UNIT 1 DATA LINK LAYER FUNDAMENTALS

Structure **Page Nos.**

1.0 INTRODUCTION

Data Link Layer (DLL) is the second layer of the OSI model which takes services from the Physical layer and provides services to the network layer. DLL transfers the data from a host to a node or a node to another node.

The main task of the data link layer is to take a raw data from transmission facility and transform it into a line that appears free of transmission errors to the network layer. Data packets are encoded and decoded into bits. These are called frames. The **Functions of the Data Link Layer** are Framing, Frame synchronisation, Error Handling Flow Regulation, Addressing and Access Control. The data link layer is divided into two sublayers: The Media Access Control (MAC) layer and the Logical Link Control (LLC) layer. The MAC sublayer controls how a computer on the network gains access to the link resources and grants permission to transmit it. The LLC layer controls frame synchronisation, flow control and error checking.

Frame is a data structure used in transmissions at DLL consisting of a header and a trailer bracketing a data frame. Packets are the fundamental unit of information transport in all modern computer networks, and increasingly, in other communication networks as well these are converted to frame at DLL. The destination host not receiving the data bits as transmitted by the source host is termed as error in data. The error needs to be detected or corrected, as the data on the network must be error free and reliable which is one of the key features of the DLL. There are various methods used for detection and correction of these errors viz.: FEC (Forward Error Correction) and CRC (Cyclic Redundancy Check). To transmit data from a source to a destination node, a node in a network should be uniquely identified. This identification is known as the address of the node. Access control is related to addressing the problem of "Which channel or node on a LAN or WAN would transmit data?" The data link layer also deals with the problem of difference in the speed of transmission of data by the sender and receiver. Very often the speed of the receiver is slower than the speed of the sender. To overcome the same, the DLL has many flow control mechanism as part of its functionality.

1.1 OBJECTIVES

After going through this unit, you should be able to understand:

- the functionality of the Data Link Layer;
- the concept of framing and how framing is done;

- the types of errors that can be generated in frames during transmission;
- error in a data frame;
- methods for detecting errors;
- methods for correcting errors;
- forwarding the error correction method for error correction;
- following the CRC method for error detection;
- the meaning of flow control, and
- the methods for Managing Flow Control and error control.

1.2 FRAMING

As you already know, the physical layer deals with raw transmission of data in the form of bits and gives services to the data link layer. The data link layer provides services to the network layer as shown in the *Figure 1*:

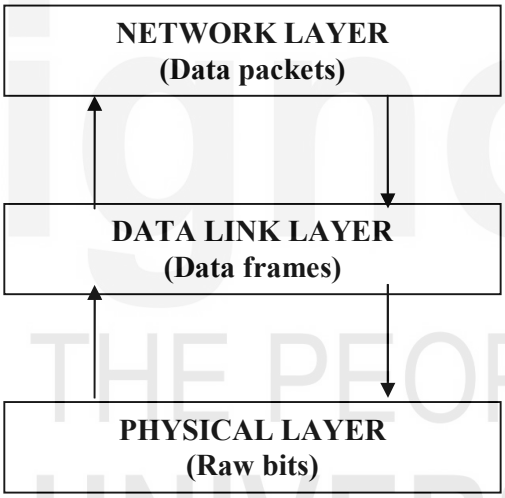


Figure 1: Data link layer providing services to network layer

The raw data coming from the Physical layer is converted into frames for forwardingto the network layer. This is done to ensure that the transmission of data is error free. Error detection and correction (if required) is done by the Data link layer, which is discussed in the following sections. The structure of the frame is shown in *Figure 2*.

Converting the bit stream into frames is a tedious process. The beginning and end of each frame should be explicitly marked. The easiest way to do so is to insert some time gap between frames.

Flag	Control Information	Data Value	Control Information	Flag

Figure 2: Frame format

But, it is difficult to keep track of counts on timing to mark the start and end time of each frame. So to overcome the same, we will discuss the following methods for framing.

- Character Count
- Character Patterns
- Bit Patterns
- Framing by Illegal code (code violation)

Character Count

The first framing method, Character count, uses a header field to specify the number of characters in the frame. The Data Link Layer at the destination checks the header field to know the size of the frame and hence, the end of frame. The process is shown in *Figure 3* for a four-frame of size 4, 5, 5 and 9 respectively.

--FRAME 1-- ------FRAME 2---- -----FRAME 3--- ----------FRAME 4---------

4	1	2	3	**5**	4	5	6	7	**5**	8	9	1	2	**9**	3	4	5	6	7	8	9	0

Figure 3: Character count

However, problems may arise due to changes in character count value during transmission. For example, in the first frame if the character count 4 changes to 6, the destination will receive data out of synchronisation and hence, it will not be able to identify the start of the next frame. This example is shown in *Figure. 4.* Even, after a request from the destination to the source for retransmission comes, it does not solve the problem because the destination does not know from where to start retransmission.

-------FRAME 1 ------ ----FRAME 2---

6	1	2	3	**5**	4	5	6	7	**5**	8	9	1	2	**9**	3	4	5	6	7	8	9	0

ERROR-----DATA Received Out of Order----------

Figure 4: Problems in character count

Character Patterns

The second framing method, Character stuffing, solves the problem of out of synchronisation. Here, each frame starts with special character set e.g., a special ASCII character sequence.

The frame after adding the start sequence and the end sequence is shown in *Figure 5.*

Begin each frame with DLE STX.

End each frame with DLE ETX.

DLE stands for Data Link Escape

STX stands for Start of Text

ETX stands for End of Text

DLE	STX	……	DLE	ETX

Figure 5: Character patterns (Character stuffing)

If a DLE ETX occurs in the middle of the data and interferes with the data during framing then, insert an ASCII DLE character just before DLE character in the data. The Receiver interprets the single DLE as an escape indicating that the next character is a control character.

If two DLE's appear in succession at the receiver's end, the first is discarded and the second is regarded as data. Thus, framing DLE STX or DLE ETX is distinguished by whether DLE is present once or twice.

Here, the example is explained with the help of *Figures 6(a), 6(b) and 6(c).*

Figure 6(a) : Before stuffing

Figure 6(b) : After stuffing

Figure 6(c) : After destuffing

A problem with character stuffing is that not all bit streams are character oriented (e.g., Unicode is a 16-bit code). Hence, for arbitrary sized characters the process of character stuffing becomes more complex. So next we will discuss a new method known as the bit stuffing, which solves the problem of arbitrary sized character.

Bit Patterns

This method is similar to the one discussed above, except that, the method of bit stuffing allows insertion of bits instead of the entire character (8 bits). Bit pattern framing uses a particular sequence of bits called a ***flag*** for framing. The flag is set as the start and the end of the frame.

Use of ***bit patterns*** is to keep the sequence of data in the same order.

Flag = 0 1 1 1 1 1 1 0 (begins and ends frame.)

In this case the transmitter automatically inserts a 0 after 5 consecutive 1's in the data. This is called Bit stuffing. The receiver discards these stuffed 0's as soon as it encounters 5 consecutive 1's in the received data as shown with the help of an example described in *Figure 7 (a), (b) and (c).*

11111111110010011

Figure 7(a) : Original data ready to be sent

011111101111101111100010011 01111110

Figure 7(b) : Data after adding Flag and stuffing

11111111110010011

Figure 7(c) : Data after destuffing

Framing by Illegal Code (Code violation)

A fourth method is based on any redundancy in the coding scheme. In this method, we simply identify an illegal bit pattern, and use it as a beginning or end marker, i.e., certain physical layers use a line code for timing reasons.

For example: Manchester Encoding

1 –It can be coded into two parts i.e., high to low ⊓⊔ ≡ 1 0

0 – It can be coded into two parts i.e., low to high ⊔⊓ ≡ 0 1

Codes of all low (000) or all high (111) aren't used for the data and therefore, can be used for framing.

☞ Check Your Progress 1

1) Name different framing methods.

……

……

……

2) Write the bit sequence after bit stuffing for the data stream
110001111111100001111100

……

……

……

3) Why bit stuffing is advantageous over character stuffing?

……

……

……

1.3 BASICS OF ERROR DETECTION

The Network should ensure complete and accurate delivery of data from the source node to destination node. But many times data gets corrupted during transmission. As already discussed in the previous block, many factors can corrupt or alter the data that leads to an error. A reliable system should have methods to detect and correct the errors. Firstly, we will discuss what the error could be then, in the later section we will discuss the process of detecting and correcting them.

Types of Error

Several types of error may occur during transmission over the network:

- 1-bit error
- burst error
- lost message (frame)

1-bit error: **1**-bit error/Single bit error means that only one bit is changed in the data during transmission from the source to the destination node i.e., either 0 is changed to 1 or 1 is changed to 0 as shown in *Figure 8*.

0	0	0	1	0	**0**	0	0	(Sent by source)
0	0	0	1	0	**1**	0	0	(Received by destination)

Figure 8: 1 bit error

This error will not appear generally in case of serial transmission. But it might appear in case of parallel transmission.

Burst error: Burst error means that 2 or more bits of data are altered during transmission from the source to the destination node. But, it is not necessary that error will appear in consecutive bits. Size of burst error is from the first corrupted bit to the last corrupted bit as shown in *Figure 9*.

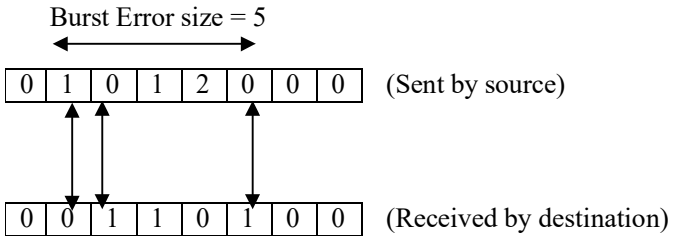


Figure 9: Burst error

An *n*-bit burst error is a string of bits inverted during transmission. This error will hardly occur in case of parallel transmission. But, it is difficult to deal with all corrupted bits at one instance.

Lost Message (Frame): The sender has sent the frame but that is not received properly, this is known as loss of frame during transmission. To deal with this type of error, a retransmission of the sent frame is required by the sender. We will discuss retransmission strategies in the next unit.

Now we will discuss some methods for error detection.

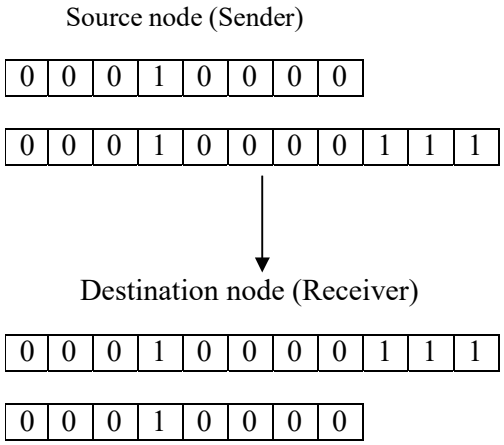
Error Detection

As already discussed in the beginning of this section accurate delivery of data at the receiver's site is, one of the important goals of this layer. This implies that the receivers should get the data that is error free. However, due to some factors if, the data gets corrupted, we need to correct it using various techniques. So, we require error detection methods first to detect the errors in the data before correcting it. Error detection is an easy process.

For error detection the sender can send every data unit twice and the receiver will do bit by bit comparison between the two sets of information. Any alteration found after the comparison will, indicate an error and a suitable method can be applied to correct the error.

But, sending every data unit twice increases the transmission time as well as overhead in comparison. Hence, the basic strategy for dealing with errors is to include groups of bits as additional information in each transmitted frame, so that, the receiver can detect the presence of errors. This method is called Redundancy as extra bits appended in each frame are redundant. At the receiver end these extra bits will be discarded when the accuracy of data is confirmed.

For example:



Redundancy check methods commonly used in data transmission are:

- Parity check
- CRC
- Checksum

Parity Check

- The most common method used for detecting errors when the number of bits in the data is small, is the use of the ***parity bit.***
- A *parity bit* is an extra binary digit added to the group of data bits, so that, the total number of one's in the group is even or odd.
- Data bits in each frame is inspected prior to transmission and an extra bit (the parity bit) is computed and appended to the bit string to ensure even or odd parity, depending on the protocol being used.
- If odd parity is being used, the receiver expects to receive a block of data with an odd number of 1's.
- For even parity, the number of 1's should be even.

In the example below, even parity is used. The ninth column contains the parity bit.

0 1 0 1 0 1 0 1 **0**
0 1 1 1 1 0 0 1 **1**
1 1 1 1 0 0 1 1 **0**

- Use of parity bits a rather weak mechanism for detecting errors.
- A single parity bit can only detect single-bit errors.

Block Sum Check Method

- This is an extension to the single parity bit method. This can be used to detect up to two bit errors in a block of characters.
- Each byte in the frame is assigned a parity bit (*row parity*).
- An extra bit is computed for each bit position (*column parity*).
- The resulting set of parity bits for each column is called the *block sum check.* Each bit that makes up the character is the modulo-2 sum of all the bits in the corresponding column.

Block Sum Check, *example*

Sender's data: 0000100 0010101 0101011

1	0 0 0 0 1 0 0
1	0 0 1 0 1 0 1
0	0 1 0 1 0 1 1
0	**0 1 1 1 0 1 0**

Data after adding parity bits:

00001000 00101010 01010110 01110100

This method detects single bit errors as well as increases the probability of finding burst error.

CRC

***Cyclic Redundancy Check (CRC)* is used to detect burst errors**

In this method, you would need to treat strings of bits as coefficients of a polynomial code that uses modulo 2 arithmetic. In modulo 2 arithmetic there are no carriers for addition and borrows for subtraction. Polynomial codes treat bit strings as representative of polynomials with coefficients of 0 and 1 only.

For example, the bit sequence 1 0 01 0 1 is represented by the polynomial $x^5 + x^2 + 1$ ($1.x^5 + 0.x^4 + 0.x^3 + 1.x^2 + 0.x^2 + 1.x^0$). When the polynomial method is employed, the sender and the receiver must agree upon a generator polynomial both the high and low order bits of the generator must be 1.

In this method the Sender divides frame (data string) by a predetermined Generator Polynomial and then appends the remainder (called checksum) onto the frame before starting the process of transmission. At the receiver end, the receiver divides the received frame by the same Generator polynomial. If the remainder obtained after the division is zero, it ensure that data received at the receiver's end is error free. All operations are done modulo 2.

An example that explains finding CRC (Cyclic Redundancy Check) will follow later.

Checksum

In this method the checksum generator divides the given input data into equal segments of k bits(8 or 16). The addition of these segments using ones complement arithmetic is complimented. This result is known as the checksum and it is appended with the data stream. This appended data stream is transmitted across the network on the transmission media. At the receiver end add all received segments. If the addition of segments at the receiver end is all 1's then, the data received is error free as, complement of the same will be all 0's. Then the data can be accepted, otherwise, data can be discarded.

For example:

Sender's data 0 0 0 0 0 0 1 0 0 1 0 1 0 0 0 0

	00000010
	01010000
Sum	**01010010**

Checksum (Compliment) 10101101

Data with appended checksum: 00000010 01010000 10101101

Receiver's accept the data as

00000010 01010000 10101101

At receiver's end

	00000010
	01010000
	10101101
sum	**11111111**
Complement	**00000000**

As complement is 0 it indicates that the data received at the receiver's end is error free so, it will be accepted by the receiver.

The checksum method detects all errors as it retains all its carries.

- Odd number of bits
- Most of even number of bits.

☞ Check Your Progress 2

1) Define parity bit? Also define even and odd parity?

……..

……..

……

2) Name the method that can detect single bit error and burst error.

……..

……..

……..

3) Define checksum.

……..

……..

……

1.4 FORWARD ERROR CORRECTION

In the previous section, we have discussed detection of error that appears after the transmission of data over the network from sender to receiver. In this section, we will study how to perform correction of errors. Forward Error Correction (FEC) is a type of error correction method which improves on simple error detection schemes by enabling the receiver to correct errors once they are detected. This reduces the need for retransmissions of error frames.

Firstly we consider the simple case i.e., correcting single bit error. As we have discussed earlier that a single bit error can be detected by adding one additional bit (parity bit/ redundant bit). This additional bit can detect error in any bit stream by differentiating the two condition error or not error as a bit can have two states only i.e., 0 and 1.

For correction of detected single bit error two states are not sufficient. As an error occurs in bit stream indicates that one bit is altered from either 0 to 1 or 1 to 0. To correct the same, conversion of altered bit is required. For performing this conversion we must know the location of bit which is in error. Therefore, for error correction identification of location of error bit is required. For example, for applying error correction of single bit error in ASCII character we must find which of 7 bit is altered. For doing this we could have eight different states i.e., no error, error in bit position 1, error in bit position 2 up to error in bit position7. For this we need many redundant bits to represent all eight states.

Here, 3 bit redundancy code can represent all possible eight states because 3 bits can represent 8 states (000 to 111). But if an error occurs in redundancy bit then we need 3 Additional bits added with 7 ASCII character bits, it covers all possible error locations.

So we can generalise if we have n data bits and r redundancy bits then total n+r will be the transmittable bits and r bits must be able to represent at least n+r+1 different states, here plus 1 indicates no error state. Hence n+r+1 states are identifiable by r additional bits and r additional bits represents 2^r different states.

- $2^r >= n+r+1$

For example, in 7 bit ASCII character we need at least 4 redundant bits as $2^4>=7+4+1$.

Hamming code is one such error correcting code.

For the example discussed above for a 7 bit ASCII character and 4 redundant bit, we will have total bits as n+r i.e., 7+4 = 11. These 4 redundant bits can be inserted in 7 bit data stream in position 1,2,4 and 8 (in 11 bit sequence at 2^0,2^1,2^2,2^3) named as r1,r2,r4 and r8 respectively.

In Hamming code each redundant bit is the combination of data bits where each data bit can be included in more than one combination as shown below.

r1 : 1,3,5,7,9,11
r2 : 2,3,6,7,10,11
r3 : 4,5,6,7
r8 : 8,9,10,11

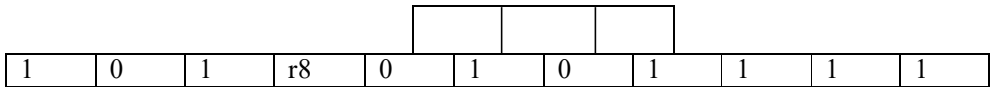
Now we, will find the values of redundant bit r1,r2, r4 and r8 for the data bit sequence1010101 as shown in following *Figure*.

11	10	9	8	7	6	5	4	3	2	1
1	0	1	r8	0	1	0	r4	1	r2	r1

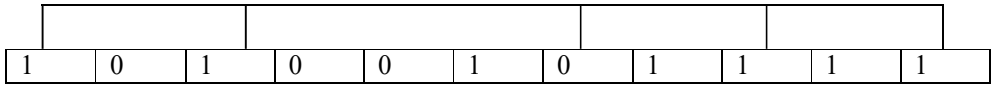
For finding values of r1, r2, r4 and r8 we will find even parities for various bit combination. The parity bit will be the value of redundant bit.

1	0	1	r8	0	1	0	r4	1	r2	1

1	0	1	r8	0	1	0	r4	1	1	1



1	0	1	r8	0	1	0	1	1	1	1



1	0	1	0	0	1	0	1	1	1	1

Assume that during transmission the number 5 bit is altered from 0 to 1. So at receiver end for error detection and correction new parities will be calculated as earlier.

Error new parity

1	0	1	0	0	1	1	1	1	1	1	1
1	0	1	0	0	1	1	1	1	1	1	0
1	0	1	0	0	1	1	1	1	1	1	1
1	0	1	0	0	1	1	1	1	1	1	0

0101 = 5. It implies 5th bit is the error bit. The binary number obtained from new parties will indicate the error bit location in the received data stream. Subsequently that bit can be altered and data can be corrected.

If all the values in the new parity column are 0, we conclude that the data is error free. In the given example if no error it should be 0000.

☞ **Check Your Progress 3**

1) Define Hamming distance.

……

……..……

…..…….

2) Generate the Hamming code for the following input data101111

……..

……..……

…..…….

1.5 CYCLIC REDUNDANCY CHECK CODES FOR ERROR DETECTION

The most commonly used method for detecting burst error in the data stream is Cyclic Redundancy Check Method. This method is based on the use of *polynomial codes.* Polynomial codes are based on representing bit strings as polynomials with coefficients as 0 and 1 only.

For example, the bit string 1110011 can be represented by the following polynomial:

$$1.x^6 + 1.x^5 + 1.x^4 + 0.x^3 + 0.x^2 + 1.x^1 + 1.x^0$$

This is equivalent to:

$$x^6 + x^5 + x^4 + x^1 + 1.$$

- The polynomial is manipulated using *modulo 2* arithmetic (which is equivalent to Exclusive OR or XOR).

- Depending on the content of the frame a set of check digits is computed for each frame that is to be transmitted. Then the receiver performs the same arithmetic as the sender on the frame and check the digits. If the result after computation is the same then the data received is error free.
- A different answer after the computation by the receiver indicates that, some error is present in the data.
- The computed check digits are called the frame check sequence (FCS) or the cyclic redundancy check (CRC).

The CRC method requires that:

- The sender and receiver should agree upon a generator polynomial before the transmission process start.
- Both the high and low bits of the generator must be 1.
- To compute the checksum for a frame with *m* bits, the size of the frame must belonger than the generator polynomial.

Algorithm for Computing the Checksum is as follows as:

> Let D(x) be the data and G(x) be the generating polynomial. Let r be the degree of generator polynomial G(x).

Step 1: Multiple the data D(x) by x^r, giving *r* zeros in the low-order end of the frame.

Step 2: Divide the result obtained in step1 by G(x), using modulo-2 division.

Step 3: Append the remainder from step 2 to D(x), thus, placing *r* terms in the *r* low-order positions.

The type of generating polynomial is important for finding the types of error that are detected.

- A generator polynomial of R bits will detect:
  - all single-bit errors
  - all double-bit errors
  - all odd-number of bit errors
  - all burst errors < R
  - most burst errors >= R

Example:

Data 1011101
Generator Polynomial G(x): $x^4+x^2+1 = 10101$
Here the size of the generator polynomial is 5 bit long, so, we will place 5 – 1 = 4 0's in the low order data stream. So the data will be:

Data 1011101 0000

Now using modulo2 division, (for getting data ready to be transmitted), we will divide the data by the generator polynomial as shown in the *Figure 10*.

Division diagram at sender's site

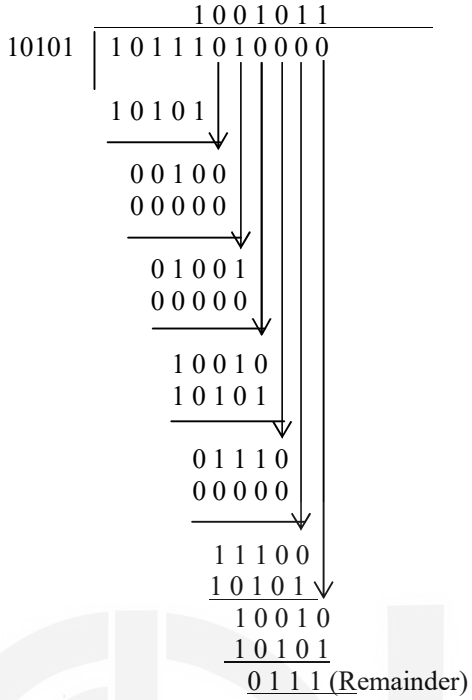


Figure 10: Cyclic redundancy check

The remainder obtained after the division of 0111, will be placed on the low order bits of the data stream. So the Data that is transmitted over the network by the sender is D(x) = 1011101 0111

Now at the receiver's end, the receiver receives D(x) as data which will be divided by the generator polynomial as shown in *Figure 11.*

Division diagram at receiver's site

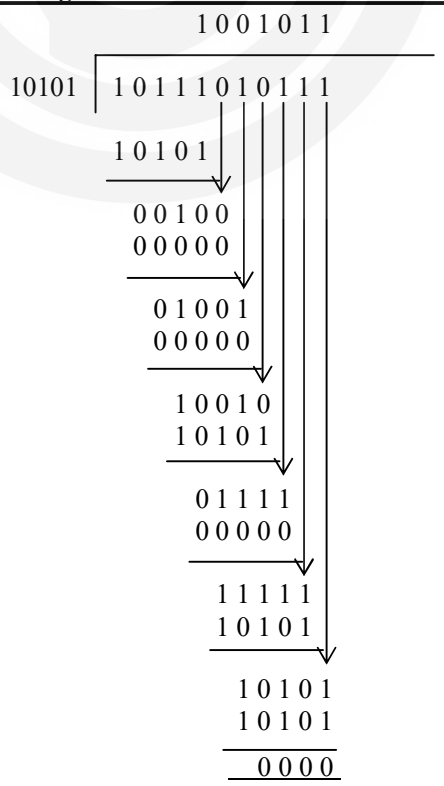


Figure 11: Cyclic redundancy check at the receiver side

Here the remainder obtained after division is 0000, so it ensure that data received at the receiver end is error free otherwise, it indicates that the data has some error in it. In this way the CRC method can detect whether the data has some error or is error free.

☞ Check Your Progress 4

1) Find the CRC for the data polynomial x^4+x^2+x+1 where generator polynomial is x^3+1.

……

……

……..

1.6 FLOW CONTROL

Another important issue for the data link layer is dealing with the situation which occurs when the sender transmits frames faster than the receiver can accept or process them. If the sender is working on a fast machine and the receiver is working on a slow machine this situation may occur in the network. In this process of transmission, some of the frames might be lost as they were not processed by the receiver due to it's low speed, while the sender might have through the transmission to be completely error free. To prevent this situation during transmission, a method is introduced called the *Flow Control.*

Flow control means using some feedback mechanism by which, the sender can be aware of when to the send next frame, and not at the usual speed of the sender. If the frame is accepted /processed by the receiver then only with the sender send the next frame. It may be said that the speed of the sender and the receiver should be compatible with each other, so that the receiver will receive or process all frames sent by the sender as every receiver has a limited block of memory called the buffer, reserved for storing incoming frames.

There are several methods available for deciding when a sender should send one frame or the next frame. Flow control ensures that the speed of sending the frame, by the sender, and the speed of processing the received frame by the receiver are compatible.

There are two basic strategies for flow control:

1) Stop-and-wait

2) Sliding window.

We shall start with the assumption that the transmission is error free and that we have an ideal channel.

Stop and Wait

Stop-and-Wait Flow Control (*Figure 12*) is the simplest form of flow control. In this method, the receiver indicates its readiness to receive data for each frame, the message is broken into multiple frames. The Sender waits for an ACK(acknowledgement) after every frame for a specified time (called time out). It is sent to ensure that the receiver has received the frame correctly. It will then send the next frame only after the ACK has been received.

Operations:

1) **Sender:** Transmits a single frame at a time.
2) **Receiver:** Transmits acknowledgement (ACK) as it receives a frame.
3) Sender receives ACK within time out.
4) Go to step 1.

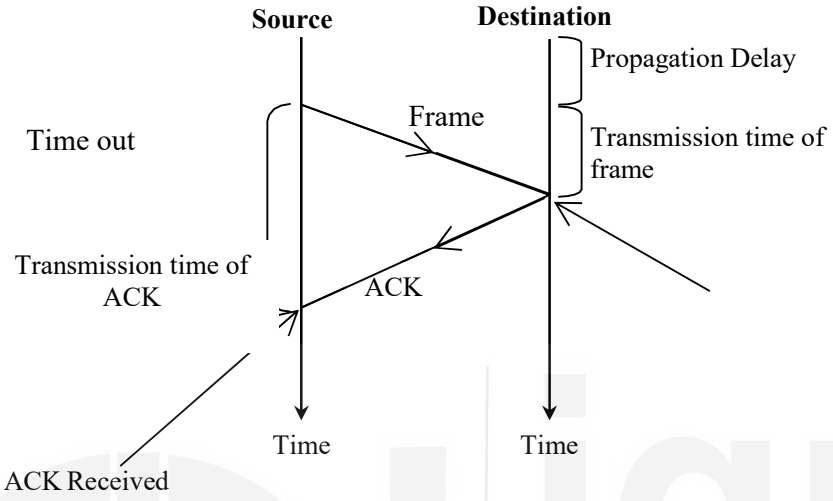


Figure 12: Stop and Wait

If a frame or ACK is lost during transmission then it has to be transmitted again by the sender. This retransmission process is known as ARQ (automatic repeat request). Stop and Wait ARQ is one of the simplest methods of flow control. It will be further discussed in detail in the next unit.

The problem with stop and wait is that only one frame can be transmitted at a time and that often leads to inefficient transmission channel till we get the acknowledgement the sender can not transmit any new packet. During this time both the sender and the channel are unutilised.

To deal with this problem, there is another flow control method i.e., sliding window protocol which is discussed below:

Sliding Window Protocol

In this flow control method, the receiver allocates buffer space for *n* frames in advance and allows transmission of multiple frames.This method allows the sender to transmit *n* frames without an ACK. A k-bit sequence number is assigned to each frame. The range of sequence number uses modulo-2 arithmetic. To keep track of the frames that have been acknowledged, each ACK has a *sequence* number. The receiver acknowledges a frame by sending an ACK that includes the Sequence number of the **next expected frame.** The sender sends the next *n* frames **starting with** the last received sequence number that has been transmitted by the receiver (ACK). Hence, a single ACK can acknowledge multiple frames as shown in the *Figure 13.*

The receiver receives frames 1, 2 and 3. Once frame 3 arrives ACK4 is sent to the sender. This ACK4 acknowledge the receipt of frame 1, 2 and 3 and informs the sender that the next expected frame is frame 4. Therefore, the sender can send multiple back-to-back frames, making efficient use of the channel.

Normal Flow diagram of a sliding window

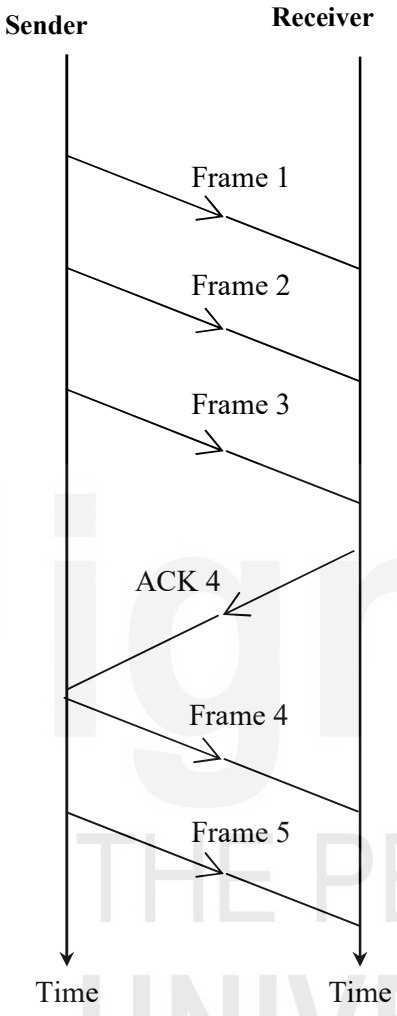


Figure 13: Normal flow diagram of sliding

Sequence number is a field in the frame that is of finite size. If *k* bits are reserved for the sequence number, then the values of sequence number ranges from 0 to $2^k - 1$ (Modulo Arithmetic).

Operation of a Sliding Window

Sending Window:

In this mechanism we maintain two types of windows (buffer) sending window and receiving windows as shown in *Figure 14 & 15.*

At any instant, the sender is permitted to send frames with sequence numbers in a certain range (3-bit sending window) as shown in *Figure 14.*

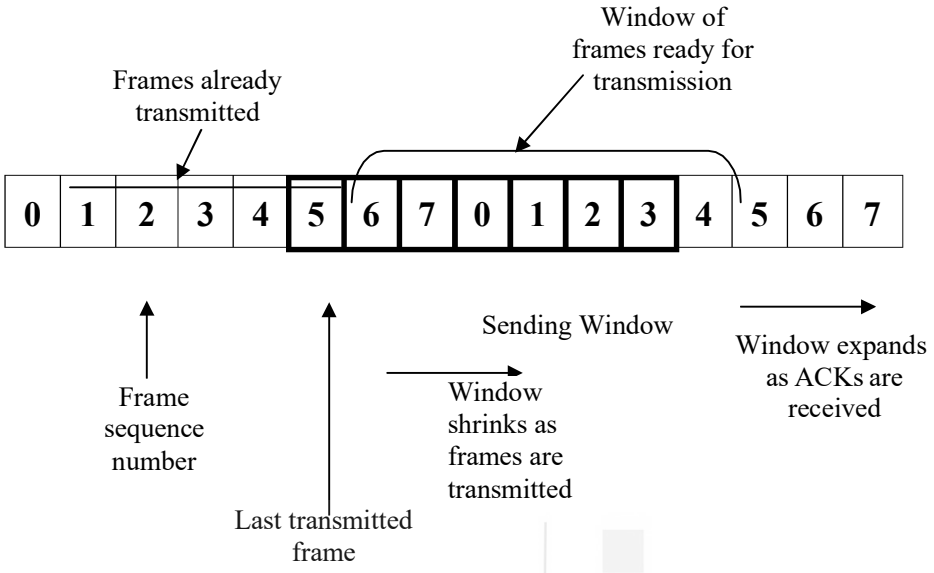


Figure 14: Sending window in a sliding window

Receiving Window

The receiver maintains a receiving window depending on the sequence numbers of frames that are accepted as shown in *Figure 15*.

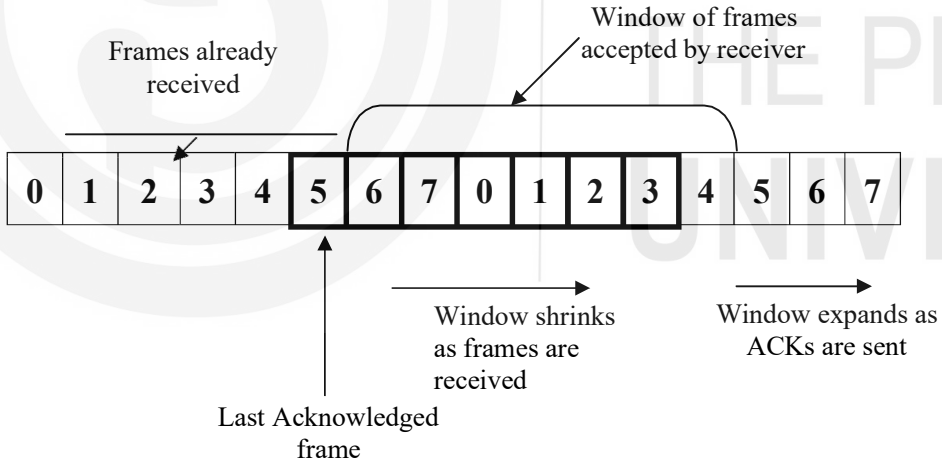


Figure 15: Receiving window in a sliding window

Functioning of Sliding Window

Flow control is achieved as the receiver can control the size of the sending window by limiting the size of the sending window. Similarly, data flow from the sender to the receiver can be limited, and that too can controls the size of receiving window as explained with the help of an example in *Figure 16.*

Example:

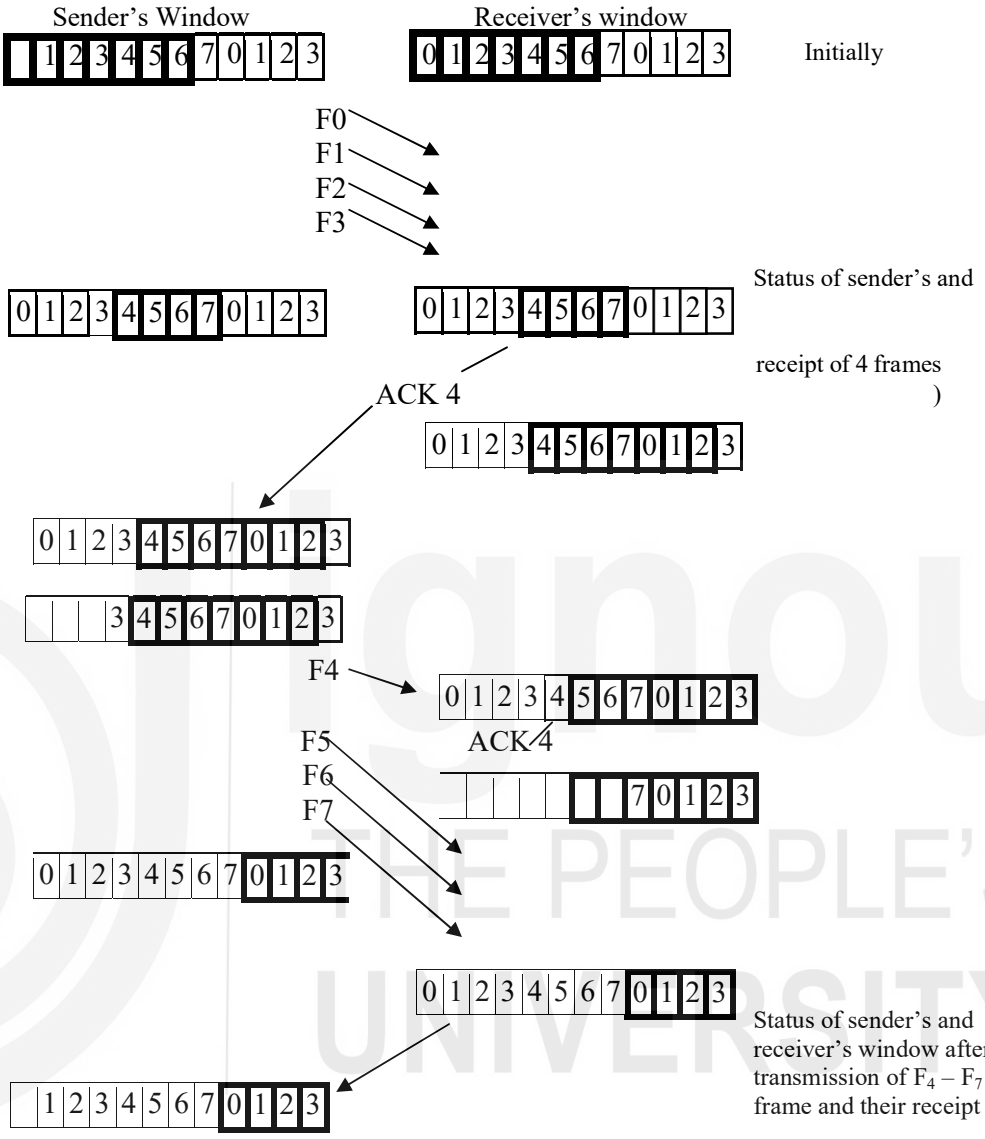


Figure 16: Function of a sliding window machine

Two types of errors are possible while transmission from source to destination takes place

1) Lost frame: Frame (data or control) fails to arrive.
2) Damaged frame: Frame is recognised, but some bits have been changed.

Two commonly used methods for sliding window are:

1) Go-back-n ARQ
2) Selective-repeat ARQ

Details of these two ARQ requests are discussed in the next unit.

☞ **Check Your Progress 5**

1) How does the sliding window protocol increase the utilisation of bandwidth?

……

……

……..

2) Define ARQ. How does ARQ facilitate the error control process?

……

……

……..

1.7 SUMMARY

This unit introduced the basic fundamental issues related to the Data Link Layer. The concept of framing and different methods of framing like Character Count, Character Stuffing, Bit stuffing and Framing by Illegal code has been focused up on. In the Data Link layer, data flow and error control is discussed. This error and flow control is required in the system for reliable delivery of data in the network. For the same, different types of error, different methods for error detection and correction are discussed. Among various methods, Block sum check method detects burst of error with high probability. CRC method is the one which detects the error with simplicity. Forward Error Correction is the error correction method that uses the parity concept. For flow control, stop and wait method tries to ensure that the speed of the sender and the receivers are matching to an extent. To overcome this sliding window protocol is introduced. Sliding window protocol can send many frames at one instance and that increases the efficiency of transmission channel. If some error occurs in the data then retransmission of the error frame is required and it is known as ARQ.

1.8 SOLUTIONS/ANSWERS

Check Your Progress 1

1) Character Count, Character Stuffing, Bit Stuffing and Framing by Illegal Code

2) 110001111101110000111111000

3) Only 1 bit used as stuffed bit in bit stuffing instead of character (8 bits) in character stuffing.

Check Your Progress 2

1) The extra bit added to the data bits to make number of 1's even or odd.
Even parity: The number of 1's in data bits including the parity bit should be even.

Odd Parity: The number of 1's in data bits including the parity bit should be odd.

2) Parity bit method is used for detecting single bit error. And CRC method is used to detect burst error.

3) Checksum is the value that is used for error detection. It is generated by adding data units using 1's compliment and then complementing the result (modulo 2 arithmetic).

Check Your Progress 3

1) A method that adds redundant bits to the data stream for error detection and correction is called Hamming Code.

2) $r_1 = 0$ $r_2 = 0$ $r_4 = 0$ $r_8 = 1$

Check Your Progress 4

1) CRC- 101

Check Your Progress 5

1) Sliding Window protocol increases the utilization of bandwidth as we can transmit many frames without waiting for an acknowledgement.

2) An error control method in which correction is made by retransmission of data by. ARQ facilitate error control process by using any of the following method:

   Stop and Wait ARQ
   Go-Back N ARQ
   Selective Repeat ARQ

1.9 FURTHER READINGS

1) *Computer Networks,* Andrew S. Tanenbaum, 4th Edition, Prentice Hall of India, New Delhi.

2) *Data and Computer Communications,* William Stalling, 5th Edition, Pearson Education, New Delhi.

3) *Data Communications and Networking,* Behrouz A. Forouzan, 3rd Edition, Tata McGraw Hill, New Delhi.

4) *Communication Networks– Fundamental Concepts and Key Architectures,* Leon Garcia and Widjaja, 3rd Edition, Tata McGraw Hill, New Delhi.

UNIT 2 RETRANSMISSION STRATEGIES

Structure **Page Nos.**

2.0 INTRODUCTION

In the previous unit we discovered that, Data Link Layer is responsible for ensuring that data is received at the receiver's end in line and error free. For the same task it has two important functions to perform that is Flow control and Error control. Flow and Error control are performed by the data link protocol. Before starting discussion on methods for flow and error control, firstly, we will define Flow control and Error control.

Flow control deals with when the sender should send the next frame and for how long the sender should wait for an acknowledgement. Data link protocol takes care of the amount of data that a sender can send and that a receiver can process as, the receiver has its own limitation in terms of speed for processing the frames. It also sees the compatibility of speed of both the sender and the receiver.

Error control deals with error detection and correction method that we have already discussed in the previous unit. If, an error is found in the frame either due to loss of frame or due to damage of frame, retransmission of the same is required by the sender. Retransmission is required when a sender does not receive a positive acknowledgement in time, due to a loss of frame or loss of acknowledgement or if, the sender receives negative acknowledgment from the receiver due to frame not been error free. This process of retransmission is called ARQ (Automatic Repeat Request). The set of rules that will determine the operations for the sender and the receiver are named the ARQ protocol. This ARQ protocol makes the network reliable, and that is, one of the important requirements of a network system if, data transmits from one node to another over the network and ensures that data received at receiver's site is complete and accurate. Here, we will refer to ACK, for positive acknowledgement (that is receiver has correct data) and NAK (REJect) to refer to negative acknowledgement (that is frame is received with some error). In this unit, you will study three commonly used methods for flow and error control that is Stop & Wait ARQ, GoBack-n ARQ and Selective Repeat ARQ.

2.1 OBJECTIVES

After going through this unit, you should be able to :

- define flow and error control;
- define is ARQ;

- define functionality of data link protocol;
- define Stop & Wait ARQ method for error and flow control;
- define GoBack-n ARQ method for error and flow control;
- perform selective Repeat ARQ method for error and flow control, and
- define pipelining.

2.2 STOP & WAIT ARQ

This is the simplest method for flow and error control. This protocol is based on the concept that, the sender will send a frame and wait for its acknowledgment. Until it receives an acknowledgment, the sender cannot send the next frame to the receiver. During transmission of frame over the network an error can appear.

Error can be due to a frame getting damaged/lost during transmission. Then, the receiver discards that frame by using error detection method. The sender will wait for acknowledgement of frame sent for a predetermined time (allotted time). If timeout occurs in the system then, the same frame is required to be retransmitted. Hence, the sender should maintain a duplicate copy of the last frame sent, as, in future it can be required for retransmission. This will facilitate the sender in retransmitting the lost/damaged frame.

At times the receiver receives the frame correctly, in time and sends the acknowledgment also, but the acknowledgment gets lost/damaged during transmission. For the sender it indicates time out and the demand for retransmission of the same frame appears in the network. If, the sender sends the last frame again, at the receiver's site, the frame would be duplicated. To overcome this problem it, follows a number mechanism and discards the duplicate frame.

Another situation when a error can appear in the network system is when the receiver receives the frame out of order, then it simply discards that frame and sends no acknowledgement. If, the acknowledgement is not received by the sender in time then, it assumes that the frame is lost during transmission, and retransmits it.

For distinguishing both data frame and acknowledgement frame, a number mechanism is used. For example A data frame 0 is acknowledged by acknowledgement frame 1, to show that the receiver has received data frame 0 and is expecting data frame1 from the sender.

Both the sender and the receiver both maintain control variable with volume 0 or 1 to get the status of recently sent or received. The sender maintains variable S that can hold 0 or 1 depending on recently sent frame 0 or 1. Similarly the receiver maintains variable R that holds 0 or 1 depending on the next frame expected 0 or 1.

Here, we are considering one directional information flow (Frame) and other direction control information (ACK) flow. If transmission of frames leads to an error then, the recovery process of the same demands, a retransmission strategy to be used that leads to four different possible outcomes, that are:

- Normal Operation
- When ACK is lost
- When frame is lost
- When ACK time out occurs

Normal Operation

If the sender is sending frame 0, then it will wait for ack 1 which will be transmitted by the receiver with the expectation of the next frame numbered frame1. As it receives ACK1 in time (allotted time) it will send frame 1. This process will be continuous till complete data transmission takes place. This will be successful transmission if ack for all frames sent is received in time. It is shown with the help of *Figure.1*

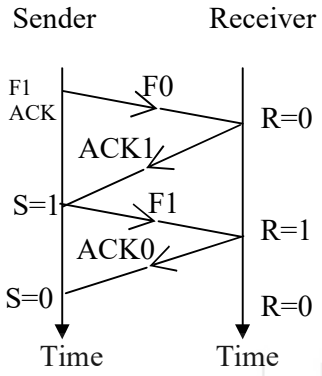


Figure 1: Stop and wait protocol

When ACK is lost

Here the sender will receive corrupted ACK1 for frame sent frame 0. It will simply discard corrupted ACK1 and as the time expires for this ACK it will retransmit frame 0. The receiver has already received frame 0 and is expecting frame1, hence, it will discard duplicate copy of frame 0. In this way the numbering mechanism solves the problem of duplicate copy of frames. Finally the receiver has only one correct copy of one frame. This is explained with the help of *Figure.2.*

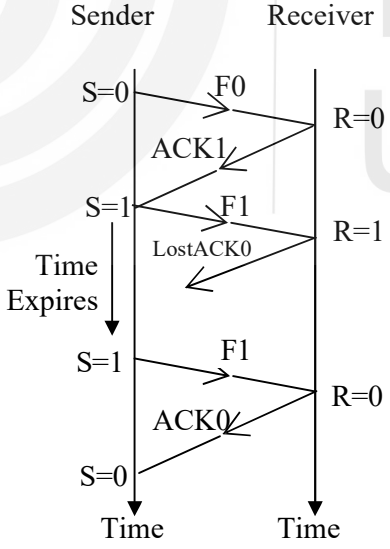


Figure 2: Loss of ACK

When Frame is lost

If the receiver receives corrupted/damaged frame1, it will simply discard it and assumes that the frame was lost on the way. And correspondingly, the sender will not get ACK0 as frame has not been received by the receiver. The sender will be in waiting stage for ACK0 till its time out occurs in the system. As soon as time out

occurs in the system, the sender will retransmit the same frame i.e frame1(F1) and the receiver will send ACK0 in reply as shown in *Figure.3.*

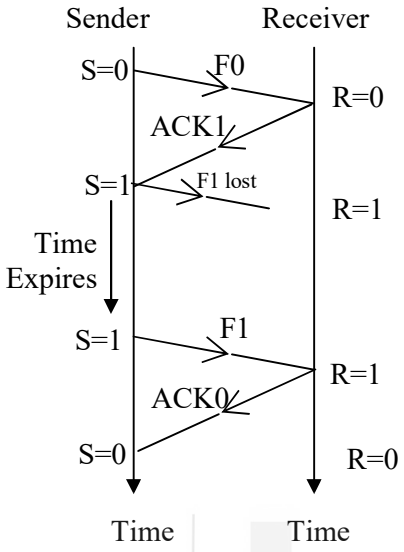


Figure 3: Loss of a frame

When ACK time out occurs

In this operation, the receiver is not able to send ACK1 for received frame0 in time, due to some problem at the receiver's end or network communication. The sender retransmits frame0 as ACK1 is not received in time. Then, the sender retransmits frame0 as ACK1 time expires. At the receiver end, the receiver discards this frame0 as the duplicate copy is expecting frame1 but sends the ACK1 once again corresponding to the copy received for frame0. At the sender's site, the duplicate copy of ACK1 is discarded as the sender has received ACK1 earlier as explained with the help of *Figure 4.*

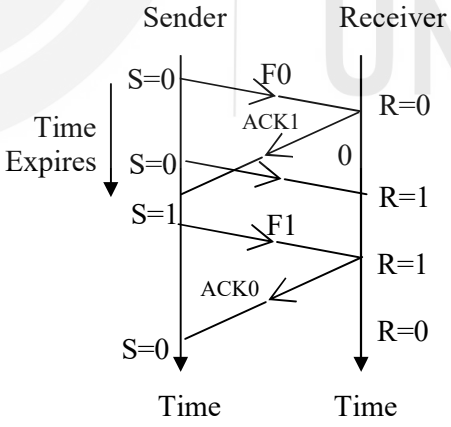


Figure 4: ACK time out

These operations indicate the importance of the numbering mechanism while, transmitting frames over the network. The method above discussed has low efficiency due to improper use of the communication channel. So, now, we will discuss the case if flow of data is bidirectional. In Bidirectional transmission both the sender and the receiver can send frames as well as acknowledgement. Hence, both will maintain S and R variables. To have an efficient use of bandwidth the ACK can be appended with the data frame during transmission. The process of combining ACK with data is

known as **Piggybacking.** The concept of piggybacking is explained in a later section. This reduces transmission overhead and increases the overall efficiency of data transmission.

The process is shown in *Figure*.5.

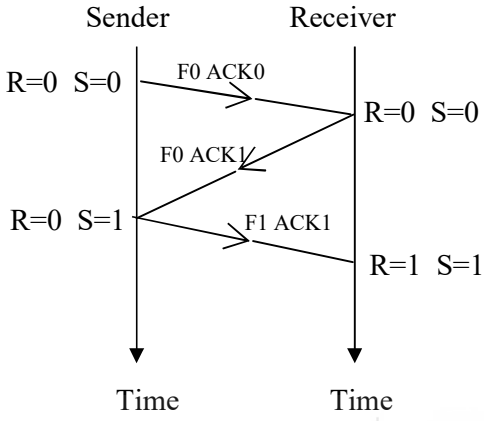


Figure 5: Piggybacking 1

The problem with stop and wait is that only one frame can be transmitted at a time and this leads to inefficiency of transmission. To deal with this, we have another error and flow control method that we will discuss in the next section.

2.3 GO-BACK-N ARQ

In this section, we will try to overcome the inefficient transmission that occurs in Stop & Wait ARQ. In this method, many frames can be transmitted during the process without waiting for acknowledgement. In this, we can send n frames without making the sender wait for acknowledgements. At the same time, the sender will maintain a copy of each sent frame till acknowledgement reaches it safely. As seen earlier in the Stop & Wait ARQ where, we used number mechanism, here also, we will use number method for transmission and receipt of frames. Each frame will have a sequence number that will be added with the frame. If, the frame can have k bit sequence number then the numbers will range between 0 to 2^k-1. For example if k is 2 bit then numbers will be 0,1,2,3,0,1,2,3,0,….

The sender can send 4 frames continuously without waiting for acknowledgment. But, the receiver will look forward to only one frame that must be in order. If, the frame received is not in order, it will simply keep on discarding the frame till, it receives the desired sequence number frame. The receiver is not bound to send an individual acknowledgment for all frames received; it can send a cumulative acknowledgment also. The receiver will send a positive acknowledgement if, the received frame is the desired sequence number frame. Otherwise, it will keep on waiting if, the frame received is corrupted or out of order. As soon as the timer expires for the frame sent by the sender, the sender will GoBack and retransmit all frames including the frame for which the timer expired, till, the last sent frame. Hence, it is named as Go-Back-N ARQ. The process of retransmission for an error frame is shown in *Figure.6.*Go-Back-N is used in HDLC protocol.

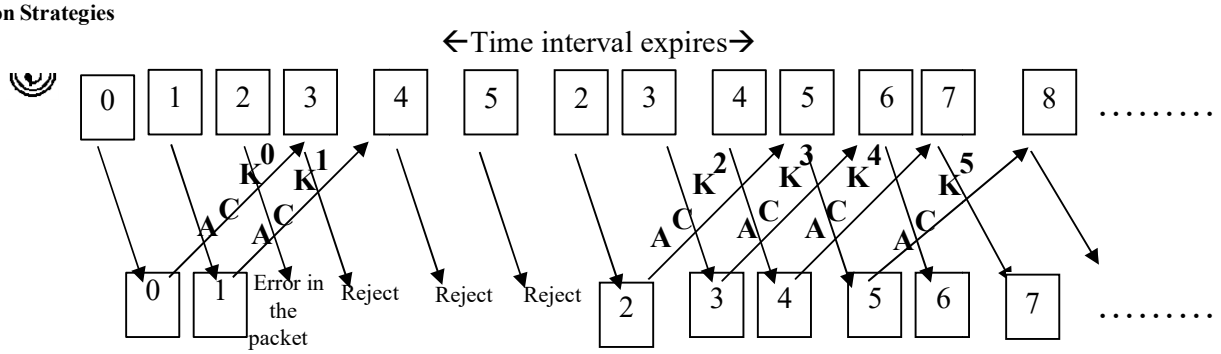


Figure 6: Go-Back-N

If, the error rate is high, then, this leads to a lot of wastage of bandwidth as the sender will retransmit all the frames from, the frame in which error appears till the last sent. To increase the efficiency of transmission, when the error rate is high, another protocol called Selective Repeat ARQ is used which is discussed in the next section.

2.4 SELECTIVE REPEAT ARQ

This method increases the efficiency of the use of bandwidth. In this method, the receiver has a window with the buffer that can hold multiple frames sent by the sender. The sender will retransmit only that frame which has some error and not all the frames as in Go-Back-N ARQ. Hence, it is named as selective repeat ARQ. Here, the size of the sender and the receiver window will be same. The receiver will not look forward only to one frame as in Go-Back-N ARQ but it will look forward to a continuous range of frames. The receiver also sends NAK for the frame which had the error and required to be retransmitted by the sender before the time out event fires. As in the earlier section we discussed, each frame will have a sequence number that will be added with the frame. If, the frame can have k bit sequence number then the sequence number of frames will range between 0 to 2^k-1. For example, if k is 2 bit then numbers will be 0,1,2,3,0,1,2,3,0,….Size of sender and receiver window would be $2^k/2$ i.e 4/2=2 or it can be written as 2^{k-1}. If the window size is 2 and acknowledgement for frame0 and frame1 both gets lost during transmission then, after timer expires the sender will retransmit frame0 though the receiver is expecting frame2 after frame1 was received without any error. Hence, frame0 will be discarded by the receiver as a duplicate frame. If the receiver window size is more than two, the receiver will accept duplicate frame0 as a new frame and hence, the size of window should be $2^k/2$. The process is shown in *Figure.7.*

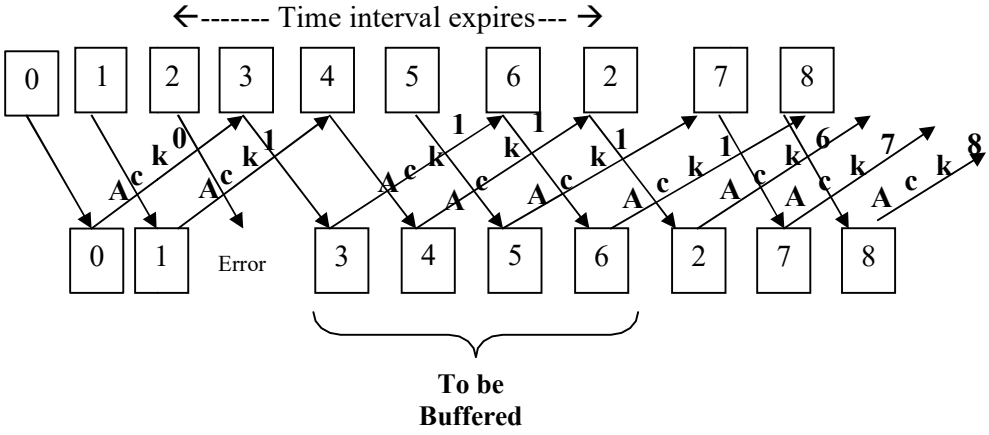


Figure 7: Selective Repeat

TCP uses Selective repeat ARQ strategy to deal with flow and error control

If, we consider bidirectional transmission i.e., data and acknowledgement flow from both sender and receiver then, the concept of Piggybacking can be used in a similar fashion as already discussed in Stop & Wait ARQ method, in order to better utilise bandwidth.

2.5 PIPELINING

Pipelining in the network is one task that starts before the previous one is completed. We might also say that the number of tasks is buffered in line, to be processed and this is called pipelining. For example, while printing, through the printer before one task printing is over we can give commands for printing second task. Stop & Wait ARQ does not use pipelining. As in Stop & Wait ARQ the sender cannot send the next frame till it receives acknowledgement for the frame sent. Here, Pipelining is used in Go-Back-N ARQ and Selective repeat ARQ as both methods can send multiple frames without holding the sender for receiving the acknowledgement for frame sent earlier. This process of pipelining improves the efficiency of bandwidth utilisation. Now, we will explain with the help of *Figure 8* how pipelining is used in Go-Back-N ARQ.

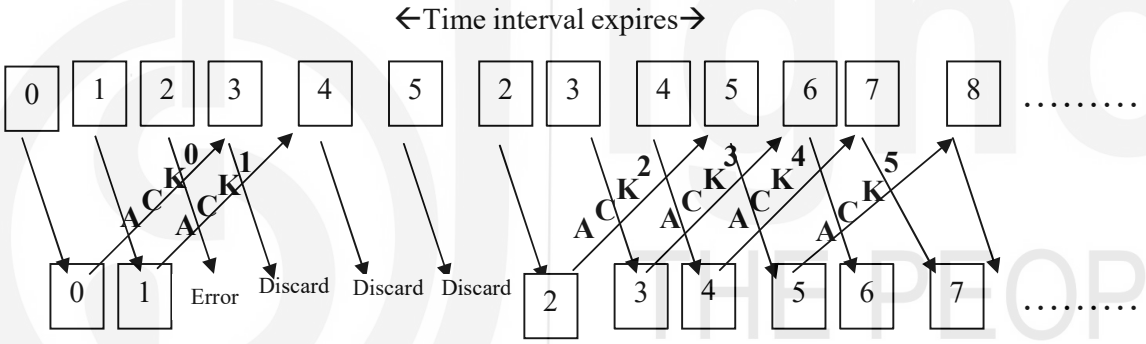


Figure 8: Pipelining in Go-Back-N

Here frame0 is received by the receiver and without waiting for acknowledgment of frame 0 sent at sender site, the sender is allowed to send frame1. This process is known as pipelining.

Similarly, selective repeat pipelining is used as shown in *Figure 9* below.

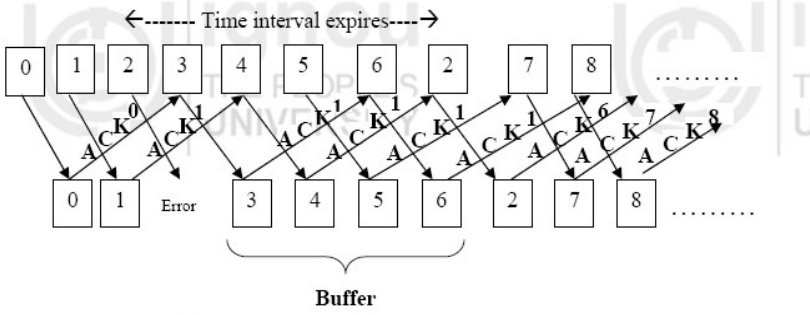


Figure 9: Pipelining in selective repeat

In this example we can show how pipelining increases the overall utilisation of bandwidth. Frame0 and frame1 are sent in continuous order without making the sender wait for acknowledgment for frame0 first.

☞ Check Your Progress 1

1) Define flow and error control.

……

……

……

2) Explain Go-Back-N ARQ and Selective repeat ARQ are better methods for retransmission.

……

……

……

3) Give an example where pipelining can be applied.

……

……

……

4) What is the significance of control variables S and R?

……

……

……

2.6 PIGGYBACKING

Piggybacking is the process which appends acknowledgement of frame with the data frame. Piggybacking process can be used if Sender and Receiver both have some data to transmit. This will increase the overall efficiency of transmission. While data is to be transmitted by both sender and receiver, both will maintain control variable S and R. We will explain the process of piggybacking when the sender and the receiver both are transmitting data with the help of *Figure 10.*

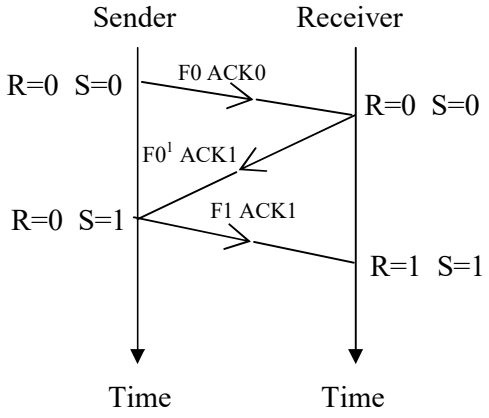


Figure 10: Piggybacking

Here, both the sender and the receiver maintain control variables S and R. The sender sends frame 0 (F0) with ACK0 appended along with it. Similarly, the receiver sends Frame 0(F0) with ACK1 appended to it. This way transmitting both frame and acknowledgement will concurrently increase optimal efficiency of bandwidth utilisation because piggybacking will get a free side.

2.7 SUMMARY

This unit focuses on one prime function of the Data link layer that is flow and error control for achieving the goal of reliable data transmission over the network. For flow and error control retransmission strategies are considered. Flow control specifically talks about the speed of sending the frame by the sender and processing the received frame by the receiver. The speed for the sender and the receiver must be compatible. So, that all frames can be received in order and processed in time. Error control technique combines two processes error detection and error correction in the data frame. In Stop & wait ARQ protocol sender waits for acknowledgment for the last frame sent. After the acknowledgment is received by the sender then only the next frame can be sent. In Go-Back-N ARQ frames can be sent continuously without waiting for the sender to send the acknowledgement. If an error is found in any frame then frames received after that will be discarded by the receiver. Retransmission of frames will start from the error frame itself. In selective repeat Process frames can be sent continuously. But here, the receiver has a buffer window that can hold the frames received after the error frame. Hence, retransmission will be only for error frame. This increases the efficiency of data transmission on a network. For better utilisation of bandwidth, piggybacking can be used. Piggybacking process appends acknowledgement along with the data frame. This will be efficient for bidirectional communication.

2.8 SOLUTIONS/ANSWERS

Check Your Progress 1

1) Flow Control: A method to control rate of flow of frames .
Error Control: A method to detect and handle the errors during data transmission.

2) Go-Back-N ARQ and Selective repeat ARQ are better method for retransmission as they keep the copy of each sent frame till the frame is acknowledged by receiver. Go-Back-N ARQ retransmits all frames in which there is an error and the following frames. Selective repeat ARQ retransmit only the error frame.

3) Pipelining is used in Go-Back-N ARQ and Selective repeat ARQ because many frames can be sent without waiting for previous frame acknowledgement.

4) Numbering system that is used by sender and receiver by using variable S and R that can hold 0 or 1 removes the occurrence of possible ambiguities like either duplicate frame or duplicate acknowledgement.

2.9 FURTHER READINGS

1) *Computer Networks,* Andrew S. Tanenbaum, 4th Edition, Prentice Hall of India, New Delhi.

2) *Data and Computer Communications,* William Stalling, 5th Edition,Prentice Hall of India, New Delhi.

3) *Data Communications and Networking,* Behrouz A. Forouzan, 3rd Edition, Tata McGraw Hill, New Delhi.

4) *Communication Networks, Fundamental Concepts and Key Architectures,* Leon Garcia and Widjaja, 3rd Edition, Tata McGraw Hill, New Delhi.

UNIT 3 CONTENTION-BASED MEDIA ACCESS PROTOCOLS

Structure **Page Nos.**

3.0 INTRODUCTION

As discussed in first unit of this block, the Data Link Layer (DLL) is divided into two sub layers i.e., the Media Access Control (MAC) layer and the Logical Link Control (LLC) layer. In a network nodes are connected to or use a common transmission media. Based on the connection of nodes, a network can be divided into two categories, that is, point-to-point link and broadcast link. In this unit, we will discuss, broadcast link and their protocols. If, we talk about broadcast network then, a control process for solving the problem of accessing a multi access channel is required. Many protocols are available for solving the problem of multi-access channel. These protocols can control an access on shared link as in broadcast network. It is an important issue to be taken into consideration that is, how to who gets access to the channel while, many nodes are in competition as shown in *Figure 1*.

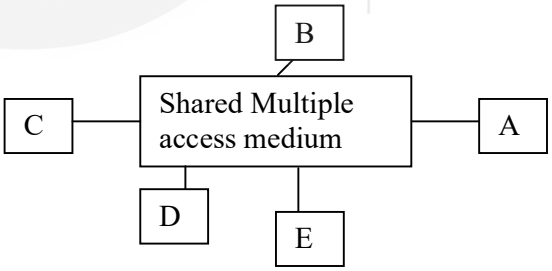


Figure 1: Shared media

The protocol which decides who will get access to the channel and who will go next on the channel belongs to MAC sub-layer of DLL. Channel allocation is categorised into two, based on the allocation of broadcast among competeting users that is Static channel allocation problem and Dynamic Channel allocation problem as shown in *Figure 2*. In this unit, we will also discuss whether some access conflict or collision comes in the network, and how to deal with these conflicts. This is an important issue for LAN.

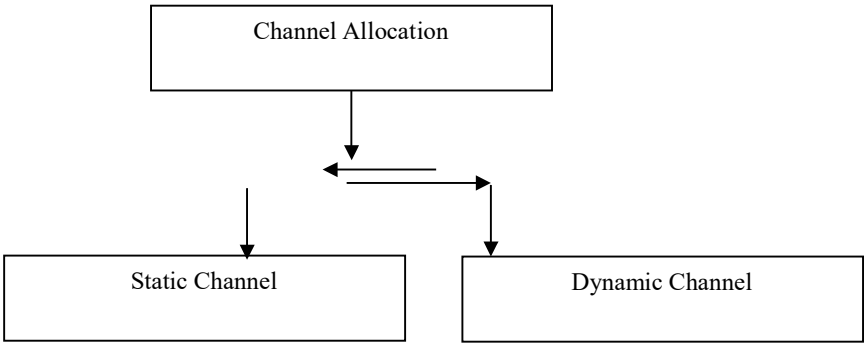


Figure 2: Channel allocation technique

Here the transmission of frames can occupy the medium or any arbitrary time or in slotted time intervals (time is divided into slots). When the transmission station senses whether the channel is busy or free, this is called *carrier sensing.*

3.1 OBJECTIVES

After going through this unit, you should be able to learn:

- the need for accessing multi-access channel;
- common methods for accessing multi-access channel like FDM, TDM;
- the need for Dynamic channel allocation method;
- pure ALOHA method for channel allocation;
- slotted ALOHA method for channel allocation;
- carrier sensing method CSMA to improve performance;
- carrier sensing with collision detection method CSMA/CD;
- IEEE 802.3 standard and their Different Cabling types, and
- basics of Giga Bit Ethernet.

3.2 ADVANTAGES OF MULTIPLE ACCESS SHARING OF CHANNEL RESOURCES

MAC sub layer's primary function is to manage the allocation of one broadcast channel among N competing users. For the same, many methods are available such as static, and dynamic allocation method.

In the static channel allocation method, allocating a single channel among N competing users can be either FDM (Frequency division multiplexing) or TDM (Time division multiplexing). In FDM the total bandwidth will be divided into N equal parts for N users. This way, every user will have their own frequency band so no conflict or collision will occur among user in the network. But, this is feasible only when the number of users are small and traffic is also limited. If, the number of users becomes large this system has face many problems like either one user is gets one frequency band that is not used at all or the other user does not get a frequency band for transmission. Hence, it is simple and efficient for a small number of users. Similarly, in TDM (Time Division Multiplexing), discussed with first Block every user will get a fixed N^{th} time slot.

In the dynamic channel allocation the important issues to be considered are whether,

time is continuous or discrete or whether the station is carrier sensing large number of stations each with small and bursty traffic.

Many methods are available for multiple access channel like ALOHA, CSMA etc. that we will discuss in the following section.

3.3 PURE ALOHA

As we have discussed earlier in the previous unit, if, one node sends a frame to another node, there can be some error in the frame. For the same we discussed some retransmission strategies to deal with the error. But, in case of allocating a single channel among N uncoordinated competing users, then the probability of collision will be high. Station accesses the channel and when their frames are ready. This is called random access. In an ALOHA network one station will work as the central controller and the other station will be connected to the central station. If, any of stations want to transmit data among themselves, then, the station sends the data first to the central station, which broadcast it to all the stations.

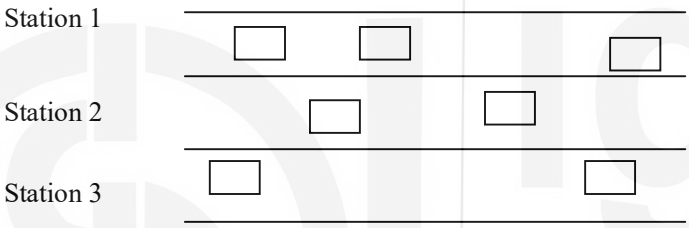


Figure 3: ALOHA

Here, the medium is shared between the stations. So, if two stations transmit a frame at overlapping time then, collision will occur in the system. Here, no station is constrained, any station that has data /frame to transmit can transmit at any time. Once one station sends a frame (when it receives its own frame and assumes that the destination has received it) after 2 times the maximum propagation time. If the sender station does not receive the its own frame during this time limit then, it retransmit this frame by using backoff algorithm that we will discuss later on. And if, after a number of repeats if it does receive own pocket then the station gives up and stops retransmitting the same frame.

Let R be the bit rate of the transmission channel and L be the length of the frame. Here, we are assuming that the size of frame will be constant and hence, it will take constant time t= L/R for transmission of each packet.

As in the case of Pure ALOHA protocol frames can be sent any time so, the probability of collision will be very high. Hence, to present a frame from colliding, no other frame should be sent within its transmission time. We will explain this with the help of the concept of vulnerable period as shown in *Figure 4*. Let a frame is that transmitted at time t_0 and t be the time required for its transmission. If, any other station sends a frame between t_0 and t_0+t then the end of the frame will collide with that earlier sent frame. Similarly, if any other station transmits a frame between the time interval t_0+t and t_0+2t again, it will result in a garbage frame due to collision with the reference frame. Hence, 2t is the vulnerable interval for the frame. In case a frame

meets with collision that frame is retransmitted after a random delay.

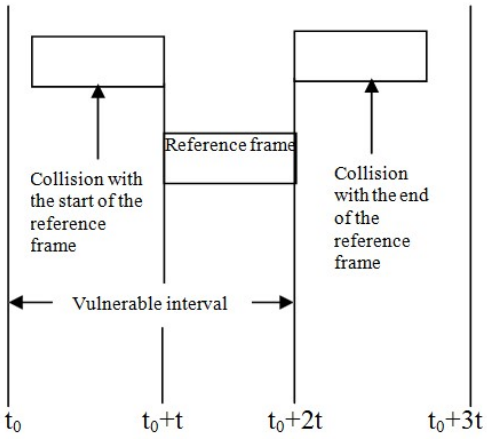


Figure 4: Vulnerable Period

Hence, for the probability of successful transmission, no additional frame should be transmitted in the vulnerable interval 2t.

To find the probability of no collision with a reference a frame, we assume that a number of users are generating new frames according to Poisons distribution. Let S be the arrival rate of new frames per frame time. As we find probability of no collision, S will represent the throughput of the system. Let G be the total arrival rate of frames including retransmission frames (also called load of the system). For finding the probability of transmission from the new and retransmitted frame. It is assumed that frames arrival is Poisson distributed with an average number of arrivals of G frames/ frame time. The probability of k frames transmission in 2t seconds is given by the Poisson distribution as follows:

The throughput of the system S is equal to total arrival rate G times the probability of successful transmission with no collision,

That is $S = G * P$

$S=G * P$ (zero frame transmission in the vulnerable interval i.e.,2t seconds)

Since

$$P[K \text{ frame in vulnerable interval } 2t] = \frac{(2G)\, e^{-2G}}{K!}, \quad K = 0, 1, 2,3$$

Thus

$P[K = 0 \text{ in } 2t] = -2G$

Hence, $S = G * P = G \cdot e^{-2G}$

Note that the averages load is G. Hence it is 2G in 2t

$S=G * e^{-2G}$

The relationship between S vs. G can be shown in *Figure 5*.

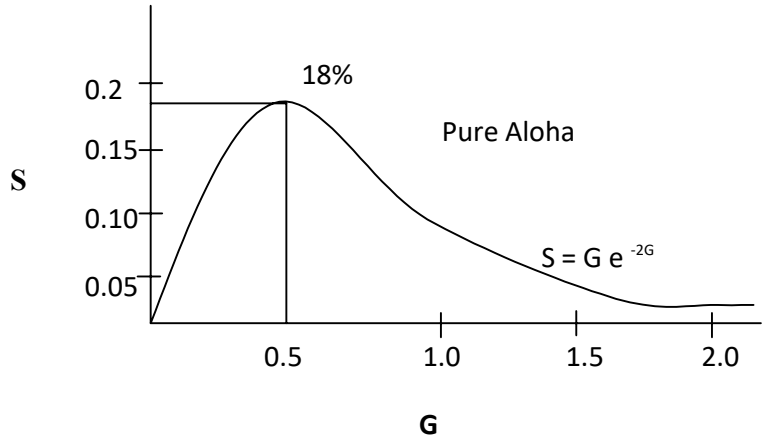


Figure 5: Throughput vs. load graph of pure ALOHA

As G is increasing, S is also increasing for small values of G. At G=1/2, S attains its peak value i.e., S=1/2e i.e., 0.18(approx). After that, it starts decreasing for increasing values of G. Here, the average number of successful transmission attempts/frames can be given as $G/S = e^{2G}$.

An average number of unsuccessful transmission attempts/frame is
$G/S - 1 = e^{2G} - 1$.

By this, we know that the performance of ALOHA is not good as unsuccessful transmission are increasing exponentially with load G. So, we will discuss Slotted ALOHA in the next section to see how performance can be improved.

3.4 SLOTTED ALOHA

In this, we can improve the performance by reducing the probability of collision. In the slotted ALOHA stations are allowed to transmit frames in slots only. If more than one station transmit in the same slot, it will lead to collision This reduces the occurrence of collision in the network system. Here, every station has to maintain the record of time slot. The process of transmission will be initiated by any station at the beginning of the time slot only. Here also, frames are assumed to be of constant length and with the same transmission time. Here the frame will collide with the reference frame only if, it arrives in the interval t_0-t to t_0. Hence, here the vulnerable period is reduced that is to t seconds long.

The throughput of the system S is equal to the total arrival rate G times the probability of successful transmission with no collision

That is $S = G * P$
$S=G * P$ (zero frame transmission in t seconds)

The probability of k frames transmission in t seconds and is given by the Poisson distribution as follows:

$P[k]= (G)^k * e^{-G}/k!$, k=0,1,2,3……

Here average load in the vulnerable interval is G (one frame time) Hence, the probability of zero frames in t seconds = e^{-G}

$S= G * e^{-G}$

The relationship between S vs G can be shown in *Figure 6*.

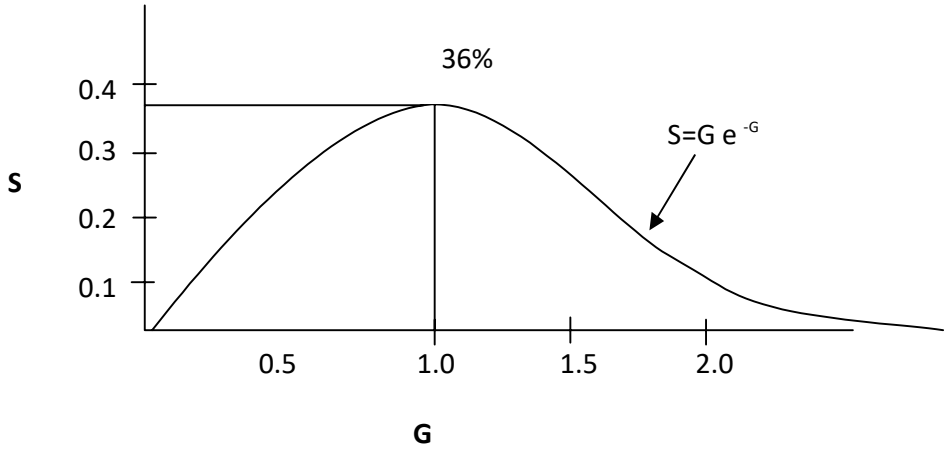


Figure 6: Throughput vs. load graph of slotted ALOHA

From the figure we can see that the system is exhibiting its performance. Maximum throughput that can be achieved with Slotted ALOHA S=1/e= 36 % (Approx.)

However, with this performance also we are not able to utilise the medium in an efficient manner. Due to the high rate of collision systems the bandwidth is which was designed to implement random access in LANs. So, we will discuss a new protocol called CSMA in the next section.

3.5 CARRIER SENSE MULTIPLE ACCESS (CSMA)

As we have seen in previous section, the Slotted ALOHA maximum throughput that can be achieved is 1/e only, though, the stations do not keep track of what the other station is doing or what's going on in the medium. Then also, many frames meet and collide. So in LANs we will observe the behavior of other station as well and try to reduce the number of collision to achieve better throughput of the network. To achieve maximum throughput here, we will try to restrict transmission that will cause collision by sensing whether the medium has some data or not. Protocols in which station senses the channel before starting transmission are in the category of CSMA protocols (also known as listen before talk protocols).

CSMA have many variants available that are to be adapted according to the behaviour of the station that has frames to be transmitted when the channel is busy or that some transmission is going on. The following are some versions of CSMA protocols:

- 1-Persistent CSMA
- Non-Persistent CSMA
- p-Persistent CSMA

1-Persistent CSMA

In this protocol a station i.e., who wants to transmit some frame will sense the channel

first, if it is found busy than that some transmission is going on the medium, then, this station will continuously keep sensing that the channel. And as soon as this station finds that the channel has become idle it will transmit its frame. But if more than one station is in waiting state and keeps track of the channel then a collision will occur in the system because both waiting station will transmit their frames at the same time.
The other possibility of collision can be if the frame has not reached any other station then, it indicates to the second station that the channel is free. So the second station also starts its transmission and that will lead to collision. Thus 1-persistent CSMA a greedy protocol as to capture the channel as soon as it finds it idle. And, hence, it has a high frequency of collision in the system. In case of collision, the station senses the channel again after random delay.

Non-Persistent CSMA

To reduce the frequency of the occurrence of collision in the system then, another version of CSMA that is non-persistent CSMA can be used. Here, the station who has frames to transmit first sense whether the channel is busy or free. If the station finds that channel to be free it simply transmits its frame. Otherwise, it will wait for a random amount of time and repeat the process after that time span is over. As it does not continuously senses the channel to be, it is less greedy in comparison of 1-Persistent CSMA. It reduces the probability of the occurrence of collision as the waiting stations will not transmit their frames at the same time because the stations are waiting for a random amount of time, before restarting the process. Random time may be different for different stations so, the likelihood waiting station will start their transmission at the same time is reduced. But, it can lead to longer delays than the 1-Persistent CSMA.

p-Persistent CSMA

This category of CSMA combines features of the above versions of CSMA that is 1-persistent CSMA and non-persistent CSMA. This version is applicable for the slotted channel. The station that has frames to transmit, senses the channel and if found free then simply transmits the frame with p probability and with probability 1-p it, defers the process. If the channel is found busy then, the station persists sensing the channel until it became idle. Here value of p is the controlling parameter.

After studying the behaviour of throughput vs load for persistent CSMA it is found that Non-Persistent CSMA has maximum throughput. But we can using collision detection mechanism improve upon this to achieve more throughput in the system using collision defection mechanism and for the same we will discuss CSMA/CD in the next section.

3.6 CSMA WITH COLLISION DETECTION (CSMA/CD)

As before here also any transmission in the system needs to sense the channel to see whether it is busy or free. The stations ensure that the transmission will start only when it finds that the channel is idle. In CSMA/CD the station aborts the process of transmission as soon as they detect some collision in the system. If two stations sense that the channel is free at the same time, then, both start transmission process immediately. And after that, both stations get information that collision has occurred in the system. Here, after the station detecting the collision, the system aborts the process of transmission. In this way, time is saved and utilisation of bandwidth is

optimised. This protocol is known as CSMA/CD and, this scheme is commonly used in LANs. Now, we will discuss the basic operation of CSMA/CD. Let, t be the maximum transmission time between two extreme ends of a network system (LAN). At t_0 station A, at one extreme end of the LAN begins the process of transmitting a frame F_A. This frame reaches the station E which at another extreme end of the same network system in t propagation delay away. If no other station in between has started its frame transmission, it implies that A has captured the channel successfully. But, in case E_F station E starts its frame transmission just before the arrival of frame from station A frame then, collision will take place. Station A will get the signal of collision after 2t time. Hence, 2t time is required to ensure that station A has captured the channel successfully as shown with the help of *Figure 8*.

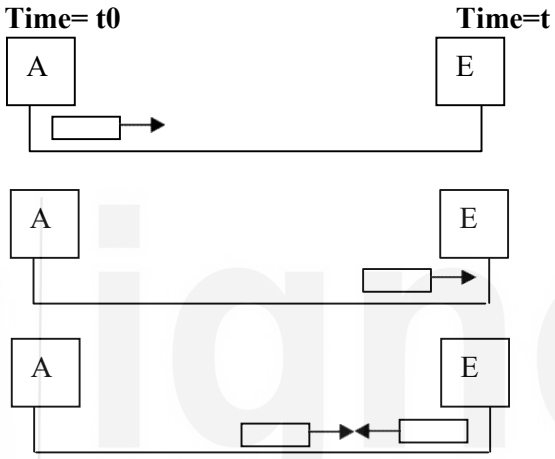


Figure 8: Collision detection

Collision of frames will be detected by looking at the strength of electric pulse or signal received after collision. After a station detects a collision, it aborts the transmission process and waits for some random amount of time and tries the transmission again with the assumption that no other station has started its transmission in the interval of propagation time. And hence, in CSMA/CD the channel can be any of the following three states as it can be shown with the *Figure 9*.

- Transmission of frame is going on.
- Idle slot.
- Contention period/slot.

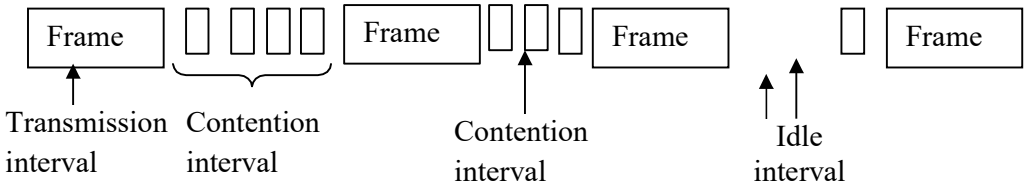


Figure 9: Transmission states

In CSMA/CD a station with a frame ready to begin transmission senses the channel and starts transmission if it finds that the channel is idle. Otherwise, if it finds it busy, the station can persist and use backoff algorithm which will be discussed in the next paragraph.

Backoff Algorithm

With the help of backoff algorithm we will see how the randomisation process occurs as soon as collision detection takes place. Time is divided into discrete slots with the length of worst case propagation time (propagation time between two extreme ends of LAN) 2t. After the first collision in the system, each station waits for 0 or 1 slot time before trying transmission for the next time. If, two stations that collide is select the same random number then collision will be repeated. After the second collision, the station will select 0,1,2 or 3 randomly and wait for these many number of slots. If, the process of collision will occur repeatedly, then, the random number interval would be between 0 and 2^i-1 for i^{th} collision and this number of slots will be the waiting time for the station. This algorithm is known as the binary exponential algorithm.

☞ **Check Your Progress 1**

1) Why is DLL divided into two sub layers? What are the key functions of those sub layers?

……

……

……

2) How does Slotted ALOHA improve the performance of the system over Pure ALOHA?

……

……

……

3) How has non-persistent reduced the probability of collision?

……

……

……

4) Explain Back off Algorithm and give one example of where it is used.

……

……

……

3.7 ETHERNET FRAME FORMAT (IEEE 802.3)

Ethernet protocol is a MAC sublayer protocol. Ethernet stands for cable and IEEE 802.3 Ethernet protocol was designed to operate at 10 Mbps. Here, we will begin discussing the Ethernet with various types of cabling. With the help of *Figure 9*, we will try to summarise the cabling used for baseband 802.3 LANs.

Characteristic\Cable	10Base5	10Base2	10BaseT	10BaseF
Medium	Thick Coaxial Cable	Thin Coaxial Cable	Twisted Pair	Optical Fiber
Maximum Length of segment	500 m	200 m	100 m	2 Km
Topology	Bus	Bus	Star	Star
Advantages	Used for connecting workstation with tap on the cable	Low cost	Existing environment can use Hub and connect the stations	Good noise immunity and good to use

Figure 9: Characteristics ethernet cable

IEEE 802.3 Ethernet accesses the channel using 1-persistent CSMA/CD method in LAN. Now we will discuss MAC frame structure for IEEE 802.3 with the help of *Figure 10*.

Preamble	Start Delimiter of frame	Destination Address	Source Address	Length of Data Field	Data	Pad	Frame Check Sum

Figure 10: Ethernet Frame Format

Each frame has seven fields explained as follows:

Preamble: The first field of 802.3 frame is 7 byte (56 bits) long with a sequence of alternate 1 and 0 i.e., 10101010. This pattern helps the receiver to synchronise and get the beginning of the frame.
Starting Delimiter (SD): The second field start delimiter is 1 byte (8 bit) long. It has pattern 10101011. Again, it is to indicate the beginning of the frame and ensure that the next field will be a destination address. Address, here, can be a single address or a group address.

Destination Address (DA): This field is 6 byte (48 bit) long. It contains the physical address of the receiver.

Source Address (SA): This filed is also 6 byte (48 bit) long. It contains the physical address of the sender.

Length of Data Field: It is 2 byte (16 bit) long. It indicates the number of bytes in the information field. The longest allowable value can be 1518 bytes.

Data: This field size will be a minimum of 46 bytes long and a maximum of 1500 bytes as will be explained later.

Pad: This field size can be 0 to 46 bytes long. This is required if, the data size is less than 46 bytes as a 802.3 frame must be at least 64 bytes long.

Frame Checksum (FCS): This field is 4 bytes (32 bit) long. It contains information about error detection. Here it is CRC-32.

Minimum and Maximum Length of Frame

Minimum frame length = 64 bytes = 512 bits

Maximum frame length= 1518 bytes = 12144 bits

Minimum length or lower limit for frame length is defined for normal operation of CSMA/CD. This is required so that, the entire frame is not transmitted completely before its first bit has been received by the receiver. If, this happens then the probability of the occurrence of collision will be high (the same has been explained earlier in the previous section CSMA/CD).

Hence, Ethernet frame must be of 64 bytes long. Some of the bytes are header and trailer parts of the frame. If, we consider 6 bytes destination address, 6 bytes source address, 2 bytes length and 4 bytes FCS (6+6+2+4=18) then, the minimum length of data will be 64-18= 46 bytes. If, frame is less than 46 bytes then, padding bits fill up this difference.

As per 802.3 standard, the frames maximum length or upper limit of frame is = 1518 bytes (excluding preamble and SD). If we subtract 18 bytes of header and trailer then, the maximum length will be 1500 bytes.

3.8 SUMMARY

In some networks, if a single channel and many users use that channel, then, allocation strategy is required for the channel. We have discussed FDM and TDM allocation method. They are the simplest methods for allocation. They work efficiently for a small number of user. For a large number of users the ALOHA protocol is considered. There are two versions of ALOHA that is Pure ALOHA and Slotted ALOHA. In Pure ALOHA no slotting was done but the efficiency was poor. In Slotted ALOHA, slots have been made, so that every frame transmission starts at the beginning of the slot and throughput is increased by a factor of 2. For avoiding collision and to increase efficiency in sensing the channel, CSMA is used. Many versions of CSMA are persistent and non-persistent. In CSMA/CD collision detection process is added so that process can be aborted just after a collision is detected.
Ethernet is a commonly used protocol for LAN. IEEE 802.3 Ethernet uses 1 persistent CSMA/CD access method.

3.9 SOLUTIONS/ANSWERS

Check Your Progress 1

1) DLL is divided into two sub layers LLC and MAC as IEEE has defined LLC for standard LANs and MAC for avoiding the conflict and collision on a network to access to the medium at any time. LLC does error flow control and MAC deals with channel allocation problem.

2) Slotted ALOHA follows synchronization for transmitting the frames thatreduces the probability of collision and hence improves the efficiency.

3) In non persistent strategy, station waits for random amount of time after sensing the collision on multiple access channels. Hence are stations attempts for retransmission after random time that reduces the probability of collision.

4) After a collision is sensed by the channel, time is divided up into discrete slots. For example, if first collision identified then each station waits for either 0 or 1 slot time. Similarly, if third collision occurs then random interval will be 0 to 7

and for i^{th} collision random number interval will be 0 to 2^i-1. Subsequently, these many numbers of slots will be skipped before retransmission. This is called as binary exponential back off algorithm. CSMA/CD uses back off algorithm.

3.10 FURTHER READINGS

1) *Computer Networks,* Andrew S. Tanenbaum, 4th Edition, Prentice Hall of India, New Delhi.

2) *Data and Computer Communications,* William Stalling, 5th Edition, Pearson Education, New Delhi.

3) *Data Communications and Networking,* Behrouz A. Forouzan, 3rd Edition,Tata McGraw Hill, New Delhi.

4) *Communication Networks– Fundamental Concepts and Key Architectures,* Leon Garcia and Widjaja 3rd Edition, Tata McGraw Hill, New Delhi.

UNIT 4 WIRELESS LAN AND DATALINK LAYER SWITCHING

Structure **Page Nos.**

4.0 INTORDUCTION

The Previous discussion, we had in this block was related to wired LANs but recently, wireless LANs are taking a dominant position due to coverage of location difficult to wire, to satisfy the requirement of mobility and adhoc networking. In a few years from now, we will notice a broader range of wireless devices accessing the Internet, such as digital cameras, automobiles, security systems, kitchen appliances. **KUROSE** and **ROSS** [reference] writes that some day wireless devices that communicate with the Internet may be present everywhere: on walls, in our cars, in our bedrooms, in our pockets and in our bodies.

In this unit, we cover two broad topics: Wireless LAN, its protocols, its standard and Data Link Layer Switching. In organisation we need an interconnection mechanism so that all nodes can talk to each other. Bridges and switches are used for this purpose. The spanning tree algorithms are used to build plugs and act as bridges.

4.1 OBJECTIVES

After going through this unit, you should be able to:

- understand what is a wireless LAN;
- describe the various LAN Protocols;
- understand the IEEE 802.11 Standard, and
- describe the operation of bridges (transparent, spanning tree and remote bridges).

4.2 INTRODUCTION TO WIRELESS LAN

As the number of mobile computing and communication devices grows, so does the demand to connect them to the outside world. A system of notebook computers that communicate by radio can be regarded as a wireless LAN. These LANs have different properties than conventional LANs and require special MAC sublayer protocols. The 802.11b standard defines the physical layer and media access sublayer for wireless local area network.

To understand the concept, we will take a simplistic view that, all radio transmitters have some fixed range. When a receiver is within a range of two active transmitters, the resulting signal will generally be garbled and of no use. It is important to realise that in some wireless LANs, not all stations are within the range of one another, which leads to a variety of complications, which we will discuss in the next sections.

Wireless LANs are commonly being used in academic institutions, companies, hospitals and homes to access the internet and information from the LAN server while on roaming.

4.3 WIRELESS LAN ARCHITECTURE(IEEE 802.11)

The wireless LAN is based on a cellular architecture where the system is subdivided into cells as shown in *Figure1*. Each cell (called basic service set or BSS, in the 802.11) is controlled by a base station (called access point or AP). Wireless LAN may be formed by a single cell, with a single access point (it can also work within an AP), most stations will be formed by several cells, where the APs are connected through some kind of backbone (called distribution system or DS). This backbone may be the Ethernet and in some cases, it can be the wireless system. The DS appears to upper-level protocols (for example, IP) as a single 802 network, in much the same way as a bridge in wire 802.3. The Ethernet network appears as a single 802 network to upper-layer protocols.

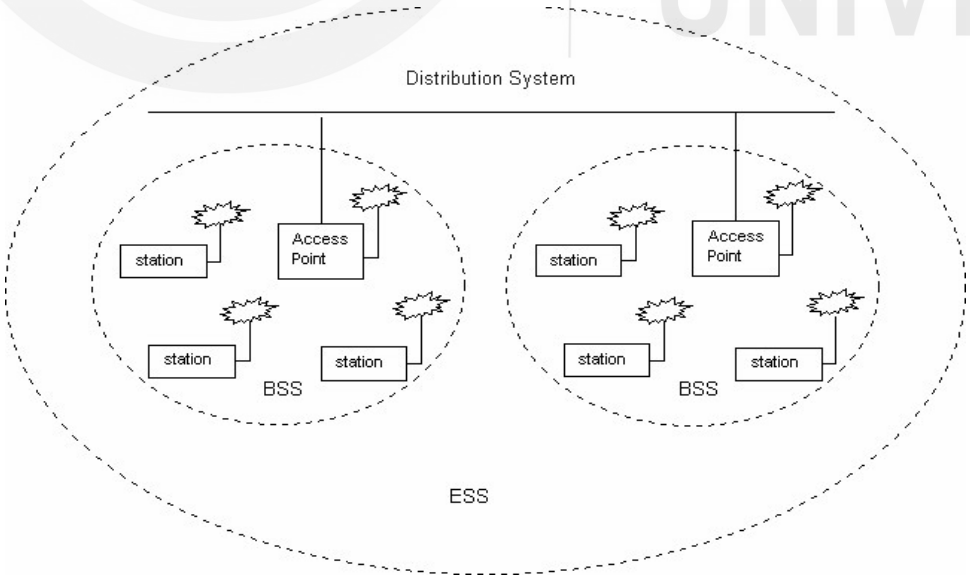


Figure 1: Wireless LAN architecture

Stations can also group themselves together to form an *ad hoc* network: a network with no central control and with no connections to the "outside world." Here, the network is formed "on the fly," simply because, there happens to be mobile devices that have found themselves in proximity to each other, that have a need to communicate, and that find no preexisting network infrastructure. An *ad hoc* network might be formed when people with laptops meet together (for example, in a conference room, a train, or a car) and want to exchange data in the absence of a centralised AP. There has been tremendous interest in *ad hoc* networking, as communications between devices continue to proliferate.

4.4 HIDDEN STATION AND EXPOSED STATION PROBLEMS

There are two fundamental problems associated with a wireless network. Assume that there are four nodes A, B, C and D. B and C are in the radio range of each other. Similarly A and B are in the radio range of each other. But C is not in the radio range of A.

Now, suppose that there is a transmission going on between A and B (*Figure 2 (a)*). If C also wants to transmit to B, first, it will sense the medium but will not listen to A's transmission to B because, A is outside its range. Thus, C will create garbage for the frame coming from A if, it transmits to B. This is called the **hidden station problem.** The problem of a station not being able to detect another node because that node is too far away is called hidden station problem.

Now, let us consider the reverse situation called the **exposed station problem.** (*Figure 2 (b).*)

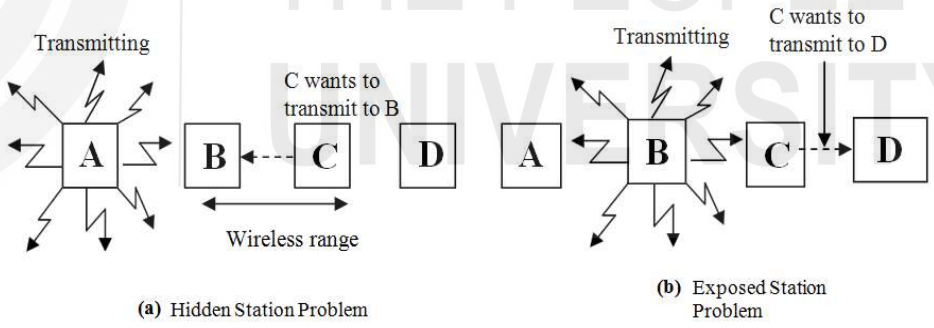


Figure 2: Hidden and exposed station problem

In this case, B is transmitting to A. Both are within radio range of each other. Now C wants to transmit to D. As usual, it senses the channel and hears an ongoing transmission and falsely concludes that it cannot transmit to D. But the fact is transmission between C and D would not have caused any problems because, the intended receivers C and D are in a different range. This is called exposed station problem.

4.5 WIRELESS LAN PROTOCOLS: MACA AND MACAW

In this section, we, will describe two wireless LAN protocols: MACA and MACAW. MACA is the oldest protocol of the two. MACA was proposed as an alternative to CSMA Protocol which has certain drawbacks:

First, it senses the channel to see if the channel is free, it transmits a packet, otherwise it waits for a random amount of time.

- Hidden Station Problems leading to frequent collision.
- Exposed terminal problems leading to worse bandwidth utilisation. MACA eliminates the hidden and exposed station problems using RTS (Repeat to Send) and CTS (Clear to Send) handshake mechanism, which is explained below through a *Figure 3*. RTS and CTS packets carry the expected duration of the data transmission, which will have some implications. All nodes near the sender/receiver after hearing RTS/CTS will defer the transmission. Therefore, it avoids some cases of hidden and exposed station problems. However, it does not always avoid these problems. If, the neighbour hears the RTS only, it is freeto transmit once the waiting interval is over.

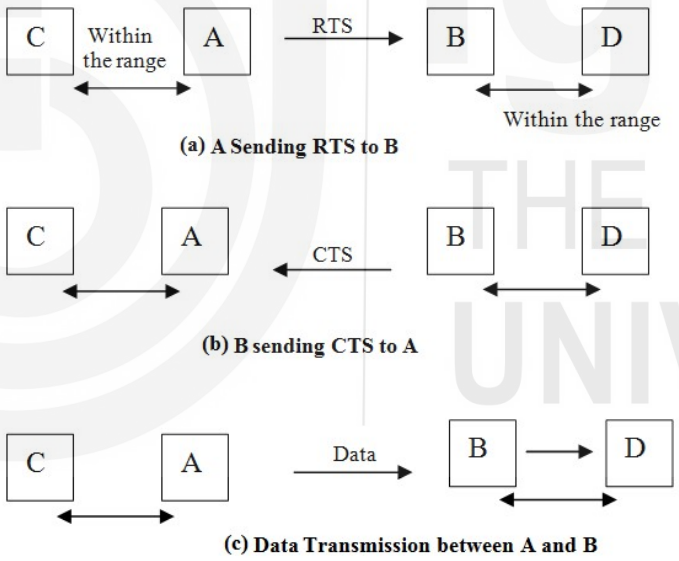


Figure 3: MACA Protocol

Just assume that, there are four nodes A, B, C, and D in a wireless LAN (*Figure 3*). A is a sender and B is a receiver. The station C is within range of A but not within range of B. Therefore, it can hear transmission from A (i.e., RTS) but not transmission from B (i.e., CTS) (*Figure 3(a) and 3(b)*). Therefore, it must remain silent long enough for the CTS to be transmitted back to A without conflict. The station D is within the range of B but not A so it hears CTS but not RTS. Therefore, it must remain silent during the upcoming data transmission, whose length it can tell by examining the CTS frame. This is illustrated through a diagram (*Figure 3(a) and 3(c)*) A sends RTS to B. Then B sends CTS to A. Then, follows data between A and B.

C 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, the station 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.

Despite these precautions, collisions can still occur. For example, B and C could both send RTS frames to A at the same time. These frames will collide and will 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. MACAW (MACA for wireless) extends MACA to improve its performance which will have the following handshaking mechanism. RTS-CTS-DS-Data → ACK. It is also illustrated through the following diagram (Figure 4).

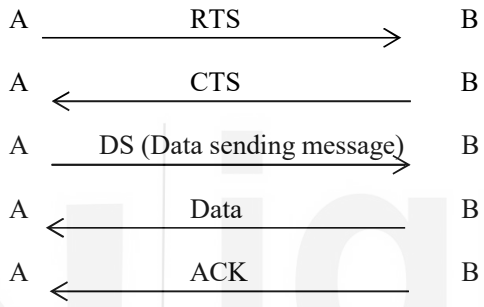


Figure 4: The MACAW protocol

MACAW extends MACA with the following features:

- **Data link layer acknowledgement:** It was noticed that without data link layer acknowledgements, lost frames were not retransmitted until the transport layer noticed their absences, much later. They solved this problem by introducing anACK frame after each successful data frame.
- **Addition of carrier sensing:** They also observed that CSMA has some use, namely, to keep a station from transmitting a RTS while at the same time another nearby station is also doing so to the same destination, so carrier sensing was added.
- **An improved backoff mechanism:** It runs the backoff algorithm separately for each data stream (source-destination pair), rather than for each station. This change improves the fairness of the protocol.
- **DS (Data sending) message:** Say a neighbour of the sender overhears an RTS but not a CTS from a receiver. In this case it can tell if RTS-CTS was successful or not. When it overhears DS, it realises that the RTS-CTS was successful and itdefers its own transmission.

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

☞ Check Your Progress 1

1) What is Hidden Station Problem?

……

……

……..

2) Why CSMA/CD cannot be used in wireless LAN environment? Discuss.

……

……

……..……..……..……..……

3) How is MACAW different from MACA?

……

……

……..……..……..……..……

4.6 IEEE 802.11 PROTOCOL STACK

IEEE 802.11 standard for wireless LAN is similar in most respects to the IEEE 802.3 Ethernet standard addresses. As shown in the *Figure 5* the physical layer of 802.11 corresponds to the OSI physical layer fairly well but, the data link layer in all the 802 protocols is split into two or more sublayers. In 802.11, the MAC (Medium Access Control) sublayer determines how the channel is allocated, that is, who gets to transmit next. Above it is the LLC (Logical Link Control) sublayer, whose job is to hide the differences between the different 802 variants and make them indistinguishable as far as the network layer is concerned.

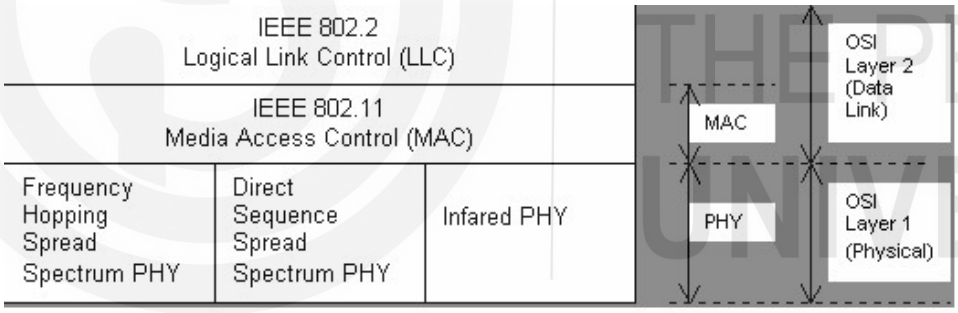


Figure 5: Part of the 802.11 protocol stack

4.6.1 The 802.11 Physical Layer

The 802.11 physical layer (PHY) standard specifies three transmission techniques allowed in the physical layer. The infrared method used much the same technology as television remote controls do. The other two use short-range radio frequency (RF) using techniques known as FHSS (Frequency Hopped Spread Spectrum) and DSSS (Direct Sequence Spread Spectrum). Both these techniques, use a part of the spectrum that does not require licensing (the 2.4 –GHz ISM band). Radio-controlled garage door openers also use the same piece of the spectrum, so your notebook computer may find itself in competition with your garage door. Cordless telephones and microwave ovens also use this band. All of these techniques operate at 1 or 2 Mbps and at low enough power that they do not conflict too much. RF is capable of being used for 'not line of sight' and longer distances.

The other two short range radio frequency techniques are known as spread spectrum. It was initially developed for military and intelligence requirement. The essential idea is to spread information signal over a wider bandwidth to make jamming and interception more difficult. The spread spectrum is ideal for data communication because, it is less susceptible to radio noise and creates little interference. It is used to comply with the regulations for use with ISM Band.

There are two types of spread spectrum:

(i) Frequency Hopping (FH), and
(ii) Direct Sequence (DS)

Both these techniques are used in wireless data network products as well as other communication application such as, a cordless telephone please refer to [5] for further studies.

Under this scheme, the signal is broadcast over a seemingly random data sequence RF hopping from frequency to frequency at split second intervals. A receiver hopping between frequencies in synchronisation with the transmitter, picks up the message.
Using FHSS (Frequency Hopped Spread Spectrum) the 2.4 GHz is divided into 75 MHz Channel. In this scheme, a pseudorandom number generator is used to produce the sequence of the frequencies hopped to. As long as all stations use the same seed to the pseudorandom number generator and stay synchronised in time, they will hop to the same frequencies simultaneously. FHSS' randomisation provides a fair way to allocate spectrum in the unregulated ISM band. It also provides some sort of security. Because an intruder does not know the hopping sequence it cannot eavesdrop on transmissions. Over longer distance, multipath fading can be an issue, and FHSS offers good resistance to it. It is also relatively insensitive to radio interference, which makes it popular for building-to-building links. Its main disadvantage is its low bandwidth. FHSS allows for a less complex radio design than DSSS but FHSS is limited to 2 Mbps data transfer rate due to FCC regulations that restrict subchannel bandwidth to 1 MHz causing many hops which means a high amount of hopping overhead. The DSSS is a better choice for WLAN application. It is also restricted to 1 or 2 Mbps.

DSSS divides 2.4 GHz band into 14 channels. Channels using at the same location should be separated 25 MHz from each other to avoid interference. FHSS and DHSS are fundamentally different signaling techniques and are not capable of interoperating with each other. Under this scheme, each but in the original signal is represented by multiple bits in the transmitted signal, which is known as chipping code. The chipping code spreads the signal across a wider frequency band in direct proportion to the number of bits used. Therefore, a 10 bit chopping code spreads signal across a frequency band that is 10 times greater than 1 bit chipping code (Ref. 3).

4.6.2 The 802.11 MAC Sub-layer Protocol

After the discussion on the physical layer it is time to switch over to the IEEE 802.11 MAC sublayer protocols which are quite different from that of the Ethernet due, to the inherent complexity of the wireless environment compared to that of a wired system. With Ethernet (IEEE 802.3) a node transmits, in case, it has sensed that the channel is free. If, it does not receive a noise burst back within the first 64 bytes, the frame has almost assuredly been delivered correctly. With wireless technology, this situation

does not hold.

As mentioned earlier 802.11 does not use CSMA/CD due to the following problems:

(i) The Hidden Station Problem: CSMA does not avoid the hidden station problem (*Figure (a) & (b)*)

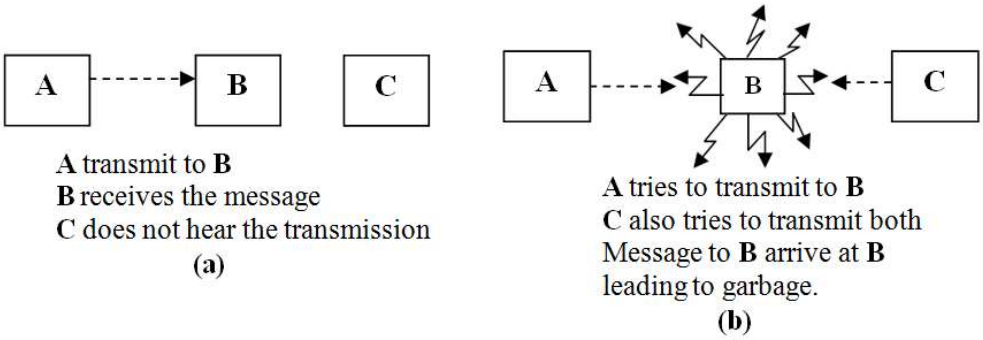
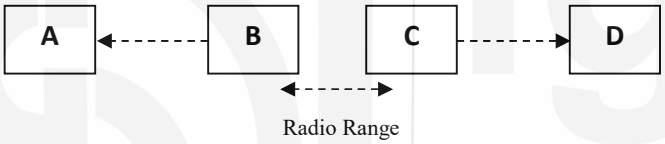


Figure 6: (a) The hidden station problem

(ii) The exposed Station Problem CSMA may cause nodes to unnecessarily refrainfrom accessing the channel as shown in the *Figure below:*



B Transmits to **A** which is heard by **C**
C unnecessarily avoids sending a message to **D** even though there would be no collision.

Figure 6: (b) The exposed station problem

(iii) In addition, most radios are half duplex, meaning that they cannot transmit and listen for noise bursts at the same time on a single frequency as Ethernet does.

To deal with this problem, 802.11 supports two modes of operation.

- DCF (Distributed Coordination Function)
- PCF (Point Coordinated Function) (optional)

Now, we, will examine IEEE 802.11 DCF separately. It does not use any central control. In this respect it is similar to Ethernet.

When DCF is employed, 802.11 uses a protocol called CSMA/CA (CSMA with Collision Avoidance). In this protocol, both **physical channel sensing** and **virtual channel sensing** are used. Two methods of operation are supported by CSMA/CA. In the first method (Physical sensing), before the transmission, it senses the channel. If the channel is sensed idle, it just wants and then transmitting. But it does not sense the channel while transmitting but, emits its entire frame, which may well be destroyed at the receiver's end due to interference there. If, the channel is busy, the sender defers transmission until it goes idle and then starts transmitting. If, a collision occurs, the

colliding stations wait for a random time, using the Ethernet binary exponential backoff algorithm and then try again later.

The second mode of CSMA/CA operation is based on MACAW and uses virtual **channel sensing**, as illustrated in *Figure 6*. In this example, there are four stations A,

B, C, and D. A is a transmitter and B is a receiver, C is within the radio range of A (and possibly within the range of B) where as D is within the range of B but not within range of A.

When A has to send data to B, it begins by sending an RTS frame to B to request permission to send it a frame. When B receives this request, it may decide to grant permission, in which case it sends the

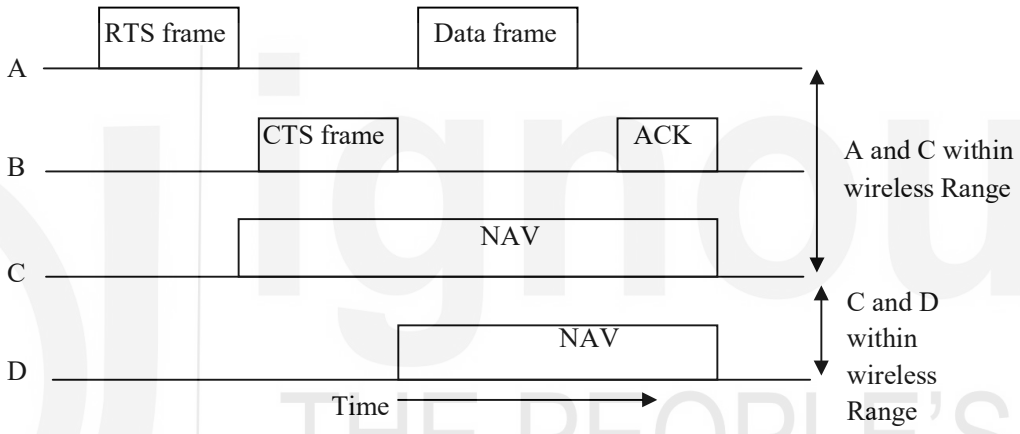


Figure 7: The use of virtual channel sensing using CSMA/CA

CTS frame back. Upon receipt of the CTS, A now sends its frame and starts an ACK timer. Upon correct receipt of the data frame, B responds with an ACK frame leading to the closure of data transfer operation between A & B. In case A's ACK timer expires before the ACK gets back to it, the whole protocol is run again.

Now, how will C and D nodes react to it? Node C is within range of A, so it may receive the RTS frame. C may receive the RTS frame because it is in the rage of A. From the information in the RTS frame it estimates how long the transfer will take, including the final ACK and asserting a kind of virtual channel busy for itself, indicated by NAV (Network Allocation Vector) as shown in *Figure 7.* Similarly, D also asserts the NAV signal for itself because it hears the CTS. The NAV signals are not for transmission. They are just internal reminders to keep quite for a certain period of time.

In contrast to wired networks, wireless networks are noisy and unreliable.
To deal with the problem of noisy channels, 802.11 allows frames to be fragmented into smaller pieces, each with its own checksum because, if a frame is too long, it has very little chance of getting through undamaged and will, probably have to be retransmitted. The fragments are individually numbered and acknowledged using a stop and wait protocol at LLC (i.e., the sender may not transmit fragment k+1 until it

has received the acknowledgement for fragment k). Once the channel has been acquired using RTS and CTS, multiple fragments can be sent in a row, as shown in Fig. 8. A sequence of fragments is called a **fragment burst.**

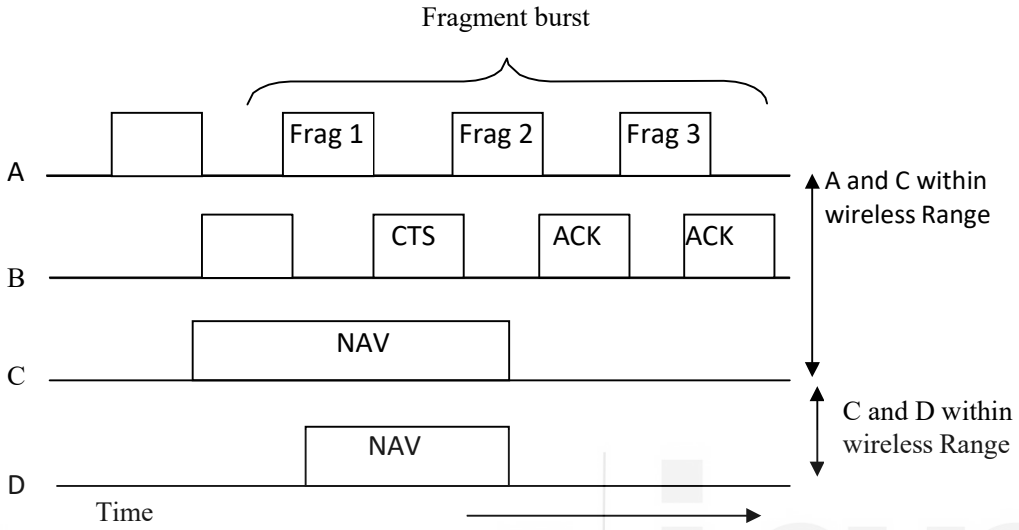


Figure 8: A fragment burst

The second advantage of fragmentation is that it increases the channel throughput by not allowing retransmission to the bad fragments rather than the entire frame. The size of C fragment can be adjusted by a base station in a cell. A base station in a cell can adjust the size of C fragment. The NAV mechanism keeps other stations quiet, only until the next acknowledgement, but another mechanism (described below) is used to allow a whole fragment burst to be sent without interference.

So, far we have discussed DCF in which, the base station polls the other stations, asking them if they have any frames to send. Since, transmission order is completely controlled by the base station in PCF mode, no collisions ever occurs. The standard prescribes the mechanism for polling, but not the polling frequency, polling order, or even whether all stations need to get equal service.

The basic mechanism is for the base station to broadcast a beacon frame periodically (10 to 100 times per second). The beacon frame contains system parameters, such as hopping sequences and dwell times (for FHSS), clock synchronisation, etc. It also invites new stations to sign up for polling services. Once a station has signed up for polling service at a certain rate, it is effectively guaranteed a certain fraction of the bandwidth, thus making it possible to give guarantee of quality services.

Battery life is always an issue with mobile wireless devices, so 802.11 pays attention to the issue of power management. In particular, the base station can direct a mobile station to go into sleep state until explicitly awakened by the base station or the user. Having told a station to go to sleep, however, means that the base station has the responsibility for buffering any frames directed at it while the mobile station is asleep. These can be collected later.

PCF and DCF can coexist within one cell. At first it might seem impossible to have central control and distributed control operating at the same time, but 802.11 provides a way to achieve this goal. It works by carefully defining the interframe time interval. After a frame has been sent, a certain amount of dead time is required before any station can send a frame.

4.7 SWITCHING AT DATA LINK LAYER

Before discussing data link layer switching devices, let us talk about repeaters which are layer 1 devices. Repeaters provide both physical and electrical connections. Their functions are to regenerating and propagate a signal in a channel. Repeaters are used to extend the length of the LAN which depends upon the type of medium. For example 10 mbps 802.3 LAN that uses UTP cable (10 BASE-T) has a maximum

restriction of 100 meters. Many organisations have multiple LANs and wish to connect them. LANs can be connected by devices called **bridges,** which operate as the data link layer. Unlike repeaters, bridges connect networks that have different physical layers. It can also connect networks using either the same or different types of architecture at the MAC. (Token ring, FDDI, Ethernet etc).

Bridges have some other characteristics:

(i) Store and forward device.
(ii) Highly susceptible to broadcast storms.

Bridges are store and forward devices to provide error detection. They capture an entire frame before deciding whether to filter or forward the frame, which provides a high level of error detection because a frame's CRC checksum can be calculated by the bridge. Bridge are highly succeptable to broadcast storms. A broadcast storm occurs when several broadcasts are transmitted at the same time. It can take up huge bandwidth.

Before looking at the technology of bridges, it is worthwhile taking a look at some common situations in which bridges are used. **Tanenbaun** [Ref.1] has six reasons why a single organisation may end up with multiple LANs.

1) **Multiple LANs in organisation:** 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. But there is a need for interaction,so bridges are needed.

2) **Geographical difference:** The organisation 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 later link themto run a single cable over the entire site.

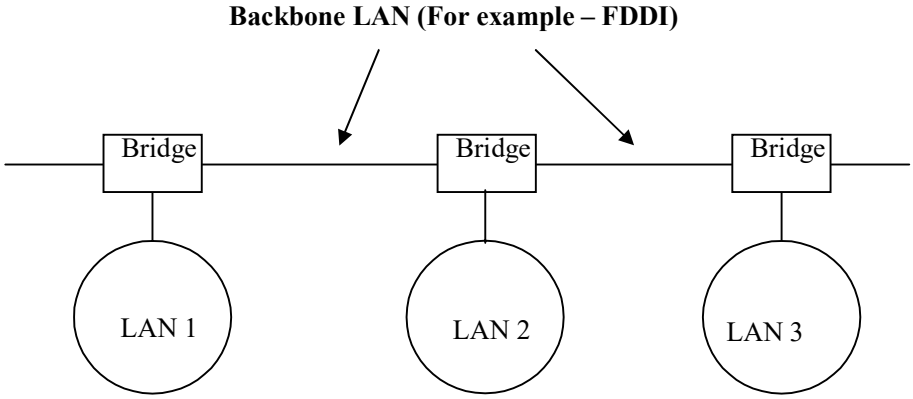


Figure 9: Multiple LANs connected by a backbone to handle a total load higher than thecapacity of single LAN

3) **Load distribution**: It may be necessary to split what is logically a single LANinto separate LANs to accommodate the load.

4) **Long Round Trip delay:** In some situations, a single LAN would be adequate in terms of the load, but the physical distance between the most distant machines is can be 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.

5) **Reliability:** By inserting bridges at critical points reliability can be enforced in the network by isolating a defective node. Unlike a repeater, which just copies whatever it sees, a bridge can be programmed to exercise some discretion regarding what forwards and what it does not forward.

6) **Security** is a very important feature in bridges today, and can control the movement of sensitive traffic by isolating the different parts of the network.

In an ideal sense, a bridge should have the following characters:

(i) **Fully transparent:** It means that it should allow the movement of a machine from one cable segment to another cable segment without change of hardware and software or configuration tools.

(ii) **Interpretability:** It should allow a machine on one LAN segment to talk to another machine on another LAN segment.

4.7.1 Operation of Bridges in Different LAN Environment

Having studied the features of bridges and why we require multiple LANs, let's learn how they work. Assume that, there are two machines A and B. Both of them are attached to a different LANs. Machine A is on a wireless LAN (Ethernet IEEE 802.3). While B is on both LANs are connected to each other through a bridge. Now, A has a packet to be sent to B. The flow of information at Host A is shown below:

1) The packet at machine A arrives at the LLC sublayer from the application layerthrough the transport layer and network layer.

2) LLC Sublayer header get attached to the packet.

3) Then it moves to the MAC Sublayer. The packet gets MAC sublayer header forattachment.

4) Since the node is part of a wireless LAN, the packet goes to the air using GRF.

5) The packet is then picked up by the base station. It examines its destination address and figures that it should be forwarded to the fixed LAN (it is Ethernetin our case).

6) When the packet arrives at the bridge which connects the wireless LAN and Ethernet LAN, it starts at the physical layer of the bridge and moves to its LLClayer. At the MAC sublayer its 802.11 header is removed.

7) The packet arrives at the LLC of a bridge without any 802.11 header.

8) Since the packet has to go to 802.3 LAN, the bridge prepares packets accordingly.

Note that a bridge connecting *k* different LANs will have *K* different MAC sublayers and *k* different physical layers. One for each type.

So far we have presented a very simplistic scenario in forwarding a packet from one LAN to another through a bridge. 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 due to focus on the following reasons:

1) **Different frame Format :**To start with, each of the LANs uses a different frame format. Unlike the differences between Ethernet, token bus, and token ring, which were due to history and big corporate egos, here the differences are to some extent legitimate. For example, the *Duration* field in 802.11 is there, due to the MACAW protocol and that makes no sense in Ethernet. As a result, any copying between different LANs requires reformatting, which takes CPU time, requires a new checksum calculation, and introduces the possibility of undetected errors due to bad bits in the bridge's memory.

2) **Different data rates**: When forwarding a frame 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. Therefore, it has to be buffered. For example, if a gigabit Ethernet is pouring bits into an 11-Mbps 802.11 LAN at top speed, the bridge will have to buffer them, hoping not to run out of memory.

3) **Different frame lengths:** An obvious problem arises when a long frame must be forwarded onto a LAN that cannot accept it. This is the most serious problem. The problem comes when a long frames arrives. An obvious solution is that the frame must be split but, such a facility is not available at the data link layer. Therefore the solution is that such frames must be discarded. Basically, there is no solution to this problem.

4) **Security:** Both 802.11 and 802.16 support encryption in the data link layer, but the Ethernet does not do so. This means that the various encryption services available to the wireless networks are lost when traffic passes over the Ethernet.

5) **Quality of service:** The Ethernet has no concept of quality of service, so traffic from other LANs will lose its quality of service when passing over an Ethernet.

4.7.2 Transparent Bridges

In, the previous section, we dealt with the problems encountered in connecting two different IEEE 802 LANs via a single bridge. However, in large organisations with many LANs, just interconnecting them all raises a variety of issues, even if they are all just Ethernet. In this section, we introduce a type of bridge called Transparent Bridge, which is a plug and play unit which you connect to your network and switch it on. There is no requirement of hardware and software changes, no setting of address switches , no downloading of routing tables, just plug and play. Furthermore, the operation of existing LANs would not be affected by the bridges at all. In other words, the bridges would be completely transparent (invisible to all the hardware and software). Operating in a promiscuous mode, a transparent bridge captures every frame that is transmitted in all the networks to which the bridge is connected. The bridge examines every frame it receives and extracts each frame's source address, which it adds to the backward address table.

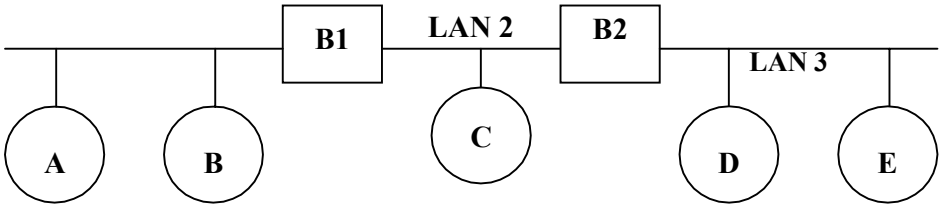


Figure 10: A configuration with four LANs and two bridges.

As an example take the following Confirmation (Figure10). There are 3 LANs: LAN1, LAN2 and LAN3 and two Bridges B1 and B2. B1 is connected to LAN1 and LAN2 and bridges B2 is connected to LAN2 and LAN3. 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. (For example packet from A is going to B. Both are on the same LAN i.e. LAN1).

2) If the destination and source LANs are different, forward the frame. (For example, a packet from A on LAN1 has to go to D on LAN 3)

3) If the destination LAN is unknown, use flooding.

When the bridges are first plugged in, all its hash tables are empty. None of the bridges know where these destination nodes are exactly. Therefore, 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 to the one it arrived on. Gradually, the bridges learn where destinations are. Once the destination is known there is no more flooding and the packet is forwarded on the proper LAN.

The algorithm used by the transparent bridges is called **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 B2 in Figure 10 sees a frame on LAN 3 coming from D, it knows that D must be reachable via LAN 3, so it makes an entry in its hash table noting that frames going to D should use LAN 3.

4.7.3 Spanning Tree Bridges

For reliability, some networks contain more than one bridge, which increases the likelihood of *networking loops.* A networking loop occurs when frames are passed from bridge to bridge in a circular manner, never reaching its destination. To prevent networking loops when multiple bridges are used, the bridges communicate with each other and establish a map of the network to derive what is called a spanning tree for all the networks. A spanning tree consists of a single path between source and destination nodes that does not include any loops. Thus, a spanning tree can be considered to be a loop-free subset of a network's topology. The spanning tree algorithm, specified in IEEE 802.1d, describes how bridges (and switches) can communicate to avoid network loops.

4.7.4 Source Routing Bridges

IBM introduced source routing bridges for use in token ring networks. With source routing, the sending machine is responsible for determining whether, a frame is destined for a node on the same network or on a different network. If, the frame is destined for a different network, then, the source machine designates this by setting the high-order bit of the group address bit of the source address to 1. It also includes in the frame's header the path the frame is to follow from source to destination.
Source routing bridges are based on the assumption that a sending machine will provide routing information for messages destined for different networks. By making the sending machine responsible for this task, a source routing bridge can ignore frames that have not been "marked" and forward only those frames with their high-order destination bit set to 1.

☞ **Check Your Progress 2**

1) What are the features of a transparent bridge?

……

……

……

2) What are the difficulties in building a bridge between the various 802 LANs ?

……

……

……

4.8 SUMMARY

In this unit we discussed two major topics wireless LANs and switching mechanism at the data link layer with IEEE at the data link layer. With IEEE 802.11 standardisation, wireless LANs are becoming common in most of the organisations but, they have their own problems and solutions CSMA/CD does not work due to hidden station problem. To make CSMA work better two new protocols, MACA and MACAW were discussed. The physical layer of wireless LAN standard i.e. IEEE 802.11 allows five different transmission modes, including infrared, various spread spectrum schemes etc. As a part of inter LANs connecting mechanism we discussed different types of bridges. Bluetooth was not taken up in this unit, although, it is a very important topic today. It is a also a wireless network used for connecting handsets and other peripherals to computers without wires.

4.9 SOLUTIONS/ANSWERS

Check Your Progress 1

1) The problem of a station not being able to detect another node for the medium because the competitor is outside its wireless range is called hidden station problem.

2) The crux of the issue is that the CSMA can be applied to wireless environment. Because it simply informs whether there is a transmission activity around the sender node that senses the carrier. Whereas, in a wireless environment, before sending a transmission, a station needs whether there is activity around the receiver or not. With a wire, all signals inform all stations, so, only one transmission can take place at a time, anywhere in the system.

3) It is different in the following ways:

- Addition of a data link layer acknowledgement
- Addition of carrier sensing
- An improved backoff mechanism
- Addition of DS message

Check Your Progress 2

1) It is a plug and play device. There are no additional requirement of Hardware and Software changes, no setting of address switches, no downloading of routing table, in case, it is used to connect LANs. It does not affect the operationof LANs.

2) The following are the difficulties in building a bridge between the various 802 LAN:

i) Different frame formats
ii) Different data rates
iii) Different frame length
iv) Security.

4.10 FURTHER READINGS

1) *Computer Networks*, A. S. *Tanenbaum* 4th Edition, Practice Hall of India, New Delhi. 2003.

2) *Computer Networking*, J.F. Kurose & K.W. Ross, A Top Down Approach Featuring the Internet, Pearson Edition, 2003.

3) *Introduction to Data Communication & Networking,* Behrouz Forouzan, Tata McGraw Hill, 1999.

4) *Communications Networks*, Leon Garcia, and Widjaja, Tata McGraw Hill, 2000.

5) *Data and Computer Communications,* Willian Stallings, 6th Edition, Pearson Education, New Delhi.