Adaptive Tree Walk Protocol:

- partition the group of station and limit the contention for each slot.
- Under light load, everyone can try for each slot like aloha
- Under heavy load, only a group can try for each slot
- **How do we do it:**

1. Treat every station as the leaf of a binary tree
2. First slot (after successful transmission), all stations can try to get the slot (under the root node).
3. If no conflict, fine
4. n case of conflict, only nodes under a subtree get to try for the next one. (depth first search)

For Example:

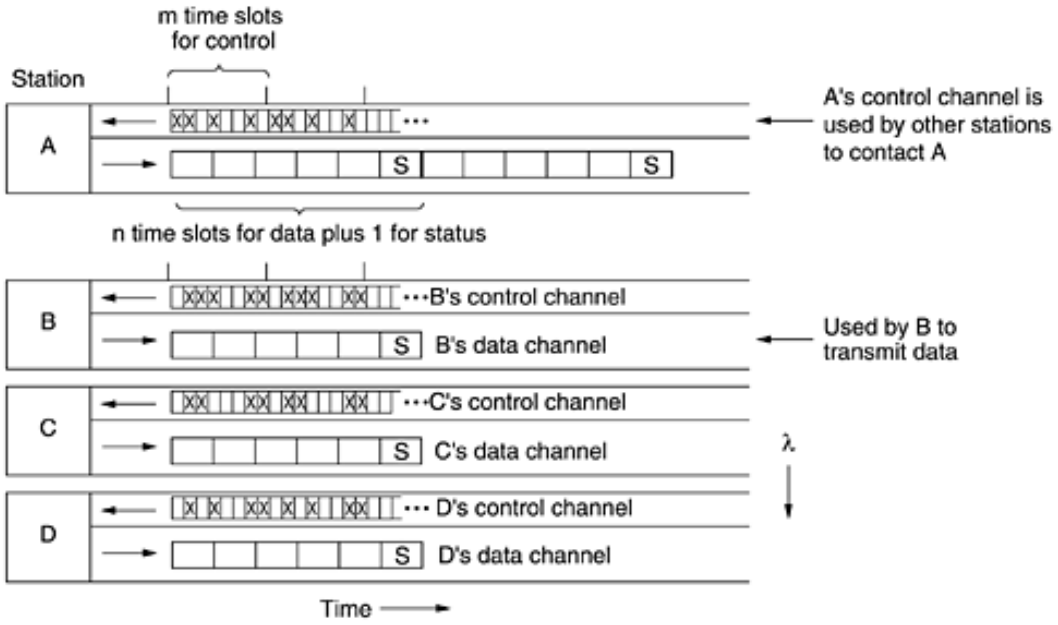


- **Slot-0:** C*, E*, F*, H* (all nodes under node 0 can try which are going to send), conflict
- **Slot-1:** C* (all nodes under node 1can try}, C sends
- **Slot-2:** E*, F*, H*(all nodes under node 2 can try}, conflict
- **Slot-3:** E*, F* (all nodes under node 5 can try to send), conflict
- **Slot-4:** E* (all nodes under E can try), E sends
- **Slot-5:** F* (all nodes under F can try), F sends
- **Slot-6:** H* (all nodes under node 6 can try to send), H sends.

4.1.3.3 Wavelength Division Multiple Access Protocols:

A different approach to channel allocation is to divide the channel into subchannels using FDM, TDM, or both, and dynamically allocate them as needed. Schemes like this are commonly used on fiber optic LANs to permit different conversations to use different wavelengths (i.e., frequencies) at the same time.

In this protocol, **WDMA** (**Wavelength Division Multiple Access**), each station is assigned two channels. A narrow channel is provided as a control channel to signal the station, and a wide channel is provided so the station can output data frames.



The protocol supports three traffic classes: (1) constant data rate connection-oriented traffic, such as uncompressed video, (2) variable data rate connection-oriented traffic, such as file transfer, and (3) datagram traffic, such as UDP packets. For the two connection-oriented protocols, the idea is that for *A* to communicate with *B*, it must first insert a CONNECTION REQUEST frame in a free slot on *B*'s control channel. If *B* accepts, communication can take place on *A*'s data channel.

Each station has two transmitters and two receivers, as follows:

1. A fixed-wavelength receiver for listening to its own control channel.

2. A tunable transmitter for sending on other stations' control channels.

3. A fixed-wavelength transmitter for outputting data frames.

4. A tunable receiver for selecting a data transmitter to listen to.

4.1.3.4 Wireless LAN Protocols:

In our discussions below, we will make the simplifying assumption that all radio transmitters have some fixed range. When a receiver is within range of two active transmitters, the resulting signal will generally be garbled and useless, in other words, we will not consider CDMA-type systems further in this discussion. It is important to realize that in some wireless LANs, not all stations are within range of one another, which leads to a variety of complications. Furthermore, for indoor wireless LANs, the presence of walls between stations can have a major impact on the effective range of each station.

A naive approach to using a wireless LAN might be to try CSMA: just listen for other transmissions and only transmit if no one else is doing so. The trouble is, this protocol is not really appropriate because what matters is interference at the receiver, not at the sender. The radio range is such that A

and B are within each other's range and can potentially interfere with one another. C can also potentially interfere with both B and D, but not with A.

First consider what happens when A is transmitting to B, as depicted in Fig. 4-11(a). If C senses the medium, it will not hear A because A is out of range, and thus falsely conclude that it can transmit to B. If C does start transmitting, it will interfere at B, wiping out the frame from A. The problem of a station not being able to detect a potential competitor for the medium because the competitor is too far away is called the hidden station problem.

Now let us consider the reverse situation: B transmitting to A, as shown in Fig. 4-11(b). If C senses the medium, it will hear an ongoing transmission and falsely conclude that it may not send to D, when in fact such a transmission would cause bad reception only in the zone between B and C, where neither of the intended receivers is located. This is called the exposed station problem.

The problem is that before starting a transmission, a station really wants to know whether there is activity around the receiver. CSMA merely tells it whether there is activity around the station sensing the carrier. In a system based on short-range radio waves, multiple transmissions can occur simultaneously if they all have different destinations and these destinations are out of range of one another.

MACA and MACAW:

An early protocol designed for wireless LANs is **MACA (Multiple Access with Collision Avoidance)**. The basic idea behind it is for the sender to stimulate the receiver into outputting a short frame, so stations nearby can detect this

transmission and avoid transmitting for the duration of the upcoming (large) data frame.

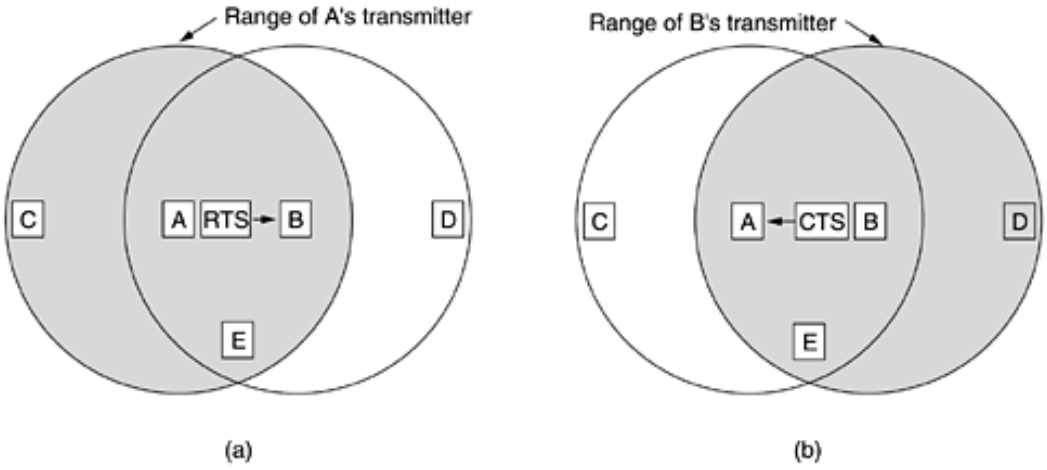


***Figure. The MACA protocol. (a)* A *sending an RTS to* B. *(b)* B *responding with a CTS to* A.**

Let us now consider how A sends a frame to B. A start by sending an RTS (Request To Send) frame to B, as shown in Fig.(a). This short frame (30 bytes) contains the length of the data frame that will eventually follow. Then B replies with a CTS (Clear to Send) frame, as shown in Fig.(b). The CTS frame contains the data length (copied from the RTS frame). Upon receipt of the CTS frame, A begins transmission.

In Fig, C is within range of A but not within range of B. Therefore, it hears the RTS from A but not the CTS from B. As long as it does not interfere with the CTS, it is free to transmit while the data frame is being sent. In contrast, D is within range of B but not A. It does not hear the RTS but does hear the CTS. Hearing the CTS tips, it off that it is close to a station that is about to receive a frame, so it defers sending anything until that frame is expected to be finished. Station E hears both control messages and, like D, must be silent until the data frame is complete.

Despite these precautions, collisions can still occur. For example, B and C could both send RTS frames to A at the same time. These will collide and be lost. In the event of a collision, an unsuccessful transmitter (i.e., one that does not hear a CTS within the expected time interval) waits a random

amount of time and tries again later. The algorithm used is binary exponential backoff, which we will study when we come to Ethernet.

Based on simulation studies of MACA, fine-tuned MACA to improve its performance and renamed their new protocol **MACAW (MACA for Wireless).** To start with, they noticed that without data link layer acknowledgements, lost frames were not retransmitted until the transport layer noticed their absence, much later.

They solved this problem by introducing an ACK frame after each successful data frame. They also observed that CSMA has some use, namely, to keep a station from transmitting an RTS at the same time another nearby station is also doing so to the same destination, so carrier sensing was added.

In addition, they decided to run the backoff algorithm separately for each data stream (source-destination pair), rather than for each station. This change improves the fairness of the protocol.

Finally, they added a mechanism for stations to exchange information about congestion and a way to make the backoff algorithm react less violently to temporary problems, to improve system performance.

4.2 Ethernet:

IEEE has standardized a number of local area networks and metropolitan area networks under the name of IEEE 802. The most important are 802.3 (Ethernet) and 802.11 (wireless LAN). We will focus on the technical details of Ethernet, the protocols, and recent developments in high-speed (gigabit) Ethernet.

1. Ethernet Cabling:

The most common kinds of Ethernet cabling are:

Name	Cable	Max. seg.	Nodes/seg.	Advantages
10Base5	Thick coax	500 m	100	Original cable; now obsolete
10Base2	Thin coax	185 m	30	No hub needed
10Base-T	Twisted pair	100 m	1024	Cheapest system
10Base-F	Fiber optics	2000 m	1024	Best between buildings

2. Manchester Encoding:

None of the versions of Ethernet uses straight binary encoding with 0 volts for a 0 bit and 5 volts for a 1 bit because it leads to ambiguities. This problem can be solved by using +1 volts for a 1 and -1 volts for a 0, but there is still the problem of a receiver sampling the signal at a slightly different frequency than the sender used to generate it. Different clock speeds can cause the receiver and sender to get out of synchronization about where the bit boundaries are, especially after a long run of consecutive 0s or a long run of consecutive 1s.

What is needed is a way for receivers to unambiguously determine the start, end, or middle of each bit without reference to an external clock. Two such approaches are called **Manchester encoding** and **differential Manchester encoding**.

Manchester encoding:

With Manchester encoding, each bit period is divided into two equal intervals. A binary 1 bit is sent by having the voltage set high during the first interval and low in the second one. A binary 0 is just the reverse: first low and then high.

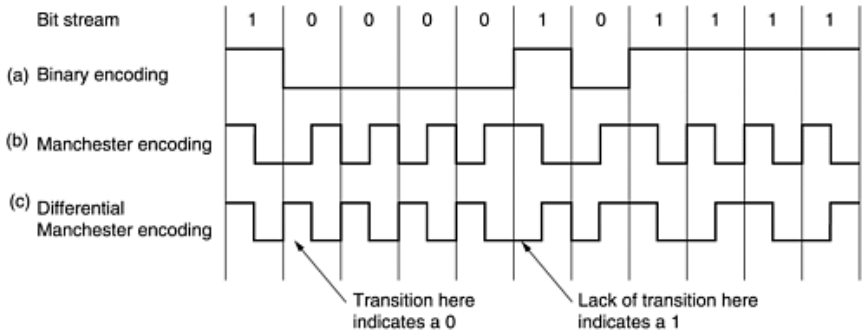
This scheme ensures that every bit period has a transition in the middle, making it easy for the receiver to synchronize with the sender.

A disadvantage of Manchester encoding is that it requires twice as much bandwidth as straight binary encoding because the pulses are half the width.

Differential Manchester encoding:

Differential Manchester encoding, is a variation of basic Manchester encoding. In it, a 1 bit is indicated by the absence of a transition at the start of the interval. A 0 bit is indicated by the presence of a transition at the start of the interval.

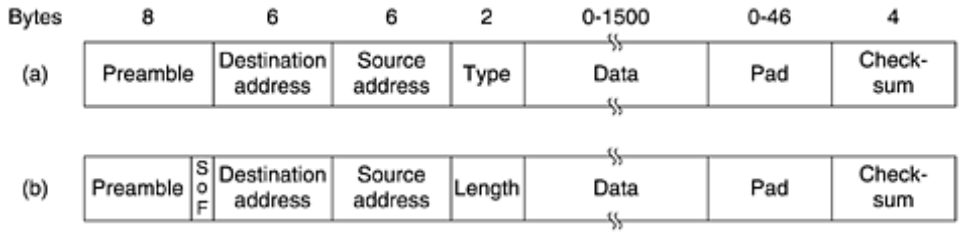
In both cases, there is a transition in the middle as well. The differential scheme requires more complex equipment but offers better noise immunity. All Ethernet systems use Manchester encoding due to its simplicity.



3. The Ethernet MAC Sublayer Protocol:

The original DIX (DEC, Intel, Xerox) frame structure is shown below in Fig.(a). Each frame starts with a *Preamble* of 8 bytes, each containing the bit pattern 10101010.

The Manchester encoding of this pattern produces a 10-MHz square wave for 6.4 µsec to allow the receiver's clock to synchronize with the senders. They are required to stay synchronized for the rest of the frame, using the Manchester encoding to keep track of the bit boundaries.



The frame contains two addresses, one for the destination and one for the source. The standard allows 2-byte and 6-byte addresses, but the

parameters defined for the 10-Mbps baseband standard use only the 6-byte addresses. The high-order bit of the destination address is a 0 for ordinary addresses and 1 for group addresses. Group addresses allow multiple stations to listen to a single address. When a frame is sent to a group address, all the stations in the group receive it. Sending to a group of stations is called multicast. The address consisting of all 1 bits is reserved for broadcast. A frame containing all 1s in the destination field is accepted by all stations on the network. The difference between multicast and broadcast is important enough to warrant repeating. A multicast frame is sent to a selected group of stations on the Ethernet; a broadcast frame is sent to all stations on the Ethernet. Multicast is more selective, but involves group management. Broadcasting is coarser but does not require any group management.

Next comes the *Type* field, which tells the receiver what to do with the frame. Multiple network-layer protocols may be in use at the same time on the same machine, so when an Ethernet frame arrives, the kernel has to know which one to hand the frame to. The *Type* field specifies which process to give the frame to.

Next come the data, up to 1500 bytes. This limit was chosen somewhat arbitrarily at the time the DIX standard was cast in stone, mostly based on the fact that a transceiver needs enough RAM to hold an entire frame and RAM was expensive in 1978. A larger upper limit would have meant more RAM, hence a more expensive transceiver.

Another interesting feature of the addressing is the use of bit 46 (adjacent to the high-order bit) to distinguish local from global addresses. Local addresses are assigned by each network administrator and have no significance outside the local network. Global addresses, in contrast, are assigned centrally by IEEE to ensure that no two stations anywhere in the world have the same global address. The final Ethernet field is the *Checksum*. It is effectively a 32-bit hash code of the data. If some data bits are erroneously received (due to

noise on the cable), the checksum will almost certainly be wrong and the error will be detected.

When IEEE standardized Ethernet, the committee made two changes to the DIX format, as shown in Fig.(b). The first one was to reduce the preamble to 7 bytes and use the last byte for a *Start of Frame* delimiter, for compatibility with 802.4 and 802.5. The second one was to change the *Type* field into a *Length* field.

Unfortunately, by the time 802.3 was published, so much hardware and software for DIX Ethernet was already in use that few manufacturers and users were enthusiastic about converting the *Type* field into a *Length* field. In 1997 IEEE threw in the towel and said that both ways were fine with it.

Fortunately, all the *Type* fields in use before 1997 were greater than 1500. Consequently, any number there less than or equal to 1500 can be interpreted as *Length*, and any number greater than 1500 can be interpreted as *Type*.

4. The Binary Exponential Backoff Algorithm:

Let us now see how randomization is done when a collision occurs. After a collision, time is divided into discrete slots whose length is equal to the worst-case round-trip propagation time on the ether (2τ). To accommodate the longest path allowed by Ethernet, the slot time has been set to 512 bit times, or 51.2 μsec as mentioned above.

After the first collision, each station waits either 0 or 1 slot times before trying again. If two stations collide and each one picks the same random number, they will collide again. After the second collision, each one picks either 0, 1, 2, or 3 at random and waits that number of slot times. If a third collision occurs (the probability of this happening is 0.25), then the next time the number of slots to wait is chosen at random from the interval 0 to $2^3 - 1$.

In general, after *i* collisions, a random number between 0 and $2^i - 1$ is chosen, and that number of slots is skipped. However, after ten collisions have been

reached, the randomization interval is frozen at a maximum of 1023 slots. After 16 collisions, the controller throws in the towel and reports failure back to the computer. Further recovery is up to higher layers.

This algorithm, called **binary exponential backoff**, was chosen to dynamically adapt to the number of stations trying to send. If the randomization interval for all collisions was 1023, the chance of two stations colliding for a second time would be negligible, but the average wait after a collision would be hundreds of slot times, introducing significant delay.

On the other hand, if each station always delayed for either zero or one slots, then if 100 stations ever tried to send at once, they would collide over and over until 99 of them picked 1 and the remaining station picked 0. This might take years.

By having the randomization interval grow exponentially as more and more consecutive collisions occur, the algorithm ensures a low delay when only a few stations collide but also ensures that the collision is resolved in a reasonable interval when many stations collide. Truncating the backoff at 1023 keeps the bound from growing too large.

As described so far, CSMA/CD provides no acknowledgements. Since the mere absence of collisions does not guarantee that bits were not garbled by noise spikes on the cable, for reliable communication the destination must verify the checksum, and if correct, send back an acknowledgement frame to the source.

Normally, this acknowledgement would be just another frame as far as the protocol is concerned and would have to fight for channel time just like a data frame. However, a simple modification to the contention algorithm would allow speedy confirmation of frame receipt.

5. Switched Ethernet:

With the growth of multimedia, transmission using 100-Mbps or 1-Gbps Ethernet can even become saturated.

Fortunately, there is an additional way to deal with increased load: switched Ethernet, as shown in Fig. The heart of this system is a **switch** containing a high-speed backplane and room for typically 4 to 32 plug-in line cards, each containing one to eight connectors. Most often, each connector has a 10Base-T twisted pair connection to a single host computer.

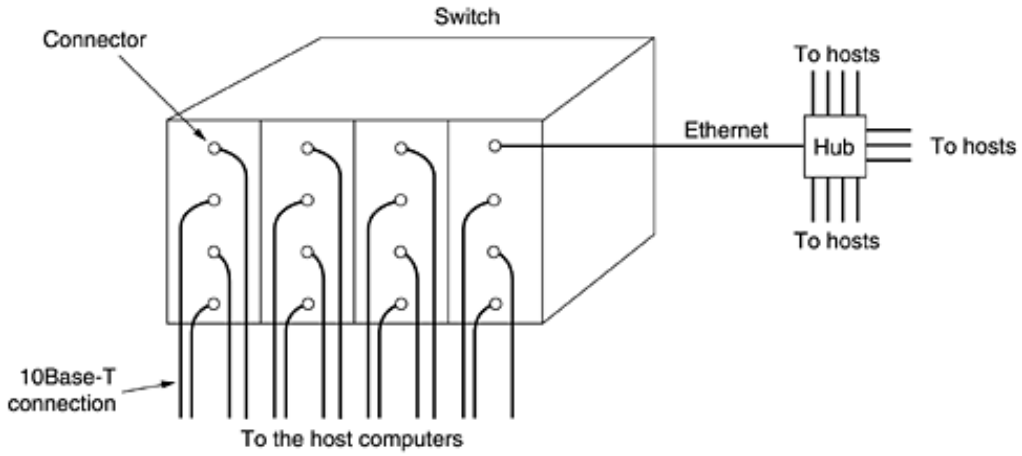


Figure (a). A simple example of switched Ethernet.

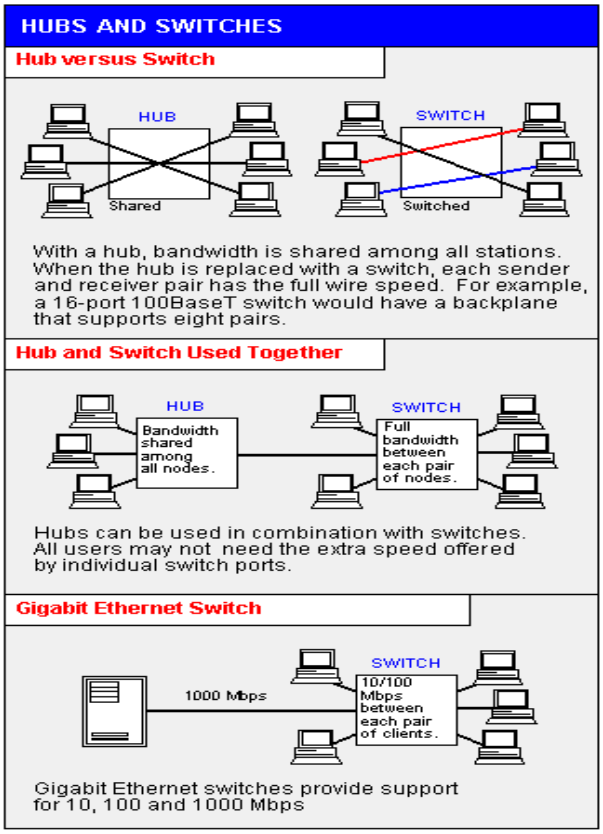


Figure (b). Hubs and Switches used in Ethernet.

When a station wants to transmit an Ethernet frame, it outputs a standard frame to the switch. The plug-in card getting the frame may check to see if it is destined for one of the other stations connected to the same card. If so, the frame is copied there. If not, the frame is sent over the high-speed backplane to the destination station's card. The backplane typically runs at many Gbps, using a proprietary protocol.

If two machines attached to the same plug-in card transmit frames at the same time. It depends on how the card has been constructed. One possibility is for all the ports on the card to be wired together to form a local on-card LAN. Collisions on this on-card LAN will be detected and handled the same as any other collisions on a CSMA/CD network—with retransmissions using the binary exponential backoff algorithm.

With this kind of plug-in card, only one transmission per card is possible at any instant, but all the cards can be transmitting in parallel. With this design, each card forms its own **collision domain**, independent of the others. With only one station per collision domain, collisions are impossible and performance is improved.

With the other kind of plug-in card, each input port is buffered, so incoming frames are stored in the card's on-board RAM as they arrive. This design allows all input ports to receive (and transmit) frames at the same time, for parallel, full-duplex operation, something not possible with CSMA/CD on a single channel. Once a frame has been completely received, the card can then check to see if the frame is destined for another port on the same card or for a distant port.

In the former case, it can be transmitted directly to the destination. In the latter case, it must be transmitted over the backplane to the proper card. With this design, each port is a separate collision domain, so collisions do not occur.

Since the switch just expects standard Ethernet frames on each input port, it is possible to use some of the ports as concentrators. In Fig (a), the port in

the upper-right corner is connected not to a single station, but to a 12-port hub. As frames arrive at the hub, they contend for the ether in the usual way, including collisions and binary backoff. Successful frames make it to the switch and are treated there like any other incoming frames: they are switched to the correct output line over the high-speed backplane. Hubs are cheaper than switches, but due to falling switch prices, they are rapidly becoming obsolete. Nevertheless, legacy hubs still exist.

6. IEEE 802.2: Logical Link Control

IEEE has defined one that can run on top of Ethernet and the other 802 protocols. In addition, this protocol, called LLC (Logical Link Control), hides the differences between the various kinds of 802 networks by providing a single format and interface to the network layer. LLC forms the upper half of the data link layer, with the MAC sublayer below it, as shown in Fig.

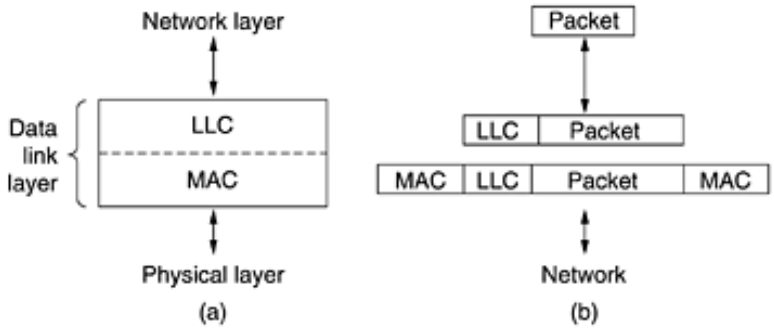


Figure. (a) Position of LLC. (b) Protocol formats.

Typical usage of LLC is as follows. The network layer on the sending machine passes a packet to LLC, using the LLC access primitives. The LLC sublayer then adds an LLC header, containing sequence and acknowledgement numbers. The resulting structure is then inserted into the payload field of an 802 frame and transmitted. At the receiver, the reverse process takes place.

LLC provides three service options: unreliable datagram service, acknowledged datagram service, and reliable connection-oriented service. The LLC header contains three fields: a destination access point, a source access point, and a control field.

The access points tell which process the frame came from and where it is to be delivered, replacing the DIX *Type* field. For the Internet, best-efforts attempts to deliver IP packets is sufficient, so no acknowledgements at the LLC level are required.

4.3 Data Link Layer Switching:

Many organizations have multiple LANs and wish to connect them. LANs can be connected by devices called bridges, which operate in the data link layer. Bridges examine the data layer link addresses to do routing. Before getting into the technology of bridges, it is worthwhile taking a look at some common situations in which bridges are used. We will mention six reasons why a single organization may end up with multiple LANs.

First, many university and corporate departments have their own LANs, primarily to connect their own personal computers, workstations, and servers. Since the goals of the various departments differ, different departments choose different LANs, without regard to what other departments are doing. Sooner or later, there is a need for interaction, so bridges are needed. In this example, multiple LANs came into existence due to the autonomy of their owners.

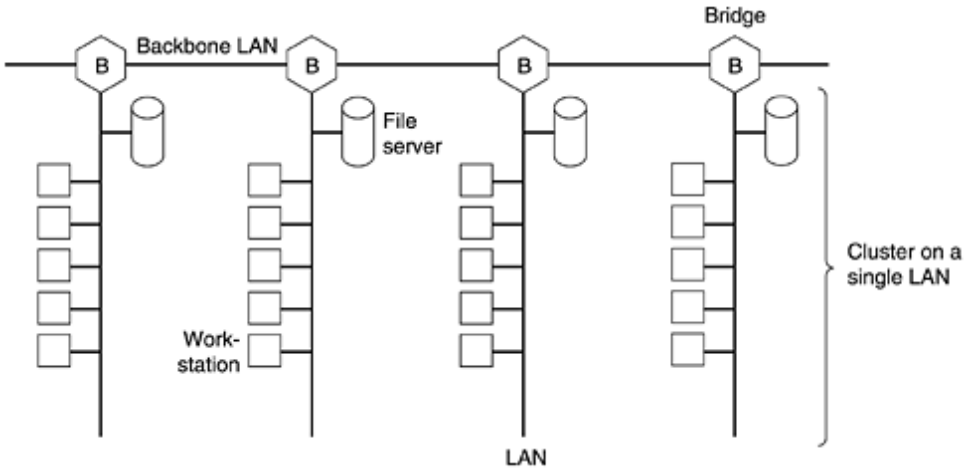
Second, the organization may be geographically spread over several buildings separated by considerable distances. It may be cheaper to have separate LANs in each building and connect them with bridges and laser links than to run a single cable over the entire site.

Third, it may be necessary to split what is logically a single LAN into separate LANs to accommodate the load. At many universities, for example, thousands of workstations are available for student and faculty computing.

Files are normally kept on file server machines and are downloaded to users' machines upon request.

The enormous scale of this system precludes putting all the workstations on a single LAN—the total bandwidth needed is far too high. Instead, multiple LANs connected by bridges are used, as shown in Fig below.

Each LAN contains a cluster of workstations with its own file server so that most traffic is restricted to a single LAN and does not add load to the backbone.



It is worth noting that although we usually draw LANs as multidrop cables as in Fig. 4-39 (the classic look), they are more often implemented with hubs or especially switches nowadays. However, a long multidrop cable with multiple machines plugged into it and a hub with the machines connected inside the hub are functionally identical. In both cases, all the machines belong to the same collision domain, and all use the CSMA/CD protocol to send frames. Switched LANs are different, however, as we saw before and will see again shortly.

Fourth, in some situations, a single LAN would be adequate in terms of the load, but the physical distance between the most distant machines is too

great (e.g., more than 2.5 km for Ethernet). Even if laying the cable is easy to do, the network would not work due to the excessively long round-trip delay. The only solution is to partition the LAN and install bridges between the segments. Using bridges, the total physical distance covered can be increased.

Fifth, there is the matter of reliability. On a single LAN, a defective node that keeps outputting a continuous stream of garbage can cripple the LAN. Bridges can be inserted at critical places, like fire doors in a building, to prevent a single node that has gone berserk from bringing down the entire system. Unlike a repeater, which just copies whatever it sees, a bridge can be programmed to exercise some discretion about what it forwards and what it does not forward.

Sixth, and last, bridges can contribute to the organization's security. Most LAN interfaces have a promiscuous mode, in which all frames are given to the computer, not just those addressed to it. Spies and busybodies love this feature. By inserting bridges at various places and being careful not to forward sensitive traffic, a system administrator can isolate parts of the network so that its traffic cannot escape and fall into the wrong hands.

Ideally, bridges should be fully transparent, meaning it should be possible to move a machine from one cable segment to another without changing any hardware, software, or configuration tables.

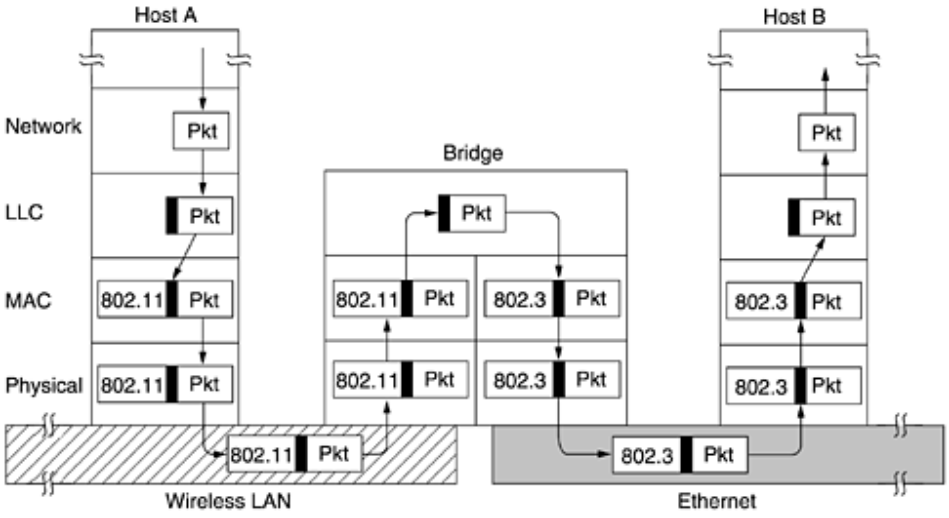
4.1 Bridges from 802.x to 802.y:

Having seen why bridges are needed, let us now turn to the question of how they work. Figure 4-40 illustrates the operation of a simple two-port bridge. Host A on a wireless (802.11) LAN has a packet to send to a fixed host, B, on

an (802.3) Ethernet to which the wireless LAN is connected. The packet descends into the LLC sublayer and acquires an LLC header (shown in black in the figure). Then it passes into the MAC sublayer and an 802.11 header is prepended to it (also a trailer, not shown in the figure).

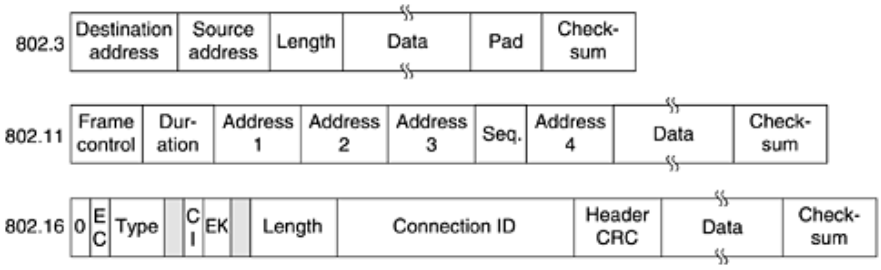
This unit goes out over the air and is picked up by the base station, which sees that it needs to go to the fixed Ethernet. When it hits the bridge connecting the 802.11 network to the 802.3 network, it starts in the physical layer and works

its way upward. In the MAC sublayer in the bridge, the 802.11 header is stripped off. The bare packet (with LLC header) is then handed off to the LLC sublayer in the bridge.



So far it looks like moving a frame from one LAN to another is easy. Such is not the case. In this section we will point out some of the difficulties that one encounters when trying to build a bridge between the various 802 LANs (and MANs). We will focus on 802.3, 802.11, and 802.16, but there are others as well, each with its unique problems.

1) To start with, each of the LANs uses a different frame format.



2) A second problem is that interconnected LANs do not necessarily run at the same data rate. When forwarding a long run of back-to-back frames from a fast LAN to a slower one, the bridge will not be able to get rid of the frames as fast as they come in. For example, if a gigabit Ethernet is pouring bits into an 11-Mbps 802.11b LAN at top speed, the bridge will have to buffer them, hoping not to run out of memory.

3) A third problem, and potentially the most serious of all, is that different 802 LANs have different maximum frame lengths. An obvious problem arises when a long frame must be forwarded onto a LAN that cannot accept it. Splitting the frame into pieces is out of the question in this layer. All the protocols assume that frames either arrive or they do not. There is no provision for reassembling frames out of smaller units.

4) Another point is security. Both 802.11 and 802.16 support encryption in the data link layer. Ethernet does not. This means that the various encryption services available to the wireless networks are lost when traffic passes over an Ethernet. Worse yet, if a wireless station uses data link layer encryption, there will be no way to decrypt it when it arrives over an Ethernet. One solution to the security problem is to do encryption in a higher layer.

5) A final point is quality of service. Both 802.11 and 802.16 provide it in various forms, the former using PCF mode and the latter using constant bit rate connections. Ethernet has no concept of quality of service, so traffic from

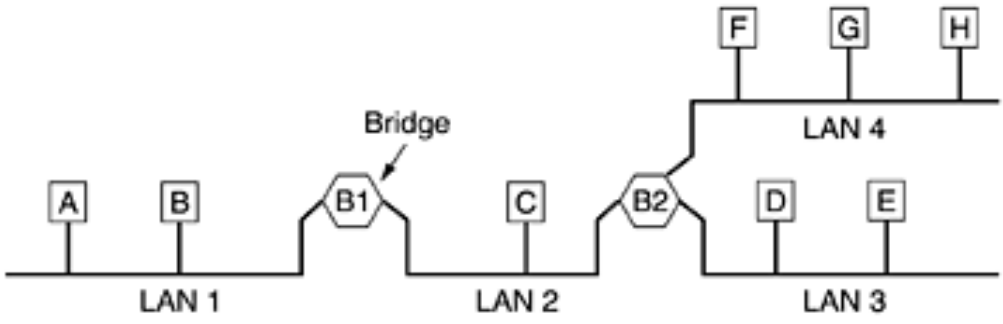
either of the others will lose its quality of service when passing over an Ethernet.

4.2 Local Internetworking:

The previous section dealt with the problems encountered in connecting two different IEEE 802 LANs via a single bridge. However, in large organizations with many LANs, just interconnecting them all raises a variety of issues, even if they are all just Ethernet. Ideally, it should be possible to go out and buy bridges designed to the IEEE standard, plug the connectors into the bridges, and everything should work perfectly, instantly.

There should be no hardware changes required, no software changes required, no setting of address switches, no downloading of routing tables or parameters, nothing. Just plug in the cables and walk away. Furthermore, the operation of the existing LANs should not be affected by the bridges at all.

In its simplest form, a transparent bridge operates in promiscuous mode, accepting every frame transmitted on all the LANs to which it is attached. As an example, consider the configuration of Fig. Bridge B1 is connected to LANs 1 and 2, and bridge B2 is connected to LANs 2, 3, and 4. A frame arriving at bridge B1 on LAN 1 destined for A can be discarded immediately, because it is already on the correct LAN, but a frame arriving on LAN 1 for C or F must be forwarded.



When a frame arrives, a bridge must decide whether to discard or forward it, and if the latter, on which LAN to put the frame. This decision is made by looking up the destination address in a big (hash) table inside the bridge. The table can list each possible destination and tell which output line (LAN) it belongs on. For example, B2's table would list A as belonging to LAN 2, since all B2 has to know is which LAN to put frames for A on. That, in fact, more forwarding happens later is not of interest to it.

When the bridges are first plugged in, all the hash tables are empty. None of the bridges know where any of the destinations are, so they use a flooding algorithm: every incoming frame for an unknown destination is output on all the LANs to which the bridge is connected except the one it arrived on. As time goes on, the bridges learn where destinations are, as described below. Once a destination is known, frames destined for it are put on only the proper LAN and are not flooded.

The algorithm used by the transparent bridges is backward learning. As mentioned above, the bridges operate in promiscuous mode, so they see every frame sent on any of their LANs. By looking at the source address, they can tell which machine is accessible on which LAN.

For example, if bridge B1 in Fig. sees a frame on LAN 2 coming from C, it knows that C must be reachable via LAN 2, so it makes an entry in its hash table noting that frames going to C should use LAN 2. Any subsequent frame addressed to C coming in on LAN 1 will be forwarded, but a frame for C coming in on LAN 2 will be discarded.

The topology can change as machines and bridges are powered up and down and moved around. To handle dynamic topologies, whenever a hash table entry is made, the arrival time of the frame is noted in the entry. Whenever a

frame whose source is already in the table arrives, its entry is updated with the current time. Thus, the time associated with every entry tells the last time a frame from that machine was seen.

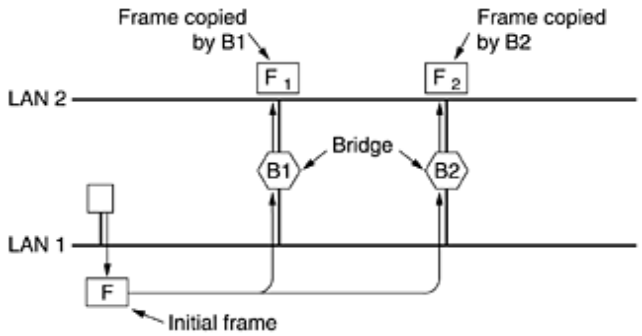
Periodically, a process in the bridge scans the hash table and purges all entries more than a few minutes old. In this way, if a computer is unplugged from its LAN, moved around the building, and plugged in again somewhere else, within a few minutes it will be back in normal operation, without any manual intervention.

The routing procedure for an incoming frame depends on the LAN it arrives on (the source LAN) and the LAN its destination is on (the destination LAN), as follows:

1. If destination and source LANs are the same, discard the frame.
2. If the destination and source LANs are different, forward the frame.
3. If the destination LAN is unknown, use flooding.

4.3 Spanning Tree Bridges:

To increase reliability, some sites use two or more bridges in parallel between pairs of LANs, as shown in Fig. 4-43. This arrangement, however, also introduces some additional problems because it creates loops in the topology. Fig. Shows two parallel transparent bridges.



A simple example of these problems can be seen by observing how a frame, F, with unknown destination is handled in Fig. 4-43. Each bridge, following the normal rules for handling unknown destinations, uses flooding, which in this example just means copying it to LAN 2. Shortly thereafter, bridge 1 sees F2, a frame with an unknown destination, which it copies to LAN 1, generating F3 (not shown). Similarly, bridge 2 copies F1 to LAN 1 generating F4 (also not shown). Bridge 1 now forwards F4 and bridge 2 copies F3. This cycle goes on forever.

The solution to this difficulty is for the bridges to communicate with each other and overlay the actual topology with a spanning tree that reaches every LAN. In effect, some potential connections between LANs are ignored in the interest of constructing a fictitious loop-free topology.

For example, in Fig.(a) we see nine LANs interconnected by ten bridges. This configuration can be abstracted into a graph with the LANs as the nodes. An arc connects any two LANs that are connected by a bridge. The graph can be reduced to a spanning tree by dropping the arcs shown as dotted lines in Fig.(b).

Using this spanning tree, there is exactly one path from every LAN to every other LAN. Once the bridges have agreed on the spanning tree, all forwarding between LANs follows the spanning tree. Since there is a unique path from each source to each destination, loops are impossible.

To build the spanning tree, first the bridges have to choose one bridge to be the root of the tree. They make this choice by having each one broadcast its serial number, installed by the manufacturer and guaranteed to be unique worldwide.

The bridge with the lowest serial number becomes the root. Next, a tree of shortest paths from the root to every bridge and LAN is constructed. This tree is the spanning tree. If a bridge or LAN fails, a new one is computed.

The result of this algorithm is that a unique path is established from every LAN to the root and thus to every other LAN. Although the tree spans all the LANs, not all the bridges are necessarily present in the tree (to prevent loops). Even after the spanning tree has been established, the algorithm continues to run during normal operation in order to automatically detect topology changes and update the tree.

4.4 Remote Bridges:

A common use of bridges is to connect two (or more) distant LANs. For example, a company might have plants in several cities, each with its own LAN. Ideally, all the LANs should be interconnected, so the complete system acts like one large LAN.

This goal can be achieved by putting a bridge on each LAN and connecting the bridges pairwise with point-to-point lines (e.g., lines leased from a telephone company). A simple system, with three LANs, is illustrated in Fig. 4-45. The usual routing algorithms apply here. The simplest way to see this is to regard the three point-to-point lines as host less LANs. Then we have a normal system of six LANS interconnected by four bridges.

Various protocols can be used on the point-to-point lines. One possibility is to choose some standard point-to-point data link protocol such as PPP, putting complete MAC frames in the payload field. This strategy works best if all the LANs are identical, and the only problem is getting frames to the correct LAN. Another option is to strip off the MAC header and trailer at the source bridge and put what is left in the payload field of the point-to-point

protocol. A new MAC header and trailer can then be generated at the destination bridge. A disadvantage of this approach is that the checksum that arrives at the destination host is not the one computed by the source host, so errors caused by bad bits in a bridge's memory may not be detected.

4.5 Repeaters, Hubs, Bridges, Switches, Routers, and Gateways: [To be read from Text book Pg. No. 242]

4.6 Virtual LANs: [To be read from Text book Pg. No. 244]

****